When Does Sim-and-Real Co-Training Make Simulation a Reliable Evaluator?

Anonymous authors
Paper under double-blind review

Abstract

Evaluating robot policies in the real world is expensive. Simulation offers a scalable alternative, but its value depends on reliably predicting real-world performance. A common belief is that sim-and-real co-training can improve this predictiveness, but when and why remain unclear. We identify co-training's two opposing effects on sim-and-real agreement: *sim-transferring* carries real-learned behaviors into simulation, whereas *sim-overwriting* lets simulated data determine simulated behavior regardless of the policies' real-domain behavioral differences. We use controlled synthetic experiments to isolate how these effects influence sim-to-real agreement across different scenarios. We then test the predictions with manipulation policies across behavioral and visual sim-to-sim gaps and with real-robot evaluation. For behaviorally diverse policies, small simulation-data doses can improve behavior-wise agreement through sim-transferring. Larger doses can hurt agreement through sim-overwriting on states covered by simulated data, while continuing to improve agreement on Sim-OOD states, where direct sim-overwriting is avoided. Step-wise correlation between simulated and target-domain scores generally improves with sim-and-real co-training, but this does *not* necessarily imply behavioral alignment. These results provide practical guidance for reliably designing and interpreting simulation-based policy evaluation.

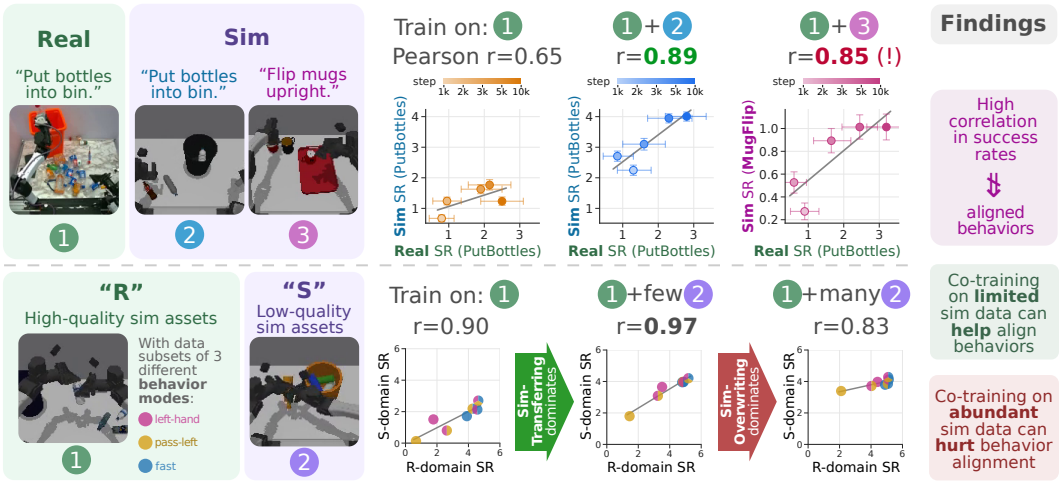


Figure 1: **(Top) real-to-sim evaluation:** (co-)training data and success rates in the two domains at different steps of π_0 post-training. Co-training raises correlation even when the simulated demonstrations come from a task different from the real-world task. **(Bottom) sim-to-sim evaluation:** domains S and R differ in their simulation setups; the policies compared are trained on different subsets of R-domain behavior modes.

1 Introduction

Evaluating generalist robot policies is essential as they increasingly operate across tasks and embodiments (Black et al., 2025; Physical Intelligence et al., 2025; ONeill et al., 2024; Octo Model

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.

Team et al., 2024). Real-world evaluation provides direct evidence of policy performance but requires numerous rollouts that are slow, labor-intensive, and difficult to reproduce. Simulation offers a promising alternative, but its value as an evaluation proxy depends on *reliability*: simulated policy performance must predict real-world performance (Li et al., 2024).

Existing real-to-sim evaluators are often zero-shot, deploying real-trained checkpoints into simulation as-is, which places high demands on SysID quality and simulation fidelity (Li et al., 2024; Zhang et al., 2025). Still, no simulator is perfect, and a residual real-to-sim gap is inevitable. A seemingly promising recent direction is to narrow the gap from the policy side as well: *co-training* on both simulated and real data adapts the policy to the simulation domain while real data keep the policy's real-domain behavior anchored (Jain et al., 2026; Allshire et al., 2026). This alters the policies before evaluating them, making the co-training paradigm less suitable for benchmarking checkpoints as-is, but it is natural for comparing base models or training recipes (Yuan et al., 2026; Wang et al., 2026).

However, it remains unclear how co-training affects evaluation reliability. Prior mechanistic studies of *performance-oriented* co-training explain when and how co-training improves policy performance in a single target domain, typically the real world, assuming abundant simulated data and scarce real data (Lei et al., 2026). In this paper, we instead study co-training in an *evaluation-oriented* setting, examining how it affects success-rate correlation and behavioral correspondence across two domains, assuming a large, diverse real dataset as the primary training corpus and a limited simulated dataset as an evaluation anchor.

This evaluation-oriented use of co-training introduces critical nuances when interpreting evaluation results. Existing simulation-based evaluation studies measure reliability using sim-real correlations and rank-violation metrics computed from *success rates* (Li et al., 2024). Yet simulated evaluation should provide more than a scalar success rate: rollout trajectories can reveal valuable information about policy *behaviors* and failure modes (Allshire et al., 2026). In particular, as we demonstrate in Secs. 4 and 5, a high correlation in success rates does not necessarily imply behavioral alignment across domains. Co-training can produce spuriously high correlations when the policy learns two different behavioral modes, one specific to real-world inputs and the other to simulated inputs, in which case simulated rollouts reveal little about real-world behavior. A solid interpretation therefore requires a mechanistic understanding of how co-training affects policy behavior across domains.

In this study, we identify two effects of co-training that jointly shape evaluation reliability but can pull it in opposite directions: **sim-transferring** carries behavior learned from real data into simulation, while **sim-overwriting** lets shared simulated data determine each policy's simulated behavior independent of their different real data, making their differences unobservable from simulated rollouts. We isolate these effects in a controlled synthetic study and then validate the findings in cross-domain evaluations of manipulation policies, including real-to-sim experiments and scalable sim-to-sim experiments. Our contributions are:

- We formulate an evaluation-oriented framework for analyzing sim-and-real co-training;
- We identify sim-transferring and sim-overwriting as concurrent mechanisms that can improve or degrade evaluation reliability;
- Across controlled synthetic and manipulation experiments, we show that reliability varies with the policy-comparison regime, real-to-sim gap type, and simulation-data dose, explaining why success-rate correlation alone does not establish behavioral alignment across domains.

2 Related Work

Real-to-sim evaluation for manipulation. Simulation-based manipulation benchmarks (Liu et al., 2023; Nasiriany et al., 2024; Mu et al., 2025) measure policy performance within simulation; real-to-sim evaluation instead uses simulation to predict the real-world performance and behavior of the same policies. SIMPLER (Li et al., 2024) introduces a simulated evaluation suite for manipulation policies trained on real-world data and validates performance correlation and ranking consistency with real-world results. Subsequent work spans real-validated simulation benchmarks (Sedlacek et al., 2025; Zhu et al., 2026) and real-to-sim evaluation and scene-construction frameworks (Zhang et al., 2025; Jain et al., 2026; Ranawaka et al., 2026).

Training on sim-and-real data mixtures. Beyond simulator-side methods, sim-and-real co-training can improve real-world success (Maddukuri et al., 2025; Yan et al., 2025). Co-training can also help preserve real-domain policy capabilities and support cross-domain evaluation and diagnosis: Yuan et al. (2026) co-post-train base models on both simulated benchmark data and pre-training data to mitigate policy degradation during simulated evaluation, while Allshire et al. (2026) use large-scale sim-and-real co-training and inspect both success rates and rollout behaviors to verify and diagnose policies. For real-to-sim evaluation, co-training and simulation-only post-training can improve proxy reliability: PolaRiS co-fine-tunes policies on real and simulated data for a limited number of steps (Jain et al., 2026), whereas Wang et al. (2026) post-train policies on limited simulated data. Both report non-monotonic reliability: moderate adaptation helps, while further adaptation can hurt. PolaRiS attributes the improvement to visual adaptation, whereas Wang et al. attribute the downturn to overfitting to simulated data.

Prior real-to-sim studies typically measure reliability over independently developed policies or checkpoints from one training run (Li et al., 2024; Zhang et al., 2025; Jain et al., 2026) while evaluating across only the aggregate real-to-sim gaps of their systems. We complement this work by treating policy-set construction, co-training data support, gap alignability, and gap components as experimental variables. We adopt the mechanistic perspective of Lei et al. (2026), but shift the focus from performance in a single target domain to cross-domain evaluation agreement.

