RECYCLE: TRAINING VISUOMOTOR DIFFUSION POLICIES ON UNLABELED DATA WITH CYCLE-CONSISTENCY

Anonymous authors
Paper under double-blind review

ABSTRACT

Visuomotor diffusion policies are trained by imitation on action-labeled demonstrations, which are orders of magnitude scarcer than action-free video. Classical control couples forward and inverse dynamics through a shared notion of action: the action that explains a transition must also produce it. Many recent diffusion policies already represent this structure: their policy, forward-dynamics, and inverse-dynamics roles are conditionals of one learned distribution, linked by a shared latent action. Yet imitation constrains these conditionals only on the demonstrations, so the denoised trajectory space is optimized for imitation accuracy alone. We constrain it to also encode action-consistent transition structure by enforcing that latent actions inferred from a transition are sufficient for forward evolution under the same policy latent space. The resulting cycle-consistency objective turns action-free video into supervision for the policy without fabricating action labels. Enforced naively on out-of-domain video, however, it degrades performance, for two reasons: unlabeled transitions may not be realizable by the robot, and the cycle gradients conflict with the supervised signal. We address the first with slot-wise latent mixup, which brings unlabeled transitions in-domain while leaving the action variable untouched, and the second with a staged schedule and batch-relative variance reweighting. The resulting recipe, ReCycle, plugs into Cosmos Policy (CP), Unified Video Action Model (UVA), and Video Policy (VP) without changing inference. Using action-free video from physics simulation, other robots, or humans, ReCycle improves CP, UVA, and VP by up to $+5.4$, $+8.3$, and $+40.0$ pp on LIBERO-10 and RoboCasa, and improves CP's real-robot success by $+18.3$ pp on average. Its gains also persist under shifted friction, contact softness, and damping. Gains are largest when the unlabeled transitions contain dynamics relevant to the target task and shrink or reverse when the transition statistics are mismatched.

1 INTRODUCTION

Visuomotor policies are typically trained by imitation on teleoperated demonstrations (Chi et al., 2023; Nasiriany et al., 2024; Liang et al., 2025; Kim et al., 2026), which are expensive to collect and orders of magnitude scarcer than action-free video from simulation, other robots, and humans. We ask: *how can action-free video supervise a visuomotor policy without being converted into pseudo-action labels?* Existing answers fall into three families. (i) *Visual pretraining* (Nair et al., 2022; Ma et al., 2023; Xiao et al., 2022; Radosavovic et al., 2023) shapes what the policy sees but never supervises its actions. (ii) *Forward prediction* (Zhu et al., 2025; Li et al., 2025a) learns to predict future frames, often with the action masked (Fig. 1b), so it need not produce a meaningful action variable. (iii) *Pseudo-action labeling* (Baker et al., 2022; Schmidt & Jiang, 2024; Bruce et al., 2024) infers actions or latent actions before policy training (Fig. 1a), committing to them and inheriting their errors.

We propose a fourth one. Classical control couples forward and inverse dynamics through a shared notion of action: the action that explains an observed transition must also produce that transition under forward simulation (Featherstone, 2008; Khalil & Dombre, 2002). We turn this coupling into a training signal. From two frames of action-free video, an inverse pass infers a latent action and a forward pass must reconstruct the next frame from it (Fig. 1c). The inferred action must therefore be

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.

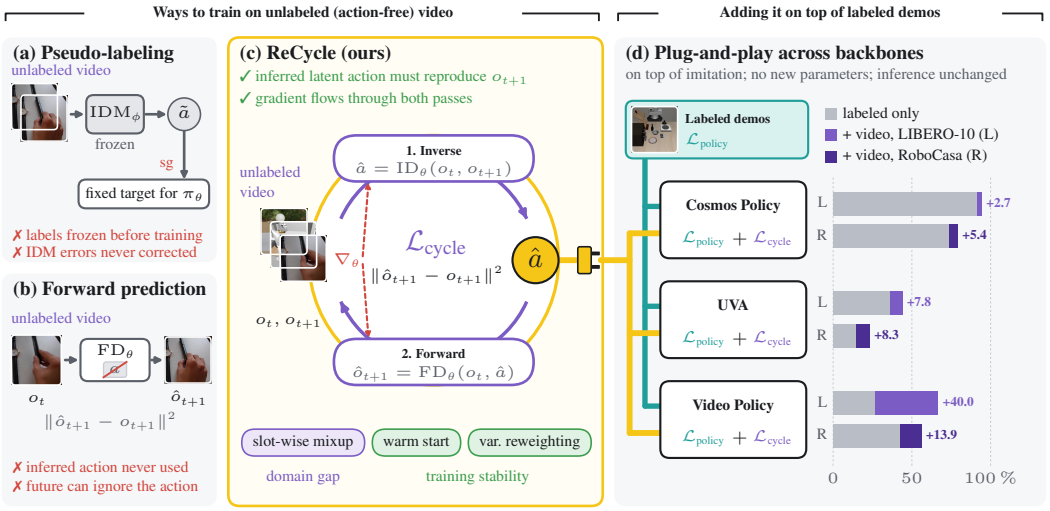


Figure 1: **Training policies on unlabeled video. (a)** Pseudo-labeling freezes IDM-inferred actions. **(b)** Forward prediction can ignore the action. **(c) ReCycle** requires an inferred latent action $\hat{a}$ to reproduce the transition, training both passes end-to-end. **(d) Plug-and-play:** added on top of $\mathcal{L}_{\text{policy}}$ with no new parameters or inference changes, it improves Cosmos Policy, UVA, and Video Policy on LIBERO-10 (L) and RoboCasa (R).

sufficient to reconstruct the transition: unlike forward prediction, the forward model must use it, and unlike pseudo-labeling, it is never frozen. Both passes are trained end-to-end in the representation the policy uses, and the latent is grounded to executable robot actions by the demonstrations rather than recovering the source embodiment's motor commands. Many diffusion policies already support this: Cosmos Policy (CP) through conditioning masks over one denoiser (Kim et al., 2026), Unified Video Action Model (UVA) through decoupled heads over a joint video–action latent (Li et al., 2025a), and Video Policy (VP) through a video model with an action decoder (Liang et al., 2025). In each, the policy $p_\theta(a \mid s)$, forward model $p_\theta(s' \mid s, a)$, and inverse model $p_\theta(a \mid s, s')$ are conditionals of one distribution linked by a *shared latent action*. Yet imitation optimizes the denoised trajectory space for imitation accuracy alone, so nothing makes these conditionals agree beyond the demonstrations. **ReCycle** constrains this space to also encode *action-consistent transition structure* across policy generation, forward roll-out, and inverse inference, by enforcing that latent actions inferred from a transition are sufficient for forward evolution under the same policy latent space. Because the constraint is defined on this shared interface rather than on a particular network, it applies to any generative policy with compatible inverse and forward paths.

Enforced naively on real corpora, the constraint *degrades* performance, for two reasons. Unlabeled video differs from robot data in viewpoint, embodiment, and scene statistics (Yang et al., 2025; O'Neill et al., 2023; Stone et al., 2023), and some of its transitions are not *realizable* by any robot action. In addition, the imitation and cycle losses *conflict* on shared parameters early in training. We address realizability with *slot-wise latent mixup*, which blends only observation latents and leaves the action variable untouched. We address gradient conflict with a supervised warm start and *batch-relative variance reweighting*. None of these adds parameters.

Instantiated in CP, UVA, and VP without changing inference, ReCycle improves each over its own labeled-only baseline, by up to $+5.4$, $+8.3$, and $+40.0$ pp on LIBERO-10 and RoboCasa. It also improves CP's real-robot success by $+18.3$ pp on average. It extracts supervision from physics-diverse synthetic video (PhyCo, rendered with Kubric (Narayanan et al., 2026)), cross-domain robot video, and human video (SSv2 (Goyal et al., 2017)). Its gains also persist under shifted friction, contact softness, and damping. The benefit is not unconditional, however: arbitrary video is not always useful. Action-free video improves the policy when its transition distribution contains dynamics relevant to the target task, as in the contact-rich coffee and sink tasks ($+26.7$ and $+20.0$ pp). It is neutral when demonstrations already cover the task's dynamics, and it can hurt when the transition statistics are mismatched, as with stove control. ReCycle provides a mechanism for extracting policy supervision precisely where the unlabeled transition structure overlaps with the target task.

In summary, our contributions are:

- We introduce ReCycle, a way to learn from action-free video: inferred actions must suffice to reconstruct the transition, with both passes trained end-to-end in the policy's shared transition space, which gives label-free supervision without fixed pseudo-actions.
- We identify transition realizability and policy–cycle gradient conflict as the main reasons naive cycle training fails, and address them with slot-wise latent mixup, a supervised warm start, and batch-relative variance reweighting.
- ReCycle improves Cosmos Policy, UVA, and Video Policy on LIBERO-10 and RoboCasa, and Cosmos Policy on a real robot, using physics-rich synthetic video, cross-domain robot video, or in-the-wild human video. Gains persist under shifted friction, contact softness, and damping.
- We show that the value of action-free video depends on transition overlap: it helps when the video contains dynamics that the target task requires and the demonstrations under-cover, and it can hurt when the transition statistics are mismatched. ReCycle converts this overlap into policy supervision.

2 Related Work

Action-free video for robot policies. Prior work pretrains frozen visual encoders for lightweight policy heads (Nair et al., 2022; Ma et al., 2023; Xiao et al., 2022; Radosavovic et al., 2023; Sermanet et al., 2018), often on large egocentric corpora (Kim et al., 2025; Zheng et al., 2026), or learns actions from video via pseudo-labeling (Baker et al., 2022) or unsupervised latent actions (Schmidt & Jiang, 2024; Bruce et al., 2024; Ye et al., 2025; Pathak et al., 2017; Agrawal et al., 2016). These methods separate pretraining from policy learning or commit to inferred actions. ReCycle instead trains the policy and the self-supervised objective jointly, and keeps updating both inverse and forward paths through shared parameters.

Cycle-consistency and inverse–forward dynamics. Cycle-consistency supervises image translation (Zhu et al., 2017) and video correspondence (Dwibedi et al., 2019; Wang et al., 2019). In robotics, inverse–forward consistency has been used for sim-to-real (Christiano et al., 2016), learning from observation (Edwards et al., 2019), and embodiment-agnostic flow representations (Wang et al., 2025). We apply it inside a diffusion policy whose policy, forward, and inverse roles share parameters, so cycle gradients shape the policy directly.

Diffusion policies and unified video–action models. Diffusion policies (**?**Ze et al., 2024) are increasingly built on video models, either by attaching action modules to pretrained video generators (Liang et al., 2025; Hu et al., 2024; Yang et al., 2025; Feng et al., 2025; Zhong et al., 2025) or by training unified video–action models (Li et al., 2025a; Zhu et al., 2025). VLAs (Brohan et al., 2023; Kim et al., 2024; Black et al., 2024; Intelligence et al., 2025; Bu et al., 2025; Li et al., 2025b) carry weaker temporal priors, and model-based methods (Sutton, 1991; Janner et al., 2019; Hafner et al., 2023; Hansen et al., 2023; Zheng et al., 2025; Jain et al., 2025) keep policy and world model separate. Cosmos Policy (Kim et al., 2026) unifies policy, world model, and value in one video model. ReCycle adds a cycle objective on action-free video to it, and to UVA and Video Policy alike.

3 Method

Section 3.1 formalizes diffusion policies with forward and inverse heads as coupled conditionals over a shared latent action. Section 3.2 derives the action-consistency constraint and its cycle objective on action-free video. Section 3.3 explains the domain gap issue and mitigation.

