CAUSALLY DEBIASED LATENT ACTION MODEL FOR EMBODIED ACTION-CONDITIONED WORLD MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Action-conditioned world models (ACWMs) aim to simulate future observations conditioned on embodied actions. However, learning controllable ACWMs requires large-scale action-labeled data, which remains costly to collect. Latent action models (LAMs) mitigate this bottleneck by inferring latent actions from videos without executable action labels, but existing LAMs are typically trained with reconstruction-only objectives and therefore entangle action-relevant dynamics with action-irrelevant visual confounders. With such objectives, LAM favors visual context over action dynamics, therefore downstream ACWMs exhibit residual motion under zero actions and fail to reproduce supplied dynamics. We propose CD-LAM, a causally debiased framework for LAM-based ACWMs. CD-LAM introduces three debiasing objectives: embodiment-centric reconstruction, action-centric contrastive learning, and latent space calibration, which together encourage embodiment-focused, action-aware, and well-calibrated representations. CD-LAM reduces action-following error by up to 42% and 35% when conditioned on latent and robot actions, while improving fidelity, and matching DreamDojo reference with over 12$\times$ fewer adaptation updates. CD-LAM also adapts LTX-2.3-22B into an ACWM with competitive quality using only 3.2% of DreamDojo-14B's sample exposures. On X-VLA, pretraining with CD-LAM instead consistently improves success rates by up to 5% across LIBERO, LIBERO-Plus, and RoboTwin C2R.

1 INTRODUCTION

Action-conditioned world models (ACWMs) (Ha & Schmidhuber, 2018; Hafner et al., 2024; Xue et al., 2026; Wu et al., 2024) have emerged as a promising paradigm for simulating the physical world directly from visual observations and embodied control signals. Given the current observation and an action trajectory, an ACWM aims to forecast the resulting future observation sequence, thereby simulating how the environment and embodiment would evolve under that intervention. By predicting futures under different candidate action trajectories, ACWMs have the potential to serve as general-purpose simulators for robot planning, policy evaluation, and data augmentation. However, the controllability of ACWMs relies on large-scale action-labeled data, yet collecting robot videos with action annotations remains costly in the real world (Khazatsky et al., 2024). In contrast, human videos are easier to collect, far more scalable, and rich in physical interactions, but they typically lack executable action annotations (Grauman et al., 2022; 2025; Damen et al., 2021). Latent action models (LAMs) offer a bridge to such large-scale data sources by inferring compact latent action representations from unlabeled video transitions (Bruce et al., 2024; Schmidt & Jiang, 2024; Ye et al., 2025). Through large-scale pre-training conditioned on these latent actions, ACWMs can be effectively warmed up to learn action-controllable dynamics and achieve stronger action-following performance after fine-tuning on limited labeled robot data (Gao et al., 2025; 2026).

Despite this promise, existing LAMs are typically trained solely with the video reconstruction objective, which cannot prevent action-irrelevant visual factors from being encoded into the latent actions. In particular, action-irrelevant factors such as background scenes and non-interacted objects may leak into the latent action representation. Our analysis in Section 3 shows that such action-irrelevant factors induce biased latent action representations, which ultimately confound the ACWM: the model can generate visually plausible rollouts, but fails to reliably follow the supplied actions (Zhang et al., 2025a; Nikulin et al., 2025; Dai et al., 2026). The inclusion of these confounders in the action condition can leave the ACWM with a biased state for robot-action adaptation and

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.

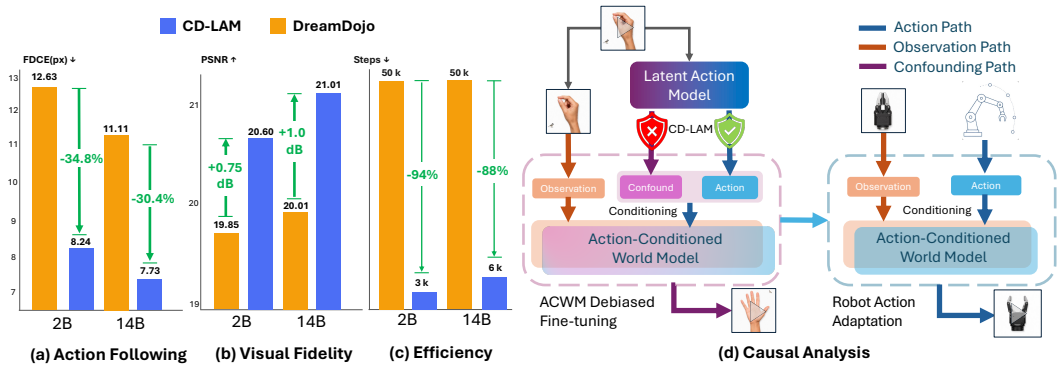


Figure 1: **Performance overview and confounding analysis.** CD-LAM lowers action-following error (FDCE) and raises PSNR at both 2B and 14B scales, with much lower training cost than DreamDojo. A reconstruction-trained LAM can leak action-irrelevant confounders into the action condition, and CD-LAM reduces this shortcut dependence.

impose an additional alignment burden. We introduce latent-space diagnostics and intervention-based evaluations to visualize and quantify such bias. Our diagnostic results reveal substantial responses to small camera shifts and shortcuts that favor visual context over action similarity. Evaluation results further expose weak action following under both latent-action and robot-action conditioning, including residual motion under zero-action inputs and failure to reproduce target dynamics.

Motivated by these findings, we propose CD-LAM to address action-irrelevant confounding in LAM-based ACWMs by explicitly shaping the latent action space. Specifically, it combines embodiment-centric reconstruction to emphasize the embodiment and interacted objects over background, action-centric contrastive learning to group transitions by coarse primitives across visual confounders and mitigate visual-context-driven shortcuts, and latent space calibration to suppress spurious responses to unchanged observations. CD-LAM consists of an efficient three-stage fine-tuning pipeline (Figure 3) that first debiases the LAM, then the ACWM, and finally adapts the ACWM to real-world robot actions. In Stage 1, we fine-tune the LAM with these three debias objectives to emphasize embodiment-centric action dynamics. In Stage 2, we fine-tune the ACWM on debiased latent actions extracted from human videos without executable action labels, to debias the learned controllability and improve action following under the corrected latent conditions. In Stage 3, we add a lightweight action-to-latent-action mapping layer to bridge robot actions into the debiased latent space as the conditioning input and adapt the ACWM to executable robot controls.

Experiments on 2B and 14B backbones show that upstream latent-action debiasing improves downstream world-model controllability (Figure 1; Section 5). Before robot-action alignment, CD-LAM reduces action-following error, measured by Foreground Displacement Chamfer Error (FDCE), by 42% and 26%, respectively. These gains persist after robot action adaptation, with FDCE reductions of 35% and 30% alongside improved visual fidelity. The complete pipeline also matches the released DreamDojo reference with more than 12$\times$ fewer robot action adaptation updates. Beyond DreamDojo, CD-LAM enables efficient adaptation of the pretrained 22B LTX-2.3 video backbone (Lightricks, 2026) into an ACWM conditioned on executable robot actions. The resulting model achieves comparable rollout quality while using 3.2% of the sample exposures reported for DreamDojo-14B across latent-action finetuning and robot-action adaptation. For policy learning, pretraining X-VLA (Zheng et al., 2026) with CD-LAM rather than the original LAM improves success rates across LIBERO, LIBERO-Plus and RoboTwin C2R (Liu et al., 2023; Fei et al., 2026; Chen et al., 2025a).

This paper makes three contributions:

- We identify and analyze a confounding mechanism in LAM-based ACWMs: action-irrelevant visual factors enter the latent action condition and undermine downstream controllability.
- We propose CD-LAM, a causally motivated framework that targets this bias at its source by learning debiased action representations for world-model fine-tuning and executable robot-action alignment.
- Our CD-LAM yields persistent improvements on action-conditioned world models DreamDojo-2B/14B and LTX-2.3-22B, also improving policy models in three robot manipulation benchmarks.

2 PRELIMINARIES

Action-conditioned world models. An action-conditioned world model (ACWM) predicts future observations under an action sequence:

$$p_\theta(o_{t+1:t+H} \mid o_{\le t}, u_{t:t+H-1}), \tag{1}$$

where o_t denotes the visual observation at time t and u_t denotes a recorded executable robot action, such as a relative end-effector displacement.

Latent action models. Given adjacent frames $\langle o_t, o_{t+1}\rangle$, the LAM infers a latent action vector $z_t \in \mathbb{R}^d$, i.e., a summary of the observed action-related information (Bruce et al., 2024; Schmidt & Jiang, 2024; Ye et al., 2025). A LAM can be written as an encoder paired with a decoder:

$$z_t \sim q_\phi(z \mid o_t, o_{t+1}), \qquad \hat{o}_{t+1} = D_\psi(o_t, z_t). \tag{2}$$

We write $\mu_\phi(o_t, o_{t+1})$ for the posterior mean of q_ϕ, used whenever a deterministic latent action is needed. The standard LAM objective is next-frame reconstruction, optionally with a bottleneck or prior regularizer:

$$\mathcal{L}_{\mathrm{LAM}} = \mathbb{E}\big[\ell(D_\psi(o_t, z_t), o_{t+1})\big] + \mathrm{Reg}(z_t). \tag{3}$$

Here ℓ is the observation reconstruction loss, and Reg denotes the capacity-control mechanism used by the particular LAM, such as a KL bottleneck. This objective defines z_t by information that helps reconstruct or predict the observed transition.

Latent action conditioning. Prior LAM-based ACWM systems use latent actions as substitutes for missing robot actions during ACWM training on egocentric video: transitions are encoded as $(o_t, o_{t+1}) \mapsto z_t$, and the world model is trained as

$$p_\theta(o_{t+1:t+H} \mid o_{\le t}, z_{t:t+H-1}). \tag{4}$$

This pipeline requires two properties. The latent action should capture the embodied action in the observed transition, and the world model should make future motion follow the latent action condition (Gao et al., 2025; 2026). Note that z_t plays a different role from u_t: a recorded robot action u_t is an executable control signal defined by an embodiment and its controller, whereas z_t is a video-derived conditioning variable defined solely by the LAM training objective.

3 CONFOUNDER ANALYSIS ON LATENT ACTIONS

In a LAM-based ACWM, z_t is the only action signal the world model receives during ACWM training, so any bias in z_t becomes a bias of the action condition itself. We first analyze why the reconstruction objective admits such bias, then verify the resulting failures in existing ACWMs.

3.1 WHY A RECONSTRUCTION-CENTRIC LOSS ADMITS CONFOUNDERS

For embodied ACWMs, z_t should act as the control signal whose dominant variation reflects key embodiment dynamics: body and object motion, contact, and displacement. A reconstruction-trained LAM does not enforce this property. Its objective only requires predictive sufficiency, i.e., that z_t helps model $p(o_{t+1} \mid o_t, z_t)$, so any transition-predictive factor can enter the latent:

$$z_t = \mu_\phi(o_t, o_{t+1}) \approx f(A_t, C_t, V_t), \tag{5}$$

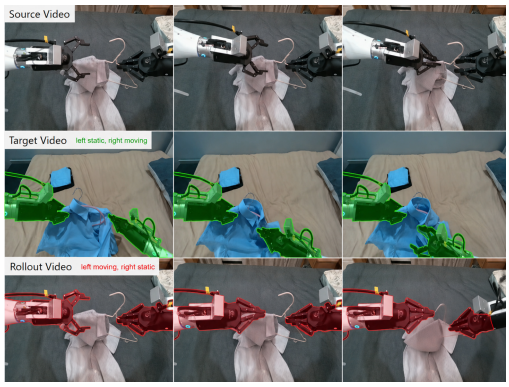
where A_t denotes embodied action effects, C_t denotes scene context, and V_t denotes source-side visual factors such as background continuation and appearance continuity. The notation $f(\cdot)$ records that nothing in the objective prevents z_t from carrying non-trivial dependence on C_t and V_t in addition to A_t. This induces ***action-irrelevant confounding***, illustrated in Figure 1(d). Scene context and source-side visual factors are valid evidence in the observation history, where they help render the current scene. Once encoded into z_t, however, they enter the action side of the model, and the condition partly specifies video continuation rather than embodiment dynamics. Robot action adaptation then inherits this contaminated latent space. A robot action u_t can ground the embodied dynamics in A_t, but it cannot specify source-specific context, background continuation, or camera-like variation, so executable robot actions are forced into alignment with a partly non-actionable latent. This mismatch can increase the difficulty of robot-action alignment.



