Real2Render2Real: Scaling Robot Data Without Dynamics Simulation or Robot Hardware

Justin Yu[1,*], **Max Letian Fu**[1,*], **Huang Huang**[1], **Karim El-Refai**[1], **Rares Andrei Ambrus**[2], **Richard Cheng**[2], **Muhammad Zubair Irshad**[2], **Ken Goldberg**[1]

[1]University of California, Berkeley, [2]Toyota Research Institute

Abstract: Scaling robot learning requires vast and diverse datasets. Yet the prevailing data collection paradigm—human teleoperation—remains costly and constrained by manual effort and physical robot access. We introduce **Real2Render2Real (R2R2R)**, a novel approach for generating robot training data without relying on object dynamics simulation or teleoperation of robot hardware. The input is a smartphone-captured scan of one or more objects and a single video of a human demonstration. R2R2R renders thousands of high visual fidelity robot-agnostic demonstrations by reconstructing detailed 3D object geometry and appearance, and tracking 6-DoF object motion. R2R2R uses 3D Gaussian Splatting (3DGS) to enable flexible asset generation and trajectory synthesis for both rigid and articulated objects, converting these representations to meshes to maintain compatibility with scalable rendering engines like IsaacLab but with collision modeling off. Robot demonstration data generated by R2R2R integrates directly with models that operate on robot proprioceptive states and image observations, such as vision-language-action models (VLA) and imitation learning policies. Physical experiments suggest that models trained on R2R2R data from a single human demonstration can match the performance of models trained on 150 human teleoperation demonstrations. Project page: https://real2render2real.com

Keywords: Robot Datasets, Imitation Learning, Data Augmentation

1 Introduction

> The great power of general purpose methods ... [is that they] continue to scale with increased computation.
>
> — Richard Sutton, *The Bitter Lesson* (2019)

Robotics has long benefited from computational scalability—methods like probabilistic planning, trajectory optimization, and reinforcement learning have driven significant progress in agile locomotion [1, 2, 3, 4, 5, 6, 7]. Dexterous manipulation, however, presents unique challenges: it requires fine-grained visual perception that is tightly coupled with robot control and kinematics to interact with objects and alter the environment. Many systems address this by explicitly separating perception from planning and control, achieving strong performance in structured environments [8, 9, 10, 11], especially when assumptions about scene geometry, object placement, and sensing modalities hold. Yet such pipelines often rely on task-specific perception modules and carefully controlled environments, limiting flexibility in more unstructured, dynamic, or visually diverse settings.

In the hope of addressing open-world manipulation tasks, inspired by large language models (LLMs) and vision-language models (VLMs) [12, 13, 14, 15], recent efforts have explored end-to-end generalist robot policies [16, 17, 18, 19, 20, 21, 22, 23, 24, 25]—models that learn directly from raw sensory input and promise capabilities like language instruction following, task transfer, and

[*]Equal Contribution

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

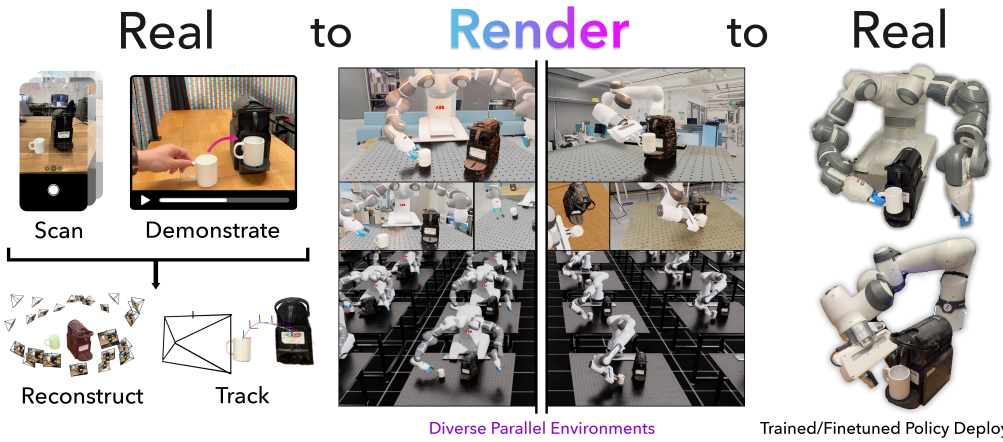

Figure 1: Real2Render2Real generating robot training data for the task of "Put the Mug on the Coffee Maker". R2R2R takes as input a multi-view object scan and a monocular human demonstration video. R2R2R then synthesizes diverse, domain-randomized robot executions through parallel rendering and outputs paired image-action data for policy training. This pipeline enables scalable learning across tasks and embodiments without teleoperation or object dynamics simulation.

in-context learning. Yet training such models at scale remains limited by data: the largest human teleoperation datasets are over 100,000× smaller than the corpora used to train frontier LLMs and VLMs [26, 27], and are constrained by the cost, speed, and embodiment-specific nature of human teleoperated data collection.

Other vision-language subfields have faced similar data scarcity—and overcome it through *computational* data generation. Structure-from-motion, detection, and depth pipelines now routinely produce pseudo-labels to bootstrap large models; for instance, SpatialVLM synthesizes two billion spatial-reasoning QA pairs [28], while RAFT [29], DUSt3R [30], MonST3R [31], Zero-1-to-3 [32], and MVGD [33] all rely on pseudo ground-truth derived from multi-view geometry pipelines (e.g., COLMAP [34]) to supervise dense 3D prediction tasks. These successes suggest an analogous question for robotics:

Can we computationally scale robot vision-action data – while not requiring dynamics simulation or human teleoperation – to train robot learning models?

Prior efforts have turned to physics-based simulation, where trajectories are synthesized via reinforcement learning or motion planning in virtual environments [35, 36, 37]. While modern simulators offer high throughput and support large-scale parallelization, they face several fundamental limitations: many commonly used simulators fail to satisfy basic Lagrangian mechanics, such as conservation of energy or momentum [38]; accurately modeling complex object interactions often demands extensive parameter tuning and hand-crafting of contact properties [39]; generating high-quality, compliant, and intersection-free assets for simulation remains labor-intensive, as collision modeling requires careful handling of geometry, friction, and deformation [40, 41]. R2R2R avoids these challenges by discarding dynamics: instead of simulating forces or contacts, we directly set object and robot poses per frame using the IsaacLab package [42] purely as a photorealistic, parallelized rendering engine by setting all objects as kinematic rather than dynamic bodies. This approach respects robot kinematics while avoiding the complexities of contact modeling, naturally aligning with contemporary vision-based policies trained from RGB images and proprioceptive inputs.

We introduce **Real2Render2Real (R2R2R)**, a pipeline for generating large-scale synthetic robot training data from a smartphone object scan and a human demonstration video. R2R2R scales *trajectory diversity* while preserving visual accuracy: it extracts 6-DoF object part trajectories from the video and generates corresponding robot executions via inverse kinematics under randomized object initializations. Starting from a multi-view scan, it reconstructs 3D object geometry and appearance, supports both rigid and articulated objects via part-level decomposition, and uses 3D Gaussian Splatting to produce mesh assets. The resulting trajectories include robot proprioception, end-effector actions, and paired RGB observations rendered under varied lighting, camera pose,

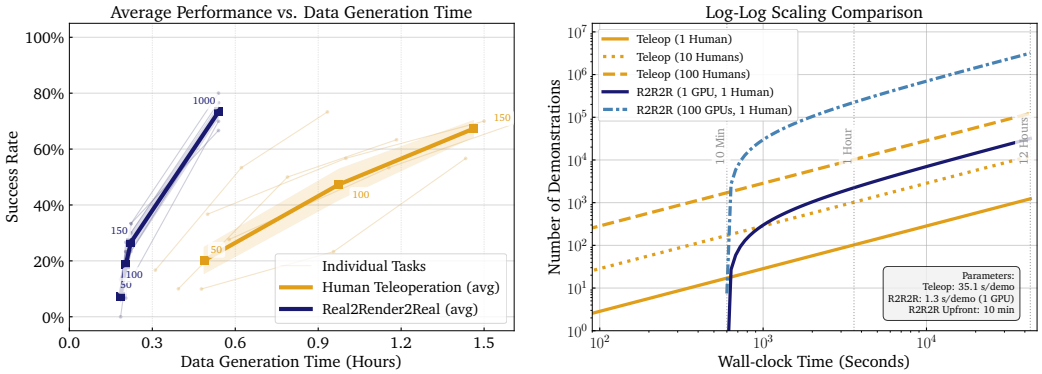

Figure 2: Data Generation Efficiency and Average Policy Performance Across Manipulation Tasks. (Left) Performance visualization displaying both task-specific outcomes (faint background lines) and cross-task averages (bold lines with error shading) for policies trained on real (1 human teleoperator) vs. synthetic data (1 human, 1 GPU). The points labeled by demonstration count (50-1000) highlight the scaling in performance and R2R2R's significant throughput advantage, with individual task trajectories illustrating the variance across different manipulation scenarios. **(Right)** Log-log scale comparison showing data generation throughput between R2R2R (1-100 GPUs) and human teleoperation (1-100 operators) over a 12-hour period. R2R2R needs an upfront time of 10 minutes for human to scan the objects, demonstrate the task, reconstruct the objects and track their trajectory, where subsequentially no human is involved. On a single NVIDIA 4090 GPU, on average, trajectories will be generated at 27x the speed of a single human teleoperator without needing robot hardware.