3.1 Diffusion Policies as Coupled Conditionals over a Shared Latent Action

Classical control couples forward (FD) and inverse dynamics (ID) through a shared notion of action. For a deterministic system with forward dynamics $s' = \mathrm{FD}(s, a)$ and any inverse $a = \mathrm{ID}(s, s')$ that selects an action explaining the transition,

$$\mathrm{FD}\big(s,\ \mathrm{ID}(s, s')\big) = s' \quad \text{for every realizable transition } (s, s'). \tag{1}$$

The action is the variable through which the two maps agree. Consider a diffusion policy that denoises a packed latent sequence

$$z = [\, s,\ a,\ s',\ V\,], \tag{2}$$

where s and s' are current and future observation latents, a is the action latent, and V an optional value. By choosing which slots are observed and which are generated, the same denoiser f_θ realizes

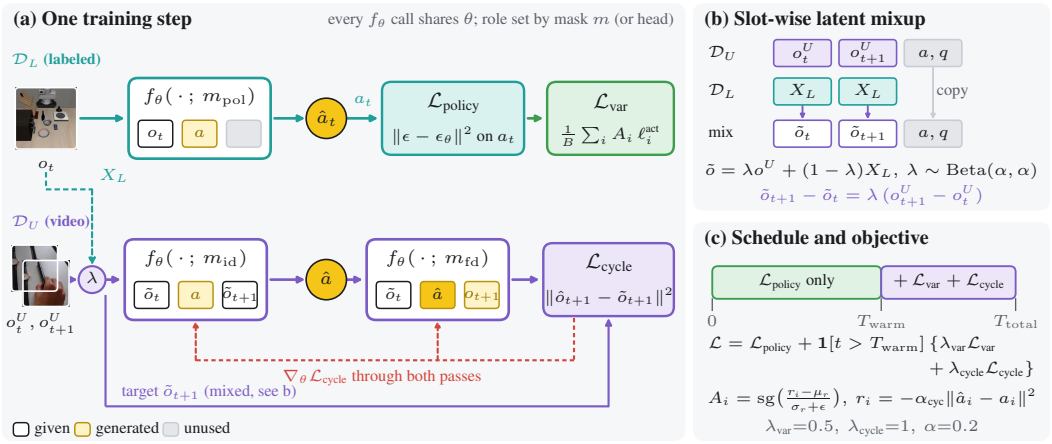


Figure 2: **Method.** (a) One training step. A single network f_θ plays three roles, selected by mask or head: policy, inverse dynamics, and forward dynamics. On labeled demonstrations it is trained with the imitation loss $\mathcal{L}_{\text{policy}}$, reweighted per sample by $\mathcal{L}_{\text{var}}$. On action-free video, pass 1 infers a latent action $\hat{a}$ and pass 2 must reproduce the next frame from it. Gradients of $\mathcal{L}_{\text{cycle}}$ flow through both passes into the shared weights. (b) Slot-wise mixup (with $\alpha = 0.2$) blends only observation latents with a single labeled partner X_L, so the mixed pair remains one real motion scaled by λ; action and proprioception slots are never mixed. Frames in (a) show the blend in pixel space for illustration; mixing is applied to VAE latents. (c) The cycle loss is enabled only after a supervised warm start; without it, the inverse head collapses on out-of-domain video.

several conditionals of the learned joint $p_\theta(s, a, s')$:

$$\text{Policy:} \quad a \sim p_\theta(a \mid s), \qquad \hat{a} = f_\theta([\,s, \cdot, \cdot, \cdot\,];\, m_{\text{pol}}), \tag{3}$$

$$\text{Forward dynamics:} \quad s' \sim p_\theta(s' \mid s, a), \qquad \hat{s}' = f_\theta([\,s, a, \cdot, \cdot\,];\, m_{\text{fd}}), \tag{4}$$

$$\text{Inverse dynamics:} \quad a \sim p_\theta(a \mid s, s'), \qquad \hat{a} = f_\theta([\,s, \cdot, s', \cdot\,];\, m_{\text{id}}), \tag{5}$$

where m is a conditioning mask and '$\cdot$' a slot to be generated. We write $\text{FD}_\theta(s, a)$ and $\text{ID}_\theta(s, s')$ for the denoised outputs of Eqs. (4) and (5). This masked-denoiser form follows Cosmos Policy (Kim et al., 2026), to which we add the inverse mask m_{id}. How this structure is applied to UVA Li et al. (2025a) and VP Liang et al. (2025) is described in the supplementary. A.10.

The action slot is the literal variable linking the three roles: the forward model consumes it to predict the future, the inverse model produces it to explain a transition, and the policy produces it as its imitation target. Standard training, however, supervises these roles only on labeled demonstrations $\mathcal{D}_{\text{L}} = \{(s, a, s')\}$. Each conditional is then fit where demonstrations exist, and nothing forces the three to agree with one another—or with Eq. (1)—on transitions the demonstrations do not cover. The iteratively denoised trajectory space is therefore optimized for imitation accuracy alone: it learns to reproduce expert action distributions but is not constrained to encode how actions, observations, and transitions relate.

3.2 Cycle-Consistency on Unlabeled Video

Our goal is to use abundant action-free video to supervise the policy, but a label mismatch stands in the way: standard policy learning requires action labels that unlabeled video cannot provide. The unified denoiser (Kim et al., 2026) resolves this mismatch architecturally–because the policy, world model, and inverse-dynamics heads share parameters, a constraint enforced through the latter two heads directly updates the policy backbone, with no action labels required. The iteratively denoised trajectory space of a diffusion policy is typically optimized only for imitation accuracy: it learns to reproduce expert action distributions but is not constrained to encode the dynamical relationships between actions, observations, and transitions. We constrain this space to also encode *action-consistent transition* structure by enforcing that the latent action inferred from a transition is sufficient for forward evolution under the same policy latent space. This is the natural self-supervised objective in the shared latent space established above, and it requires no action labels. Let $\mathcal{D}_{\text{L}}$ denote the labeled robot dataset of trajectories with ground-truth actions and $\mathcal{D}_{\text{U}}$ denote the action-free video dataset without ground-truth labels.

Concretely, given a frame pair $(o_t, o_{t+1}) \sim \mathcal{D}_{\mathrm{U}}$, we run a two-pass cycle. The inverse-dynamics mask m_{id} first predicts a latent action that explains the observed transition,

$$\hat{a}_t = f_\theta\big([\, o_t,\, \cdot,\, o_{t+1}\,];\, m_{\mathrm{id}}\big), \tag{6}$$

where the action slot is masked and the model conditions only on the two image latents. The forward-dynamics mask m_{fd} then reconstructs the future from the inferred action,

$$\hat{o}_{t+1} = f_\theta\big([\, o_t,\, \hat{a}_t\,];\, m_{\mathrm{fd}}\big). \tag{7}$$

The cycle loss is the reconstruction error between the predicted and observed next-frame latents,

$$\mathcal{L}_{\text{cycle}} = \mathbb{E}_{(o_t, o_{t+1})\sim\mathcal{D}_{\mathrm{U}}}\Big[\big\|\, \hat{o}_{t+1} - o_{t+1}\,\big\|_2^2\Big], \tag{8}$$

and gradients flow through both passes, training the inverse-dynamics and forward-dynamics heads jointly from video alone.

Unlike forward-only video prediction, this objective enforces two properties. The inferred action $\hat{a}_t$ must be *executable*: passed through the forward model, it must recover the observed future. The forward model must be *action-faithful*: a future that ignores $\hat{a}_t$ fails the reconstruction. The same action variable must therefore explain a transition (inverse), produce it (forward), and stay consistent with the imitation target (policy).

The latent $\hat{a}_t$ is not a robot command; grounding to the robot's action space is left to $\mathcal{L}_{\text{policy}}$ on $\mathcal{D}_{\mathrm{L}}$. What $\mathcal{L}_{\text{cycle}}$ adds is a forward model trained on far richer physical motion than the demonstrations cover. It also yields a representation that explains transitions through controllable latent actions rather than visual correlations. The policy inherits both through the shared backbone.

3.3 Mitigating the Domain Gap with Unlabeled Data

The unlabeled corpus $\mathcal{D}_{\mathrm{U}}$ differs from the labeled robot dataset $\mathcal{D}_{\mathrm{L}}$ in viewpoint, lighting, embodiment, and scene statistics, which makes the cycle objective non-trivial to deploy in practice. Naively activating $\mathcal{L}_{\text{cycle}}$ alongside the supervised policy loss $\mathcal{L}_{\text{policy}}$ from the first training iteration in fact *degrades* performance: cycle gradients from out-of-domain video pull the shared representation away from the manifold on which the policy must act before reliable action features have formed (Tab. 8). The two objectives compete for the same parameters, and no single loss weight resolves the trade-off. We address this with three mechanisms, each targeting a distinct failure mode of the underlying policy/cycle trade-off: *temporal scheduling* of when the auxiliary objective is active, *distributional alignment* of its inputs, and *gradient stabilization* of the supervised signal in its presence.

Curriculum scheduling. We separate the two objectives in time. In the first stage, the model is trained only with $\mathcal{L}_{\text{policy}}$ on labeled demonstrations for a fixed warmup of T_{warm} iterations, allowing the action head to form stable task-relevant representations. In the second stage, analogous to a post-training or fine-tuning phase, we activate $\mathcal{L}_{\text{cycle}}$ and $\mathcal{L}_{\text{var}}$ on $\mathcal{D}_{\mathrm{U}}$ together with the slot-wise mixup described below. Empirically, this staged curriculum is essential: without it, unlabeled video can hurt performance substantially (Tab. 8); with it, the model is able to leverage the unlabeled data to improve generalization.

Slot-wise latent mixup. We narrow the visual domain gap between $\mathcal{D}_{\mathrm{U}}$ and $\mathcal{D}_{\mathrm{L}}$ by mixing Zhang et al. (2018); Verma et al. (2019) only image-style latents, leaving the action variable that carries the structural claim of the method untouched. For each training step, we sample a single mixing coefficient $\lambda \sim \mathrm{Beta}(\alpha, \alpha)$ and use it for all image slots. For every image-style slot $k \in \mathcal{K}_{\text{img}}$, we form a mixed latent

$$\tilde{x}^{(k)} = \lambda x_{\mathrm{U}}^{(k)} + (1-\lambda) x_{\mathrm{L}}^{(k)}, \quad k \in \mathcal{K}_{\text{img}}, \tag{9}$$

where $x_{\mathrm{U}}^{(k)}$ comes from the unlabeled video batch and $x_{\mathrm{L}}^{(k)}$ is a detached latent from the labeled batch in the same training step. We do not mix action, proprioceptive, or value slots: these remain at their unlabeled values throughout the cycle pass, so no ground-truth action information from $\mathcal{D}_{\mathrm{L}}$ leaks into $\mathcal{L}_{\text{cycle}}$. When mixup is enabled, the cycle reconstruction target in Eq. (8) is the mixed future latent $\tilde{o}_{t+1}$ rather than the original o_{t+1}^{U}.

Variance reweighting. The cycle objective can introduce noisy gradients during the cycle-training stage, particularly when the unlabeled videos are visually or physically distant from $\mathcal{D}_{\mathrm{L}}$. To stabilize the supervised signal in the presence of this noise, we add a lightweight reweighting of the labeled action loss using the model's own current action error as a reliability signal. For each labeled sample i in a mini-batch of size B, we compute the action prediction error and a detached proxy reward

$$e_i = \|\hat{a}_i - a_i\|_2^2, \qquad r_i = -\alpha_{\text{cyc}}\, e_i, \tag{10}$$

and form a batch-relative advantage

$$A_i = \text{sg}\left(\frac{r_i - \mu_r}{\sigma_r + \epsilon}\right), \qquad \mathcal{L}_{\text{var}} = \frac{1}{B}\sum_{i=1}^{B} A_i\, \ell_i^{\text{act}}, \tag{11}$$

where μ_r and σ_r are the batch mean and standard deviation of $\{r_i\}_{i=1}^B$, ℓ_i^{act} is the per-sample action denoising loss and $\text{sg}(\cdot)$ denotes stop-gradient. Better-than-average samples receive positive advantage and reinforce their action predictions; worse-than-average samples receive negative advantage and are downweighted. This reweighting requires no added forward passes and reduces the variance of the supervised gradient without a learned critic.

Full training objective. Combining the three terms gives the final loss,

$$\mathcal{L} = \underbrace{\mathcal{L}_{\text{policy}}}_{\text{supervised denoising}} + \mathbf{1}[\,t > T_{\text{warm}}\,]\{\lambda_{\text{var}} \underbrace{\mathcal{L}_{\text{var}}}_{\text{batch-relative reweighting}} + \lambda_{\text{cycle}} \underbrace{\mathcal{L}_{\text{cycle}}}_{\text{unlabeled cycle}}\} \tag{12}$$

where $\mathcal{L}_{\text{policy}}$ and $\mathcal{L}_{\text{var}}$ operate on $\mathcal{D}_{\text{L}}$ and $\mathcal{L}_{\text{cycle}}$ operates on $\mathcal{D}_{\text{U}}$. All three terms share the same denoiser f_θ and add no trainable parameters; the slot-wise mixup modifies the inputs of $\mathcal{L}_{\text{cycle}}$ at training time and likewise adds none.

booktabs,multirow,xcolor

4 Experiments

4.1 Dataset and Benchmark

Simulation benchmarks. We evaluate on **RoboCasa** (Nasiriany et al., 2024) (24 Franka kitchen tasks, 50 human demos each, 10 trials per task across five scenes with unseen objects and held-out styles) and **LIBERO** (Liu et al., 2024) (four Franka suites—Spatial, Object, Goal, Long—of 10 tasks each with 50 demos, following the CP (Kim et al., 2026) evaluation setup).

Table 1: **Within-backbone gains from ReCycle.** Each block compares one backbone with and without ReCycle under identical labeled data, training budget, and evaluation episodes. Mean ± std over 3 seeds.

