PolaRiS: Scalable Real-to-Sim Evaluations for Generalist Robot Policies

Arhan Jain[1*], **Mingtong Zhang**[2*,α], **Kanav Arora**[1], **William Chen**[3], **Marcel Torne**[4], **Muhammad Zubair Irshad**[5], **Sergey Zakharov**[5], **Yue Wang**[6], **Sergey Levine**[3,8], **Chelsea Finn**[4,8], **Wei-Chiu Ma**[7], **Dhruv Shah**[2], **Abhishek Gupta**[1,5‡], **Karl Pertsch**[3,4,8‡]

[1]University of Washington, [2]Princeton University, [3]University of California, Berkeley, [4]Stanford University, [5]Toyota Research Institute, [6]University of Southern California, [7]Cornell University, [8]Physical Intelligence
Equal contribution., ‡Equal advising., αWork partially done at TRI.

A significant challenge for robot learning research is our ability to accurately measure and compare the performance of robot policies. Benchmarking in robotics is historically challenging due to the stochasticity, reproducibility, and time-consuming nature of real-world rollouts. This challenge is exacerbated for recent generalist policies, which has to be evaluated across a wide variety of scenes and tasks. Evaluation in simulation offers a scalable complement to real world evaluations, but the visual and physical domain gap between existing simulation benchmarks and the real world has made them an unreliable signal for policy improvement. Furthermore, building realistic and diverse simulated environments has traditionally required significant human effort and expertise. To bridge the gap, we introduce **Policy Evaluation and Environment Reconstruction in Simulation (PolaRiS)**, a scalable real-to-sim framework for high-fidelity simulated robot evaluation. PolaRiS utilizes neural reconstruction methods to turn short video scans of real-world scenes into interactive simulation environments. Additionally, we develop a simple simulation data co-training recipe that bridges remaining real-to-sim gaps and enables zero-shot evaluation in *unseen* simulation environments. Through extensive paired evaluations between simulation and the real world, we demonstrate that PolaRiS evaluations provide a much stronger correlation to real world generalist policy performance than existing simulated benchmarks. Its simplicity also enables rapid creation of diverse simulated environments. As such, this work takes a step towards distributed and democratized evaluation for the next generation of robotic foundation models.

Website: polaris-evals.github.io

1 Introduction

Recently, *generalist robotic policies* [1, 2, 3, 4, 5, 6, 7, 8], trained on diverse real-robot datasets, have shown an impressive degree of generalization to novel scenarios. However, with the advent of generalist policies comes a critical demand for large-scale empirical validation, since absolute guarantees are challenging to provide. To enable generalization, evaluations must be conducted across a large variety of tasks, environments and rollouts. While performing evaluations directly on real-world robots is the gold standard, this quickly becomes tedious, hard to control systematically, and impractical for individual groups to perform due to cost and labor constraints. Therefore, a more scalable paradigm for generalist robot policy evaluation becomes an emerging demand.

Simulation provides an appealing alternative—cheap, safe, fast, and unburdened by physical hardware, making large-scale policy evaluation far more feasible. However, despite considerable efforts [9, 10,
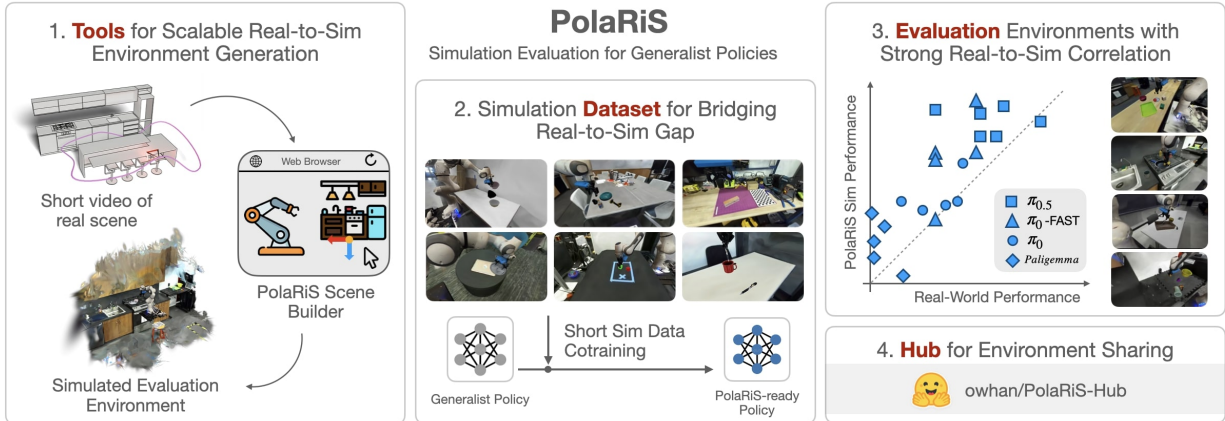
Figure 1 | Overview of PolaRiS. PolaRiS is a real-to-sim approach that turns short videos of real-world scenes into high-fidelity simulated environments for scalable robot policy evaluation. We demonstrate that co-finetuning with simulation data for a small number of iterations enables generalist policies to be evaluated accurately in *unseen* PolaRiS environments *out of the box*. As part of this work, we provide: (1) **tools** for quickly creating new PolaRiS environments, (2) an open-source simulation **dataset** for finetuning generalist policies for PolaRiS evaluation, (3) off-the-shelf **evaluation environments** with strong real-to-sim correlation, and (4) a **hub** for sharing new PolaRiS environments with the community.

11, 12, 13, 14, 15, 16], these platforms have not been well suited for evaluating the aforementioned *generalist* policies. To support a wide range of scenarios required to test generality, existing robot-simulation benchmarks typically sacrifice realism—including dynamics, visuals, and scene geometry—and thus fail to capture the visual complexity of the physical world [9, 11, 12, 14, 17]. Our experiments show that there remains a poor correlation between simulation and real world policy performance of state-of-the-art generalist policies. Moreover, as generalist policies get deployed across diverse environments, roboticists are often interested not only in *average* model performance *across* environments, but also in understanding a policy's performance within a *specific target environment*, which necessitates a tailored simulation of that scene. Constructing such an environment typically requires considerable manual effort. Effective simulated evaluation of generalist policies requires a new set of tools that allow us to rapidly construct high-fidelity targeted simulation environments, reflecting the diversity of real-world environments.

To this end, we introduce PolaRiS, a scalable real-to-sim framework for evaluating generalist robot policies. PolaRiS leverages neural reconstruction such as Gaussian splatting [18, 19] and 3D object generation [20] to construct simulated evaluation environments that closely match their real-world counterparts with high visual and geometric fidelity (Figure 1). This enables advanced generalist policies to perform various tasks in PolaRiS simulations out of the box. Additionally, our experiments show that lightweight finetuning of these policies with a small dataset of demonstrations in simulation can significantly boost real-to-sim correlation. Importantly, unlike existing simulation benchmarks, this simulation data is used only to align visual representations instead of teaching new tasks, and therefore does not have to be collected in the target environment or for the target task. As a result, once finetuned, policies can be evaluated *zero-shot* in *unseen* PolaRiS environments, facilitating rapid development of simulated evaluations that are customized for individual deployments yet shared broadly to assess generalist competence.

In our experiments across diverse real-world environments using the DROID platform [21], PolaRiS evaluations match the real-world performance of state-of-the-art generalist policies out of the box with an average Pearson correlation of $r = 0.9$. PolaRiS evaluations also strongly correlate ($r = 0.98$) with policy scores in RoboArena [22], the most comprehensive real-world benchmark for generalist

robot policies to date. As such, our experiments suggest that PoLaRiS takes a significant step towards rapid, scalable, simulated evaluation of generalist policies. We open-source all tools required to create, use, and share PoLaRiS evaluation environments, to foster the community-driven development of a shared suite of realistic simulation evaluation environments.

The contributions of this paper are as follows: (1) we introduce PoLaRiS, a scalable framework for creating high-fidelity simulated evaluation environments through neural scene reconstruction, (2) we develop a simple recipe for simulation data co-training that bridges the real-to-sim domain gap, ensuring that real-world generalist policies are accurately evaluatable in unseen simulation scenes out of the box, (3) we perform empirical evaluations across diverse real-world scenes demonstrating strong performance correlation for widely used generalist policies between PoLaRiS simulation and the real world, (4) we conduct detailed ablations demonstrating key design decisions for faithful simulated evaluation.

2 Related Work

Robot Policy Evaluation: Accurately evaluating the performance of robot policies is important for guiding research, yet faces challenges in practice: from reproducing initial environment conditions to robot wear and tear, and the tedium of supervising many hours of evaluations. Nevertheless, numerous real-robot benchmarks have been proposed over the years [22, 23, 24, 25]. With the rise of increasingly general policies that can perform many tasks across many environments [1, 2, 5, 6, 26], the challenges for performing comprehensive real-world evaluations are exacerbated. *Simulated* evaluations pose a promising alternative: initial conditions in simulation are perfectly repeatable, simulated robots do not wear out, and evaluations can be rapidly scaled through parallelization. As a result, many simulated robot benchmarks have been proposed [9, 11, 12, 13, 17]. These benchmarks are invaluable for algorithmic development. However, due to large visual and physical domain gaps between these simulated environments and the real world, they struggle to evaluate the performance of *generalist* policies, trained on real-world data. We instead design simulated environments that are capable of evaluating *real*-robot, generalist policies. They share the aforementioned benefits of simulated evaluations with respect to usability, reproducibility, and scalability, but are also reflective of the real-world performance of the evaluated generalist policies. More recently, advances in world models have enabled the use of action-conditioned video models as an alternative tool for robot policy evaluation [27, 28, 29, 30]. Like "classic" simulation, video model evaluations can be run at scale. Yet, our experiments demonstrate that today's (open) models struggle to reliably model physical interactions, and thus struggle to estimate policy performance. In contrast, our method leverages an accurate physics simulator together with high-fidelity neural rendering, making policy evaluation more faithful to real-world dynamics, reliable, efficient, and cost-effective.

