Causally Debiased Latent Action Model for Embodied Action Conditioned World Models

Yufan Wei[1,2*], Kun Zhou[1†], Lingjun Mao[1,2*], Zijun Zhang[1], Ziming Xu[1], Ziqiao Xi[1], Shuang Liang[1,2*], Ruobing Han[1], Yuchen Yan[1], Xinyue Wang[1,2*], Fan Feng[1], Biwei Huang[1]

[1]Aether AI [2]University of California, San Diego

[*]Done during an internship at Aether AI.

[†]Project Lead & Corresponding author: franciskunzhou@gmail.com

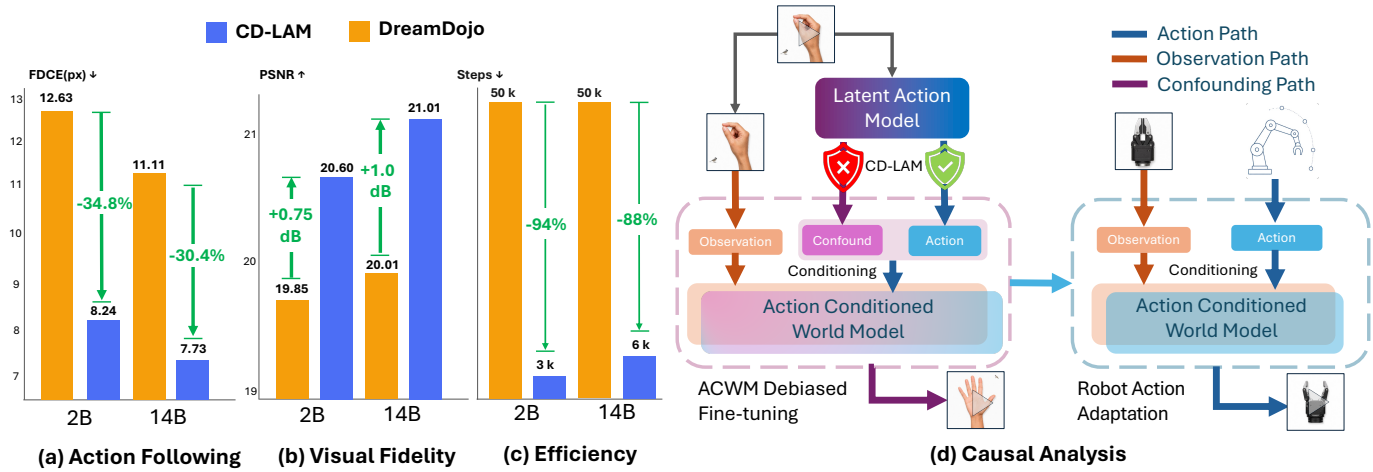


Fig. 1. **Performance overview and the underlying confounding mechanism.** CD-LAM substantially improves action following, visual fidelity, and data efficiency on DreamDojo. (a) CD-LAM lowers embodiment action-following error (FDCE) at both 2B and 14B. (b) CD-LAM also raises PSNR at both scales (gains annotated in dB). (c) CD-LAM uses 3k and 6k robot action adaptation updates for the final 2B and 14B checkpoints, compared with DreamDojo's 50k-update reference; crossing curves in Fig. 8. (d) Causal analysis: a reconstruction-trained LAM can leak action-irrelevant confounders into the action condition; CD-LAM reduces the measured shortcut dependence along this path, and (a)–(c) quantify the resulting gains.

***Abstract*—Action-conditioned world models (ACWMs) aim to simulate future observations conditioned on embodied actions, offering a promising foundation for robot planning, policy evaluation, and data augmentation. However, learning controllable ACWMs requires large-scale action-labeled data, which remains costly to collect in the real world. 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 factors such as backgrounds and non-interacted objects. In this work, we identify this action-irrelevant bias as a key obstacle to controllable ACWMs and introduce evaluation metrics to measure latent-action bias, action following, and robustness. We propose CD-LAM, a causally debiased framework for LAM-based ACWMs. CD-LAM introduces three debiasing objectives used during fine-tuning: embodiment-centric reconstruction, action-centric contrastive learning, and latent space calibration, which together encourage embodiment-focused, action-aware, and well-calibrated, non-collapsed latent action representations. Experiments on 2B and 14B ACWM backbones show that CD-LAM substantially improves latent-action controllability, downstream robot action following, visual fidelity, and adaptation efficiency: at 14B, CD-LAM matches the DreamDojo reference with more than 12× fewer robot action adaptation updates and surpasses it at the 6k final checkpoint.**

Code **Project Page** **Model**

***Index Terms*—latent action models, world models, causal debiasing**

I. Introduction

Action conditioned world models (ACWMs) [1]–[4] have emerged as a promising paradigm for simulating the physical world directly from visual observations and embodied control signals. Given the current visual 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 the futures of 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, whereas collecting robot videos with action annotations remains costly in the real world [5], [6]. In

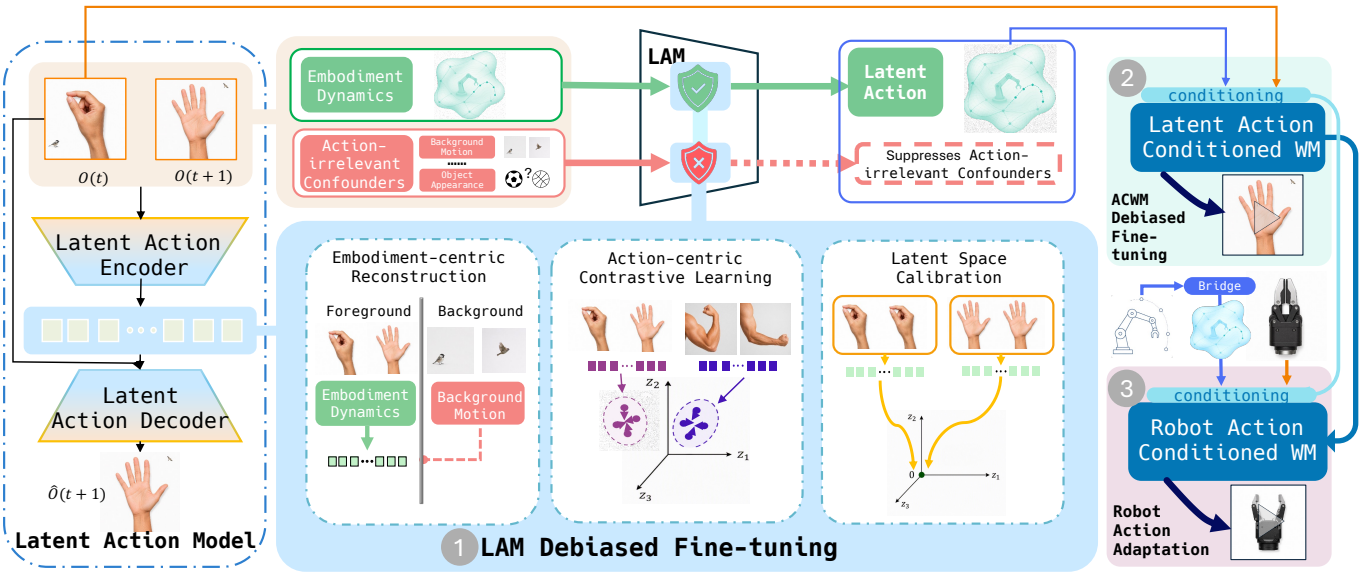


Fig. 2. **Overview of CD-LAM.** CD-LAM debiases the LAM's latent action space in three stages while keeping the downstream action conditioning format unchanged. **Stage 1 (LAM debiased fine-tuning)** debiases the LAM with the three CD-LAM objectives; **Stage 2 (ACWM debiased fine-tuning)** trains the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

contrast, human videos are easier to collect and far more scalable, and contain rich physical interactions, but they typically lack executable action annotations [7]–[9]. To leverage such large-scale data sources, latent action models (LAMs) offer a bridge by inferring compact latent action representations from unlabeled video transitions [10]–[12]. Through large-scale pretraining 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 [13], [14].

