JEPA-VA: JEPA IS A GOOD VISUAL ACTION VERIFIER FOR GENERALIZABLE ROBOT POLICIES

Anonymous authors
Paper under double-blind review

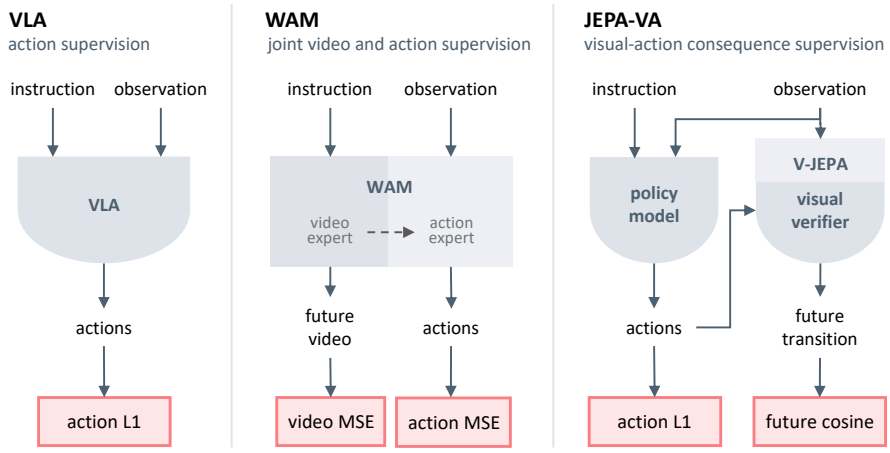


Figure 1: **Comparison of training paradigms: VLA, WAM, and JEPA-VA.** VLAs use only action supervision; WAMs jointly supervise video and actions. JEPA-VA adds visual-action consequence supervision: a V-JEPA verifier predicts the feature change associated with a policy action chunk, supervised by change-weighted cosine alignment with the demonstrated change.

ABSTRACT

Vision-language-action (VLA) policies are commonly trained to align predicted actions with demonstrations, while world-action models (WAMs) jointly predict future video and action; neither explicitly checks whether an action produces its intended visual consequence. We study this complementary training signal: does a policy's proposed action lead to the expected visual change? We introduce JEPA-VA, a JEPA visual verifier that evaluates proposed action chunks through their predicted feature changes relative to demonstrated transitions. Conditioned on the current observation and an action chunk, the verifier predicts the feature change at the end of the chunk. During policy training, a change-weighted cosine objective aligns the predicted change direction with the demonstrated one, sending a consequence-consistency gradient through the verifier's action input back to the policy. The verifier is not needed at deployment and adds no inference cost. Verifier-guided policy learning raises LIBERO-Plus success from 62.68% to 71.73% and clean-to-randomized RoboTwin success from 12.00% to 17.80%, suggesting that JEPA-VA helps policies learn more generalizable robot manipulation.

1 INTRODUCTION

Robot policies must choose actions that bring a scene closer to the task goal. Imitation learning usually trains them by matching demonstrated actions, although action error is only an indirect measure of what happens in the scene: a small error at contact can matter more than a larger error during free-space motion, and different actions can lead to similar outcomes. This gap motivates

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.

supervising actions through their visual consequences, especially when policies must generalize to changes in camera viewpoint, lighting, background, or object layout.

Vision-language-action (VLA) policies align predicted actions with demonstrations, while world-action models (WAMs) jointly predict future video and action. These objectives motivate a complementary question: can consistency between the visual consequence predicted for a policy's proposed action and the demonstrated transition provide a useful policy-learning signal? An action-conditioned visual predictor offers a differentiable route from consequence error back to the action that produced its prediction. This route lets visual transition supervision act directly on the policy's action output.

Turning visual prediction into action supervision requires an objective that emphasizes the scene transition. Persistent scene content can dominate an absolute future-feature objective, allowing a predictor to obtain low error by approximately copying the current observation. We represent the transition as the difference between future and current features. Aligning the direction of this residual with the demonstrated change, while weighting patches by their demonstrated change magnitude, focuses supervision on changes over the action horizon.

We introduce JEPA-VA, a policy-learning framework built around a frozen, action-conditioned JEPA visual verifier. We first train the verifier to predict the visual feature change at the end of a demonstrated action chunk. Its change-weighted cosine objective aligns residual directions and assigns greater weight to patches with larger demonstrated feature changes. We then freeze the verifier and use it to predict the change associated with the policy's proposed action. Discrepancy from the demonstrated change supplements action imitation, with gradients flowing through the verifier's action input to the policy. At deployment, the policy runs on its own, with no verifier computation or future observations required.

We use frozen V-JEPA 2.1 features as the visual representation, motivated by prior work on pretrained visual features for manipulation and video-predictive representations (Nilaksh et al., 2026; Zhang et al., 2025; Assran et al., 2025; Mur-Labadia et al., 2026; Miao et al., 2026). Our encoder comparison with SigLIP2, DINOv3, and VGGT supports this choice on LIBERO and LIBERO-Plus (Tschannen et al., 2025; Siméoni et al., 2025; Wang et al., 2025). We evaluate consequence supervision on single-arm LIBERO-Plus and bimanual RoboTwin (Fei et al., 2025; Chen et al., 2025), comparing policies trained w/o and w/ the verifier at three capacities. The reported Base-policy success on LIBERO-Plus increases from 62.68% to 71.73%, while Giant-policy success on clean-to-randomized RoboTwin increases from 12.00% to 17.80%. On a Unitree G1D robot, JEPA-VA obtains the highest weighted score on both a simple toy-to-box task (69.50 vs. 43.50 for the strongest baseline, FastWAM) and an advanced instruction-conditioned object-to-basket task (41.90 vs. 29.52).

Our contributions are summarized as follows:

- We propose JEPA-VA, a policy-learning framework that supplements action imitation with visual consequence supervision through a frozen, action-conditioned JEPA verifier, adding no inference cost at deployment.
- We formulate a change-sensitive verification objective that aligns predicted and demonstrated feature-change directions, weighted by demonstrated change magnitude to emphasize visual transitions.
- We evaluate visual consequence supervision on single-arm LIBERO-Plus and bimanual RoboTwin, reporting improved performance under distribution shift and analyzing how the gains vary across three policy capacities.

2 Related Work

Imitation Learning and Generalist Robot Policies. Action Chunking with Transformers predicts temporally extended actions for manipulation (Zhao et al., 2023), while Diffusion Policy models action sequences with a conditional denoising process (Chi et al., 2023). Vision-language-action models such as OpenVLA further exploit large-scale pretrained representations and robot data (Kim et al., 2024). These approaches learn control from demonstrated actions through different policy representations and action parameterizations. JEPA-VA augments action imitation with a loss on the

visual change predicted for the policy's action output. The imitation objective anchors actions to demonstrations, while the consequence objective supplies an additional gradient through an action-conditioned visual predictor.

Predictive Visual Representations and Latent Dynamics. V-JEPA 2 combines predictive visual pretraining with an action-conditioned model for planning, and V-JEPA 2.1 develops dense visual features through video self-supervised learning (Assran et al., 2025; Mur-Labadia et al., 2026). DINO-WM learns dynamics over pretrained patch features and optimizes action sequences toward visual goals (Zhou et al., 2024). These studies motivate modeling visual dynamics in feature space. In JEPA-VA, the latent predictor serves as a training objective for a policy: it is fitted independently, frozen, and differentiated through its action input to update policy parameters. Demonstrated futures supply the reference transitions during training, and deployment requires only the policy. To emphasize change over persistent scene content, the verifier predicts residual visual features with change-weighted matching.

World-Action Models and Robustness. World-action models connect visual prediction and action learning in several ways. Fast-WAM studies video co-training and supports action-only inference (Yuan et al., 2026); ImageWAM uses image-editing features as action context (Zhang et al., 2026); CSWAM integrates V-JEPA-derived semantic history into a world-action architecture to improve robustness (Liu et al., 2026). Our focus is the supervision applied to the policy's proposed actions: a frozen verifier maps them to predicted visual changes, and discrepancy from demonstrated changes propagates back to the policy. This mechanism complements the use of predictive visual features and joint video–action training by making consequence consistency an explicit objective on the action output. We study its generalization behavior using LIBERO, LIBERO-Plus, and RoboTwin, which provide manipulation, distribution-shift, and bimanual domain-randomization evaluations (Liu et al., 2023; Fei et al., 2025; Chen et al., 2025).

3 JEPA-VA

3.1 Problem Setup and Visual Representation

A demonstration supplies the current multi-view observation o_t, robot state s_t, instruction ℓ, an h-step action chunk $A_t = (a_t, \dots, a_{t+h-1}) \in \mathbb{R}^{h \times D_a}$, and the observation o_{t+h} at the end of the chunk. Here D_a is the action dimension and h is the prediction horizon. The verifier predicts a single endpoint transition for these h actions.