Real-to-sim Policy Evaluation: A few recent works have recently explored whether increased visual and physical realism can improve the ability of simulated environments to be predictive of real-world policy performance: [31, 32] demonstrated that navigation policies can be evaluated in realistic reconstructions of indoor environments, and [33] demonstrated the same for autonomous driving policies. In robot manipulation, where policies closely *interact* with their environment, the requirements for physical and visual realism of simulated evaluation environments increase: SIMPLER [9] first showed that simulation can be used for evaluation of real-robot manipulation policies, but it fails to deliver strong real-world performance correlation for recent generalist policies [8]. Additionally, the "green-screening" approach used by [9] and follow-up works [34], which copy-pastes background pixels from real-robot images, does not support evaluation for robot setups with moving cameras. Unfortunately, this excludes most generalist policies [1, 2, 5, 26], which feature moving

wrist cameras attached to the robot's end-effector. Our work uses 3D Gaussian splatting techniques to create high-fidelity simulation environments. The resulting environments show strong real-to-sim performance correlation *and* support the evaluations of policies with wrist cameras.

Real-to-sim Environment Creation: A number of works have developed methods for digitizing real-world environments. Some approaches focus solely on room layout, and fill the scene by sampling from a large 3D object database [35], often lacking in visual realism. For higher visual quality, early approaches relied on specialized and expensive sensory equipment [36, 37], and more recently, learned models have been used to bypass such requirements and instead infer 3D models of environments from raw image or video inputs [32, 38]. In this context, Gaussian splatting has become a popular tool for inferring scene geometry and appearance from raw sensory observations [18, 19, 39, 40]. Multiple works have explored the use of Gaussian splatting techniques in robotics [41, 42, 43, 44, 45, 46]. While these works have shown that splatting can produce environments that can be used for augmenting the training of robot policies, in this work we investigate the utility of real-to-sim techniques in the context of robot policy *evaluation*. Most closely related to our work, [47, 48] used simulated environments created via Gaussian splatting for policy evaluation, but focused on evaluating narrow, single-task policies in environments seen in the training data. In contrast, comprehensive evaluation of modern *generalist* policies requires evaluations across a large number of tasks in unseen environments [22]. Our work is the first to develop a pipeline that enables evaluation of real-robot policies in *unseen* simulated environments, with strong performance correlation to real-world evaluations.

3 Preliminaries

The goal of this work is to evaluate *real-world* generalist robot policies in *simulated* environments such that the simulated evaluation faithfully reflects real-world performance. Let $\Pi = \{\pi_1, \ldots, \pi_N\}$ denote a set of policies, each of which can be executed both in the real world and in simulation. Let $\mathcal{E}_{\text{real}}$ denote the real-world environment distribution, and $\mathcal{E}_{\mathcal{S}}$ denote the corresponding simulated distribution constructed by a generative simulation pipeline $\mathcal{S}$.

For each policy $\pi_i \in \Pi$, we can define a metric of expected real-world performance (e.g., success rate) as $R_i = \mathbb{E}_{e \sim \mathcal{E}_{\text{real}}}\big[r(e, \pi_i) \big]$, where $r(e, \pi_i)$ denotes the episodic performance when executing policy π_i in environment e. Analogously, we define the policy's performance in simulation as $R_{\mathcal{S},i} = \mathbb{E}_{e \sim \mathcal{E}_{\mathcal{S}}}\big[r(e, \pi_i) \big]$. The goal of simulation-based evaluation is to design simulation environments $\mathcal{E}_{\mathcal{S}}$ such that the ranking and relative performance of policies in simulation closely approximate their real-world behavior. We follow prior work [9] to quantify the *faithfulness* of policy evaluation using two metrics.

(i) Pearson correlation coefficient (r): This measures the linear correlation between real and simulated performance:

$$r(R, R_{\mathcal{S}}) = \frac{\sum_{i=1}^{N}(R_i - \bar{R})(R_{\mathcal{S},i} - \bar{R}_{\mathcal{S}})}{\sqrt{\sum_{i=1}^{N}(R_i - \bar{R})^2}\sqrt{\sum_{i=1}^{N}(R_{\mathcal{S},i} - \bar{R}_{\mathcal{S}})^2}},$$

where $\bar{R}$ and $\bar{R}_{\mathcal{S}}$ are the mean real and simulated scores.

(ii) Mean Maximum Rank Violation (MMRV): This captures the severity of rank inversions between real and simulated evaluations. Intuitively, it measures the average worst-case real-world performance
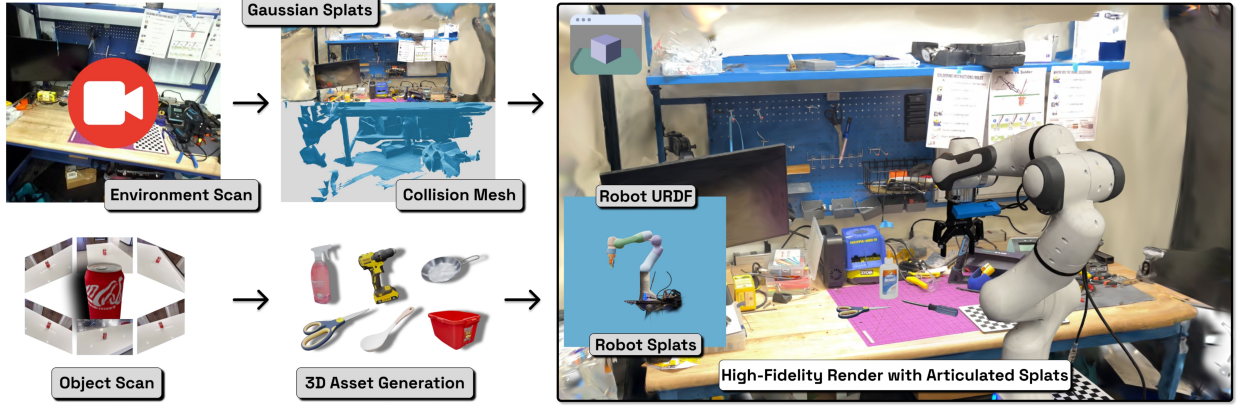
Figure 2 | Environment creation in PolaRiS. A user first scans in a real-world environment, robot and objects. They then use PolaRiS to create articulated Gaussian splat representations of the real-world environment that capture geometry and visual appearance of the environment. Finally, they compose the evaluation scene by combining scanned-in scene, objects and robot to use for policy evaluation.

gap between any two policies that are mis-ranked under simulation:

$$\text{MMRV}(R, R_{\mathcal{S}}) = \frac{1}{N} \sum_{i=1}^{N} \max_{1 \le j \le N} |R_i - R_j| \cdot \mathbf{1}\Big[(R_{\mathcal{S},i} < R_{\mathcal{S},j}) \ne (R_i < R_j)\Big].$$

The ideal simulation pipeline $\mathcal{S}$ has high Pearson correlation ($r \to 1$) and a low MMRV ($\to 0$). This indicates a well-calibrated simulation, in which relative performance ordering and absolute success trends of real-world policies are faithfully preserved.

2D Gaussian Splatting for Scene Reconstruction: To faithfully recover the scene geometry, we resort to recent neural reconstruction tools [18, 19, 39, 49, 50]. To achieve the best trade-off between high visual fidelity and accurate mesh geometry, we leverage 2D Gaussian Splatting (2DGS) [18], which represents a scene as a collection of oriented planar Gaussian disks embedded in 3D space, enabling improved geometric fidelity compared to other representations. A 2D Gaussian splat is represented as a tuple $g_j = (\boldsymbol{\mu}_j, \mathbf{R}_j, \mathbf{S}_j, \mathbf{c}_j, \alpha_j)$, where $\boldsymbol{\mu}_j \in \mathbb{R}^3$ is the disk center, $\mathbf{R}_j \in SO(3)$ encodes its orientation, $\mathbf{S}_j \in \mathbb{R}^{3\times3}$ defines its covariance in the disk plane, and $(\mathbf{c}_j, \alpha_j)$ denote its color and opacity. A full scene is represented as a collection of these Gaussian splats $\mathcal{D} = \{g_j\}_{j=1}^{M}$, optimized so that a differentiable renderer reproduces the input views on aggregate. Once optimized, the resulting representation supports both real-time photorealistic rendering and accurate mesh extraction, making it well suited for physics-based simulation of robotic policies. Additional implementation details are provided in Appendix A.1.

Generalist Robot Policies: While *generalist* policies is generally an ill-defined term, in this work we specifically use it to refer to a class of language-conditioned multi-task robot policies known as Vision-Language-Action models (VLAs) [1, 2, 5, 26]. VLAs are built on top of transformer based Vision-Language Models (VLMs), fine-tuned to produce robot actions conditioned on visual observations and textual instructions. We denote a generalist policy as $\pi_\theta : (O_{\text{vis}}, O_{\text{lang}}) \to \mathcal{A}$, where O_{vis} is the space of observations, O_{lang} is the space of language instructions, and $\mathcal{A}$ is the action space.