Method	Eval. scope (L / R)	LIBERO-10	RoboCasa
Main benchmark evaluation			
CP Kim et al. (2026)	full: 10 / 24 tasks	91.7 ± 2.8	73.75 ± 3.1
CP + ReCycle	full: 10 / 24 tasks	$\mathbf{94.4 \pm 3.2}$ (+2.7)	$\mathbf{79.17 \pm 2.4}$ (+5.4)
Plug-in check on other architectures (task subsets)			
UVA Li et al. (2025a)	subset: 3 / 3 tasks	36.1 ± 4.5	15.0 ± 2.8
UVA + ReCycle	subset: 3 / 3 tasks	$\mathbf{43.9 \pm 4.8}$ (+7.8)	$\mathbf{23.3 \pm 4.1}$ (+8.3)
VP Liang et al. (2025)	subset: 3 / 3 tasks	26.7 ± 4.1	42.2 ± 2.4
VP + ReCycle	subset: 3 / 3 tasks	$\mathbf{66.7 \pm 6.7}$ (+40.0)	$\mathbf{56.1 \pm 3.3}$ (+13.9)

Unlabeled video corpora. We use three sources of unlabeled video for $\mathcal{D}_{\text{U}}$: (i) PhyCo (Narayanan et al., 2026), 100K+ synthetic clips with varied friction, restitution, deformation, and applied force under randomized appearance, viewpoint, and lighting (our default); (ii) LIBERO videos with action labels removed, as cross-domain robot video; and (iii) Something-Something v2 (SSv2) (Goyal et al., 2017), ∼220K human hand–object clips across 174 interaction templates. Unless otherwise noted, we sample 2k clips from each source.

4.2 Plug-and-play across methods

Because ReCycle needs only an inverse and a forward path, the same objective plugs into CP, UVA, and VP without changing inference. We measure *within-backbone* gains: each backbone is compared only with itself, with and without ReCycle, under identical labeled data, budget, and evaluation, to show transfer across architectures rather than to rank them. CP, our main evaluation, uses the full suites (LIBERO-10: 10 tasks; RoboCasa: 24 tasks × 10 episodes, trained on 40k iter + 5k iter with 2k PhyCo data). UVA and VP use 3-task subsets of LIBERO-10 (stove + moka pot, both moka pots, cream cheese + butter) and RoboCasa (OpenDrawer, PnPCounterToCab, TurnOnSinkFaucet), with 3 seeds × 20 episodes per task from identical initial states. UVA and VP use the same 2k-clip SSv2 subset. Each baseline is trained for 10k iterations on labeled data, then for 5k more with $\mathcal{L}_{\text{var}}$ and $\mathcal{L}_{\text{cycle}}$ enabled (Eq. 12).

Table 2: Effect of unlabeled data source on success rate under friction and softness perturbations, on coffee tasks. We compare PhyCo (in-domain simulation), LIBERO (cross-domain robot video), and SSv2 (in-the-wild human video) against the Cosmos Policy baseline. On coffee, all three substantially improve over CP, with SSv2 yielding the largest gain (**+17.78** pp avg.).

		Unlabeled Data Source				**Δ vs. CP (pp)**		
Pert.	**Scale**	**None (CP)**	**PhyCo**	**LIBERO**	**SSv2**	**PhyCo**	**LIBERO**	**SSv2**
Coffee tasks (CoffeeSetupMug, CoffeeServeMug, CoffeePressButton)								
Friction	0.50	60.00	73.33	70.00	**76.67**	+13.33	+10.00	**+16.67**
	1.00	50.00	76.67	66.67	**83.33**	+26.67	+16.67	**+33.33**
	2.00	56.67	63.33	**70.00**	**70.00**	+6.67	+13.33	+13.33
Softness	2	60.00	**73.33**	70.00	**73.33**	+13.33	+10.00	+13.33
	4	56.67	66.67	70.00	**73.33**	+10.00	+13.33	**+16.67**
	8	60.00	63.33	**76.67**	73.33	+3.33	**+16.67**	+13.33
Avg.		57.22	69.44	70.56	**75.00**	+12.22	+13.33	**+17.78**

ReCycle improves every backbone on both benchmarks (Tab. 1), most where supervised training leaves the most headroom. VP gains +40.0 pp on LIBERO-10 and +13.9 pp on RoboCasa, while

Table 3: Task-category success rates comparing CP with ReCycle trained on 2k PhyCo videos, weighted by category size. Our method improves overall, with strong gains on Coffee and Sink, but regresses on Stove and Drawer–motivating further study with larger unlabeled corpora.

Category	n	CP	ReCycle	Δ (pp)
Pick & Place	8	60.00	63.75	+3.75
Door	4	97.50	97.50	0.00
Drawer	2	95.00	90.00	−5.00
Stove	2	70.00	55.00	−15.00
Sink	3	76.67	96.67	+20.00
Coffee	3	50.00	76.67	+26.67
Microwave	2	95.00	95.00	0.00
Overall	**24**	**73.75**	**79.17**	**+5.42**

Table 4: Per-task success rates on RoboCasa under varying amounts of unlabeled PhyCo data, each configuration trained for 1k iterations. Adding data generally improves performance, with Coffee saturating beyond 10k videos and Stove improving monotonically through 50k.

Category	Task	5k	10k	50k
Coffee	CoffeeSetupMug	40	50	60
	CoffeeServeMug	70	80	70
	CoffeePressButton	90	90	90
	Average	**66.7**	**73.3**	**73.3**
Stove	TurnOnStove	50	55	65
	TurnOffStove	20	25	25
	Average	**35.0**	**40.0**	**45.0**

a matched control trained for the same 5k iterations without the cycle term degrades to $\approx 20\%$ on RoboCasa, so the gain is not due to extra training. The warm start is essential: without it, UVA's inverse head collapses to a near-constant action on SSv2 (-13.9 pp).

4.3 Effect of Unlabeled Data Source

We test our cycle-consistency framework with three sources of unlabeled video–physics-rich simulation (PhyCo, controlled friction/restitution/deformation), cross-domain robot video (LIBERO with action labels stripped), and in-the-wild human video (SSv2) w.r.t. RoboCasa tasks (coffee)–to disentangle whether *unlabeled video* helps in general or whether *physics-rich video* specifically helps, holding all other settings fixed. Table 2 reports results under friction and softness perturbations. On coffee tasks, all three sources improve over the Cosmos Policy baseline by $+12.2$, $+13.3$, and $+17.8$ pp respectively: gains do not require domain-matched simulation, and SSv2 surprisingly outperforms PhyCo, suggesting motion diversity in human video can compensate for the absence of explicit physical-parameter randomization.

4.4 Task dependency of unlabeled data and cycle loss

Action-free video is not uniformly useful: it helps when its transitions contain dynamics relevant to the target policy and can hurt when their statistics are mismatched (Tab. 3, 2k PhyCo videos). The largest gains are on Coffee ($+26.67$ pp) and Sink ($+20.00$ pp), which depend on sustained contact under variable physics. PhyCo spans exactly these dynamics, and demonstrations cover them only thinly. Door and Microwave (0 pp) involve stereotyped articulated motion that demonstrations already cover, so the cycle signal neither helps nor hurts. Stove (-15.00 pp) and Drawer (-5.00 pp) are mismatched: stove control hinges on small knob rotations and fine visual cues absent from PhyCo, and drawer opening is a constrained rigid-body motion outside its coverage. **The benefit of action-free video is thus governed by the overlap between its transitions and the target task's dynamics, not by the volume of video.** Where the two overlap, ReCycle transfers that structure to the policy by requiring latent actions to reproduce the unlabeled transitions.

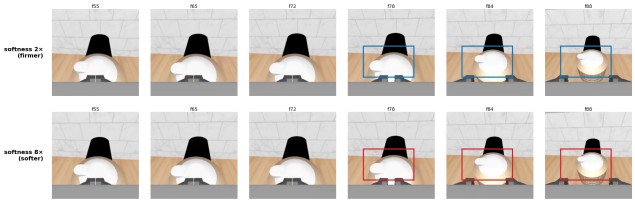


Figure 3: Compliant-button physics. Softness $2\times$ (top, firm) vs. $8\times$ (bottom, soft) on `CoffeePressButton` ep. 0. Around $f_{72} \rightarrow f_{88}$ the gripper contacts the button (boxed): at $8\times$ it indents the button deeper into the dome and keeps descending after the $2\times$ run has bottomed out. Both episodes complete the press.

Scaling unlabeled physical data. CP underperforms on coffee tasks such as `CoffeeSetupMug` and `PressButton` (Tab. 3), which we hypothesize demand greater physical awareness–e.g., `PressButton` (Fig. 3) requires reasoning about contact compliance to apply appropriate force. Tab. 4 shows adding PhyCo data improves performance with diminishing returns beyond 10k videos: success rate saturates between 10k and 50k, indicating moderate amounts of physically grounded data suffice. Stove tasks show the opposite pattern, improving monotonically through 50k–some task families benefit from substantially larger unlabeled corpora. All variants use batch size 20, 1k iterations, and 10 episodes per task.

Contact physics determines success. To isolate the effect of physical properties, we run a controlled comparison on the same policy and scene (Fig. 5, `PnPCabToCounter`, ep. 7), varying only

the global tangential-friction coefficient μ in MuJoCo. Under low friction ($\mu = 0.25$), the hot dog is grasped but slips by f_{184}; the policy continues executing its planned motion with an empty gripper and times out at 500 frames. Under high friction ($\mu = 2.0$), the same policy succeeds in 162 frames. The failure is not caused by an incorrect trajectory but by a change in contact dynamics–motivating incorporating physical awareness into the policy itself.

4.5 How to Incorporate Unlabeled Video

Unlabeled video helps on the right tasks, but how that signal is consumed matters. We compare our cycle objective with two natural alternatives, all using the same backbone, labeled corpus, PhyCo subset, and training schedule. (1) **Action pseudo-labeling** trains an inverse-dynamics model on $\mathcal{D}_{\mathrm{L}}$, predicts pseudo-actions for PhyCo videos, and trains the policy on the union with the supervised diffusion loss; unlike our cycle objective, it freezes the pseudo-actions before policy training and inherits any biases of the IDM. (2) **Forward-only video prediction** trains next-frame prediction with the action slot masked, $\left\| f_\theta([o_t, \cdot]; m_{\mathrm{fd}}) - o_{t+1} \right\|_2^2$, which can improve visual representations but does not require inferred actions to be executable. On coffee tasks (Tab. 5), our cycle objective achieves 76.7%, outperforming both alternatives (each 70.0%) and the baseline (50.0%): the gain over pseudo-labeling shows end-to-end gradient flow through the IDM and world model beats freezing inferred actions, and the gain over forward-only prediction shows the round-trip executability constraint provides action grounding beyond visual representation learning.

Table 5: Per-task success rates (%) on coffee-related RoboCasa tasks, comparing the CP baseline against alternative ways of using the PhyCo unlabeled corpus.

Task	CP	Pseudo-label	Forward-only	ReCycle (ours)
CoffeeSetupMug	30	40	50	**60**
CoffeeServeMug	50	80	70	**80**
CoffeePressButton	70	90	90	**90**
Mean	50.0	70	70	**76.7**

4.6 Standard Benchmark Evaluation with and w/o Physical Perturbation

Although ReCycle is not trained for robustness, its gains carry over to MuJoCo perturbations in friction, softness, damping, and controller gain (Tab. 6). On RoboCasa, ReCycle improves over CP on average in every perturbation family: +2.78 pp (friction), +2.64 pp (softness), +4.45 pp (damping), and +1.95 pp (gain). On LIBERO, where nominal success is near saturation, gains appear mainly under perturbation, e.g., +3.00 pp at friction 2.0 and +6.50 pp at gain 1.5×. We attribute this by-product to the cycle objective shaping the policy's dynamics representation rather than only its visual features (Sec. 3.2). Consistent with this, ReCycle's latent action linearly encodes physics parameters (friction, restitution) better than Cosmos Policy (+6.5 pp avg.; App. A.7). Qualitative rollouts (Figs. 6 and 3) show ReCycle recovering from contact failures that defeat the baseline.

4.7 Real-robot experiment

We use a Stretch 3 robot. We collected data for three tasks: (1) pick up apple and move it to black bowl, (2) pick cube and move it to the black patch on table, (3) pick up food and place it into wooden basket, with about 100 demonstrations per task. CP baseline and ReCycle are trained for $2k$ steps under matched action space and observations. For ReCycle, we use SSv2 as unlabeled data. We observe that the task success rate (averaged over 20 trials per task) improves for ReCycle as shown in Tab. 7. Some success rollouts are shown in Fig. 4. More details and rollouts are provided in the supplementary A.3. **We also provide robot videos in the supplementary material zip folder titled 'ReCycle_supplementary.zip'. Please unzip the folder and click on the 'index.html' file to play the rollouts.**

Table 6: Robustness to physical-parameter perturbations on RoboCasa and LIBERO. We evaluate the CP baseline and our method (+ReCycle) under varied friction, softness, damping, and controller-gain coefficients. Adding unlabeled videos to the baseline improves robustness across most settings, with the largest gains concentrated at extreme parameter values and on RoboCasa. Best per row within each benchmark in **bold**.