and object placement—making them directly compatible with modern imitation learning policies such as vision-language-action models and diffusion models. By eliminating the need for dynamics simulation or robot hardware, R2R2R enables accessible and scalable robot data collection, allowing anyone to contribute by capturing everyday object interactions with a smartphone.

This paper makes three contributions. First, we present *Real2Render2Real (R2R2R)*, a novel framework that synthesizes diverse, physically grounded observation–action pairs using only smartphone-captured videos: a multi-view object scan and a human demonstration video—without requiring dynamics simulation or robot hardware. Second, we demonstrate that this data is compatible with modern vision-language-action (VLA) and imitation learning policies, including both transformer-based and diffusion-based architectures that operate from RGB and proprioceptive input. Third, we show that policies trained on R2R2R-generated data based on one human demonstration can match the performance of those trained on 150 human teleoperation demonstrations, across 1,050 physical robot evaluations, while requiring significantly less time to generate.

2 Related Work

Robot Data Collection Paradigms. Scaling robot learning has traditionally relied on two paradigms: data from industrial deployments and data from human teleoperation. Industrial robot logs [43, 44, 45] scale with production throughput but are often task- and embodiment-specific. In contrast, teleoperation datasets [46, 47, 48, 49, 50] offer greater visual and task diversity but remain bottlenecked by human effort and real-time collection. At the same time, the rise of generalist robot policies [16, 19, 51, 17, 18, 20, 21, 22, 23, 24, 25]—capable of performing diverse manipulation tasks from raw observations—has amplified the need for scalable, diverse, and high-quality training data. Yet the scale of current robot datasets remains orders of magnitude below that of their vision and language counterparts [26, 27].

Procedural Robot Data Generation. To address the challenge of robot data scaling, many works have studied procedural data generation to automate robot data collection for pre-defined tasks. Many works [52, 53, 54, 55, 56, 57] use pre-defined motion primitives, optionally with a perception module, to automate data collection using a real robot, with automatic scene reset. While reducing human interventions, they still require robot hardware for data collection, limiting scalability. More recently, simulation data generation has emerged as a scalable alternative to real-world collection,

	Tele-Op Free	RL Free	Phys. Engine Free	Robot Agnostic	One-to-Many Trajectories	Articulated Objects
CASHER [39]	✗	✗	✗	✓	✓	✓
RoboVerse [38]	✗	✗	✗	✓	✗	✓
RoboGSim [61]	✗	✓	✓	✗	✗	✗
RoVi-Aug [62]	✗	✓	✓	✓	✗	✓
Video2Policy [37]	✓	✗	✗	✓	✓	✓
MimicGen [58]	✗	✓	✗	✓	✓	✗
DexMimicGen [63]	✗	✓	✗	✓	✓	✗
Phantom [64]	✓	✓	✓	✓	✗	✓
DemoGen [65]	✗	✓	✓	✗	✓	✗
AR2-D2 [66]	✓	✓	✓	✓	✗	✓
Real2Render2Real	✓	✓	✓	✓	✓	✓

Table 1: Comparison of Robot Data Generation Methods. *Real2Render2Real* requires no teleoperation, eliminates reliance on reward engineering, reinforcement learning, or accurate asset physics modeling, and provides object-centric demonstrations directly extracted from a video where humans interact with the objects. It also supports various robot embodiments, and generates multiple varied trajectories from a single demonstration.

parallelizing data generation without physical robot hardware. Utilizing the privileged information from the simulator, Mahler et al. [10] generates large and diverse data for robot grasping. Katara et al. [35], Wang et al. [36] generate large-scale robot data in simulation using reinforcement learning, trajectory optimization, and motion planning. MimicGen [58] synthesizes diverse simulations from a single human tele-operation sequence, combining motion planning and trajectory replaying. Despite efforts to bridge the sim-to-real gap through domain randomization [59], improved asset and scene generation [35, 36], the resulting simulation data often exhibit significant visual discrepancies from real-world observations, requiring co-training on real data to enable effective transfer [60].

Real2Synthetic Data Generation. To mitigate this visual domain gap, some works augment and repurpose real RGB data instead of synthesizing it from scratch. For example, Chen et al. [62] employs generative models for inpainting robot embodiment features into real images, enabling data synthesis for robots with different morphologies. However, such approaches still require human teleoperation for initial demonstrations. Further, they lack the ability to generate additional diverse trajectories beyond the provided demonstrations. Similarly, Lepert et al. [64], Duan et al. [66] use hand-pose tracking to guide inpainted robot end-effector trajectories from human demonstrations. While these methods reduce the need for direct teleoperation, they typically generate only a single trajectory per video and lack support for computationally-scaled trajectory diversity. In contrast, R2R2R can generate multiple, diverse robot trajectory renderings and action rollouts from a single human demonstration. Policies trained solely on R2R2R-generated data achieve comparable real-world performance with those trained on human teleoperation data.

Real2Sim2Real Data Generation. To generate diverse trajectories from a single demonstration while bridging the sim-to-real gap, many methods follow a Real2Sim2Real paradigm—using real-world observations to build simulated environments to train policies deployed back in the real-world. Prior work [67] shows that tuning physics parameters can reduce dynamics mismatch, but large visual domain gaps often still necessitate test-time perception modules. Recent methods [37, 68, 38, 63] reduce this visual gap by constructing digital twins or "digital cousins"[69] from real scans. These approaches vary in their reliance on teleoperation, simulation, and trajectory diversity—but many still depend on teleoperated demos, handcrafted rewards, or accurate physics models, limiting scalability. For example, DexMimicGen[63] uses fixed simulation assets; RoboVerse [38] supports only rigid objects; and RialTo [70] and CASHER [39] require manual articulation labeling and reward engineering. While Video2Policy [37] avoids reward tuning via vision-language models, it still requires test-time object detection due to visual mismatches. These pipelines also rely on physics engines, which demand high-fidelity meshes for collision checking and extensive tuning. RoboGSim [61] avoids simulation but lacks support for trajectory diversity from a single demo. In contrast, R2R2R addresses these limitations by: (1) extracting object trajectories from human videos, (2) segmenting object parts automatically, (3) rendering realistic observations to remove reliance on

Figure 3: 3D Gaussian Splat Object Reconstructions with part-level segmentations derived from feature-based grouping. Objects are reconstructed and segmented into rigid or articulated components using GARField [74].

additional perception models, (4) eliminating the need for collision modeling and detailed meshes, and (5) generating diverse trajectories from a single demonstration.

3 Assumptions

We assume objects are rigid or articulated, and manipulated on a table-top setup under quasi-static conditions. Object surfaces are assumed to exhibit low specularity to support robust geometry reconstruction and visual feature extraction. We also assume that during human demonstrations, objects are not placed in configurations that lead to complete mutual occlusion. Approximate camera poses relative to the robot in the physical setup are assumed to be available, enabling the generation of observations from nearby viewpoints during data collection. Learned policies take RGB image observations and robot's proprioceptive states as inputs.

4 Method

Real2Render2Real (R2R2R) is a data generation pipeline for synthesizing diverse robot demonstration data consisting of RGB-action pairs from a single human demonstration and multi-view object scan. R2R2R consists of three primary stages: (1) **real-to-sim asset and trajectory extraction**, where rigid or articulated object geometry and part trajectories are extracted from real-world smartphone captures; (2) **augmentation**, where object initialization is randomized and object motion trajectories are interpolated if appropriate; and (3) **parallelized rendering**, where diverse photorealistic robot executions are generated using IsaacLab [42], scalable with the amount of available GPU memory and the numbers of GPUs.