A frozen visual encoder E maps the camera views to patch features. Concatenating patches across cameras gives $Z_t = E(o_t) \in \mathbb{R}^{P \times D_z}$, where P is the total number of patches and D_z is their feature dimension. Each index p identifies both a camera and a spatial patch. The endpoint features $Z_{t+h} = E(o_{t+h})$ use the same encoder and patch ordering. The demonstrated feature change is

$$\Delta Z_t = Z_{t+h} - Z_t, \qquad \Delta z_{t,p} = z_{t+h,p} - z_{t,p}. \tag{1}$$

The policy predicts $\widehat{A}_t = \pi_\theta(o_t, s_t, \ell) = f_\theta(Z_t, s_t, \ell)$, where f_θ denotes the trainable policy operating on frozen visual features. The action-conditioned predictor maps the current features and a complete action chunk to an endpoint feature-change estimate,

$$\widehat{\Delta Z}_t = \Psi_\phi(Z_t, A_t) \in \mathbb{R}^{P \times D_z}. \tag{2}$$

We use this predictor as a *visual action verifier*: it evaluates an action chunk by the directional consistency of its predicted feature change with the demonstrated transition. Language and robot state condition the policy's actions, while the verifier evaluates those actions in the current visual context.

3.2 Stage I: Learn an Action-Conditioned Visual Verifier

Architecture. The verifier projects visual patches and action vectors into a common token space of width d_Ψ. Its memory contains P visual tokens and h action tokens. Camera embeddings distinguish views, and rotary positions encode spatial patch locations and action time indices. A set of P endpoint queries, indexed by camera and patch location at time $t + h$, attends to this memory. The

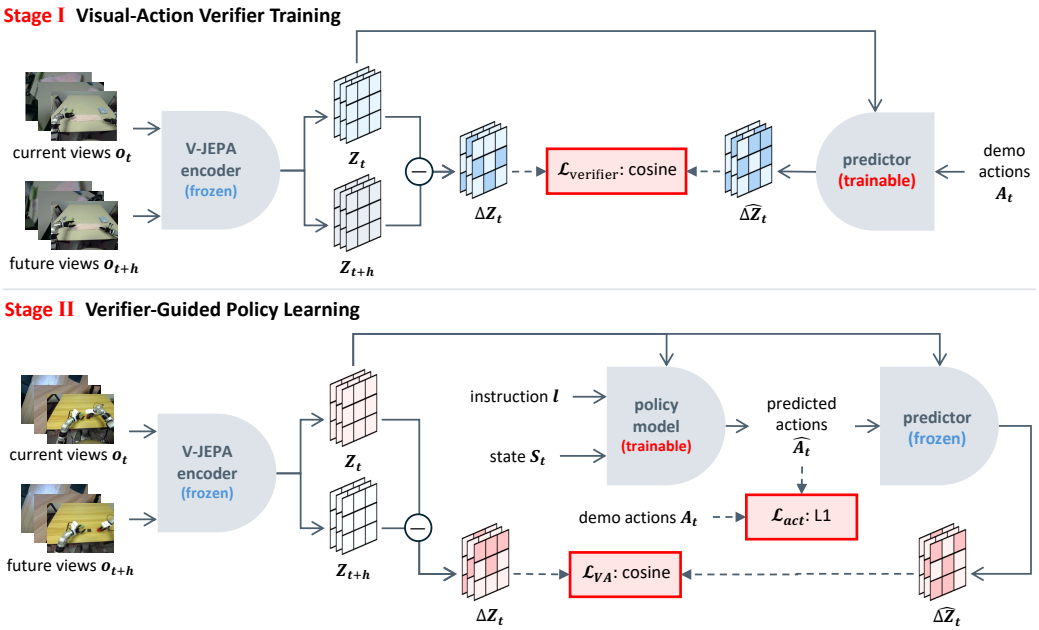


Figure 2: **Overview of JEPA-VA. Stage I:** the frozen V-JEPA encoder produces current and endpoint features Z_t and Z_{t+h}. Given Z_t and an h-step demonstrated action chunk A_t, the predictor learns the residual $\Delta Z_t = Z_{t+h} - Z_t$ using $\mathcal{L}_{\text{verifier}}$. **Stage II:** the policy predicts $\widehat{A}_t$ from current visual features, instruction ℓ, and state s_t. Training combines action L1 loss $\mathcal{L}_{\text{act}}$ with $\mathcal{L}_{\text{VA}}$, which aligns the predicted and demonstrated residual directions through the frozen predictor. Both cosine losses use the change weights in Equation 4. Gradients through the predictor's action input update the policy; deployment uses only the policy.

decoder outputs are projected to D_z dimensions, giving one predicted residual per patch. Padding masks exclude padded action tokens from the verifier's action memory. The verifier has its own action projection, so the same fitted verifier can supervise any policy that predicts action chunks.

Change-Sensitive Objective. We supervise the direction of the predicted feature change and weight each patch by the magnitude of its demonstrated change. For a minibatch $\mathcal{B}$, let $i = (b, p)$ index examples and patches, and set $w_i = \|\Delta z_i\|_2$. Define the stabilized cosine as

$$\cos_\epsilon(u, v) = \frac{u^\top v}{\max(\|u\|_2, \epsilon)\max(\|v\|_2, \epsilon)}, \qquad \epsilon = 10^{-8}. \tag{3}$$

The change-weighted discrepancy is

$$\mathcal{D}(\widehat{\Delta Z}, \Delta Z) = \frac{\sum_i w_i \left[1 - \cos_\epsilon(\widehat{\Delta z}_i, \Delta z_i)\right]}{\max(\sum_i w_i, \epsilon_w)}, \qquad \epsilon_w = 10^{-6}. \tag{4}$$

The cosine term aligns residual directions, while the target norm determines each patch's contribution to the minibatch loss. Patches with zero feature change receive zero weight. For a nonzero target, predicting zero change gives a per-patch cosine discrepancy of one, so directly copying the current features into an endpoint prediction does not yield a low residual discrepancy. This construction concentrates supervision on visual transitions over the action horizon.

We define the Stage-I verifier loss and its fitted parameters as

$$\mathcal{L}_{\text{verifier}}(\phi; \mathcal{B}) = \mathcal{D}\big(\Psi_\phi(Z, A), \Delta Z\big), \tag{5}$$

$$\phi^* = \arg\min_\phi \; \mathbb{E}_{\mathcal{B}}\left[\mathcal{L}_{\text{verifier}}(\phi; \mathcal{B})\right]. \tag{6}$$

Here Z, A, and ΔZ denote the batched tensors. Stage I fits ϕ on successful demonstration transitions while keeping E fixed. The encoder supplies both current and target features in the same representation space.

3.3 Stage II: Train the Policy through the Frozen Verifier

After Stage I, we freeze ϕ^* and use the policy's predicted chunk as the verifier input. The resulting endpoint residual is

$$\widehat{\Delta Z}^{\pi} = \Psi_{\phi^*}(Z, \widehat{A}). \tag{7}$$

Policy training combines action imitation and visual transition consistency:

$$\mathcal{L}_{\text{act}} = \frac{\sum_{b,k,d} m_{bk} |\widehat{a}_{bkd} - a_{bkd}|}{\max(D_a \sum_{b,k} m_{bk}, 1)}, \tag{8}$$

$$\mathcal{L}_{\text{VA}} = \mathcal{D}\big(\widehat{\Delta Z}^{\pi}, \Delta Z\big), \tag{9}$$

$$\mathcal{L}_{\pi} = \mathcal{L}_{\text{act}} + \lambda \mathcal{L}_{\text{VA}}. \tag{10}$$

The sums use $k = 0, \ldots, h-1$ and $d = 1, \ldots, D_a$; $m_{bk} = 1$ for a valid action step and 0 for padding. Demonstrated and predicted actions use the same per-dimension normalization in both stages. The imitation term anchors action predictions to demonstrations. The visual term encourages the predicted endpoint change to follow the demonstrated direction, emphasizing patches with larger demonstrated changes.

During this stage, E, ϕ^*, and the target features are fixed. The verifier retains derivatives with respect to its action input, yielding

$$\nabla_{\theta} \mathcal{L}_{\text{VA}} = \left(\frac{\partial \widehat{A}}{\partial \theta} \right)^{\top} \left(\frac{\partial \Psi_{\phi^*}}{\partial \widehat{A}} \right)^{\top} \nabla_{\widehat{\Delta Z}^{\pi}} \mathcal{D}. \tag{11}$$

Thus the visual objective updates the policy through the action Jacobian of the fitted verifier. The action prediction remains connected to this computation graph while the verifier parameters stay fixed.

Policy and Inference. The policy uses a shared multimodal transformer. State, language, and visual features are projected to a common width d_π, forming tokens S_t, T_ℓ, and V_t. We prepend h learnable action-query tokens $Q_{\text{act}} \in \mathbb{R}^{h \times d_\pi}$ and process the joint sequence with the policy transformer F_θ:

$$U_t = F_\theta\big([Q_{\text{act}}; S_t; T_\ell; V_t]\big), \qquad \widehat{A}_t = g_\theta\big(U_t[1{:}h]\big). \tag{12}$$

Here $[\cdot;\cdot]$ denotes token concatenation and g_θ maps each action-query output to a D_a-dimensional action. Modality embeddings identify token types, and rotary positions encode language order, visual patch coordinates, and action time indices. The action queries interact with the observation tokens within the shared transformer. At deployment, the policy executes a prefix of its predicted chunk and replans from a new observation. Inference consists of the visual encoder and policy; the verifier and demonstrated endpoint targets are used during training.