		RoboCasa			LIBERO		
Perturbation	Coeff.	CP (%)	+ReCycle (%)	Δ (pp)	CP (%)	+ReCycle (%)	Δ (pp)
Friction	0.25	75.00	**77.08**	+2.08	97.00	**98.00**	+1.00
	0.50	75.83	**78.33**	+2.50	97.20	**98.20**	+1.00
	2.00	73.75	**77.50**	+3.75	91.75	**94.75**	+3.00
	Avg.	74.86	**77.64**	**+2.78**	95.32	**96.98**	**+1.67**
Softness	2.0	74.58	**77.50**	+2.92	**97.00**	96.25	−0.75
	4.0	73.33	**78.75**	+5.42	95.75	**97.75**	+2.00
	8.0	**76.67**	76.25	−0.42	96.80	**97.50**	+0.70
	Avg.	74.86	**77.50**	**+2.64**	96.51	**97.16**	**+0.65**
Damping	0.25×	77.08	**80.42**	+3.34	**96.35**	95.75	−0.60
	1.0×	73.75	**79.17**	+5.42	97.55	**98.25**	+0.70
	4.0×	67.92	**72.50**	+4.58	**54.25**	54.00	−0.25
	Avg.	72.92	**77.36**	**+4.45**	**82.72**	82.67	−0.05
Gain	0.5×	0.83	**1.25**	+0.42	2.25	**2.50**	+0.25
	1.0×	73.75	**79.17**	+5.42	**98.50**	98.25	−0.25
	1.5×	13.75	13.75	0.00	42.75	**49.25**	+6.50
	Avg.	29.44	**31.39**	**+1.95**	47.83	**50.00**	**+2.17**

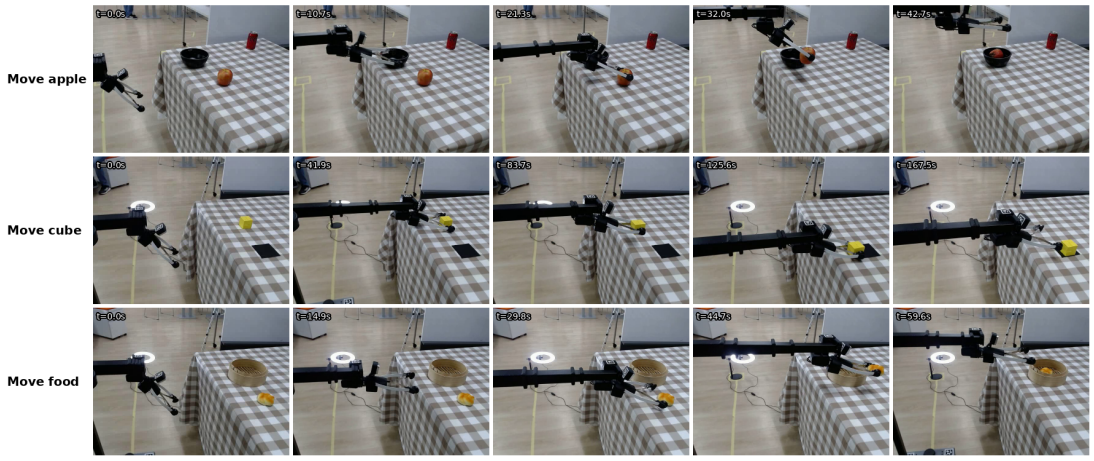


Figure 4: Real-robot Stretch 3 evaluation. Top: Move apple to the black bowl, Middle: Move cube to the black patch, Bottom: Move food to the wooden basket.

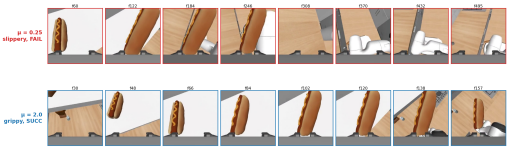


Figure 5: Object dragging at low friction (`PnPCabToCounter` ep. 7). Wrist frames from the same policy, differing only in μ. Top ($\mu = 0.25$, slippery): hot dog slips by f_{184}, policy times out at 500 frames (FAIL). Bottom ($\mu = 2.0$, grippy): places at f_{157} in 162 frames (SUCCESS).

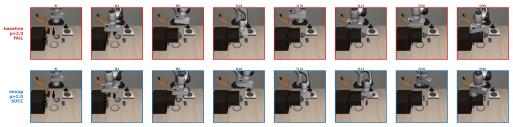


Figure 6: LIBERO-Goal: open drawer & place bowl ($\mu = 2.0$, soft $= 1.0$). At high friction the bowl sticks to the gripper/drawer lip during insertion: baseline loses contact alignment and times out (FAIL); ReCycle releases cleanly into the drawer (SUCCESS).

Table 7: Real world comparison (%)

Method	Pick apple	Pick Cube	Pick food
CP	60	50	55
ReCycle	80	65	75

Limitations. ReCycle requires a backbone that exposes both an inverse and a forward path, and it relies on a supervised warm start before the cycle term is enabled. Its benefit depends on transition overlap with the target task, and mismatched video can hurt (e.g., stove), so selecting video by relevance is left to future work.

Table 8: Loss-component ablation on RoboCasa

$\mathcal{L}_{\text{policy}}$	$\mathcal{L}_{\text{cycle}}$	Curriculum	Mixup	$\mathcal{L}_{\text{var}}$	Success rate (%)
✓					73.75
✓	✓				40.30
✓	✓	✓			62.18
✓	✓	✓	✓		78.50
✓	✓	✓	✓	✓	**79.17**

5 Conclusion

We presented ReCycle, a novel way to supervise visuomotor diffusion policies with action-free video, as an alternative to visual pretraining, forward prediction, and pseudo-labeling. Diffusion policies with forward and inverse dynamics already share a latent action across their policy, forward, and inverse roles, which makes cycle consistency a natural self-supervised objective: a latent action inferred from a transition must be sufficient to reconstruct it, and both passes are trained end-to-end in the policy's own representation. We showed that this objective works only once two problems are resolved: early gradient conflict, resolved by a supervised warm start, and cross-domain transition mismatch, resolved by slot-wise latent mixup. Batch-relative reweighting adds further stabilization. Because ReCycle only requires an inverse and a forward path, it plugs into Cosmos Policy, UVA, and Video Policy without changing inference. It improves each over its labeled-only baseline, with gains that persist under shifted friction, contact softness, and damping, which indicates that it shapes the dynamics representation rather than only visual features. Finally, the value of action-free video depends on transition overlap: it helps when the video contains dynamics relevant to the target task and can hurt when they are mismatched. ReCycle turns this overlap into policy supervision, reducing dependence on action-labeled demonstrations.

AI USE STATEMENT

In this work, we used generative AI tools for refining the writing. We have not used generative AI tools for anything else. Additionally, we used generative AI tools for polishing writing. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies training robot manipulation policies from action-labeled demonstrations and action-free video. All experiments are conducted in simulation (RoboCasa, LIBERO) and on publicly available datasets (PhyCo, LIBERO, Something-Something v2). Human video from SSv2 is used under its original license solely as an unlabeled training signal. We do not foresee specific ethical concerns beyond the general risks of deploying learned robot policies in the physical world.

REPRODUCIBILITY STATEMENT

The unified denoiser, training objectives, curriculum schedule, and slot-wise latent mixup are described in Section 3. Datasets, benchmarks, evaluation protocols, and physics-perturbation settings are described in Section 4, with additional implementation details, hyperparameters, and dataset visualizations in the Appendix. We will release code and trained checkpoints upon publication.

REFERENCES

Pulkit Agrawal, Ashvin V. Nair, Pieter Abbeel, Jitendra Malik, and Sergey Levine. Learning to poke by poking: Experiential learning of intuitive physics. In *Advances in Neural Information Processing Systems*, pp. 5074–5082, 2016.

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*, 2022.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. *arXiv preprint arXiv:2402.15391*, 2024.

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Univla: Learning to act anywhere with task-centric latent actions. *arXiv preprint arXiv:2505.06111*, 2025.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, pp. 02783649241273668, 2023.

Paul Christiano, Zain Shah, Igor Mordatch, Jonas Schneider, Trevor Blackwell, Joshua Tobin, Pieter Abbeel, and Wojciech Zaremba. Transfer from simulation to real world through learning deep inverse dynamics model. *arXiv preprint arXiv:1610.03518*, 2016.

Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Temporal cycle-consistency learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

Ashley D. Edwards, Himanshu Sahni, Yannick Schroecker, and Charles L. Isbell. Imitating latent policies from observation. In *International Conference on Machine Learning*, 2019.

Roy Featherstone. *Rigid body dynamics algorithms*. Springer, 2008.

Yao Feng, Hengkai Tan, Xinyi Mao, Guodong Liu, Shuhe Huang, Chendong Xiang, Hang Su, and Jun Zhu. Vidar: Embodied video diffusion model for generalist bimanual manipulation. *arXiv preprint arXiv:2507.12898*, 2025.

Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE international conference on computer vision*, pp. 5842–5850, 2017.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Nicklas Hansen, Hao Su, and Xiaolong Wang. Td-mpc2: Scalable, robust world models for continuous control. *arXiv preprint arXiv:2310.16828*, 2023.

Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, Jianke Zhang, Koushil Sreenath, Chaochao Lu, and Jianyu Chen. Video prediction policy: A generalist robot policy with predictive visual representations. *arXiv preprint arXiv:2412.14803*, 2024.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: a Vision-Language-Action Model with Open-World Generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Arnav Kumar Jain, Vibhakar Mohta, Subin Kim, Atiksh Bhardwaj, Juntao Ren, Yunhai Feng, Sanjiban Choudhury, and Gokul Swamy. A smooth sea never made a skilled sailor: Robust imitation via learning to search. *arXiv preprint arXiv:2506.05294*, 2025.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. *Advances in neural information processing systems*, 32, 2019.

Wisama Khalil and Etienne Dombre. *Modeling, identification and control of robots*. Butterworth-Heinemann, 2002.

Hanjung Kim, Jaehyun Kang, Hyolim Kang, Meedeum Cho, Seon Joo Kim, and Youngwoon Lee. UniSkill: Imitating human videos via cross-embodiment skill representations. In *Conference on Robot Learning*, pp. 4269–4294, 2025.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, et al. Cosmos policy: Fine-tuning video models for visuomotor control and planning. *arXiv preprint arXiv:2601.16163*, 2026.

Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model. *arXiv preprint arXiv:2503.00200*, 2025a.

Wei Li, Renshan Zhang, Rui Shao, Jie He, and Liqiang Nie. Cogvla: Cognition-aligned vision-language-action model via instruction-driven routing & sparsification. *arXiv preprint arXiv:2508.21046*, 2025b.

Junbang Liang, Pavel Tokmakov, Ruoshi Liu, Sruthi Sudhakar, Paarth Shah, Rares Ambrus, and Carl Vondrick. Video generators are robot policies. *arXiv preprint arXiv:2508.00795*, 2025.

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, 2024.

Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. Vip: Towards universal visual reward and representation via value-implicit pre-training. In *International Conference on Learning Representations*, 2023.

Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations. *arXiv preprint arXiv:2310.17596*, 2023.

Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. In *CoRL*, 2022.

Sriram Narayanan, Ziyu Jiang, Srinivasa Narasimhan, and Manmohan Chandraker. Phyco: Learning controllable physical priors for generative motion, 2026. URL https://arxiv.org/abs/2604.28169. CVPR 2026.

Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots. *arXiv preprint arXiv:2406.02523*, 2024.

Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023.

Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International Conference on Machine Learning*, pp. 2778–2787, 2017.

Ilija Radosavovic, Tete Xiao, Stephen James, Pieter Abbeel, Jitendra Malik, and Trevor Darrell. Real-world robot learning with masked visual pre-training. In *Conference on Robot Learning*, 2023.

Dominik Schmidt and Minqi Jiang. Learning to act without actions. In *International Conference on Learning Representations*, 2024.

Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, and Sergey Levine. Time-contrastive networks: Self-supervised learning from video. In *IEEE International Conference on Robotics and Automation*, 2018.

Austin Stone, Ted Xiao, Yao Lu, Keerthana Gopalakrishnan, Kuang-Huei Lee, Quan Vuong, Paul Wohlhart, Sean Kirmani, Brianna Zitkovich, Fei Xia, et al. Open-world object manipulation using pre-trained vision-language models. *arXiv preprint arXiv:2303.00905*, 2023.