4 Scalable Simulation Evaluations with Real-World Video Scans

In this section, we describe POLARIS, our real-to-sim system for scalable creation of high-fidelity simulated environments (Figure 2). We structure our discussion around two components: first, we detail the construction pipeline that transforms a video recording of the real-world scene into an interactive simulation using 2DGS and automatic articulation; second, we address remaining real-to-sim discrepancies by proposing a lightweight simulation data co-training strategy to ensure well-correlated evaluations. Together, these components enable *zero-shot* evaluation of real-world generalist policies in previously unseen simulation environments.

4.1 Environment Generation with PolaRiS

To create an environment with POLARIS, we take a compositional approach. We first construct the visuals and geometry of the environments, then construct the visuals, geometry, and articulation of the robot, and finally add object assets into the scene. We outline these steps below:

Environment Creation via Reconstruction: We start by capturing a short (2–5 minutes) monocular video recording of the scene. We then run a reconstruction pipeline based on 2DGS [18] to reconstruct scene visuals and geometry. To facilitate compositional scene construction, we record separate video sequences for (i) the static background scene and (ii) the robot. This enables training of distinct Gaussian splat models for each component, which can later be composed and articulated in Euclidean space. During capture, the user slowly moves the camera to avoid motion blur while ensuring full coverage of all visible surfaces. To assist in automatically resolving global orientation and metric scale, a ChArUco board is placed in the environment (see Figure 2). This data acquisition process for a scene typically requires only 5–10 minutes. Following data capture, we estimate the camera parameters for all frames using COLMAP [51] and train a 2D Gaussian Splatting model $\mathcal{D} = \{g_j\}_{j=1}^{M}; g_j = (\boldsymbol{\mu}_j, \mathbf{R}_j, \mathbf{S}_j, \mathbf{c}_j, \alpha_j)$ for each recording by optimizing a standard photometric reconstruction loss as described in Appendix A.1.

To enable closed-loop robotic evaluation in a physics-based simulator, we require the geometric and photometric components of the reconstructed scene. We obtain a Truncated Signed Distance Field (TSDF) using a combination of rasterizing the splats and volumetric depth fusion [52]. An explicit mesh representation $\mathcal{M}$ is then extracted from the TSDF using marching cubes [53]. The resulting meshes are imported into IsaacSim [54] to enable realistic contact dynamics and object interactions, while the underlying Gaussian splats $\mathcal{D}$ provides the corresponding photometric renders for high-fidelity visual affects. The environment creation framework can be fully automated and typically completes within 30 minutes on an NVIDIA RTX 4090. Please refer to Appendix B for implementation details.

Automatic Articulation: While the environment remains static, the robot mesh and the corresponding Gaussian splats must be *articulated* for interactive evaluation. To enable this, we anchor subsets of Gaussian splat primitives to the robot's kinematic links, allowing them to move according to the robot's joint configuration. Let the robot have links $\{\mathcal{L}_1, \ldots, \mathcal{L}_L\}$ with joint configuration $\mathbf{q} \in \mathbb{R}^d$. Each link $\mathcal{L}_\ell$ is associated with a set of Gaussian primitives $\mathcal{G}_\ell = \{g_{\ell j}\}_{j=1}^{M_\ell}$ and a rigid-body transform $T_\ell(\mathbf{q}) = [R_\ell(\mathbf{q}) \mid \mathbf{t}_\ell(\mathbf{q})] \in \mathrm{SE}(3)$ obtained via forward kinematics. For a configuration $\mathbf{q}$, the position and covariance of each Gaussian is updated with the corresponding transform: $\boldsymbol{\mu}'_{\ell j} = R_\ell(\mathbf{q})\,\boldsymbol{\mu}_{\ell j} + \mathbf{t}_\ell(\mathbf{q}), \Sigma'_{\ell j} = R_\ell(\mathbf{q})\,\Sigma_{\ell j}\,R_\ell(\mathbf{q})^\top$. The articulated robot splat is then given by $\mathcal{G}(\mathbf{q}) = \bigcup_{\ell=1}^{L}\{(\boldsymbol{\mu}'_{\ell j}, \Sigma'_{\ell j}, \mathbf{c}_{\ell j}, \alpha_{\ell j})\}_{j=1}^{M_\ell}$, which can be rendered using the standard differentiable 2DGS pipeline. This enables interactive, photorealistic robots whose Gaussian splat components follow the true physical kinematics of the underlying mesh model. This allows the evaluation to be fully interactive in closed loop, while

6
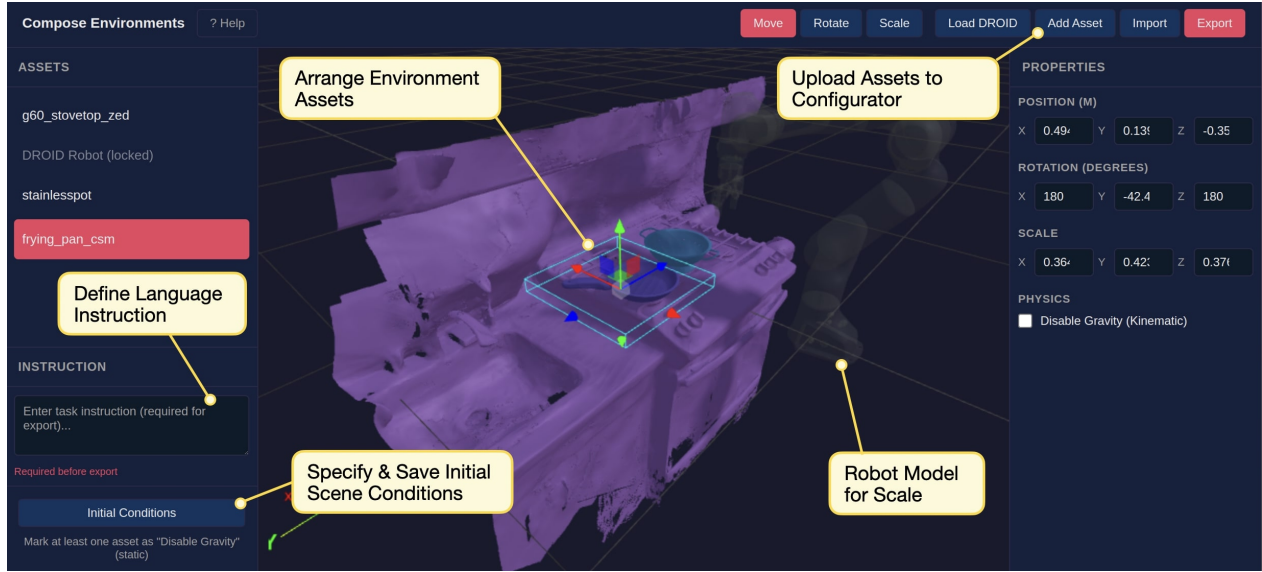
Figure 3 | Example of a scene being composed in our scene composition GUI. Users can easily import environment and object assets into the tool, automatically load the DROID robot, and save the simulation ready environment into a USD. The procedure typically takes less than 5 minutes.

retaining the visual realism of Gaussian splatting. The fully scanned and articulated robot model can be reused across simulation environments, making this a one-time process per robot system.

Object Creation with Generative Models: We find that directly extracting high-quality, watertight assets from real-world scenes is a time-consuming and tedious process that often requires depth-enabled hardware and precise registration tools. Although PoLaRıs is not tied to any specific object reconstruction methods, we employ the TRELLIS 3D image-to-3D generative model [20] for simplicity and flexibility. TRELLIS introduces a structured representation that enables mapping from a single or a few multiview 2D images to diverse 3D representations (such as Gaussian splats, radiance fields, and meshes) with high fidelity, while recovering unobservable object regions. With multiview images captured with a standard camera as inputs, we segment the object using SAM2 [55] and feed the segmented views into TRELLIS to obtain a structured latent representation. This latent is decoded into a Gaussian splat for appearance and an untextured mesh for geometry, after which appearance is baked onto the mesh to produce a fully textured object for simulation.