3 Problem Formulation

Let $D \in \{S, R\}$ denote simulated proxy and real target domains sharing a task. Domain D has an observation function $f_o^D(s)$, transition model $p^D(s'|s,a)$, initial-state distribution, instruction, success criterion y^D, and demonstration dataset $\mathcal{D}_D$. A policy $\pi_\theta(a|o)$ maps observations to action-chunk distributions, and $\mathrm{SR}^D(\pi)$ is its probability of satisfying y^D. We co-train each policy on real and simulated demonstrations with weight w:

$$\min_\theta \; (1-w)\mathbb{E}_{\mathcal{D}_R}[\ell(\theta)] + w\,\mathbb{E}_{\mathcal{D}_S}[\ell(\theta)]. \tag{1}$$

Across a policy set Π, $\mathcal{D}_R$ may vary between training recipes, while $\mathcal{D}_S$ is fixed so that simulation evaluates every policy under the same protocol. Reliability is therefore a property of (S, R, Π): the proxy is reliable when SR^S agrees with SR^R across Π in relative magnitude and rank order.

Metrics. Let $\Pi = \{\pi_i\}_{i=1}^N$, $R_i = \mathrm{SR}^R(\pi_i)$, and $S_i = \mathrm{SR}^S(\pi_i)$. We quantify relative-magnitude agreement with Pearson correlation r and rank-order disagreement with mean maximum rank violation (MMRV) (Li et al., 2024):

$$r = \frac{\mathrm{cov}_i(R_i, S_i)}{\sigma_i(R_i)\sigma_i(S_i)}, \quad V_{ij} = |R_i - R_j|\,\mathbf{1}_{[(S_i<S_j)\neq(R_i<R_j)]}, \quad \mathrm{MMRV} = \frac{1}{N}\sum_i \max_j V_{ij}. \tag{2}$$

Higher r and lower MMRV indicate better agreement.

Evaluation-oriented co-training. In performance-oriented co-training, the same objective in Eq. (1) is used to raise SR^R, assuming $\mathcal{D}_R$ is scarce and $\mathcal{D}_S$ abundant (Lei et al., 2026). Our evaluation-oriented setting instead targets raising agreement between SR^S and SR^R across a policy set, with a larger, more diverse $\mathcal{D}_R$ serving as the primary training corpus and a limited $\mathcal{D}_S$ collected specifically to support evaluation.

3.1 A Taxonomy of Real-to-Sim Gaps

Classifying the gaps first requires a domain-specific notion of which actions succeed. Following Gao et al. (2026), let π_E^D denote an *expert* policy that solves the task optimally in domain $D \in \{S, R\}$. An action chunk is *viable* at state s if executing it and then handing control to π_E^D yields the same probability of success as letting π_E^D act from s onward. We denote the set of such chunks by $\mathcal{V}^D(s)$. A policy's success rate is high when its chunks lie in $\mathcal{V}^D(s)$ at the states it visits and drops as they leave it.

We characterize the real-to-sim gap by behavioral and visual components, paralleling the classification that Gao et al. (2026) applies to task perturbations. We further divide each component by *alignability*: whether it can potentially be repaired from the policy side through co-training (Fig. 2).

Behavioral gap (B). A behavioral gap changes which actions succeed: $\mathcal{V}^S(s) \neq \mathcal{V}^R(s)$. Under STAR-Gen (Gao et al., 2026), such a gap is purely behavioral when observations are unchanged and visual–behavioral (VB) when it also changes observations ($f_o^S \neq f_o^R$). The former case, e.g., an unobservable friction change, corresponds to the *overlapping* regime of Lei et al. (2026): when policies cannot distinguish the domains, co-training does not help and can even cause negative transfer. Rather than revisit this input-indiscernibility failure mode, we study behavioral gaps with similar but discernible observations: the synthetic study uses an explicit domain flag (Sec. 4), while the sim-to-sim experiments use behavior-dominant VB gaps with small visual differences (Sec. 5.1). Within this scope, let $\delta(s)$ map viable actions and near misses in R to their counterparts in S. The behavioral gap is alignable (**B-A**) when a single δ generalizes across states, and unalignable (**B-U**) when the correction at one state predicts nothing about another. For example, omitted gripper compliance in simulation may be corrected by a uniformly firmer grasp throughout (B-A), whereas omitted screw threads turn screwing into peg insertion, for which no consistent δ exists (B-U).

Visual-only gap (V). A visual-only gap changes observations ($f_o^S \neq f_o^R$) without changing which actions succeed ($\mathcal{V}^S(s) = \mathcal{V}^R(s)$). The gap is alignable (**V-A**) when both renderings retain the same task-relevant information but encode it differently, for example through texture, lighting, or camera parameters. The gap can become unalignable (**V-U**) when the proxy omits a required cue or provides a privileged cue that the policy can exploit. The remainder of this paper focuses on V-A, the common real-to-sim rendering setting.

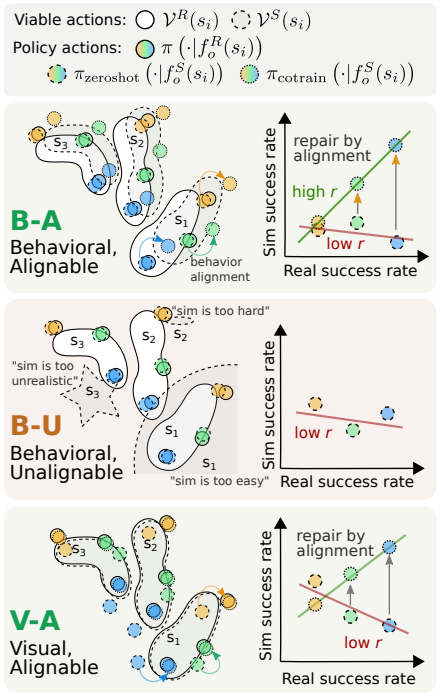


Figure 2: **Taxonomy of real-to-sim gaps.**

3.2 Two Effects of Co-Training

We now study co-training as a policy-side repair for alignable gaps. Under this protocol, co-training must bridge the domains while preserving the real-domain differences among policies that the proxy measures, giving rise to two concurrent effects.

Sim-transferring. Simulated demonstrations show the policy what succeeds in S, allowing it to learn δ. When the gap is alignable, δ generalizes across evaluated states, mapping for each (π, s) the action chunk predicted from $f_o^R(s)$ to its counterpart under $f_o^S(s)$. Actions in $\mathcal{V}^R(s)$ map into $\mathcal{V}^S(s)$, whereas nonviable actions transfer the same failure; thus, differences in R-domain competence remain reflected in S, resulting in high reliability.

Sim-overwriting. Enough simulated data can instead teach the behavior S demands directly. Because $\mathcal{D}_S$ is shared across Π, all policies may learn the same simulated behavior despite behaving differently in R. Then SR^S reflects what $\mathcal{D}_S$ taught rather than what $\mathcal{D}_R$ did, weakening the behavioral coupling between R and S and reducing reliability. This need not lower performance in either domain and may even raise R-domain performance. This effect is concurrent with the first one: the simulated data that enables sim-transferring can also produce sim-overwriting.

The balance between the effects depends on the amount and coverage of $\mathcal{D}_S$: limited simulated supervision favors sim-transferring, whereas dense supervision enables sim-overwriting. Whether overwriting harms reliability depends on Π: it can erase differences among policies trained on different real-domain behavior modes, yet still yield a high success-rate correlation among policies that differ mainly in fit quality. Sec. 4 isolates these regimes with a controlled study on synthetic data.

4 A Controlled Synthetic Study

We build on the construction of Lei et al. (2026) by training a small diffusion model to fit a closed-form map with a domain flag, and add evaluation-oriented controls over gap alignability, nested data supports, and policy-set variations defining different comparison regimes. The two regimes mirror

common evaluation use cases: comparing policy architectures or training recipes while holding the training data fixed, and comparing training datasets through their effects on learned policy behavior.

Experiment design. We co-train a conditional diffusion model $\pi_\theta(a|o)$ to fit a closed-form map $\psi : \mathbb{R}^3 \to \mathbb{R}^2$. Fig. 3 visualizes ψ. We interpret its inputs as observations and outputs as actions, making it an analog of a policy. The observation $o = (o_u, o_v, o_D)$ contains coordinates (o_u, o_v) shared across domains and a domain flag $o_D \in \{o_R, o_S\}$. The ground-truth action is $a = \psi(o_u, o_v|o_D)$, with $\psi(\cdot|o_S) = \psi(\cdot|o_R) + \delta(\cdot)$. A constant δ produces an *alignable* gap, whereas varying δ with (o_u, o_v) produces an *unalignable* one. Each map traces a narrow band of several segments, with o_u along its arc length and o_v across its width. Varying segment curvature changes the local state-to-action structure, analogous to different stages of a manipulation task.

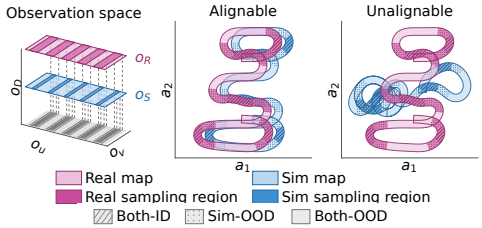


Figure 3: **Synthetic data.** Left: observation space, with nested data sampling regions (projected onto the same plane for visualization). Middle and right: ground-truth actions $\psi(\cdot|o_R)$ and $\psi(\cdot|o_S)$ with their respective sampling regions, under alignable and unalignable gaps.