4.1 Real-to-Sim Asset Extraction

We extract 3D object assets from smartphone scans using a two-stage process inspired by [71, 72]. First, we reconstruct object geometry and appearance using 3D Gaussian Splatting (3DGS) [73], then apply GARField [74] to segment the scene into semantically meaningful parts by lifting 2D masks into 3D. This enables both object-level and part-level decomposition, including articulated components. To support mesh-based rendering, the resulting Gaussian groups are converted into textured triangle meshes via an extended version of [75].

4.2 Real-to-Sim Trajectory Extraction

Given a smartphone video of a human manipulating the scanned objects, R2R2R extracts the 6-DoF part motion of the object and its parts using 4D Differentiable Part Modeling (4D-DPM) introduced in [71]. Each 3DGS object part is embedded with pre-trained DINO features, enabling part pose optimization through differentiable rendering. We extend [71]'s implementation to track single or multiple rigid objects, as well as articulated ones, from demonstration videos.

While there are many alternative pipelines that convert real images into 3D assets, we adopt 3DGS-to-mesh conversion for two key reasons: (1) it enables background–foreground segmentation and part decomposition via 3D grouping [74], which is critical for extracting object part-specific trajectories from monocular human demonstrations; and (2) it maintains compatibility with both 4D-DPM trajectory reconstruction and instanceable mesh-based rendering engines, allowing seamless integration

into our large-scale rendering pipeline. This process requires no fiducials or hardware beyond a smartphone camera, making it well-suited for scalable and accessible real-to-sim data generation.

Interpolation Methods for Object Trajectory Diversity: A key contribution of Real2Render2Real is the ability to synthesize multiple valid 6-DoF object trajectories from a single human demonstration. In the case of multiple rigid objects that interact, (e.g. putting a mug on a coffee-maker) the original demonstration is valid only for a specific initial object configuration, and naively replaying it from a new initial pose would fail. To address this, we introduce a suite of trajectory interpolation and resampling techniques that adapt the original trajectory to new start and end poses while preserving its semantic intent.

We begin with a reference trajectory $\tau \in \mathbb{R}^{T\times 7}$ consisting of T waypoints provided by the part tracking from the demonstration video, each encoding an object orientation (quaternion) and position. Given a new initial pose $\mathbf{x}_{\text{start}}$ and the desired end pose $\mathbf{x}_{\text{end}}$ from human demonstration, we apply a spatial normalization that transforms the original trajectory into a canonical space. We compute the affine transform between the original and target endpoint poses, apply it to the translational component of the trajectory, and interpolate keyframe orientations using spherical linear interpolation (Slerp). This results in a new trajectory τ' that begins and ends at the desired poses while respecting the structure of the original motion. While these trajectories preserve high-level semantic intent, they are generated without explicit collision avoidance and may result in infeasible paths when initialized behind occluding objects. To mitigate this, we apply a sampling heuristic that biases the distribution of initial placements away from the goal pose (see Fig 4).

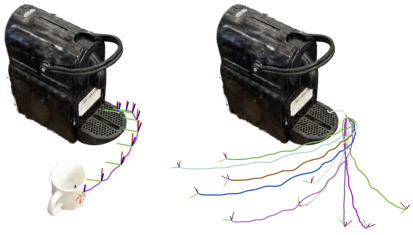

Figure 4: Trajectory Interpolation R2R2R adapts object motion to varied start/end configurations via spatial normalization and Slerp.

Grasp Pose Sampling: R2R2R estimates 3D hand keypoints from the demonstration video using [76], then determines object-hand interactions by computing the Euclidean distance between keypoints (index fingertip and thumb) and the centroids of all segmented object parts. This produces a distance matrix indexed over time and object parts. We identify the grasped part as the one with the minimum aggregate distance across the trajectory, effectively selecting the object most consistently proximal to the hand throughout the demonstration. To generate physically plausible grasps, we sample 3DGS means to construct a coarser triangle mesh (distinct from the high-resolution rendering mesh), apply surface smoothing and decimation to obtain consistent normals, and use an analytic antipodal grasp sampler following [10] to determine candidate grasp axes. For bimanual tasks, this process is applied independently per hand to infer separate object associations and grasps, supporting coordinated actions such as lifting or stabilization.

Differential Inverse Kinematics: For each grasp and object trajectory pair, we solve a differentiable inverse kinematics problem using PyRoki [77]. The solver computes smooth joint-space trajectories that induce the desired object motion across the pre-grasp, grasp, and post-grasp phases. Crucially, our method does *not require modeling object dynamics or simulating physics interactions*. Instead of solving for joint torques that would physically induce object movement (as in dynamic simulation), we assume the object rigidly follows the trajectory during contact. This kinematic assumption avoids challenges like contact modeling, compliance, or friction estimation. The solver simply ensures that robot kinematics can track the desired object motion subject to joint limits, and during pre- and post-grasp phases we additionally add smoothness and velocity costs to the differentiable IK problem, which generates valid grasp approach motions.

Rendering Diverse Environment Contexts: We apply domain randomization across both scene geometry and rendering parameters. This includes randomized lighting conditions (e.g., intensity, color temperature, background images), camera extrinsics (uniformly sampled up to 2cm translation and 5° rotation), and object initial poses (sampled within a workspace-relevant range). By modeling 3D object-centric representations, we can apply these augmentations directly during rendering.

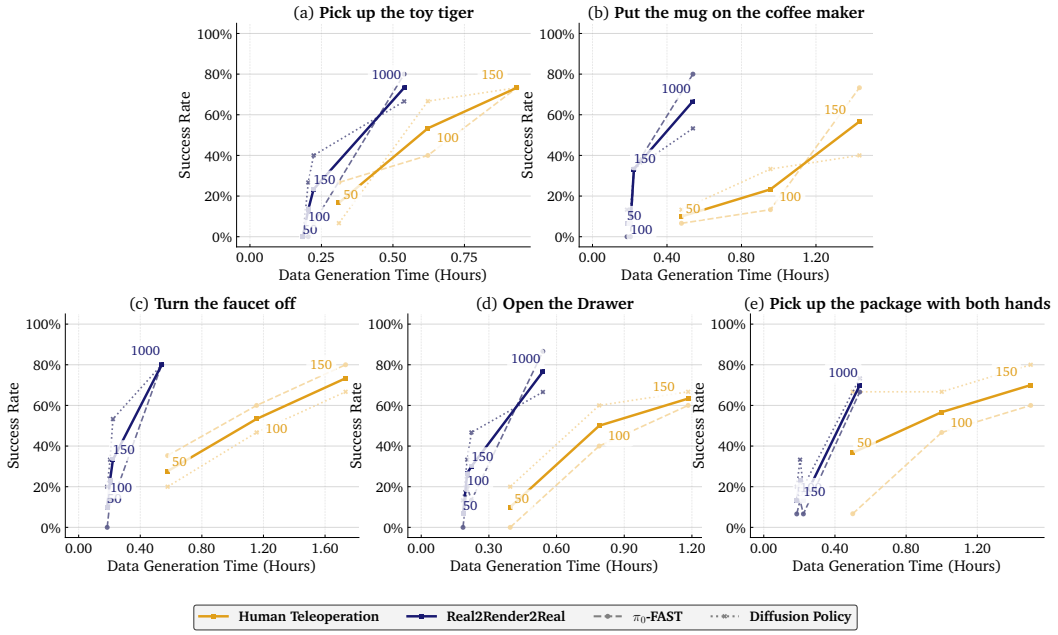

Figure 5: Physical Experiments Comparing Real2Render2Real to Human Teleoperation Data Efficiency Task success rate is plotted against data generation time in hours. Solid lines represent performance averaged across π_0-FAST and Diffusion Policy. The Real2Render2Real line (blue square) includes points corresponding to 50, 100, 150, and 1000 trajectories generated by a single Nvidia RTX 4090. The Human Teleoperation line (gold square) includes points corresponding to 50, 100, and 150 trajectories. For each task, each model is evaluated 15 times on the same set of pre-determined, in-distribution object pose. The Real2Render2Real data generation time includes a 10-minute setup cost, while the Human Teleoperation time is based on the real trajectory collection time of 150 demonstrations. Exact numbers for evaluation results can be found in Section 9.1.