(a) Zero-action input.

Table 1: **LAM encoder confounding audit**. A clean LAM encoder barely responds to action irrelevant changes (more details at Table C.1).

Diagnostic	DreamDojo	CD-LAM
Zero-transition (static pair (o_t, o_t))		
Rel. norm, median↓	0.527	**0.043**
Abs. norm, median↓	3.119	**0.226**
Camera shift (rel. norm, mean/median)		
Horizontal↓	0.555/0.536	**0.156/0.096**
Vertical↓	0.545/0.529	**0.110/0.064**
Action-neighbor structure		
Shortcut leakage↓	0.151	**0.014**



(b) Transferred target action.

Figure 2: **Action-following failures under zero-input and target-action probes.** The initial frame is fixed and only the action is replaced. (a) Zero action: the rollout still moves. (b) Actions transferred from a target video (middle, green): the rollout (bottom, red) misses the target dynamics.

3.2 Empirical Study on a Representative LAM-based ACWM

We next check whether the analyzed failure appears in practice, using DreamDojo (Gao et al., 2026) as a representative LAM-based ACWM. The evidence has two layers:

Weak Robot Action Following. After robot action adaptation, the generated embodiment dynamics should follow the conditioning action. However, DreamDojo ACWMs tend to violate this requirement in two basic settings, both of which keep the initial frame fixed and replace only the conditioning action. First, in the zero-action setting, we replace the conditioning action with zero, denoted $\mathrm{do}(u_t = 0)$. Although the embodiment motion should be suppressed, the rollout keeps moving (Figure 2a). Second, under target-action transfer, where the conditioning action is replaced with one taken from a different target video, the rollout should reproduce the target embodiment dynamics, yet it does not (Figure 2b). These examples indicate that the ACWM does not reliably follow the supplied robot actions. Both settings are interventions in the causal sense: the observation history is held fixed and only the action condition is replaced. These probes test the model's response to replacing the action condition while keeping the observation fixed, turning the disentanglement that Equation (5) calls for into a behavioral test, and Section 5.4 applies the same two interventions.

Action-irrelevant Confounding in Latent Actions. We further examine the latent action space directly. A clean latent action space should satisfy three basic properties: (1) the zero-transition latent $z_t^0 = \mu_\phi(o_t, o_t)$, encoded from a duplicated-frame pair, should remain close to zero; (2) camera-like image shifts should induce only small latent responses; and (3) local neighborhoods should reflect action similarity rather than shared visual context. To quantify these three properties, we design corresponding evaluation metrics (Appendix C) and apply them to the DreamDojo LAM encoder. As shown in Table 1, the DreamDojo LAM falls short of these desired properties and exhibits substantial bias: duplicated-frame pairs and synthetic camera shifts produce large latent responses, while same-episode visual context pulls latents closer even when their action primitives differ. The DreamDojo latent actions are thus significantly biased by action-irrelevant confounders.

4 CD-LAM: Causal Debiasing of Latent Actions

Section 3 shows that reconstruction-only LAM training can confound latent actions. We propose CD-LAM, a causally motivated debiasing approach with an efficient staged fine-tuning pipeline.

4.1 Design Principles

The analysis in Section 3 leads to three design principles for keeping action-irrelevant factors out of the latent action space while visual context stays available through the observation path.

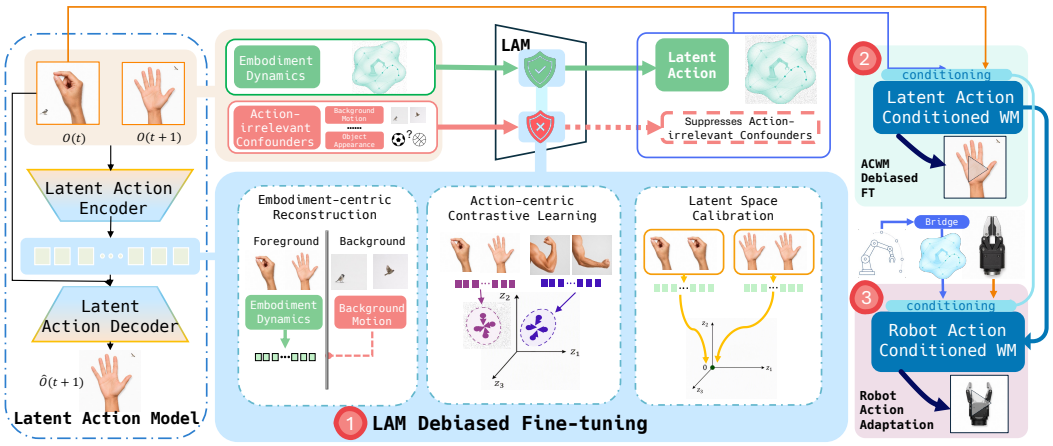


Figure 3: **Overview of CD-LAM.** CD-LAM debiases the LAM's latent action space in three stages while keeping the downstream action conditioning format unchanged. **Stage 1 (LAM debiased fine-tuning)** debiases the LAM with the three CD-LAM objectives. **Stage 2 (ACWM debiased fine-tuning)** trains the ACWM on the debiased latent actions. **Stage 3 (robot action adaptation)** aligns executable robot actions to the same space through a lightweight bridge.

• **Embodiment-centric Reconstruction.** The reconstruction signal should emphasize regions tied to embodiment dynamics over background regions.
• **Action-centric Structure.** The latent action space should group videos with similar actions, not visually similar videos with different actions.
• **Calibrated Latent Space.** Duplicated-frame inputs should map to a designated zero-transition reference, while ordinary transitions retain enough variation to encode embodiment dynamics.

CD-LAM implements these principles with three matching LAM objectives: embodiment-centric reconstruction ($\mathcal{L}_{\text{emb}}$), action-centric contrastive learning ($\mathcal{L}_{\text{ctr}}$), and latent space calibration ($\mathcal{L}_{\text{cal}}$).

4.2 CD-LAM Objective

Given a transition (o_t, o_{t+1}), CD-LAM produces a debiased latent action

$$z_t^{\text{CD}} = \mu_\phi(o_t, o_{t+1}) \in \mathbb{R}^d, \tag{6}$$

which is fed into the same ACWM conditioning format as the original DreamDojo latent. The LAM is fine-tuned with

$$\mathcal{L}_{\text{CD}} = \mathcal{L}_{\text{emb}} + \lambda_{\text{ctr}}(k)\mathcal{L}_{\text{ctr}} + \mathcal{L}_{\text{cal}}. \tag{7}$$

Here $\lambda_{\text{ctr}}(k)$ weights the contrastive term and varies with the training step k (we reserve t for frame time); the calibration term carries its own weights (Equation (10)).

4.2.1 Embodiment-centric Reconstruction Loss

We use an embodiment-centric weighted MSE for frame-pair reconstruction. Given a transition (o_t, o_{t+1}), CD-LAM predicts $\hat{o}_{t+1} = D_\psi(o_t, z_t^{\text{CD}})$ as in Equation (2). Let $M_t \in [0,1]^{h\times w}$ be the embodiment–object foreground mask obtained by SAM3 (Carion et al., 2026), where $h \times w$ is the frame resolution. We optimize the weighted reconstruction loss

$$\mathcal{L}_{\text{emb}} = \frac{1}{|\Omega|}\sum_{p\in\Omega} W_t(p)\left\|\hat{o}_{t+1}(p) - o_{t+1}(p)\right\|_2^2, \qquad W_t = \alpha_{\text{fg}} M_t + \alpha_{\text{bg}}(1 - M_t). \tag{8}$$

where Ω is the pixel grid and $\alpha_{\text{fg}} > \alpha_{\text{bg}}$: embodiment and object errors cost more than background errors, while the nonzero background weight keeps global visual consistency.

4.2.2 Action-centric Contrastive Learning

Reconstruction alone treats each transition independently. CD-LAM adds a pairwise contrastive loss over coarse manipulation primitives so that action-consistent transitions remain close across visual

contexts. Let $v_i = \mathrm{norm}(r_\omega(z_i^{\mathrm{CD}}))$ with an auxiliary head r_ω, let $s_{ij} = \tau\, v_i^\top v_j + b$ be their scaled cosine similarity, and let $\mathcal{P}_\pm$ be the valid same-/different-primitive pairs in a batch:

$$\mathcal{L}_{\mathrm{ctr}} = \underbrace{\frac{1}{W_+} \sum_{(i,j)\in\mathcal{P}_+} w_{ij}\, \mathrm{softplus}(-s_{ij})}_{\text{pull same primitive together}} + \underbrace{\frac{1}{W_-} \sum_{(i,j)\in\mathcal{P}_-} w_{ij}\, \mathrm{softplus}(s_{ij})}_{\text{push different primitives apart}}. \tag{9}$$

Each term is normalized by its own total pair weight, $W_\pm = \max\big(1, \sum_{(i,j)\in\mathcal{P}_\pm} w_{ij}\big)$, and batches without positives contribute zero. The learned scale τ, bias b (Zhai et al., 2023), and pair weights w_{ij} are specified in the released configuration. The primitive labels are coarse verb-level categories (e.g., pick–place, pour, open, close) obtained by mapping the datasets' task and verb annotations into a coarse primitive space (Appendix D) without involving any executable robot actions. Transitions without annotations still contribute to the reconstruction and calibration objectives.

4.2.3 Latent Space Calibration

The latent space calibration loss combines capacity control with zero-transition calibration:

$$\mathcal{L}_{\mathrm{cal}} = \beta_{\mathrm{kl}}(k)\, \mathcal{L}_{\text{KL-fb}} + \lambda_{\mathrm{zero}}\, \mathcal{L}_{\mathrm{zero}}, \qquad \mathcal{L}_{\mathrm{zero}} = \mathbb{E}_{o_t}\big[\, [\rho_t - m_{\mathrm{zero}}]_+^2 \,\big]. \tag{10}$$

The free-bit KL term $\mathcal{L}_{\text{KL-fb}}$ is a KL penalty applied only above a per-dimension free-bits floor. It provides capacity control while preserving variation among ordinary transitions. The zero-transition term $\mathcal{L}_{\mathrm{zero}}$ acts on the relative norm $\rho_t = \|z_t^0\|_2 / (\mathrm{sg}(s_\Delta) + \epsilon)$ of the zero-transition latent $z_t^0 = \mu_\phi(o_t, o_t)$ (Section 3.2), where s_Δ is the running RMS norm of ordinary transition latents. It is nonzero only when ρ_t exceeds a small margin m_{zero}, so duplicated-frame inputs are encouraged toward the designated zero-transition reference. Here $\mathrm{sg}(\cdot)$ is stop-gradient, $[x]_+ = \max(x, 0)$, and $\beta_{\mathrm{kl}}(k)$, λ_{zero} are term weights (Table B.2).

4.3 Multi-stage Training

CD-LAM is used in a three-stage training pipeline (Figure 3):

• **Stage 1 (LAM Debiased Fine-tuning).** We first fine-tune the LAM on videos without executable action labels using the CD-LAM objective of Equation (7) with only coarse task annotation.

• **Stage 2 (ACWM Debiased Fine-tuning).** We then extract debiased latents z_t^{CD} from videos and continue ACWM training under the conditioning of Equation (4), with z_t^{CD} in place of z_t. This adapts the world model to the debiased latent action space before executable robot actions are introduced.

• **Stage 3 (Robot Action Adaptation).** Finally, for action labeled robot data (o_t, u_t, o_{t+1}), we align actions to the debiased latent action space. A lightweight MLP bridge g_η is trained to regress the gradient-stopped CD-LAM latent, $g_\eta(u_t) \approx \mathrm{sg}\big(\mu_\phi(o_t, o_{t+1})\big)$. An auxiliary action readout with a cycle-consistency objective encourages the mapped latent to retain information about u_t. During robot action adaptation, the ACWM receives $\hat{z}_t = g_\eta(u_t)$, so robot actions enter the same debiased latent action space used in Stage 2.