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. As a result, action-irrelevant factors such as background scenes and non-interacted objects may leak into the latent action representation. Our empirical analysis in Section III 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 target action condition and becomes fragile under small perturbations [15]–[17]. We therefore aim to debias LAMs by making latent action representations more causally grounded in embodiment-centered action features, while disentangling action-irrelevant factors and mitigating representation collapse. Concretely, we introduce three causally debiased objectives for LAMs: embodiment-centric reconstruction, action-centric contrastive learning, and latent space calibration, which respectively encourage embodiment-focused, action-aware, and calibrated non-collapsed action representations.

To this end, we propose CD-LAM, a causally debiased method for LAM-based ACWMs. CD-LAM consists of an efficient three-stage fine-tuning pipeline (Fig. 2) that first debiases the LAM, then debiases the ACWM, and finally adapts it to real-world robot actions. In Stage 1, we fine-tune the LAM with embodiment-centric reconstruction, action-centric contrastive learning, and latent space calibration to emphasize embodiment-centered action dynamics. In Stage 2, we fine-tune the ACWM on debiased latent actions extracted from unlabeled videos, to debias the learned controllability. In Stage 3, we add a lightweight action-to-latent mapping layer to bridge robot actions into the latent space and adapt the ACWM to executable robot controls.

With these objectives, the two debiasing stages require only 1k and 2k updates, respectively. Empirically, CD-LAM improves all three aspects summarized in Fig. 1 across both 2B and 14B backbones (Section V). First, ACWM rollouts conditioned on the debiased latent actions exhibit lower action-following error: FDCE drops by 42% and 26% at the 2B and 14B scales. Second, the gains persist after robot action adaptation: FDCE further drops by 35% and 30%, accompanied by improved visual fidelity, with the debiased 14B model achieving the best performance on every metric. Third, the debiased latent space is substantially cheaper to adapt: CD-LAM matches the DreamDojo baseline with more than 12× fewer robot action adaptation updates. Together, these results show that targeted LAM debiasing is an effective approach to improving controllability with limited robot action data.

This paper makes four contributions:

- We analyze action-irrelevant bias in existing LAMs and introduce metrics for measuring latent-action bias, controllability, and robustness in LAM-based ACWMs.
- We propose CD-LAM, a causally debiased framework built on three LAM objectives: embodiment-centric reconstruction, action-centric contrastive learning, and latent space calibration.

- We show that CD-LAM achieves effective and efficient debiasing, improving action following, visual fidelity, and adaptation efficiency across 2B and 14B backbones while matching the DreamDojo reference with more than 12× fewer robot action adaptation updates.
- We release our debiased LAMs, debiased ACWMs, evaluation protocols, and training code to facilitate future research on controllable embodied world models.

II. Preliminaries

A. 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 denote the visual observation at time t and u_t denotes a recorded executable robot action, such as a relative end-effector displacement. The action sequence $u_{t:t+H-1}$ may come from a policy, a planner, or a human operator. Typically, ACWMs are trained by conditional next-video prediction, using diffusion-style video losses for this conditional distribution.

B. 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 [10]–[12]. 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.

C. ACWM Debiased Fine-tuning of the ACWM

Prior LAM-based ACWM systems use latent actions as substitutes for missing robot actions during ACWM debiased fine-tuning 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 [10]–[14]. 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. We examine whether such a variable is fit for real robot action conditioning in Section III.

III. Confounder Analysis on Latent Actions

In a LAM-based ACWM, z_t is the only action signal the world model receives during ACWM debiased fine-tuning, 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.

A. Why 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 such purity. 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 Fig. 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. The result is weak robot action following.

B. Empirical Study on Representative LAM-based ACWMs

We next check whether the analyzed failure appears in practice, using DreamDojo [14] as the representative LAM-based ACWM. The evidence has two layers: weak action following ability and confounded latent action representations.

Weak Robot Action Following. After robot action adaptation, the generated embodiment dynamics should follow the conditioning action. However, DreamDojo ACWMs are prone to violate this requirement in two basic settings, both of which keep the initial frame fixed and replace only the conditioning action. We select several examples for qualitative analysis. First, under a zero action, we perform the action replacement, denoted as $\mathrm{do}(u_t = 0)$. Although the embodiment motion should be suppressed, the rollout keeps moving (Fig. 3). 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 (Fig. 4). These examples indicate that the ACWM does not reliably follow the supplied robot actions.

Action-irrelevant Confounding in Latent Actions. We further examine the latent action space directly. A clean latent

Fig. 3. **Zero robot action inputs still produce motion.** Frames are generated by the 2B DreamDojo ACWM after robot action adaptation on the AgiBot dataset [18], with the initial frame fixed and all relative robot action inputs replaced with zero, $\mathrm{do}(u_t = 0)$. The rollout still produces embodiment motion.

Fig. 4. **The rollout does not follow the transferred target action.** 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.

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 types of bias, we design corresponding evaluation metrics (Appendix A) and apply them to the DreamDojo LAM. As shown in Table I, 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. These results indicate that the latent actions learned by the DreamDojo LAM are significantly biased by action-irrelevant confounders.

IV. CD-LAM: Causal Debiasing of Latent Actions

Section III shows that the LAMs trained with reconstruction-only objective can turn z_t into a confounded latent action. Motivated by this analysis, we propose CD-LAM, a causally motivated debiasing approach implemented through an efficient staged fine-tuning pipeline.

Table I. **LAM action-irrelevant confounding audit.** CD-LAM (our method, Section IV) reduces zero-transition, camera-shift, and episode-context responses. Shortcut leakage is the gap between different-action same-episode pairs and same-action different-episode pairs. All diagnostics evaluate the LAM encoder alone, before any world model rollout; a clean latent action should respond weakly on all three diagnostics (lower is better).

Diagnostic	DreamDojo LAM	CD-LAM
Zero-transition response (static pair (o_t, o_t); rel. norm ↓)		
Median response	0.527	**0.043**
Absolute latent norm (median)	3.119	**0.226**
Camera-shift robustness (synthetic translation; rel. norm ↓)		
Horizontal shift (mean / median)	0.555/0.536	**0.156/0.096**
Vertical shift (mean / median)	0.545/0.529	**0.110/0.064**
Action-neighbor structure (hard-negative diagnostic, ↓)		
Shortcut leakage	0.151	**0.014**

A. Design Principles

The analysis in Section III leads to three design principles: action-irrelevant factors should be kept out of the latent action space, while visual context stays available through the observation path.

- **Embodiment-centric Reconstruction.** The reconstruction signal should emphasize regions tied to embodiment dynamics over background regions.
- **Action-centric Structure.** The latent action space should have the structure to group similar action videos, instead of visually similar but action-different ones.
- **Calibrated, Non-collapsed 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 ($\mathcal{L}_{\text{cal}}$).

B. 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

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

Here $\lambda_{\text{ctr}}(k)$ and λ_{cal} weight the contrastive and calibration terms, where $\lambda_{\text{ctr}}(k)$ varies with the training step k (we reserve t for frame time).