Changes in camera pose or lighting do not affect the underlying kinematic rollout, allowing R2R2R to generate diverse visual contexts from a single demonstration. These augmentations expand the data distribution and improve generalization by mitigating the appearance gap and covariate shift between synthetic demonstrations and real-world deployment.

High-Throughput Rendering: IsaacLab [42] supports GPU-parallel execution of multiple environment contexts using tile-based rendering, deep-learning super sampling (DLSS), and mesh asset instancing. On a single NVIDIA RTX 4090, R2R2R uses the IsaacLab framework to render complete robot demonstrations at an average rate of 51 demonstrations per minute—compared to 1.7 demonstrations per minute via human teleoperation—yielding over a **27**$\times$ speedup. This throughput scales linearly with the number of rendering GPUs, as depicted in Figure 2. Data generation/collection time per task can be found in Table 7.

4.3 Policy Learning

We consider two modern imitation learning architectures: Diffusion Policy [20] and π_0-FAST [78]. We train Diffusion Policy from scratch for 100k steps conditioned on a 4-timestep history of proprioception and 448px RGB observations to iteratively denoise 16 future absolute end-effector poses in SE(3). We finetune π_0-FAST for 30k steps using Low Rank Adaptation (LoRA) [79] (rank=16), which takes 224px square image observations (to match pretraining resolution) and predicts a 10-step relative joint angle action-chunk. Training the diffusion policy takes approximately 3 hours on a single NVIDIA GH200, while π_0-FAST finetuning takes 11 hours. At deployment, both models receive raw RGB images and robot proprioception—SE(3) absolute end-effector pose for diffusion and joint positions for π_0-FAST—and output the corresponding action targets. To improve temporal consistency between actions predicted at different timesteps, we apply temporal ensembling [23] to predicted action-chunks during execution for both models. More training details can be found in Section 9.7.

5 Experiments

We conduct 1,050 physical robot evaluations on an ABB YuMi IRB14000 Bimanual Robot (a robot embodiment unseen during π_0-FAST pre-training) across five manipulation tasks using the trained policies. Policies are trained on either human teleoperation data or synthetic demonstrations generated by R2R2R. To assess how policy performance scales with training data, we train models with 50, 100, 150, and 1,000 rendered trajectories and up to 150 teleoperation trajectories per task. To ensure a fair comparison, all models are trained for a fixed number of training steps using only third-person RGB observations. For each task, each model is evaluated 15 times on the same set of pre-determined, in-distribution object pose.

We deliberately selected tasks to highlight R2R2R's ability to scale across diverse manipulation scenarios that involve varying physical and kinematic structures. Specifically, the tasks span: single-object picking (*"pick up the toy tiger"*), multi-object interaction (*"put the mug on the coffee maker"*), articulated object manipulation (*"turn the faucet off"* and *"open the drawer"*), and bimanual coordination (*"pick up the package with both hands"*). These categories correspond directly to R2R2R 's support for part-level segmentation, articulated object reconstruction, and multi-arm grasp planning, and are visualized in appendix Sections 9.3.1 to 9.3.5. We provide additional ablation experiments on trajectory interpolation (Section 9.2.1), increased background randomization (Section 9.2.2), and sim-real co-training (Section 9.2.3) in the appendix.

5.1 Performance Scaling and Comparison

To evaluate how well R2R2R-generated data supports policy learning compared to human teleoperated data, we analyze performance trends as a function of dataset size across the five tasks described above. Results are summarized in Figure 5. We observe that R2R2R-generated data scales predictably with dataset size: success rates increase monotonically for most tasks as the number of demonstrations grows. On the "Put the mug on the coffee maker" task (see Figure 5b), performance of Diffusion Policy trained on R2R2R data improves from 33.3% at 150 demos to 53.3% at 1000 demos, while π_0-FAST jumps from 33.3% to 80.0%. While higher quality, real-world data offers better performance in low-data regimes (e.g., π_0-FAST reaches 73.3% at 150 real demos vs. 33.3% at 150 R2R2R demos as shown in Figure 5b), as the scale increases to 1000 demos, R2R2R achieves performance that matches or surpasses teleoperation across multiple tasks. This suggests that while real data is more efficient per demonstration, R2R2R's generation enables scaling *trajectory diversity* far beyond human throughput, achieving competitive final performance with less collection effort.

To assess whether this performance is statistically comparable, we conduct formal significance and equivalence testing across all tasks and models. Appendix 9.9 shows that on the evaluated tasks, there are no statistically significant differences between policies trained on R2R2R versus human teleoperation data on the tasks we evaluated. Two One-Sided Tests (TOST) further suggest that the observed differences fall within a $\pm 5\%$ margin, indicating similar overall performance.

6 Conclusion

We propose R2R2R, a scalable data generation pipeline that creates robot training data from an object scan and a human demonstration video. R2R2R mitigates limitations of prior work by removing the need for teleoperation, robot hardware, or dynamics simulation. It leverages 3D Gaussian Splatting to represent both rigid and articulated objects, enabling parallel rendering using Gaussian-converted meshes and scalable rendering engines. These realistic renderings serve as visual observations for policy training. Given the robot's URDF, R2R2R synthesizes diverse robot trajectories with extracted object motion from one human demonstration using differential inverse kinematics. Experiments on five robotic tasks suggest that policies trained on data generated by R2R2R scale with data volume and perform comparably to those trained on teleoperated demonstrations, demonstrating that R2R2R is a practical and scalable pipeline for real-world robot dexterous manipulation policy learning.

7 Limitations

Real2Render2Real (R2R2R) enables scalable data generation and competitive real-world performance, but several limitations remain.

Reconstruction and Simulation Fidelity. R2R2R relies on vision-based reconstruction methods—such as 3D Gaussian Splatting and mesh conversion—that yield high-fidelity appearance but often lack watertight or physically plausible geometry. These limitations make it difficult to simulate realistic physical interactions, especially in contact-rich settings. As a result, R2R2R forgoes physics simulation entirely. While this design choice boosts scalability, it also restricts modeling of important dynamics such as friction, compliance, and force feedback. As real-to-sim pipelines mature in their physical realism [80], we anticipate extending R2R2R toward non-prehensile and dynamic tasks.

Two-Stage Input. While a single dynamic video (moving objects, moving camera) as input would offer higher scalability, current image-to-mesh pipelines lack the necessary multi-view consistency and self-supervised articulated part segmentation, both crucial for tasks like opening drawers or turning faucets. Future directions include unified pipelines based on emerging 4D-tracking and reconstruction works.

Scene Diversity and Collision Awareness. Trajectory generation in R2R2R is performed via geometric interpolation, without considering environmental context such as distractor objects or obstacles. As a result, synthesized trajectories may intersect with the scene geometry, leading to physically infeasible plans. Incorporating fast motion planning techniques during trajectory synthesis could improve collision avoidance and robustness, particularly in cluttered or multi-object scenes.

Scope of Manipulation Tasks. The current framework focuses exclusively on rigid and articulated objects using prehensile manipulation. It does not support deformable object handling or non-prehensile strategies such as pushing, toppling, or sliding. These interactions often demand accurate metric depth estimates and fine-grained physical modeling—both of which are challenging with monocular video and approximate geometry. Extending R2R2R to these broader manipulation regimes remains an open direction.

Grasping Generality. R2R2R's grasp generation module currently uses antipodal grasp sampling, which limits compatibility to parallel-jaw grippers. This restricts the generality of trained policies and excludes multi-fingered or anthropomorphic hands, which require richer grasp representations and contact models. Supporting these more complex end-effectors would require advances in grasp synthesis and simulation.

Tracking Robustness. Like other object-centric pipelines, R2R2R is vulnerable to tracking failures under fast motion, heavy occlusion, poor texture, or reflective surfaces. In such cases, object reconstructions and pose tracks may be inaccurate, resulting in degraded data quality. These failures can lead to invalid grasps or trajectories that do not transfer well to the real world. Robustifying tracking and adding confidence-aware filtering or correction is an important area for future work.

Policy Failure. Precise perception is still a major issue for visual imitation learning models (i.e. they will still miss the handle of the mug; sometimes an off-center grasp will lead to rotation of the object, resulting in missed grasps). Since we are only using third-person cameras, we hypothesize the models may achieve better performance by adding wrist cameras to increase grasp approach precision.

Addressing these limitations—through richer physical modeling, context-aware planning, expanded manipulation capabilities, and improved reconstruction robustness—offers a plausible path toward more general and reliable robot learning at scale.