5 Experiments

5.1 Experimental Setup

Models and baseline. We apply CD-LAM to DreamDojo (Gao et al., 2026) at the 2B and 14B scales, which share a single latent action model, while keeping the downstream ACWM architecture, latent dimensionality, and conditioning interface unchanged. DreamDojo serves as the matched baseline to isolate the effect of latent-action debiasing within the same LAM–ACWM pipeline. For Stage 2, we continue the original DreamDojo training on the same video subset with matched update counts. For Stage 3, we compare against its released robot-action-adapted checkpoints under the same evaluation protocol. Recent latent-action methods span different levels of the learning stack, from alternative latent formulations and joint latent/world-model learning to policy/VLA pretraining (Wang et al., 2026; Ayalew et al., 2026; Zhang et al., 2026), so their reported scores also reflect differences

in downstream architecture, action grounding, embodiment, and evaluation. We therefore use DreamDojo for the debiasing study, while LTX-2.3 and X-VLA serve as complementary transfer studies evaluating whether the action-conditioning interface and the learned latent representation remain useful beyond DreamDojo pipeline (HaCohen et al., 2026; Zheng et al., 2026).

Training and data. The main study runs on 128 H100 GPUs. Our three-stage fine-tuning starts from released DreamDojo checkpoints. By default, LAM debiasing uses 100h of DreamDojo pretraining human video, and robot-action adaptation uses approximately 300h of paired AgiBot video and actions (Bu et al., 2025). We evaluate latent-action rollouts on EgoDex (Hoque et al., 2026) and robot-action rollouts on held-out AgiBot clips. Training and evaluation details are provided in Appendix B, with separate LTX and X-VLA protocols in Appendices F and G.

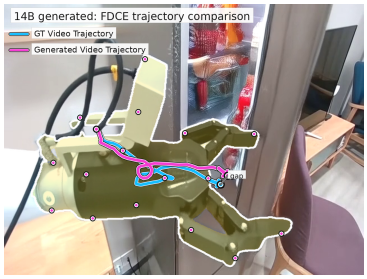


Figure 4: **FDCE computation:** symmetric Chamfer distance of foreground displacement tracks.

Metrics. We evaluate visual fidelity with PSNR, SSIM (Wang et al., 2004), and LPIPS (Zhang et al., 2018), and action following with Foreground Displacement Chamfer Error (FDCE). Pixel metrics alone do not certify action following: as Figure C.1 shows, high PSNR can coexist with incorrect embodiment motion. Following trajectory-based motion evaluation for controllable video generation (Wang et al., 2024; Zhang et al., 2025b), FDCE measures the symmetric Chamfer distance between generated and reference foreground displacement tracks in pixels; Figure 4 illustrates the computation and Appendix C provides the full definition. As a tracking-based proxy for action following, we interpret FDCE jointly with zero-action and target-action interventions, visual-fidelity metrics, and qualitative rollouts. For VLA policies, we report success rate.

5.2 Latent Action Diagnostics (Stage 1)

We first audit the latent action before any world model rollout. This directly probes the upstream source identified in Section 3: whether the encoder responds to action-irrelevant confounders as if they were embodied action. Table 1 shows that CD-LAM strongly reduces responses to action-irrelevant confounders: the median zero-transition response drops from 0.527 to 0.043, synthetic camera shifts induce 3.6–8.3$\times$ smaller responses. Shortcut leakage is computed as the difference between the mean cosine similarity of same-episode pairs with different primitives and that of different-episode pairs with the same primitive. It drops from 0.151 to 0.014, indicating substantially less context-driven structure in the latent space.

Table 2: **Latent action conditioned rollouts after ACWM debiased fine-tuning.** Left: conditioned on z_t from the video transition. Right: target-action transfer $\mathrm{do}(z_t = z_t^{\mathrm{tar}})$.

		latent action z_t rollout				$\mathrm{do}(z_t = z_t^{\mathrm{tar}})$			
Backbone	Model	FDCE↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	FDCE↓
2B	DreamDojo	34.00	20.88	0.780	0.413	13.02	**0.598**	0.643	42.74
	CD-LAM	**19.63**	**24.29**	**0.827**	**0.308**	**13.15**	0.588	**0.600**	**33.81**
14B	DreamDojo	40.29	21.04	0.792	0.398	**13.11**	0.593	0.631	50.27
	CD-LAM	**29.87**	**23.18**	**0.814**	**0.342**	13.03	**0.597**	**0.617**	**33.22**

5.3 ACWM Debiased Fine-tuning Rollouts (Stage 2)

We then evaluate ACWM rollouts conditioned on debiased LAM latents before introducing the robot-action bridge. Table 2 shows gains at two scales. Compared with the matched continued-training baseline, CD-LAM reduces FDCE by 42% at 2B (34.00 to 19.63) and by 26% at 14B (40.29 to 29.87) while also improving PSNR, SSIM, and LPIPS. Under target-action transfer $\mathrm{do}(z_t = z_t^{\mathrm{tar}})$, FDCE drops by 21% and 34%, respectively, while pixel metrics show smaller, mixed changes. These results show that CD-LAM improves action following before robot-action alignment. We also found that scaling the baseline from 2B to 14B does not improve action following, whereas CD-LAM reduces FDCE at both scales.

Table 3: **Robot action rollouts and interventions on AgiBot (Stage 3).** Zero-action FDCE is residual motion w.r.t. the static initial frame. DreamDojo: released 50k-update checkpoints.

		robot action u_t rollout					$\mathrm{do}(u_t = 0)$	$\mathrm{do}(u_t = u_t^{\mathrm{tar}})$
Backbone	Model	$\mathrm{FDCE}_{\mathrm{mean}}\downarrow$	$\mathrm{FDCE}_{\mathrm{med}}\downarrow$	PSNR ↑	SSIM ↑	LPIPS ↓	FDCE ↓	FDCE ↓
2B	DreamDojo	12.63	8.15	19.85	0.798	0.271	10.71	24.36
	CD-LAM	**8.24**	**6.75**	**20.60**	**0.806**	**0.269**	**5.03**	**22.55**
14B	DreamDojo	11.11	8.98	20.01	0.808	0.263	9.36	24.82
	CD-LAM	**7.73**	**5.99**	**21.01**	**0.818**	**0.247**	**2.18**	**21.11**

Table 4: **Debiasing-data scaling at 2B.** All tiers use 1k Stage-1 updates and differ only in video hours. DreamDojo is the 50k-update robot adaptation reference.

Model	PSNR ↑	$\mathrm{FDCE}_{\mathrm{mean}}\downarrow$	$\mathrm{FDCE}_{\mathrm{med}}\downarrow$
DreamDojo	19.85	12.63	8.15
CD-LAM-1h	20.54	8.91	6.88
CD-LAM-10h	20.61	8.87	6.23
CD-LAM-100h	20.60	8.24	6.75
CD-LAM-1000h	**20.64**	**7.97**	**6.12**

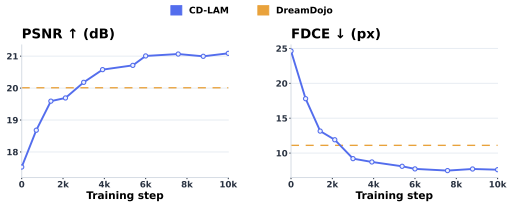


Figure 5: **Robot adaptation efficiency.** 14B PSNR and FDCE over training steps. Dashed lines mark DreamDojo at 50k updates.

Table 5: **CD-LAM objective ablation.** Each row omits the indicated LAM loss term. FG denotes FG-PSNR and $Cam.x/y$ are camera-shift responses.

Model	PSNR ↑	FG ↑	$Cam.x\downarrow$	$Cam.y\downarrow$	FDCE ↓
Full	**29.64**	**26.57**	**.133**	.101	17.23
w/o $\mathcal{L}_{\mathrm{emb}}$	29.53	26.26	.134	**.100**	18.40
w/o $\mathcal{L}_{\mathrm{ctr}}$	29.64	26.57	.134	.101	19.07
w/o $\mathcal{L}_{\mathrm{cal}}$	29.31	26.48	.637	.637	**16.10**

Table 6: **X-VLA policy success (%).** CD-LAM improves over DreamDojo LAM on LIBERO, LIBERO-Plus, and RoboTwin C2R.

Training	LIBERO	LIBERO-Plus	RoboTwin C2R Clean	RoboTwin C2R Random
X-VLA	98.05	69.74	85.67	25.00
DreamDojo LAM	95.72	75.69	84.00	28.83
CD-LAM	**96.30**	**78.25**	**86.67**	**33.83**
Δ (pp)	+0.58	+2.56	+2.67	+5.00

5.4 Robot Action Adaptation Rollouts (Stage 3)

Table 3 shows that CD-LAM also improves action following after robot-action adaptation. Under robot actions, CD-LAM reduces mean FDCE by 35% at 2B (12.63 to 8.24) and 30% at 14B (11.11 to 7.73), while improving median FDCE and all three visual-fidelity metrics. At 14B, CD-LAM achieves the best result in every column of the table, whereas baseline scaling is mixed: $\mathrm{FDCE}_{\mathrm{med}}$ *worsens* from 8.15 to 8.98 even as PSNR improves. Figure E.1 further shows that the gains span action categories and shift the rollout distribution toward lower FDCE and higher PSNR. Figure E.2 provides the full per-action breakdown and an additional target-action transfer example, while Figure E.3 shows representative rollouts at both scales. Under zero-input conditioning $\mathrm{do}(u_t = 0)$, residual FDCE decreases from 10.71 to 5.03 at 2B and from 9.36 to 2.18 at 14B. Zero FDCE corresponds to the static reference used in this probe. Under target-action transfer $\mathrm{do}(u_t = u_t^{\mathrm{tar}})$, FDCE improves from 24.36 to 22.55 and from 24.82 to 21.11. Together, these interventions show that CD-LAM improves action following under both recorded and transferred actions while suppressing residual motion under zero-action conditioning.

5.5 Robot Action Adaptation Efficiency and Debiasing Data Scaling

Figure 5 shows that at 14B, CD-LAM reaches the DreamDojo reference within roughly 3k updates on FDCE and 4k updates on PSNR (more than 12× fewer than the reference's 50k updates), and clearly surpasses it on both metrics by the 6k checkpoint in Table 3. These results suggest that correcting the latent action representation before robot-action adaptation reduces the number of adaptation updates needed to reach the reference performance. Table 4 evaluates the data side of the same effect. All tiers share the same 1k-step Stage-1 budget, so the comparison isolates data quantity at fixed compute. Even the 1h CD-LAM tier reduces $\mathrm{FDCE}_{\mathrm{mean}}$ from 12.63 to 8.91, and the 1000h tier further lowers

it to 7.97. The 1h tier captures about 80% of the 1000h tier's FDCE improvement: limited debiasing data yields most of the gain, while more data refines action consistency.

5.6 Objective Ablation

Table 5 reports a separate ablation setup: PSNR and FG-PSNR evaluate Stage-2 rollouts, camera-shift norms probe the LAM, and FDCE evaluates Stage-3 rollouts. The L_{emb} ablation lowers FG-PSNR by 0.31 dB and increases FDCE by 1.17 px. Removing L_{ctr} leaves the reconstruction and camera-shift scores nearly unchanged but increases FDCE by 1.84 px. Removing the combined latent-space calibration term L_{cal} increases horizontal/vertical shifted-pair norms from 0.133/0.101 to 0.637/0.637 and lowers PSNR by 0.33 dB, while improving FDCE from 17.23 to 16.10. This reveals a trade-off: L_{cal} improves latent-space calibration and visual fidelity but does not uniformly improve downstream motion metrics, while the full model still achieves lower FDCE than the L_{emb} and L_{ctr} ablations.

5.7 Efficient Robot-Action Adaptation of a Pretrained Video Backbone

Table 7: **Efficient adaptation on LTX-2.3.** Training exposure and robot-action rollout performance relative to DreamDojo-14B.

	Latent-action pretraining			Robot-action adaptation			Robot-action rollout		
Model	GPUs	Updates	Exposure Samples	GPUs	Updates	Exposure Samples	PSNR ↑	$\mathrm{FDCE}_{\mathrm{mean}}$ ↓	$\mathrm{FDCE}_{\mathrm{med}}$ ↓
DreamDojo 14B	256 H100	140k	143.360M	128 H100	50k	25.600M	20.01	11.11	8.98
CD-LAM + LTX-2.3 22B	8 H200	60k	3.840M	7 H200	28k	1.568M	**20.38**	11.75	**7.82**

We further test whether CD-LAM can efficiently adapt a generic pretrained video backbone into an embodied action-conditioned world model. Starting from LTX-2.3 22B, we perform only 60k latent-action pretraining updates, followed by 28k robot-action adaptation updates. This corresponds to 37.3 $\times$ fewer latent-action training sample exposures and 16.3 $\times$ fewer robot-action adaptation exposures than DreamDojo-14B. Despite this substantially reduced training exposure, the adapted LTX-2.3 achieves comparable robot-action rollout quality, with higher PSNR (20.38 vs. 20.01) and lower median FDCE (7.82 vs. 8.98). These results show that CD-LAM supports efficient adaptation of a pretrained video backbone to robot-action conditioning, achieving comparable rollout quality with substantially fewer training cost than DreamDojo-14B.