4 Experiments

4.1 Experimental Setup

Benchmarks. We evaluate single-arm manipulation on the Spatial, Object, Goal, and Long suites of LIBERO (Liu et al., 2023), and bimanual manipulation on RoboTwin 2.0 (Chen et al., 2025). LIBERO contains 40 tasks with 50 trials per task. LIBERO-Plus evaluates the same tasks under seven visual, language, and state perturbations; its seven category success rates are averaged equally as Plus-7, rather than pooled across trials. RoboTwin evaluates 50 tasks with 100 trials per task in both clean and randomized scenes. Full task counts, trial protocols, and aggregation details are given in Appendix A.

Protocol. Our frozen visual encoder is V-JEPA 2.1 ViT-G (gigantic); Base/Large/Giant denote the *policy transformer*, not the visual encoder or verifier size. For external comparisons, we report the Base policy on LIBERO-Plus and the Giant policy on RoboTwin. LIBERO's single-arm demonstrations are comparatively easy to overfit, and added policy capacity brings little robustness gain there

Table 1: **LIBERO-Plus comparison by perturbation.** Success rates (%) on the seven perturbation categories and their mean (Plus-7), with parameter counts and action-chunk inference latency. Bold and underline mark the best and second-best results; for parameters and latency, lower is better.

Method	Params (B)	Lat. (ms)	Plus-7	Perturbation category						
				Cam.	Robot	Lang.	Light	BG	Noise	Layout
UniVLA (Bu et al., 2025)	7.55	144	45.8	1.8	46.2	69.6	69.0	81.0	21.2	31.9
π_0 (Black et al., 2024)	3.24	162	56.1	13.8	6.0	58.8	85.0	<u>81.4</u>	**79.0**	<u>68.9</u>
π_0-FAST (Pertsch et al., 2025)	<u>2.92</u>	233	62.5	<u>65.1</u>	21.6	61.0	73.2	73.2	74.4	68.8
OpenVLA-OFT-m (Kim et al., 2025)	7.71	<u>141</u>	69.9	55.6	21.7	**81.0**	**92.7**	**91.0**	<u>78.6</u>	68.7
FastWAM (Yuan et al., 2026)	6.72	349	51.4[†]	16.4	44.5	68.9	78.2	53.7	37.7	60.7
ImageWAM (Zhang et al., 2026)	4.60	216	<u>71.4</u>[†]	63.3	<u>58.4</u>	<u>75.4</u>	86.3	66.7	75.2	**74.6**
JEPA-VA-Base	**1.95**	**97**	**71.73**	**76.1**	**79.5**	61.1	<u>87.7</u>	57.2	72.8	67.7

Note: Results for π_0, π_0-FAST, and OpenVLA-OFT-m are from LIBERO-Plus (Fei et al., 2025); UniVLA from Allen Institute for AI (2026); FastWAM and ImageWAM (Ovis-U1 variant) from Zhang et al. (2026). [†]Recomputed from the published category rates (Appendix F). Latencies are internal H20 PyTorch-eager medians; our parameters and latency cover the per-step inference path (Table 4). Cam. = Camera, Lang. = Language, BG = Background.

Table 2: **Clean-trained RoboTwin comparison.** Success rates (%) over 50 tasks for models trained on clean scenes and evaluated on clean (C2C) or randomized (C2R) scenes, with parameter counts and action-chunk inference latency. Per-task results are in Appendix D.3. Bold and underline mark the best and second-best results; for parameters and latency, lower is better.

Method	Params (B)	Latency (ms)	C2C	C2R
DP3 (Ze et al., 2024)	**0.26**	**63**	55.24	4.96
GalaxeaVLA (Galaxea AI, 2026)	3.24	245	62.72	12.72
RDT (Liu et al., 2024)	<u>1.66</u>	186	34.50	13.72
π_0 (Black et al., 2024)	3.24	162	46.42	<u>16.34</u>
AHA-WAM (Cai et al., 2026)	7.23	415.77	64.32	3.20
FastWAM (Yuan et al., 2026)	6.72	344	**77.76**	1.92
JEPA-VA-Giant	2.93	<u>142</u>	<u>77.50</u>	**17.80**

Note: External success rates are means of the per-task RoboTwin 2.0 Leaderboard results (RoboTwin Team, 2026) (Appendix D.3). DP3, RDT, and π_0 use single-task SFT; the others use joint training. Latencies are internal H20 PyTorch-eager medians, except AHA-WAM, reported in its paper on an RTX 5090D (Cai et al., 2026).

(Table 4); RoboTwin's 50 bimanual tasks are more complex and benefit consistently from capacity. We report LIBERO policies at 60K EMA steps and RoboTwin policies at 100K; full training and evaluation settings are in Appendix A. The verifier is trained once and adds no computation at deployment.

Evidence. External baselines represent latent-action, VLA, and world-action policy families; we report their published success rates and measure their latency with our protocol. Within our study, the w/o verifier reference, encoder substitutions, and w/ verifier policies are trained in the same codebase. The encoder ablation uses 20K LIBERO and 60K RoboTwin policy steps; the main comparisons use 60K and 100K, respectively. Sources of public results are listed in Appendix F.

4.2 Quantitative Comparison

LIBERO-Plus. Table 1 reports Plus-7 and all seven LIBERO-Plus perturbation categories. JEPA-VA-Base reaches 71.73% Plus-7, the highest in the table, with fewer parameters (1.95B) and lower latency (97 ms) than every baseline. It achieves the best camera-viewpoint (76.1%) and robot-initial-state (79.5%) robustness by large margins, the two categories where most VLAs degrade sharply.

RoboTwin. JEPA-VA-Giant obtains 77.50% C2C and 17.80% C2R (Table 2). Its clean-scene score is within 0.26 points of FastWAM, while randomized-scene success is 15.88 points higher. It

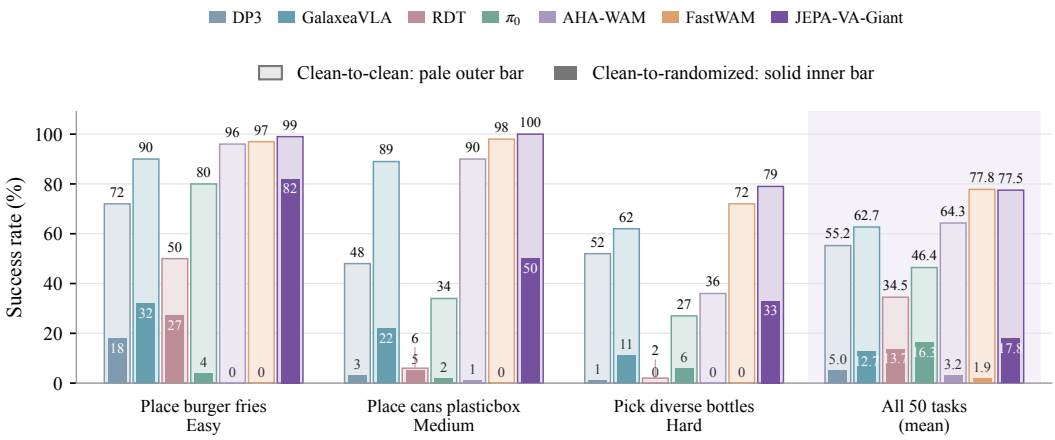


Figure 3: **Clean and randomized RoboTwin performance.** Pale outer bars show C2C success and solid inner bars show C2R success; bars are overlaid, not stacked. Methods follow the order of Table 2. The first three groups show one selected task per difficulty level; the last reports the mean over all 50 tasks. Full per-task results are in Appendix D.3.

also exceeds the listed GalaxeaVLA and AHA-WAM results on both settings. These observations show that comparable nominal success can coexist with substantially different transfer behavior, and that consequence supervision achieves strong transfer without generating future video.

Figure 3 highlights generalization strengths at three empirical difficulty levels, alongside the full-benchmark mean. We rank all 50 tasks by mean C2C success across the seven methods and split them into easy, medium, and hard groups of 17, 17, and 16 tasks. From each group we show the task with the largest C2R margin over the strongest baseline (+50, +28, and +22 points, respectively); these are selected strengths rather than official difficulty categories.

4.3 Ablation Studies

Encoder. We first examine the representation used by the w/o verifier policy (Table 3). The encoder ablation uses a short budget, 20K steps on LIBERO and 60K on RoboTwin, rather than the 60K and 100K used for the main results. Because the question is which frozen representation a policy can exploit most readily, an under-trained regime is more informative: it reflects how quickly each encoder supports useful behavior, before extended training lets the policy compensate for a weaker representation. V-JEPA 2.1 obtains 93.80% LIBERO and 39.51% LIBERO-Plus, compared with 91.75/24.17% for SigLIP2 Giant, 90.65/18.17% for DINOv3, and 84.30/11.89% for VGGT. The separation is larger under LIBERO-Plus perturbation than on nominal LIBERO tasks. On RoboTwin, V-JEPA has the highest reported C2C aggregate (31.46%), whereas SigLIP2 has the highest C2R aggregate (7.36%). The LIBERO-Plus category breakdown appears in Table 11.

