GAE-WAM: GEOMETRIC ACTION EFFECT MODELING FOR WORLD ACTION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

World action models (WAMs) use future visual prediction to support robot action learning. Complete future observations contain object motion, backgrounds, and textures, but not all of this information is directly relevant to action selection. How can action prediction emphasize task-relevant information in visual predictions? We propose **Gae-WAM**, a WAM architecture that connects visual and action predictions through action-induced geometric changes in target objects. Modeling these changes as *geometric action effects* brings object-focused information from visual predictions into action-side supervision without substantial additional inference overhead. We find that geometric action effect modeling is effective for robot manipulation. We compare Gae-WAM with other methods on multiple simulation benchmarks and Astribot S1 real-world tasks. Gae-WAM achieves higher aggregate success, with gains under camera-viewpoint and spatial-configuration changes. Shared-representation and training-objective ablations support the contributions of geometric references and cross-branch consistency to manipulation performance.

1 INTRODUCTION

Vision-language-action (VLA) models commonly map visual observations and language instructions to robot actions. World action models (WAMs) add future visual prediction to action learning. This additional supervision helps policy representations capture object motion and scene evolution during manipulation. Fast-WAM uses future-video prediction during training (Yuan et al., 2026b). ImageWAM predicts an endpoint observation through image editing and uses intermediate visual representations for action generation (Zhang et al., 2026a). Future observations, however, include backgrounds, textures, and lighting variations alongside object motion. These elements differ in their relevance to action selection. This raises a question: *How can action prediction emphasize task-relevant information in visual predictions?*

Prior studies highlight the value of object-focused visual information for action learning (Zhang & Ma, 2025; Yu et al., 2026). Object-focused prediction targets and motion-based supervision further emphasize task-relevant visual changes (Zhang & Ma, 2025; Yu et al., 2026; Ji et al., 2026). However, these methods focus on object changes in 2D images and do not explicitly supervise target displacement in 3D. They also do not directly constrain agreement between the target changes implied by visual and action predictions. Action effect modeling offers a way to describe action outcomes through object states and spatial relationships (Guo et al., 2026). For manipulation, changes in object position and extent describe spatial outcomes that can be estimated from observations. We therefore explore target-object geometry as a shared basis for visual and action predictions.

Based on this idea, we define a *geometric action effect* as the geometric change in a target object caused by an action chunk. We introduce **Gae-WAM**, a WAM architecture that uses geometric action effects to guide action learning toward task-relevant information in visual predictions (Figure 1). Using target masks and camera calibration, we derive a shared geometric reference from current and endpoint demonstration observations. The reference combines per-view centroid and projected-area changes with estimated 3D displacement. Reference grounding trains one lightweight auxiliary head to estimate the geometric changes depicted by the visual prediction and the other to estimate those expected from the predicted action chunk. Cross-branch consistency aligns these effect estimates to supervise the action branch with target changes captured by visual predictions. This encourages action learning to draw more on the task-relevant object changes represented in visual predictions.

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.

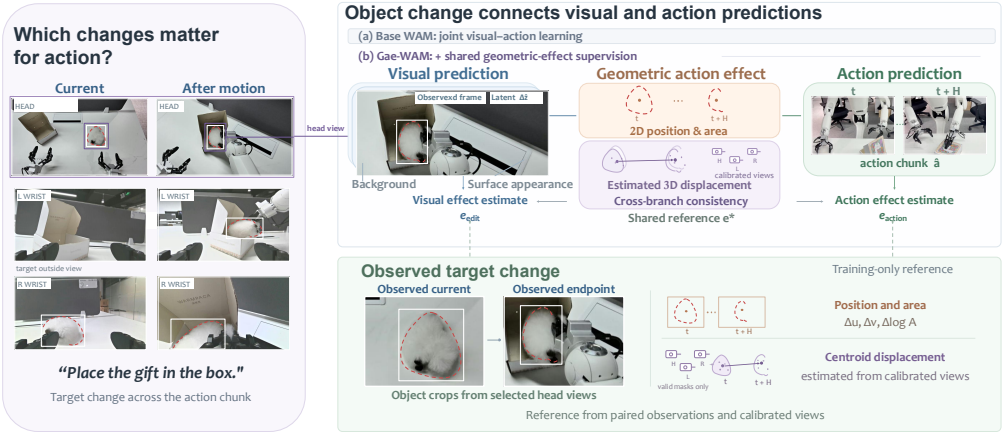


Figure 1: **Geometric action effects relate visual and action predictions through the target object.** The gift is the object acted upon; the box specifies its destination. (a) Base WAM jointly learns visual and action predictions. (b) Gae-WAM estimates the action-induced geometric change from each branch. A reference constructed from observed 2D position and area changes and 3D displacement grounds both estimates, while cross-branch consistency aligns them during training.

All added components are restricted to training, preserving ImageWAM's deployment inputs and action-generation procedure.

Empirically, we find that geometric action effect modeling is effective for robot manipulation. Under matched training and evaluation settings, Gae-WAM achieves higher aggregate success than ImageWAM in simulation and on Astribot S1 real-world tasks. On LIBERO-Plus (Fei et al., 2026), we observe gains under camera-viewpoint and spatial-configuration changes. Ablations further show that reference grounding and cross-branch consistency contribute to the improvements. These results support geometric action effects as a useful training signal for connecting visual prediction with action learning.

Our contributions are as follows:

- We formulate geometric action effect modeling as a shared prediction task. It connects visual and action predictions through action-induced geometric changes in target objects.
- We develop Gae-WAM with observation-derived reference grounding and cross-branch consistency. These objectives encourage greater use of task-relevant visual information in action learning. The method retains the baseline inference procedure.
- We demonstrate the effectiveness of Gae-WAM on simulation benchmarks. Real-world experiments further show improved overall manipulation success on Astribot S1.

2 Related Work

World action models. Vision–language–action (VLA) models combine visual–language understanding and action prediction (Zitkovich et al., 2023; Kim et al., 2025); diffusion and flow matching generate continuous action sequences (Chi et al., 2023; Black et al., 2025b;a). Future visual prediction supports planning for control (Du et al., 2023) and WAM training through joint generation or auxiliary objectives (Zhang et al., 2025; Bi et al., 2026; Li et al., 2026a; Kim et al., 2026; Yuan et al., 2026b). ImageWAM learns current-to-endpoint transformations through image editing, conditioning its action expert on intermediate editing-branch key–value (KV) representations (Zhang et al., 2026a). Gae-WAM augments this joint training with geometric action effect modeling, using action-induced target-object changes to supervise both prediction branches.

Target-object and motion representations. Control-oriented world modeling separates task information from visual distractions (Fu et al., 2021; Wang et al., 2022). Localization and mask prediction

identify manipulated objects for policy learning (Stone et al., 2023; Zhang & Ma, 2025; Yu et al., 2026); point trajectories and 3D tracking supply motion information (Wen et al., 2024; Guan et al., 2026; He et al., 2026; Wang et al., 2026a). Multi-view geometry relates image projections to spatial coordinates (Hartley & Zisserman, 2004), and 3D representations provide policy inputs and geometric supervision (Ze et al., 2024; Zhao et al., 2026). We combine target-object localization with observed geometric changes to construct a reference for the consequences of an action chunk.

Action effects and visual–action alignment. AEM models action outcomes through object-state and spatial-relation cues for procedural mistake detection (Guo et al., 2026). These cues supervise an effect-aware representation of an observed action segment. For robot learning, prior work relates visual and action predictions through representation alignment and latent-space modeling (Qiu et al., 2026; Wang et al., 2026b), action-conditioned prediction, future-prior transfer, and auxiliary motion tasks (Pan et al., 2026; Bai et al., 2026; Li et al., 2026b). Imagine2Act derives 3D object transformations from generated goal images for policy conditioning and aligns predicted end-effector motion with object transformations while the object is grasped (Heng et al., 2026). Gae-WAM combines a target-object focus with action outcome modeling: both branches estimate the geometric action effect of the same chunk. Demonstration observations ground these estimates in a shared reference, and cross-branch consistency aligns them during training.

3 METHOD

3.1 PROBLEM FORMULATION

Given multi-view observations $O_t = \{I_t^{(v)}\}_{v=1}^{V}$, a language instruction ℓ, and robot proprioceptive state q_t at time t, the policy predicts an action chunk spanning the next H control steps. Training uses paired transitions $(O_t, \ell, q_t, \mathbf{a}^*, O^*_{t+H})$, where $\mathbf{a}^* = \mathbf{a}^*_{t:t+H-1} \in \mathbb{R}^{H\times d_a}$ is the demonstrated action chunk and O^*_{t+H} its endpoint observation. ImageWAM jointly predicts actions and endpoint visual states (Zhang et al., 2026a).

Task semantics and offline mask rules specify the target object or object set; each action chunk determines the current and endpoint observations used to measure its change. A *geometric action effect* is the geometric change in the target caused by an action chunk. Current–endpoint demonstration observations provide a reference e^* for two auxiliary estimates: e_{edit} from the visual prediction and e_{action} from the action prediction. Grounding both estimates in the observed change and enforcing their consistency introduces target-object geometry into joint visual–action training.

3.2 GEOMETRIC ACTION EFFECT REFERENCE

For LIBERO, SAM3 (Carion et al., 2026) segments and tracks task-specified targets offline in each camera video. Shared target semantics associate the views; masks are indexed by episode, frame, and camera. RoboTwin uses task-specified simulator actor masks. Appendix C details target selection and mask construction. The reference concatenates per-view 2D changes and world-coordinate 3D displacement:

$$e^* = \left[(e_{\text{2D}}^{*,(1)})^\top, \ldots, (e_{\text{2D}}^{*,(V_{\max})})^\top, (e_{\text{3D}}^{*})^\top\right]^\top, \tag{1}$$

where $V_{\max}$ is the maximum number of camera views and $D_e = 3V_{\max} + 3$.

Observable 2D changes. Let $M_t^{(v)}$ and $M_{t+H}^{(v)}$ be the target's current and endpoint masks in view v. Define

$$m(M) = [\bar{u}(M), \bar{v}(M), \log(A(M) + \varepsilon)]^\top, \tag{2}$$

where $\bar{u}, \bar{v}$ are normalized centroid coordinates, A is the target's image area fraction, and ε ensures numerical stability. The per-view component is

$$e_{\text{2D}}^{*,(v)} = [\Delta u_v, \Delta v_v, \Delta \log A_v]^\top = m(M_{t+H}^{(v)}) - m(M_t^{(v)}). \tag{3}$$

3D spatial reference. Using camera intrinsics and extrinsics $K_\tau^{(v)}, T_\tau^{(v)}$ and multi-view target masks, we construct sparse visual hulls at $\tau \in \{t, t+H\}$. We sample 3D points along rays through

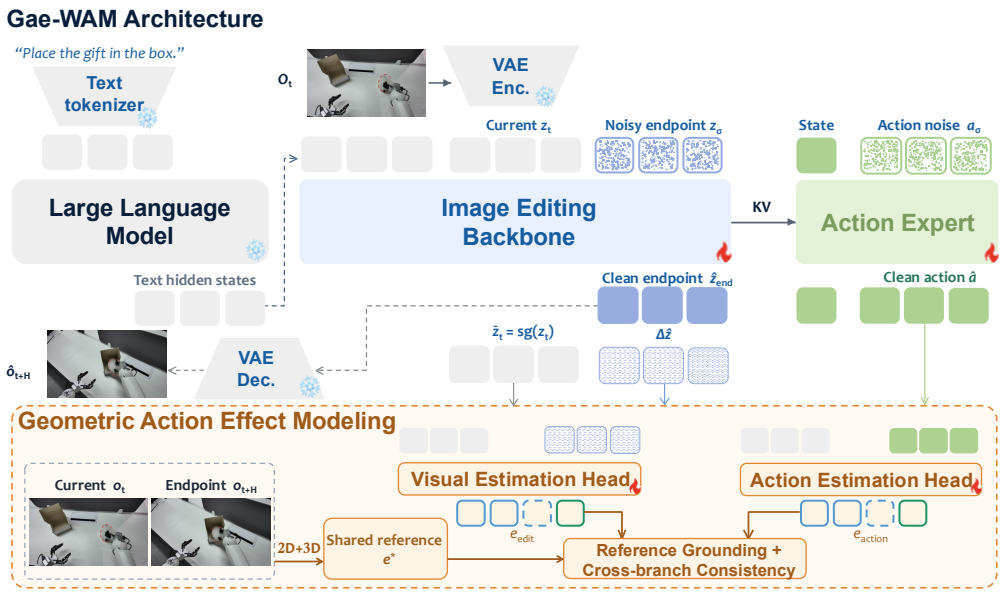


Figure 2: **Gae-WAM architecture.** Two auxiliary heads estimate geometric action effects from predicted editing differences and action chunks, conditioned on the current scene. Current–endpoint target observations provide a shared reference e^* for grounding e_{edit} and e_{action}; cross-branch consistency aligns the estimates. Reference construction and both auxiliary heads are used during training.

the target region and filter them by silhouette support in other valid views, yielding world-coordinate centroid estimates c_t, c_{t+H}. The world-coordinate component is

$$e^*_{3\text{D}} = S_p^{-1}(c_{t+H} - c_t), \tag{4}$$

where $S_p \in \mathbb{R}^{3\times3}$ is a predefined matrix normalizing world-coordinate displacement components. The 2D terms preserve view-specific changes, while the 3D term expresses displacement in a common world frame.