5.8 Debiased Latent Actions as a Policy Pretraining Signal

Finally, we compare X-VLA policies trained with latent actions from either DreamDojo LAM or CD-LAM, using matched recipes within each benchmark (Table 6). Across three paired post-training seeds, CD-LAM improves LIBERO success from 95.72% to 96.30% and LIBERO-Plus overall success from 75.69% to 78.25%, with positive mean gains across all seven perturbation categories. On RoboTwin 2.0, we evaluate C2R robustness on a predefined 12-task subset comprising ten eligible tasks with the largest public X-VLA clean-to-random gaps and two anchor tasks to assess policy robustness after latent-action pretraining. With CD-LAM, randomized-scene success increases from 28.8% to 33.8%, with comparable clean performance (84.0% vs. 86.7%). These results show that CD-LAM also improves downstream policy success. Full protocols, task-selection criteria, and task-level results are provided in Appendix G.

6 Conclusion

We presented CD-LAM, a causally motivated framework that improves world model controllability by reducing visual bias in latent action representations. Across 2B and 14B backbones, CD-LAM improves action following, visual fidelity, and adaptation efficiency, with benefits persisting from latent action conditioning to executable robot controls. Experiments with LTX-2.3 and X-VLA further demonstrate its applicability to video model adaptation and policy learning. Together, these findings identify upstream representation bias as a practical obstacle to downstream control and show that correcting it yields benefits across models and tasks. By correcting bias in latent actions, CD-LAM offers a practical path toward more controllable world models and more robust embodied policies.

AI USE STATEMENT

In this work, we used generative AI tools as coding assistants during implementation (code completion, refactoring, and debugging of the training and evaluation scripts) and for editing the manuscript (grammar, phrasing, and LaTeX formatting), as well as for assistance with plotting and summarizing existing experimental results. We have not used generative AI tools for generating research ideas, designing the method or the experiments, or producing experimental results. All AI-assisted code was tested by the authors, and all AI-assisted text was reviewed and revised by the authors; we take full responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies the controllability of action-conditioned world models and trains and evaluates on publicly released egocentric human-video and robot-manipulation datasets (Section 5.1), used under their respective licenses and terms of use; no new data involving human subjects were collected. Debiased latent action models and world models are intended for robot planning, policy evaluation, and data augmentation; we do not foresee direct misuse beyond the general dual-use considerations of video generation models.

REPRODUCIBILITY STATEMENT

The three CD-LAM objectives are fully specified in Section 4 (Equations (7), (8), (9), and (10)), and the three-stage training pipeline in Section 4.3. Backbones, the baseline, debiasing data tiers, optimizer step budgets, batch sizes, and evaluation splits are listed in Section 5.1 and Appendix B; the intervention protocol is defined in Section 3.2 and Appendix B. Complete definitions of FDCE, the pixel metrics, and the three LAM diagnostics are given in Appendix C, and the construction of the coarse action-primitive labels used by the contrastive term in Appendix D. The training recipe and evaluation protocol of the VLA policy experiment, including three-seed results and per-category breakdowns are given in Appendix G. Loss weights and schedules of the debiasing stage are listed in Table B.2. Training and evaluation code, configuration files, and the debiased LAM and ACWM checkpoints are provided in an anonymized repository at `https://anonymous.4open.science/r/CD-LAM-anonymous-373C/`.

REFERENCES

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Tewodros Ayalew, Matthew Jeung, Samuel Wheeler, Xiao Zhang, Andre de la Cruz Arce, Kaylene Stocking, Michael Maire, and Matthew R Walter. Claw: Learning continuous latent action world models via adversarial latent regularization. *arXiv preprint arXiv:2606.04130*, 2026.

Bowen Baker, Ilge Akkaya, Peter Zhokhov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampedro, and Jeff Clune. Video PreTraining (VPT): Learning to Act by Watching Unlabeled Online Videos. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. arXiv:2206.11795.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first international conference on machine learning*, 2024.

Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xuan Hu, Xu Huang, et al. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. SAM 3: Segment anything with concepts. In *International conference on learning representations*, volume 2026, pp. 138846–138923, 2026.

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

Xiaoyu Chen, Junliang Guo, Tianyu He, Chuheng Zhang, Pushi Zhang, Derek Cathera Yang, Li Zhao, and Jiang Bian. IGOR: Image-GOal Representations are the Atomic Control Units for Foundation Models in Embodied AI. *arXiv preprint arXiv:2411.00785*, 2024.

Yi Chen, Yuying Ge, Weiliang Tang, Yizhuo Li, Yixiao Ge, Mingyu Ding, Ying Shan, and Xihui Liu. Moto: Latent Motion Token as the Bridging Language for Learning Robot Manipulation from Videos. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025b.

Weisheng Dai, Kai Lan, Jianyi Zhou, Bo Zhao, Xiu Su, Junwen Tong, Weili Guan, and Shuo Yang. Conla: Contrastive latent action learning from human videos for robotic manipulation. *arXiv preprint arXiv:2602.00557*, 2026.

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. The EPIC-KITCHENS Dataset: Collection, Challenges and Baselines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):4125–4141, 2021. doi: 10.1109/TPAMI.2020.2991965.

Pim De Haan, Dinesh Jayaraman, and Sergey Levine. Causal confusion in imitation learning. *Advances in neural information processing systems*, 32, 2019.

Ashley D. Edwards, Himanshu Sahni, Yannick Schroecker, and Charles L. Isbell. Imitating Latent Policies from Observation. In *International Conference on Machine Learning (ICML)*, 2019. ILPO; arXiv:1805.07914.

Marcus Fechner, Hamza Adnan, Constantin C Lüth, Matthew T Jackson, Alexey Zakharov, and J Marius Zöllner. Segment to focus: Guiding latent action models in the presence of distractors. *arXiv preprint arXiv:2602.02259*, 2026.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, et al. Libero-plus: A progressive robustness benchmark for visual-language-action models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 38574–38583, 2026.

Shenyuan Gao, Siyuan Zhou, Yilun Du, Jun Zhang, and Chuang Gan. Adaworld: Learning adaptable world models with latent actions. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=QQegZj99sk.

Shenyuan Gao, William Liang, Kaiyuan Zheng, Ayaan Naveed Malik, Seonghyeon Ye, Sihyun Yu, Wei-Cheng Tseng, Yuzhu Dong, Kaichun Mo, Chen-Hsuan Lin, Jiannan Xiang, Yuqi Xie, Ruijie Zheng, Dantong Niu, Pooya Jannaty, Jinwei Gu, Jun Zhang, Jitendra Malik, Pieter Abbeel, Ming-Yu Liu, Yuke Zhu, Joel Jang, and Linxi Fan. Dreamdojo: A generalist robot world model from large-scale human videos. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=FuvU7PTyED.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Abrham Gebreselasie, Cristina González, James Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jáchym Kolář, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz, Merey Ramazanova, Leda

Sari, Kiran Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Ziwei Zhao, Yunyi Zhu, Pablo Arbeláez, David Crandall, Dima Damen, Giovanni Maria Farinella, Christian Fuegen, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18995–19012, June 2022.

Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, Eugene Byrne, Zach Chavis, Joya Chen, Feng Cheng, Fu-Jen Chu, Sean Crane, Avijit Dasgupta, Jing Dong, Maria Escobar, Cristhian Forigua, Abrham Gebreselasie, Sanjay Haresh, Jing Huang, Md Mohaiminul Islam, Suyog Jain, Rawal Khirodkar, Devansh Kukreja, Kevin J. Liang, Jia-Wei Liu, Sagnik Majumder, Yongsen Mao, Miguel Martin, Effrosyni Mavroudi, Tushar Nagarajan, Francesco Ragusa, Santhosh Kumar Ramakrishnan, Luigi Seminara, Arjun Somayazulu, Yale Song, Shan Su, Zihui Xue, Edward Zhang, Jinxu Zhang, Angela Castillo, Changan Chen, Xinzhu Fu, Ryosuke Furuta, Cristina González, Prince Gupta, Jiabo Hu, Yifei Huang, Yiming Huang, Weslie Khoo, Anush Kumar, Robert Kuo, Sach Lakhavani, Miao Liu, Mi Luo, Zhengyi Luo, Brighid Meredith, Austin Miller, Oluwatumininu Oguntola, Xiaqing Pan, Penny Peng, Shraman Pramanick, Merey Ramazanova, Fiona Ryan, Wei Shan, Kiran Somasundaram, Chenan Song, Audrey Southerland, Masatoshi Tateno, Huiyu Wang, Yuchen Wang, Takuma Yagi, Mingfei Yan, Xitong Yang, Zecheng Yu, Shengxin Cindy Zha, Chen Zhao, Ziwei Zhao, Zhifan Zhu, Jeff Zhuo, Pablo Arbeláez, Gedas Bertasius, David Crandall, Dima Damen, Jakob Engel, Giovanni Maria Farinella, Antonino Furnari, Bernard Ghanem, Judy Hoffman, C. V. Jawahar, Richard Newcombe, Hyun Soo Park, James M. Rehg, Yoichi Sato, Manolis Savva, Jianbo Shi, Mike Zheng Shou, and Michael Wray. Ego-Exo4D: Understanding Skilled Human Activity from First- and Third-Person Perspectives. *International Journal of Computer Vision*, 133:8356–8435, 2025. doi: 10.1007/s11263-025-02557-6.

David Ha and Jürgen Schmidhuber. World Models. *arXiv preprint arXiv:1803.10122*, 2018.

Yoav HaCohen, Benny Brazowski, Nisan Chiprut, Yaki Bitterman, Andrew Kvochko, Avishai Berkowitz, Daniel Shalem, Daphna Lifschitz, Dudu Moshe, Eitan Porat, et al. Ltx-2: Efficient joint audio-visual foundation model. *arXiv preprint arXiv:2601.03233*, 2026.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models, 2024. URL https://arxiv.org/abs/2301.04104.

Ryan Hoque, Peide Huang, David J. Yoon, Mouli Sivapurapu, and Jian Zhang. Egodex: Learning dexterous manipulation from large-scale egocentric video. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=FFxkFMU89E.

Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. CoTracker: It is Better to Track Together. In *European Conference on Computer Vision (ECCV)*, 2024.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, Peter David Fagan, Joey Hejna, Masha Itkina, Marion Lepert, Yecheng Jason Ma, Patrick Tree Miller, Jimmy Wu, Suneel Belkhale, Shivin Dass, Huy Ha, Arhan Jain, Abraham Lee, Youngwoon Lee, Marius Memmel, Sungjae Park, Ilija Radosavovic, Kaiyuan Wang, Albert Zhan, Kevin Black, Cheng Chi, Kyle Beltran Hatch, Shan Lin, Jingpei Lu, Jean Mercat, Abdul Rehman, Pannag R Sanketi, Archit Sharma, Cody Simpson, Quan Vuong, Homer Rich Walke, Blake Wulfe, Ted Xiao, Jonathan Heewon Yang, Arefeh Yavary, Tony Z. Zhao, Christopher Agia, Rohan Baijal, Mateo Guaman Castro, Daphne Chen, Qiuyu Chen, Trinity Chung, Jaimyn Drake, Ethan Paul Foster, Jensen Gao, David Antonio Herrera, Minho Heo, Kyle Hsu, Jiaheng Hu, Donovon Jackson, Charlotte Le, Yunshuang Li, Roy Lin, Zehan Ma, Abhiram Maddukuri, Suvir Mirchandani, Daniel Morton, Tony Nguyen, Abigail O'Neill, Rosario Scalise, Derick Seale, Victor Son, Stephen Tian, Emi Tran, Andrew E. Wang, Yilin Wu, Annie Xie, Jingyun Yang, Patrick Yin, Yunchu Zhang, Osbert Bastani, Glen Berseth, Jeannette Bohg, Ken Goldberg, Abhinav Gupta, Abhishek

Gupta, Dinesh Jayaraman, Joseph J Lim, Jitendra Malik, Roberto Martín-Martín, Subramanian Ramamoorthy, Dorsa Sadigh, Shuran Song, Jiajun Wu, Michael C. Yip, Yuke Zhu, Thomas Kollar, Sergey Levine, and Chelsea Finn. DROID: A Large-Scale In-The-Wild Robot Manipulation Dataset. In *Proceedings of Robotics: Science and Systems*, 2024. doi: 10.15607/RSS.2024.XX.120.