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 Eq. (2). Let $M_t \in [0,1]^{h \times w}$ be the embodiment–object foreground mask obtained by

SAM3 [19], where $h \times w$ is the frame resolution. We define the spatial weight

$$W_t = \alpha_{\text{fg}} M_t + \alpha_{\text{bg}}(1 - M_t), \qquad \alpha_{\text{fg}} > \alpha_{\text{bg}}, \tag{8}$$

and optimize

$$\mathcal{L}_{\text{emb}} = \frac{1}{|\Omega|} \left\| W_t^{1/2} \odot (\hat{o}_{t+1} - o_{t+1}) \right\|_2^2. \tag{9}$$

Here Ω denotes the pixel grid of the frame. This loss emphasizes reconstruction of embodiment-dynamics regions while retaining a nonzero background weight for global visual consistency.

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 $v_i = \text{norm}(r_\omega(z_i^{\text{CD}}))$, and let $y_{ij} = +1$ for same-primitive pairs and $y_{ij} = -1$ otherwise. We use

$$\mathcal{L}_{\text{ctr}} = \frac{1}{|\mathcal{P}|} \sum_{(i,j)\in\mathcal{P}} \text{softplus}\big(- y_{ij}(\tau v_i^\top v_j + b)\big). \tag{10}$$

Here r_ω is an auxiliary projection head used only by this loss, and τ and b are a learned temperature and bias [20]. Same-primitive pairs are pulled together and different-primitive pairs are pushed apart. The primitive labels are coarse verb-level categories (e.g., pick–place, pour, open, close) obtained by clustering caption verbs into a 12-way primitive space (Appendix B); they contain no executable robot actions, so this term shapes the latent action representation without turning CD-LAM into supervised robot action learning.

3) Latent Space Calibration

The latent space calibration loss has two roles:

$$\mathcal{L}_{\text{cal}} = \mathcal{L}_{\text{KL-fb}} + \mathcal{L}_{\text{zero}}. \tag{11}$$

The free-bit KL term $\mathcal{L}_{\text{KL-fb}}$, as a KL penalty applied only above a per-dimension free-bits floor, provides capacity control, preventing z_t from storing arbitrary transition information while preserving variation among ordinary transitions. The zero-transition calibration term $\mathcal{L}_{\text{zero}}$ anchors duplicated-frame inputs to a zero-transition reference:

$$\mathcal{L}_{\text{zero}} = \mathbb{E}_{o_t}\left[\left(\left[\frac{\|z_t^0\|_2}{\text{sg}(s_\Delta) + \epsilon} - m_{\text{zero}} \right]_+ \right)^2 \right]. \tag{12}$$

Here $z_t^0 = \mu_\phi(o_t, o_t)$ is the zero-transition latent from Section III-B, s_Δ is the running RMS norm of ordinary transition latents, $\text{sg}(\cdot)$ denotes stop-gradient, $[x]_+ = \max(x, 0)$, m_{zero} is a small margin, and ϵ is a small constant. $\mathcal{L}_{\text{zero}}$ pushes the norm of zero-transition latents below m_{zero} times the running RMS of ordinary transition latents. This calibrates the zero-transition reference without forcing ordinary transition latents to collapse.

C. Multi-stage Training

CD-LAM is used in a three-stage training pipeline (Fig. 2). The first two stages operate on action-unlabeled video (ACWM debiased fine-tuning), while the final stage introduces paired robot actions (robot action adaptation).

1) **Stage 1 (LAM Debiased Fine-tuning).** We first fine-tune the LAM on action-unlabeled videos using the CD-LAM objective of Eq. (7).
2) **Stage 2 (ACWM Debiased Fine-tuning).** We then extract debiased latents z_t^{CD} from video transitions and continue ACWM training under the conditioning of Eq. (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.
3) **Stage 3 (Robot Action Adaptation).** Finally, for paired robot action data (o_t, u_t, o_{t+1}), we align executable robot actions to the debiased latent action space. A lightweight MLP bridge g_η is trained to regress the gradient-stopped CD-LAM latent, $g_\eta(u_t) \approx \text{sg}\big(\mu_\phi(o_t, o_{t+1})\big)$, 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 executable robot actions enter the same debiased latent action space used in Stage 2.

V. Experiments

We evaluate CD-LAM along three aspects. First, latent action understanding: does the LAM's representation capture embodiment dynamics rather than action-irrelevant confounders? Second, downstream action following: does the ACWM follow latent action and robot action conditions under fixed visual context? Third, robot action adaptation efficiency and data leverage: can a small LAM debiasing stage reduce the robot action adaptation cost and produce large downstream gains?

A. Experimental Setup

Models and baseline. We apply CD-LAM debiasing at two ACWM backbone scales, 2B and 14B, which share a single debiased LAM. DreamDojo [14] with its original reconstruction-trained LAM is the baseline; CD-LAM replaces only the LAM's latent action space and keeps the ACWM architecture, latent dimension, and conditioning format unchanged, so all comparisons isolate the effect of the LAM debiasing stage. The debiasing stage additionally uses SAM3 foreground masks and coarse primitive labels (Appendix B) as training signals; the baseline does not use these.

Training. Training follows the three stages of Section IV-C and runs on 96 H100 GPUs. LAM debiased fine-tuning uses 1k optimizer steps with per-GPU batch size 32; unless otherwise stated, CD-LAM models use the 100h debiasing data tier (Table IV varies this tier from 1h to 1000h). ACWM Debiased Fine-tuning uses 2k optimizer steps, with per-GPU batch size 12 (2B) and 2 (14B). For the main results after robot action adaptation, we report the final checkpoints of the aligned runs: 3k steps at 2B and 6k at 14B. In all efficiency comparisons,