Reference construction also outputs per-dimension validity $\nu^* \in \{0,1\}^{D_e}$ and confidence $\kappa^* \in [0,1]^{D_e}$ to weight supervision. Missing views and invalid 3D estimates receive zero weight.

3.3 Geometric Action Effect Modeling

Gae-WAM retains ImageWAM's image-editing backbone and action expert, adding two lightweight auxiliary heads for geometric action effect modeling (Figure 2).

Image editing and action generation. A frozen VAE encodes multi-view RGB observations concatenated in fixed order into z_t. Language, proprioception, and current image tokens form the conditioning prefix. The action expert reads intermediate key–value representations from the editing branch, as in ImageWAM. From noisy endpoint latents z_{σ_z} and actions $\mathbf{a}_{\sigma_a}$, the branches predict respective flow velocities $\hat{v}_{\text{img}}$ and $\hat{v}_{\text{act}}$, yielding one-step denoised estimates:

$$\hat{z}_{\text{end}} = z_{\sigma_z} - \sigma_z \hat{v}_{\text{img}}, \qquad \hat{\mathbf{a}} = \mathbf{a}_{\sigma_a} - \sigma_a \hat{v}_{\text{act}}. \tag{5}$$

Two estimates of the action outcome. With stop-gradient operator $\text{sg}(\cdot)$, define $\bar{z}_t = \text{sg}(z_t)$ and predicted editing difference $\Delta\hat{z} = \hat{z}_{\text{end}} - \bar{z}_t$. The heads compute

$$\begin{aligned} e_{\text{edit}} &= H_{\text{edit}}(\bar{z}_t, \Delta\hat{z}), \\ e_{\text{action}} &= H_{\text{action}}(\bar{z}_t, \hat{\mathbf{a}}), \end{aligned} \tag{6}$$

where $e_{\text{edit}}, e_{\text{action}} \in \mathbb{R}^{D_e}$ estimate the geometric action effect from complementary predictions. The visual head estimates the target change depicted by the predicted endpoint; the action head

estimates the change that the predicted control sequence would produce in the current scene. The two independently parameterized heads use learnable attention pooling and lightweight multilayer perceptrons. Within these auxiliary paths, gradients through the current latent are stopped; gradients through the endpoint and action estimates carry geometric supervision to their respective branches.

3.4 Grounding and Cross-Branch Consistency

Base WAM objective. ImageWAM's flow-matching objective uses ground-truth target x_0:

$$x_\sigma = (1-\sigma)x_0 + \sigma\epsilon, \qquad v^* = \epsilon - x_0, \tag{7}$$

where ϵ is sampled noise and σ the noise level. The visual and action targets x_0 are the ground-truth endpoint latent and demonstrated action chunk, respectively. Their respective flow-matching losses give $\mathcal{L}_{\text{WAM}} = \lambda_{\text{img}}\mathcal{L}_{\text{img}} + \lambda_{\text{act}}\mathcal{L}_{\text{act}}$.

Geometric action effect supervision. To account for reference reliability and prediction noise, we use per-dimension weights $\omega_{n,d} = \chi_n \nu^*_{n,d}\kappa^*_{n,d}$ for sample n and reference dimension d. Here, χ_n selects one-step denoised estimates for supervision based on both branches' noise levels. These weights define a confidence-weighted distance using the Smooth-L_1 loss ρ_β with parameter β:

$$D_\omega(x, y) = \frac{\sum_{n,d} \omega_{n,d}\rho_\beta(x_{n,d} - y_{n,d})}{\sum_{n,d}\omega_{n,d} + \varepsilon}. \tag{8}$$

Reference grounding supervises both geometric action effect estimates with the demonstration-derived reference e^*. Cross-branch consistency further couples the two prediction paths by comparing their estimated effects for the same input. We use a symmetric stop-gradient formulation so that each consistency term updates one path while treating the other path's estimate as a fixed target:

$$\begin{aligned} \mathcal{L}_{\text{ground}} &= \lambda_e D_\omega(e_{\text{edit}}, e^*) + \lambda_a D_\omega(e_{\text{action}}, e^*), \\ \mathcal{L}_{\text{cons}} &= \frac{1}{2} D_\omega(e_{\text{edit}}, \text{sg}(e_{\text{action}})) + \frac{1}{2} D_\omega(\text{sg}(e_{\text{edit}}), e_{\text{action}}). \end{aligned} \tag{9}$$

We incorporate the geometric losses with an overall weight λ_E and a relative consistency weight λ_c. Their contribution is introduced gradually through $\alpha(k)$, which increases linearly over the initial training steps k:

$$\mathcal{L} = \mathcal{L}_{\text{WAM}} + \lambda_E \alpha(k)\left(\mathcal{L}_{\text{ground}} + \lambda_c \mathcal{L}_{\text{cons}}\right). \tag{10}$$

3.5 Inference

Gae-WAM retains ImageWAM's action generation procedure, conditioned on current multi-view observations, language, and proprioception:

$$\hat{\mathbf{a}}_{t:t+H-1} = \pi_\theta(O_t, \ell, q_t; \epsilon_a), \tag{11}$$

where ϵ_a is initial action-sampling noise. Masks, camera parameters, visual hull and reference construction, and auxiliary heads serve training only.

4 Experiments

We evaluate Gae-WAM on three simulation benchmarks and Astribot S1 real-world tasks.

4.1 Experimental Setup

Benchmarks and evaluation. LIBERO comprises Spatial, Object, Goal, and Long (Liu et al., 2023), with 1,000 rollouts per evaluation seed (250 per suite). LIBERO-Plus provides 10,030 instances across seven perturbation axes: camera viewpoint, robot initial state, language, lighting, background texture, sensor noise, and object layout (Fei et al., 2026). We evaluate each LIBERO-Plus instance once per seed. For both benchmarks, we pool successes across three evaluation seeds of each fixed trained checkpoint. On RoboTwin2.0, we follow the Clean-to-Randomized (C2R) protocol (Chen et al., 2026), training from the shared base initialization on clean demonstrations and evaluating bimanual manipulation in Clean (Easy) and Randomized (Hard) scenes. Real-world evaluation uses Astribot S1 for towel folding, pen storage, gift packing, and basketball placement (Figure 3(b)).



(a) Target-region attention on LIBERO. (b) Astribot S1 real-world tasks.

Figure 3: **Attention comparison and real-world tasks.** (a) ImageWAM and an early Gae-WAM variant share the color scale and Q95 threshold. Cyan boxes mark target regions; green contours show target masks. (b) Towel folding, pen storage, gift packing, and basketball placement, in reading order.

Comparison settings. We compare primarily with ImageWAM retrained with matched demonstrations, initialization, base training budget, and deployment inputs. LIBERO follows ImageWAM's official training configuration; RoboTwin2.0 uses a shared local configuration (Appendix N). Real-world comparisons match initial conditions, success criteria, and time limits. Image, video, and vision–language pretraining are part of model initialization. Our matched runs use no additional robot-policy pretraining. Appendices B, M, and N detail baseline pretraining, training settings, and evaluation protocols. Reported differences use unrounded rates.

Table 1: **LIBERO success rates (%).**

Method	Spatial	Object	Goal	Long	Avg.
$\pi_{0.5}$ (Black et al., 2025a)	98.80	98.20	98.00	92.40	96.90
Motus (Bi et al., 2026)	96.80	99.80	96.60	97.60	97.70
LingBot-VA (Li et al., 2026a)	98.50	99.60	97.20	**98.50**	**98.50**
RynnVLA-002 (Cen et al., 2026)	**99.00**	99.80	96.40	94.40	97.40
Fast-WAM (Yuan et al., 2026b)	98.20	**100.00**	97.00	95.20	97.60
ImageWAM†	97.47	99.60	**98.40**	96.27	97.93
Gae-WAM	98.53	99.73	98.13	96.00	98.10

Table 2: **RoboTwin2.0 C2R (%).**

Method	Clean	Rand.	Avg.
GR00T-N1.7 (NVIDIA, 2026)	43.60	20.70	32.15
$\pi_{0.5}$ (Black et al., 2025a)	73.10	**47.90**	**60.50**
StarVLA (Community, 2026)	58.10	10.60	34.35
AHA-WAM (Cai et al., 2026)	64.30	3.20	33.80
Fast-WAM (Yuan et al., 2026b)	77.80	1.90	39.90
ImageWAM†	80.20	13.32	46.76
Gae-WAM	**81.44**	13.02	47.23

Methods above the separator use additional robot-policy pretraining; those below do not. †: matched retraining. Bold: column maximum. RoboTwin2.0 Avg. weights Clean and Randomized equally. Sources: Appendix M.

4.2 Main Results

LIBERO and LIBERO-Plus. Gae-WAM reaches 98.10% success on standard LIBERO, modestly improving over the already strong matched ImageWAM baseline (Table 1). The largest gain is in Spatial, while Goal and Long success remain slightly lower.

Gae-WAM improves overall LIBERO-Plus success, particularly under spatial perturbations (Table 3). Camera-viewpoint success rises from ImageWAM's 74.94% to 78.82%, the highest among the reported methods. Success increases from 45.25% to 48.47% under robot-initial-state changes and from 79.48% to 80.79% under object-layout changes. ImageWAM remains stronger under

Table 3: **LIBERO-Plus success rates (%).** †: matched retraining. P.T.: additional robot-policy pretraining. Bold: column maximum. Avg. pools successful/total rollouts. Baseline sources are in Appendix M.

Method	P.T.	Cam.	Robot	Lang.	Light	Bg.	Noise	Layout	Avg.
OpenVLA-OFT (Fei et al., 2026)	✓	56.40	31.90	79.50	88.70	93.30	75.80	74.20	69.60
π_0-FAST (Fei et al., 2026)	✓	65.10	21.60	61.00	73.20	73.20	74.40	68.80	61.60
SG-WAM (Zhao et al., 2026)	×	58.60	48.90	81.40	89.80	86.10	80.70	74.20	73.00
Cosmos Policy (Zhang et al., 2026b)	×	75.80	63.30	81.70	96.50	88.90	92.70	82.20	82.20
$\pi_{0.5}$ (Zhang et al., 2026b)	✓	75.40	**77.50**	85.60	**96.90**	**94.60**	89.70	**85.70**	**85.70**
ImageWAM†	×	74.94	45.25	**90.13**	96.64	86.40	95.23	79.48	80.31
Gae-WAM	×	**78.82**	48.47	88.01	96.06	85.75	**95.36**	80.79	81.18

language, lighting, and background perturbations. These spatial gains are consistent with grounding target-object changes in image and world coordinates.