Zihang Lai, Eldar Insafutdinov, Edgar Sucar, and Andrea Vedaldi. Cowtracker: Tracking by warping instead of correlation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 42571–42580, June 2026.

Jung Min Lee, Dohyeok Lee, Seokhun Ju, Taehyun Cho, Jin Woo Koo, Li Zhao, Sangwoo Hong, and Jungwoo Lee. MVP-LAM: Learning action-centric latent action via cross-viewpoint reconstruction. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=r0SYyjDhxY.

Lightricks. LTX-2.3, 2026. URL https://huggingface.co/Lightricks/LTX-2.3.

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

Alexander Nikulin, Ilya Zisman, Denis Tarasov, Nikita Lyubaykin, Andrei Polubarov, Igor Kiselev, and Vladislav Kurenkov. Latent action learning requires supervision in the presence of distractors. *arXiv preprint arXiv:2502.00379*, 2025.

NVIDIA, Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, Daniel Dworakowski, Jiaojiao Fan, Michele Fenzi, Francesco Ferroni, Sanja Fidler, Dieter Fox, Songwei Ge, Yunhao Ge, Jinwei Gu, Siddharth Gururani, Ethan He, Jiahui Huang, Jacob Huffman, Pooya Jannaty, Jingyi Jin, Seung Wook Kim, Gergely Klár, Grace Lam, Shiyi Lan, Laura Leal-Taixe, Anqi Li, Zhaoshuo Li, Chen-Hsuan Lin, Tsung-Yi Lin, Huan Ling, Ming-Yu Liu, Xian Liu, Alice Luo, Qianli Ma, Hanzi Mao, Kaichun Mo, Arsalan Mousavian, Seungjun Nah, Sriharsha Niverty, David Page, Despoina Paschalidou, Zeeshan Patel, Lindsey Pavao, Morteza Ramezanali, Fitsum Reda, Xiaowei Ren, Vasanth Rao Naik Sabavat, Ed Schmerling, Stella Shi, Bartosz Stefaniak, Shitao Tang, Lyne Tchapmi, Przemek Tredak, Wei-Cheng Tseng, Jibin Varghese, Hao Wang, Haoxiang Wang, Heng Wang, Ting-Chun Wang, Fangyin Wei, Xinyue Wei, Jay Zhangjie Wu, Jiashu Xu, Wei Yang, Lin Yen-Chen, Xiaohui Zeng, Yu Zeng, Jing Zhang, Qinsheng Zhang, Yuxuan Zhang, Qingqing Zhao, and Artur Zolkowski. Cosmos world foundation model platform for physical ai, 2025. URL https://arxiv.org/abs/2501.03575.

Open X-Embodiment Collaboration. Open X-Embodiment: Robotic Learning Datasets and RT-X Models. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6892–6903, 2024. doi: 10.1109/ICRA57147.2024.10611477.

Ryan Punamiya, Simar Kareer, Zeyi Liu, Josh Citron, Ri-Zhao Qiu, Xiongyi Cai, Alexey Gavryushin, Jiaqi Chen, Davide Liconti, Lawrence Y Zhu, et al. Egoverse: An egocentric human dataset for robot learning from around the world. *arXiv preprint arXiv:2604.07607*, 2026.

Dominik Schmidt and Minqi Jiang. Learning to act without actions. In *International Conference on Learning Representations*, volume 2024, pp. 9379–9395, 2024.

Yuhong Shi, Zhenhao Chu, Jie Wei, Jun Hao, Jianyi Liu, and Jingwen Fu. Overcoming statistical bias in action-controllable world models. *arXiv preprint arXiv:2608.04653*, 2026.

Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges, 2019. URL https://arxiv.org/abs/1812.01717.

Yucen Wang, Fengming Zhang, De-Chuan Zhan, Li Zhao, Kaixin Wang, and Jiang Bian. Co-evolving latent action world models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=iec4FIZyrx.

Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861.

Zhouxia Wang, Ziyang Yuan, Xintao Wang, Yaowei Li, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. Motionctrl: A unified and flexible motion controller for video generation. In *ACM SIGGRAPH 2024 Conference Papers*, pp. 1–11, 2024.

Hangxing Wei, Xiaoyu Chen, Chuheng Zhang, Tim Pearce, Jianyu Chen, Alex Lamb, Li Zhao, and Jiang Bian. Learning additively compositional latent actions for embodied ai. *arXiv preprint arXiv:2604.03340*, 2026.

Jialong Wu, Shaofeng Yin, Ningya Feng, Xu He, Dong Li, Jianye Hao, and Mingsheng Long. iVideoGPT: Interactive VideoGPTs are Scalable World Models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. arXiv:2405.15223.

Haotian Xue, Yipu Chen, Liqian Ma, Zelin Zhao, Lama Moukheiber, Yuchen Zhu, and Yongxin Chen. Acwm-phys: Investigating generalized physical interaction in action-conditioned video world models. *CoRR*, abs/2605.08567, May 2026. URL https://doi.org/10.48550/arXiv.2605.08567.

Sherry Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Leslie Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning Interactive Real-World Simulators. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/c4d66eae503694424123b93ac0fbaf17-Abstract-Conference.html.

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Se June Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, et al. Latent action pretraining from videos. In *International Conference on Learning Representations*, volume 2025, pp. 28213–28239, 2025.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid Loss for Language Image Pre-Training. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Chuheng Zhang, Tim Pearce, Pushi Zhang, Kaixin Wang, Xiaoyu Chen, Wei Shen, Li Zhao, and Jiang Bian. What do latent action models actually learn? *Advances in Neural Information Processing Systems*, 38:146676–146697, 2025a.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Tianqiu Zhang, Muyang Lyu, Yufan Zhang, Fang Fang, and Si Wu. DiLA: Disentangled latent action world models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=BRBHruBDkb.

Zhongwei Zhang, Fuchen Long, Zhaofan Qiu, Yingwei Pan, Wu Liu, Ting Yao, and Tao Mei. MotionPro: A Precise Motion Controller for Image-to-Video Generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025b.

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

A Related Work

Latent Action Learning from Video. Latent action learning builds on observation-only imitation (Edwards et al., 2019) and video-based pretraining with inferred action labels (Baker et al., 2022). Genie (Bruce et al., 2024) learns discrete latent controls for interactive generation, while LAPO (Schmidt & Jiang, 2024) uses inferred actions to bootstrap policies and shows that improved reconstruction need not yield better policy learning. LAPA (Ye et al., 2025) and Moto (Chen et al., 2025b) use quantized transition representations for policy pretraining: LAPA replaces its latent prediction head during robot-action fine-tuning, whereas Moto retains motion-token supervision. These works establish the value of video-derived action targets, while leaving different requirements on the representations ultimately used for control.

World Models and Latent-Action Conditioning. World models predict environment dynamics for planning and control (Ha & Schmidhuber, 2018; Hafner et al., 2024), with UniSim (Yang et al., 2024) and Cosmos (NVIDIA et al., 2025) scaling generative modeling to diverse video data. Action-conditioned architectures (Xue et al., 2026; Wu et al., 2024) and robot datasets spanning diverse tasks and scenes (Open X-Embodiment Collaboration, 2024; Khazatsky et al., 2024) support learning from executable controls. To exploit action-unlabeled video, IGOR (Chen et al., 2024) uses image-goal representations for both policy learning and video generation, while AdaWorld (Gao et al., 2025) and DreamDojo (Gao et al., 2026) use continuous latent actions for world-model pretraining and subsequent adaptation. CoLA-World (Wang et al., 2026) couples quantized latent-action learning with a pretrained world model and analyzes distribution mismatch and codebook degeneration during real-action adaptation. CLAW (Ayalew et al., 2026) jointly learns continuous actions and a diffusion world model, adversarially suppressing future-observation leakage. These approaches improve latent–world-model learning, but suppressing future appearance does not by itself exclude exogenous motion, and codebook stability does not determine which physical factors the codes represent.

Action-Relevant and Structured Latent Representations. Zhang et al. (2025a) analyze how reconstruction-driven learning can encode exogenous variation and observation-dependent shortcuts. LAOM (Nikulin et al., 2025) studies distractors and shows that action decodability and usefulness as a policy target are distinct considerations. More recent methods impose explicit structure. ConLA (Dai et al., 2026) combines category-based contrastive learning and temporal-reversal constraints in a quantized RGB-reconstruction LAM, while MVP-LAM (Lee et al., 2026) uses cross-view reconstruction of frozen DINO features. Category agreement and cross-view consistency constrain semantic organization and viewpoint dependence, but do not by themselves separate all embodiment-controlled changes from other predictable variation. AC-LAM (Wei et al., 2026) imposes additive composition and evaluates identity, inverse consistency, and normalized zero-transition responses; these algebraic properties provide useful structure without uniquely determining the represented physical factors. MaskLAM (Fechner et al., 2026) restricts reconstruction to agent regions to reduce distractor encoding. DiLA (Zhang et al., 2026) learns continuous actions through content–structure disentanglement and dynamics prediction in DINO feature space, supporting RGB generation and visual planning. Its separation of appearance from structural change leaves open how reliably transferred motion is assigned to the intended entity. These distinctions motivate evaluating latent actions according to their downstream use as embodied control conditions.

Causal Perspectives on Bias Propagation. Invariant prediction (Arjovsky et al., 2019), shortcut learning (Geirhos et al., 2020), and causal confusion in imitation learning (De Haan et al., 2019) motivate examining which information a predictor uses and how that dependence affects its behavior. We investigate a specific mechanism in LAM-based ACWMs. CoCo (Shi et al., 2026) addresses statistical shortcuts through counterfactual consistency constraints on action-conditioned world models. Source-specific visual information can be useful through the observation pathway yet also enter the latent action. If the world model relies on this information through its action condition, alignment with robot commands must accommodate variation that those commands cannot generally determine. CD-LAM targets this bias at its source through embodiment-centric reconstruction, action-centric contrastive learning, and latent-space calibration. We connect encoder diagnostics with action interventions and conditioned rollouts before and after robot action adaptation, evaluating how upstream representation correction affects controllability and whether its benefits persist under executable robot controls. Our study focuses on correcting the upstream latent-action representation and evaluating its effects before and after robot-action adaptation.

Evaluating Controllability and Embodied Motion. Visual-fidelity metrics (e.g., PSNR and FVD (Unterthiner et al., 2019)) do not by themselves establish whether a rollout follows its supplied action condition. Genie measures action sensitivity using ΔPSNR (Bruce et al., 2024). MotionCtrl evaluates object-motion control with ObjMC (Wang et al., 2024), while MotionPro measures trajectory alignment using mean-distance metrics, including MD-Vid computed with CoTracker (Zhang et al., 2025b; Karaev et al., 2024). Our FDCE metric compares foreground displacements between generated and reference rollouts, using SAM3 masks (Carion et al., 2026) and tracks from CoWTracker (Lai et al., 2026). We interpret this motion measure jointly with visual fidelity, latent diagnostics, and zero-action and target-action interventions to assess the downstream effects of latent-action debiasing.

B Training and Evaluation Details

This appendix specifies the main DreamDojo-based experiments. The LTX and X-VLA protocols are given separately in Appendices F and G.

B.1 Training Data and Evaluation Sets

Training data. Unless otherwise stated, Stage 1 uses 100h of debiasing video drawn from DreamDojo's pretraining data. The data-scaling study varies this amount over 1h, 10h, 100h, and 1000h while keeping the Stage-1 budget at 1k updates. Stage 2 compares CD-LAM with continued DreamDojo training on the same video subset and with matched update counts. Stage 3 uses approximately 300h of paired AgiBot video and executable actions (Bu et al., 2025).

Evaluation sets. Stage-2 latent-action rollouts are evaluated on 300 EgoDex clips (Hoque et al., 2026), an egocentric human-manipulation corpus. Stage-3 robot-action rollouts and interventions are evaluated on 300 held-out AgiBot clips drawn from distinct episodes. Metric definitions and validity filtering are given in Appendix C.

B.2 Stage-specific Training Configuration

The three stages follow Section 4.3. All main-study runs use 128 H100 GPUs. Table B.1 summarizes the update budgets and per-GPU batch sizes. Stage-3 entries identify the checkpoints used for the main results.

Table B.1: **Main-study training budgets.** The same debiased LAM serves both ACWM scales. Stage-3 counts exclude the preceding Stage-1 and Stage-2 updates. The released DreamDojo Stage-3 reference uses 50k updates; Stage-3 batches match this reference.