Richard S. Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *SIGART Bull.*, 2(4):160–163, July 1991. ISSN 0163-5719. doi: 10.1145/122344.122377. URL https://doi.org/10.1145/122344.122377.

Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning*, pp. 6438–6447, 2019.

Xiaolong Wang, Allan Jabri, and Alexei A. Efros. Learning correspondence from the cycle-consistency of time. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

Yiqi Wang, Mrinal Verghese, and Jeff Schneider. Latent policy steering with embodiment-agnostic pretrained world models. *arXiv preprint arXiv:2507.13340*, 2025.

Tete Xiao, Ilija Radosavovic, Trevor Darrell, and Jitendra Malik. Masked visual pre-training for motor control. In *arXiv preprint arXiv:2203.06173*, 2022.

Liudi Yang, Yang Bai, George Eskandar, Fengyi Shen, Mohammad Altillawi, Dong Chen, Soumajit Majumder, Ziyuan Liu, Gitta Kutyniok, and Abhinav Valada. Roboenvision: A long-horizon video generation model for multi-task robot manipulation. *arXiv preprint arXiv:2506.22007*, 2025.

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Sejune Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, Lars Liden, Kimin Lee, Jianfeng Gao, Luke Zettlemoyer, Dieter Fox, and Minjoon Seo. Latent action pretraining from videos. In *International Conference on Learning Representations*, 2025.

Yanjie Ze, Gu Yan, Yueh-Hua Wu, Annabella Macaluso, Yuying Ge, Jianglong Ye, Nicklas Hansen, Li Erran Li, and Xiaolong Wang. 3d diffusion policy. *arXiv preprint arXiv:2403.03954*, 2024.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

Ruijie Zheng, Jing Wang, Scott Reed, Johan Bjorck, Yu Fang, Fengyuan Hu, Joel Jang, Kaushil Kundalia, Zongyu Lin, Loic Magne, et al. Flare: Robot learning with implicit world modeling. *arXiv preprint arXiv:2505.15659*, 2025.

Ruijie Zheng, Dantong Niu, Yuqi Xie, Jing Wang, Mengda Xu, Yunfan Jiang, Fernando Castañeda, Fengyuan Hu, You Liang Tan, Letian Fu, Trevor Darrell, Furong Huang, Yuke Zhu, Danfei Xu, and Linxi Fan. EgoScale: Scaling dexterous manipulation with diverse egocentric human data. *arXiv preprint arXiv:2602.16710*, 2026.

Zhide Zhong, Haodong Yan, Junfeng Li, Xiangchen Liu, Xin Gong, Wenxuan Song, Jiayi Chen, and Haoang Li. Flowvla: Thinking in motion with a visual chain of thought. *arXiv preprint arXiv:2508.18269*, 2025.

Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets. *arXiv preprint arXiv:2504.02792*, 2025.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision*, 2017.

A Appendix

In this appendix, we provide the following details.

- More related works
- Training details.
- Real robot training details.
- Analysis.
- Quantitative and qualitative results.

We also provide robot videos in the supplementary material zip folder titled 'ReCycle_supplementary.zip'. Please unzip the folder and click on the 'index.html' file to play the rollouts.

A.1 More Related Work

Action-free video pretraining for robot policies. A long line of work has sought to leverage abundant action-free video for downstream robot learning. Visual-representation pretraining methods such as R3M (Nair et al., 2022), VIP (Ma et al., 2023), MVP (Xiao et al., 2022; Radosavovic et al., 2023), and Time-Contrastive Networks (Sermanet et al., 2018) train frozen encoders from human videos and attach lightweight policy heads. A second line learns latent action spaces: VPT (Baker et al., 2022) pseudo-labels videos using an inverse-dynamics model, while LAPO (Schmidt & Jiang, 2024) and GENIE (Bruce et al., 2024) learn latent actions in fully unsupervised fashion. More recent efforts including UniSkill (Kim et al., 2025) and EgoScale (Zheng et al., 2026) pretrain on egocentric human videos at scale. We differ from these threads in two ways: prior approaches separate pretraining from policy learning into distinct stages with frozen encoders or auxiliary modules, while we jointly train the policy and self-supervised objective end-to-end through a unified diffusion backbone; and pseudo-labeling pipelines commit to inferred actions at IDM-training time and inherit any biases of the predictor, whereas our cycle objective continues to update both the IDM and world-model heads through gradient flow on shared parameters.

Cycle consistency and inverse–forward dynamics. Cycle consistency has been widely used as a self-supervised signal in image translation (Zhu et al., 2017), video correspondence learning (Dwibedi et al., 2019; Wang et al., 2019), and motion estimation. Several works in robotics apply related ideas: Christiano et al. (Christiano et al., 2016) use inverse–forward consistency for sim-to-real transfer, Edwards et al. (Edwards et al., 2019) learn latent policies from observation by enforcing dynamics consistency, and Latent Policy Steering (Wang et al., 2025) uses optical flow as an embodiment-agnostic action representation that implicitly enforces forward consistency. Our contribution is to apply the cycle-consistency principle within a unified diffusion denoiser whose policy, world-model, and inverse-dynamics heads share a single set of parameters: cycle consistency in this shared latent space provides a natural self-supervised objective for action-free video without requiring separate IDM or WM modules, and gradients from the cycle directly shape the policy backbone.

Diffusion policies and unified video–action models. Diffusion policies (**?**Ze et al., 2024) have emerged as a strong paradigm for visuomotor control. Recent work increasingly adapts pretrained video models for manipulation: some methods fine-tune video models and attach separate action modules (Liang et al., 2025; Hu et al., 2024; Yang et al., 2025; Feng et al., 2025; Zhong et al., 2025), while others train unified video-action models from scratch (Li et al., 2025a; Zhu et al., 2025). Vision-language-action models such as RT-2 (Brohan et al., 2023), OpenVLA (Kim et al., 2024), π_0 (Black et al., 2024), $\pi_{0.5}$ (Intelligence et al., 2025), UniVLA (Bu et al., 2025), and CogVLA (Li et al., 2025b) fine-tune image-text backbones on robot data but lack strong temporal priors. Most directly relevant, Cosmos Policy (Kim et al., 2026) fine-tunes the Cosmos-Predict2 video foundation model into a unified policy, world model, and value function via latent frame injection. Other unified-dynamics approaches (Sutton, 1991; Janner et al., 2019; Hafner et al., 2023; Hansen et al., 2023; Zheng et al., 2025; Jain et al., 2025) maintain distinct policy, world-model, and value modules trained from scratch. We adopt Cosmos Policy's unified parameterization and exploit the shared latent space across heads to introduce a cycle-consistency objective on action-free video, an axis along which prior work has not extended this architecture.

A.2 Training details.

A.2.1 Datasets.

RoboCasa. RoboCasa (Nasiriany et al., 2024) consists of 24 static kitchen manipulation tasks using a Franka Emika Panda arm. Although the benchmark includes 50 human demonstrations per task and additional MimicGen-generated demonstrations (Mandlekar et al., 2023), we train using only the 50 human demonstrations to evaluate data efficiency. Each task is evaluated over 50 trials across five scenes (10 trials per scene), with unseen object instances and some test scenes containing visual styles not present during training.

LIBERO. LIBERO (Liu et al., 2024) is a Franka manipulation benchmark with four suites—LIBERO-Spatial, LIBERO-Object, LIBERO-Goal, and LIBERO-Long—evaluating generalization to new spatial layouts, objects, language goals, and long-horizon tasks. Each suite contains 10 tasks with 50 demonstrations per task. We follow the evaluation setup of Cosmos Policy (Kim et al., 2026).

PhyCo Dataset: We instantiate the action-free corpus $\mathcal{D}_{\mathrm{U}}$ with the PhyCo dataset (Narayanan et al., 2026), a large-scale synthetic corpus of over 100K photorealistic clips generated with Kubric, PyBullet, and Blender (Fig. 7). Each clip systematically varies one or more of four physical attributes—friction, restitution, deformation, and applied force—across six canonical interaction regimes spanning rigid sliding, elastic rebound, gravity-driven soft drops, deformable impacts, and multi-body collisions. Object color, surface material, camera placement, and HDRI illumination are independently randomized across 50 environments, so visual appearance is decoupled from the underlying dynamics. This factorization is precisely what $\mathcal{L}_{\text{vid}}$ requires: the same physical attribute is observed under widely varying visual conditions, forcing the inverse-dynamics and world-model heads to explain transitions through controllable latent actions rather than appearance correlations, which is the property the cycle loss is designed to reward. Unless noted otherwise, we draw a random subset of 2K clips as $\mathcal{D}_{\mathrm{U}}$ for all experiments.

Something-Something v2 (SSv2) (Goyal et al., 2017) is a large-scale crowdsourced video dataset of $\sim$220K clips depicting humans performing 174 everyday hand-object interaction templates, such

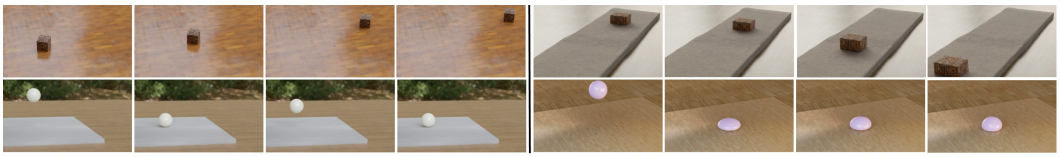
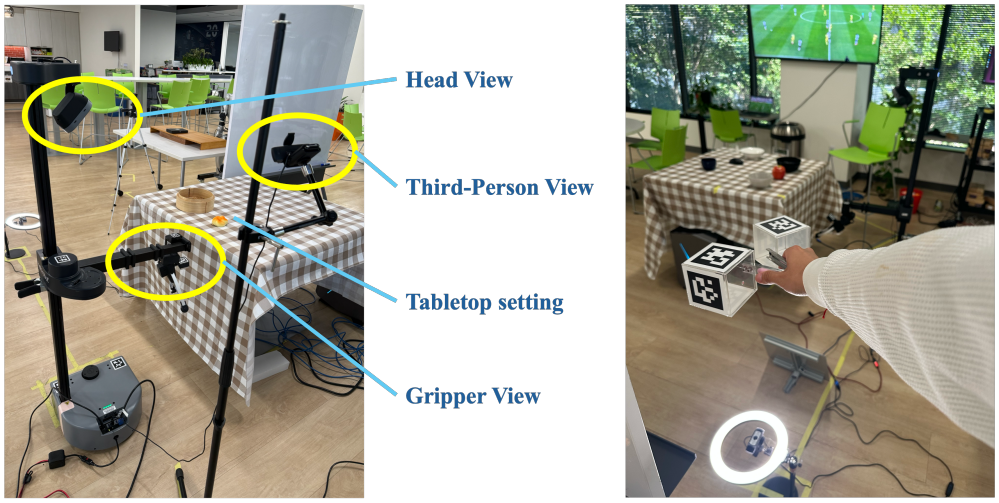


Figure 7: Example visualizations from the PhyCo dataset (Narayanan et al., 2026), illustrating controlled variations in key physical attributes, including friction, restitution, applied force, and deformation.



(a) Real World Testing Environment

(b) Real World Data Collection Environment

Figure 8: **Left:** Stretch 3 with the three camera views: head view, third-person view on a fixed mount, and egocentric gripper view. **Right:** Teleoperation with the ArUco-marker cube tracked by a ring-lit webcam.

as "pushing X from left to right," "putting X behind Y," "moving X closer to Y," or "pretending to pick X up." Each template is instantiated by many crowd workers using arbitrary household objects, viewpoints, and lighting, producing rich variation in scene composition, hand morphology, object identity, and motion style while preserving the underlying interaction semantics. Crucially for action-free policy learning, SSv2 was designed around *action concepts*—the templates target specific manipulation primitives rather than passive scenes—so the corpus is densely populated with goal-directed contact events even though no robot embodiment, joint trajectories, or controlled physical parameters are recorded. We use SSv2 as our most out-of-distribution unlabeled corpus: it shares neither robot embodiment with the labeled demonstrations nor controlled physics with PhyCo, yet its sheer diversity of human manipulation provides a strong test of whether the cycle objective extracts useful structure from any sufficiently rich source of physical interaction.

A.3 Real-World Robot Experiments

We build a real-world environment to evaluate multiple tabletop manipulation in a dynamic and everyday setting (Fig. 8). It uses the Hello Robot Stretch 3 platform.

Scene. The workspace is a table in an open office lounge, covered with a brown-and-white gingham tablecloth. The patterned cloth gives a textured background that is not trivially segmentable. On the table are household objects in varied colors, shapes and materials: a white ceramic bowl, a black bowl, a dark mug, and an apple. We mark the robot's base footprint and the table position on the floor with tape, so the robot–table pose stays the same across sessions and episodes can be reset repeatably.