RoboTwin2.0. With clean-only downstream training, Gae-WAM achieves the highest reported Clean success of 81.44% (Table 2). This improvement outweighs slightly lower Randomized success, yielding a higher equally weighted average than matched ImageWAM.

Astribot S1 real-world experiments. Gae-WAM improves overall Astribot S1 success from ImageWAM's 51.67% to 56.67% (Table 4). Gains are largest in basketball placement, followed by gift packing and pen storage, highlighting benefits for object transport and placement. Towel-folding success decreases, indicating task-dependent benefits.

Table 4: **Results on four Astribot S1 tasks.** Success in %; 30 trials per task and method. Overall pools all 120 trials per method: ImageWAM 62 successes; Gae-WAM 68. Δ denotes the difference in percentage points.

Task	Trials per method	ImageWAM	Gae-WAM	Δ
Fold Towel	30	50.00	43.33	-6.67
Pen Storage	30	50.00	53.33	+3.33
Pack Gift	30	46.67	56.67	+10.00
Basketball Placement	30	60.00	73.33	+13.33
Overall	120	51.67	56.67	+5.00

4.3 Analysis

Success by task type. To examine where the gains arise, we group RoboTwin2.0 tasks by manipulation operation and pool Clean and Randomized trials, giving 200 trials per task and method (Appendix N.1). Transport, placement, and storage account for the largest share of additional successes, with a prominent gain on placing an object in a basket and then lifting the basket. Stacking, sorting, and bimanual handovers also improve, while trash dumping shows the largest category-level increase in success rate. Success decreases for articulated-object manipulation and for tool use and alignment. These differences suggest that the usefulness of the current geometric reference depends on the type of interaction.

Visual emphasis and closed-loop behavior. Figure 3(a) compares target-region attention for ImageWAM and an early Gae-WAM variant on identical selected LIBERO inputs. The maps share layers, head aggregation, color scales, and display thresholds. Target-region measurements show greater attention in several selected states, with similar allocation at contact onset (Appendix J). Figure 4 shows matched closed-loop gripper–object interactions and final configurations through contact, transport, and placement.

We further examine the final checkpoints under camera changes while holding the physical state fixed (Appendix K). Gae-WAM exhibits lower mean translation and rotation drift than ImageWAM

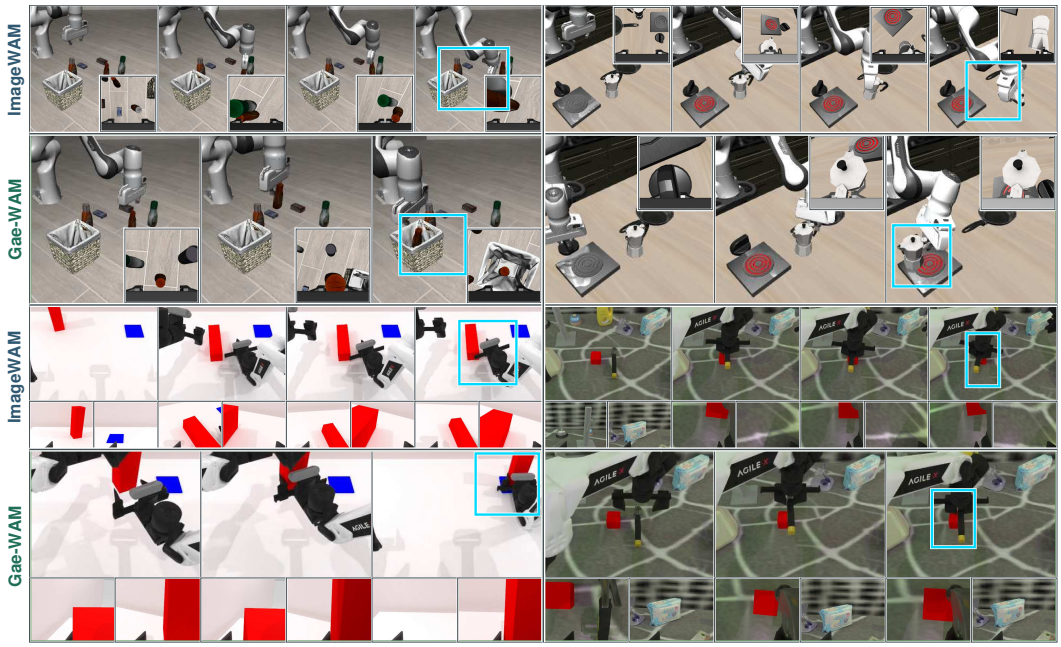


Figure 4: **Closed-loop manipulation under perturbations.** Matched ImageWAM/Gae-WAM keyframes show contact, object manipulation, and final states in task-progress order. Cyan boxes mark gripper–object interactions and final object configurations.

Table 5: **Shared-representation ablation on LIBERO-Plus (%).** Variants are described in Section 4.4. Bold: column maximum.

Variant	Cam.	Robot	Lang.	Light	Bg.	Noise	Layout	Avg.
ImageWAM	74.94	45.25	90.13	96.64	86.40	95.23	79.48	80.31
DirectAlign	76.46	48.43	89.72	96.82	84.05	**95.48**	80.59	80.96
FullAlign	73.98	46.97	**93.30**	**97.75**	87.55	94.34	78.45	80.86
Gae-WAM (2D-only)	74.63	47.66	90.44	95.68	**88.04**	95.02	79.50	80.71
Gae-WAM	**78.82**	**48.47**	88.01	96.06	85.75	95.36	**80.79**	**81.18**

across the sampled states, alongside a comparable scale of state-dependent motion and distance to demonstration commands. This observation complements the closed-loop viewpoint results by showing reduced average sensitivity of continuous motion commands to camera changes.

4.4 Ablation Studies

We examine the choice of shared prediction target, the contribution of 3D displacement, and the roles of reference grounding and cross-branch consistency. All variants use ImageWAM's base model and training settings.

Q1: How does target geometry compare with alternative shared representations? We design DirectAlign and FullAlign as control variants (Appendix D.1). DirectAlign aligns freely learned low-dimensional representations without an observation-derived target. FullAlign uses pooled future VAE latents as the shared target for reference grounding and cross-branch consistency. Gae-WAM instead uses action-induced geometric changes in target objects as the shared prediction target.

Both DirectAlign and FullAlign improve overall success over ImageWAM (Table 5). Gae-WAM achieves the highest overall success among these variants, although the aggregate differences are small. Its clearest advantage is under camera-viewpoint changes, and it also outperforms FullAlign under object-layout changes. FullAlign performs better under language and lighting perturbations. These comparisons show that the choice of shared target affects which perturbations benefit from

Table 6: **Training-objective ablation on LIBERO-Plus (%).** Ground.: shared geometric-reference supervision; Cons.: cross-branch consistency. Bold: column maximum.

Method	Ground.	Cons.	Cam.	Robot	Lang.	Light	Bg.	Noise	Layout	Avg.
ImageWAM	×	×	74.94	45.25	**90.13**	96.64	**86.40**	95.23	79.48	80.31
DirectAlign	×	✓	76.46	48.43	89.72	**96.82**	84.05	**95.48**	80.59	80.96
Grounding-only	✓	×	75.05	48.15	88.85	96.64	85.10	94.84	**81.18**	80.63
Gae-WAM	✓	✓	**78.82**	**48.47**	88.01	96.06	85.75	95.36	80.79	**81.18**

cross-branch training, with target geometry providing the strongest viewpoint results among the evaluated representations.

Q2: What does 3D displacement add beyond per-view 2D changes? The 2D-only variant retains reference grounding and cross-branch consistency but uses only per-view centroid and projected-area changes. Adding estimated 3D displacement improves overall success on both LIBERO-Plus and standard LIBERO (Table 5; Appendix G). The largest improvement on LIBERO-Plus occurs under camera-viewpoint changes, where success rises from 74.63% to 78.82%. Robot-initial-state and object-layout success also improve. World-coordinate displacement provides a common spatial reference for changes projected into different views, and these results support its use alongside per-view 2D geometry.

Q3: Does cross-branch consistency help beyond reference grounding? We compare Gae-WAM with Grounding-only to isolate the contribution of cross-branch consistency while keeping the geometric reference unchanged (Table 6). DirectAlign provides a further comparison using consistency between freely learned representations.

Reference grounding alone improves overall success over ImageWAM, and adding cross-branch consistency yields the highest overall success among the compared variants. The main improvement over Grounding-only occurs under camera-viewpoint changes, while Grounding-only remains slightly stronger under object-layout changes. This comparison supports directly coupling the visual and action estimates even when both are already supervised by the same geometric reference.

4.5 Limitations

The current geometric action effect representation captures projected centroid and area changes together with 3D translation. It provides no explicit supervision for rotation, deformation, contact, or object relations. These properties matter for interactions such as articulated-object manipulation and tool alignment, where Gae-WAM does not improve over ImageWAM. Richer references based on object pose, contact states, and object–receptacle relations could better describe these interactions. Intermediate observations could also capture changes within an action chunk.

Reference construction depends on target segmentation and camera calibration. Occlusion, limited view coverage, and visual-hull discretization can reduce the accuracy of the reconstructed displacement. Temporal fusion and confidence estimates that account for visibility and calibration could improve the reference used by both prediction branches. Automatic target selection could simplify reference construction for new demonstrations.

5 Conclusion

In this paper, we presented Gae-WAM, which connects visual and action predictions through geometric action effects. Grounding and aligning effect estimates brings target changes from visual predictions into action-side supervision. The method improves manipulation performance without substantial additional inference overhead. Experiments demonstrate its effectiveness across simulation benchmarks and Astribot S1 tasks, with clear gains under camera-viewpoint changes on LIBERO-Plus. Ablations further support the contributions of geometric reference grounding and cross-branch consistency. These results highlight the value of target-object geometry in translating visual prediction into useful supervision for action learning.

AI USE STATEMENT

Generative AI tools were used to polish the language of the manuscript.

REPRODUCIBILITY STATEMENT

Source code is provided in the supplementary material. Training and evaluation settings are detailed in Appendices B, N, and O. Appendix C covers target selection, mask construction, geometric references, and training pseudocode. Appendices D and G document ablation variants and results by evaluation seed.

REFERENCES

Yongjie Bai, Hanting Wang, Mingtong Dai, Qijun Zhong, Yang Liu, and Liang Lin. Bridge-WA: Predicting where and how the world changes for robotic action, 2026. URL https://arxiv.org/abs/2607.02195.

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, Hongyan Zhao, Hanyu Liu, Zhizhong Su, Lei Ma, Hang Su, and Jun Zhu. Motus: A unified latent action world model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 35101–35113, June 2026.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, brian ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization. In Joseph Lim, Shuran Song, and Hae-Won Park (eds.), *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 17–40. PMLR, 27–30 Sep 2025a. URL https://proceedings.mlr.press/v305/black25a.html.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, Laura Smith, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A Vision-Language-Action Flow Model for General Robot Control. In *Proceedings of Robotics: Science and Systems*, LosAngeles, CA, USA, June 2025b. doi: 10.15607/RSS.2025.XXI.010.

Jisong Cai, Long Ling, Shiwei Chu, Zhongshan Liu, Jiayue Kang, Zhixuan Liang, Wenjie Xu, Yinan Mao, Weinan Zhang, Xiaokang Yang, Ru Ying, Ran Zheng, and Yao Mu. AHA-WAM:asynchronous horizon-adaptive world-action modeling with observation-guided context routing, 2026. URL https://arxiv.org/abs/2606.09811.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, RISHI HAZRA, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollar, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. SAM 3: Segment anything with concepts. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 138846–138923, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/e0982cbc81401df3430ee1ff780dc7a2-Paper-Conference.pdf.