8 Acknowledgement

This research was performed at the AUTOLAB at UC Berkeley in affiliation with the Berkeley AI Research (BAIR) Lab, and the CITRIS "People and Robots" (CPAR) Initiative. In their academic roles at UC Berkeley, Justin Yu, Letian Fu, Huang Huang, Karim El-Refai, and Ken Goldberg are supported

in part by donations from Toyota Research Institute, Autodesk, Meta, Google, Siemens, Bosch, and by equipment grants from NVIDIA, PhotoNeo, NSF AI4OPT Centre, and Intuitive Surgical. This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE 2146752. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. We deeply appreciate Jon Kuroda, Ion Stoica, and Joey Gonzalez for their generous support with computing hardware.

9 Appendix

Contents for This Section

9.1 Raw Evaluation Results

We report raw task success rates for each policy in Table 2.

Task / Policy	Teleop Trajectories			R2R2R Trajectories			
	50	100	150	50	100	150	1000
Pick up the tiger							
Diffusion Policy	6.7%	66.7%	73.3%	0.0%	26.7%	40.0%	66.6%
π_0-FAST (Finetuned)	26.7%	40.0%	73.3%	0.0%	0.0%	6.7%	80.0%
Put the mug on the coffee maker							
Diffusion Policy	13.3%	33.3%	40.0%	13.3%	13.3%	33.3%	53.3%
π_0-FAST (Finetuned)	6.6%	13.3%	73.3%	0.0%	0.0%	33.3%	80.0%
Pick up the package with both hands							
Diffusion Policy	66.7%	66.7%	80.0%	20.0%	33.3%	20.0%	73.3%
π_0-FAST (Finetuned)	6.7%	46.7%	60.0%	6.6%	13.3%	6.6%	66.7%
Open the drawer							
Diffusion Policy	20.0%	60.0%	66.7%	13.3%	33.3%	46.7%	66.7%
π_0-FAST (Finetuned)	0.0%	40.0%	60.0%	0.0%	20.0%	13.3%	86.6%
Turn the faucet off							
Diffusion Policy	20.0%	46.7%	66.7%	20.0%	33.3%	53.3%	80.0%
π_0-FAST (Finetuned)	35.3%	60.0%	80.0%	0.0%	13.3%	13.3%	80.0%

Table 2: Comparison of Physical Policy Success Rates Across Training Sources. Task success rates for Diffusion Policy and π_0-FAST trained exclusively on either human teleoperation data (left) or R2R2R-generated data (right). Each policy was evaluated on 15 trials per task using a binary success metric: a score of 1 is assigned for successful task completion, and 0 otherwise.

9.2 Additional Ablation Experiments

9.2.1 Trajectory Interpolation

R2R2R generates diverse trajectories by adapting a single human demonstration to new object poses through interpolation and spatial transformation (see Section 4.2 and Figure 4 for visualization). To evaluate the impact of this trajectory interpolation step, we ablate it by replaying only the original object motion track without adapting to varied initial and goal poses. Table 3 shows a substantial drop in performance when interpolation is disabled: on the "Put the mug on the coffee maker" task, success rates fall from 80.0% to 0.0% for π_0-FAST and from 53.3% to 6.7% for Diffusion Policy. This highlights that simply replaying object motion from a single demonstration is insufficient for generating transferable robot behaviors—trajectory adaptation is crucial to scaling data diversity in object-centric manipulation.

Policy	w/o Trajectory Interpolation (1k)	w/ Trajectory Interpolation (1k)
π_0-FAST (Finetuned)	0.0%	**80.0%**
Diffusion Policy	6.7%	**53.3%**

Table 3: Success rates on "Put the mug on the coffee maker" using R2R2R-generated data with and without trajectory interpolation (1,000 demos). Interpolation enables adapting object motion to varied contexts, which is critical for policy generalization.

9.2.2 Background and Tabletop Texture Augmentation

Our default data generation pipeline includes moderate visual augmentation, such as randomized lighting, camera pose, and object placement, as well as sampling from a limited set of lightbox-style background environments. To study the effect of stronger visual perturbations, we apply more aggressive augmentation that includes a wider variety of lightbox backgrounds and diverse tabletop textures (see Figure 6).

Table 4 reports the success rates on the task *Put the mug on the coffee maker* under this more varied visual setting. We observe a consistent drop in policy performance across both π_0-FAST and Diffusion Policy when trained on data with aggressive background and surface augmentation. This result suggests that while visual diversity is generally beneficial, overly strong appearance perturbations may harm policy learning when not properly balanced. Future work may investigate more principled augmentation schedules or adaptive augmentation strategies to preserve generalization while maintaining performance.

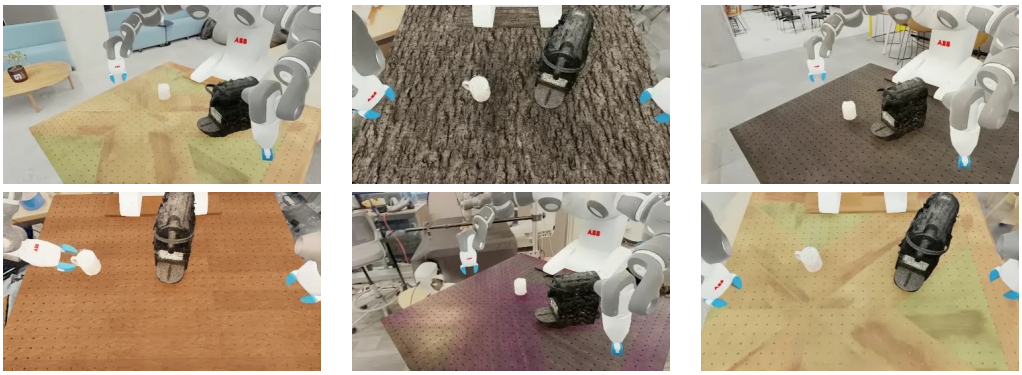

Figure 6: Background and Tabletop Texture Augmentation – Each image corresponds to a different environment.

Policy	Less Background Aug. (1k)	More Background Aug. (1k)
π_0-FAST (Fine-tuned)	**73.3%**	35.3%
Diffusion Policy	**53.3%**	33.3%

Table 4: Success rate comparison on *Put the mug on the coffee maker* with and without background and tabletop texture augmentation. We generated 1000 trajectories for each setting and evaluated across 15 trials.

9.2.3 Sim-and-Real Co-training

While sim-and-real co-training is not the main focus of this paper, we included additional results comparing policies exclusively on either R2R2R-generated data, human teleoperation data, and co-training setup that combines data from both sources. Specifically, for the task *Put the mug on the coffee maker*, we trained a policy using 1,000 R2R2R-generated demonstrations together with 150 human teleoperation demonstrations. We do not perform additional importance sampling or re-weighting of human teleoperation data. For the π_0-FAST policy, co-training achieved a success rate of 73.3%, which is on par with training using only R2R2R data or only real demonstrations individually. Co-training for diffusion policy yields a significant improvement over either real data only or R2R2R-generated data only, where the performance improved from 40.0% to 86.7%. We hypothesize that since LoRA [79] serves as a significant regularizer for π_0-FAST, end-to-end fine-tuning with completely unfrozen model with additional hyperparameters tuning could lead to better performance. For more in-depth analysis on how co-training can improve policy performance, please refer to [60, 81].

Policy	Real Data Only (150)	R2R2R Data Only (1k)	Co-Training (150+1k)
π_0-FAST	73.3%	**80.0%**	73.3%
Diffusion Policy	40.0%	53.3%	**86.7%**

Table 5: Success rate comparison on *Put the mug on the coffee maker* under different training datasets mixtures.

9.3 Task Visualizations

Physical policy rollout figures show model input RGB frames from real policy evaluation successes using either Diffusion Policy [20] or π_0-FAST [78]. The depicted policies were trained *exclusively* on R2R2R synthetic data.

9.3.1 Put the Mug on the Coffee Maker

Figure 7: Put the Mug on the Coffee Maker – Demonstration Video Frames.