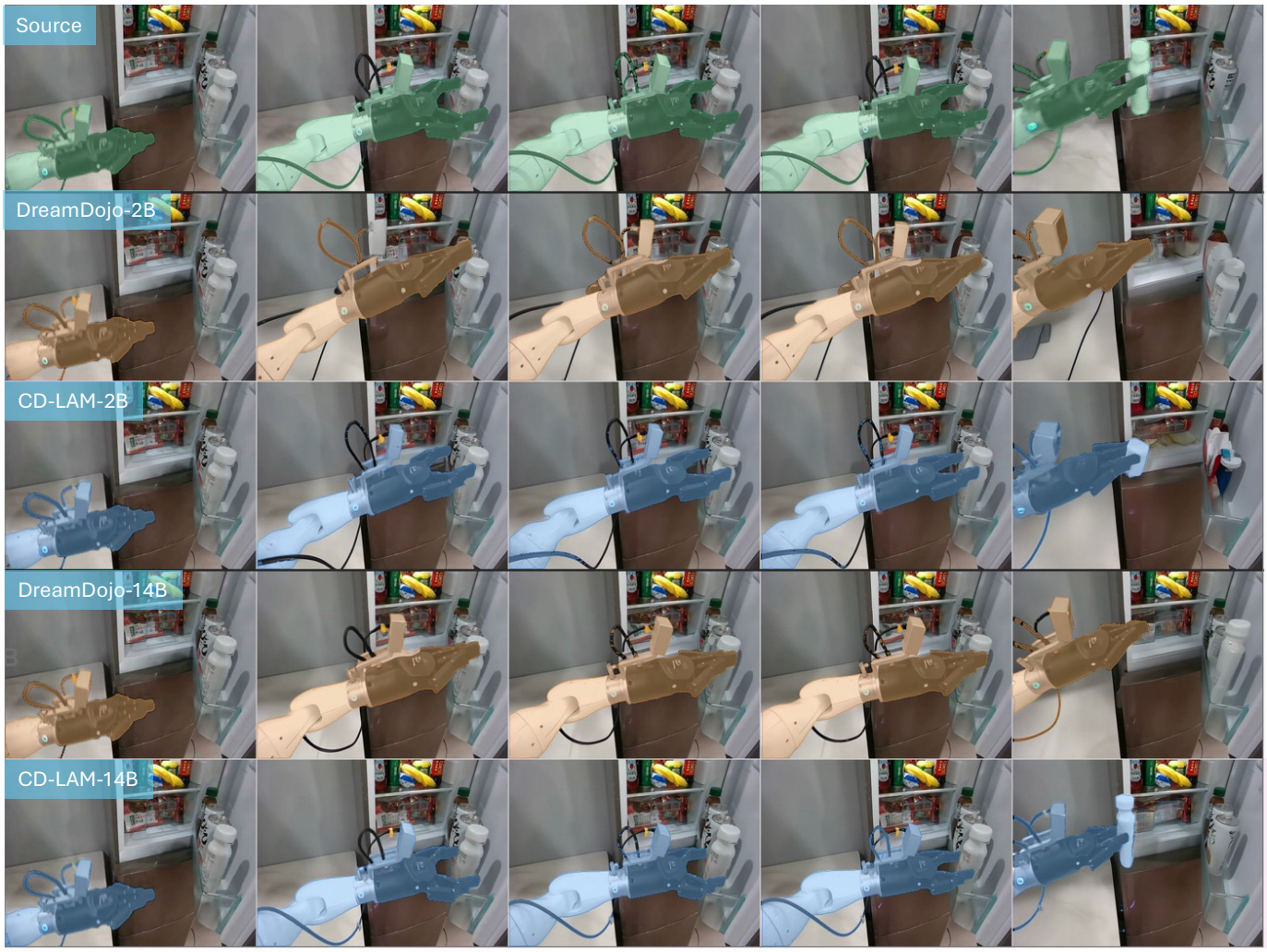


Fig. 5. **Representative rollouts after robot action adaptation at 2B and 14B scale.** Rows: ground truth, DreamDojo, and CD-LAM at 2B and 14B; all model rows start from the same initial frame and receive the same robot action sequence. DreamDojo's arm pose drifts from the ground-truth trajectory and scaling to 14B does not repair the drift, while 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.

step counts denote optimizer updates under the *aligned* protocol: both models use the same robot action adaptation data, batch size, and optimizer settings, so step counts are directly comparable. The 50k-step reference in Fig. 1(c) and the dashed lines in Fig. 8 denote DreamDojo's full original robot action adaptation budget.

Evaluation data. Latent action conditioned rollouts after ACWM debiased fine-tuning are evaluated on 300 held-out clips from EgoDex [21], an egocentric human-manipulation corpus matching the action-unlabeled stages; robot action rollouts and direct interventions after robot action adaptation are evaluated on 300 clips drawn from distinct episodes of AgiBot [18], a real-robot dataset with paired executable robot actions.

Intervention protocol. Interventions follow Section III-B: 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.

Metrics. We evaluate visual fidelity with PSNR, foreground PSNR (FG-PSNR), SSIM, and LPIPS. These metrics measure whether a rollout is visually close to the reference, but they do not directly test whether embodiment dynamics follow the action condition. To quantify action following, we report Foreground Displacement Chamfer Error (FDCE), a symmetric Chamfer distance between generated and reference foreground displacement tracks, where lower is better. Foreground masks select embodiment and interacted-object regions using SAM3 [19], and point tracks are computed only within valid foreground regions using CoWTracker [22]. We sample up to 16 valid foreground anchors per rollout pair. As illustrated in

Fig. 6(a), FDCE compares induced foreground displacement rather than raw pixel appearance. FDCE is measured in pixels; we report both the mean, which is sensitive to occasional large failures, and the median, which reflects typical behavior. The full metric definition and reporting conventions are provided in Appendix A. As Fig. 6(b) shows, a rollout can score well on PSNR while missing the commanded motion, so reconstruction quality alone is not a sufficient test of controllability.

B. Latent Action Understanding Audit (Stage 1)

We first audit the latent action before any world model rollout. This directly tests the upstream source identified in Section III: whether the encoder $(o_t, o_{t+1}) \mapsto z_t$ responds to action-irrelevant confounders as if they were embodied action.

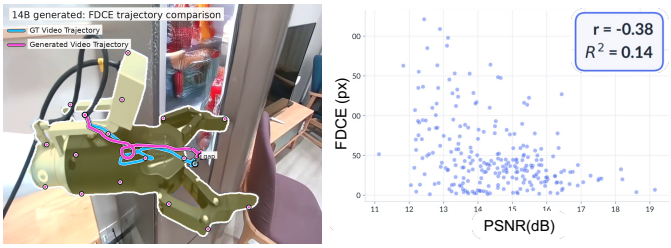
Each diagnostic in Table I targets one confounder in Eq. (5): the zero-transition response tests whether a duplicated-frame input still yields an action-like latent, the camera-shift response tests source-side visual factors V_t, and shortcut leakage tests scene context C_t. Table I shows that CD-LAM strongly reduces responses to action-irrelevant confounders while preserving local action structure: the median zero-transition response drops from 0.527 to 0.043, synthetic camera shifts induce 3.6–8.3$\times$ smaller responses, and shortcut leakage falls from 0.151 to 0.014. Preservation is verified with the same hard-negative probe: same-primitive pairs from different episodes keep an essentially unchanged cosine similarity (0.132 vs. 0.131), so CD-LAM debiases the latent action space toward embodied transition semantics rather than uniformly shrinking it. Stage 1 therefore delivers the repair that Section III calls for; the next two subsections track how it propagates downstream.

C. ACWM Debiased Fine-tuning Rollouts (Stage 2)

We next test whether the debiased LAM also debiases the world model through ACWM debiased fine-tuning, using latent action conditioned rollouts. The world model consumes latent actions z_t extracted by the LAM directly, without the robot action bridge, so any gain is attributable to the latent action condition itself rather than to robot action alignment.

Table II shows consistent gains at both scales. On latent action rollouts, 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. Transfer pixel metrics remain nearly unchanged, whereas FDCE improves substantially, indicating that the main gain is in motion following rather than pixel similarity.

The debiasing indeed propagates: no LAM output is scored here, only ACWM rollouts, so the gains show that Stage 2 carries the Stage-1 repair into the world model. Moreover, scale does not substitute for debiasing: the baseline's FDCE *worsens* from 2B to 14B (34.00 to 40.29, and 42.74 to 50.27 under transfer), so a larger backbone amplifies visual capability but not action following, whereas CD-LAM improves both scales.



(a) FDCE computation (b) PSNR–FDCE scatter

Fig. 6. **Visual fidelity and action following are complementary.** (a) FDCE measures foreground motion consistency by comparing generated and reference foreground displacement tracks with a symmetric Chamfer distance. (b) Pixel-level fidelity does not certify action following: PSNR explains only limited variance in FDCE ($r = -0.38$, $R^2 = 0.14$).

D. Robot Action Adaptation Rollouts (Stage 3)

We then evaluate downstream action following after robot action adaptation. Here the experimental input is the executable robot action u_t, which enters the ACWM through $\hat{z}_t = g_\eta(u_t)$, so this test asks whether the ACWM follows executable robot actions after they are mapped into the debiased latent action space.