Jun Cen, Siteng Huang, Yuqian Yuan, Kehan Li, Hangjie Yuan, Chaohui Yu, Bohan Hou, Yuming Jiang, Jiayan Guo, Xin Li, Hao Luo, Fan Wang, Deli Zhao, and Hao Chen. RynnVLA-002: A unified vision-language-action and world model, 2026. URL https://arxiv.org/abs/2511.17502.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, Weiliang Deng, Yubin Guo, Tian Nian, Xuanbing Xie, Qiangyu Chen, Kailun Su, Tianling Xu, Guodong Liu, Mengkang Hu, Huan ang Gao, Kaixuan Wang, Zhixuan Liang, Yusen Qin, Xiaokang Yang, Ping Luo, and Yao Mu. RoboTwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=itonej9GIV.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin CM Burchfiel, and Shuran Song. Diffusion Policy: Visuomotor Policy Learning via Action Diffusion. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023. doi: 10.15607/RSS.2023.XIX.026.

StarVLA Community. StarVLA: A Lego-like codebase for vision-language-action model developing, 2026. URL https://arxiv.org/abs/2604.05014.

Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 9156–9172. Curran Associates, Inc., 2023. doi: 10.52202/075280-0403. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/1d5b9233ad716a43be5c0d3023cb82d0-Paper-Conference.pdf.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 38574–38583, June 2026.

Xiang Fu, Ge Yang, Pulkit Agrawal, and Tommi Jaakkola. Learning task informed abstractions. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 3480–3491. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/fu21b.html.

Jiarui Guan, Wenshuai Zhao, Yue Pei, Ziliang Chen, Arno Solin, and Juho Kannala. Point tracking improves world action models. In *Robot World Models*, 2026. URL https://openreview.net/forum?id=BltZguOz3Z.

Wenliang Guo, Yujiang Pu, and Yu Kong. Procedural mistake detection via action effect modeling. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 43009–43030, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/473ac8cf1fe7059c81af069abb558007-Paper-Conference.pdf.

Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2 edition, 2004.

Yuxin He, Ruihao Zhang, Xianzu Wu, Zhiyuan Zhang, Cheng Ding, and Qiang Nie. 3D dynamics-aware manipulation: Endowing manipulation policies with 3D foresight, 2026. URL https://arxiv.org/abs/2502.10028.

Liang Heng, Jiadong Xu, Yiwen Wang, Xiaoqi Li, Muhe Cai, Yan Shen, Juan Zhu, Guanghui Ren, and Hao Dong. Imagine2Act: Leveraging object-action motion consistency from imagined goals for robotic manipulation, 2026. URL https://arxiv.org/abs/2509.17125.

Haoyuan Ji, Lingxiang Fan, Shang Su, Yinqiao Lu, Mengkai Shi, Jun Gao, and Shuo Feng. Dc-wam: Dynamic-centric visual supervision and reasoning for world-action models, 2026. URL https://arxiv.org/abs/2607.25918.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. In Pulkit Agrawal, Oliver Kroemer, and Wolfram

Burgard (eds.), *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2679–2713. PMLR, 06–09 Nov 2025. URL https://proceedings.mlr.press/v270/kim25c.html.

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 C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 71531–71552, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/748becc400a57c0e31cfe6a2e7951467-Paper-Conference.pdf.

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 *Proceedings of Robotics: Science and Systems*, Sydney, Australia, July 2026a. doi: 10.15607/RSS.2026.XXII.016.

Wei Li, Peijin Jia, Yuan Ma, Xuefeng Jiang, Titong Jiang, Sheng Sun, Yujian Li, Xin Wen, Han Hong, Zhikang Liu, Bailin Li, and Kun Zhan. FoMoVLA: Bridging visual foresight and motion guidance for vision-language-action models, 2026b. URL https://arxiv.org/abs/2607.14739.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 44776–44791. Curran Associates, Inc., 2023. doi: 10.52202/075280-1939. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/8c3c666820ea055a77726d66fc7d447f-Paper-Datasets_and_Benchmarks.pdf.

NVIDIA. GR00T-N1.7-3B: Model card. Official model repository, 2026. URL https://huggingface.co/nvidia/GR00T-N1.7-3B. Accessed September 24, 2026.

OpenWAM-Official. OpenWAM: RoboTwin benchmark results. Official project repository, 2026. URL https://github.com/OpenWAM-Official/OpenWAM/blob/main/benchmarks/robotwin/README.md. Accessed September 24, 2026.

Bikang Pan, Fan Liu, Haotao Lu, Jingya Wang, and Ye Shi. SelfWAM: A self-grounded unified world action model for fast robot control, 2026. URL https://arxiv.org/abs/2608.00725.

Lu Qiu, Yizhuo Li, Yi Chen, Yuying Ge, Yixiao Ge, and Xihui Liu. Making foresight actionable: Repurposing representation alignment in world action models, 2026. URL https://arxiv.org/abs/2606.12217.

Austin Stone, Ted Xiao, Yao Lu, Keerthana Gopalakrishnan, Kuang-Huei Lee, Quan Vuong, Paul Wohlhart, Sean Kirmani, Brianna Zitkovich, Fei Xia, Chelsea Finn, and Karol Hausman. Open-world object manipulation using pre-trained vision-language models. In Jie Tan, Marc Toussaint, and Kourosh Darvish (eds.), *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 3397–3417. PMLR, 06–09 Nov 2023. URL https://proceedings.mlr.press/v229/stone23a.html.

Chenyi Wang, Xinkai Wang, Bokai Lin, Jialin Tian, Fucheng Zhang, Cewu Lu, and Lixin Yang. Track4Action: Distilling world-centric 3D tracker into vision-language-action policies, 2026a. URL https://arxiv.org/abs/2608.03727.

Junke Wang, Qihang Zhang, Shuai Yang, Yiming Luo, Yujun Shen, Zuxuan Wu, Yu-Gang Jiang, and Yinghao Xu. RepWAM: World action modeling with representation visual-action tokenizers, 2026b. URL https://arxiv.org/abs/2606.13674.

Tongzhou Wang, Simon Du, Antonio Torralba, Phillip Isola, Amy Zhang, and Yuandong Tian. Denoised MDPs: Learning world models better than the world itself. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 22591–22612. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/wang22c.html.

Chuan Wen, Xingyu Lin, John Ian Reyes So, Kai Chen, Qi Dou, Yang Gao, and Pieter Abbeel. Any-point Trajectory Modeling for Policy Learning. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. doi: 10.15607/RSS.2024.XX.092.

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, 2026. URL https://arxiv.org/abs/2606.13515.

Haoqi Yuan, Zhixuan Liang, Anzhe Chen, Ye Wang, Haoyang Li, Pei Lin, Yiyang Huang, Zixing Lei, Tong Zhang, Jiazhao Zhang, Jie Zhang, Jingyang Fan, Gengze Zhou, Qihang Peng, Chenxu Lv, Xiaoyue Chen, An Yang, Fei Huang, Junyang Lin, Dayiheng Liu, Jingren Zhou, Chenfei Wu, and Xiong-Hui Chen. Qwen-RobotManip technical report: Alignment unlocks scale for robotic manipulation foundation models, 2026a. URL https://arxiv.org/abs/2606.17846.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-WAM: Do world action models need test-time future imagination?, 2026b. URL https://arxiv.org/abs/2603.16666.

Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3D Diffusion Policy: Generalizable Visuomotor Policy Learning via Simple 3D Representations. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. doi: 10.15607/RSS.2024.XX.067.

Junbo Zhang and Kaisheng Ma. Mask2Act: Predictive multi-object tracking as video pre-training for robot manipulation. In *36th British Machine Vision Conference 2025, BMVC 2025, Sheffield, UK, November 24-27, 2025*. BMVA, 2025. URL https://bmva-archive.org.uk/bmvc/2025/assets/papers/Paper_124/paper.pdf.

Wenyao Zhang, Hongsi Liu, Zekun Qi, Yunnan Wang, XinQiang Yu, Jiazhao Zhang, Runpei Dong, Jiawei He, He Wang, Zhizheng Zhang, Li Yi, Wenjun Zeng, and Xin Jin. DreamVLA: A vision-language-action model dreamed with comprehensive world knowledge. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 24195–24228. Curran Associates, Inc., 2025. doi: 10.52202/085713-0818. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/22d4f952efa13970f0b1ffb22170d416-Paper-Conference.pdf.

Yuyang Zhang, Wenyao Zhang, Zekun Qi, He Zhang, Haitao Lin, Jingbo Zhang, Yao Mu, Xiaokang Yang, Wenjun Zeng, and Xin Jin. ImageWAM: Do world action models really need video generation, or just image editing?, 2026a. URL https://arxiv.org/abs/2606.19531.

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, 2026b. URL https://arxiv.org/abs/2603.22078.

Ruiteng Zhao, Zhengshen Zhang, Yue Su, Wenshuo Wang, Jiahui Li, Zhiyuan Yang, Francis E. H. Tay, Marcelo H. Ang, Jr., and Haiyue Zhu. SG-WAM: Self-guided world modeling in geometry-aware policy space, 2026. URL https://arxiv.org/abs/2608.01397.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. RT-2: Vision-language-action models transfer web knowledge to robotic control. In Jie Tan, Marc Toussaint, and Kourosh Darvish (eds.), *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2165–2183. PMLR, 06–09 Nov 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.

A Overview of the Supplementary Material

This supplement details reference construction, training and evaluation, the contributions of geometric grounding and cross-branch consistency, and policy behavior under controlled changes in observation. LIBERO and LIBERO-Plus results use evaluation seeds 42, 43, and 44 with checkpoints trained using seed 42. The RoboTwin2.0 comparison uses training seed 42, evaluation seed 42, and final step-30,000 checkpoints. The five-method reference-source comparison uses evaluation seed 42, and the fixed-state camera probe uses a single training and inference seed. Tables identify the evaluation set and aggregation used in each comparison.

B LIBERO and LIBERO-Plus Protocol

Training configuration. The LIBERO experiments follow the official ImageWAM FLUX.2 Klein 4B training configuration (Zhang et al., 2026a). Our retrained ImageWAM and Gae-WAM share initialization and base optimization, with the following settings: two 224×224 RGB views concatenated in a fixed order, current–endpoint observation pairs with 16-step action chunks (7 dimensions per step), a global batch size of 80, 8 GPUs, 10 epochs, 34,720 optimizer steps, AdamW, a learning rate of 10^{-4}, a cosine scheduler, weight decay of 10^{-2}, bf16 precision, gradient clipping at 1.0, training seed 42, and image/action loss weights of 0.5/1.0. The LIBERO Qwen data, model, and cache configurations all use a context length of 512. RoboTwin2.0 uses the separate configuration in Appendix N.

Gae-WAM adds the 2D+3D reference sidecar and training-only geometric action effect predictors and losses. Geometric labels are transformed together with spatial RGB augmentations to preserve their correspondence in Gae-WAM training. Deployment uses ImageWAM's original observation inputs and action-generation modules.

Evaluation protocol. The main LIBERO and LIBERO-Plus results and ablations use final checkpoints trained with seed 42. Each checkpoint is evaluated with seeds 42, 43, and 44; the same trained weights are reused across these evaluations. Reported aggregate success rates pool successful and total rollouts over the three evaluation seeds. LIBERO uses 1,000 rollouts per seed (250 per suite), and LIBERO-Plus uses one rollout for each of 10,030 instances per seed. Appendix G gives per-seed results, and Appendix H reports all six methods by suite and perturbation difficulty. Appendix F reports the grounding ablation and its breakdown by task suite.

LIBERO and LIBERO-Plus inference uses a 512-token Qwen context, a 16-step action horizon, replanning every 12 steps, 10 denoising steps, and a 30-step wait after reset. Gripper commands are binarized. For Grounding-only on LIBERO-Plus with evaluation seed 44, an environment-constructor repair retained 9,534 completed rollouts and enabled evaluation of the remaining 496 instances. Environment construction used up to eight sampling attempts, followed by one policy rollout per instance.