Data sampling. Each dataset $\mathcal{D}_D$ contains pairs $(o + \epsilon, a)$ with $o \sim P_D$, $a = \psi(o_u, o_v|o_D)$, and noise ϵ. Every segment has nested supports, $\mathrm{supp}(P_S) \subset \mathrm{supp}(P_R) \subset \mathrm{supp}(P_E)$, where P_E is the full evaluation distribution. This models simulated data bounded by renderer and physics coverage versus broader real variation. The nesting defines three regions: ***Both-ID***, where both domains train; ***Sim-OOD***, where only real data trains; and ***Both-OOD***, where neither does. Fig. 3 shows these regions and their ground-truth actions. Appendix C provides full implementation details.

4.1 Scenario 1: Policies differ by capacity and training

We first construct Π from capacity and training length alone: MLPs of 10 widths, each contributing its last three checkpoints, yield 30 policies that differ only in how well they fit the same data. As real-to-sim evaluation is commonly used to compare checkpoints at different training steps or different model architectures trained on the same data (Allshire et al., 2026; Zhang et al., 2025; Li et al., 2024), we test whether co-training improves its reliability for this policy set. With $|\mathcal{D}_R| = 3000$ fixed, we sweep $|\mathcal{D}_S|{:}|\mathcal{D}_R|$ from 1:600 to 1:1 over eight values, plus an R-only reference ($w = 0$); co-trained arms fix the sim sampling weight to $w = 0.1$ following the practice in Jain et al. (2026). We define SR^D as a smoothed average score based on the distance between predicted actions and $\psi(\cdot|o_D)$. Fig. 4(a,b) reports Pearson r and MMRV between SR^S and SR^R across Π at each ratio in Both-ID and Sim-OOD.

Observations. With R-only training, the domain gap collapses every policy's SR^S to near zero, leaving the proxy unable to discriminate among them. Adding even a small amount of sim data restores the policy ordering in Both-ID: under both gaps, r rises sharply and MMRV falls, remaining near their ceiling and floor throughout the ratio sweep (Fig. 4(a1,a2)). This agreement does *not* require behavioral alignment. Under the unalignable gap, co-trained policies produce different actions in S and R, yet their Both-ID r stays close to 1.0 (Fig. 4(a2)). Because the policies differ only in how well they fit the shared data, each policy's capacity and training length govern its fit to both targets, causing SR^S and SR^R to vary together. In the language of Sec. 3, these Both-ID scores are produced by *sim-overwriting*, which is benign here: it decouples SR^S from the R-domain training data, but no policy distinction is lost because every policy shares that data. Outside sim support, only *sim-transferring* can preserve the ordering: at larger co-training doses, Sim-OOD r is near 1.0 under the alignable gap (Fig. 4(a3)) but remains lower under the unalignable one (Fig. 4(a4)). We conclude that, for policy sets varying only in capacity and training progress, co-training can improve sim-and-real success-rate agreement even without behavioral alignment; high agreement in Both-ID regions therefore does not by itself establish behavioral correspondence across domains.

4.2 Scenario 2: Policies differ by behavior

We next construct a Π whose members differ by *which behaviors they learned* in R, rather than by fit quality. We create this variation through $\mathcal{D}_R$ coverage: 16 variants share one capacity, but each

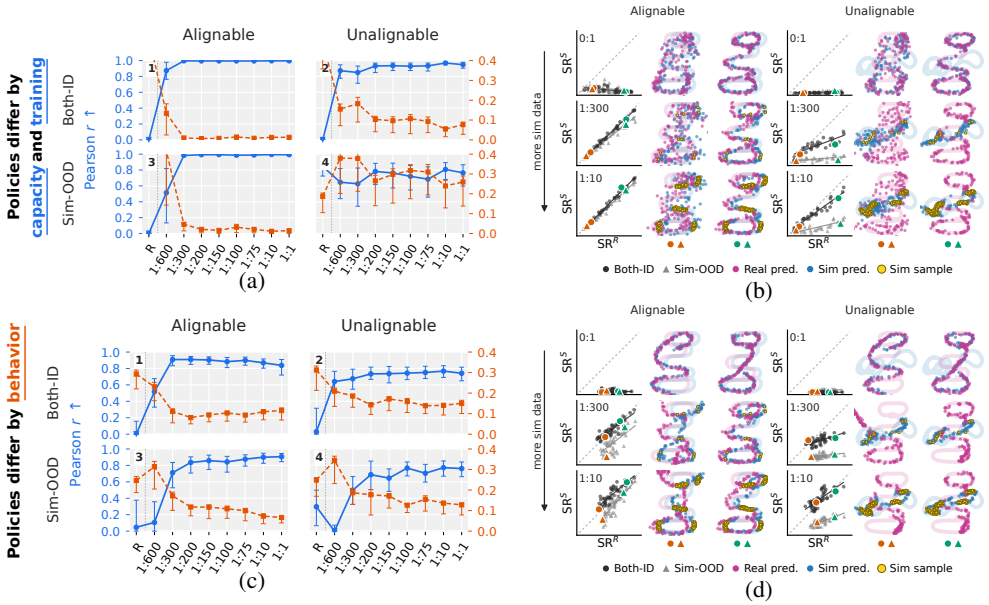


Figure 4: **Two synthetic policy sets expose different co-training effects.** (a,b) Policies differ by capacity and training (Scenario 1); (c,d) policies differ by behavior, realized through different $\mathcal{D}_R$ coverage (Scenario 2). (a,c) Pearson r and MMRV between SR^S and SR^R against the $|\mathcal{D}_S|$:$|\mathcal{D}_R|$ ratio; error bars are 95% bootstrap CIs over Π. (b,d) SR^S vs. SR^R over Π at selected $|\mathcal{D}_S|$:$|\mathcal{D}_R|$ ratios, with selected policies' action predictions drawn over the ground-truth actions for S and R domains. Appendix C provides additional visualizations for (b,d).

omits a different subset of the five segments, with up to two omitted. The shared $\mathcal{D}_S$ covers all five segments, and each variant contributes its final three checkpoints as in Scenario 1. We sweep the same $|\mathcal{D}_S|$:$|\mathcal{D}_R|$ ratios and report the same metrics in Fig. 4(c,d).

Observations. The two effects of Sec. 3 now oppose each other, with outcomes depending on the region. In Sim-OOD under the alignable gap, no simulated data covers the evaluated states, so SR^S can reflect $\mathcal{D}_R$ coverage only through *sim-transferring*. Here, r rises monotonically with $|\mathcal{D}_S|$ as the learned alignment increasingly carries real-learned behavior into S (Fig. 4(c3)). In Both-ID, the effects compete: r first rises, then slowly falls as *sim-overwriting* takes over (Fig. 4(c1)). A large $|\mathcal{D}_S|$ lets every policy fit the shared simulated data directly, so SR^S depends less on which segments it observed in $\mathcal{D}_R$, which is the axis along which Π varies. Under the unalignable gap, Both-ID r is lower than in Scenario 1 because there is no shared fit quality to preserve agreement, while the varying S–R shift blocks sim-transferring from tying SR^S to $\mathcal{D}_R$ coverage (Fig. 4(c2)). Thus, for behaviorally diverse policy sets, co-training on a limited sim dataset improves reliability, while a large one erodes it on the states sim data covers (Both-ID), but Sim-OOD reliability grows consistently with the dose. This suggests keeping the simulated corpus small and, where possible, evaluating scene and object variations beyond its support, where scores can only reflect transferred real behavior.

5 Experiments

We next test whether the observations from the controlled synthetic study extend to VLA policies for manipulation tasks, whose domain gaps are not captured by a closed-form δ. To do so, we first run sim-to-sim experiments that isolate a behavioral gap and a visual gap, evaluating each setting and policy checkpoint at scale for statistically meaningful correlation comparisons. We then extend to real-to-sim evaluation. All co-training runs use $w = 0.1$; Appendix B provides the full policy-training recipes.

5.1 Sim-to-Sim Behavioral Gaps

Tasks and domains. We adapt two simulated tasks from ABC-Sim (Allshire et al., 2026) as target domains (R) and construct their paired S domains shifted by a behavioral gap (Fig. 5). Both tasks

are scored by progress (referred to as SR hereafter): *PutBottles* by the peak number of bottles in the bin out of six, and *MugFlip* by the peak number of upright mugs out of two. The manipulated objects in both R domains are scanned meshes from RoboCasa (Nasiriany et al., 2024). To mimic the behavioral component of a real-to-sim gap, we pair each R with an S domain whose objects are *crude* stand-ins assembled from simple box and cylinder primitives: PutBottles adopts a legacy version of the same task in ABC-Sim as its S domain, and MugFlip's S domain uses a family of hexagonal-prism mugs. For PutBottles we additionally enlarge the physical gap by lowering bottle friction to $\mu = 0.5$ from 4.0 and raising bottle mass from 15 g to 50 g. These shifts in assets and physics change what a policy must do; directly replaying R-domain demonstrations would fail in S. Both domains use MJWarp simulation (Google DeepMind & NVIDIA, 2025).