Composing simulated evaluation scenes. The final step is to assemble these components into an interactive evaluation scene using our Scene Composition GUI (Figure 3). Since all components are represented as Gaussian primitives, the environment, robot and objects can be composed and transformed in Euclidean space. Moreover, we set standard physics parameters for contact dynamics using default values from IsaacSim, and use manufacturer-provided defaults for the robot arm. Object mass parameters can be estimated by the users. Once the scene is assembled, it can be exported as a USD file for evaluation. Alongside scene composition, the user also defines a success criterion for scoring evaluation rollouts – e.g., whether a particular object is placed within a certain receptacle. The total human effort required for creating a new evaluation environment is typically **less than 20 minutes**, and the total wall time for creating a new evaluation environment, including training time of the Gaussian splats, is typically **less than one hour**. This is more scalable than prior simulated evaluation works [9], which require hand-defined construction of the environment geometry and extensive tuning of lighting and textures.
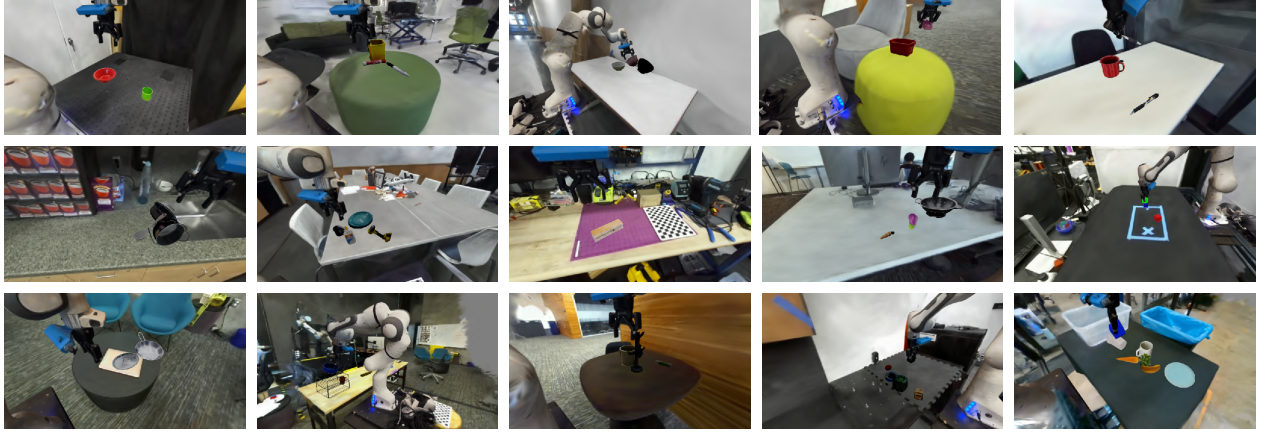
Figure 4 | PolaRiS simulated co-training dataset environments. We collect a small number of demonstrations in simulation and co-finetune policies with this data for improved real-to-sim evaluation correlation. Importantly, once finetuned, policies evaluated in POLARIS show strong real-to-sim correlation, even in *unseen* environments. New simulated evaluation environments are easily added without needing to collect additional demonstrations.

An important facet of the constructed simulation is the ability to provide policies with rich sensor inputs. As opposed to prior work [9, 34] which copied image backgrounds in 2D via green-screen, POLARIS reconstructs the entire scene in 3D, providing full 3D rendering and novel view synthesis. This allows us to easily render third person views from any angle and provide wrist camera inputs to the policy. This opens up a new class of simulation evaluations for generalist policies that prior real-to-sim evaluation frameworks that rely on background inpainting [9, 34] were unable to support.

4.2 Enabling High Correlation Policy Evaluation with PolaRiS using Co-Training

While the pipeline outlined above allows for the creation of high-fidelity environments, our empirical validation in Section 5 shows that the resulting environments do not reliably lead to high real-to-sim correlation out-of-the-box. We hypothesize, that this mainly stems from subtle domain gaps between the real world and simulation, such as minor lighting variations or missing inter-object shadows. Although these differences may appear negligible to human observers, they make a considerable difference in policy performance. To better understand this effect, we conducted hardware-in-the-loop experiments that use real-world dynamics but simulated visuals, which indicate that visual mismatch is indeed the dominant source of error (see Section C for details). To mitigate these gaps and improve real-to-sim correlation, we propose a simple, policy-agnostic simulation data co-training scheme.

Concretely, we collect a small simulation dataset via human teleoperation: $\mathcal{D}_{\mathcal{S}} = \{(o_t^{\mathcal{S}}, a_t^{\mathcal{S}}, p_t^{\mathcal{S}})\}_{t=1}^{M}$, consisting of simulated visual observations $o_t^{\mathcal{S}}$, expert actions $a_t^{\mathcal{S}}$, and proprioceptive states $p_t^{\mathcal{S}}$, gathered over a set of simulated co-training environments $\mathcal{E}_{\mathcal{S}}^{\text{train}} \subset \mathcal{E}_{\mathcal{S}}$. Given a pretrained generalist policy π_θ with original training data $\mathcal{D}_{\text{pre}}$, we update the policy parameters via *co-training* by sampling batches that mix simulation data with the original pretraining data:

$$\theta' \leftarrow \theta - \eta \nabla_\theta \mathbb{E}_{(o,a,p) \sim (1-\lambda)\,\mathcal{D}_{\text{pre}} + \lambda\,\mathcal{D}_{\mathcal{S}}} \left[\mathcal{L}_{\text{BC}}(\pi_\theta(o, p), a) \right],$$

where $\mathcal{L}_{\text{BC}}$ is a standard behavior cloning objective and $\lambda \in [0, 1]$ controls the ratio of simulation to pretraining samples in each batch. Intuitively, co-training with simulation data improves the policy's robustness to subtle visual gaps between simulation and the real world, leading to more reliable policy evaluation in simulation. In practice, we find that only a few hundred simulated episodes and a few hundred fine-tuning steps are sufficient to substantially improve real-to-sim correlation,

8

Figure 5 | Evaluation environments. Top: real-world environments, **Bottom**: PoLaRiS simulated evaluation replicas. We create high visual fidelity environments utilizing Gaussian Splats for environment reconstruction and TRELLIS for object asset generation.

typically completed in < 25 mins. This is orders of magnitude less data and compute than what is used during generalist pretraining. Importantly, because simulation co-training aims to refine visual representations instead of acquire new task-specific skills, the simulated demonstrations *are not required* to be collected in the target simulation environment(s) or task(s) $\mathcal{E}_S^{\text{test}}$. Instead, they can be collected apriori, on a completely separate set of environments and tasks - in our experiments co-training environments ($\mathcal{E}_S^{\text{train}}$) and evaluation environments ($\mathcal{E}_S^{\text{test}}$) are disjoint. As a result, the co-trained policies can be evaluated *zero-shot* in unseen simulation environments.

5 Experiments

Our experimental analysis aims to answer the following questions: (1) Does PoLaRiS enable zero-shot evaluation in *unseen* simulation environments with strong correlation to real-world performance? (2) Which design decisions affect zero-shot real-to-sim correlation? (3) Does PoLaRiS provide stronger real-to-sim evaluation correlation than current simulated evaluation tools in robotics?

5.1 Experimental Setup

Robot platform. We perform our experiments using the DROID [21] platform–one of the mostly widely used robot platforms for generalist policies [2, 22, 56]–which consists of a 7-DoF Franka Panda robot arm, and a set of two ZED RGB cameras with one mounted to the robot's wrist. While DROID offers both joint velocity and joint position action spaces, we focus on evaluating policies using joint *position* actions, which we found easier to simulate accurately since they suffer from less accumulating error.

Policies. We use a representative set of DROID VLAs from prior works for testing the efficacy of PoLaRiS as an evaluation tool: (1) π_0 [1] and π_0-**FAST** [56], which are diffusion-based and auto-regressive VLAs respectively, (2) **PaliGemma-binning**, an auto-regressive VLA with binning action tokenization from [56], and (3) $\pi_{0.5}$ [2], the current state-of-the-art VLA on DROID [22].

Tasks. To enable precise evaluation of real-to-sim correlations, we set up 6 *paired* real-world and simulated environments across two institutions (showcases in Figure 5). The environments span
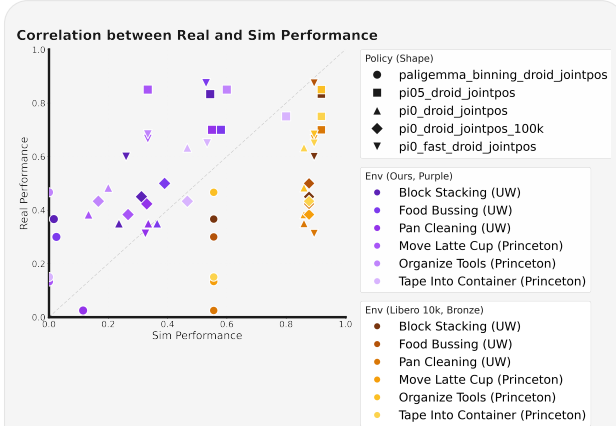
Figure 6 | Correlation of Evaluations between Real and Sim: PoLaRiS achieves strong real-to-sim correlation in *unseen* environments, accurately ranking real-world policy performance via simulation.
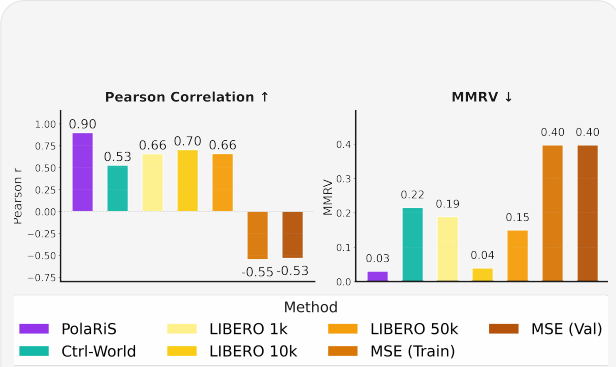
Figure 7 | Comparison of Evaluation Metrics: We compare Pearson correlation coefficient and Mean Maximum Rank Violation (MMRV). PoLaRiS achieves the best performance.

diverse visual scenes and a representative sample of rigid object manipulation tasks. We also create an additional 15 real-to-sim scenes through our pipeline, which we use to collect a dataset of around 350 simulation demonstrations for co-training (Section 4.2). We ensure that these "co-training environments" are *completely separate from our evaluation environments and tasks* and none of the scenes or objects are shared.
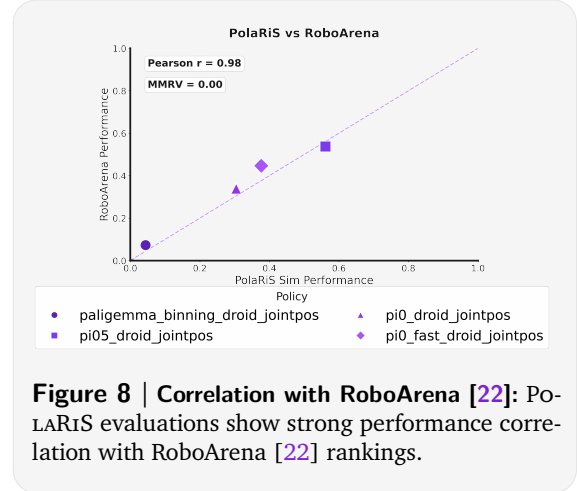
PolaRiS evaluation procedure. We finetune all policies for 1k steps on the co-training dataset using supervised learning, then run 50 rollouts per task in simulation to compute the policy's performance. Each task is automatically scored on a progress scale from $[0 \ldots 1]$. For details on all tasks, progress scales, and hyperparameters, please see Appendix C.