C Geometric Reference Construction and Training

C.1 Target Selection and Mask Construction

Targets and temporal windows. Task semantics and offline selection rules specify a target object or object set for each demonstration. The selected targets are shared across its training chunks, with multiple targets combined by pixelwise mask union. A chunk $\mathbf{a}_{t:t+15}$ uses observations and masks at t and $t + 16$ to construct the reference. At episode boundaries, cached indices are clipped to the final frame; training masks out padded endpoints in the geometric loss.

LIBERO masks. A task-specific prompt map supplies target descriptions, text aliases, and instance-selection rules for each camera. SAM3 (Carion et al., 2026) processes each camera video offline and associates instances across frames through video tracking IDs. Track selection uses temporal coverage, segmentation confidence, continuity, and image motion; some prompt entries select tracks by maximum motion or area. Selected instance masks are merged for each frame. Across cameras, correspondence follows the shared target semantics and view-specific prompts. Masks, observations,

and camera geometry are indexed by episode, frame, and camera order. The final cache covers 1,712 demonstrations and 277,713 frames from 40 tasks, with two 224×224 mask views.

RoboTwin masks. A task-to-actor mapping selects scene actors and articulated links. Simulator instance segmentation provides their binary masks, with multiple targets merged into a union. The same actor set identifies the targets across cameras. Masks and camera geometry are exported during demonstration replay and associated with training RGB by frame index; sampled RGB frames are checked for alignment. The reference is reconstructed from these masks and camera parameters.

C.2 Geometric Reference Construction and Availability

Geometric reference and validity. The reference combines per-view centroid and projected-area changes with 3D displacement, giving $D_e = 3V_{\max} + 3$ dimensions. At each endpoint, sparse visual-hull reconstruction filters sampled points by multi-view silhouette support and parallax. Valid reconstructions contain at least two points, with at most eight retained, and use confidence-weighted centroids. Their displacement is normalized by $0.20\,\mathrm{m}$ per axis. Each reference dimension carries validity ν^* and confidence κ^*. LIBERO additionally checks agreement between changes in projected reconstructed centroids and mask-centroid changes, using an error scale of 0.2 in normalized image coordinates, a confidence threshold of 0.1, and at least two views. RoboTwin uses the base reconstruction-validity checks. Valid dimensions contribute to the auxiliary objectives, while all training windows retain image and action supervision under the original padding rules.

Offline 3D reference availability. Task-balanced audits sample 4,096 windows from each simulation dataset and evaluate reference validity before random training augmentation and noise gating (Table 7). The LIBERO audit uses seed 20260822 and evaluates clipped terminal endpoints directly. The RoboTwin audit uses seed 42, applies endpoint-padding validity checks, and samples the prepared 2,500-episode collection before its training–validation split. The reported rates describe reference availability in these audited windows.

Table 7: **Offline 3D reference availability in simulation.** Counts are valid reconstructed references among sampled windows.

Dataset / suite	Sampled windows	Valid 3D	Availability (%)
LIBERO Spatial	1,020	152	14.90
LIBERO Object	1,020	389	38.14
LIBERO Goal	1,026	284	27.68
LIBERO Long	1,030	94	9.13
LIBERO overall	4,096	919	22.44
RoboTwin2.0	4,096	3,085	75.32

Real-world reference caches. Astribot targets are specified per task: the basketball, towel, union of pens and brushes, and gift. Camera calibration is obtained through the Astribot S1 SDK. We combine this calibration with target masks, recorded joint states, and forward kinematics to construct 3D references in a base-fixed frame. Table 8 reports validity over all training-window starts in each cache, using endpoint $\min(t + 16, T - 1)$ for an episode of length T, before training-time gating.

Table 8: **Candidate 3D reference availability in Astribot training caches.** Each row corresponds to a separately trained task model.

Task	Training starts	Valid candidate 3D	Availability (%)
Basketball placement	182,952	12,119	6.62
Towel folding	266,022	59,699	22.44
Pen storage	301,732	55,150	18.28
Gift packing	335,836	38,573	11.49

C.3 Training Procedure

Algorithm 1 uses references (e^*, ν^*, κ^*) constructed from offline masks and camera geometry or loaded from reference caches. The noise gate follows both branches' noise levels and the configured admissible range. Samples with invalid reference dimensions still contribute to the WAM losses.

Algorithm 1 Gae-WAM training with geometric action effect supervision.

Inputs: Demonstrations with references (e^*, ν^*, κ^*); trainable WAM parameters θ; head parameters ϕ_e, ϕ_a; frozen VAE encoder E; loss weights and warmup schedule $\alpha(k)$.

for optimization step $k = 1, \dots, K$ **do**
  Sample a batch $(O_t, \ell, q_t, \mathbf{a}^*, O^*_{t+H}, e^*, \nu^*, \kappa^*)$.
  Apply configured augmentation and update the affected reference components and weights.
  $z_t \leftarrow E(O_t)$; $z^*_{\text{end}} \leftarrow E(O^*_{t+H})$.
  Sample $\sigma_z, \sigma_a, \epsilon_z, \epsilon_a$ following the benchmark's training configuration.
  $z_{\sigma_z} \leftarrow (1-\sigma_z) z^*_{\text{end}} + \sigma_z \epsilon_z$; $\mathbf{a}_{\sigma_a} \leftarrow (1-\sigma_a)\mathbf{a}^* + \sigma_a \epsilon_a$.
  Predict $\hat{v}_{\text{img}}, \hat{v}_{\text{act}}$ from the noisy inputs, conditioned on $z_t, \ell, q_t, \sigma_z, \sigma_a$.
  Compute $\mathcal{L}_{\text{img}}$ and $\mathcal{L}_{\text{act}}$ with targets $\epsilon_z - z^*_{\text{end}}$ and $\epsilon_a - \mathbf{a}^*$.
  $\hat{z}_{\text{end}} \leftarrow z_{\sigma_z} - \sigma_z \hat{v}_{\text{img}}$; $\hat{\mathbf{a}} \leftarrow \mathbf{a}_{\sigma_a} - \sigma_a \hat{v}_{\text{act}}$.
  $\bar{z}_t \leftarrow \text{sg}(z_t)$; $\Delta\hat{z} \leftarrow \hat{z}_{\text{end}} - \bar{z}_t$.
  $e_{\text{edit}} \leftarrow H_{\text{edit},\phi_e}(\bar{z}_t, \Delta\hat{z})$; $e_{\text{action}} \leftarrow H_{\text{action},\phi_a}(\bar{z}_t, \hat{\mathbf{a}})$.
  $\chi_n \leftarrow \text{NoiseGate}(\sigma_{z,n}, \sigma_{a,n})$; $\omega_{n,d} \leftarrow \chi_n \nu^*_{n,d} \kappa^*_{n,d}$.
  Define $D_\omega(x, y) = \dfrac{\sum_{n,d} \omega_{n,d} \rho_\beta(x_{n,d} - y_{n,d})}{\sum_{n,d} \omega_{n,d} + \varepsilon}$.
  $\mathcal{L}_{\text{ground}} \leftarrow \lambda_e D_\omega(e_{\text{edit}}, e^*) + \lambda_a D_\omega(e_{\text{action}}, e^*)$.
  $\mathcal{L}_{\text{cons}} \leftarrow \frac{1}{2} D_\omega(e_{\text{edit}}, \text{sg}(e_{\text{action}})) + \frac{1}{2} D_\omega(\text{sg}(e_{\text{edit}}), e_{\text{action}})$.
  $\mathcal{L}_{\text{WAM}} \leftarrow \lambda_{\text{img}} \mathcal{L}_{\text{img}} + \lambda_{\text{act}} \mathcal{L}_{\text{act}}$.
  $\mathcal{L} \leftarrow \mathcal{L}_{\text{WAM}} + \lambda_E \alpha(k)(\mathcal{L}_{\text{ground}} + \lambda_c \mathcal{L}_{\text{cons}})$.
  Update θ, ϕ_e, ϕ_a jointly using $\nabla\mathcal{L}$.
end for

Deployment: Retain the trained WAM and follow ImageWAM's action-generation procedure.

Here, sg stops gradients and ρ_β denotes Smooth-L_1. Only the auxiliary current context is detached; endpoint and action predictions retain gradients. The symmetric terms stop gradients through their respective targets, and $\alpha(k)$ increases linearly during warmup.

D Ablation Variants and Reference Sources

D.1 Definition of the Ablation Variants

The ablations vary the shared prediction target or the auxiliary training objective. All variants retain ImageWAM's base image and action losses and follow the LIBERO training and evaluation protocol in Appendix B. The auxiliary components are used only during training; deployment follows ImageWAM's action-generation procedure.

Gae-WAM. The full method uses an observation-derived reference that combines per-view target centroid and projected-area changes with estimated world-coordinate 3D displacement. Two auxiliary heads estimate geometric action effects from the predicted visual change and action chunk, conditioned on the current visual context. Reference grounding supervises both estimates, and cross-branch consistency aligns them. The following variants modify this shared target or the objectives applied to the two branches.

DirectAlign. DirectAlign aligns freely learned low-dimensional representations computed from the visual and action predictions. Its auxiliary objective encourages agreement between the two representations without supervising them with an observation-derived reference. It therefore provides a comparison for cross-branch alignment without a prescribed geometric or future-observation target.

FullAlign. FullAlign replaces the geometric reference with the pooled VAE latent of the ground-truth endpoint observation. The visual estimate is obtained by directly pooling the predicted clean

Table 9: **Shared supervision and 3D reference source (%).** Final step-34,720 checkpoints with training seed 42 and evaluation seed 42. Standard LIBERO uses 1,000 rollouts; LIBERO-Plus uses 10,030. Rates pool successful/total rollouts; bold marks the column maximum.

Method	Shared target	Standard LIBERO	LIBERO-Plus
ImageWAM	Base image–action objective	98.00	80.54
DirectAlign	Learned low-dimensional features	97.20	80.76
2D-only	Per-view 2D object changes	**98.30**	80.41
GT-3D	2D + simulator-derived 3D	97.80	**81.65**
Gae-WAM	2D + reconstructed 3D	98.20	81.28

endpoint latent. A separate predictor estimates the same target from the current latent and predicted action chunk. Both estimates receive reference supervision and cross-branch consistency. This variant compares geometric changes with a full future-observation representation as the shared target; Appendix E gives the pooling and predictor details.

Gae-WAM (2D-only). This variant retains reference grounding and cross-branch consistency, with both objectives applied only to per-view centroid and projected-area changes. World-coordinate 3D displacement is excluded from the shared target. The comparison with the full method examines the contribution of 3D displacement beyond per-view 2D geometry. We abbreviate this variant as 2D-only in the supplementary tables.

Grounding-only. Grounding-only retains the two geometric action effect predictors and the full observation-derived 2D+3D reference. Both predictions are supervised by reference grounding, while cross-branch consistency is disabled by setting $\lambda_c = 0$. The comparison with Gae-WAM isolates the contribution of directly aligning the two estimates in addition to grounding them in a common reference. Appendix F reports the detailed results.

D.2 Comparison of Geometric Reference Sources

Five-method comparison. We compare ImageWAM, DirectAlign, 2D-only, GT-3D, and Gae-WAM to examine how shared supervision and the source of the 3D reference affect manipulation performance. ImageWAM provides the base image–action training objective; DirectAlign couples learned low-dimensional predictions; 2D-only grounds both branches in target projection changes. GT-3D and Gae-WAM additionally supervise displacement in world coordinates, using simulator-derived and observation-reconstructed references, respectively. Their comparison examines how the source of spatial supervision affects the benefit of the shared geometric action effect representation.