Table 3: **Visual encoder comparison without consequence verification.** Recorded success rates (%) on LIBERO and LIBERO-Plus after 20K policy steps and on RoboTwin 2.0 (Chen et al., 2025) after 60K steps. The longer-trained capacity and verifier results in Table 4 are separate experiments. Plus-7 is the unweighted mean of the seven category rates in Table 11. Bold and underline mark the best and second-best results.

Encoder	LIBERO		RoboTwin	
	Original	Plus-7	C2C	C2R
V-JEPA 2.1 (Mur-Labadia et al., 2026)	**93.80**	**39.51**	**31.46**	<u>6.60</u>
SigLIP2 Giant (Tschannen et al., 2025)	<u>91.75</u>	<u>24.17</u>	26.80	**7.36**
DINOv3 (Siméoni et al., 2025)	90.65	18.17	<u>29.12</u>	4.14
VGGT (Wang et al., 2025)	84.30	11.89	14.60	4.94

Table 4: **Policy capacity and visual consequence verification.** Success rates (%), parameter counts (B), and action-chunk inference latency (ms). Base/Large/Giant share the V-JEPA 2.1 ViT-G encoder and differ in policy-head size. Bold marks the higher success rate within each size pair.

Policy	Training	Params (B)	LIBERO			RoboTwin		
			Latency (ms)	Original	Plus-7	Latency (ms)	C2C	C2R
Base	w/o verifier	1.95	97	92.95	62.68	97	**71.38**	6.68
	w/ verifier			**95.00**	**71.73**		70.18	**8.34**
Large	w/o verifier	2.19	105	94.00	64.02	109	**73.22**	8.32
	w/ verifier			**94.35**	**72.25**		73.14	**11.10**
Giant	w/o verifier	2.93	124	**93.80**	68.35	142	77.20	12.00
	w/ verifier			93.65	**68.50**		**77.50**	**17.80**

Capacity. The w/o verifier RoboTwin policies improve monotonically from Base to Large to Giant: C2C rises from 71.38 to 73.22 to 77.20%, and C2R from 6.68 to 8.32 to 12.00%. The verifier-trained policies follow the same ordering on RoboTwin. LIBERO behaves differently: w/o verifier Plus-7 rises from Base to Giant (62.68 to 68.35%), but verifier-trained Plus-7 peaks at Large (72.25%) rather than Giant (68.50%). Capacity thus consistently benefits bimanual manipulation, while on LIBERO-Plus the verifier lets Base and Large policies surpass the w/o verifier Giant policy.

Verifier. At size Base, verification improves LIBERO by 2.05 points and Plus-7 by 9.05 points; at Large, the corresponding gains are 0.35 and 8.23 points. The Giant w/o verifier policy is already strong on LIBERO-Plus (68.35%), and verification yields a further +0.15 points. On RoboTwin, C2R increases at all three capacities, by 1.66, 2.78, and 5.80 points for Base/Large/Giant, respectively, while C2C stays within 1.2 points. For Giant, C2R rises from 12.00 to 17.80%, a 48% relative gain. Consequence supervision thus consistently improves robustness under randomization, and its RoboTwin gain grows with policy capacity.

Efficiency. Latency covers the visual encoder and policy head (single H20, batch 1, bf16, PyTorch eager; two LIBERO and three RoboTwin cameras). All three sizes run faster than every external baseline except DP3, and the training-only verifier is absent from the inference path.

Perturbation Profile. Figure 4 compares the Base policy with and without consequence verification across all seven perturbation categories. For the Base policy, the recorded LIBERO-Plus category differences are positive in all seven categories (Appendix C.1). The largest are robot initial state (+14.3 points), lighting (+13.6), and camera viewpoint (+10.9); backgrounds (+2.4) and language (+4.1) improve less. The largest gains appear in perturbations that alter the visual transition an action produces, consistent with the method's emphasis on visual consequences.

4.4 Real-World Evaluation

We deploy JEPA-VA on a Unitree G1D with Dex1 grippers on two tasks (Figure 5, Appendix E): toy-to-box, and object-to-basket, where three instructions each name a different target among three objects.

Each trial records whether the gripper reaches the target, grasps it, and completes the placement. We summarize these stages with a weighted score, $\mathrm{Score} = (20N_{\mathrm{reach}} + 30N_{\mathrm{grasp}} + 50N_{\mathrm{success}})/N_{\mathrm{trials}}$, which ranges from 0 to 100. Table 5 reports 20 trials for toy-to-box and 21 trials for object-to-basket (7 per instruction). JEPA-VA obtains the highest score on both tasks: on toy-to-box it succeeds in 12 of 20 trials, twice as often as FastWAM or ACT, and on object-to-basket it completes 7 of 21 placements. Per-instruction results appear in Appendix E.

5 Discussion and Limitations

The experiments support a two-part account of robustness: JEPA features provide a useful visual foundation, and supervision through a frozen consequence model is associated with additional robustness gains. The Base policy improves on LIBERO-Plus beyond the w/o verifier JEPA reference,

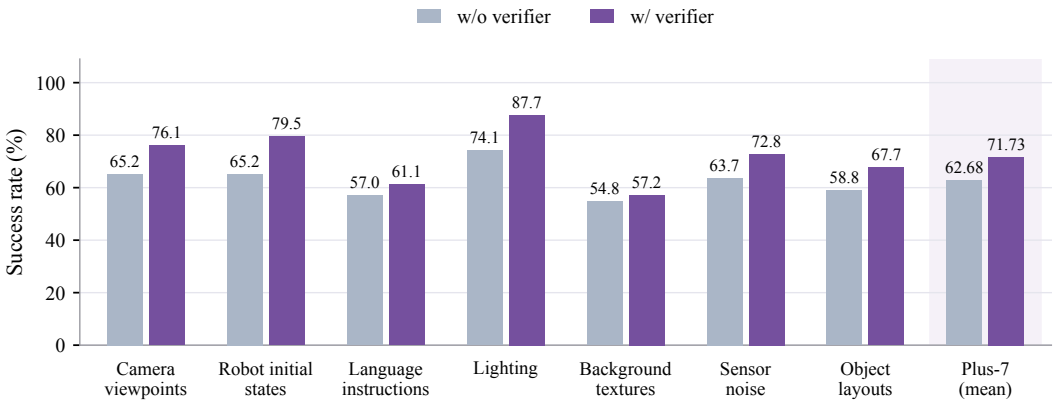


Figure 4: **LIBERO-Plus robustness with consequence verification.** Success rates (%) for the Base policy w/o and w/ verifier at 60K EMA. The seven categories contain 10,030 evaluation variants in total. Plus-7 is the recorded category-balanced mean, computed before rounding the category rates. Verification improves the recorded point estimate in every category, with the largest differences in robot initial states and lighting. Numerical values appear in Appendix C.1.

Table 5: **Real-robot results on the Unitree G1D.** Stage counts and weighted score; trial counts in parentheses. Bold and underline mark the best and second-best results.

Method	Toy-to-box (20)				Object-to-basket (21)			
	Reach	Grasp	Success	Score	Reach	Grasp	Success	Score
ACT (Zhao et al., 2023)	15	<u>7</u>	<u>6</u>	40.50	12	3	2	20.48
FastWAM (Yuan et al., 2026)	<u>18</u>	<u>7</u>	<u>6</u>	<u>43.50</u>	<u>15</u>	<u>4</u>	<u>4</u>	<u>29.52</u>
JEPA-VA	**20**	**13**	**12**	**69.50**	**16**	**7**	**7**	**41.90**

while Giant improves clean-to-randomized RoboTwin transfer with nearly unchanged clean-scene performance. These observations connect visual consequence supervision to measurable policy behavior under distribution shift.

Tighter Fitting Does Not Imply Robustness. We also observe that training policies beyond the reported checkpoints keeps reducing action-imitation error but degrades both LIBERO-Plus and real-robot performance. Fitting demonstrations more closely therefore does not translate into robustness under distribution shift, and may indicate overfitting to the demonstration distribution.

JEPA-VA trains its verifier on successful demonstrations, so its guidance is strongest near the demonstration distribution. Training the verifier on policy rollouts or failure data could broaden this coverage, and larger-scale real-robot evaluation is a natural next step.

6 Conclusion

JEPA-VA uses an independently learned JEPA predictor as a frozen differentiable visual action verifier during policy training. The framework supplements action imitation with consistency between predicted actions and demonstrated visual changes, while omitting the verifier at inference. Experiments show the strongest LIBERO-Plus robustness among the compared policies and consistent gains in RoboTwin randomized-scene success over training w/o verifier, with lower inference latency than VLA and WAM baselines. These findings establish visual consequence supervision as an effective route to generalizable robot policies.

AI USE STATEMENT

An OpenAI Codex assistant was used to organize experiment records, inspect implementation details, retrieve bibliographic metadata, draft and revise text, and prepare the LaTeX manuscript. It was not used to generate experimental measurements. The authors verified all claims, references, and data.

REPRODUCIBILITY STATEMENT

The appendix specifies the objectives, available training and evaluation settings, checkpoint convention, and aggregation rule. Sources for all external results are listed in Appendix F. We will release code and checkpoints upon publication.