Comparisons. We compare our real-to-sim evaluation approach to existing methods used for offline policy evaluation: **MSE**, which computes training / validation set action MSE for each policy; **Libero-Score**, which evaluates policies on the Libero simulation benchmark [11] that is often used in VLA benchmarking [8, 57, 58, 59]; and **Video model evaluation**, which rolls out policies in the open-source **Ctrl-World** [29] video model and uses humans to score the rollouts. For Libero, to get the highest-quality evaluations possible, we finetune all policies on Libero-90, the largest of the Libero datasets, for 50k steps and evaluate on all 90 tasks, running 50 rollouts per task or a total of 4,500 evaluations per policy. Notably, this is a more thorough evaluation than done by most existing VLA papers using Libero, which typically evaluate on much smaller Libero splits [8, 57, 58]. Additionally, we evaluate multiple Libero-finetuned checkpoints (1k, 10k, 50k) and report results for the best-correlated checkpoint in Figure 6. As a result, the Libero-Score approach uses approximately 10x more finetuning and evaluation compute than our simulated evaluation approach. We note that existing real-to-sim evaluation approaches [9, 34], including SIMPLER, do not support evaluation of policies with wrist cameras, and thus cannot even be used to evaluate most state of the art VLA models.

Metrics. For each environment we run 20 real-world evaluation rollouts per policy to establish a ground-truth ranking of policy performance. As described in Section 3, we report Pearson correlation r and MMRV between the real-world and simulation performance.

5.2 Real-to-Sim Correlation in Unseen Environments

We report real-world correlation results for all tested approaches in Figure 6 and Figure 7. In line with existing literature [9], we find that MSE is a poor metric for real-world policy performance and shows low correlation. Libero-score too is poorly correlated with real-world performance, with nearly all tested policies scoring between 90 and 95% success in Libero, but spanning the full spectrum from high-performing to low-performing policies in the real world. Finally, we find that today's open-source video models are not yet reliable enough for policy evaluation in diverse scenes: in our experiments, Ctrl-World often produces heavy hallucinations during object interaction, making accurate scoring chal-

Figure 8 | **Correlation with RoboArena [22]:** PoLaRiS evaluations show strong performance correlation with RoboArena [22] rankings.

lenging and ultimately leading to clear policy mis-rankings (MMRV: 0.22). While recent works have shown promising video model evaluation results in narrower scene settings [29, 30], our experiments suggest that more advances are needed for these models to be widely applicable tools for robot policy evaluation in diverse real-world tasks.

In contrast, PoLaRiS achieves strong correlation results across all environments: in our experiments, differences in PoLaRiS scores closely reflect real-world performance differences, and rank the performance of policies accurately. Importantly, this result does not only hold in terms of *average* performance, but holds *for each individual environment*: the *worst-case* correlation across all six tested environments is $r = 0.81$. See individual environment results in Figure 13. In contrast to hand-designed sim benchmarks, PoLaRiS's pipeline can be used to estimate both: average performance of generalist policies and performance in specific deployment scenarios.

Furthermore, we find that PoLaRiS scores exhibit strong correlation to RoboArena [22] real-world evaluation scores (see Figure 8). We report correlation to the average progress scores reported as part of the RoboArena evaluations. While we find that scores correlate well, we emphasize that the simulated evaluations in our experiments only capture a small subset of the capabilities tested in RoboArena—particularly those involving non-rigid body dynamics, soft-body interactions, and complex contact behaviors that remain challenging for current simulators. Thus, we believe that PoLaRiS can serve as a useful tool for cheaply and rapidly estimating VLA performance during research and development, but *does not* entirely replace real-world evaluation.

5.3 Ablation Studies and Analysis

In this section, we investigate the importance of different components of our PoLaRiS framework.

Visual fidelity. To ablate the impact of the visual fidelity of simulated environments on real-to-sim correlation, we create versions of our simulated evaluation scenes with different levels of visual quality (see Table 5), using ray-tracing instead of Gaussian splatting, and removing background and foreground textures. We report real-to-sim correlation after co-training all policies with the respective simulated dataset in Figure 9, left. First, we find that after co-training, all simulation variants can provide decently correlated results, suggesting that the simulation data co-training can alleviate some visual sensitivity observed in prior works [9]. Yet, we find that PoLaRiS with Gaussian splatting rendering leads to the strongest correlation. While these results suggest visual fidelity is not the *sole* driver of high-quality simulated evaluations, the comparison to Libero-Score (Section 5.2) underlines the benefit of real-to-sim evaluation techniques compared to hand-crafting low-fidelity simulated
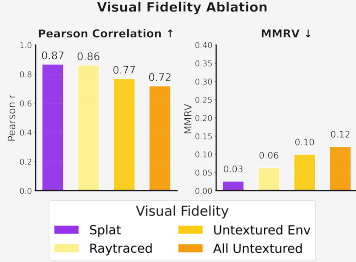
Figure 9 | Visual fidelity ablation: High-fidelity rendering is key for strong real-to-sim correlation: evaluations with texture-less simulation environments or raytraced rendering underperform Gaussian splatting based rendering evaluations, even after simulation data co-training.
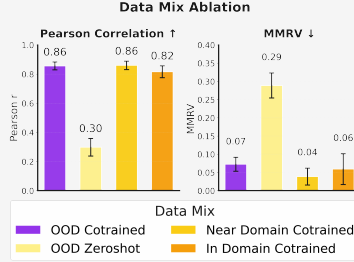
Figure 10 | Co-training dataset ablation: Simulation data co-training is key for strong real-to-sim evaluation correlation, but importantly, once co-trained with our simulated demonstration dataset, POLARIS provides strong correlation even for unseen ("OOD") simulation environments.
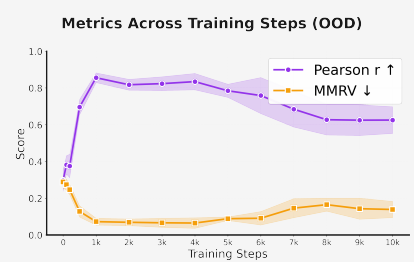
Figure 11 | Finetuning steps ablation: Real-to-sim correlation across five finetuning seeds. POLARIS reliably achieves strong real-to-sim correlation after just a few hundred steps of simulation data co-finetuning. Finetuning policies for too long reduces real-to-sim correlation.

benchmarks: real-to-sim evaluation scans *real-world* environments and thus allows evaluation in environments that are closer to the training distribution.

Co-training dataset. We evaluate the impact of different simulation co-training data mixtures. In our original experiments, we collected data in 15 held-out simulation environments. For this experiment, we additionally collect data in the 3 target environments, on both, the target tasks and non-target tasks. We then compare 3 settings: "**OOD**" uses only data from the 15 held-out environments (the setting from the previous sections); "**near-domain**" additionally uses data from *non-target tasks*, collected in the *target environments*; "**in-domain**" combines the data from the held-out environments with data collected on the *target tasks* in the *target environments*. We also compare to directly evaluating the pre-trained checkpoints without any simulation data co-training ("**no-cotrain**") (see Table 4). We report real-to-sim correlation in Figure 10. First, we find that the base VLA models ("no-cotrain") achieve non-trivial performance, but correlation is too low to accurately rank policy performance, underlining the importance of simulation co-training. Comparing the different simulation data mixes, we find that the "OOD" data mix (ours) leads to approximately comparable correlations to in-domain data. This underlines that the function of the simulation data is primarily to bridge subtle visual gaps between simulation and real-world environments, for which OOD scenes are sufficient. Notably, this result make POLARIS evaluations very scalable: we can add new evaluation environments *without* collecting targeted simulation data and *without* needing to finetune generalist policies for every new scenario. Such "out of the box" evaluation in unseen simulation environments makes sharing and accumulating of targeted simulated evaluations across the community feasible, like it is commonplace e.g. for LLM research today. Lastly, we find that "near-domain" data leads to slightly stronger correlations than target task demonstrations, possibly because the latter lead to quick overfitting.

Fine-tuning steps. We evaluate the robustness of our pipeline to different number of finetuning steps. Concretely, we run five finetuning runs for all policies, training for 10k steps each. We then evaluate the mean and variance of Pearson correlation and MMRV across all checkpoints (see Figure 11). Our experiments suggest that 1k steps of co-finetuning consistently lead to strong real-to-sim correlation, but even just a few hundred steps of finetuning can significantly improve correlation results. Figure 11 shows that too much finetuning can lead to overfitting and worsen correlation.