Table III is the main system-level result: CD-LAM strengthens embodiment action following while improving visual fidelity. $\mathrm{FDCE}_{\mathrm{mean}}$ drops by 35% at 2B (12.63 to 8.24) and by 30% at 14B (11.11 to 7.73), with consistent improvements in $\mathrm{FDCE}_{\mathrm{med}}$, PSNR, SSIM, and LPIPS at both scales. At 14B, CD-LAM further improves on its 2B counterpart in every column of Table III, whereas baseline scaling is mixed: $\mathrm{FDCE}_{\mathrm{med}}$ *worsens* from 8.15 to 8.98 even as PSNR improves. This complements Section V-C: scale does not substitute for debiasing, but it compounds with it. Qualitative rollouts are shown in Fig. 5; Fig. 7 confirms that the gain (a) spans action categories rather than a single primitive and (b) shifts the whole rollout distribution toward lower FDCE and higher PSNR. A full per-action breakdown and an additional transfer example are provided in Fig. A.1, where CD-LAM is on par with or slightly behind the baseline on a few categories. Evaluation categories follow AgiBot's action annotations and are distinct from the 12-way training primitives of Appendix B.

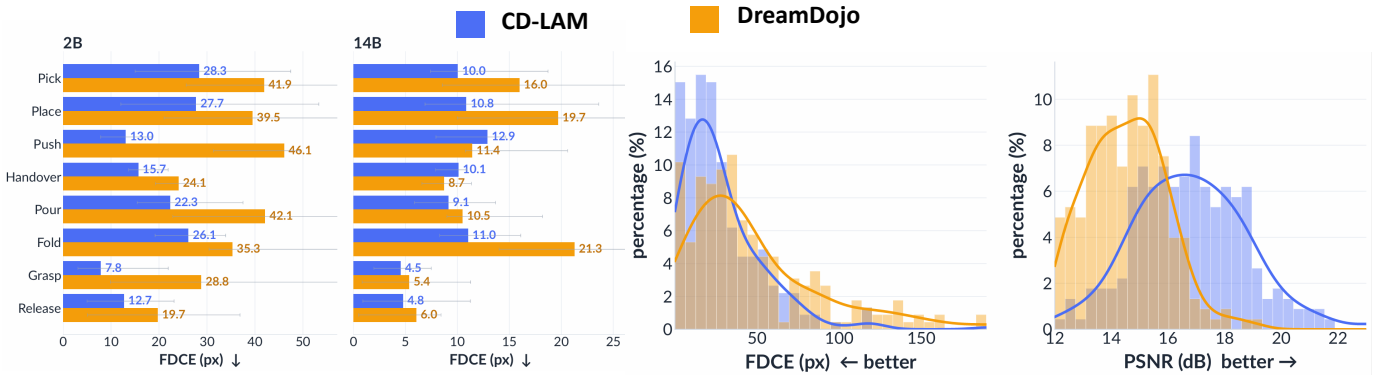
Direct Action Following Interventions. The intervention columns of Table III apply the two interventions of the evaluation protocol and show directly that the rollout is more sensitive to the supplied action condition. Under the zero-action intervention $\mathrm{do}(u_t = 0)$, residual FDCE is halved at 2B (10.71 to 5.03) and cut to less than a quarter at 14B (9.36 to 2.18): the rollout now largely stays still when the action specifies no movement, which is precisely the behavior that Fig. 3 showed the baseline lacks. 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. That the largest relative gains appear exactly where conditioning is stress-tested is consistent with improved sensitivity to the action input: robot action adaptation aligns robot actions to a space that now encodes embodiment dynamics rather than action-irrelevant confounders.

Table II. **Latent action conditioned rollouts after ACWM debiased fine-tuning.** Left block: rollouts conditioned on latent actions z_t from the video transition. Right block: target-action transfer $\mathrm{do}(z_t = z_t^{\mathrm{tar}})$ under a fixed source context. Right-block pixel differences are marginal; the operative comparison there is FDCE. Bold marks the better model within each backbone; lower FDCE is better.

		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**

Table III. **2B/14B ACWM results after robot action adaptation and direct interventions.** Evaluated on AgiBot. The zero-action intervention $\mathrm{do}(u_t = 0)$ tests whether a zero action suppresses embodiment motion; its FDCE is computed against a static reference (initial frame held fixed), so the column reads as residual motion and is not comparable with the rollout columns. Target-action transfer $\mathrm{do}(u_t = u_t^{\mathrm{tar}})$ tests action following under a fixed source context. Bold marks the better model within each backbone; lower FDCE is better.

		robot action u_t rollout					$\mathrm{do}(u_t = 0)$	$\mathrm{do}(u_t = u_t^{\mathrm{tar}})$
Backbone	Model	$\mathrm{FDCE}_{\mathrm{mean}}$ ↓	$\mathrm{FDCE}_{\mathrm{med}}$ ↓	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**



(a) FDCE by action type

(b) Rollout distribution

Fig. 7. **Action-following behavior after robot action adaptation, beyond aggregate scores.** (a) CD-LAM reduces FDCE across the eight action categories shown (full breakdown in Fig. A.1) at both 2B and 14B scales, showing that 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.

E. Robot Action Adaptation Efficiency and Debiasing Data Scaling

The main results above compare final checkpoints. We further ask how quickly the downstream ACWM benefits once the LAM's latent action space is debiased. Under the aligned robot action adaptation protocol, Fig. 8 shows that at 14B, CD-LAM reaches the DreamDojo reference within roughly 3k updates on FDCE and 4k updates on PSNR, and clearly surpasses it on both metrics by the 6k final checkpoint. This is more than 12× fewer updates than the 50k reference. It is the compute side of the less-is-more effect: because the ACWM no longer has to unlearn a confounded action condition, a short Stage-1 debiasing pass replaces a large amount of downstream robot action adaptation compute. Fig. 1(c) summarizes the endpoint: the final aligned checkpoints use 3k (2B) and 6k (14B) updates against the 50k reference.

Table IV evaluates the data side of the same effect. Even the 1h CD-LAM tier reduces $\mathrm{FDCE}_{\mathrm{mean}}$ from 12.63 to 8.91 while preserving PSNR, and the 1000h tier further lowers it to 7.97. The 1h tier thus already captures about 80% of the 1000h tier's FDCE improvement (3.72 of 4.66 points), so the benefit comes mainly from debiasing itself rather than from debiasing-data scale. The large early gain and the smaller but consistent gains from larger tiers support a less-is-more pattern: targeted LAM debiasing with limited data unlocks most of the downstream controllability improvement, while additional data continues

to refine embodiment action consistency.

F. Objective Ablation

Finally, we ablate the three CD-LAM objective components to connect the empirical gains back to the design principles in Section IV-A. Table V reports three readouts to identify which failure mode each term controls: rollout fidelity after ACWM debiased fine-tuning, LAM camera-shift response, and robot action FDCE. FDCE in this ablation is measured under a separate ablation evaluation setup, so its absolute values are not comparable with Table III.

Embodiment-centric weighting protects foreground fidelity and action following: removing it reduces FG-PSNR by $0.31\,\mathrm{dB}$ and worsens robot action FDCE by $1.17\,\mathrm{px}$, while leaving the camera-shift response nearly unchanged. Action-centric contrast has little effect on PSNR or camera-shift response (identical at reported precision), but removing it worsens robot action FDCE by $1.84\,\mathrm{px}$, indicating that its main role is to organize action-consistent transition neighborhoods rather than sharpen frames.

Zero-transition calibration is the dominant term for suppressing action-irrelevant camera-shift response. Removing it increases the horizontal/vertical camera-shift response from $0.133/0.101$ to $0.637/0.637$, about $4.8\times/6.3\times$ larger. Removing it nevertheless lowers FDCE on this split; the table footnote explains why we do not read this as better debiasing. Each term thus controls the failure mode it was designed for, matching the design principles of Section IV-A: the three objectives are complementary rather than redundant.