REFERENCES

Allen Institute for AI. VLA Evaluation Harness Leaderboard, 2026. URL https://allenai.github.io/vla-evaluation-harness/leaderboard/. Snapshot accessed September 21, 2026; data last updated August 10, 2026.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-JEPA 2: Self-Supervised Video Models Enable Understanding, Prediction and Planning. *arXiv preprint arXiv:2506.09985*, 2025. URL https://arxiv.org/abs/2506.09985.

Kevin Black, Noah Brown, Danny Driess, et al. π_0: A Vision-Language-Action Flow Model for General Robot Control. *arXiv preprint arXiv:2410.24164*, 2024. URL https://arxiv.org/abs/2410.24164.

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. UniVLA: Learning to Act Anywhere with Task-centric Latent Actions. *arXiv preprint arXiv:2505.06111*, 2025. URL https://arxiv.org/abs/2505.06111.

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. *arXiv preprint arXiv:2606.09811*, 2026. URL https://arxiv.org/abs/2606.09811.

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. *arXiv preprint arXiv:2506.18088*, 2025. URL https://arxiv.org/abs/2506.18088.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion Policy: Visuomotor Policy Learning via Action Diffusion. *arXiv preprint arXiv:2303.04137*, 2023. URL https://arxiv.org/abs/2303.04137.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. LIBERO-Plus: In-depth Robustness Analysis of Vision-Language-Action Models. *arXiv preprint arXiv:2510.13626*, 2025. URL https://arxiv.org/abs/2510.13626.

Galaxea AI. GalaxeaVLA: Model and Code Release, 2026. URL https://github.com/OpenGalaxea/GalaxeaVLA.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An Open-Source Vision-Language-Action Model. *arXiv preprint arXiv:2406.09246*, 2024. URL https://arxiv.org/abs/2406.09246.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-Tuning Vision-Language-Action Models: Optimizing Speed and Success. *arXiv preprint arXiv:2502.19645*, 2025. URL https://arxiv.org/abs/2502.19645.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking Knowledge Transfer for Lifelong Robot Learning. *arXiv preprint arXiv:2306.03310*, 2023. URL https://arxiv.org/abs/2306.03310.

Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. RDT-1B: a Diffusion Foundation Model for Bimanual Manipulation. *arXiv preprint arXiv:2410.07864*, 2024. URL https://arxiv.org/abs/2410.07864.

Tianbin Liu, Jian Zhu, Taiyi Su, Jianjun Zhang, Chong Ma, Zitai Huang, Weiyi Lu, and Yi Xu. CSWAM: Better Causal Semantic Representations for Out-of-Distribution Generalization in World Action Models. *arXiv preprint arXiv:2609.18462*, 2026. URL https://arxiv.org/abs/2609.18462.

Shangchen Miao, Ningya Feng, Jialong Wu, Ye Lin, Xu He, Dong Li, and Mingsheng Long. JEPA-VLA: Video predictive embedding is needed for VLA models. *arXiv preprint arXiv:2602.11832*, 2026.

Lorenzo Mur-Labadia, Matthew Muckley, Amir Bar, Mido Assran, Koustuv Sinha, Mike Rabbat, Yann LeCun, Nicolas Ballas, and Adrien Bardes. V-JEPA 2.1: Unlocking Dense Features in Video Self-Supervised Learning. *arXiv preprint arXiv:2603.14482*, 2026. URL https://arxiv.org/abs/2603.14482.

Nilaksh, Saurav Jha, Artem Zholus, and Sarath Chandar. Reconstruction or semantics? what makes a latent space useful for robotic world models. *arXiv preprint arXiv:2605.06388*, 2026.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. FAST: Efficient Action Tokenization for Vision-Language-Action Models. *arXiv preprint arXiv:2501.09747*, 2025. URL https://arxiv.org/abs/2501.09747.

RoboTwin Team. RoboTwin 2.0 Leaderboard, 2026. URL https://robotwin-platform.github.io/leaderboard. Snapshot accessed September 25, 2026; board updated September 24, 2026.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025. URL https://arxiv.org/abs/2508.10104.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. SigLIP 2: Multilingual Vision-Language Encoders with Improved Semantic Understanding, Localization, and Dense Features. *arXiv preprint arXiv:2502.14786*, 2025. URL https://arxiv.org/abs/2502.14786.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual Geometry Grounded Transformer. *arXiv preprint arXiv:2503.11651*, 2025. URL https://arxiv.org/abs/2503.11651.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-WAM: Do World Action Models Need Test-time Future Imagination? *arXiv preprint arXiv:2603.16666*, 2026. URL https://arxiv.org/abs/2603.16666.

Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3D Diffusion Policy: Generalizable Visuomotor Policy Learning via Simple 3D Representations. *arXiv preprint arXiv:2403.03954*, 2024. URL https://arxiv.org/abs/2403.03954.

Jianke Zhang, Yucheng Hu, Yanjiang Guo, Xiaoyu Chen, Yichen Liu, Wenna Chen, Chaochao Lu, and Jianyu Chen. UniJEPA: Enhancing robot policy via unified continuous and discrete representation learning. *arXiv preprint arXiv:2510.10642*, 2025.

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? *arXiv preprint arXiv:2606.19531*, 2026. URL https://arxiv.org/abs/2606.19531.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware. *arXiv preprint arXiv:2304.13705*, 2023. URL https://arxiv.org/abs/2304.13705.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World Models on Pre-trained Visual Features enable Zero-shot Planning. *arXiv preprint arXiv:2411.04983*, 2024. URL https://arxiv.org/abs/2411.04983.

A Implementation and Training Protocol

This appendix instantiates the visual representation, action-conditioned verifier, and frozen-verifier policy training described in the method. The following recipes distinguish the LIBERO, RoboTwin, and real-robot training settings.

A.1 LIBERO Training Recipe

Table 6: Base LIBERO recipe.

Component	Setting
Visual representation	Frozen V-JEPA 2.1 ViT-G
Policy architecture	Shared multimodal transformer with action-query tokens
Policy output	16-step normalized action chunk
Endpoint prediction horizon	16 action steps
Verifier input	Current visual patches and action chunk
Verifier target	Endpoint minus current frozen patch features
Verifier loss	Change-magnitude-weighted cosine discrepancy
Default verifier scale	Giant width, 2 decoder blocks
Stage-II objective	Action L1 + $0.1\mathcal{L}_{\mathrm{VA}}$
Stage-II learning rate / weight decay	10^{-4} / 0.05
Policy EMA decay	0.9999
Default global training batch	$8 \times 16 = 128$
Main reporting checkpoint	60K policy EMA
Training suites	Spatial, Object, Goal, Long; exclude LIBERO-90
Base Plus execution / episode limit	10 actions / 520 steps
Base Plus language encoding	Online UMT5
Base Plus rendering / precision	MuJoCo 2.3.7, EGL / bf16 AMP

The verifier checkpoint is learned separately before policy training. A shared verifier can supervise policy Base/Large/Giant runs; policy size does not imply verifier size. The dataset constructs residual targets from clean cached features before feature augmentation. At episode boundaries it holds the final available feature frame when the endpoint lies beyond the episode end. Padded actions are masked in imitation and verifier attention. For frame-subsampled data, action timestamps and encoded-frame indices are aligned explicitly.

Compact Training Procedure. (1) Encode demonstration observations with a fixed visual encoder and retain aligned actions. (2) Build the feature-change target at the end of each action chunk. (3) Train only Ψ using demonstrated actions and $\mathcal{L}_{\mathrm{verifier}}$ in Equation 5. (4) Freeze Ψ's parameters. (5) Train a policy on successful demonstrations using Equation 10, retaining gradients through Ψ's action input. (6) Evaluate the selected EMA policy using observations, states, and instructions only.

A.2 RoboTwin Training Recipe

The w/o verifier reference and the w/ verifier Base/Large/Giant policies share these settings; the reference minimizes action imitation alone. The shared Ψ is frozen during policy training, and the verifier gradient passes through predicted actions. Both the verifier and the policies are trained on successful demonstrations only. Clean and Randomized refer to evaluation scenes, rather than separate training datasets.

Table 7: RoboTwin recipe.

Component	Setting
Visual representation	Frozen V-JEPA 2.1 ViT-G, pre-encoded
Training data	`aloha-agilex_clean_50` (50 tasks)
Camera views	3
Policy output	32-step absolute action chunk
Endpoint prediction horizon	32 action steps
Default verifier scale	Giant width, 2 decoder blocks
Verifier training	Clean demonstrations, nominal 100K steps
Stage-II objective	Action L1 + $0.1\mathcal{L}_{\mathrm{VA}}$; L1 only w/o verifier
Stage-II learning rate / weight decay	10^{-4} / 0.05
Policy EMA decay	0.9999
Default global training batch	$8 \times 16 = 128$
Policy training budget	Nominal 100K steps
Main reporting checkpoint	100K policy EMA

A.3 Real-Robot Fine-Tuning Recipe

Table 8: Real-robot fine-tuning recipe.