6 Discussion

In this work, we introduce PoLaRiS, a system for developing scalable evaluation benchmarks in simulation from real-world videos. PoLaRiS allows for personalized environment creation in simulation, corresponding to particular real-world environments. With a small amount of policy co-training on simulation data, these simulated environments can be made directly usable for policy evaluation. We perform a thorough empirical investigation, demonstrating strong correlation between simulated and real evaluations, out of the box even in unseen simulation environments. These simulation environments can serve as a useful tool for scalable evaluations of generalist policies. As advances are made in the field of 3D reconstruction and generation, the PolaRiS framework will benefit from higher fidelity environments and greater ease of use.

While a promising step for policy evaluation, there are still several points of improvement for PoLaRiS. Going beyond Gaussian splat rendering to richer diffusion based renderings can further bridge the real-to-sim visual gaps. Moreover, PoLaRiS performs a relatively simplistic identification of system dynamics. We can draw on a rich literature of system identification [60] to better identify the robot and environment dynamics. Lastly, we can consider hybrid simulation techniques, merging elements of world modeling with physics-based simulation to generate more scalable simulated evaluations, including more complex objects and scene configurations.

Acknowledgments

References

[1] Physical Intelligence. π_0: A vision-language-action flow model for general robot control, 2024.

[2] Physical Intelligence. $\pi_{0.5}$: a vision-language-action model with open-world generalization, 2025.

[3] Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE, 2024.

[4] Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis Armstrong, Ashwin Balakrishna, Robert Baruch, Maria Bauza, Michiel Blokzijl, et al. Gemini robotics: Bringing ai into the physical world. *arXiv preprint arXiv:2503.20020*, 2025.

[5] Gemini Robotics Team. Gemini robotics 1.5: Pushing the frontier of generalist robots with advanced embodied reasoning, thinking, and motion transfer, 2025.

[6] TRI LBM Team. A careful examination of large behavior models for multitask dexterous manipulation. *arXoiv preprint*, 2025.

[7] 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*, Delft, Netherlands, 2024.

[8] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[9] 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 Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard, editors, *Conference on Robot Learning, 6-9 November 2024, Munich, Germany*, volume 270 of *Proceedings of Machine Learning Research*, pages 3705–3728. PMLR, 2024.

[10] Pushkal Katara, Zhou Xian, and Katerina Fragkiadaki. Gen2sim: Scaling up robot learning in simulation with generative models. In *IEEE International Conference on Robotics and Automation, ICRA 2024, Yokohama, Japan, May 13-17, 2024*, pages 6672–6679. IEEE, 2024.

[11] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: benchmarking knowledge transfer for lifelong robot learning. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[12] 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. *CoRR*, abs/2406.02523, 2024.

[13] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J. Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics Autom. Lett.*, 5(2):3019–3026, 2020.

[14] Oier Mees, Lukás Hermann, Erick Rosete-Beas, and Wolfram Burgard. CALVIN: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics Autom. Lett.*, 7(3):7327–7334, 2022.

[15] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine.

Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pages 1094–1100. PMLR, 2020.

[16] Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Wensi Ai, Benjamin Martinez, et al. Behavior-1k: A human-centered, embodied ai benchmark with 1,000 everyday activities and realistic simulation. *arXiv preprint arXiv:2403.09227*, 2024.

[17] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In Leslie Pack Kaelbling, Danica Kragic, and Komei Sugiura, editors, *3rd Annual Conference on Robot Learning, CoRL 2019, Osaka, Japan, October 30 - November 1, 2019, Proceedings*, volume 100 of *Proceedings of Machine Learning Research*, pages 1094–1100. PMLR, 2019.

[18] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *Proceedings of the 2024 ACM SIGGRAPH Conference on Computer Graphics and Interactive Techniques*. Association for Computing Machinery, 2024.

[19] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. In *ACM SIGGRAPH 2023 Conference Proceedings*. Association for Computing Machinery, 2023.

[20] Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21469–21480, 2025.

[21] DROID Team. DROID: A large-scale in-the-wild robot manipulation dataset. In *Robotics: Science and Systems XX, Delft, The Netherlands, July 15-19, 2024*, 2024.

[22] Pranav Atreya, Karl Pertsch, Tony Lee, Moo Jin Kim, Arhan Jain, Artur Kuramshin, Clemens Eppner, Cyrus Neary, Edward Hu, Fabio Ramos, et al. Roboarena: Distributed real-world evaluation of generalist robot policies. In *Proceedings of the Conference on Robot Learning (CoRL 2025)*, 2025.

[23] Berk Çalli, Aaron Walsman, Arjun Singh, Siddhartha S. Srinivasa, Pieter Abbeel, and Aaron M. Dollar. Benchmarking in manipulation research: The YCB object and model set and benchmarking protocols. *CoRR*, abs/1502.03143, 2015.

[24] Michael Ahn, Henry Zhu, Kristian Hartikainen, Hugo Ponte, Abhishek Gupta, Sergey Levine, and Vikash Kumar. ROBEL: robotics benchmarks for learning with low-cost robots. In Leslie Pack Kaelbling, Danica Kragic, and Komei Sugiura, editors, *3rd Annual Conference on Robot Learning, CoRL 2019, Osaka, Japan, October 30 - November 1, 2019, Proceedings*, volume 100 of *Proceedings of Machine Learning Research*, pages 1300–1313. PMLR, 2019.

[25] Zhiyuan Zhou, Pranav Atreya, You Liang Tan, Karl Pertsch, and Sergey Levine. Autoeval: Autonomous evaluation of generalist robot manipulation policies in the real world. *CoRR*, abs/2503.24278, 2025.

[26] NVIDIA. Gr00t n1: An open foundation model for generalist humanoid robots, 2025.

[27] Julian Quevedo, Percy Liang, and Sherry Yang. Evaluating robot policies in a world model. *arXiv preprint arXiv:2506.00613*, 2025.

[28] Wei-Cheng Tseng, Jinwei Gu, Qinsheng Zhang, Hanzi Mao, Ming-Yu Liu, Florian Shkurti, and Lin Yen-Chen. Scalable policy evaluation with video world models. *arXiv preprint arXiv:2511.11520*, 2025.

[29] Yanjiang Guo, Lucy Xiaoyang Shi, Jianyu Chen, and Chelsea Finn. Ctrl-world: A controllable generative world model for robot manipulation. *arXiv preprint arXiv:2510.10125*, 2025.

[30] Gemini Robotics Team, Coline Devin, Yilun Du, Debidatta Dwibedi, Ruiqi Gao, Abhishek Jindal, Thomas Kipf, Sean Kirmani, Fangchen Liu, Anirudha Majumdar, Andrew Marmon, Carolina Parada, Yulia Rubanova, Dhruv Shah, Vikas Sindhwani, Jie Tan, Fei Xia, Ted Xiao, Sherry Yang, Wenhao Yu, and Allan Zhou. Evaluating gemini robotics policies in a veo world simulator, 2025.

[31] Gunjan Chhablani, Xiaomeng Ye, Muhammad Zubair Irshad, and Zsolt Kira. Embodiedsplat: Personalized real-to-sim-to-real navigation with gaussian splats from a mobile device, 2025.

[32] Alejandro Escontrela, Justin Kerr, Arthur Allshire, Jonas Frey, Rocky Duan, Carmelo Sferrazza, and Pieter Abbeel. Gaussgym: An open-source real-to-sim framework for learning locomotion from pixels, 2025.

[33] Ze Yang, Yun Chen, Jingkang Wang, Sivabalan Manivasagam, Wei-Chiu Ma, Anqi Joyce Yang, and Raquel Urtasun. Unisim: A neural closed-loop sensor simulator. *arXiv preprint arXiv:2308.01898*, 2023.

[34] Yash Jangir, Yidi Zhang, Kashu Yamazaki, Chenyu Zhang, Kuan-Hsun Tu, Tsung-Wei Ke, Lei Ke, Yonatan Bisk, and Katerina Fragkiadaki. Robotarena ∞: Scalable robot benchmarking via real-to-sim translation, 2025.

[35] Matt Deitke, Rose Hendrix, Luca Weihs, Ali Farhadi, Kiana Ehsani, and Aniruddha Kembhavi. Phone2proc: Bringing robust robots into our chaotic world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[36] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, and Vladlen Koltun. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[37] Fei Xia, Amir R Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: Real-world perception for embodied agents. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9068–9079, 2018.

[38] Marcel Torne, Anthony Simeonov, Zechu Li, April Chan, Tao Chen, Abhishek Gupta, and Pulkit Agrawal. Reconciling reality through simulation: A real-to-sim-to-real approach for robust manipulation. *Arxiv*, 2024.

[39] Guikun Chen and Wenguan Wang. A survey on 3d gaussian splatting, 2025.

[40] Muhammad Zubair Irshad, Mauro Comi, Yen-Chen Lin, Nick Heppert, Abhinav Valada, Rares Ambrus, Zsolt Kira, and Jonathan Tremblay. Neural fields in robotics: A survey. *arXiv preprint arXiv:2410.20220*, 2024.

[41] Justin Yu, Kush Hari, Karim El-Refai, Arnav Dalil, Justin Kerr, Chung-Min Kim, Richard Cheng, Muhammad Z. Irshad, and Ken Goldberg. Persistent object gaussian splat (pogs) for tracking human and robot manipulation of irregularly shaped objects. *ICRA*, 2025.