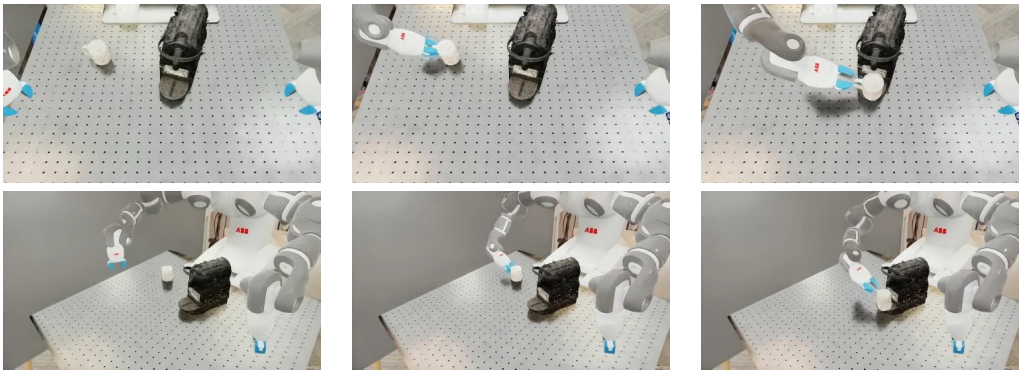

Figure 8: Put the Mug on the Coffee Maker – Example R2R2R Frames.

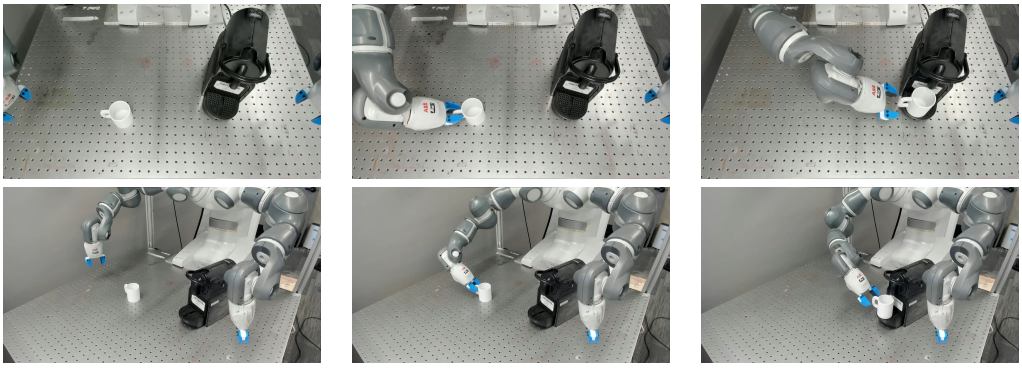

Figure 9: Put the Mug on the Coffee Maker – Physical Policy Rollout.

9.3.2 Turn off the Faucet

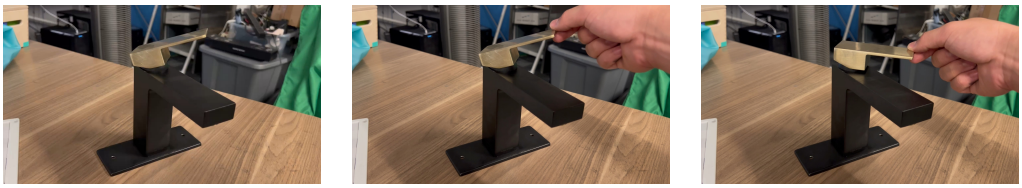

Figure 10: Turn off the Faucet - Demonstration Video Frames.

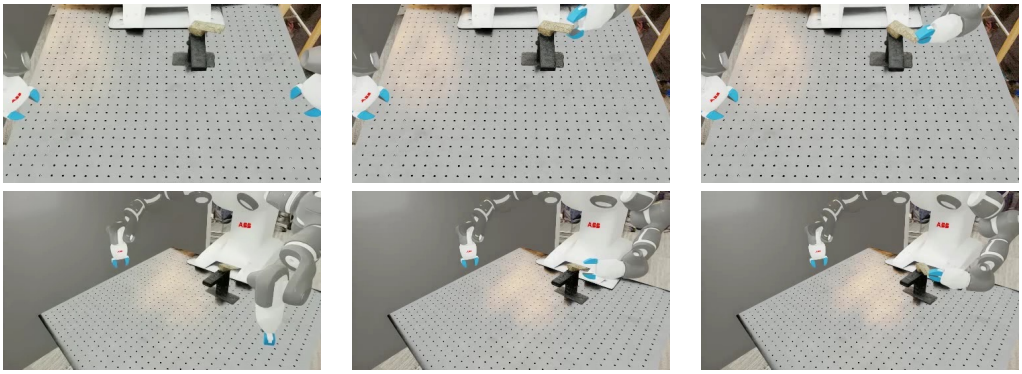

Figure 11: Turn off the Faucet - Example R2R2R Frames.

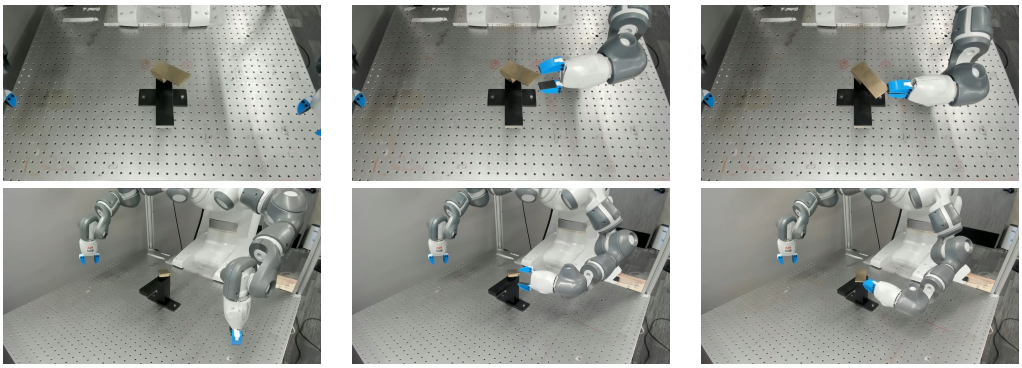

Figure 12: Turn off the Faucet - Physical Policy Rollout.

Note: For human teleoperated demonstrations, the teleoperator would push down on the faucet handle in a non-prehensile motion to turn it off instead of grasping the handle and twisting it closed as is done with R2R2R–where only prehensile grasping is currently supported.

9.3.3 Open the Drawer

Figure 13: Open the Drawer - Demonstration Video Frames. **Note:** The true video order–and thus the tracked trajectory–was in reverse, as a full multi-view scan for the inner drawer requires it to first be in an open configuration.

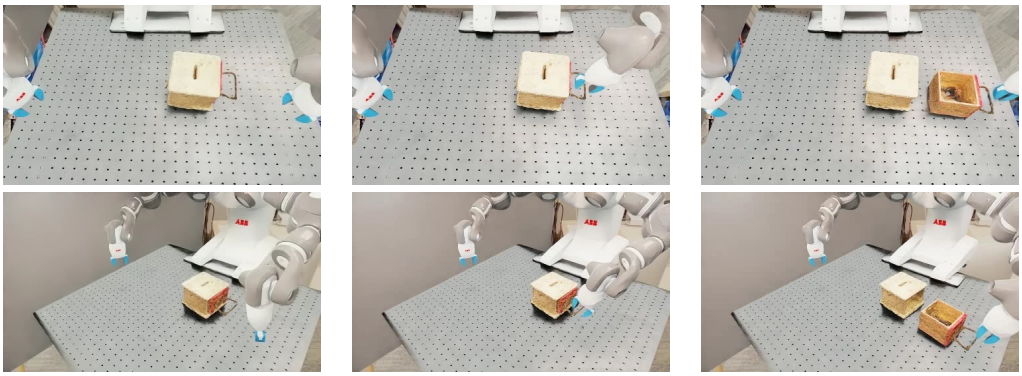

Figure 14: Open the Drawer - Example R2R2R Frames.

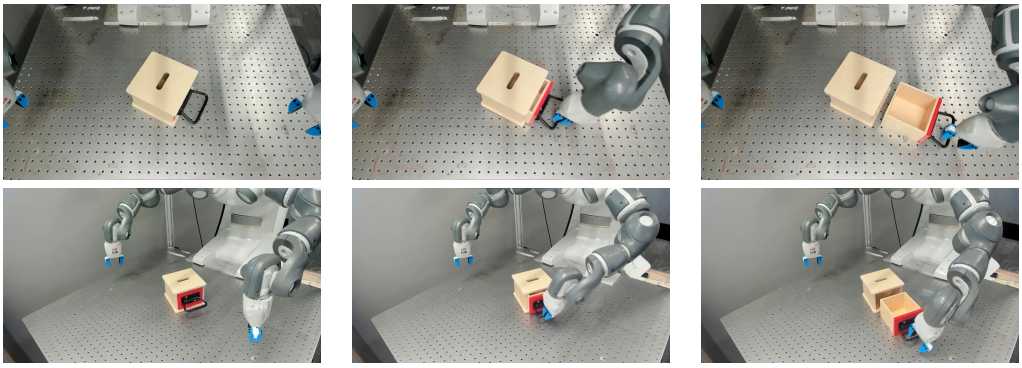

Figure 15: Open the Drawer - Physical Policy Rollout.