Component	Setting
Robot	Unitree G1D with Dex1 grippers
Camera views	Head-left, left-wrist, right-wrist
State / action	16 arm-and-gripper joint coordinates / absolute joint positions
Visual features	Frozen, pre-encoded at 5 FPS from 30-Hz demonstrations
Policy	Base policy transformer; V-JEPA 2.1 ViT-G encoder
Demonstrations	58 (toy-to-box) / 150 (object-to-basket)
Training steps / checkpoint	10K per task / 10K EMA
Learning rate	OneCycle, peak 10^{-4}
Weight decay / EMA decay	0.05 / 0.9999
Global training batch	$8 \times 16 = 128$
Action chunk	120 steps
Validation	Held-out split

Each task is fine-tuned separately on its own demonstrations with the settings above.

B Evaluation Protocol and Metrics

This section defines the evaluation settings used by the main-text comparisons. LIBERO contains the Spatial, Object, Goal, and Long suites with 40 tasks and 50 trials per task. LIBERO-Plus evaluates the same policy under seven perturbation categories; the reported Plus-7 score is the unweighted arithmetic mean of the seven category success rates. RoboTwin uses 50 bimanual tasks, 100 trials per task, and reports success separately in Clean and Randomized scenes. Unless stated otherwise, LIBERO verifier results use the 60K EMA checkpoint and RoboTwin results use the recorded 100K checkpoint.

For latency, we report the median single-H20, batch-1, bf16 PyTorch-eager GPU-forward time for the stated camera and language configuration. The verifier is training-only and is excluded from all deployment measurements. External baselines are timed with the same protocol and each method's own input configuration for the benchmark, so LIBERO and RoboTwin latencies can differ; the only exception is AHA-WAM, whose latency is taken from its paper (RTX 5090D).

C Additional Quantitative Results

The following tables provide the numerical breakdowns behind the main-text quantitative comparison. Nominal LIBERO results and the category-balanced Plus-7 calculation are reported separately so that clean performance is not conflated with robustness under perturbation.

Table 9: Base 60K EMA nominal LIBERO suite breakdown. Each suite has 500 trials.

Suite	w/o verifier	w/ verifier
Spatial	478/500 (95.6%)	478/500 (95.6%)
Object	489/500 (97.8%)	500/500 (100.0%)
Goal	480/500 (96.0%)	495/500 (99.0%)
Long	412/500 (82.4%)	427/500 (85.4%)
Overall	1859/2000 (92.95%)	1900/2000 (95.00%)

For category c, let s_c and n_c be successful and total LIBERO-Plus trials. Then

$$\text{Plus7} = \frac{100}{7}\sum_{c=1}^{7}\frac{s_c}{n_c}, \qquad \text{Pooled} = 100\frac{\sum_c s_c}{\sum_c n_c}. \tag{13}$$

The category sizes are 1,599 (camera), 1,550 (robot), 1,537 (language), 1,142 (light), 1,076 (background), 1,601 (noise), and 1,525 (layout). For the base verifier-trained checkpoint, pooled success is $7203/10030 = 71.81\%$, while Plus-7 is 71.73%.

C.1 LIBERO-Plus Breakdown

Table 10 reports the category profile underlying the base-model comparison. Camera and initial-state perturbations probe spatial robustness, while appearance and language changes test complementary sources of shift. Each category is weighted equally in Plus-7, irrespective of its trial count.

Table 10: **Base-model robustness by perturbation category.** Rates (%) for the Base policy w/o and w/ verifier at 60K EMA. Category rates are rounded to one decimal; aggregate Plus-7 values use unrounded category rates.

Perturbation	w/o verifier	w/ verifier	Difference (pp)
Camera viewpoints	65.2	76.1	+10.9
Robot initial states	65.2	79.5	+14.3
Language instructions	57.0	61.1	+4.1
Lighting	74.1	87.7	+13.6
Background textures	54.8	57.2	+2.4
Sensor noise	63.7	72.8	+9.1
Object layouts	58.8	67.7	+8.9
Plus-7	62.68	71.73	+9.05

D Ablation Details

This section expands the main-text ablations with input-processing controls and complete RoboTwin task-level records. These tables document the recorded comparisons; they do not introduce additional causal claims.

D.1 Observation Pipeline Control

Codec matching yields only +1.35 points on nominal LIBERO. Gripper-transition weighting changes the w/o verifier reference from 92.95 to 93.65% on LIBERO and from 62.68 to 62.31%

Table 11: **LIBERO-Plus visual-encoder ablation by perturbation.** Success rates (%) after 20K policy-training steps. These runs correspond to the LIBERO-Plus column of Table 3; the 60K policy results in Table 4 use a longer training budget. Column abbreviations follow Table 1. Bold and underline mark the best and second-best results.

Encoder	Perturbation category							Plus-7
	BG	Cam.	Robot	Lang.	Light	Layout	Noise	
V-JEPA 2.1 (Mur-Labadia et al., 2026)	17.84	**39.65**	**21.81**	**40.73**	**60.42**	**45.11**	**51.03**	**39.51**
SigLIP2 (Tschannen et al., 2025)	**35.04**	14.51	<u>19.16</u>	<u>22.84</u>	<u>36.87</u>	<u>26.36</u>	14.43	<u>24.17</u>
DINOv3 (Siméoni et al., 2025)	<u>22.86</u>	<u>14.57</u>	13.23	15.29	27.41	18.62	<u>15.24</u>	18.17
VGGT (Wang et al., 2025)	17.84	2.31	6.26	11.97	27.06	12.72	5.06	11.89

Table 12: Separate AV1 codec-matching control with an unchanged w/o verifier Base checkpoint. Each suite has 500 trials. The matched frames are encoded and decoded using the training codec before policy input.

Suite	Original input	Codec-matched	Difference (pp)
Spatial	478 (95.6%)	483 (96.6%)	+1.0
Object	489 (97.8%)	493 (98.6%)	+0.8
Goal	480 (96.0%)	486 (97.2%)	+1.2
Long	412 (82.4%)	424 (84.8%)	+2.4
Total	1859 (92.95%)	1886 (94.30%)	+1.35

on Plus-7. Neither input-processing nor loss-reweighting modifications approach the +9.05-point Plus-7 gain from consequence verification.

D.2 RoboTwin Comparison Protocol

The main comparison uses clean-to-clean and clean-to-randomized results from the leaderboard's clean-only training track. Results trained on randomized demonstrations, such as ImageWAM's main RoboTwin scores, follow a different protocol and are therefore not included in Table 2. Joint rows train one policy across tasks; single-task SFT rows use task-specific policies.

Verifier-trained policies obtain Clean 70.18/73.14/77.50% and Randomized 8.34/11.10/17.80% for Base/Large/Giant. Each setting contains 50 tasks with 100 trials per task; the Giant policy succeeds in 3875/5000 Clean and 890/5000 Randomized trials, and the Large policy in 555/5000 Randomized trials.

D.3 Per-Task Results

Tables 13 and 14 report all 50 tasks for the six baseline methods and JEPA-VA-Giant. Baseline entries follow the RoboTwin leaderboard (24 September 2026); JEPA-VA-Giant uses the 100K EMA checkpoint. Each JEPA-VA-Giant task has 100 completed trials in each setting.

D.4 Per-Task Ablations

Tables 15 and 16 report per-task results for the 50-task ablations. V-JEPA 2.1, SigLIP2, DINOv3, and VGGT are the encoder runs of Table 3; the Model Scale columns are w/o verifier Base/Large/Giant runs, and the w/ Verifier columns are the corresponding verifier-trained runs. Each task uses 100 trials per setting.

E Real-World Protocol

The real-robot study uses a Unitree G1D with Dex1 grippers and head-left, left-wrist, and right-wrist cameras. In the toy-to-box task, the robot picks up a toy and places it inside a box. In the advanced

Table 13: **RoboTwin clean-to-clean success by task.** Rates (%). Each task uses 100 trials. Bold and underline mark the best and second-best results in each row.