[42] Jad Abou-Chakra, Lingfeng Sun, Krishan Rana, Brandon B. May, Karl Schmeckpeper, Maria Vittoria Minniti, and Laura Herlant. Real-is-sim: Bridging the sim-to-real gap with a dynamic digital twin for real-world robot policy evaluation. *CoRR*, abs/2504.03597, 2025.

[43] Justin Yu, Kush Hari, Kishore Srinivas, Karim El-Refai, Adam Rashid, Chung Min Kim, Justin Kerr, Richard Cheng, Muhammad Zubair Irshad, Ashwin Balakrishna, et al. Language-embedded gaussian splats (legs): Incrementally building room-scale representations with a mobile robot. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 13326–13332. IEEE, 2024.

[44] Mohammad Nomaan Qureshi, Sparsh Garg, Francisco Yandún, David Held, George Kantor, and Abhisesh Silwal. Splatsim: Zero-shot sim2real transfer of RGB manipulation policies using gaussian splatting. In *IEEE International Conference on Robotics and Automation, ICRA 2025, Atlanta, GA, USA, May 19-23, 2025*, pages 6502–6509. IEEE, 2025.

[45] Prithwish Dan, Kushal Kedia, Angela Chao, Edward Weiyi Duan, Maximus Adrian Pace, Wei-Chiu Ma, and Sanjiban Choudhury. X-sim: Cross-embodiment learning via real-to-sim-to-real. *CoRR*, abs/2505.07096, 2025.

[46] Justin Yu, Letian Fu, Huang Huang, Karim El-Refai, Rares Andrei Ambrus, Richard Cheng, Muhammad Zubair Irshad, and Ken Goldberg. Real2render2real: Scaling robot data without dynamics simulation or robot hardware, 2025.

[47] Guangqi Jiang, Haoran Chang, Ri-Zhao Qiu, Yutong Liang, Mazeyu Ji, Jiyue Zhu, Zhao Dong, Xueyan Zou, and Xiaolong Wang. Gsworld: Closed-loop photo-realistic simulation suite for robotic manipulation. *arXiv preprint arXiv:2510.20813*, 2025.

[48] 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.

[49] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision (ECCV)*, pages 405–421, 2020.

[50] Lior Yariv, Peter Hedman, Christian Reiser, Dor Verbin, Pratul P. Srinivasan, Richard Szeliski, Jonathan T. Barron, and Ben Mildenhall. Bakedsdf: Meshing neural sdfs for real-time view synthesis. *arXiv*, 2023.

[51] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, 2016.

[52] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '96)*, pages 303–312. ACM, 1996.

[53] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '87)*, pages 163–169. ACM, 1987.

[54] NVIDIA. Isaac Sim.

[55] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloé Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross B. Girshick, Piotr Dollár, and Christoph Feichtenhofer. SAM 2: Segment anything in images and videos. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

[56] Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025.

[57] Huang Huang, Fangchen Liu, Letian Fu, Tingfan Wu, Mustafa Mukadam, Jitendra Malik, Ken Goldberg, and Pieter Abbeel. OTTER: A vision-language-action model with text-aware visual feature extraction. *CoRR*, abs/2503.03734, 2025.

[58] Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *CoRR*, abs/2502.19645, 2025.

[59] Jason Lee, Jiafei Duan, Haoquan Fang, Yuquan Deng, Shuo Liu, Boyang Li, Bohan Fang, Jieyu Zhang, Yi Ru Wang, Sangho Lee, Winson Han, Wilbert Pumacay, Angelica Wu, Rose Hendrix, Karen Farley, Eli VanderBilt, Ali Farhadi, Dieter Fox, and Ranjay Krishna. Molmoact: Action reasoning models that can reason in space, 2025.

[60] Taeyoon Lee, Jaewoon Kwon, Patrick M. Wensing, and Frank C. Park. Robot model identification and learning: A modern perspective. *Annu. Rev. Control. Robotics Auton. Syst.*, 7(1), 2024.

[61] Physical Intelligence. openpi, 2025.

Appendix

A   Additional Details for Preliminaries

A.1   2DGS

Given a set of calibrated RGB images $\{I_i\}_{i=1}^N$ with known camera poses $\{T_i\}_{i=1}^N$, 2DGS optimizes a set of primitives

$$\mathcal{D} = \left\{ (\boldsymbol{\mu}_j, \ \mathbf{R}_j, \ \mathbf{S}_j, \ \mathbf{c}_j, \ \alpha_j) \right\}_{j=1}^M,$$

where $\boldsymbol{\mu}_j \in \mathbb{R}^3$ is the disk centre, $\mathbf{R}_j \in SO(3)$ defines the disk's orientation (via two tangent vectors) in world space, $\mathbf{S}_j = \mathrm{diag}(s_{j,u}, s_{j,v}, 0)$ encodes the elliptical variances in the disk plane, $\mathbf{c}_j \in \mathbb{R}^3$ is the view-dependent colour and $\alpha_j \in [0, 1]$ the opacity. Rendering is carried out by computing a perspective-correct ray–splat intersection for each disk, projecting the elliptical footprint into image space, and then performing differentiable alpha-blending to produce a rendered image $\hat{I}_i(\mathcal{D}, T_i)$.

The training objective is then given by a photometric loss plus geometry-regularisation terms:

$$\mathcal{L}(\mathcal{D}) = \sum_{i=1}^N \left\| \hat{I}_i(\mathcal{D}, T_i) - I_i \right\|_2^2$$
$$+ \ \lambda_{\mathrm{dist}} \, \mathcal{R}_{\mathrm{dist}}(\mathcal{D}) \ + \ \lambda_{\mathrm{norm}} \, \mathcal{R}_{\mathrm{norm}}(\mathcal{D}).$$

Here $\mathcal{R}_{\mathrm{dist}}$ is a *depth* regularizer that encourages each disk to align closely along the viewing ray to reduce thickness and multi-view inconsistency, and $\mathcal{R}_{\mathrm{norm}}$ is a *normal* regularizer that minimizes the discrepancy between disk normals and the gradient of the rendered depth map. Once optimized, the disks give rise to both high-quality novel-view synthesis and accurate geometry that can be extracted (e.g., via meshing) and used for simulation of robotic policies. While closely related to 3D Gaussian splatting [19], the 2DGS algorithm more faithfully captures object geometries while maintaining high fidelity visuals. As we show in our experimental analysis, this ensures better zero-shot evaluation performance for robot policies trained in the real world.

B   Environment Generation with PolaRiS

B.1   Reconstruction

To ensure all reconstructions are expressed in a consistent world coordinate frame, we define the ChArUco board origin as the canonical base frame $\mathcal{F}_0$. For each video, we get recovered camera poses $\{T_i\}_{i=1}^N$ for almost all video images in an arbitrary frame from COLMAP, and a subset of these frames will have measured camera coordinates $\{C_i\}_{i=1}^M; M \leq N$ with respect to the ChArUco board. Using the COLMAP's *model aligner* functionality [51], we are able to transform the collective camera poses' arbitrary scale and orientation into an accurately scaled and oriented set of camera poses in the ChArUco board canonical base frame.

B.2   Scene Composition

After training a Gaussian Splat and extracting a mesh, we turn all assets into USD format. For raytraced experiments, we baked the vertex colors from 2DGS mesh extraction into a texture. All scanned objects are also turned into USD format. Finally, scenes are put together using our scene composition GUI (Fig. 3).
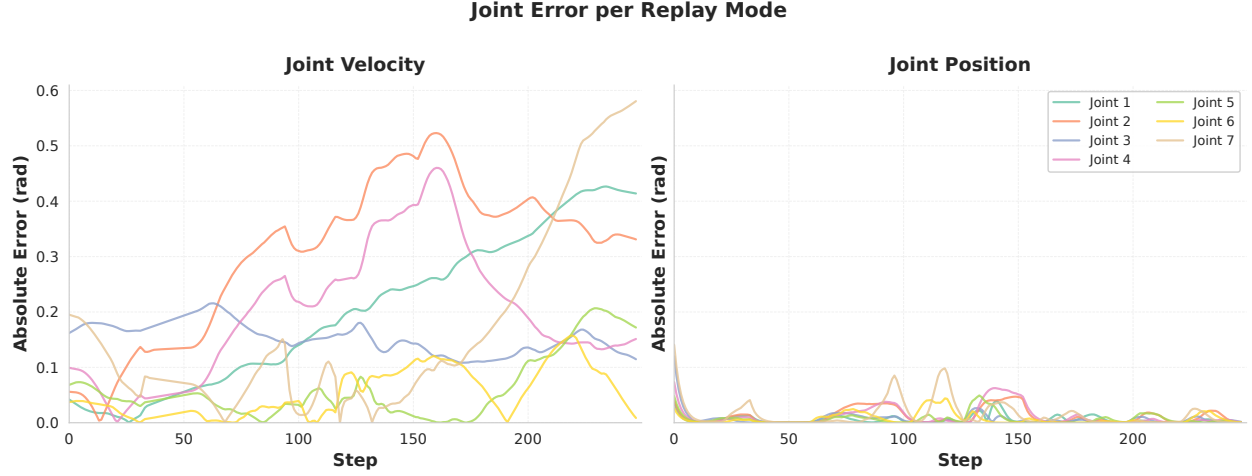
Figure 12 | Comparison of absolute joint error overtime in open loop replay between joint velocity and joint position control modes.