Construction of the GT-3D reference. GT-3D retains the 2D reference, architecture, losses, training budget, and inference procedure of Gae-WAM. Its 3D target is the target object's simulator-derived displacement over the same action-chunk horizon and in the same world coordinate frame:

$$p_{t+H}^{\mathrm{GT}} - p_t^{\mathrm{GT}}, \qquad \text{with the observation-derived counterpart} \quad c_{t+H}^{\mathrm{recon}} - c_t^{\mathrm{recon}}.$$

The simulator coordinates are recovered by recreating the recorded states and checking sampled target-object mappings. These coordinates provide auxiliary training targets. At deployment, both variants generate actions from the same visual, language, and proprioceptive inputs using ImageWAM's action-generation procedure.

Geometric supervision and task performance. On LIBERO-Plus with evaluation seed 42, Gae-WAM achieves 81.28% success, compared with 80.54% for ImageWAM, 80.76% for DirectAlign, and 80.41% for 2D-only. GT-3D reaches 81.65% ($8{,}189/10{,}030$ successful rollouts), 0.37 percentage points above Gae-WAM. Both 2D+3D references improve success over these controls in this evaluation, supporting world-coordinate displacement as a useful component of geometric action effect supervision. The Gae-WAM result further shows that reconstructing this reference from observations provides a useful geometric training signal. On Standard LIBERO, 2D-only achieves 98.30%, Gae-WAM 98.20%, and GT-3D 97.80%; the table reports all five methods under the same evaluation seed.

Table 10: **Contribution of cross-branch consistency on LIBERO-Plus (%).** Pooled results over evaluation seeds 42, 43, and 44, with training seed 42. Δ is Gae-WAM minus Grounding-only, in percentage points.

Perturbation	Grounding-only	Gae-WAM	Δ
Camera	75.05	78.82	+3.77
Robot	48.15	48.47	+0.32
Language	88.85	88.01	-0.85
Light	96.64	96.06	-0.58
Background	85.10	85.75	+0.65
Noise	94.84	95.36	+0.52
Layout	81.18	80.79	-0.39
Overall	80.63	81.18	+0.55

Choice of the reference source. We adopt observation-derived displacement to construct geometric action effect supervision from recorded manipulation demonstrations. GT-3D accesses target-object positions through the simulator's state interface. Obtaining an equivalent direct reference for physical demonstrations requires measurements or annotations of the object's world-coordinate positions at the two endpoints. Gae-WAM estimates this displacement from current–endpoint target masks and camera calibration, using the same observations that supply its per-view 2D reference. This provides a common procedure for constructing the full 2D+3D training target from multi-view demonstrations in simulation and on real robots, with dataset-specific camera geometry described in Appendix C. Validity indicators exclude unavailable geometric coordinates, and confidence weights control their contribution to supervision. Our choice combines the observed benefit of spatial supervision with a reference-construction procedure that can be applied across these demonstration sources.

E Detailed Definition of FullAlign

FullAlign is a control variant designed in this work. Its shared target is the latent representation of the ground-truth future image encoded by the frozen VAE and compressed by adaptive average pooling to an $8 \times 16 \times 128$ grid with 16,384 dimensions. The visual branch directly pools the predicted clean endpoint latent. The action branch uses a state predictor with approximately 2.18M parameters to estimate the future grid from the current latent and predicted actions. FullAlign retains reference supervision on both branches and cross-branch consistency, allowing us to compare two concrete auxiliary-supervision designs: a full future-observation representation and a low-dimensional geometric action effect representation.

Across evaluation seeds 42, 43, and 44, FullAlign achieves 80.86% on LIBERO-Plus and Gae-WAM achieves 81.18%, using the same aggregation as the main shared-representation comparison. Gae-WAM improves camera-view success by 4.84 percentage points and object-layout success by 2.34 points. FullAlign achieves higher success for language and lighting perturbations. The two designs pair their shared targets with corresponding auxiliary predictors and loss configurations.

F Contributions of Grounding and Cross-Branch Consistency

Grounding-only supervises both geometric action effect predictors with the full 2D+3D reference. Gae-WAM adds cross-branch consistency between the two predictions. Both use final checkpoints trained with seed 42, evaluated with seeds 42, 43, and 44.

Performance across evaluation seeds. The complete objective achieves 81.18% pooled success on LIBERO-Plus, compared with 80.63% for Grounding-only, a gain of 0.55 percentage points. The per-seed gains are 0.88, 0.38, and 0.39 points for evaluation seeds 42, 43, and 44, respectively. These positive gains across three evaluations support coupling the prediction branches in addition to supervising each branch with the common geometric reference.

Table 11: **Cross-branch consistency: overall, camera-view, and task-suite results.** All rows pool LIBERO-Plus evaluations with seeds 42, 43, and 44. Differences are Gae-WAM minus Grounding-only in percentage points.

Evaluation subset	Δ
Overall	$+0.548$
Camera perturbations	$+3.773$
Spatial-suite tasks	$+1.915$
Object-suite tasks	-2.608
Goal-suite tasks	$+3.036$
Long-suite tasks	-0.159

Where consistency contributes. The largest gain occurs under camera-view perturbations: success increases from 75.05% to 78.82%. Success also increases under robot-initial-state, background, and sensor-noise changes. Grounding-only retains higher success under language, lighting, and object-layout perturbations. Thus, cross-branch consistency improves aggregate performance with its strongest observed contribution under changes in viewpoint.

Results by original task suite. Table 11 reports the Gae-WAM-minus-Grounding-only differences after pooling successful and total rollouts across evaluation seeds 42, 43, and 44. Suite rows group LIBERO-Plus perturbation instances by their original LIBERO suite. The pooled gain combines higher success on Spatial- and Goal-suite tasks with lower success on Object- and Long-suite tasks.

G Main Results by Evaluation Seed

Gae-WAM improves LIBERO-Plus success over ImageWAM by 0.74, 1.33, and 0.56 percentage points for evaluation seeds 42, 43, and 44, respectively. Each method uses the same trained checkpoint across these evaluations.

Table 12: **Six-method results by evaluation seed (%).** All methods use training seed 42 and the final step-34,720 checkpoint. Each seed contributes 1,000 LIBERO and 10,030 LIBERO-Plus rollouts per method.

	LIBERO			LIBERO-Plus		
Method	42	43	44	42	43	44
ImageWAM	98.00	97.70	98.10	80.54	79.99	80.40
DirectAlign	97.20	98.20	97.10	80.76	81.03	81.09
FullAlign	97.60	98.10	97.30	81.01	81.25	80.32
2D-only	98.30	97.00	97.30	80.41	80.83	80.91
Grounding-only	98.00	98.40	98.10	80.40	80.94	80.57
Gae-WAM	98.20	97.60	98.50	81.28	81.32	80.96

Pooling evaluation seeds 42, 43, and 44, 2D-only achieves $2{,}926/3{,}000$ successes (97.53%) on Standard LIBERO, compared with $2{,}938/3{,}000$ (97.93%) for ImageWAM and $2{,}943/3{,}000$ (98.10%) for Gae-WAM. Adding the observation-derived 3D reference therefore yields 17 additional successes over 2D-only ($+0.57$ percentage points). The per-seed Gae-WAM-minus-2D-only differences are -0.10, $+0.60$, and $+1.20$ points, respectively.

H Complete Six-Method Results

We report six methods using their final step-34,720 checkpoints trained with seed 42. Each checkpoint is evaluated with seeds 42, 43, and 44. The tables below pool successful and total rollouts across these evaluations. The accompanying CSV files contain per-seed totals, suite and perturbation breakdowns, the 40 original tasks, and perturbation-axis cross-tabulations.

Table 13: **Standard LIBERO by suite.** Success rates (%) pool evaluation seeds 42, 43, and 44. Bold marks each column maximum.

Method	Spatial	Object	Goal	Long	Overall
Trials per method	750	750	750	750	3,000
ImageWAM	97.47	99.60	98.40	96.27	97.93
DirectAlign	98.00	99.60	97.07	95.33	97.50
FullAlign	96.67	**99.73**	98.40	95.87	97.67
2D-only	97.20	99.60	97.33	96.00	97.53
Grounding-only	97.60	99.60	**99.07**	**96.40**	**98.17**
Gae-WAM	**98.53**	**99.73**	98.13	96.00	98.10

On Standard LIBERO, Grounding-only achieves 98.17% success ($2{,}945/3{,}000$), followed by Gae-WAM at 98.10% ($2{,}943/3{,}000$), a difference of 0.07 percentage points. Gae-WAM achieves the highest Spatial success among the six methods and shares the highest Object success with FullAlign.

Table 14: **LIBERO-Plus by original task suite.** Success rates (%) pool evaluation seeds 42, 43, and 44. Bold marks each column maximum.

Method	Spatial	Object	Goal	Long	Overall
Trials per method	7,206	7,554	7,773	7,557	30,090
ImageWAM	86.34	83.77	74.09	77.49	80.31
DirectAlign	86.48	84.01	**77.61**	76.08	80.96
FullAlign	85.94	85.28	75.56	77.04	80.86
2D-only	85.25	84.83	75.57	**77.57**	80.71
Grounding-only	86.15	**85.88**	74.10	76.86	80.63
Gae-WAM	**88.07**	83.27	77.14	76.70	**81.18**

Gae-WAM has the highest pooled LIBERO-Plus success among the six methods at 81.18%. Relative to ImageWAM, success increases by 1.72 percentage points on Spatial and 3.05 points on Goal. Grounding-only leads on Object, DirectAlign on Goal, and 2D-only on Long.

Table 15: **LIBERO-Plus by annotated difficulty.** Success rates (%) pool evaluation seeds 42, 43, and 44. Bold marks each column maximum.

Method	Level 1	Level 2	Level 3	Level 4	Level 5	Unlabelled
Trials per method	4,932	6,606	6,282	5,658	6,249	363
ImageWAM	92.94	91.66	82.28	77.31	58.36	92.56
DirectAlign	92.94	**92.14**	84.78	78.30	57.15	**99.72**
FullAlign	92.62	91.52	84.02	77.80	58.81	99.45
2D-only	92.96	91.72	83.57	78.05	58.12	95.04
Grounding-only	**93.11**	91.42	84.15	77.06	58.07	98.35
Gae-WAM	92.46	91.95	**84.80**	**78.33**	**58.86**	98.35

The difficulty groups follow the benchmark annotations. Gae-WAM achieves the highest success at levels 3, 4, and 5. Among the five labelled difficulty levels, its largest gain over ImageWAM occurs at level 3, increasing from 82.28% to 84.80% ($+2.52$ percentage points). Level 4 improves by 1.03 points and level 5 by 0.50 points.

I LIBERO-Plus Perturbation Breakdown

Table 16 reports the seven-axis differences between Gae-WAM and each of the five controls. Positive values indicate higher success for Gae-WAM. Rates and differences are computed from successful and total rollouts, then rounded to two decimal places. Gae-WAM achieves higher camera-view

Table 16: **LIBERO-Plus success-rate differences by perturbation axis.** Values are Gae-WAM minus each control in percentage points, pooling evaluation seeds 42, 43, and 44.

Axis	ImageWAM	DirectAlign	FullAlign	2D-only	Grounding-only
Background	-0.65	+1.70	-1.80	-2.29	+0.65
Camera	+3.88	+2.36	+4.84	+4.19	+3.77
Language	-2.13	-1.71	-5.29	-2.43	-0.85
Light	-0.58	-0.76	-1.69	+0.38	-0.58
Layout	+1.31	+0.20	+2.34	+1.29	-0.39
Robot	+3.23	+0.04	+1.51	+0.82	+0.32
Noise	+0.12	-0.12	+1.02	+0.33	+0.52

success than all five controls, with gains ranging from 2.36 to 4.84 percentage points. Robot-initial-state success also increases across all five comparisons. Gae-WAM and DirectAlign achieve similar success under robot-initial-state perturbations, while their largest difference favors Gae-WAM under viewpoint changes. Across these comparisons, the complete geometric action effect objective yields its largest observed advantage under viewpoint perturbations.