Camera configuration. We record three synchronized RGB streams from different viewpoints:

- **Head view:** an Intel Realsense D435i depth camera on the robot's mast/head. It looks down at the table and shows the whole workspace, including the arm and all objects. We take the RGB observation from this camera.
- **Third-person view:** a static external RGB Webcam camera mounted on a rigid stand ("sturdy" view) at the table edge, facing the robot. We keep it fixed during both data collection and evaluation so the viewpoint does not drift.
- **Gripper view (egocentric):** the wrist-mounted Intel RealSense D405. It gives a close-range view of the gripper approaching and contacting objects.

Teleoperation. We collect demonstrations with Stretch's dexterous teleoperation interface. An upward-facing webcam under a ring light tracks a handheld cube covered in ArUco markers. The operator moves the cube, and the tracked 6-DoF pose is mapped to the robot's base, lift, arm extension and wrist joints.

Here are three tasks that we collect for evaluation:

- **"Move apple:"** pick up the red apple on the table and place it into the black bowl.
- **"Move cube:"** pick up the cube and place it on the black patch on the table.
- **"Move food:"** pick up the orangish food item and place it to the wooden basket.

Environmental variation. The environment is deliberately left uncontrolled. Lighting comes mainly from natural light through large windows and changes over the day. The background contains distractors: chairs and furniture, people walking by, and a television playing video. We keep these conditions during evaluation to test how the learned latent actions hold up under real-world appearance changes.

Training cosmos policy and Recycle on robot benchmark. We collected around 100 rollouts for each tasks - move apple, move cube, move food. Then using the dataset we trained baseline CP and ReCycle on top of that. The details are provided below.

Training. All policies are Cosmos Policy models Kim et al. (2026) initialized from the generic Cosmos-Predict2-2B Video2World checkpoint (480p, 16 fps). They use the three-camera latent layout of the ALOHA configuration: 11 latent frames holding the current and future proprioception, the three camera views, a 50-step action chunk, and a value estimate. Images are resized to 224×224 with image augmentation, and T5 embeddings of the task instruction are used for conditioning. Each model is trained for 10,000 iterations with an effective batch size of 128 (Move Apple and Move Cube: 4 H100 GPUs $\times$ 16 samples $\times$ 2 gradient-accumulation steps; Move Food: 2 GPUs $\times$ 16 $\times$ 4). We use a learning rate of 10^{-4} with the default Cosmos Policy schedule. For each task we train two models with identical data and hyperparameters: the *baseline*, trained with the standard Cosmos Policy objective, and *ReCycle*, which adds the cycle-consistency loss with $\lambda_{\text{cycle}}=1$. On the real robot, the unlabeled video for ReCycle comes from the task's own demonstrations with their actions stripped. One eighth of each batch is drawn from this action-free stream. The inverse-dynamics pass infers actions for these clips, and the forward-dynamics pass must reconstruct the observed future frames from those inferred actions.

Inference. The policy runs on a remote GPU server and communicates with the robot over a websocket (msgpack) interface. At each query, the robot sends the three camera images, the 8-D joint state, and the instruction. The server returns a 50-step chunk of absolute joint targets, sampled with 5 action-denoising steps. The robot executes the first 15 targets open-loop at 15 Hz (1 s) and then re-queries. For safety, each commanded target is clamped to within 0.03 of the current joint position per control tick. Execution also stops if the server returns non-finite actions.

Evaluation. Each policy is evaluated for 20 trials per task, with object and robot initial poses varied across positions. The same initial configurations are used for the baseline and ReCycle. A trial is successful if the object (apple, cube, food) ends in the target (bowl, pad, basket) within 3000 steps. We report success rates in Table 7. Some of the rollout comparison are provided in Fig. 15, Fig. 16, Fig. 17, Fig. 18, Fig. 19, Fig. 20, Fig. 21, Fig. 21, Fig. 22, Fig. 23, Fig. 24, Fig. 25, Fig. 26. More rollout videos are provided in the supplementary video (click on the index.html file).

A.4 HYPERPARAMETER AND DESIGN ABLATIONS.

We further verify that each design choice in our method is load-bearing rather than incidental.

Curriculum warmup length (Tab. 9): we vary T_{warm} between 0 and 44k iterations within a fixed 45k-iteration budget. Performance rises from 33.21% at $T_{\text{warm}} = 0$ (cycle gradients destabilize the backbone before action features form) to 79.17% at $T_{\text{warm}} = 40$k, then drops slightly at 44k where insufficient cycle iterations remain to shape the dynamics representation—confirming that curriculum scheduling is not just a convenience but the gating condition for the auxiliary objective.

Table 9: Sensitivity to curriculum warmup length on RoboCasa (24 tasks, 10 episodes per task). Total training is held fixed at 45k iterations; T_{warm} controls how long the policy trains on $\mathcal{L}_{\text{policy}}$ alone before $\mathcal{L}_{\text{cycle}}$ activates. Too short ($\leq$20k) lets cycle gradients destabilize the backbone before action features form; too long ($\geq$44k) leaves insufficient cycle iterations to shape the dynamics representation.

T_{warm}	Success rate (%)
0 (no curriculum)	33.21
10k	50.73
20k	60.12
30k	68.38
40k (ours)	**79.17**
44k (cycle for 1k iter)	78.25

Mixup coefficient α (Tab. 10): we sweep $\alpha \in \{0.05, 0.2, 1.0\}$ and find a flat plateau around $\alpha = 0.2$ (79.17%), with degradation at both extremes—small α approximates an unmixed pair (under-bridging the domain gap), large α pushes toward $\lambda \approx 0.5$ (over-blending and corrupting the cycle target)—indicating the method is robust to small mixup-strength perturbations but does require some blending.

Table 10: Effect of mixup coefficient α on RoboCasa, with $\lambda \sim \text{Beta}(\alpha, \alpha)$. Larger α pulls the mixing distribution toward $\lambda = 0.5$ (heavy blending); $\alpha \to 0$ makes the distribution bimodal at $\{0, 1\}$ (mostly unmixed). $\alpha = \infty$ corresponds to the no-mixup configuration in Tab. 8.

α	Success rate (%)
0.05	50.43
0.2 (ours)	**79.17**
1.0	64.50

Loss-weight sensitivity (Tab. 11): sweeping λ_{cycle} over $\{0.0, 0.5, 1.0\}$ and λ_{var} over $\{0.0, 0.5, 1.0\}$ shows the method is robust within a factor of two of the chosen values ($\lambda_{\text{cycle}} = 1.0$, $\lambda_{\text{var}} = 0.5$), with $<$1 pp variation across the sweep—ruling out the concern that the result depends on precise loss-weight tuning.

Table 11: Loss-weight sensitivity on RoboCasa.

λ_{cycle}	λ_{var}	SR (%)
1.0	0.0	40.30
0.0	1.0	35.78
0.5	1.0	**64.72**
1.0	0.5	79.17

A.5 DISCUSSIONS

Loss-component ablation and policy/cycle trade-off. Because the policy, world-model, and inverse-dynamics heads share a single set of parameters, $\mathcal{L}_{\text{policy}}$ and $\mathcal{L}_{\text{cycle}}$ update the same representation and can pull it in conflicting directions: the former optimizes for features that map labeled observations to ground-truth actions, the latter for features that explain transitions on the

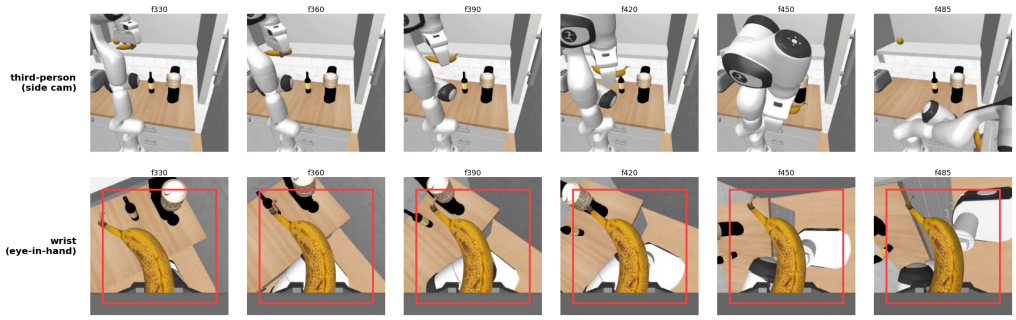


Figure 9: Object toggling under high friction (RoboCasa task `PnPCabToCounter`, ep. 8). Once the gripper releases the banana, $\mu = 2.0$ ($2\times$ nominal) prevents a clean settle. The banana keeps catching on the counter while the still-moving gripper drags it through stick-slip cycles, so its yaw snaps frame-to-frame in the wrist view ($\approx 30° \to 60° \to 100° \to 130°$ between f_{330} and f_{485}). The late-rollout motion signal in the wrist panel is ≈ 7.5—about $5\times$ higher than any other clip in the entire sweep—confirming the toggle artifact is friction-driven, not a policy idiosyncrasy.

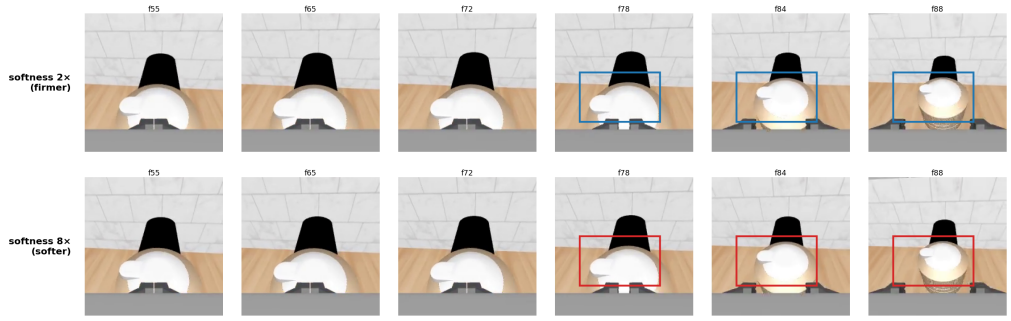


Figure 10: Compliant-button physics, softness $2\times$ vs. $8\times$ (RoboCasa task `CoffeePressButton`, ep. 0). MuJoCo's softness is the inverse of effective contact stiffness, so increasing it lets the button-spring deflect further before reacting. Around $f_{72} \to f_{88}$ the gripper is in contact with the button (boxed). At softness $= 8\times$ (bottom row) the gripper visibly indents the button deeper into the dome and continues to descend after the firmer $2\times$ run has already bottomed out—by f_{88} the button cap is clearly squashed against the machine body in the soft case but still upright in the firm case. Both episodes complete the press, but the contact event under high softness lasts longer and the spring keeps relaxing through the final frames.

out-of-domain PhyCo distribution. Tab. 8 makes this trade-off quantitatively explicit. Activating $\mathcal{L}_{\text{cycle}}$ from iteration zero collapses success from 73.75% to 40.30%, because cycle gradients destabilize the backbone before reliable action features form. The warm start recovers 62.18%, slot-wise mixup raises it to 78.50% by narrowing the visual domain gap, and variance reweighting yields the final 79.17%. Each component therefore addresses a distinct failure mode—temporal scheduling, distributional alignment, and gradient stabilization–and we conclude that the central challenge in using action-free video for diffusion-policy training is not designing a useful auxiliary objective but *scheduling* it so it complements rather than competes with the supervised signal.

A.6 Visualizations

We provide more visualizations in Fig. 9, Fig. 10, Fig. 11, Fig. 12, Fig. 13, Fig. 14.

A.7 Interpretability: Probing the latent action space for physical structure.

A cycle-consistent video objective should push the inverse-dynamics model (IDM) to encode the physical state of the world rather than surface visual appearance. We test this directly by asking whether the IDM's own latent action $\hat{a}_t$ is linearly predictive of the underlying physics parameter that produced the transition, while controlling for visual confounds. To isolate physics

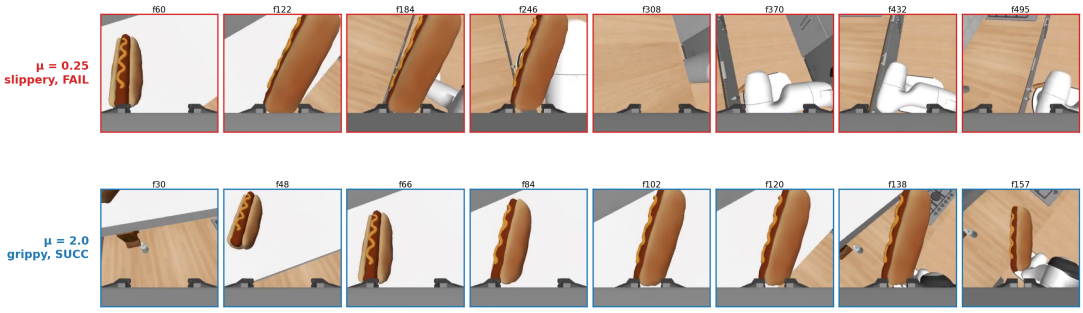


Figure 11: Object dragging is more visible at low friction (hot-dog placement, `PnPCabToCounter` ep. 7). Wrist (eye-in-hand) frames from the same scene and policy, differing only in μ. Top ($\mu = 0.25$, slippery): the hot dog is in the gripper at f_{60} and f_{122} but has slipped out by f_{184}; the gripper view is empty for every later frame, and the policy uses the full 500-frame budget without ever placing cleanly (FAIL). Bottom ($\mu = 2.0$, grippy): the hot dog stays locked from f_{30} through f_{138} and is set down at f_{157}, completing the place in 162 frames (SUCCESS). Because μ in MuJoCo applies to all tangential contacts, lowering it simultaneously degrades gripper-object retention and surface stability—the slippery rollout is over $3\times$ longer and visibly fails because the object drags out of the gripper rather than following the planned trajectory.