Task	DP3	GalaxeaVLA	RDT	π_0	AHA-WAM	FastWAM	JEPA-VA-Giant
Adjust Bottle	99	**100**	81	90	**100**	99	**100**
Beat Block Hammer	72	82	77	43	52	82	**95**
Blocks Ranking RGB	3	42	3	19	60	81	**90**
Blocks Ranking Size	2	19	0	7	32	52	**57**
Click Alarmclock	77	79	61	63	98	99	**100**
Click Bell	90	77	80	44	**100**	95	99
Dump Bin Bigbin	85	88	64	83	90	94	**98**
Grab Roller	98	**100**	74	96	98	**100**	97
Handover Block	70	10	45	45	63	66	**88**
Handover Mic	**100**	85	90	98	99	98	99
Hanging Mug	17	20	23	11	13	14	**34**
Lift Pot	**97**	75	72	84	62	91	96
Move Can Pot	70	58	25	58	48	**96**	74
Move Pillbottle Pad	41	69	8	21	33	**92**	**92**
Move Playingcard Away	68	88	43	53	82	**94**	90
Move Stapler Pad	12	8	2	0	5	33	**42**
Open Laptop	82	74	59	85	**89**	84	74
Open Microwave	61	41	37	**80**	59	17	36
Pick Diverse Bottles	52	62	2	27	36	72	**79**
Pick Dual Bottles	60	77	42	57	60	**87**	86
Place A2B Left	46	56	3	31	65	**69**	57
Place A2B Right	49	47	1	27	50	**60**	54
Place Bread Basket	26	77	10	17	54	**87**	82
Place Bread Skillet	19	70	5	23	68	**85**	83
Place Burger Fries	72	90	50	80	96	97	**99**
Place Can Basket	67	67	19	41	61	**77**	70
Place Cans Plasticbox	48	89	6	34	90	98	**100**
Place Container Plate	86	97	78	88	99	**100**	98
Place Dual Shoes	13	44	4	15	48	**67**	58
Place Empty Cup	65	91	56	37	64	90	**98**
Place Fan	36	41	12	20	58	59	**62**
Place Mouse Pad	4	19	1	7	12	**55**	33
Place Object Basket	65	70	33	16	71	82	**84**
Place Object Scale	15	35	1	10	34	**83**	59
Place Object Stand	60	79	15	36	79	**94**	91
Place Phone Stand	44	53	15	35	61	85	**87**
Place Shoe	58	67	35	28	74	**87**	82
Press Stapler	69	74	41	62	82	**90**	68
Put Bottles Dustbin	60	54	21	54	58	55	**79**
Put Object Cabinet	**72**	37	33	68	36	42	42
Rotate QRcode	74	66	50	68	61	**81**	77
Scan Object	31	42	4	18	51	**68**	**68**
Shake Bottle Horizontally	**100**	**100**	84	99	**100**	**100**	**100**
Shake Bottle	98	**100**	74	97	**100**	**100**	**100**
Stack Blocks Three	1	28	2	17	57	**89**	71
Stack Blocks Two	24	85	21	42	89	**99**	97
Stack Bowls Three	57	59	51	66	71	**80**	78
Stack Bowls Two	83	92	76	91	93	**94**	**94**
Stamp Seal	18	33	1	3	13	30	**46**
Turn Switch	**46**	20	35	27	42	39	32
Mean	55.24	62.72	34.50	46.42	64.32	**77.76**	77.50

Table 14: **RoboTwin clean-to-randomized success by task.** Rates (%). Each task uses 100 trials. Bold and underline mark the best and second-best results in each row.

Task	DP3	GalaxeaVLA	RDT	π_0	AHA-WAM	FastWAM	JEPA-VA-Giant
Adjust Bottle	3	10	**75**	56	0	0	4
Beat Block Hammer	8	31	**37**	21	0	0	3
Blocks Ranking RGB	0	1	0	**5**	1	0	3
Blocks Ranking Size	0	0	0	1	0	0	**3**
Click Alarmclock	14	7	12	11	17	**60**	55
Click Bell	0	27	9	3	19	7	**55**
Dump Bin Bigbin	**53**	40	32	24	27	0	35
Grab Roller	2	33	43	**80**	3	2	53
Handover Block	0	0	**14**	8	1	0	0
Handover Mic	3	2	**31**	13	3	0	3
Hanging Mug	1	1	**16**	3	0	0	0
Lift Pot	0	0	9	**36**	0	0	2
Move Can Pot	6	5	12	**21**	0	0	0
Move Pillbottle Pad	0	**2**	0	1	0	0	**2**
Move Playingcard Away	3	13	11	**22**	0	0	2
Move Stapler Pad	0	1	0	**2**	0	0	0
Open Laptop	7	13	32	**46**	9	0	4
Open Microwave	22	13	20	**50**	1	0	6
Pick Diverse Bottles	1	11	0	6	0	0	**33**
Pick Dual Bottles	1	21	13	12	1	1	**44**
Place A2B Left	2	**5**	1	1	1	0	2
Place A2B Right	0	3	1	**6**	0	1	**6**
Place Bread Basket	1	**32**	2	4	3	0	31
Place Bread Skillet	0	**14**	1	1	0	0	**14**
Place Burger Fries	18	32	27	4	0	0	**82**
Place Can Basket	2	**6**	**6**	5	2	0	3
Place Cans Plasticbox	3	22	5	2	1	0	**50**
Place Container Plate	1	38	17	**45**	11	0	35
Place Dual Shoes	0	9	4	0	0	0	**14**
Place Empty Cup	1	3	7	11	2	0	**22**
Place Fan	1	2	2	**10**	0	0	0
Place Mouse Pad	**1**	**1**	0	**1**	0	0	0
Place Object Basket	0	3	**17**	2	0	0	3
Place Object Scale	0	2	0	0	0	0	**4**
Place Object Stand	0	**27**	5	11	0	0	22
Place Phone Stand	2	**8**	6	7	0	0	1
Place Shoe	2	**22**	7	6	5	0	9
Press Stapler	3	17	24	29	2	3	**44**
Put Bottles Dustbin	21	3	4	13	1	0	**30**
Put Object Cabinet	1	5	**18**	**18**	0	0	**18**
Rotate QRcode	1	2	5	**15**	0	0	2
Scan Object	1	**6**	1	1	0	0	5
Shake Bottle Horizontally	25	46	51	51	15	9	**56**
Shake Bottle	19	45	45	**60**	14	10	59
Stack Blocks Three	0	**1**	0	0	**1**	0	**1**
Stack Blocks Two	0	**9**	2	1	2	0	6
Stack Bowls Three	5	5	17	**24**	0	0	18
Stack Bowls Two	6	27	30	**41**	5	0	34
Stamp Seal	0	2	0	**4**	0	0	1
Turn Switch	8	8	15	**23**	13	3	11
Mean	4.96	12.72	13.72	16.34	3.20	1.92	**17.80**

Table 15: **RoboTwin clean-to-clean ablations by task.** Success rates (%).

Task	Vision Encoder				Model Scale			w/ Verifier		
	V-JEPA	SigLIP2	DINOv3	VGGT	Base	Large	Giant	Base	Large	Giant
Adjust Bottle	59	69	74	2	100	100	100	100	99	100
Beat Block Hammer	33	15	28	8	88	84	92	60	85	95
Blocks Ranking Rgb	7	10	15	0	79	76	87	73	75	90
Blocks Ranking Size	6	1	1	0	41	44	46	29	45	57
Click Alarmclock	90	59	82	68	97	98	92	96	97	100
Click Bell	75	59	84	25	74	92	93	98	94	99
Dump Bin Bigbin	76	72	62	11	100	98	96	96	98	98
Grab Roller	86	83	85	29	100	98	98	99	98	97
Handover Block	0	0	1	1	54	77	75	55	64	88
Handover Mic	44	42	69	37	99	98	98	97	99	99
Hanging Mug	1	1	1	2	21	35	45	45	32	34
Lift Pot	5	6	3	10	98	98	93	97	94	96
Move Can Pot	16	4	14	9	66	73	71	78	74	74
Move Pillbottle Pad	25	11	5	1	86	79	87	84	85	92
Move Playingcard Away	29	35	46	2	85	86	86	83	89	90
Move Stapler Pad	4	0	0	0	19	40	42	23	30	42
Open Laptop	41	42	67	27	88	87	86	73	85	74
Open Microwave	15	41	45	37	76	59	61	6	39	36
Pick Diverse Bottles	10	4	4	2	58	53	78	62	61	79
Pick Dual Bottles	11	3	7	19	74	72	78	81	72	86
Place A2B Left	16	11	9	2	41	33	52	41	43	57
Place A2B Right	14	21	6	3	34	33	49	29	33	54
Place Bread Basket	44	21	23	5	75	82	85	76	78	82
Place Bread Skillet	23	32	23	0	85	83	75	83	92	83
Place Burger Fries	60	29	44	5	98	96	98	99	93	99
Place Can Basket	30	15	25	25	64	69	66	70	74	70
Place Cans Plasticbox	9	14	9	3	98	100	100	100	97	100
Place Container Plate	90	82	69	58	98	100	99	96	99	98
Place Dual Shoes	4	3	6	1	53	56	55	49	56	58
Place Empty Cup	61	25	35	4	94	96	89	97	96	98
Place Fan	6	7	4	0	45	51	60	52	53	62
Place Mouse Pad	5	5	0	0	23	27	33	25	34	33
Place Object Basket	35	37	33	14	82	85	87	70	79	84
Place Object Scale	13	8	7	0	38	46	54	42	47	59
Place Object Stand	39	34	25	10	78	78	88	85	76	91
Place Phone Stand	31	10	31	6	74	80	81	75	68	87
Place Shoe	9	11	10	3	75	78	83	67	84	82
Press Stapler	14	32	57	32	66	74	83	56	79	68
Put Bottles Dustbin	5	5	0	6	69	57	85	62	74	79
Put Object Cabinet	4	1	3	17	31	37	44	60	37	42
Rotate Qrcode	24	32	33	4	73	77	79	78	75	77
Scan Object	29	18	9	3	69	77	74	71	74	68
Shake Bottle	90	87	76	86	100	99	100	100	99	100
Shake Bottle Horizontally	90	84	73	75	99	99	100	100	99	100
Stack Blocks Three	1	0	1	0	41	45	75	39	55	71
Stack Blocks Two	6	2	14	1	96	80	99	87	88	97
Stack Bowls Three	54	43	40	5	71	73	79	74	72	78
Stack Bowls Two	86	89	79	38	99	98	96	97	94	94
Stamp Seal	5	1	2	0	40	42	39	39	35	46
Turn Switch	43	24	17	34	57	63	49	55	59	32
Mean	31.46	26.80	29.12	14.60	71.38	73.22	77.20	70.18	73.14	77.50

Table 16: **RoboTwin clean-to-randomized ablations by task.** Success rates (%) from the internal experiment record; these are not leaderboard submissions. Encoder runs and scale/verifier runs use separately recorded configurations. The w/ verifier Large column is the completed 100K EMA phase-2 evaluation with 100 trials per task.