Stage	Model / backbone	Optimizer updates	Batch / GPU
1: LAM debiasing	Shared LAM	1k	48
2: ACWM fine-tuning	2B	2k	12
2: ACWM fine-tuning	14B	2k	2
3: Robot-action adaptation	2B	3k checkpoint	Matched
3: Robot-action adaptation	14B	6k checkpoint	Matched

Objectives and optimization. Stage-3 batch sizes and optimizer settings match the DreamDojo reference (Section B.3). Table B.2 gives the Stage-1 loss weights, schedules, and remaining reported LAM settings. Bridge fitting has its own optimizer and batch size (Section B.4); these are distinct from ACWM training.

B.3 Baseline and Efficiency Comparison

For Stage 2, CD-LAM and the continued-training DreamDojo baseline use the same video subset and matched update counts. For Stage 3, the main results compare the adapted CD-LAM pipeline with DreamDojo's released robot-action-adapted checkpoints under the same evaluation protocol.

In the robot-action adaptation efficiency comparison, step counts denote optimizer updates under the aligned protocol: both models use the same adaptation data, batch size, and optimizer settings. The

Table B.2: **CD-LAM objective hyperparameters.** Loss weights and schedules behind Equations (7), (8), (9), and (10), in the notation of Section 4. Pair weighting for Equation (9) is specified in the released configuration. Stage-specific update budgets are summarized in Table B.1.

Term	Hyperparameter	Value
Embodiment-centric Reconstruction $\mathcal{L}_{\mathrm{emb}}$ (foreground / background)		
foreground recon weight	α_{fg}	1.0
background recon weight	α_{bg}	0.02
Action-centric Contrastive Learning $\mathcal{L}_{\mathrm{ctr}}$		
projection head	r_ω	$32 \to 128 \to 32$
log-scale / bias init	$\log \tau,\ b$	$2.30,\ -3.0$
coefficient schedule	$\lambda_{\mathrm{ctr}}(k)$	$0 \to 0.020$ (steps 150–400) $\to 0.005$ (cos, by 1000)
Latent Space Calibration $\mathcal{L}_{\mathrm{cal}}$ (free-bit KL + zero-transition calibration)		
KL free bits	κ	0.5
KL weight (ramped)	$\beta_{\mathrm{kl}}(k)$	$10^{-4} \to 10^{-3}$ by step 500
zero-transition weight	λ_{zero}	1.0
zero-transition margin	m_{zero}	0.05
zero-transition constant	ϵ	10^{-8}
running RMS s_Δ	—	EMA of squared norm, update weight 0.05
Latent / optimization		
latent-action dim	—	32
frame size / precision	—	240×320 / bf16
weight decay / grad clip	—	10^{-4} / 1.0
initialization	—	DreamDojo LAM checkpoint

50k-step reference in Figure 1(c) and the dashed lines in Figure 5 refer to the released DreamDojo AgiBot checkpoint, trained for 50k updates on 128 GPUs with the same per-GPU batch as our adaptation runs. The reported update-count reduction concerns Stage 3; it excludes the preceding 1k LAM debiasing and 2k ACWM fine-tuning updates and is not an end-to-end wall-clock speedup. The LTX comparison uses different backbones and hardware and is not compute-normalized (Appendix F).

B.4 Robot-action Bridge

The bridge g_η and the auxiliary readout h are three-layer MLPs (Linear–GELU–Linear–GELU–Linear, width 256). g_η maps the full 22-D action vector u^{in} to the 32-D latent, and h maps a latent back to u^{act}, the 15 action dimensions that vary in the training split; u and z are standardized per dimension with training-split statistics. They are fitted by minimizing $\lambda_{\mathrm{enc}}\|g_\eta(u^{\mathrm{in}}) - z\|_2^2 + \lambda_{\mathrm{cyc}}\|h(g_\eta(u^{\mathrm{in}})) - u^{\mathrm{act}}\|_2^2 + \lambda_{\mathrm{dec}}\|h(z) - u^{\mathrm{act}}\|_2^2$ with weights $1.0/3.0/0.3$, using Adam (learning rate 10^{-3}, weight decay 10^{-5}), batch 2,048 and 300 epochs on 10,560 training and 1,440 validation pairs; the LAM is frozen and its latents are pre-computed. During robot action adaptation the bridge is frozen as well, so the video loss does not back-propagate into g_η.

B.5 Intervention Protocol

Interventions follow Section 3.2: the observation history is held fixed and only the conditioning action is replaced. Under a *zero action*, $\mathrm{do}(u_t = 0)$, embodiment motion should be suppressed. Under *target-action transfer*, the conditioning action is replaced with one encoded from a different target video, and the rollout should reflect the direction and magnitude of the transferred target action under the fixed source context. For latent actions this means $\mathrm{do}(z_t = z_t^{\mathrm{tar}})$ with $z_t^{\mathrm{tar}} = \mu_\phi(o_t^{\mathrm{tar}}, o_{t+1}^{\mathrm{tar}})$; for robot actions, $\mathrm{do}(u_t = u_t^{\mathrm{tar}})$ with u_t^{tar} the target video's recorded action.

C METRIC DETAILS

PSNR Reporting. We report PSNR as the visual-fidelity metric, computed on full frames in dB. For images normalized to $[0, 1]$,

$$\mathrm{PSNR}(x, \hat{x}) = 10 \log_{10} \frac{1}{\mathrm{MSE}(x, \hat{x})}. \tag{C.1}$$

Figure 1(b) reports absolute PSNR for each setting, with CD-LAM gains annotated in dB. Because PSNR is logarithmic, a fixed dB gain corresponds to a multiplicative MSE reduction: the $+1.0$ dB gain at 14B after robot action adaptation is a 20.6% reduction in MSE ($10^{-1/10} \approx 0.794$).

FDCE Definition. FDCE measures foreground action following by comparing foreground displacement tracks rather than raw pixels. It draws on trajectory-based evaluation for controllable video generation (Wang et al., 2024; Zhang et al., 2025b), where generated trajectories are compared with prescribed motion controls. Our adaptation compares foreground displacement tracks extracted from generated and reference videos. Foreground masks select embodiment and interacted-object regions using SAM3 (Carion et al., 2026), and point tracks are computed only within valid foreground regions using CoWTracker (Lai et al., 2026).

For a reference foreground point p_j^s and a generated foreground point $\hat{p}_i^s$ at rollout step s, we define displacement vectors relative to the initial frame as

$$a_j^s = p_j^s - p_j^0, \qquad \hat{a}_i^s = \hat{p}_i^s - \hat{p}_i^0. \tag{C.2}$$

For rollout step s, let $\hat{\mathcal{A}}^s = \{\hat{a}_i^s\}_{i=1}^{N_g^s}$ and $\mathcal{A}^s = \{a_j^s\}_{j=1}^{N_r^s}$ be the displacements of the N_g^s generated and N_r^s reference foreground tracks that are valid at that step. The per-step symmetric Chamfer distance is

$$d^s = \frac{1}{2N_g^s} \sum_{i=1}^{N_g^s} \min_j \left\| \hat{a}_i^s - a_j^s \right\|_2 + \frac{1}{2N_r^s} \sum_{j=1}^{N_r^s} \min_i \left\| \hat{a}_i^s - a_j^s \right\|_2, \tag{C.3}$$

and FDCE averages it over the retained rollout steps $\mathcal{S}_{\mathrm{valid}}$ (steps with fewer than four valid points on either side are dropped, see below):

$$\mathrm{FDCE}(\hat{o}, o) = \frac{1}{|\mathcal{S}_{\mathrm{valid}}|} \sum_{s \in \mathcal{S}_{\mathrm{valid}}} d^s. \tag{C.4}$$

In our evaluation, we sample up to 16 valid foreground anchors per rollout pair and average the bidirectional nearest-neighbor distances. Lower FDCE indicates that the generated rollout induces foreground displacement closer to the reference action. Figure C.1 contrasts FDCE with PSNR on the same rollouts.

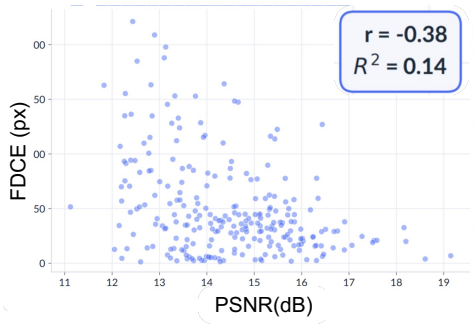


Figure C.1: **Pixel-level fidelity does not certify action following.** PSNR explains only limited variance in FDCE ($r = -0.38$, $R^2 = 0.14$). Each point is one rollout.

Robustness Properties. Anchors are seeded inside an eroded foreground mask. Tracks are linearly interpolated over short invisible spans and Savitzky–Golay smoothed before scoring; at step s only points visible at both step 0 and step s enter $\hat{\mathcal{A}}^s$ and $\mathcal{A}^s$, steps with fewer than four valid points

on either side are skipped, and a rollout pair with fewer than two valid steps is marked invalid and excluded from all means (596 of the 600 arm–pairs of the 300-clip evaluation are valid). The symmetric Chamfer form in Equation (C.3) scores displacement geometry rather than matched point counts, so the metric is robust to differing numbers of valid tracks. Tracking can be unreliable under heavy occlusion or fast motion, which can affect FDCE estimates.

Reporting Conventions. FDCE is reported in pixels at the evaluation resolution. The mean is sensitive to large outliers while the median reflects typical behavior; we report both. Pixel metrics (PSNR, SSIM, LPIPS) are computed on full frames against the reference rollout, and FG-PSNR restricts the same computation to the foreground mask.

LAM Diagnostic Definitions. The three diagnostics of Table 1 are computed from the posterior mean μ_ϕ on the audit split and reported as medians (camera shifts also as means; shortcut leakage is a difference of means) over evaluation pairs. The *zero-transition* and *camera-shift* responses are computed from a duplicated pair and a synthetically shifted pair, respectively, normalized by the typical latent norm of ordinary transitions:

$$R_{\text{zero}} = \frac{\|\mu_\phi(o_t, o_t)\|_2}{D}, \qquad R_{\text{shift}} = \frac{\|\mu_\phi(o_t, T_3(o_t))\|_2}{D}, \tag{C.5}$$

where $D = \text{RMS}\left(\|\mu_\phi(o_t, o_{t+1})\|_2\right) + \epsilon$ over the audit split and T_3 translates the frame horizontally or vertically by 3 pixels. The *shortcut leakage* is the cosine gap

$$\begin{aligned} L_{\text{shortcut}} = \mathbb{E}\big[\cos(z_i, z_j) \,\big|\, \text{same episode, diff. primitive}\big] \\ - \mathbb{E}\big[\cos(z_i, z_j) \,\big|\, \text{diff. episode, same primitive}\big], \end{aligned} \tag{C.6}$$

where pairs are drawn from the audit split using the coarse primitive labels of Appendix D; the second term alone is the same-primitive cosine similarity reported in the main text (0.132 vs. 0.131).

Reading Table 1. All three diagnostics feed the encoder a change that is not an embodied action, so an encoder whose latent carries only embodiment dynamics should respond weakly to all of them. Table C.1 expands Table 1 with the input of each diagnostic and its ideal value.

- *Zero-transition.* A duplicated pair (o_t, o_t) contains no motion, so its latent should lie near zero. R_{zero} expresses its norm as a fraction of a typical transition latent. The DreamDojo LAM gives 0.527, i.e., a static input is encoded as roughly half of a typical action, whereas CD-LAM gives 0.043. The absolute norm $\|\mu_\phi(o_t, o_t)\|_2$ (3.119 vs. 0.226) is reported alongside to show that the relative drop does not come from a larger normalizer D.
- *Camera shift.* T_3 moves the whole frame by 3 pixels, which mimics a small camera motion while no hand or object moves. The DreamDojo LAM responds to such a shift with more than half the norm of a typical transition (0.53–0.56), and CD-LAM responds 3.6–8.3$\times$ more weakly.
- *Shortcut leakage.* This diagnostic asks what makes two latents similar. For the DreamDojo LAM, pairs from the same episode with different primitives have a mean cosine similarity of 0.283, higher than pairs with the same primitive from different episodes (0.132), so shared visual context outweighs shared action. CD-LAM reduces this gap from 0.151 to 0.014 while the same-primitive term stays essentially unchanged (0.131). The reduction therefore comes from removing context-driven similarity, not from shrinking all similarities. A value near zero means episode context no longer outweighs action in the latent neighborhood structure.