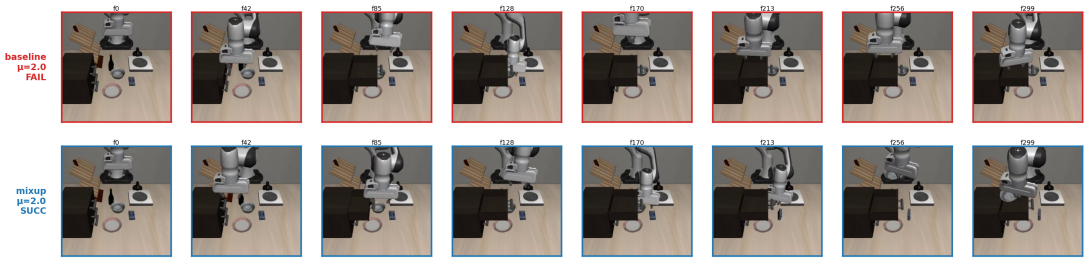


Figure 12: LIBERO-Goal task: open drawer & place bowl ($\mu = 2.0$, soft $= 1.0$). At high tangential friction the bowl sticks to the gripper / drawer lip during the insertion phase; the baseline loses contact alignment and times out (FAIL), while ReCycle completes the same conditioned trajectory with a clean release into the drawer (SUCCESS).

from appearance, we restrict each probe to a single procedurally generated Kubric scene template–holding camera, lighting, object identity, and viewpoint essentially constant–so that only one physics parameter varies across clips. We use three templates exercising different phenomena: surface friction (`friction_slide_flat_v2`), drop-impact restitution (`ball_drop_v2`), and oblique-collision restitution (`ball_wall_collision`), with each clip binned into low/medium/high tertiles of the target parameter (chance accuracy $\approx$0.33). For every clip we run the cycle pass on three independently sampled $(t, t{+}\Delta t)$ frame pairs and average the IDM latent across the three passes, denoising the diffusion-sigma stochasticity in $\hat{a}_t$ and yielding one stable latent vector per scene. We then evaluate four representations along the 28-block DiT–the IDM output x_0, blocks $14/21/27$, and the input to the final layer (`dit_final`)–using L_2-regularized logistic regression with stratified 5-fold cross-validation. Linear probing is the standard sensitivity test for self-supervised representations and, on these 256-d to 2048-d latents, is markedly more discriminative than nearest-neighbour accuracy, which is dominated by a handful of noisy coordinates.

On the most informative layer (`dit_final`, d=2048), ReCycle recovers the physics label substantially better than the Cosmos Policy baseline on every probe (Tab. 12), with an average absolute gain of $+6.5$ pp ($+14.0\%$ relative). The largest improvement is on `ball_drop_v2` ($+10.2$ pp, $+25.1\%$ relative), where the underlying physics—impact restitution–is most diagnostic of the dynamics the cycle loss explicitly supervises. The pattern across the three probes mirrors the ranking we observe at evaluation: gains concentrate where the unlabeled corpus's physical variations align with the contact regime being probed.

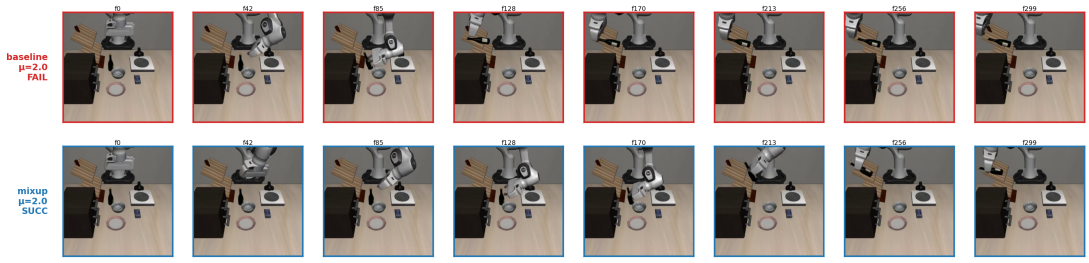


Figure 13: LIBERO-Goal task: wine bottle on rack ($\mu = 2.0$, soft $= 1.0$). High μ makes the bottle harder to re-seat on the narrow rack without rocking off; the baseline fails to stabilize the final placement, while ReCycle succeeds on the identical episode index.

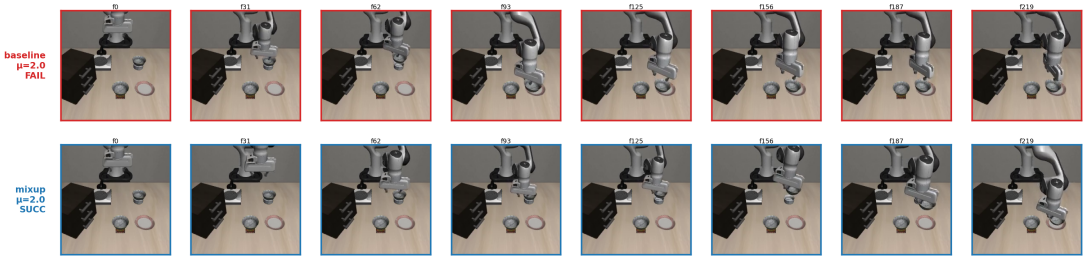


Figure 14: LIBERO-Spatial task: black bowl on ramekin ($\mu = 2.0$, soft $= 1.0$). Elevated friction changes how the bowl slides on the ramekin during the final alignment. The baseline episode fails; ReCycle completes the same indexed rollout.

A.8 Societal Impact

This is a method to train robot policy, which don't have any negative societal impact. Infact, it could aid the society if robots can be used for help.

A.9 Computation

All experiments are run in either H100 or A6000 machines with 80GB and 49GB GPU correspondingly.

A.10 Plug-and-Play ReCycle.

In this section, we describe how ReCycle is plugged into CP, UVA and VP baselines.

ReCycle on Cosmos Policy Kim et al. (2026) Cosmos Policy (CP) (Kim et al., 2026) fine-tunes the Cosmos-Predict2 video diffusion transformer into a unified policy, world model, and value function through latent frame injection. Actions, proprioception, and values are encoded as additional latent frames and packed with the image latents into one sequence $z = [s, a, s', V]$ (Eq. 2). A conditioning mask selects which frames are given and which are denoised: the policy mask m_{pol} generates a from s, and the world-model mask m_{fd} generates s' from (s, a). The forward path therefore already exists. We add the inverse path with a single new mask, m_{id}, which conditions on the current and next image latents and denoises the action frame. This requires no new parameters, because all three roles share the same denoiser f_θ. For an action-free clip, we encode the frame pair (o_t, o_{t+1}) with the Cosmos VAE and apply slot-wise mixup to the image frames only. We then run the two-pass cycle. The inverse pass under m_{id} yields the action estimate $\hat{a}_t$, which is injected into the action frame of a forward pass under m_{fd}. The predicted next-frame latent is scored against $\tilde{o}_{t+1}$ with the reconstruction loss $\mathcal{L}_{\text{cycle}}$. Gradients flow through both passes, so the inverse role learns to emit actions the world model can use, and the world model learns to use them. We retain CP's training recipe: the EDM denoising objective, its hybrid log-normal–uniform noise schedule, and modality normalization to $[-1, 1]$. Training runs for 45k iterations, of which $T_{\text{warm}} = 40$k use $\mathcal{L}_{\text{policy}}$ only. For the final 5k iterations, $\mathcal{L}_{\text{var}}$ and $\mathcal{L}_{\text{cycle}}$ are also enabled ($\lambda_{\text{var}} = 0.5$, $\lambda_{\text{cycle}} = 1$, $\alpha = 0.2$), with N_L

Table 12: Linear-probe accuracy on the IDM latent action $\hat{a}_t$ (`dit_final`, d=2048), within-template probes on Kubric. Higher is better; chance is the majority-class baseline. CP baseline: `rc24_cos_50k` at iter 45k. ReCycle (ours): `cosmos_rl_v4ws40k_mixup_grpo_5k` at iter 45k (λ_{cycle}=1.0, λ_{var}=0.05, mixup α=0.2).

Probe	Physics axis	n	Chance	CP	ReCycle	Δ	Rel.
`friction_slide_flat_v2`	platform friction	300	0.333	0.570	0.647	+0.077	+13.5%
`ball_drop_v2`	ball restitution	157	0.338	0.407	0.509	+0.102	+25.1%
`ball_wall_collision`	wall restitution	200	0.335	0.440	0.455	+0.015	+3.4%
Average			0.335	**0.472**	**0.537**	**+0.065**	**+14.0%**

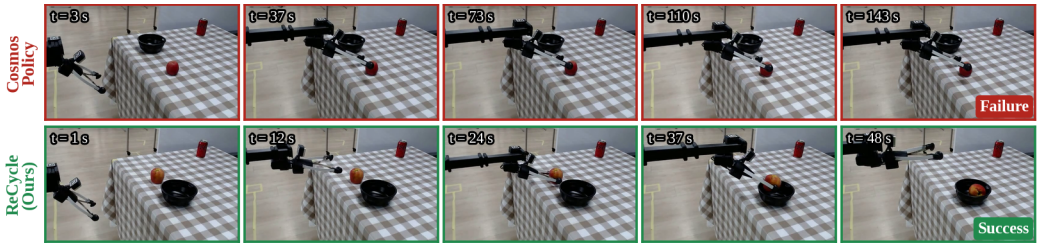


Figure 15: Success rollout for move apple task

labeled samples and N_U unlabeled clips per GPU per step. Inference is unchanged: the policy is sampled under m_{pol} with 5 denoising steps, and m_{id} is used only during training.

ReCycle on UVA Li et al. (2025a) The Unified Video Action model (UVA) Li et al. (2025a) is a masked autoregressive transformer (MAR-B, 262M parameters) that operates on VAE tokens of four past and four future frames and is trained with two diffusion heads: one that denoises the masked future-frame tokens and one that denoises a chunk of K=16 actions. A single network serves as policy, forward model, or inverse model depending on which inputs are masked: in *inverse* mode the future frames are visible and the actions are predicted, and in *dynamics* mode the past frames and the actions are visible and the future tokens are predicted. Because both directions already exist, the ReCycle loss is a direct composition of the two modes. Given an action-free clip v with past frames v_{p} and future frames v_{f} (VAE tokens $z_{\text{p}}, z_{\text{f}}$), we run the inverse mode on z_{f} and draw an action chunk from the action head with a short, differentiable S=8-step ancestral sampler, $\hat{a} = \text{IDM}_\theta(z_{\text{f}})$, keeping the sampling steps in the computation graph. We then run the dynamics mode with z_{p}, a random token mask m, and $\hat{a}$ injected through the model's action-projection layer, and apply UVA's own masked video diffusion loss to the future tokens:

$$\mathcal{L}^{\text{UVA}}_{\text{cycle}} = \mathbb{E}_{t,\epsilon,m}\Big[\, m \odot \big\|\epsilon - \epsilon_\theta\big(z_{\text{f},t};\, z_{\text{p}},\, \text{IDM}_\theta(z_{\text{f}}),\, t\big)\big\|_2^2\Big], \qquad \mathcal{L} = \mathcal{L}_{\text{UVA}} + \lambda_{\text{cycle}}\,\mathcal{L}^{\text{UVA}}_{\text{cycle}}, \quad (13)$$

where $\mathcal{L}_{\text{UVA}}$ is the original objective on labeled demonstrations (a task mode sampled uniformly per batch) and λ_{cycle}=1. Gradients flow through both passes, so the inverse head is trained to emit actions that let the dynamics head reconstruct the observed future, and the dynamics head is trained to use them. The empty-string CLIP embedding replaces the instruction for unlabeled clips. Each optimizer step uses 32 labeled samples and 4 SSv2 clips per GPU (8 frames, temporal stride 2, resized to 256^2). Inference is unchanged: the policy mode never reads the action input.