VI. Related Work

Latent Action Models from Action-unlabeled Video. A latent action model reads a frame pair and infers a compact latent action that summarizes the change between them, so that a decoder or world model can predict the next frame. The idea descends from learning to act by watching unlabeled video: imitating latent policies from observation [23] and large-scale video pretraining for control [24]. Genie [10] introduced this video-to-action abstraction at scale with a discrete latent action codebook; LAPO [11] recovers latent actions from observation alone to bootstrap policies, and LAPA [12] learns discrete latent actions by quantizing inter-frame transitions; AdaWorld [13] learns such latent actions as a transferable condition for fast world model adaptation; and embodied-manipulation variants such as Moto [25] and IGOR [26] tokenize inter-frame motion or image-goal change into a shared latent action space. More recent variants explore additively compositional latent actions [27], co-evolving latent action world models [28], and cross-viewpoint action-centric latent actions [29]. The shared principle is *associative*: the latent action is defined by what improves next-frame reconstruction (echoing masked-reconstruction representation learning [30]), and the dominant lever is to scale data and codebook size.

World Models and Action Conditioned World Models. A *world model* compresses experience into a learned latent dynamics that can be rolled out to imagine futures for planning or control [1], [2], and video foundation models such as UniSim [31] and Cosmos [32] scale this to high-resolution, long-horizon prediction over web-scale video. An *action conditioned world model* additionally conditions the rollout on an action, predicting the future *given* a control input rather than merely continuing the video. This is what turns a passive video predictor into a controllable simulator for embodied agents [3], [4], and it is increasingly trained over large cross-embodiment corpora [5], [6]. The *latent action* world model is the ACWM variant whose conditioning action is a LAM's latent action rather than a recorded control command, so the latent action condition learned during latent action based training on action-unlabeled video can later drive an embodied agent: DreamDojo [14] instantiates this LAM–ACWM recipe and is the baseline we compare against.

Metrics for Controllable Video and Embodied Motion.

Table IV. **Scaling of the debiasing data.** Each tier debiases the LAM with the stated hours of video, then repeats the same aligned robot action adaptation on the 2B backbone. All tiers share the same 1k-step Stage-1 budget, so the comparison isolates data quantity at fixed compute.

Model	PSNR ↑	$\mathrm{FDCE}_{\mathrm{mean}}$ ↓	$\mathrm{FDCE}_{\mathrm{med}}$ ↓
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**

Table V. **Objective ablation of CD-LAM.** PSNR and FG-PSNR are measured on rollouts of the latent action conditioned ACWM after ACWM debiased fine-tuning, not on LAM decoder reconstruction. Camera-shift response is the relative latent response to synthetic horizontal/vertical image shifts, and robot action FDCE is measured on rollouts after robot action adaptation. All columns are measured under a separate ablation evaluation setup and are comparable within this table only.

Objective	PSNR ↑	FG-PSNR ↑	Cam-shift x/y ↓	FDCE ↓
Full CD-LAM	**29.64**	**26.57**	**0.133 / 0.101**	17.23
w/o emb.-centric weighting	29.53	26.26	0.134 / 0.100	18.40
w/o action-centric contrast	**29.64**	**26.57**	0.134 / 0.101	19.07
w/o zero-trans calibration	29.31	26.48	0.637 / 0.637	**16.10**†

† This variant obtains lower robot action FDCE on this split, but it fails the upstream camera-shift diagnostic. We therefore do not interpret it as better action debiasing; the evidence for zero-transition calibration is its suppression of action-irrelevant camera-shift response.

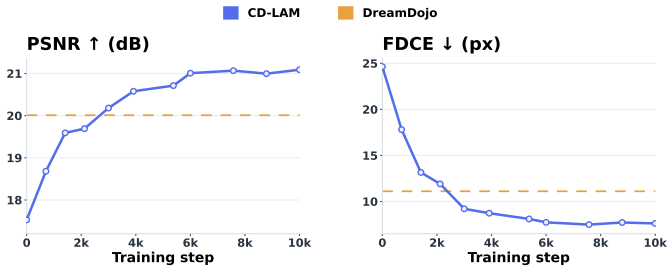


Fig. 8. **Robot action adaptation efficiency.** Under the aligned protocol, CD-LAM crosses the DreamDojo reference within 3k–4k updates (more than $12\times$ fewer than the 50k reference), and clearly surpasses it by the 6k final checkpoint. Curves show the 14B model on a monitoring subset; absolute values are not directly comparable with Table III.

Video world models are most often scored by pixel- and distribution-level fidelity (e.g., PSNR and FVD [33]), which reward sharp, plausible frames but are largely insensitive to whether the commanded action was followed: a model can copy context and score well while ignoring the latent action. Motion-following evaluation instead tracks where things actually move: MotionPro [34] scores object-motion control as an average trajectory distance (ObjMC) over points propagated by a point tracker, building on point-tracking benchmarks and methods [35]–[37]. We adopt this tracking-based stance but specialize it to embodied manipulation, where our FDCE metric measures foreground motion over Segment-Anything masks [38], [39] (SAM3 in our pipeline [19]) with CoWTracker-tracked [22] delta-trajectories. This distinction is central to our evaluation: reconstruction is necessary but not sufficient, and pixel and motion metrics can rank latent action spaces differently.

Causal and Debiased Representation Learning. A parallel line of work removes spurious or non-causal factors from learned representations, via invariance across training environments [40], analyses of shortcut learning [41], and causally motivated corrections in imitation learning, where policies latch onto nuisance correlates of expert actions [42]. These methods debias the *inputs* of a policy or classifier; CD-LAM instead debiases a *conditioning variable*: the latent action that a downstream world model treats as an intervention handle, where confounding corrupts controllability rather than accuracy.

Prior work primarily scales latent actions as predictive representations for video reconstruction. We instead focus on action-irrelevant confounding in the latent action representation space and show that targeted LAM debiasing can improve downstream controllability and robot action adaptation efficiency.

VII. Conclusion

We presented CD-LAM, a causally debiased framework for improving the controllability of LAM-based action-conditioned world models. Starting from the observation that reconstruction-only LAM training can encode action-irrelevant visual factors into latent actions, we analyzed how such biased representations confound downstream ACWMs, leading to weak action following and poor robustness despite visually plausible rollouts. To address this issue, CD-LAM introduces three lightweight debiasing objectives: embodiment-centric reconstruction, action-centric contrastive learning, and latent space calibration, which jointly promote embodiment-focused, action-aware, and calibrated non-collapsed latent action representations. Through an efficient three-stage fine-tuning pipeline, CD-LAM first debiases the LAM, then debiases the ACWM, and finally adapts the model to executable robot actions. Experiments across 2B and 14B backbones demonstrate that CD-LAM improves latent-action controllability, robot action following, visual fidelity, robustness, and adaptation efficiency while matching the DreamDojo reference with more than $12\times$ fewer robot action adaptation updates.

REFERENCES

[1] D. Ha and J. Schmidhuber, "World models," *arXiv preprint arXiv:1803.10122*, 2018.

[2] D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap, "Mastering diverse domains through world models," *arXiv preprint arXiv:2301.04104*, 2023.

[3] H. Xue, Y. Chen, L. Ma, Z. Zhao, L. Moukheiber, Y. Zhu, and Y. Chen, "ACWM-Phys: Investigating generalized physical interaction in action-conditioned video world models," *arXiv preprint arXiv:2605.08567*, 2026.

[4] J. Wu, S. Yin, N. Feng, X. He, D. Li, J. Hao, and M. Long, "iVideoGPT: Interactive VideoGPTs are scalable world models," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2024, arXiv:2405.15223.

[5] Open X-Embodiment Collaboration, "Open X-Embodiment: Robotic learning datasets and RT-X models," *arXiv preprint arXiv:2310.08864*, 2023.