In addition to those two *same-task* co-training settings with paired S and R domains, a third *cross-task* co-training setting builds an *unalignable* gap by setting S to the PutBottles target domain and R to the MugFlip target domain.

Policies and evaluation. Per task we teleoperate three *behavior modes* in the R domain (Fig. 5), 25 episodes per mode. We fine-tune π_0 (Black et al., 2025) on each mode and on each combination of modes, giving seven policies trained on different $\mathcal{D}_R$. Behavior-wise correlations compare those seven at their final checkpoints; step-wise correlations track the full-coverage (trained on all 3 modes in R; $|\mathcal{D}_R| = 75$) policy across five checkpoints per seed at different training steps. Every checkpoint is scored on 150 rollout episodes. The S-domain demonstrations that form $\mathcal{D}_S$ are teleoperated separately, always covering all behavior modes. We vary the co-training dose $|\mathcal{D}_S|$ from 0 to 15 episodes; at each dose, the same $\mathcal{D}_S$ is shared across all seven trainings.

Results. Fig. 6 and Table 1(a) summarize the behavioral-gap results, which are consistent with key observations for Both-ID regions in the synthetic study in Sec. 4.

On the two tasks with alignable S–R behavioral gaps, co-training on a small dose of $\mathcal{D}_S$ improves simulation's reliability as an evaluation proxy for *behavior-wise* comparisons among policies trained

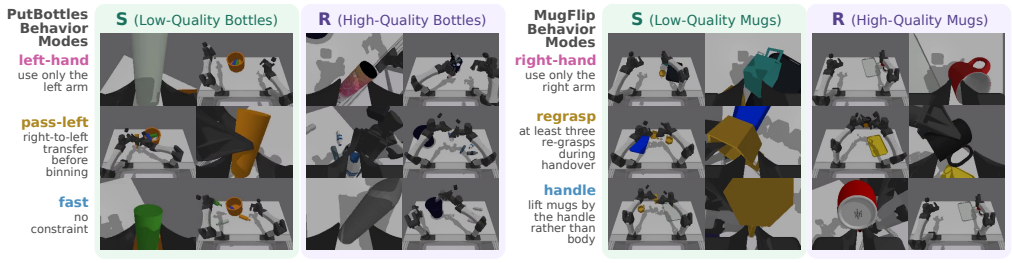


Figure 5: **Behavioral-gap domains and behavior modes.**

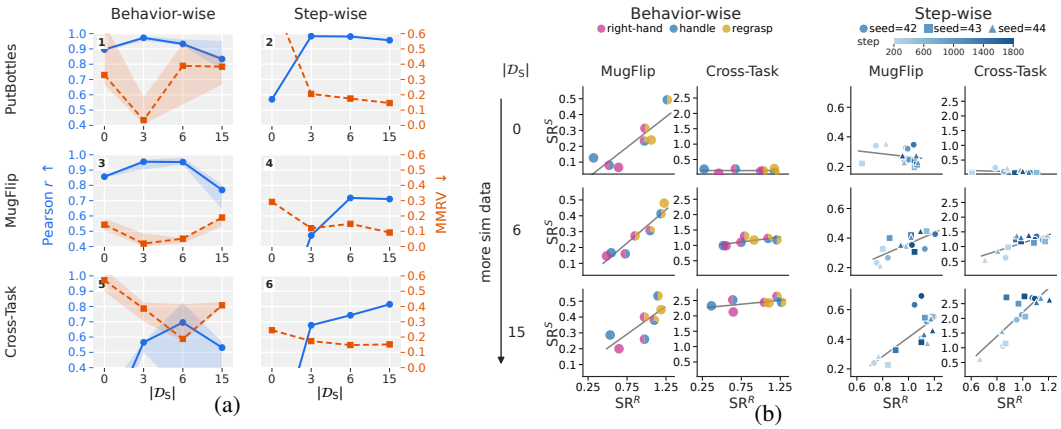


Figure 6: **Behavioral-gap results.** (a) Pearson r and MMRV versus $|\mathcal{D}_S|$ for behavior-wise and step-wise policy comparisons. For each seed, each behavior-wise data point computes over 7 training runs at their final checkpoints; curves show the median over 3 seeds, with shading spanning their min–max range. Each step-wise data point computes over all 15 checkpoints (3 seeds × 5 steps). (b) Selected SR^S versus SR^R scatters. Behavior-wise scatters show the 7 policies from the median-r seed; step-wise scatters show all 15 checkpoints. Gray lines are least-squares fits.

Table 1: **Sim-to-sim evaluation reliability.** Behavior-wise values are the separate medians over three seeds; step-wise values pool 15 checkpoints (three seeds × five steps).

(a) Behavioral gaps

		PutBottles		MugFlip		Cross-Task	
	$\lvert\mathcal{D}_S\rvert$	Behavior	Step	Behavior	Step	Behavior	Step
Pearson r (↑)	0	0.90	0.57	0.86	−0.24	−0.02	−0.22
	3	**0.97**	**0.98**	**0.95**	0.47	0.57	0.68
	6	0.93	**0.98**	**0.95**	**0.72**	**0.70**	0.74
	15	0.83	0.96	0.77	0.71	0.53	**0.81**
MMRV (↓)	0	0.33	0.84	0.14	0.29	0.57	0.24
	3	**0.03**	0.21	**0.02**	0.12	0.39	0.17
	6	0.39	0.18	0.05	0.15	**0.19**	**0.15**
	15	0.38	**0.15**	0.19	**0.09**	0.41	0.15

(b) Visual gap

		Same-task		Hold-out	
	$\lvert\mathcal{D}_S\rvert$	Behavior	Step	Behavior	Step
Pearson r (↑)	0	**0.81**	0.84	0.78	0.65
	3	0.71	0.57	0.86	0.74
	15	0.73	0.81	**0.96**	0.76
	75	−0.10	**0.87**	0.94	**0.82**
MMRV (↓)	0	0.15	0.13	0.11	0.19
	3	0.14	0.25	0.04	0.21
	15	**0.11**	**0.10**	**0.01**	**0.14**
	75	0.34	0.14	0.01	0.14

on data covering different behavior modes (Fig. 6(a1,a3)). At $|\mathcal{D}_S| = 3$, both PutBottles and MugFlip attain their highest behavior-wise r and lowest MMRV, improving over $|\mathcal{D}_S| = 0$ (training only on R-domain data, without co-training). However, co-training on a large dose of $\mathcal{D}_S$ hurts behavior-wise evaluation reliability: at $|\mathcal{D}_S| = 15$, both tasks fall below the $|\mathcal{D}_S| = 0$ baseline in behavior-wise r and rise above it in MMRV (Fig. 6(a1,a3)). Cross-task behavior-wise r also improves with co-training but remains visibly below that of the alignable gaps (Fig. 6(a5)). *Step-wise* reliability likewise improves with co-training but, unlike behavior-wise reliability, does not degrade at larger $|\mathcal{D}_S|$ (Fig. 6(a2,a4)). This improvement appears even for cross-task co-training (Fig. 6(a6)): at $|\mathcal{D}_S| \geq 6$, its step-wise r rises to the same level as MugFlip despite using a fundamentally different task as its S domain. Thus, a rise in step-wise correlation is insufficient to indicate co-training's success in aligning behaviors across two domains. In addition to using success rates as an indicator, Appendix A applies a probe test to directly reveal the sim-overwriting effect in this setting.

5.2 Sim-to-Sim Visual Gaps

Tasks and domains. We construct a visual-only gap on MugFlip by rendering S with MJWarp's Batched Renderer (Google DeepMind & NVIDIA, 2025) and R with Isaac Sim's Omniverse RTX Renderer (NVIDIA, 2025), while holding assets and physics fixed and using MJWarp's physical simulator for dynamics in both domains. Fig. 7 visualizes sample frames from both domains.

Policies and evaluation. We reuse the MugFlip environments and policy sets from Sec. 5.1. The R-domain demonstrations match those in Sec. 5.1 except for the renderer, while the S-domain demonstrations follow the same distribution and cover all three behavior modes. $\mathcal{D}_S$ and $\mathcal{D}_R$ share no common demonstration episodes. Evaluation in the R domain uses a *hybrid simulator* combining MJWarp's dynamics and Isaac Sim's renderer.