J Attention Visualization

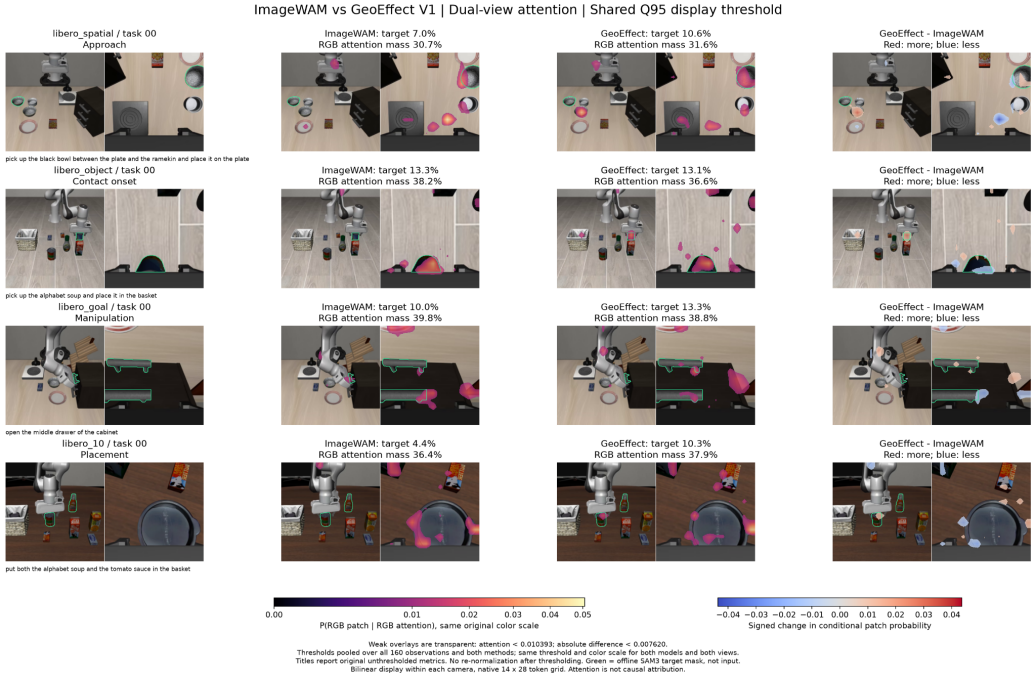


Figure 5: Representative RGB-patch attention maps from ImageWAM and an early Gae-WAM variant, labeled GeoEffect V1 in the original record. Both models use the same Q95 display threshold. Target-region fractions are computed from the raw attention weights; Q95 controls which patches are displayed.

In representative bowl-picking, drawer-opening, and placement states, attention mass in the target region changes from 7.0%→10.6%, 10.0%→13.3%, and 4.4%→10.3%, respectively. The contact-onset example changes from 13.3%→13.1%. The early variant allocates more attention to the target region in the first three examples, with similar allocation in the contact-onset example. Together, the maps show how target-region emphasis varies across manipulation states. The following camera probe evaluates the final checkpoints.

K Action Response to Camera Changes

We examine how camera changes affect the policy's action commands at a fixed physical state. This probe complements the closed-loop viewpoint evaluations by measuring translation, rotation, and gripper responses under controlled changes in visual input.

Evaluation protocol. The probe uses the final step-34,720 checkpoints of ImageWAM, Grounding-only, 2D-only, and Gae-WAM, trained with seed 42. We select 80 states from 40 LIBERO tasks using demonstration metadata, with one contact-onset and one active-manipulation state per task. The state inventory was fixed before the probe, following the benchmark evaluation. Inference uses deployment preprocessing and action denormalization, BF16 precision, a 512-token Qwen context, 10 denoising steps, and inference seed 42. Statistics average the two stages within each task and then weight the 40 tasks equally.

Fixed-state intervention. For each state, we render the nominal external camera and two alternatives at the 25th and 75th percentile positions of the canonically ordered official Camera Viewpoints configurations for that task. These interventions vary external-camera rotation, pitch, and zoom while preserving the physical state, wrist image, and proprioception. The target remains visible in all 240 observations, with a minimum area of 28 pixels at model-input resolution. Each model receives the same observations, producing $80 \times 3 \times 4 = 960$ action chunks of shape 16×7.

Command-space measurements. We analyze the first 12 commands, corresponding to the executed prefix before replanning. After clipping and scaling by the operational-space controller, let $u_{s,k}^{(v)}$, $\omega_{s,k}^{(v)}$, and $g_{s,k}^{(v)}$ denote translation increments, rotation increments, and gripper commands at state s and command k, with $v = 0$ denoting the nominal view. We measure

$$D_{\mathrm{trans}}(s) = \frac{1}{24} \sum_{v=1}^{2} \sum_{k=1}^{12} \|u_{s,k}^{(v)} - u_{s,k}^{(0)}\|_2, \tag{12}$$

$$D_{\mathrm{rot}}(s) = \frac{1}{24} \sum_{v=1}^{2} \sum_{k=1}^{12} \|\omega_{s,k}^{(v)} - \omega_{s,k}^{(0)}\|_2, \tag{13}$$

$$D_{\mathrm{grip}}(s) = \frac{1}{24} \sum_{v=1}^{2} \sum_{k=1}^{12} \mathbf{1}[g_{s,k}^{(v)} \neq g_{s,k}^{(0)}]. \tag{14}$$

Translation and rotation drift are expressed in mm/command and degrees/command. Nominal-view distances to demonstration commands describe action fidelity, and the norm of the across-state standard deviation of mean nominal translation commands measures response spread across physical states.

Lower mean motion-command drift. Gae-WAM has the lowest mean translation and rotation drift among the four evaluated policies: 6.881 mm/command and 0.706 degrees/command, compared with 7.156 and 0.742 for ImageWAM and 7.247 and 0.765 for Grounding-only (Figure 6). The paired reductions relative to Grounding-only are 0.366 mm/command and 0.058 degrees/command. These observations connect the viewpoint results in the main paper to the policy's continuous motion commands: across the sampled states, camera changes induce smaller average translation and rotation adjustments. Table 17 reports the paired differences for all three controls.

Response across states and action channels. Gae-WAM retains an across-state translation spread of 30.945 mm, within the 30.719–30.975 mm range of the four policies. Its nominal distances to demonstration commands are 9.353 mm/command and 1.150 degrees/command, within the corresponding ranges of 9.266–9.476 and 1.123–1.187. The lower mean camera drift therefore accompanies a comparable scale of state-dependent motion and demonstration distance. Gripper-flip rates are 3.385%, 3.594%, 3.854%, and 3.802% for ImageWAM, Grounding-only, 2D-only, and Gae-WAM, respectively. The channel-wise results locate the observed reduction in the continuous translation and rotation commands.

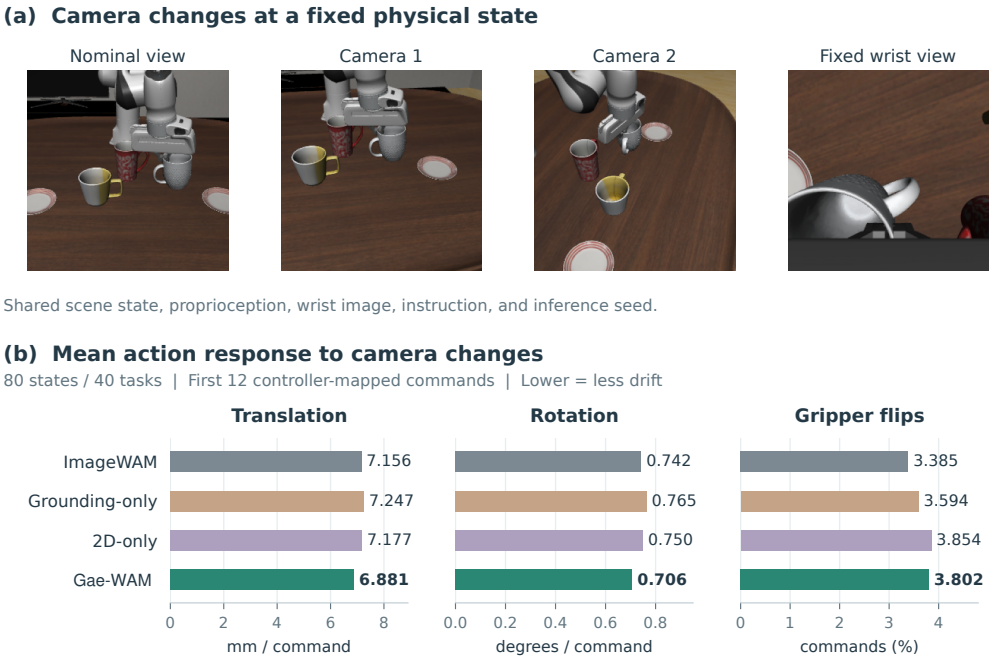


Figure 6: **Action response to camera changes.** Top: three external-camera views of the same physical state and the fixed wrist view. Bottom: task-weighted mean translation drift, rotation drift, and gripper-flip rate across 80 states from 40 tasks. Each measurement uses the first 12 controller-mapped commands. Paired differences and nominal-view control measurements appear in Table 17.

L IMAGEWAM RETRAINING AND DIFFERENCES FROM THE ORIGINAL PAPER

Our LIBERO ImageWAM baseline follows the official training configuration, with its checkpoints also evaluated on LIBERO-Plus as described in Appendix B. For RoboTwin2.0, ImageWAM and Gae-WAM use a common local training configuration specified in Appendix N. Within each benchmark, the two methods share demonstrations, base initialization, training budget, deployment inputs, and evaluation settings. All reported gains over ImageWAM refer to these matched comparisons. Published ImageWAM results serve as a separate literature reference.

M BASELINE SOURCES AND PRETRAINING

Standard LIBERO. Table 1 reports $\pi_{0.5}$, Motus, LingBot-VA, and Fast-WAM using the comparison in Yuan et al. (2026b). For RynnVLA-002, we use the continuous-action model at 256×256 image resolution (Cen et al., 2026). Its four suite rates are 99.0%, 99.8%, 96.4%, and 94.4%, with a 97.4% average. Our ImageWAM and Gae-WAM rows pool the three evaluation seeds listed in Appendix G.

LIBERO-Plus. OpenVLA-OFT and π_0-FAST rates follow Fei et al. (2026). SG-WAM rates follow Zhao et al. (2026), and Cosmos Policy and $\pi_{0.5}$ rates follow Zhang et al. (2026b). The matched ImageWAM baseline is retrained and evaluated with Gae-WAM as described in Appendix L.

RoboTwin2.0. Baseline result sources and comparison protocols are detailed in Appendix N.

Robot-policy pretraining. Robot-policy pretraining refers to training on robot actions or trajectories before downstream benchmark training. This is distinct from the image, video, or vision–language initialization of the backbone. Motus, LingBot-VA, $\pi_{0.5}$, OpenVLA-OFT, and π_0-FAST use robot-policy pretraining. GR00T-N1.7 uses cross-embodiment robot pretraining (NVIDIA, 2026). RynnVLA-002, Fast-WAM, AHA-WAM, SG-WAM, Cosmos Policy, and our matched ImageWAM

Table 17: **Camera-response statistics and nominal-view controls.** All estimates use the same 80 states from 40 tasks. Panel (a) reports mean distances to demonstration commands and across-state translation spread. Panel (b) gives control-minus-Gae-WAM differences. Gripper-flip differences are expressed in percentage points; other differences use the units shown in the table.

(a) Nominal-view controls

Method	Demo translation (mm/command)	Demo rotation (°/command)	Response spread (mm)
ImageWAM	9.476	1.187	30.975
Grounding-only	9.266	1.123	30.813
2D-only	9.405	1.134	30.719
Gae-WAM	9.353	1.150	30.945

(b) All paired comparisons with Gae-WAM

Metric	Control	Difference
Translation drift (mm/command)	ImageWAM	0.275
	Grounding-only	0.366
	2D-only	0.296
Rotation drift (°/command)	ImageWAM	0.036
	Grounding-only	0.058
	2D-only	0.043
Gripper flips	ImageWAM	−0.417
	Grounding-only	−0.208
	2D-only	0.052
Demo translation (mm/command)	ImageWAM	0.124
	Grounding-only	−0.087
	2D-only	0.053
Demo rotation (°/command)	ImageWAM	0.036
	Grounding-only	−0.027
	2D-only	−0.017

and Gae-WAM comparisons use no additional robot-policy pretraining for the reported simulation results. The StarVLA row uses the configuration evaluated in Yuan et al. (2026a), without additional robot-policy pretraining.