D Coarse Action-primitive Labels

The action-centric contrastive loss (Equation (9)) uses coarse action-primitive labels rather than executable robot actions. The label space is a 12-way canonical verb set: pick–place, insert–remove, stack–unstack, scoop–dump, open, close, turn on, turn off, wash–rinse, cut, stir, and pour. Labels come from the task-name and verb fields shipped with the source datasets. A hand-written rule table maps each verb phrase and its aliases (e.g., grab, grasp, take $\rightarrow$ pick–place) onto one of the 12 canonical primitives; clips whose annotation matches no rule remain unlabeled and join no contrastive

Table C.1: **Expanded LAM encoder audit** (Table 1 with inputs and ideal values). Norm responses are relative to D, the RMS latent norm of ordinary transitions, except the absolute norm. Camera-shift entries are mean/median. All other responses are medians, and cosine terms are means over pairs.

Diagnostic	Encoder input	Ideal	DreamDojo LAM	CD-LAM
Zero-transition, rel. norm R_{zero}	duplicated pair (o_t, o_t)	≈ 0	0.527	**0.043**
Zero-transition, abs. norm	duplicated pair (o_t, o_t)	≈ 0	3.119	**0.226**
Camera shift, horizontal R_{shift}	$(o_t, T_3(o_t))$, 3 px	≈ 0	0.555/0.536	**0.156/0.096**
Camera shift, vertical R_{shift}	$(o_t, T_3(o_t))$, 3 px	≈ 0	0.545/0.529	**0.110/0.064**
Shortcut leakage L_{shortcut}	latent pairs, Equation (C.6)	≤ 0	0.151	**0.014**
same-primitive, diff. episode cosine	latent pairs	kept	0.132	0.131

pair. The labels are verb-level only (no controller states or trajectories) and merely tell $\mathcal{L}_{\text{ctr}}$ which transitions plausibly share a primitive.

Of the 68,864 transition pairs in the clean-ego index, 25,192 (36.6%) carry a primitive label; unlabeled pairs contribute only to reconstruction and calibration. The LAM audit split covers all 12 primitives. The label distribution is long-tailed (pick–place dominates), so a typical training batch exposes the contrastive term to an effective 8–10 primitives (exponentiated label entropy). The alias table deliberately merges phrasing- and viewpoint-dependent verb variants, so $\mathcal{L}_{\text{ctr}}$ never relies on fine-grained verb semantics.

E Additional Results

Figure E.1 breaks down the robot action adaptation results of Section 5.4 by action category and rollout distribution, and Figure E.2 gives the full per-action breakdown together with an additional target-action transfer example. Figure E.3 shows representative rollouts after robot action adaptation at both scales. The baseline target-action transfer failure is shown together with the zero-action intervention in Figure 2. In Figure 2b, the first row shows the source context, and the second row shows the target video that provides the robot action sequence. The third row is the ACWM rollout conditioned on that target action under the fixed source context. The rollout does not reproduce the target embodiment dynamics, indicating target-action misalignment after robot action adaptation.

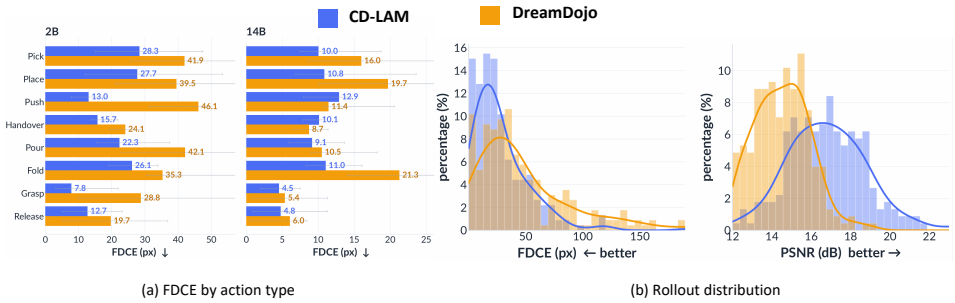


Figure E.1: **Action-following behavior after robot action adaptation, beyond aggregate scores.** (a) CD-LAM reduces FDCE in most action categories (full breakdown in Figure E.2) at both 2B and 14B scales, so the gain is not concentrated in a single primitive. (b) On the 2B ACWM after robot action adaptation, CD-LAM shifts the rollout distribution toward lower FDCE and higher PSNR.

F LTX Adaptation: Implementation Details

Integrating CD-LAM into LTX. We introduce CD-LAM conditioning into the pretrained LTX-2.3 22B backbone (HaCohen et al., 2026; Lightricks, 2026) through adaptive layer normalization (AdaLN). Given a 32-dimensional latent action extracted by CD-LAM, an action embedder produces an embedding e_a, which is projected and combined with the diffusion-timestep embedding e_τ:

$$c = e_\tau + P_a(e_a). \tag{F.1}$$

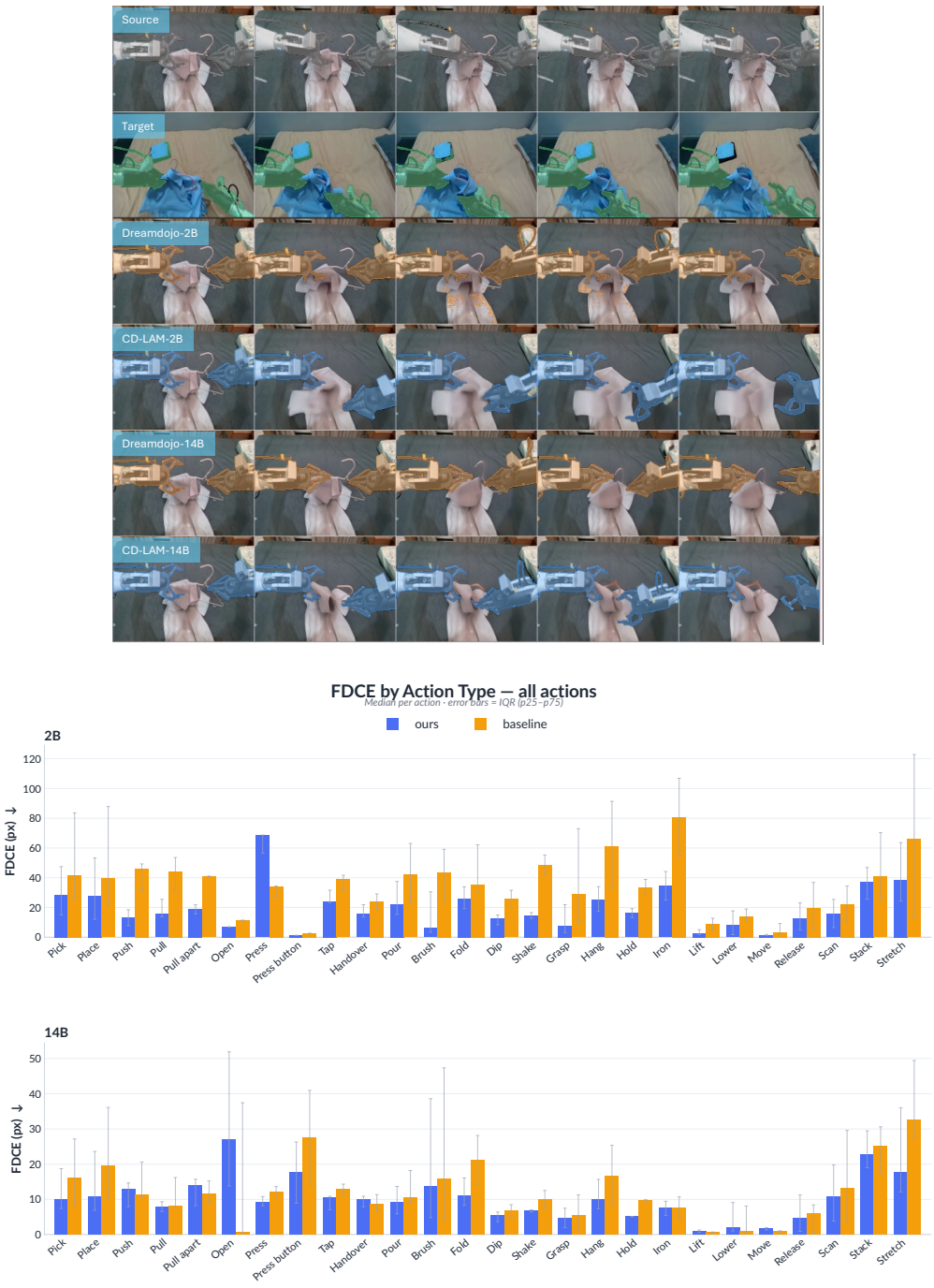


Figure E.2: **Additional target-action and per-action results. Top:** Target-action transfer example. The source-context row provides the fixed visual context, the target-motion row provides the robot action sequence, and rollouts are generated under the same target action. CD-LAM transfers the target motion more reliably at both scales. **Bottom:** Per-action FDCE breakdown (median; error bars show the inter-quartile range). CD-LAM lowers FDCE across most categories.

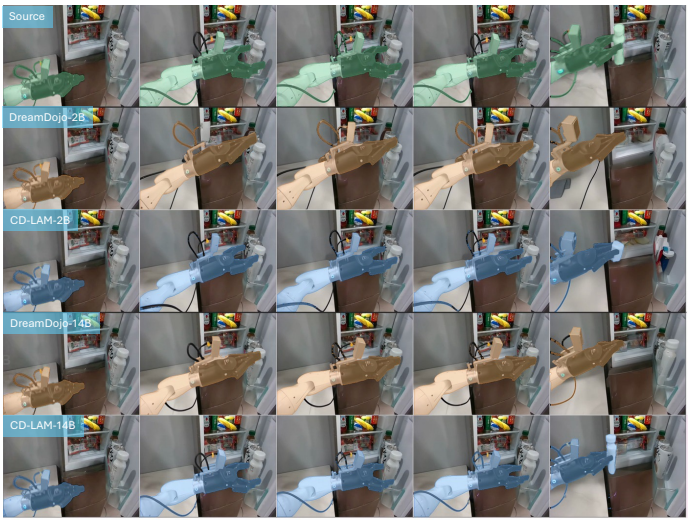


Figure E.3: **Representative rollouts after robot action adaptation at 2B and 14B scale.** Rows: source video, DreamDojo, and CD-LAM at 2B and 14B; all model rows start from the same initial frame and receive the same robot action sequence. DreamDojo's arm pose drifts from the ground-truth trajectory, and scaling to 14B does not repair the drift, whereas CD-LAM tracks the commanded motion at both scales. Scene appearance stays comparable, so the separation is in action following rather than visual quality. Colored overlays are row markers distinguishing the models, not model outputs.

Each DiT block maps c to channel-wise scale, shift, and residual-gate parameters for its attention and feed-forward sublayers. CD-LAM latents therefore condition the backbone through AdaLN modulation rather than additional cross-attention tokens. During subsequent robot-action adaptation, a learned bridge maps 22-dimensional executable actions into the same 32-dimensional conditioning space.

CD-LAM-conditioned midtraining. After introducing this conditioning pathway, we midtrain LTX with CD-LAM latent actions for 60k optimizer updates on approximately 20,000 hours of egocentric video. This stage uses 8 H200 GPUs, a per-GPU micro-batch size of 2, and gradient accumulation over 4 micro-batches, yielding an effective global batch size of 64 and 3.84M processed examples. The video duration describes the midtraining corpus, while the processed-example count includes repeated sampling.

Robot-action adaptation. We subsequently perform full-model robot-action adaptation on the same paired AgiBot video and action data used for DreamDojo adaptation. Adaptation uses 7 H200 GPUs with the same per-GPU micro-batch size and gradient accumulation, yielding an effective global batch size of 56. We evaluate checkpoints at 17k, 23k, and 28k adaptation updates, corresponding to 0.952M, 1.288M, and 1.568M processed examples, respectively; the main text reports the 28k checkpoint. All step counts refer to optimizer updates rather than individual gradient-accumulation micro-steps. Adaptation counts restart after, and exclude, the preceding 60k CD-LAM-conditioned midtraining updates.

Evaluation and additional checkpoints. Given an initial frame and a sequence of executable robot actions, the adapted model generates an autoregressive video rollout. We use the same held-out AgiBot evaluation protocol as the main robot-action study. Table F.1 reports all evaluated checkpoints, including LPIPS in addition to the PSNR and FDCE metrics shown in the main text. Metric definitions follow Appendix C. These results evaluate action-conditioned generation within the AgiBot data domain and do not establish generalization across unseen domains.