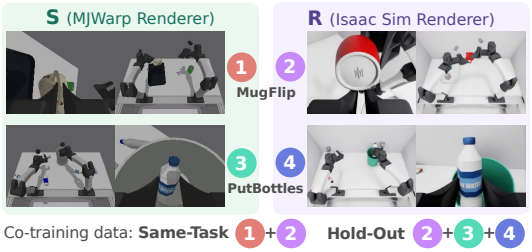


Figure 7: **Visual-gap domains and co-training data.**

Alongside the *same-task* setting, which co-trains on MugFlip demonstrations from both domains, we follow Jain et al. (2026) and study a *hold-out* setting in which PutBottles is used for co-training but not evaluation: at every dose, $\mathcal{D}_S$ contains only PutBottles demonstrations, whereas $\mathcal{D}_R$ contains each MugFlip behavior-mode combination above and a fixed set of 75 PutBottles demonstrations shared across all seven policies. $\mathcal{D}_S$ and $\mathcal{D}_R$ use disjoint PutBottles episodes from ABC-Sim (Allshire et al., 2026), rendered with their respective renderers. Evaluation remains on MugFlip, and Fig. 7 summarizes the data support. Under the definitions of Sec. 4, MugFlip evaluation is Both-ID in the same-task setting and Sim-OOD in the hold-out setting.

Results. Fig. 8 and Table 1(b) summarize the results. Unlike the behavioral gaps in Sec. 5.1, small doses of same-task co-training fail to significantly improve behavior-wise reliability across the visual gap, while the largest dose collapses it as S-domain scores cease to distinguish the policies (Fig. 8(a1)). In contrast, co-training on a held-out task consistently improves proxy reliability across the visual gap: every nonzero dose improves behavior-wise reliability over $|\mathcal{D}_S| = 0$, with no degradation at the largest dose (Fig. 8(a3)). The large-dose contrast agrees with Sec. 4. Step-wise correlation remains high even when same-task behavior-wise reliability collapses (Fig. 8(a2)), showing that agreement across training steps does not guarantee reliable comparisons among poli-

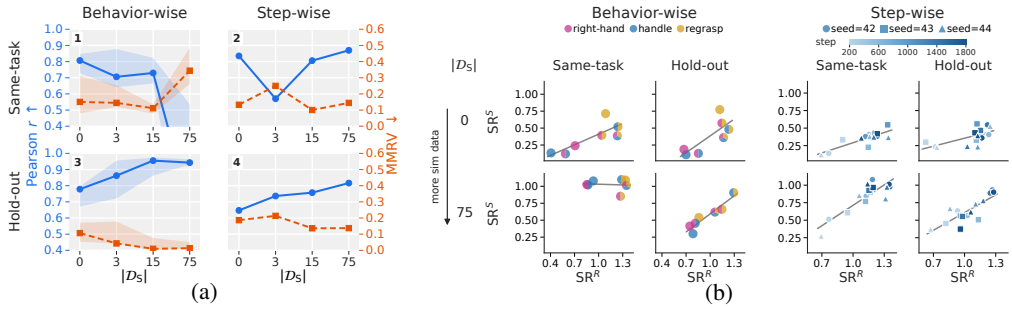


Figure 8: **Visual-gap results.** Plotting conventions follow Fig. 6. All evaluations are on MugFlip. Same-task co-training uses MugFlip data from both domains; hold-out co-training learns visual alignment from PutBottles without seeing S-domain MugFlip in training.

cies with different behaviors. Similar to Sec. 5.1, Appendix A applies a probe test to complement the observations on success rates and reveal sim-overwriting more directly.

5.3 Real-to-Sim Evaluation

We now test whether the step-wise findings from the sim-to-sim experiments hold in a real-to-sim setting, where the gap combines behavioral and visual differences.

Domains and data. Fig. 1 shows sample frames from all domains. The target domain R is real-world PutBottles, with $\mathcal{D}_R$ containing 300 real PutBottles demonstrations from ABC-130K (Allshire et al., 2026). For the *same-task* co-training setting, the simulated domain S is the MJWarp PutBottles target domain from Sec. 5.1 (with high-quality RoboCasa object assets), and $\mathcal{D}_S$ uses 30 PutBottles demonstrations sampled from ABC-Sim. Following Sec. 5.1, the *cross-task* setting keeps R as real-world PutBottles but replaces S with simulated MugFlip; its $\mathcal{D}_S$ contains 30 ABC-Sim MugFlip demonstrations.

Policies and evaluation. We compare real-only training with same-task and cross-task co-training. Each of the three settings contributes five checkpoints from one π_0 post-training run, spanning steps 1k to 10k. Every checkpoint is evaluated on 20 real PutBottles episodes and 150 simulated episodes. The simulated evaluation task is PutBottles for real-only training and same-task co-training, and MugFlip for cross-task co-training.

Results. Fig. 1 reports step-wise Pearson r and MMRV across the five checkpoints. Same-task co-training raises step-wise r from the Real-Only baseline of 0.65 to 0.89 and lowers MMRV from 0.81 to 0.18. Cross-task co-training produces gains of similar magnitude, reaching $r = 0.85$, only slightly below same-task despite using fundamentally different tasks in S and R. This extends the finding of Sec. 5.1 to real evaluation: an increase in step-wise correlation through co-training does not require behavior alignment between the simulated and real tasks.

6 Discussion

In this paper, we identify two concurrent effects through which co-training shapes evaluation reliability. Sim-transferring preserves behavioral differences from R in S, while sim-overwriting can hide them. Therefore, high sim-real success-rate correlation, especially across checkpoints at different training steps, does not by itself show that policy behaviors match across domains. When using simulated rollouts to interpret policy behavior or diagnose differences in $\mathcal{D}_R$, researchers should verify that the rollouts preserve the relevant behavioral differences.

Limitations and future work. We evaluate two regimes of policy variation: step-wise comparisons vary the training step while holding the real-domain data fixed; behavior-wise comparisons vary the behavior modes covered by the real-domain data. Comparisons among different pretrained base models, architectures, or inference settings remain untested. Our experiments cover two rigid-body manipulation tasks across three cross-domain settings; broader tasks and domain gaps, especially soft-body and more complex contact-rich manipulation, also remain to be evaluated. Future work should further compare physics engines and renderers with different levels of fidelity.

AI USE STATEMENT

In this work, we used generative AI tools to generate synthetic datasets (Sec. 4); help develop theoretical models and conceptual frameworks; design and provide feedback on research methodology and experiments; implement methods; clean and reformat datasets; support qualitative and thematic data analysis; and interpret results.

We did not use generative AI tools to assist with translation.

The following are not applicable to this work: formulating mathematical claims; providing critical ingredients for proving mathematical claims; assisting in the writing of proofs; proposing or refining hypotheses.

Additionally, we used generative AI tools to create and modify scientific figures based on experimental data; draft parts of the research paper (Appendices A, B, and C); summarize and analyze existing literature; discover research topics and identify gaps; brainstorm; source and search for information; edit the research paper to improve readability; identify relevant literature; and format references.

We have reviewed all AI-assisted work. We reviewed the LLM-generated code for policy training, policy evaluation, data generation, and data reformatting. We verified scientific figures plotted with LLM-generated code against the original data. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Arthur Allshire, Himanshu Gaurav Singh, Ritvik Singh, Adam Rashid, Hongsuk Choi, David McAllister, Justin Yu, Yiyuan Chen, Huang Huang, Pieter Abbeel, Xi Chen, Rocky Duan, Phillip Isola, Jitendra Malik, Fred Shentu, Guanya Shi, Philipp Wu, and Angjoo Kanazawa. Scalable behavior cloning with open data, training, and evaluation. *arXiv preprint arXiv:2606.27375*, 2026.

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

Jensen Gao, Suneel Belkhale, Sudeep Dasari, Ashwin Balakrishna, Dhruv Shah, and Dorsa Sadigh. A taxonomy for evaluating generalist robot manipulation policies. *IEEE Robotics and Automation Letters*, 11(3):3182–3189, 2026. doi: 10.1109/LRA.2026.3656785.

Google DeepMind and NVIDIA. MuJoCo Warp (MJWarp), 2025. URL `https://github.com/google-deepmind/mujoco_warp`.

Arhan Jain, Mingtong Zhang, Kanav Arora, William Chen, Marcel Torne, Muhammad Irshad, Sergey Zakharov, Yue Wang, Sergey Levine, Chelsea Finn, Wei-Chiu Ma, Dhruv Shah, Abhishek Gupta, and Karl Pertsch. PolaRiS: Scalable real-to-sim evaluations for generalist robot policies. In *Robotics: Science and Systems XXII*, RSS2026. Robotics: Science and Systems Foundation, July 2026. doi: 10.15607/rss.2026.xxii.062. URL `http://dx.doi.org/10.15607/RSS.2026.XXII.062`.

Yu Lei, Minghuan Liu, Abhiram Maddukuri, Zhenyu Jiang, and Yuke Zhu. A mechanistic analysis of sim-and-real co-training in generative robot policies. In *Forty-third International Conference on Machine Learning*, 2026.

Xuanlin Li, Kyle Hsu, Jiayuan Gu, Oier Mees, Karl Pertsch, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, Sergey Levine, Jiajun Wu, Chelsea Finn, Hao Su, Quan Vuong, and Ted Xiao. Evaluating real-world robot manipulation policies in simulation. In *8th Annual Conference on Robot Learning*, 2024.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 44776–44791, 2023. doi: 10.52202/075280-1939.