Task	Vision Encoder				Model Scale			w/ Verifier		
	V-JEPA	SigLIP2	DINOv3	VGGT	Base	Large	Giant	Base	Large	Giant
Adjust Bottle	3	15	1	0	0	0	4	0	0	4
Beat Block Hammer	2	4	2	5	0	1	2	1	1	3
Blocks Ranking Rgb	0	0	0	0	0	1	2	0	1	3
Blocks Ranking Size	0	0	0	0	1	0	0	0	0	3
Click Alarmclock	31	41	26	9	43	32	48	42	55	55
Click Bell	24	20	15	3	9	17	44	14	39	55
Dump Bin Bigbin	37	23	16	19	5	38	24	40	49	35
Grab Roller	18	25	10	19	6	6	1	1	8	53
Handover Block	0	0	0	0	0	0	0	0	0	0
Handover Mic	1	2	0	2	1	1	3	3	1	3
Hanging Mug	0	1	0	5	0	1	0	2	0	0
Lift Pot	0	0	1	0	0	0	0	0	0	2
Move Can Pot	0	0	1	2	1	1	0	1	1	0
Move Pillbottle Pad	0	1	0	0	0	0	0	2	0	2
Move Playingcard Away	3	1	0	2	0	0	0	0	0	2
Move Stapler Pad	0	0	0	0	0	0	0	0	1	0
Open Laptop	7	10	10	10	5	2	2	8	4	4
Open Microwave	9	16	10	13	3	8	4	8	6	6
Pick Diverse Bottles	1	2	1	0	5	8	24	8	11	33
Pick Dual Bottles	3	0	1	3	7	8	38	6	15	44
Place A2B Left	5	5	0	1	5	3	9	2	2	2
Place A2B Right	2	5	0	0	0	2	4	2	2	6
Place Bread Basket	4	6	5	4	6	6	22	9	6	31
Place Bread Skillet	1	1	0	0	1	6	8	3	5	14
Place Burger Fries	1	3	9	2	8	7	35	15	30	82
Place Can Basket	0	2	2	9	2	11	1	4	0	3
Place Cans Plasticbox	4	0	1	0	7	8	35	8	26	50
Place Container Plate	29	31	21	21	44	38	21	44	32	35
Place Dual Shoes	0	0	0	1	5	7	19	5	11	14
Place Empty Cup	8	7	9	4	15	15	12	23	10	22
Place Fan	0	0	0	0	0	0	1	0	1	0
Place Mouse Pad	0	1	0	0	0	0	0	0	0	0
Place Object Basket	0	7	1	5	0	3	4	5	2	3
Place Object Scale	0	2	0	0	1	1	2	2	3	4
Place Object Stand	6	10	2	5	10	6	16	13	18	22
Place Phone Stand	1	3	1	0	0	1	2	1	0	1
Place Shoe	1	4	0	1	10	5	10	3	5	9
Press Stapler	11	9	13	10	12	21	41	12	35	44
Put Bottles Dustbin	2	4	1	2	2	8	14	7	10	30
Put Object Cabinet	1	5	0	2	2	2	12	3	5	18
Rotate Qrcode	0	0	0	2	0	1	0	0	0	2
Scan Object	1	1	0	1	2	3	6	0	4	5
Shake Bottle	35	34	15	33	39	44	50	45	59	59
Shake Bottle Horizontally	35	28	17	23	35	41	53	40	57	56
Stack Blocks Three	0	0	0	0	0	0	0	0	0	1
Stack Blocks Two	0	0	0	0	0	3	4	0	3	6
Stack Bowls Three	3	4	2	0	6	14	4	7	5	18
Stack Bowls Two	17	17	7	5	28	32	16	18	28	34
Stamp Seal	1	0	0	0	0	0	0	0	0	1
Turn Switch	23	18	7	24	8	4	3	10	4	11
Mean	6.60	7.36	4.14	4.94	6.68	8.32	12.00	8.34	11.10	17.80

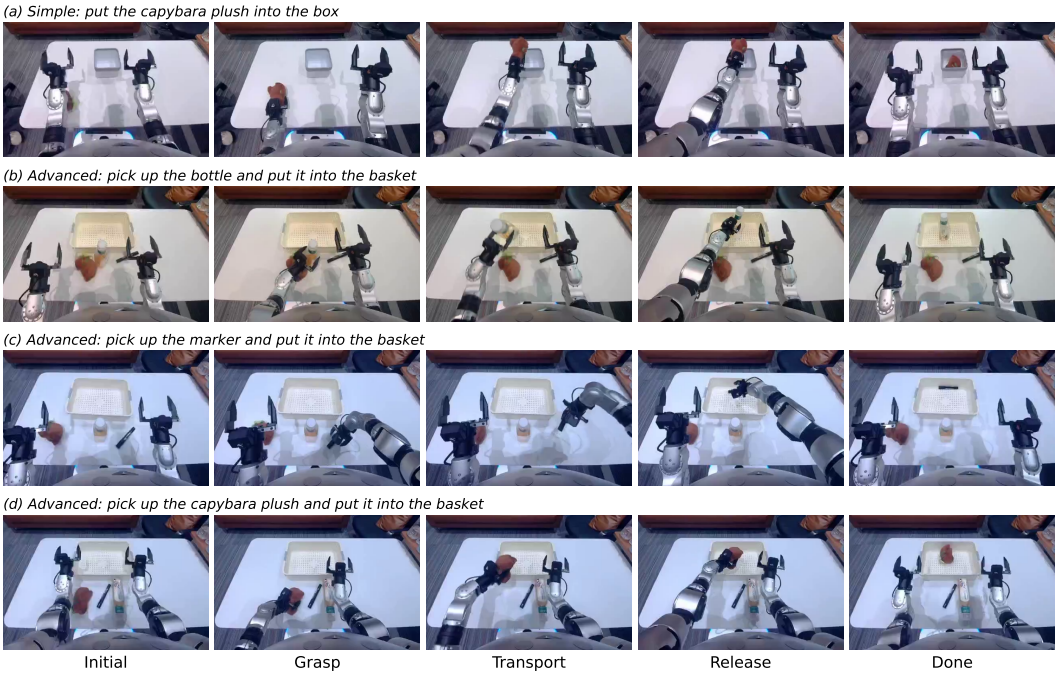


Figure 5: **Real-robot tasks on the Unitree G1D.** Simple: place a toy into a box. Advanced: three instructions, each asking the robot to pick a different object and place it into the basket.

Table 17: **Per-instruction object-to-basket results.** Counts over 7 trials per instruction and the weighted score $(20N_{\text{reach}} + 30N_{\text{grasp}} + 50N_{\text{success}})/7$. Bold and underline mark the best and second-best results within each instruction.

Instruction	Method	Reach	Grasp	Success	Score
Doll (easy)	FastWAM	**7**	<u>3</u>	<u>3</u>	<u>54.29</u>
	ACT	**7**	2	1	35.71
	JEPA-VA	<u>6</u>	**4**	**4**	**62.86**
Bottle (medium)	FastWAM	<u>6</u>	<u>1</u>	<u>1</u>	<u>28.57</u>
	ACT	5	<u>1</u>	<u>1</u>	25.71
	JEPA-VA	**7**	**3**	**3**	**54.29**
Marker (hard)	FastWAM	<u>2</u>	0	0	<u>5.71</u>
	ACT	0	0	0	0.00
	JEPA-VA	**3**	0	0	**8.57**

object-to-basket task, three objects are present and each of three instructions names a different object to pick and place into the basket, requiring instruction-conditioned target selection.

Table 17 breaks the object-to-basket results down by instruction. The three target objects differ in grasp difficulty: the doll is easiest, the bottle is intermediate, and the thin marker is hardest. Each instruction is evaluated for 7 trials, and the score uses the same weighting as Table 5. JEPA-VA has the highest score for every instruction; its largest margin is on the bottle, where it succeeds in 3 of 7 trials compared with 1 for each baseline. No method completes the marker placement.

F Reproducibility and Provenance

All LIBERO-Plus results use the category-balanced Plus-7 metric. For π_0, the benchmark reports a pooled rate of 53.6% and a category mean of 56.1%; we use the latter. The OpenVLA-OFT row (OFT-m) is the mixed-suite variant with a 69.9% category mean. For FastWAM and Image-WAM (Ovis-U1), we recompute the category mean from the published per-category rates, obtaining

51.44% and 71.41% (marked † in Table 1). RoboTwin baselines follow the machine-readable leaderboard (RoboTwin Team, 2026); their 50-task means are recomputed from its per-task entries.