9.3.4 Lift up the Package with Both Hands

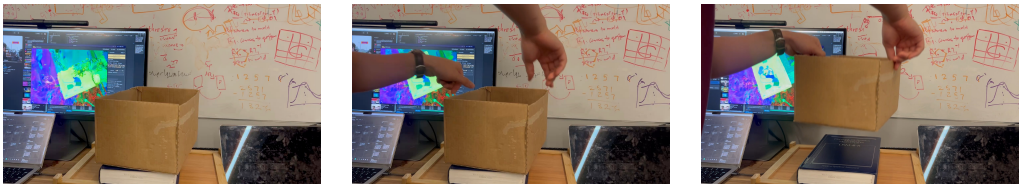

Figure 16: Lift up the Package with Both Hands - Demonstration Video Frames.

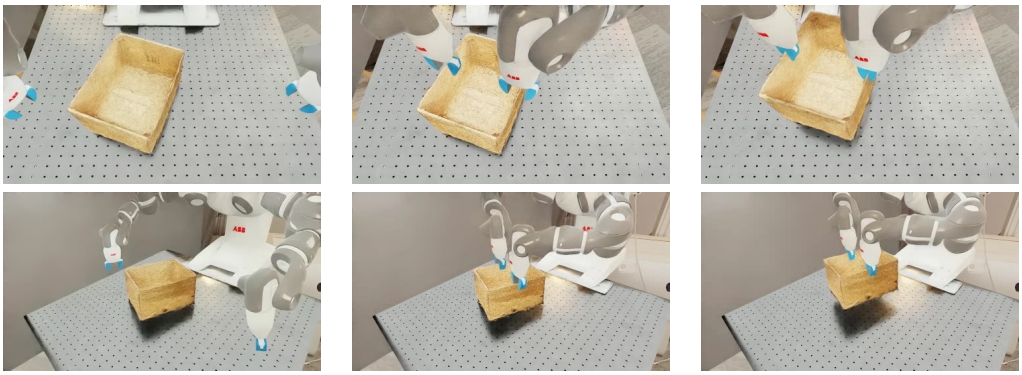

Figure 17: Lift up the Package with Both Hands - Example R2R2R Frames.

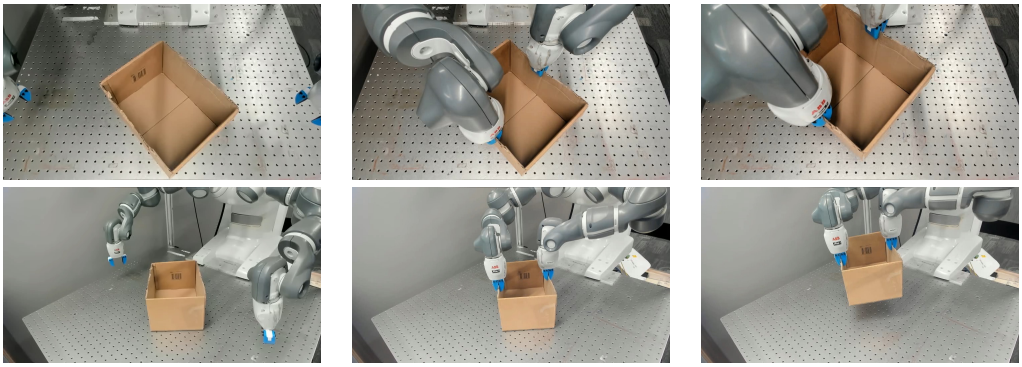

Figure 18: Lift up the Package with Both Hands - Physical Policy Rollout.

9.3.5 Pick up the Tiger

Figure 19: Pick up the Tiger - Demonstration Video Frames.

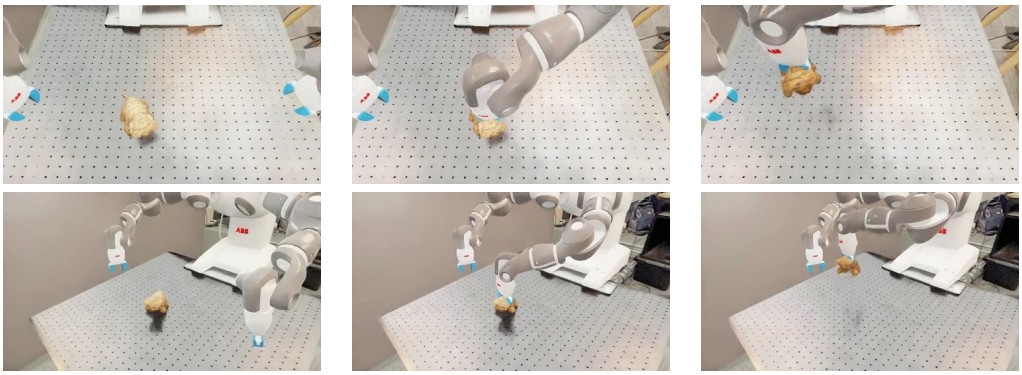

Figure 20: Pick up the Tiger - Example R2R2R Frames.

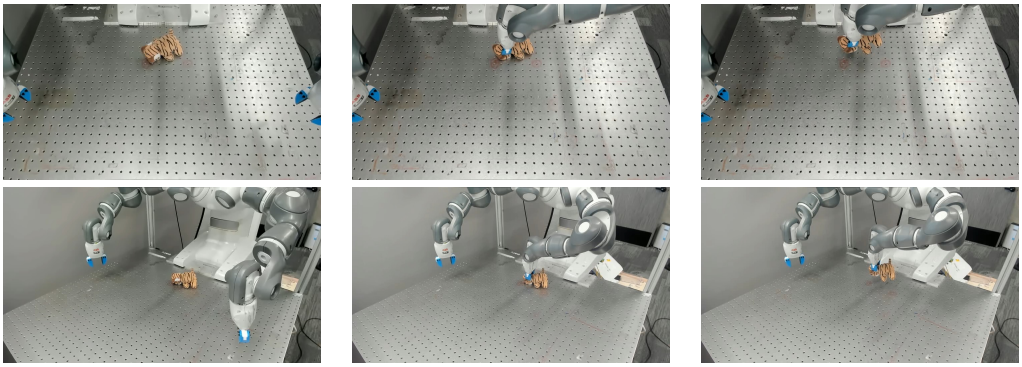

Figure 21: Pick up the Tiger - Physical Policy Rollout.

9.3.6 Put the Mug on the Coffee Maker (Franka Robot Embodiment)

Unlike vanilla π_0-FAST (DROID) [78], which operates under joint velocity control, R2R2R records only joint positions. To accommodate this difference, we fine-tuned π_0-FAST (DROID) to predict delta joint positions (and absolute gripper position, consistent with [78]). Since Franka's impedance control mode can be imprecise, we apply blocking control with temporal ensembling to improve execution accuracy.

Although the learned policy occasionally completes the task successfully, we observe several consistent failure modes, largely stemming from limitations of the default Franka gripper:

1. Collision during grasping: The grasp approach that is near parallel to the table often causes the gripper to collide with the table surface during mug pickup.
2. Off-center grasp: The gripper's wide jaws tend to produce asymmetric contacts, with one pad closer to the mug than the other. This imbalance induces rotation, leading to slippage.
3. Difficulty in precise placement: The wide gripper also makes it challenging to release the mug accurately onto the coffee machine.

To mitigate these issues, we recommend using the Robotiq 2F-85 gripper in future experiments. Its smaller form factor and improved grasping precision may reduce failure rates and improve placement consistency.

Figure 22: Put the Mug on the Coffee Maker (Franka Robot) - Example R2R2R Frames.

Figure 23: Put the Mug on the Coffee Maker (Franka Robot) - Physical Policy Rollout.