Abhiram Maddukuri, Zhenyu Jiang, Lawrence Yunliang Chen, Soroush Nasiriany, Yuqi Xie, Yu Fang, Wenqi Huang, Zu Wang, Zhenjia Xu, Nikita Chernyadev, Scott Reed, Ken Goldberg, Ajay Mandlekar, Linxi Fan, and Yuke Zhu. Sim-and-real co-training: A simple recipe for vision-based robotic manipulation. In *Robotics: Science and Systems XXI*, Los Angeles, CA, USA, 2025. doi: 10.15607/RSS.2025.XXI.109.

Yao Mu, Tianxing Chen, Zanxin Chen, Shijia Peng, Zhiqian Lan, Zeyu Gao, Zhixuan Liang, Qiaojun Yu, Yude Zou, Mingkun Xu, Lunkai Lin, Zhiqiang Xie, Mingyu Ding, and Ping Luo. Robotwin: Dual-arm robot benchmark with generative digital twins. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27649–27660, 2025. doi: 10.1109/CVPR52734.2025.02575.

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. In *Robotics: Science and Systems (RSS)*, 2024.

NVIDIA. Isaac Sim, 2025. URL `https://github.com/isaac-sim/IsaacSim`. Version 6.0.1.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Charles Xu, Jianlan Luo, Tobias Kreiman, You Liang Tan, Lawrence Yunliang Chen, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, 2024.

Abby ONeill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, Andrew Wang, Anikait Singh, Animesh Garg, Aniruddha Kembhavi, Annie Xie, Anthony Brohan, Antonin Raffin, Archit Sharma, Arefeh Yavary, Arhan Jain, Ashwin Balakrishna, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Blake Wulfe, Brian Ichter, Cewu Lu, Charles Xu, Charlotte Le, Chelsea Finn, Chen Wang, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Christopher Agia, Chuer Pan, Chuyuan Fu, Coline Devin, Danfei Xu, Daniel Morton, Danny Driess, Daphne Chen, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dinesh Jayaraman, Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Ethan Foster, Fangchen Liu, Federico Ceola, Fei Xia, Feiyu Zhao, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Gilbert Feng, Giulio Schiavi, Glen Berseth, Gregory Kahn, Guanzhi Wang, Hao Su, Hao-Shu Fang, Haochen Shi, Henghui Bao, Heni Ben Amor, Henrik I Christensen, Hiroki Furuta, Homer Walke, Hongjie Fang, Huy Ha, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jad Abou-Chakra, Jaehyung Kim, Jaimyn Drake, Jan Peters, Jan Schneider, Jasmine Hsu, Jeannette Bohg, Jeffrey Bingham, Jeffrey Wu, Jensen Gao, Jiaheng Hu, Jiajun Wu, Jialin Wu, Jiankai Sun, Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jimmy Wu, Jingpei Lu, Jingyun Yang, Jitendra Malik, João Silvério, Joey Hejna, Jonathan Booher, Jonathan Tompson, Jonathan Yang, Jordi Salvador, Joseph J. Lim, Junhyek Han, Kaiyuan Wang, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Black, Kevin Lin, Kevin Zhang, Kiana Ehsani, Kiran Lekkala, Kirsty Ellis, Krishan Rana, Krishnan Srinivasan, Kuan Fang, Kunal Pratap Singh, Kuo-Hao Zeng, Kyle Hatch, Kyle Hsu, Laurent Itti, Lawrence Yunliang Chen, Lerrel Pinto, Li Fei-Fei, Liam Tan, Linxi Jim Fan, Lionel Ott, Lisa Lee, Luca Weihs, Magnum Chen, Marion Lepert, Marius Memmel, Masayoshi Tomizuka, Masha Itkina, Mateo Guaman Castro, Max Spero, Maximilian Du, Michael Ahn, Michael C. Yip, Mingtong Zhang, Mingyu Ding, Minho Heo, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J Joshi, Niko Suenderhauf, Ning Liu, Norman Di Palo, Nur Muhammad Mahi Shafiullah, Oier Mees, Oliver Kroemer, Osbert Bastani, Pannag R Sanketi, Patrick Tree Miller, Patrick Yin, Paul Wohlhart, Peng Xu, Peter David Fagan, Peter Mitrano, Pierre Sermanet, Pieter Abbeel, Priya Sundaresan, Qiuyu Chen, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto Martín-Martín, Rohan Baijal, Rosario Scalise, Rose Hendrix, Roy Lin, Runjia Qian, Ruohan Zhang, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Shan Lin, Sherry Moore, Shikhar Bahl, Shivin Dass, Shubham Sonawani, Shuran Song, Sichun Xu, Siddhant Haldar, Siddharth Karamcheti, Simeon Adebola, Simon Guist, Soroush Nasiriany,

Stefan Schaal, Stefan Welker, Stephen Tian, Subramanian Ramamoorthy, Sudeep Dasari, Suneel Belkhale, Sungjae Park, Suraj Nair, Suvir Mirchandani, Takayuki Osa, Tanmay Gupta, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Thomas Kollar, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Trinity Chung, Vidhi Jain, Vincent Vanhoucke, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiaolong Wang, Xinghao Zhu, Xinyang Geng, Xiyuan Liu, Xu Liangwei, Xuanlin Li, Yao Lu, Yecheng Jason Ma, Yejin Kim, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Yilin Wu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yue Cao, Yueh-Hua Wu, Yujin Tang, Yuke Zhu, Yunchu Zhang, Yunfan Jiang, Yunshuang Li, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zehan Ma, Zhuo Xu, Zichen Jeff Cui, Zichen Zhang, and Zipeng Lin. Open X-Embodiment: Robotic learning datasets and RT-X models. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6892–6903, 2024. doi: 10.1109/ICRA57147.2024.10611477.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Nadun Ranawaka, Josiah Wong, Wei-Lin Pai, Wei-Teng Chu, Tianyuan Dai, Masoud Moghani, Hang Yin, Yunfan Jiang, Wesley Durbano, Brandon Huynh, Yu Fang, Danfei Xu, Ruohan Zhang, Li Fei-Fei, Linxi Fan, Bowen Wen, Ajay Mandlekar, and Yuke Zhu. SimFoundry: Modular and automated scene generation for policy learning and evaluation. *arXiv preprint arXiv:2606.28276*, 2026.

Martin Sedlacek, Pavlo Yefanov, Georgy Ponimatkin, Jai Bardhan, Simon Pilc, Mederic Fourmy, Evangelos Kazakos, Cees G. M. Snoek, Josef Sivic, and Vladimir Petrik. REALM: A real-to-sim validated benchmark for generalization in robotic manipulation. *arXiv preprint arXiv:2512.19562*, 2025.

Shuo Wang, Hanyuan Xu, Yingdong Hu, Fanqi Lin, and Yang Gao. A practical recipe towards improving sim-and-real correlation for VLA evaluation. *arXiv preprint arXiv:2606.10366*, 2026.

Qiyang Yan, Zihan Ding, Xin Zhou, and Adam J Spiers. Variable-friction in-hand manipulation for arbitrary objects via diffusion-based imitation learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 14284–14290, 2025.

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

Kaifeng Zhang, Shuo Sha, Hanxiao Jiang, Matthew Loper, Hyunjong Song, Guangyan Cai, Zhuo Xu, Xiaochen Hu, Changxi Zheng, and Yunzhu Li. Real-to-sim robot policy evaluation with gaussian splatting simulation of soft-body interactions. *arXiv preprint arXiv:2511.04665*, 2025.

Yixin Zhu, Zixiong Wang, Jian Yang, Jin Xie, Jingyi Yu, Jiayuan Gu, and Beibei Wang. Toward visually realistic simulation: A benchmark for evaluating robot manipulation in simulation. *arXiv preprint arXiv:2605.06311*, 2026.

A A PROBE TEST ON BEHAVIOR PRESERVATION

Recap. In Secs. 4 and 5, we use r and MMRV as our primary metrics, both computed from closed-loop success rates. This choice is motivated by practical real-to-sim policy evaluation settings (Li et al., 2024; Jain et al., 2026). Although success rates can distinguish selected policies with different behaviors in our behavior-wise comparison experiments (e.g., Fig. 6(b)), success rate alone is not a direct measure of behavior, since distinct behaviors can also yield similar success rates. Consequently, r and MMRV computed from success rates can indicate behavioral consistency when Π (the policy set) is behaviorally diverse, but provide only indirect evidence for sim-transferring and sim-overwriting, the two effects defined in Sec. 3.2.

This section complements these metrics with a simple classification probe that tests whether the behavior modes represented in the policies' $\mathcal{D}_R$ remain identifiable from rollouts in S. The probe examines trajectories more directly, although its input features capture only very limited behavioral information from each rollout. We therefore interpret the probe results jointly with the success-rate agreement metrics when analyzing the two effects.

Selected policies. We apply the classifier to a subset of the behavior-wise policies in Sec. 5: those whose 25 R-domain demonstrations cover exactly one behavior mode. At each $|\mathcal{D}_S|$, these selected policies differ in this mode and share the other training data, including $\mathcal{D}_S$. We use *left-hand* (left arm only) and *pass-left* (right-to-left transfer before binning) for PutBottles, omitting the unconstrained *fast* mode; for MugFlip, we use *right-hand* (right arm only), *handle* (lift by the handle), and *regrasp* (at least three regrasps per mug). Fig. 5 illustrates these behavior modes. We evaluate the final checkpoint (step 1,800), consistent with Secs. 5.1 and 5.2.