[6] A. Khazatsky, K. Pertsch *et al.*, "DROID: A large-scale in-the-wild robot manipulation dataset," in *Robotics: Science and Systems (RSS)*, 2024.

[7] K. Grauman, A. Westbury, E. Byrne *et al.*, "Ego4D: Around the world in 3,000 hours of egocentric video," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[8] K. Grauman, A. Westbury, L. Torresani *et al.*, "Ego-Exo4D: Understanding skilled human activity from first- and third-person perspectives," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[9] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray, "Scaling egocentric vision: The EPIC-KITCHENS dataset," *European Conference on Computer Vision (ECCV)*, 2018.

[10] J. Bruce, M. Dennis, A. Edwards, J. Parker-Holder, Y. Shi, E. Hughes, M. Lai, A. Mavalankar, R. Steigerwald, C. Apps, Y. Aytar, S. Bechtle, F. Behbahani, S. Chan, N. Heess, L. Gonzalez, S. Osindero, S. Ozair, S. Reed, J. Zhang, K. Zolna, J. Clune, N. de Freitas, S. Singh, and T. Rocktäschel, "Genie: Generative interactive environments," in *International Conference on Machine Learning (ICML)*, 2024, arXiv:2402.15391.

[11] D. Schmidt and M. Jiang, "Learning to act without actions," in *International Conference on Learning Representations (ICLR)*, 2024, lAPO; arXiv:2312.10812.

[12] S. Ye, J. Jang, B. Jeon, S. Joo, J. Yang, B. Peng, A. Mandlekar, R. Tan, Y.-W. Chao, B. Y. Lin, L. Liden, K. Lee, J. Gao, L. Zettlemoyer, D. Fox, and M. Seo, "Latent action pretraining from videos," in *International Conference on Learning Representations (ICLR)*, 2025.

[13] S. Gao, S. Zhou, Y. Du, J. Zhang, and C. Gan, "AdaWorld: Learning adaptable world models with latent actions," in *International Conference on Machine Learning (ICML)*, 2025.

[14] S. Gao, W. Liang, K. Zheng, A. Malik, S. Ye, S. Yu, W.-C. Tseng, Y. Dong, K. Mo, C.-H. Lin, Q. Ma, S. Nah, L. Magne, J. Xiang, Y. Xie, R. Zheng, D. Niu, Y. L. Tan, K. R. Zentner, G. Kurian, S. Indupuru, P. Jannaty, J. Gu, J. Zhang, J. Malik, P. Abbeel, M.-Y. Liu, Y. Zhu, J. Jang, and L. Fan, "DreamDojo: A generalist robot world model from large-scale human videos," *arXiv preprint arXiv:2602.06949*, 2026.

[15] C. Zhang, T. Pearce, P. Zhang, K. Wang, X. Chen, W. Shen, L. Zhao, and J. Bian, "What do latent action models actually learn?" in *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[16] A. Nikulin, I. Zisman, D. Tarasov, N. Lyubaykin, A. Polubarov, I. Kiselev, and V. Kurenkov, "Latent action learning requires supervision in the presence of distractors," in *International Conference on Machine Learning (ICML)*, 2025.

[17] W. Dai, K. Lan, J. Zhou, B. Zhao, X. Su, J. Tong, W. Guan, and S. Yang, "ConLA: Contrastive latent action learning from human videos for robotic manipulation," *arXiv preprint arXiv:2602.00557*, 2026.

[18] AgiBot-World-Contributors, Q. Bu, J. Cai, L. Chen, X. Cui, Y. Ding, S. Feng, S. Gao, X. He, X. Hu, X. Huang, S. Jiang, Y. Jiang, C. Jing, H. Li, J. Li, C. Liu, Y. Liu, Y. Lu, J. Luo, P. Luo, Y. Mu, Y. Niu, Y. Pan, J. Pang, Y. Qiao, G. Ren, C. Ruan, J. Shan, Y. Shen, C. Shi, M. Shi, M. Shi, C. Sima, J. Song, H. Wang, W. Wang, D. Wei, C. Xie, G. Xu, J. Yan, C. Yang, L. Yang, S. Yang, M. Yao, J. Zeng, C. Zhang, Q. Zhang, B. Zhao, C. Zhao, J. Zhao, and J. Zhu, "AgiBot World Colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025. [Online]. Available: https://arxiv.org/abs/2503.06669

[19] N. Carion, L. Gustafson, Y.-T. Hu, S. Debnath, R. Hu, D. Suris, C. Ryali, K. V. Alwala, H. Khedr, A. Huang *et al.*, "SAM 3: Segment anything with concepts," *arXiv preprint arXiv:2511.16719*, 2025.

[20] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer, "Sigmoid loss for language image pre-training," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[21] R. Hoque, P. Huang, D. J. Yoon, M. Sivapurapu, and J. Zhang, "EgoDex: Learning dexterous manipulation from large-scale egocentric video," 2025. [Online]. Available: https://arxiv.org/abs/2505.11709

[22] Z. Lai, E. Insafutdinov, E. Sucar, and A. Vedaldi, "CoWTracker: Tracking by warping instead of correlation," 2026. [Online]. Available: https://arxiv.org/abs/2602.04877

[23] A. D. Edwards, H. Sahni, Y. Schroecker, and C. L. Isbell, "Imitating latent policies from observation," in *International Conference on Machine Learning (ICML)*, 2019, iLPO; arXiv:1805.07914.

[24] B. Baker, I. Akkaya, P. Zhokhov, J. Huizinga, J. Tang, A. Ecoffet, B. Houghton, R. Sampedro, and J. Clune, "Video PreTraining (VPT): Learning to act by watching unlabeled online videos," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022, arXiv:2206.11795.

[25] Y. Chen, Y. Ge, W. Tang, Y. Li, Y. Ge, M. Ding, Y. Shan, and X. Liu, "Moto: Latent motion token as the bridging language for learning robot manipulation from videos," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

[26] X. Chen, J. Guo, T. He, C. Zhang, P. Zhang, D. C. Yang, L. Zhao, and J. Bian, "IGOR: Image-GOal representations are the atomic control units for foundation models in embodied AI," *arXiv preprint arXiv:2411.00785*, 2024.

[27] H. Wei, X. Chen, C. Zhang, T. Pearce, J. Chen, A. Lamb, L. Zhao, and J. Bian, "Learning additively compositional latent actions for embodied AI," *arXiv preprint arXiv:2604.03340*, 2026.

[28] Y. Wang, F. Zhang, D.-C. Zhan, L. Zhao, K. Wang, and J. Bian, "Co-evolving latent action world models," *arXiv preprint arXiv:2510.26433*, 2025.

[29] J. M. Lee, D. Lee, S. Ju, T. Cho, J. W. Koo, L. Zhao, S. Hong, and J. Lee, "MVP-LAM: Learning action-centric latent action via cross-viewpoint reconstruction," *arXiv preprint arXiv:2602.03668*, 2026.

[30] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[31] M. Yang, Y. Du, S. K. S. Ghasemipour, J. Tompson, L. P. Kaelbling, D. Schuurmans, and P. Abbeel, "Learning interactive real-world simulators," in *International Conference on Learning Representations (ICLR)*, 2024, uniSim; arXiv:2310.06114.

[32] NVIDIA, "Cosmos world foundation model platform for physical AI," *arXiv preprint arXiv:2501.03575*, 2025.

[33] T. Unterthiner, S. van Steenkiste, K. Kurach, R. Marinier, M. Michalski, and S. Gelly, "Towards accurate generative models of video: A new metric and challenges," *arXiv preprint arXiv:1812.01717*, 2018, fVD.