ReCycle on Video Policy Liang et al. (2025) Video Policy Liang et al. (2025) fine-tunes a Stable Video Diffusion U-Net to predict a 25-frame video (one conditioning frame followed by eight frames from each of three cameras) and decodes a chunk of K=32 actions from the U-Net's intermediate features with a separate pose branch: a 3D-convolutional vision encoder followed by a 1D diffusion U-Net over the action sequence, trained jointly in the EDM framework with the video and action noise levels harmonized. The inverse direction is therefore built in (features of a noised clip $\rightarrow$ actions), but the video model is not conditioned on actions, so there is no forward direction to close the cycle. We add one with a zero-initialized MLP g_ϕ that maps the action chunk into the U-Net's time-embedding space, $e \leftarrow e + g_\phi(a)$, together with a learned null token $g_\varnothing$; on labeled data the ground-truth chunk

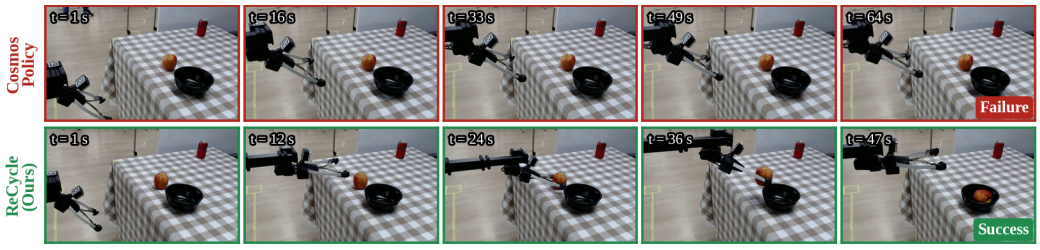


Figure 16: Success rollout for move apple task

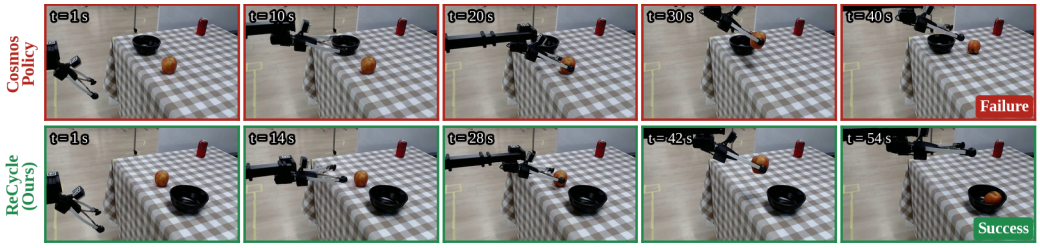


Figure 17: Success rollout for move apple task

is supplied with 50% dropout to the null token, and at inference the null token is always used, so the ReCycle model samples exactly like the baseline. For an unlabeled clip with first-stage latents x, the cycle uses two denoiser passes. First, an inverse pass at noise level $\sigma_1 \sim p(\sigma)$ with the pose input set to pure noise yields the one-shot denoised action estimate $\hat{a} = \mathrm{clip}\big(D_\theta^{\mathrm{pose}}(x+\sigma_1\epsilon,\ \sigma_1\epsilon;\ \sigma_1),\ -1, 1\big)$. Second, a forward pass at an independent level σ_2 conditions the video branch on $\hat{a}$ and is scored with the standard EDM-weighted denoising loss:

$$\mathcal{L}_{\mathrm{cycle}}^{\mathrm{VP}} = \mathbb{E}_{\sigma_1,\sigma_2,\epsilon,\epsilon'}\Big[\, w(\sigma_2)\, \big\| D_\theta^{\mathrm{video}}\big(x + \sigma_2\epsilon';\ \sigma_2,\ g_\phi(\hat{a})\big) - x \big\|_2^2 \Big], \qquad \mathcal{L} = \mathcal{L}_{\mathrm{VP}} + \lambda_{\mathrm{cycle}}\, \mathcal{L}_{\mathrm{cycle}}^{\mathrm{VP}}, \tag{14}$$

with $\lambda_{\mathrm{vid}}{=}1$ and $w(\sigma)$ the EDM weighting with $\sigma_{\mathrm{data}}{=}1$. The gradient of $\mathcal{L}_{\mathrm{cyc}}^{\mathrm{VP}}$ reaches the action-conditioning MLP, then through $\hat{a}$ into the pose U-Net, the vision encoder, and the U-Net features of the inverse pass, so both directions are trained. Each unlabeled clip is placed in all three camera streams with its first frame as the conditioning frame and the empty instruction; one clip accompanies four labeled samples per GPU per step. Following the protocol of the main experiments, the cycle term is enabled only after the baseline has converged (10k fine-tuning steps), and the ReCycle model is trained for a further 5k steps from that checkpoint.

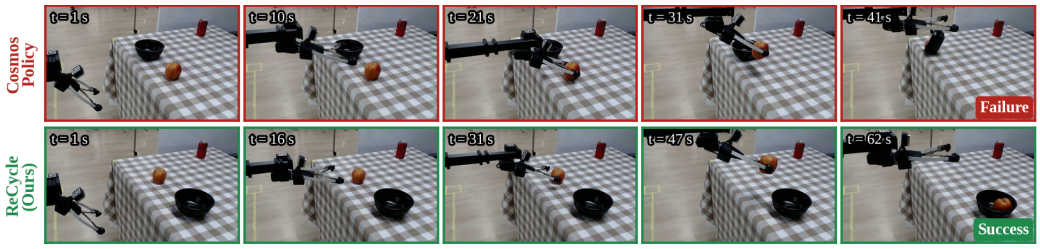


Figure 18: Success rollout for move apple task

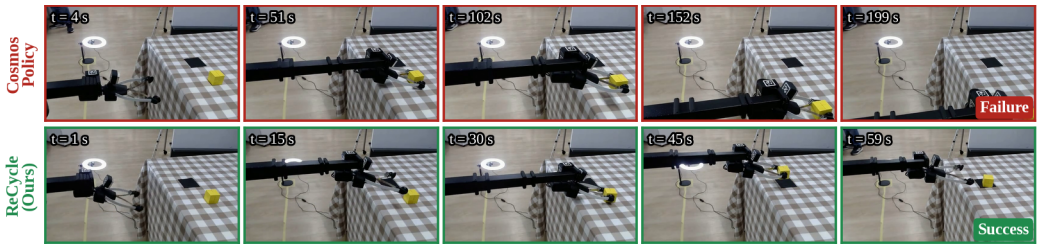


Figure 19: Success rollout for move cube task

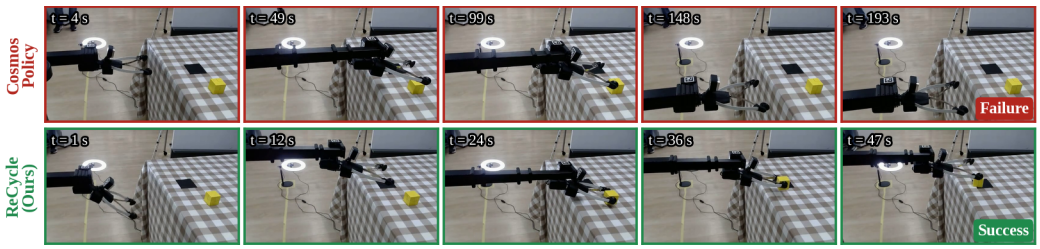


Figure 20: Success rollout for move cube task

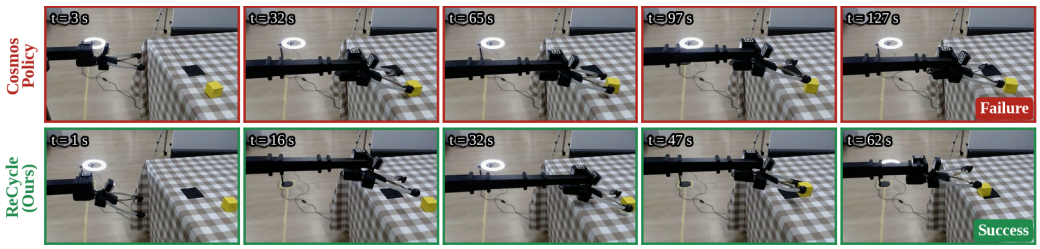


Figure 21: Success rollout for move cube task

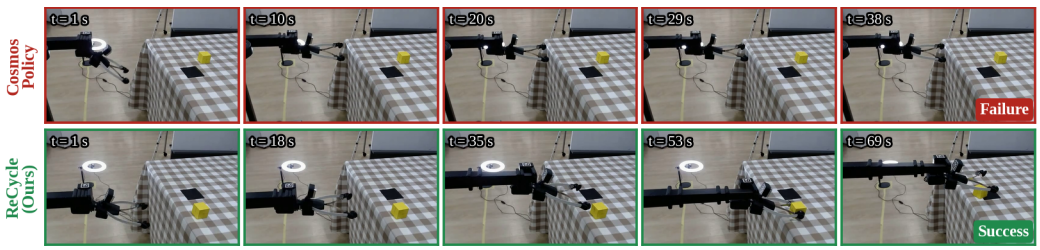


Figure 22: Success rollout for move cube task

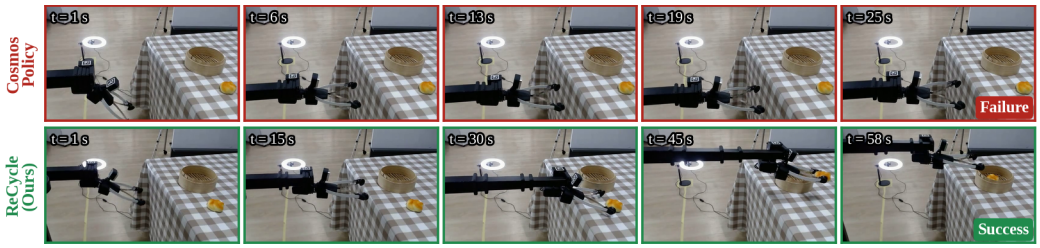


Figure 23: Success rollout for move food task

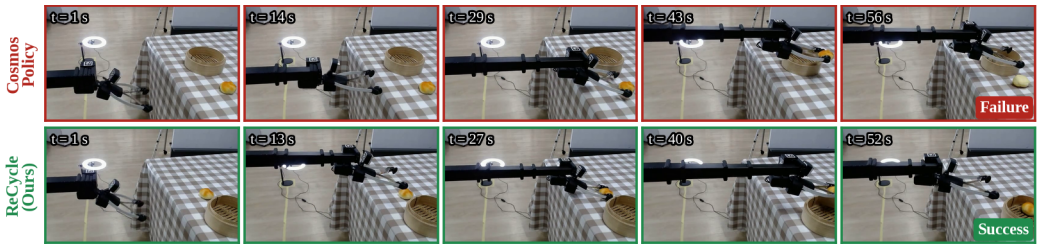


Figure 24: Success rollout for move food task

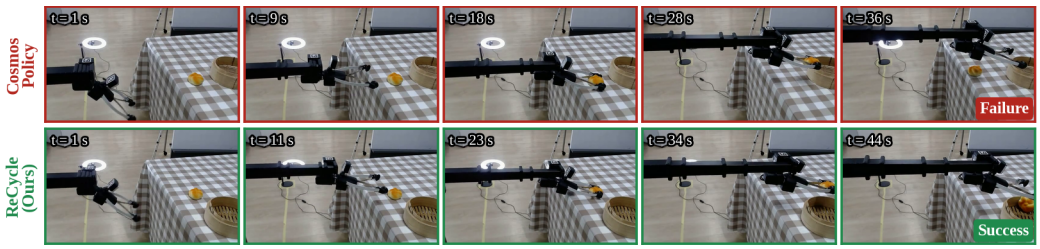


Figure 25: Success rollout for move food task

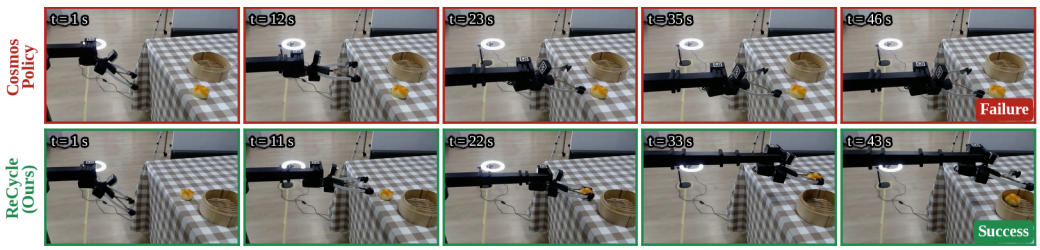


Figure 26: Success rollout for move food task