Classifier probe. We use a simple logistic regression classifier to test whether policy differences associated with $\mathcal{D}_R$ remain visible in rollouts. Each rollout's class label is the single behavior mode covered by the $\mathcal{D}_R$ of the policy that generates it. The classifier predicts this label from 14 features: the time-averaged positions of six arm joints and one gripper for each arm. It uses no images, object information, or task progress scores. For each behavior mode, data dose $|\mathcal{D}_S|$, and domain, we use 150 rollout episodes.

"$S \to S$" accuracy trains on 75 S rollouts per mode and tests on the remaining 75, measuring whether the policies remain distinguishable in S according to the behavior modes covered by their $\mathcal{D}_R$. As a control, "$R \to R$" accuracy trains on 75 R rollouts per mode and tests on the remaining 75, checking whether the policies remain distinguishable in R. "$R \to S$" accuracy trains on all 150 R rollouts per mode and tests on all 150 S rollouts per mode, indicating how similarly these behavior modes are distinguished across R and S under the probe. A decrease in $R \to S$ alone cannot determine whether an underlying cross-domain alignment remains but is not captured by the limited features, or whether sim-overwriting has decoupled behavior in S from behavior in R. We therefore interpret it jointly with $S \to S$: a concurrent decrease provides stronger evidence of sim-overwriting, whereas high $S \to S$ with low $R \to S$ indicates that the policies remain distinguishable in S but differently from R under the probe. Chance-level accuracy is 0.50 for PutBottles and 0.33 for MugFlip. This analysis uses one training seed among the three seeds in Sec. 5.

Table A.1: **Behavioral-gap classification and S-domain performance.** Mean SR^S averages over the selected policies and 150 rollouts per policy. Chance accuracy is 0.50 for PutBottles and 0.33 for MugFlip; shading ranges from lower (red) to higher (green) relative to chance.

Task	$\lvert\mathcal{D}_S\rvert$	Mean SR^S	$S \to S$	$R \to S$	$R \to R$
PutBottles	0	0.77	1.00	1.00	1.00
	3	2.77	0.83	0.71	0.93
	6	3.02	0.73	0.66	0.93
	15	3.62	0.66	0.58	0.95
MugFlip	0	0.14	1.00	0.99	0.99
	3	0.22	0.99	0.88	0.99
	6	0.20	0.95	0.79	0.93
	15	0.37	0.82	0.64	0.90

Table A.2: **Visual-gap classification and S-domain performance.** Mean SR^S averages over the three selected MugFlip policies and 150 rollouts per policy. Chance accuracy is 0.33; shading ranges from lower (red) to higher (green) relative to chance.

Setting	$\lvert\mathcal{D}_S\rvert$	Mean SR^S	$S \to S$	$R \to S$	$R \to R$
Same-task	0	0.34	1.00	0.94	0.97
	3	0.40	0.99	0.99	1.00
	15	0.61	0.92	0.73	0.98
	75	1.09	0.74	0.36	0.95
Hold-out	0	0.30	0.91	0.77	0.88
	3	0.22	0.85	0.66	0.92
	15	0.32	0.86	0.73	0.90
	75	0.44	0.87	0.67	0.90

Results and interpretation. For the behavioral gaps in Sec. 5.1, at $|\mathcal{D}_S| = 0$, the simple joint-position features are sufficient for the logistic regression classifier to distinguish the behavior modes with near-perfect $S \to S$, $R \to S$, and $R \to R$ accuracy (Table A.1). The small-dose gains in r and MMRV reported there are consistent with sim-transferring. As $|\mathcal{D}_S|$ increases, $R \to S$ decreases while $R \to R$ remains high, showing that distinctions learned in R become less recognizable in S under the probe. This decrease does not by itself establish that sim-transferring is absent or that behavior is fully decoupled across domains. The concurrent decrease in $S \to S$ provides stronger evidence of increasing sim-overwriting because the policies also become less distinguishable within S. Mean SR^S nevertheless increases from 0.77 to 3.62 bottles for PutBottles, consistent with sim-overwriting: shared $\mathcal{D}_S$ can improve performance in S while weakening the policy differences associated with $\mathcal{D}_R$ (Sec. 3.2).

For the visual gap in Sec. 5.2, in the same-task setting (Table A.2), $S \to S$ falls to 0.74 and $R \to S$ to 0.36 at the largest MugFlip dose, while $R \to R$ remains 0.95. Their concurrent decrease is consistent with sim-overwriting even though mean SR^S increases from 0.34 to 1.09 upright mugs. Increasing the dose of MugFlip demonstrations in $\mathcal{D}_S$ can therefore improve SR^S while making behavior in S less representative of the policy differences observed in R. In the hold-out setting, $S \to S$ remains 0.87 despite a lower $R \to S$ of 0.67, indicating that policy distinctions persist in S despite imperfect cross-domain recognizability under the probe. Together with the improved r and MMRV reported there, this pattern is consistent with sim-transferring from co-training on the held-out PutBottles task, without the same degree of sim-overwriting observed when $\mathcal{D}_S$ contains MugFlip demonstrations.

B Details of Policy Training

Common setup. All policies in Sec. 5 start from the public 2B-parameter `pi0_base` checkpoint[1] and are post-trained using AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.95$, $\epsilon = 10^{-8}$, and weight decay 0.1) with a peak learning rate of 5×10^{-5} and a global batch size of 256. We implement co-training weight $w = 0.1$ through dataset-level sampling: each training example is drawn from $\mathcal{D}_S$ with probability 0.1 and from $\mathcal{D}_R$ with probability 0.9, independent of the raw sizes of the two datasets.

Sim-to-sim recipes. The sim-to-sim policies use an action horizon of 16 and are trained for 1,800 steps with 200 warm-up steps followed by cosine decay to 10% of the peak learning rate. The seven PutBottles behavior combinations are: *left-hand*, *pass-left*, *fast*, *left-hand+pass-left*, *left-hand+fast*, *pass-left+fast*, and *left-hand+pass-left+fast*; the seven MugFlip combinations are *right-hand*, *handle*, *regrasp*, *right-hand+handle*, *right-hand+regrasp*, *handle+regrasp*, and *right-hand+handle+regrasp*. In the visual-gap hold-out setting, the MugFlip and PutBottles datasets in R share the total R-domain sampling weight (0.9) in proportion to their sizes.

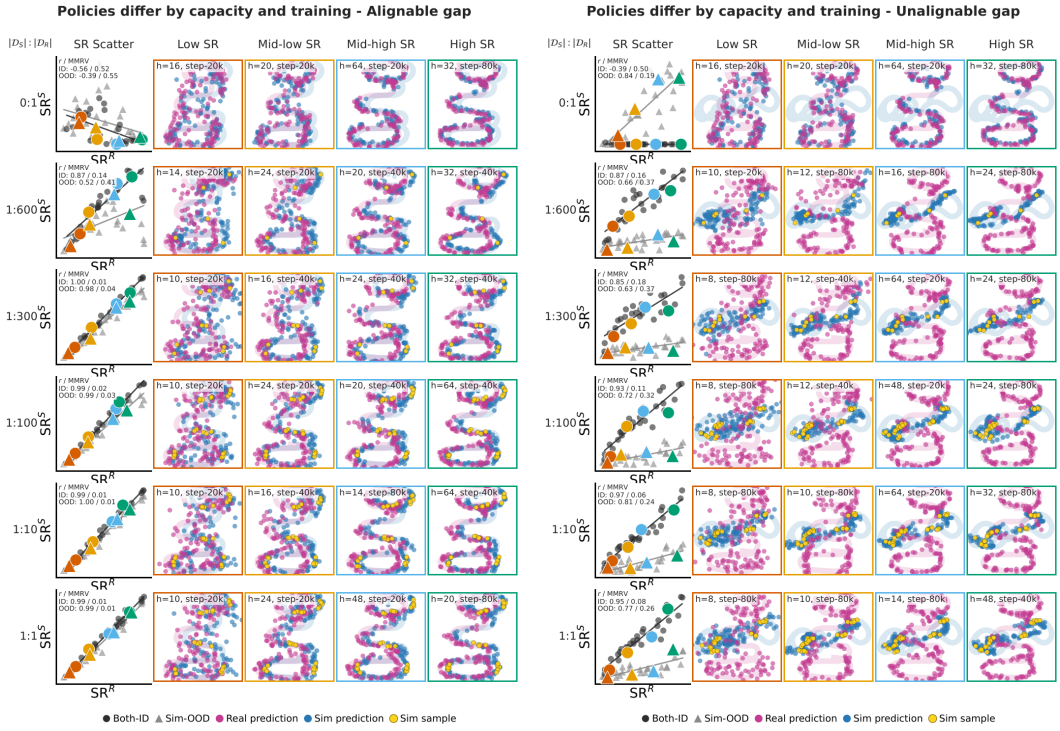
Real-to-sim recipes. The real-to-sim policies use an action horizon of 30 and are trained for 10,000 steps with 1,000 warm-up steps followed by linear decay to 20% of the peak learning rate. The full real-robot dataset ABC-130K (Allshire et al., 2026) contains demonstrations collected with

[1] `https://github.com/Physical-Intelligence/openpi`

two YAM gripper embodiments, Crank[2] and FlexPoint[3]; we sample only from the Crank subset to match our real-world evaluation setup. For cross-task co-training, we relabel every simulated MugFlip episode with the PutBottles instruction used by the real demonstrations rather than its native MugFlip instruction. Language therefore does not identify the dataset or task: the same instruction is paired with bottle-placement actions in real scenes and mug-flipping actions in simulated mug scenes, requiring the policy to distinguish them from visual context.