N RoboTwin 2.0 C2R Protocol

Data and initialization. The RoboTwin2.0 Clean-to-Randomized (C2R) protocol uses 50 `demo_clean` demonstrations per task across 50 tasks, giving 2,500 clean episodes (Chen et al., 2026). We split these episodes into 2,475 training and 25 validation episodes and use clean data for both subsets. ImageWAM and Gae-WAM start from the same FLUX.2-klein-base-4B, VAE, and RoboTwin ActionDiT initialization weights. Each model is trained for the full 30,000-step budget from this shared initialization. We use a common local RoboTwin configuration for the two methods; it differs from the official ImageWAM configuration followed in our LIBERO experiments.

Shared training configuration. Table 18 lists the settings shared by ImageWAM and Gae-WAM. Inputs comprise the main camera and both wrist cameras, using the `compact_288x256` preprocessing preset. The action and proprioceptive vectors each have 14 dimensions. The RoboTwin text configuration uses cached Qwen features with a context length of 128. Both methods apply the same data augmentation with probability 0.5, including color jitter, gamma adjustment, noise, and geometrically consistent cropping and rotation. For Gae-WAM, spatial augmentations also transform the corresponding geometric references.

Geometric auxiliary supervision. Gae-WAM adds the observation-derived 2D+3D geometric reference and the two effect prediction heads described in the method. The reference constructor uses the V1 configuration. Table 19 gives the auxiliary settings; ImageWAM retains the shared image and action objectives without these auxiliary components. Both

Table 18: **Shared RoboTwin2.0 training settings.** ImageWAM and Gae-WAM use the same data split, initialization, optimization, and augmentation settings.

Setting	Value
Training / validation episodes	2,475 / 25, all from clean demonstrations
Training seed	42
Optimizer steps	30,000
Hardware per model	$8\times$ H100 80GB
Global batch size	384: 12 per GPU, 8 GPUs, 4 accumulation steps
Precision / distributed strategy	BF16 / DeepSpeed ZeRO-1
Optimizer	AdamW, $\beta = (0.9, 0.95)$, weight decay 0.01
Gradient clipping	1.0
Peak action / proprioception learning rate	10^{-4}
Peak visual learning rate	3×10^{-5}
Learning-rate schedule	1,500 warmup steps, then cosine decay over the remainder of the 30,000-step schedule; minimum 10^{-6}
Image / action loss weights	0.5 / 1.0
Action horizon	16
Action / proprioception dimensions	14 / 14
Camera inputs	Main camera, left wrist, right wrist
Image preprocessing	`compact_288x256`
Qwen context length	128, with cached text features
Augmentation probability	0.5
Saved checkpoint	Final model weights at `step_030000.pt`

Table 19: **Gae-WAM auxiliary settings on RoboTwin2.0.** The loss weights follow the notation in the method section.

Setting	Value
Geometric reference	Per-view 2D changes and estimated 3D displacement
Visual grounding weight λ_e	1.0
Action grounding weight λ_a	1.0
Consistency weight λ_c	0.25
Overall auxiliary weight λ_E	0.01
Auxiliary-loss ramp $\alpha(k)$	Linear increase over the first 1,500 steps
Effect-head learning rate	10^{-4}
Head hidden dimension	256
Training mode	Joint optimization
Auxiliary noise range	$[0.05, 0.9]$
Shuffled-action loss weight	0 (disabled)

effect heads and the trainable WAM components are optimized jointly throughout training. The heads use `conditional_difference_rms_v1` with hidden dimension 256 and `current_symmetric` alignment. Current visual context is detached before entering the auxiliary heads (`detach_context=true`). Auxiliary supervision uses the noise range $\sigma \in [0.05, 0.9]$ with the `uniform_in_band` setting.

Masks are used for geometric reference construction; neither policy takes masks as input or predicts them. Invalid reference dimensions are excluded from the auxiliary losses, while the corresponding training windows are retained. This comparison measures the contribution of the complete geometric auxiliary objective. The separate contribution of cross-branch consistency is examined by the Grounding-only ablation on LIBERO-Plus.

Table 20: **RoboTwin2.0 success by task type, pooling Clean and Randomized.** Each task contributes 200 trials per method. Δ is Gae-WAM minus ImageWAM in percentage points; net successes give the corresponding difference in successful rollouts.

Task type	Tasks	Trials/method	Δ	Net successes
Transport, placement, and storage	24	4,800	+1.15	+55
Stacking and sorting	6	1,200	+1.25	+15
Explicit bimanual handovers	2	400	+0.50	+2
Grasping, lifting, and reorientation	6	1,200	-0.83	-10
Bottle shaking	2	400	+1.25	+5
Trash dumping	1	200	+6.50	+13
Tool use and alignment	3	600	-1.50	-9
In-place pressing and contact	3	600	+0.17	+1
Articulated-object manipulation	3	600	-4.17	-25
All tasks	**50**	**10,000**	**+0.47**	**+47**

Evaluation. Both models are evaluated using their final step-30,000 checkpoint and evaluation seed 42. Each model performs 100 rollouts per task under Clean (Easy) and Randomized (Hard) conditions, giving 5,000 rollouts per condition and 10,000 in total. The reported average weights the two conditions equally. Inference uses the action-only path with a 16-step action horizon, replanning every 16 steps, 10 denoising steps, and a text classifier-free guidance scale of 1.0. The instruction setting is `unseen`. The geometric reference constructor and effect heads are inactive during inference. Evaluation partitions the 50 tasks into four independent eight-GPU shards containing 13, 13, 12, and 12 tasks. Only the final model checkpoint is saved; intermediate checkpoints and optimizer states are not retained.

Baseline sources. The main table groups methods using clean-only downstream training; randomized downstream demonstrations define a different training condition. The C2R rates for StarVLA (Community, 2026), GR00T-N1.7 (NVIDIA, 2026), and $\pi_{0.5}$ (Black et al., 2025a) are reported by Yuan et al. (2026a). Fast-WAM (Yuan et al., 2026b) and AHA-WAM (Cai et al., 2026) use the clean-only results from the official OpenWAM benchmark repository (OpenWAM-Official, 2026): $77.8/1.9/39.9\%$ and $64.3/3.2/33.8\%$ for Clean/Randomized/Avg., respectively. Our ImageWAM and Gae-WAM results use the matched local configuration specified above.

N.1 Success by Task Type

We group the 50 RoboTwin2.0 tasks into nine categories according to the manipulation operations in their implementations. For each task, we pool 100 Clean and 100 Randomized trials, yielding 200 trials per method. For a category containing n tasks, its success rate is the total number of successes divided by $200n$. Table 20 reports the difference between Gae-WAM and ImageWAM under this aggregation. Across all tasks, ImageWAM records $4{,}676/10{,}000$ successes (46.76%) and Gae-WAM records $4{,}723/10{,}000$ (47.23%).

Category contributions. Transport, placement, and storage contribute 55 additional successes, and stacking and sorting contribute 15. Explicit bimanual handovers, bottle shaking, trash dumping, and in-place contact contribute another 21 in total. Articulated-object manipulation and tool use and alignment decrease by 4.17 and 1.50 percentage points, respectively. Across all nine categories, the net gain is 47 successes.

Task details. The object-in-basket task includes placing the object inside the basket and lifting the basket; it contributes 27 additional successes across the two conditions. Trash dumping contributes 13, and the remaining 48 tasks contribute seven in total. Articulated-object tasks include opening a microwave and operating a switch; tool-use tasks include hammering, stamping, and scanning. For the two bottle-shaking tasks, the benchmark success criterion is based on object lift height.

O REAL-ROBOT PROTOCOL

Platform and shared comparison. ImageWAM and Gae-WAM share demonstrations, deployment inputs, and initial conditions on the Astribot S1 dual-arm platform. We evaluate each method on four tasks, with 30 trials per task (120 per method). Figure 7 shows representative keyframes.

Fold Towel (`fold_towel`). The robot folds the towel twice and releases it in the demonstrated final configuration. The task involves fabric deformation and changes in projected extent.

Pen Storage (`penholder`). The robot grasps a target pen, transports it to the designated holder, aligns it with the opening, and releases it inside. Success requires the pen to remain in the holder after release. This task tests transport and orientation-sensitive placement at a narrow opening.

Pack Gift (`pack_gift`). The robot grasps the gift, places it inside the designated box, and closes the lid. Success requires the gift to remain inside with the lid closed.

Basketball Placement (`basketball`). The robot grasps a toy basketball, moves it above the hoop, and releases it through the opening. Success requires the ball to pass through the hoop after release. This task tests grasp retention and controlled release of a rounded object.

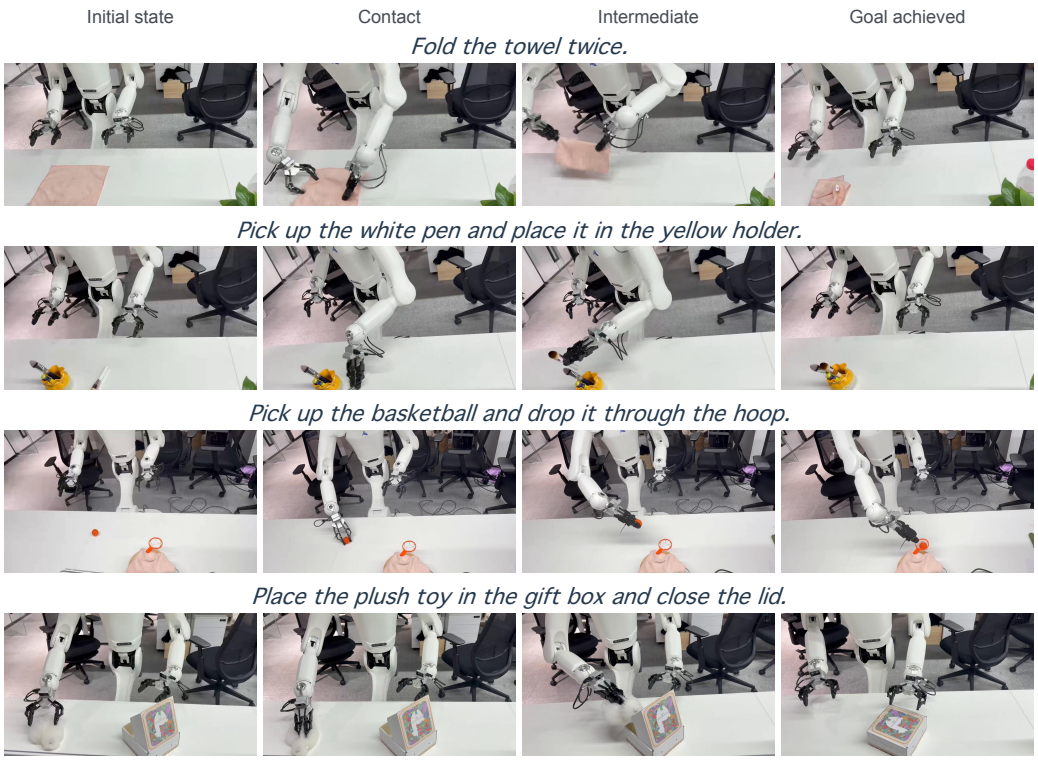


Figure 7: **Real-world manipulation keyframes on Astribot S1.** Rows show towel folding, pen storage, basketball placement, and gift packing, with the corresponding language instructions above each sequence. Columns show the initial state, contact, an intermediate stage, and task completion.