[34] Z. Zhang, F. Long, Z. Qiu, Y. Pan, W. Liu, T. Yao, and T. Mei, "MotionPro: A precise motion controller for image-to-video generation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[35] C. Doersch, A. Gupta, L. Markeeva, A. Recasens, L. Smaira, Y. Aytar, J. a. Carreira, A. Zisserman, and Y. Yang, "TAP-Vid: A benchmark for tracking any point in a video," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022, arXiv:2211.03726.

[36] C. Doersch, Y. Yang, M. Vecerik, D. Gokay, A. Gupta, Y. Aytar, J. a. Carreira, and A. Zisserman, "TAPIR: Tracking any point with per-frame initialization and temporal refinement," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, arXiv:2306.08637.

[37] N. Karaev, I. Rocco, B. Graham, N. Neverova, A. Vedaldi, and C. Rupprecht, "CoTracker: It is better to track together," in *European Conference on Computer Vision (ECCV)*, 2024.

[38] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, sAM; arXiv:2304.02643.

[39] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson *et al.*, "SAM 2: Segment anything in images and videos," *arXiv preprint arXiv:2408.00714*, 2024.

[40] M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz, "Invariant risk minimization," *arXiv preprint arXiv:1907.02893*, 2019.

[41] R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann, "Shortcut learning in deep neural networks," *Nature Machine Intelligence*, vol. 2, pp. 665–673, 2020.
[42] P. de Haan, D. Jayaraman, and S. Levine, "Causal confusion in imitation learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

APPENDIX A
METRIC DETAILS

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

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

Fig. 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 $+1.0\,\mathrm{dB}$ 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. Foreground masks select embodiment and interacted-object regions using SAM3 [19], and point tracks are computed only within valid foreground regions using CoWTracker [22].

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

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

The average distance between generated track i and reference track j is

$$c_{ij} = \frac{1}{H} \sum_{s=1}^{H} \left\| \hat{a}_i^s - a_j^s \right\|_2. \tag{A.3}$$

Given N_g generated foreground tracks and N_r reference foreground tracks, we compute the symmetric Chamfer distance as

$$\begin{aligned} \mathrm{FDCE}(\hat{o}, o) = & \frac{1}{2N_g} \sum_{i=1}^{N_g} \min_j c_{ij} \\ & + \frac{1}{2N_r} \sum_{j=1}^{N_r} \min_i c_{ij}. \end{aligned} \tag{A.4}$$

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.

Robustness Properties. The protocol is conservative by construction. Anchors are seeded inside an eroded foreground mask, and tracks with low visibility confidence are discarded before scoring, so spurious background points do not enter the distance. The symmetric Chamfer form in Eq. (A.4) scores displacement geometry rather than matched point counts, so the metric is robust to differing numbers of valid tracks. The known failure modes are tracker-limited rather than metric-limited (heavy hand–object occlusion, motion blur under fast manipulation, and tracker drift on textureless grippers), and they inflate FDCE for all compared models on the same clip, biasing comparisons toward the null.

Reporting Conventions. FDCE is reported in pixels at the evaluation resolution. Because occasional rollouts fail catastrophically, the mean is sensitive to these large outliers while the median reflects typical behavior; we therefore 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 I are computed from the posterior mean μ_ϕ on the audit split and reported as medians (camera shifts also as means) over evaluation pairs. The *zero-transition* and *camera-shift* responses feed a duplicated pair and a synthetically shifted pair, respectively, normalized by the typical latent norm of ordinary transitions:

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

where $D = \mathrm{RMS}\left(\|\mu_\phi(o_t, o_{t+1})\|_2\right) + \epsilon$ over the audit split and T_3 translates the frame horizontally or vertically by 3 pixels at the evaluation resolution 320×640. The *shortcut leakage* is the cosine gap

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

where pairs are drawn from the audit split using the coarse primitive labels of Appendix B; the second term alone serves as the action-neighbor preservation check quoted in the main text (0.132 vs. 0.131).

APPENDIX B
COARSE ACTION-PRIMITIVE LABELS

The action-centric contrastive loss (Eq. (10)) 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 videos' caption annotations in three steps. First, we extract and lemmatize the main verb (with its particle) from each clip's caption, so that "picking up" and "picked up" map to the same verb. Second, we coarsely cluster the extracted verbs by semantic similarity, merging synonyms and near-synonyms (e.g., grab, grasp, and take). Third, each cluster is assigned to one of the 12 canonical primitives; clips with no reliable verb remain unlabeled and join no contrastive pair. The labels are verb-level only (no controller states or trajectories) and merely tell $\mathcal{L}_{\mathrm{ctr}}$ 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. Expanding to the ego+robot index raises the labeled count to 91,664 while retaining 12/12 primitive coverage, and the LAM audit split also covers all 12. 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). Clustering deliberately merges phrasing- and viewpoint-dependent verb variants, so $\mathcal{L}_{\mathrm{ctr}}$ never relies on fine-grained caption semantics.

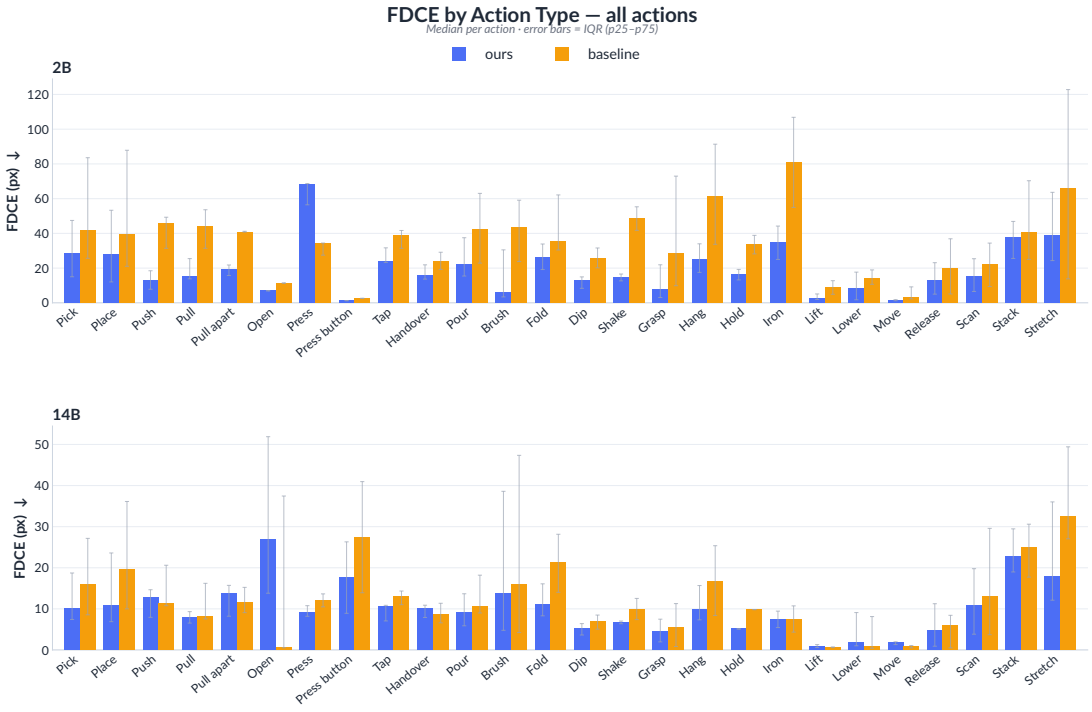


Fig. A.1. **Additional target-action and per-action results. Top:** Target-action transfer example: the source-context row provides the fixed visual context, the target-motion row provides the robot action sequence, and rollouts are generated under the same target action. CD-LAM transfers the target 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.