C Evaluation Tasks, Cotraining Hyperparameters, and Datasets

C.1 Tasks

We evaluate 6 tasks each in distinct environments. Each task has its own criteria for partial progress scores so that we can evaluate policies with a more dense metric than sparse, binary success. All criteria are autonomously graded in simulation using privileged state information. For real world evaluations, a human watches each rollout and grades them according to the same rubric used in simulation. The criteria for each task in Table 1. Each task may have a different score range, so scores and normalized between 0-1 to make allow for comparisons across environments.

C.2 Cotraining

For all of our experiments involving cotraining policies, we collected data in simulation environments using a twin of the real DROID platform [21]. To restrict behaviors from deviating significantly from the original DROID dataset, we adopted the same data collection scheme that was used in DROID. Data in simulation was collected with an Oculus Quest 2, and we setup our environments such that they could interface directly with the VR controller code shipped with the DROID code release. After data is collected, it's processed into the RLDS format, exactly matching the structure of the original DROID RLDS. Finally, policies were trained using the open-source DROID policy training code provided in *openpi* [61], with a few modifications in dataloading. All cotraining was done with a mixture of **90%** real DROID data and **10%** simulated DROID data. Additional hyperparameters for cotraining can be found in Table 2 and Table 3. For our analysis of how many finetuning steps is needed to achieve best correlations, we ran independent finetuning runs with unique seeds (1-5). For each checkpoint we do 50 independent rollouts with the same initial conditions. All cotraining (both simulation and real world) is done on joint position action data. We find joint position policies to transfer much better from real to sim, with the intuition of joint position policies avoiding compounding error. We demonstrate this intuition in Fig. 12 using open loop replay comparing joint velocity and joint position policies.

Task Name	Instruction	Step-by-Step Rubric
Food Bussing	Put all the foods in the bowl	1. Reach for any food item 2. Lift the food item 3. Place the first food item into the bowl 4. Place the second food item into the bowl
Block Stacking	Place and stack the blocks on top of the green tray	1. Reach for any block 2. Lift the block 3. Place any block on the tray 4. Place the second block on the tray 5. Second block is stacked on the first
Pan Cleaning	Use the yellow sponge to scrub the blue handle frying pan	1. Reach for the yellow sponge 2. Lift the sponge 3. Move sponge into the pan
Move Latte Cup	Put the latte art cup on top of the cutting board	1. Reach for the latte cup 2. Pick up the cup 3. Put it on top of the cutting board
Organize Tools	Put the scissor into the large container	1. Reach for the scissor 2. Pick up the scissor 3. Put it into the container in large size
Tape Into Container	Put the tape into the container	1. Reach for the tape 2. Pick up the tape 3. Put it into the container

Table 1 | Task prompts and the corresponding step-by-step rubric for the three evaluation tasks. Scores are normalized between 0 and 1 for comparability across environments.

C.3   Datasets

In our cotraining experiments, we performed some ablations on the differences in evaluation ability with different data mixtures, varying the closeness of the data to our test-time distribution. Specifically, our datasets were "*OOD*", "*near-domain*", and "*in-domain*". Details on examples and makeup of each dataset can be found in Table 4. We also performed ablations on how much visual fidelity mattered for faithful evaluation of policies. In this experiment, we had 4 datasets used for cotraining: *splat*, *raytraced*, *untextured environment*, and *untextured everything*. Examples of each can be seen in Table 5. All visual fidelity ablations were performed with *OOD* data, with only differences in rendering.

D   Additional Experiments

D.1   Hardware-in-the-loop Experiments

In order to determine what was causing the low correlation and performance in our real-to-sim environments, we set out to isolate and test the issues of visual gap and dynamics gap. We accomplished
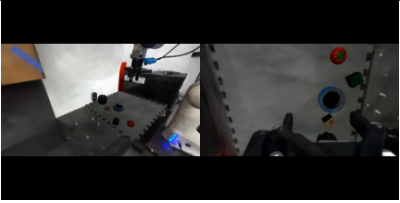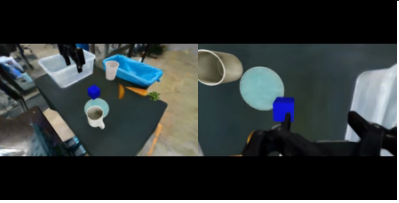
Hyperparameter	Parameter Value
Action Space	Joint Position (8-DoF robot)
Learning Rate Schedule	Cosine Decay
Warmup Steps	1,000
Peak Learning Rate	5×10^{-5}
Decay Steps	1,000,000
Decay Learning Rate	5×10^{-5}
Batch Size	128

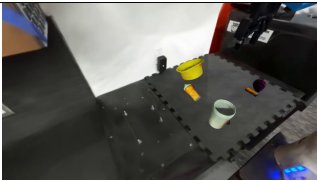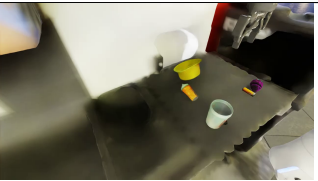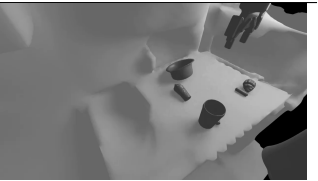**Table 2** | Shared hyperparameters used for all policies and experiments during co-training.

Model	Details
pi05	Action dim = 32, Action horizon = 15, no tokenizer
pi0_fast	Action dim = 8, Action horizon = 10, FAST Tokenizer
pi0	Action dim = 32, Action horizon = 10, no tokenizer
pi0_100k	Action dim = 32, Action horizon = 10, no tokenizer
paligemma_binning	Action dim = 8, Action horizon = 15, Binning Tokenizer

Table 3 | Policy-specific configurations for all DROID cotrained models. For LIBERO experiments, the only modification is changing action dimension to 7 for **pi0_fast** and **paligemma_binning**.

this by running rollouts in the real world and selectively switching out parts of the rollout with simulation. For testing whether simulation dynamics was the issue, we fed the policy real observations and commanded joint positions with blocking control using the achieved simulation joint positions for every step (control frequency of 15 Hz for DROID). To test the visual gap, we matched up a scene between real and simulation and fed the policy simulation renders at each step. For control, we executed predicted actions on the real robot for 1/15 seconds and commanded the achieved real joint positions to the simulation to create the effect of real dynamics with simulation observations – evaluating success in simulation. All hardware-in-the-loop experiments were performed with the π_0 FAST DROID Joint Position policy. We found that introducing simulated dynamics with real visuals led to no drop in performance; however, introducing simulation visuals with real dynamics led to about a 25% drop in performance measured across 3 tasks.

In-Domain	Near-Domain	OOD
Prompt: Put all the foods in the bowl	Prompt: Move the cubes from left to right	Prompt: Put the dishes into the plastic container
Environments: 18 Instructions: 21 Trajectories: 476 Timesteps: 79,077	Environments: 18 Instructions: 21 Trajectories: 416 Timesteps: 65,489	Environments: 15 Instructions: 18 Trajectories: 326 Timesteps: 56,041

Table 4 | Details on the makeup of the different cotraining data mixes for ablating what kind of data is necessary to keep simulation evaluation faithful.

Splat	Raytraced	Untextured Environment	Untextured Everything

Table 5 | Examples of different visual fidelity renderings.
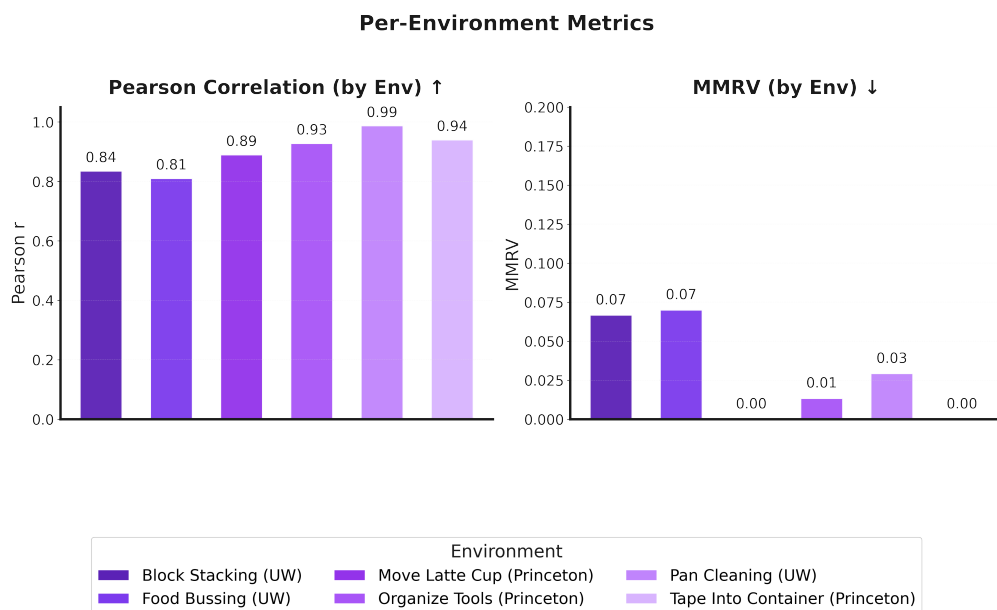
Figure 13 | Correlation metrics for each individual environment, demonstrating PolaRiS's ability to faithfully evaluate policies in targeted environments for deployment in addition to general performance.

5