Table F.1: **LTX robot-action adaptation checkpoints.** All checkpoints follow the same 60k-update CD-LAM-conditioned midtraining stage. Processed examples count robot-action adaptation only and include repeated sampling. PSNR is in dB and FDCE in pixels.

Updates	Examples	PSNR ↑	LPIPS ↓	$\text{FDCE}_{\text{mean}}$ ↓	FDCE_{med} ↓
17k	0.952M	20.48	0.165	12.57	8.11
23k	1.288M	20.64	0.160	11.78	8.01
28k	1.568M	20.38	0.172	11.75	7.82

G VLA Policy Experiment: Protocol and Full Results

Setup. We compare latent-action pretraining with the released DreamDojo LAM and with CD-LAM. Each X-VLA policy is initialized from scratch and pretrained on the same 700,695 clips from EgoVerse (Punamiya et al., 2026), without robot data at this stage, to predict the latent actions extracted by its respective LAM. The 32-dimensional latent actions from each LAM are projected to 20 dimensions using the same LAM-specific PCA procedure. We keep one policy-pretraining checkpoint fixed for each LAM and repeat LIBERO post-training with three paired random seeds. Within each pair, the two policies use the same data order, optimization recipe, and 80k-update budget. Table G.3 lists the complete recipe.

Evaluation. We evaluate the official X-VLA LIBERO checkpoint and both LAM-based policies on all 10,030 LIBERO-Plus variants using a shared task list, initial states, environment seed, per-suite action budgets (220/280/300/520 steps), 10 settling steps, 10 denoising steps, and FP32 inference. Each policy retains its training-compatible action chunk, action interface, and proprioception. An episode succeeds only when the simulator's task predicate is met within the action budget; failed episodes remain in the denominator. Our primary LIBERO-Plus metric is overall success rate, computed as the number of successful episodes divided by 10,030. We additionally report the unweighted mean across the seven perturbation categories as a secondary metric. For the two LAM-based policies, we report the mean and sample standard deviation over three paired post-training seeds. The official X-VLA checkpoint is evaluated once on the same complete benchmark.

Training Dynamics. Figure G.1 shows three paired post-training trajectories on fixed monitoring subsets. Training beyond 50k resumes from model weights with the optimizer reinitialized. The main comparisons use the fixed 80k checkpoints evaluated on the complete benchmarks.

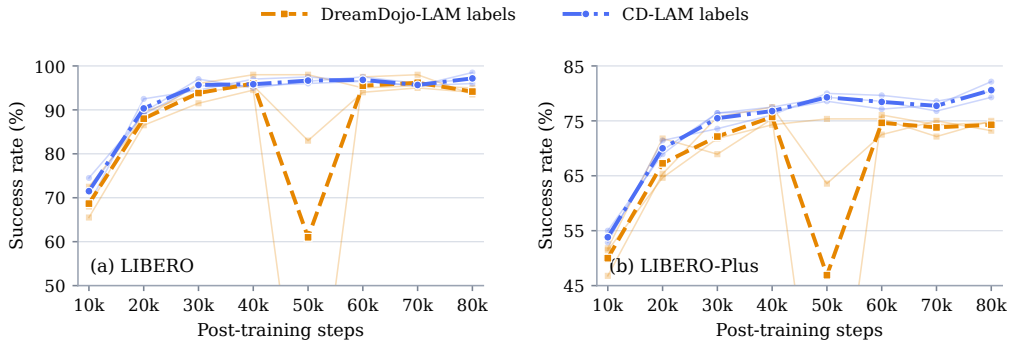


Figure G.1: **Success rate across post-training checkpoints.** Left: LIBERO. Right: LIBERO-Plus. Bold curves show means over three paired post-training seeds, thin curves show individual seeds, and shading spans their minimum and maximum. Curves use fixed subsets of 200 LIBERO episodes and 280 LIBERO-Plus variants. The main comparison uses the fixed 80k endpoint evaluated on the complete benchmark. For readability, the displayed y-axis ranges are 50–100% (LIBERO) and 45–85% (LIBERO-Plus).

Robustness Breakdown. At the fixed 80k endpoint, pretraining with CD-LAM outperforms pretraining with DreamDojo LAM in all three paired runs (Table 6). The mean improvement is also positive in every LIBERO-Plus perturbation category, with the largest gains under object-layout, background-texture, and lighting variations (Table G.1).

Table G.1: **LIBERO-Plus success rate (%) by perturbation category at 80k updates.** Mean ± sample standard deviation over three paired post-training seeds. Δ is CD-LAM minus DreamDojo LAM.

Category	n	DreamDojo LAM	CD-LAM	Δ (pp)
Background textures	1,076	67.29 ± 0.34	$\mathbf{72.49 \pm 1.61}$	$+5.20$
Camera viewpoints	1,599	92.47 ± 0.75	$\mathbf{93.14 \pm 0.72}$	$+0.67$
Language instructions	1,537	74.80 ± 0.39	$\mathbf{75.71 \pm 2.00}$	$+0.91$
Lighting conditions	1,142	88.97 ± 0.63	$\mathbf{92.88 \pm 1.54}$	$+3.91$
Object layouts	1,525	69.07 ± 0.04	$\mathbf{75.39 \pm 0.30}$	$+6.32$
Robot initial states	1,550	44.47 ± 0.39	$\mathbf{45.91 \pm 0.48}$	$+1.44$
Sensor noise	1,601	92.48 ± 1.65	$\mathbf{93.28 \pm 0.81}$	$+0.79$

Table G.2: **Overall policy success and uncertainty.** Rollout-weighted success rate (%), reported as mean ± sample standard deviation over three paired post-training seeds.

Evaluation	DreamDojo LAM	CD-LAM	Δ (pp)
LIBERO-Plus	75.69 ± 0.33	$\mathbf{78.25 \pm 0.37}$	$+2.56$
Standard LIBERO	95.72 ± 0.06	$\mathbf{96.30 \pm 0.51}$	$+0.58$

Table G.3: **Training recipe of the two pretrained policies** (only the LAM teacher changes, and latent coordinates and bridge are refit with the same procedure).

Item	DreamDojo LAM teacher	CD-LAM teacher	Relationship
Teacher	DreamDojo LAM (released 400k ckpt)	CD-LAM 100h	changed
Policy pretraining initialization	X-VLA from scratch, seed 0	X-VLA from scratch, seed 0	same
Post-training seeds	0, 1, 2	0, 1, 2	paired
Downstream training length	80,000 updates	80,000 updates	same
EgoVerse pretraining clips	700,695	700,695	same rows / order / splits
Latent coordinates	teacher-specific 32-D → PCA 20-D	teacher-specific 32-D → PCA 20-D	same fitting recipe
Policy pretraining steps / batch	32,846 / 128	32,846 / 128	same
Bridge fit steps / batch / seed	10k / 2048 / 20260720	10k / 2048 / 20260720	same
LIBERO train / validation episodes	1,592 / 120	1,592 / 120	same
Downstream steps / learning rate	80,000 / 10^{-4}	80,000 / 10^{-4}	same
VLM / bridge LR coefficients	0.1 / 0.1	0.1 / 0.1	same
Trainable scope	bridge + all X-VLA	bridge + all X-VLA	same
Bridge auxiliary / execution weights	1.0 / 0.3	1.0 / 0.3	same
Camera augmentation	agent view only, p=0.7; dropout 0.1	agent view only, p=0.7; dropout 0.1	same

Table G.4: **LIBERO-Plus macro success by post-training seed at 80k updates.** Secondary evaluation using an unweighted average across the seven perturbation categories.

LAM teacher	Seed 0	Seed 1	Seed 2
DreamDojo LAM	75.89	75.78	75.29
CD-LAM	78.62	78.49	78.09

Official checkpoint evaluation and training differences. We evaluate the official X-VLA LIBERO checkpoint ourselves, obtaining 98.05% on standard LIBERO and 69.74% on LIBERO-Plus (6,995/10,030 successful episodes). The LIBERO-Plus evaluation uses the same variant list, initial states, environment seeds, action budgets, and inference-noise rules as our LAM-based policies, while retaining each policy's native action interface. The official checkpoint uses its original pretraining and LIBERO fine-tuning recipe, with absolute end-effector actions and no camera augmentation. Our LAM-based policies use human-video latent-action pretraining, a latent-to-action bridge, and camera augmentation. Thus, the official checkpoint provides a reference under a shared evaluation protocol; the matched comparison for latent-action debiasing is CD-LAM versus DreamDojo LAM.

G.1 RoboTwin Clean-to-Random Evaluation

Evaluation setting. We evaluate policy robustness in the clean-to-random (C2R) setting of RoboTwin 2.0. Policies are post-trained only on demonstrations from clean environments, then evaluated in both clean and randomized scenes. Randomization changes clutter, textures, lighting, and table height, testing transfer beyond the visual conditions seen during post-training.

Subset construction. The 12-task subset was fixed from public leaderboard results before observing any results from our models. We first identify eligible tasks with official X-VLA clean-scene success of at least 80%, randomized-scene success of at least 35% for at least one other reported model, and an official episode horizon of at most 700 steps. Among these, we select the ten tasks with the largest X-VLA clean-to-random success drops (69–81 percentage points). We add `adjust_bottle` and `place_container_plate` as anchor tasks, with official X-VLA randomized-scene success of 33–39%, to include moderate-difficulty cases. This subset targets pronounced C2R degradation rather than average performance across all 50 RoboTwin tasks. Table G.5 lists all tasks and results.

Policy training. All three policies start from the same official pretrained X-VLA initialization. The latent-action policies run 20k intermediate updates using 31 robot datasets, with all RoboTwin data excluded, and EgoVerse/EgoTouch human videos occupying one third of each batch. Robot samples supervise both executable actions and latent actions; human samples supervise latent actions only. Data, sampling, optimization, and update budgets are matched between the two latent-action policies, differing only in whether the latent targets are provided by DreamDojo LAM or CD-LAM. The X-VLA baseline skips intermediate training. All policies then post-train on 50 clean demonstrations per task for 20k updates using absolute positions, 20-dimensional dual-arm actions, 30-step action chunks, three cameras, batch size 128, and learning rate 10^{-4}. Inference directly predicts executable robot actions without latent-action inputs.

Rollout protocol. We evaluate each policy on 25 clean and 50 randomized episodes per task, using identical initial scenes across policies. Scenes are generated following the official procedure, starting from index 100000 and retaining expert-solvable instances. Task configurations are checked against the official version, and episode horizons follow the official setup. Randomized evaluation uses task names rather than the leaderboard's unseen descriptions, so leaderboard scores are not directly comparable. All policies use one post-training seed. Each latent-training policy has one simulation timeout, counted as failure. Aggregate success is the unweighted mean over the 12 tasks, equivalent to pooled episode success within each scene condition because evaluation counts are equal across tasks.

Statistical analysis. We compute paired 95% confidence intervals using 2000 bootstrap replicates, resampling matched episodes within each task. CD-LAM improves randomized-scene success over DreamDojo LAM by 5.0 percentage points, with interval $[1.8, 8.2]$; discordant successes are 62 versus 32 episodes. Its clean-scene difference is 2.7 points, with interval $[-0.7, 6.0]$. Relative to skipping intermediate training, the randomized gain is 8.8 points, with interval $[5.3, 12.3]$. These confidence intervals quantify rollout variability for the evaluated tasks and fixed trained policies.

Table G.5: **Task-level success rates (%) on RoboTwin C2R.** All matched policies use the same initial scenes: 25 clean and 50 randomized episodes per task. X-VLA skips intermediate training; LAM and CD-LAM use matched intermediate training.

	Clean			Randomized		
Task	X-VLA	LAM	CD-LAM	X-VLA	LAM	CD-LAM
lift_pot	96	100	100	30	24	38
grab_roller	100	100	100	52	42	60
rotate_qrcode	84	88	84	22	18	26
place_phone_stand	64	76	84	4	4	4
move_playingcard_away	92	80	76	18	18	12
beat_block_hammer	88	100	96	8	10	18
place_object_basket	72	36	52	2	16	20
place_empty_cup	84	88	88	32	40	48
place_bread_skillet	72	68	76	6	6	4
open_laptop	80	92	84	38	62	58
adjust_bottle	100	100	100	62	62	66
place_container_plate	96	80	100	26	44	52
Mean	85.7	84.0	86.7	25.0	28.8	33.8