C Details of Controlled Synthetic Study

Supplementary visualizations. Figs. C.1 and C.2 expand Fig. 4 by showing success rates and representative policy predictions at six selected $|\mathcal{D}_S|{:}|\mathcal{D}_R|$ ratios under both alignable and unalignable gaps.



(a) Alignable gap (b) Unalignable gap

Figure C.1: **Policies differing by capacity and training (Sec. 4).** Each row corresponds to one $|\mathcal{D}_S|{:}|\mathcal{D}_R|$ ratio. The left column plots SR^S against SR^R across the policy set, using circles for Both-ID and triangles for Sim-OOD; the inset reports r and MMRV for both regions. The remaining columns show action predictions from four policies spanning low to high SR^R, with each label giving the network width and training checkpoint. Magenta and blue points denote predictions in R and S, respectively, and yellow points denote samples from $\mathcal{D}_S$.

[2] https://i2rt.com/products/yam-gripper-crank-shaft

[3] https://i2rt.com/products/flexpoint-adaptive-gripper-by-i2rt-robotics



(a) Alignable gap (b) Unalignable gap

Figure C.2: **Policies differing by behavior through $\mathcal{D}_R$ coverage (Sec. 4).** Each row uses the same SR scatter and four representative-policy columns as Fig. C.1. Here, all policies share the same capacity but omit different segments from $\mathcal{D}_R$, as indicated above each prediction panel. The panel below each prediction highlights that policy's $\mathcal{D}_R$ sampling region on the action band, exposing the training-data differences that produce behavioral variation across the policy set.

Model and training details. The conditional diffusion policy uses an MLP denoiser with three SiLU hidden layers of dimension h and a linear output layer. At each diffusion step t, we encode t as a 64-dimensional sinusoidal vector and concatenate it with the noisy two-dimensional action and three-dimensional observation as the MLP input. The MLP predicts the noise added to the action. We use 1,000 diffusion steps, with the diffusion noise parameter β_t increasing linearly from 10^{-4} to 0.02. All models are trained for 80k steps using Adam with a learning rate of 10^{-3} and a batch size of 256. Matching Sec. 4, we set the co-training weight to $w = 0.1$, so each batch contains 90% samples from $\mathcal{D}_R$ and 10% from $\mathcal{D}_S$. For policies differing by capacity and training, Π includes all $10 \times 3 = 30$ combinations of $h \in \{8, 10, 12, 14, 16, 20, 24, 32, 48, 64\}$ and checkpoints at steps 20k, 40k, and 80k. For policies differing by behavior, Π includes all $16 \times 3 = 48$ combinations of the $\mathcal{D}_R$-coverage variants and checkpoints at steps 20k, 40k, and 80k, with $h = 64$ throughout.

Data construction. For readers interested in reproducing the study, we provide details of the synthetic data generation. Each sample is parameterized by progress $u \in [0, 1]$ through five equal segments and position $v \in [-1, 1]$ across the action band. For domain $D \in \{S, R\}$, the observation is

$$o(u, v, D) = (2u - 1, v, o_D) + \epsilon, \qquad o_S = -0.5, \quad o_R = 0.5, \tag{3}$$

where the scalar o_D is the domain flag defined in Sec. 4. Training and evaluation observations use $\epsilon \sim \mathcal{N}(0, \mathrm{diag}(0.004^2, 0.004^2, 0.01^2))$. The centerline is the curve through the middle of the action band. Let $s_D(u)$ denote the distance traveled along this curve per unit increase in u, and let $\kappa_D(u)$ denote its signed curvature, with positive and negative values bending in opposite directions. The table specifies one nominal speed s per segment; $s_D(u)$ smoothly transitions between these values near segment boundaries. The curvature parameters are handled analogously. The tangent angle θ_D, centerline c_D, and ground-truth action map are

$$\begin{aligned} \theta_D(u) &= \int_0^u \kappa_D(t) s_D(t)\, dt, \\ c_D(u) &= b_D + \int_0^u s_D(t)\big(\cos\theta_D(t), \sin\theta_D(t)\big)\, dt, \\ \psi(2u - 1, v | o_D) &= c_D(u) + 0.04v\big(-\sin\theta_D(u), \cos\theta_D(u)\big). \end{aligned} \tag{4}$$

Here, the constant vector b_D positions the centerline in action space, and the vector multiplying $0.04v$ points perpendicular to the centerline. Consider a segment with endpoints a and b, local coordinate $q = (u - a)/(b - a) \in [0, 1]$, and parameters $(s, \bar{\kappa}, A, \nu, m, \rho)$. Let $\bar{\kappa}_D(u)$ smoothly interpolate the base-curvature values $\bar{\kappa}$ between adjacent segments. The full curvature within the segment is

$$\begin{aligned} \kappa_D(u) &= \bar{\kappa}_D(u) + A \sin^2(\pi q) \sin(2\pi\nu q) + \frac{\pi}{s} \sum_{\ell=0}^{m-1} (-1)^\ell g_\omega(u - u_\ell), \\ g_\omega(x) &= \frac{1 + \cos(\pi x/\omega)}{2\omega} \mathbf{1}\{|x| < \omega\}, \qquad \omega = \frac{0.1\pi\rho}{s}, \\ u_\ell &= a + \frac{\ell + 1/2}{m}(b - a), \qquad \ell = 0, \ldots, m - 1 \quad (m \geq 1). \end{aligned} \tag{5}$$

When $m = 0$, $\sum_{\ell=0}^{m-1} (-1)^\ell g_\omega(u - u_\ell) = 0$; otherwise, $(-1)^\ell$ alternates the turn direction and ρ controls the radius scale. Table C.3 lists the segment parameters. For the alignable gap, $s_S = s_R$ and $\kappa_S = \kappa_R$, while $b_S - b_R = (0.25, 0)$; therefore, $\psi(\cdot|o_S) - \psi(\cdot|o_R) = (0.25, 0)$. For the unalignable gap, S retains the same segment speeds but uses the second set of curvature parameters, making the action difference vary with u.

Within every segment, evaluation covers its full local coordinate $q \in [0, 1]$, $\mathcal{D}_R$ covers $q \in [0.25, 0.75]$, and $\mathcal{D}_S$ covers only $q \in [0.40, 0.60]$. Thus, $q \in [0, 0.25) \cup (0.75, 1]$ is Both-OOD, $q \in [0.25, 0.40) \cup (0.60, 0.75]$ is Sim-OOD, and $q \in [0.40, 0.60]$ is Both-ID. We allocate approximately the same number of training samples to each included region within a segment, sample uniformly inside each region, and sample v uniformly from $[-1, 1]$. We sweep $|\mathcal{D}_S| \in \{0, 5, 10, 15, 20, 30, 40, 300, 3000\}$. For policies differing by behavior, the 16 $\mathcal{D}_R$-coverage variants include one with all five segments, five that each omit one segment, and ten that each omit one pair of segments. Each retained segment contributes 600 samples to $\mathcal{D}_R$, so these variants contain 3,000, 2,400, or 1,800 samples when they omit zero, one, or two segments, respectively.

Table C.3: **Segment parameters for the closed-form ground-truth action map.** Each tuple is $(s, \bar{\kappa}, A, \nu, m, \rho)$. The R parameters also define S in the alignable gap before translation.

Segment	R and alignable S	Unalignable S
1	$(9, 0, 1.5, 2, 1, 2.4)$	$(9, 4, 0, 0, 0, 0)$
2	$(12, 0, 2, 3, 2, 2.0)$	$(12, 0, -2, 3, 2, 2.0)$
3	$(15, 0, 2.5, 4, 2, 2.2)$	$(15, -3, 0, 0, 0, 0)$
4	$(10, 0, 1.8, 2.5, 1, 2.3)$	$(10, 0, 1.8, 2.5, 3, 2.3)$
5	$(8, 0, 1.2, 2, 1, 2.6)$	$(8, 2, -1.2, 2, 0, 0)$

Evaluation. We use 1,500 evaluation samples from each domain. For a predicted action $\hat{a}$ and ground-truth action a, the per-sample score is $\exp(-\|\hat{a} - a\|_2^2 / 0.1^2)$. The SR^D of an evaluation region is the mean score over its samples from domain D.