9.4 Qualitative Ablations

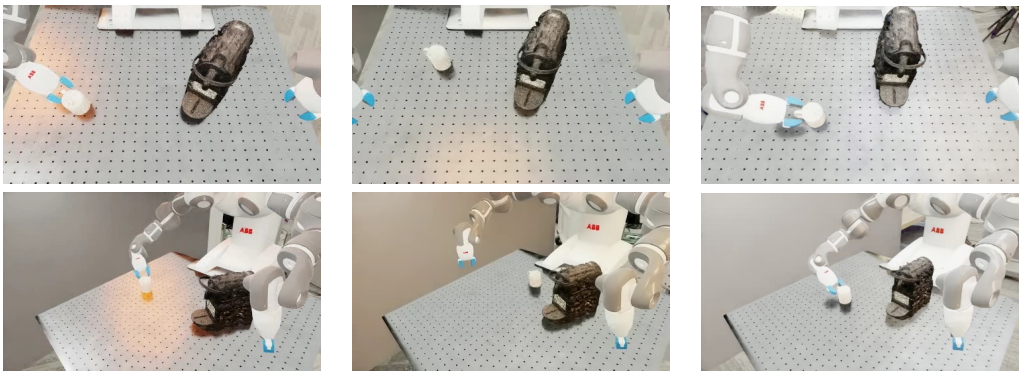

Figure 24: Real2Render2Real (Views From Both Cameras Shown). Base augmentations used in the main R2R2R experiments include: random sphere lighting, camera pose perturbation, robot initial joint perturbation, and randomized object initialization uniformly distributed via manual parameters.

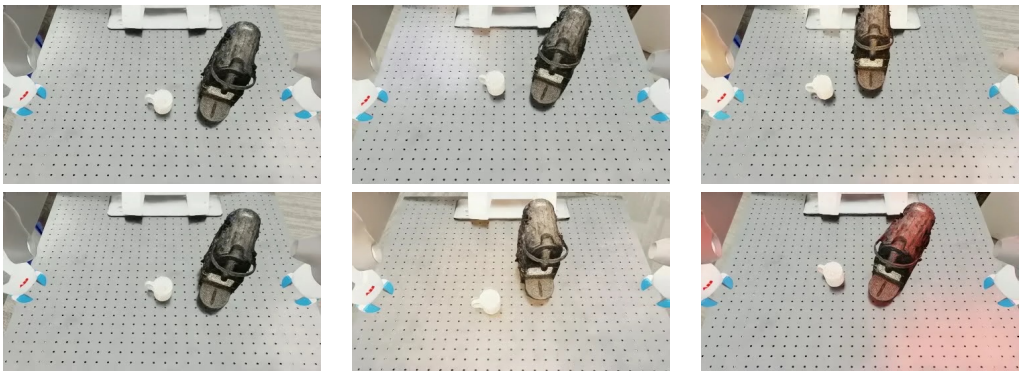

Figure 25: Trajectory Interpolation Turned Off (Top Camera Views Shown). *Note the fixed configuration of the mug with respect to the coffee maker.* With trajectory interpolation off for multiple rigid bodies, we may only densely follow the tracked trajectories shown in the video demonstration. Without it, the only method for increasing trajectory diversity would be to augment with part trajectories from adding/tracking additional demonstration videos.

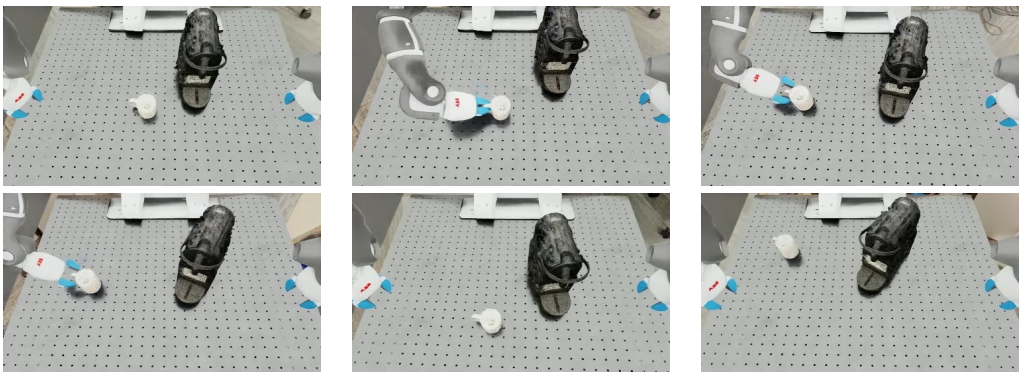

Figure 26: Random Lighting Augmentation Turned Off (Top Camera Views Shown). We turn off the randomized sphere light sources with varying colors and intensities. Uniform lighting is available from the only light source in the render scene – the skybox/dome-light asset.

9.5 Upfront Processing Time Until Generation

Task	Time to Complete
Scanning	1 min
Demonstration	$<$10 sec
GARField Segmentation [74]	2 min
3DGS Optimization	1 min
4D-DPM Tracking [71]	3 mins
SuGaR [75] Meshification	2 mins
Asset into IsaacLab	1 min

Table 6: Upfront Processing Time per Task Prior to R2R2R Data Generation. Breakdown of one-time preprocessing steps required to convert a demonstration video and scanned asset into a renderer-ready format for R2R2R. These steps include segmentation, tracking, meshification, and asset import.

9.6 Data Collection and Generation Time

Task	Teleop, 150 demos, 1 operator	R2R2R, 1k demos, 1 GPU
Pick up the tiger	60 mins	26.15 mins
Put the mug on the coffee maker	86 mins	38.22 mins
Pick up the package with both hands	90 mins	13.97 mins
Open the drawer	71 mins	16.95 mins
Turn the faucet off	104 mins	16.67 mins

Table 7: Time taken per task to either collect 150 demos through teleoperation with one human operator or to generate 1000 synthetic demos with R2R2R. Note: R2R2R generation times do not include the upfront processing time until generation.

9.7 Extended Training Details

We provide hyperparameters for training diffusion policy [20] from scratch and fine-tuning π_0-FAST [78] with LoRA in Table 8 and Table 9.

Config	Value
optimizer	AdamW [82]
base learning rate	2e-4
learning rate schedule	cosine decay [83]
batch size	64
weight decay	0.09
optimizer momentum	β_1, β_2 = 0.9, 0.999 [84]
warm up steps [85]	500
total steps	100,000
observation history	4
action dimension	20 (YuMi)
proprio format	absolute eef xyz, 6d rotation, absolute gripper position
action format	delta eef xyz, 6d rotation, absolute gripper position
action horizon	16
observation resolution	448

Table 8: Diffusion Policy Hyperparameters

Config	Value
optimizer	AdamW [82]
base learning rate	2.5e-5
learning rate schedule	cosine decay [83]
batch size	32
weight decay	0.09
LoRA Rank	16
LoRA Alpha	16
optimizer momentum	β_1, β_2 = 0.9, 0.95 [84]
warm up steps [85]	1000
total steps	30,000
action/proprio dimension	16 (YuMi) 8 (Franka)
proprio format	absolute joints positions, absolute gripper position
action format	delta joints, absolute gripper position
action horizon	10
observation resolution	224

Table 9: π_0-FAST hyperparameters

9.8 Extended Inference Details

In experiments, we use a Pi0-FAST model to predict robot actions as frequency-space tokens. While effective as an out-of-the-box VLA model, its autoregressive decoding procedure is inherently slow. Each token must be generated sequentially before the trajectory can be reconstructed. During evaluation, this latency posed a practical bottleneck: the raw observation-to-action inference time was high and resulted in slow experimental evaluations. We investigated whether it was possible to reduce inference latency without retraining the core model.

Our solution is a VLM early-stopping trick that exploits the frequency coefficient token ordering in the FAST implementation (0th and lower harmonics first) and energy compaction property of the Discrete Cosine Transform (DCT). Robot action trajectories tend to be smooth and temporally correlated, meaning that most of their information is concentrated in the low-frequency coefficients. Instead of waiting for the model to decode the full set of coefficients, we stop decoding early, reconstruct the trajectory using only the first few frequency coefficients, and immediately issue the action.

This trades a small increase in trajectory reconstruction mean squared error (since high-frequency details are dropped) for a substantial reduction in latency—often cutting inference time nearly in half. In practice, the essential structure of the action are preserved with just the first few DCT harmonics, making the trade-off favorable in latency-critical evaluation settings. Optional refinements, such as action-chunk ensembling or other smoothing techniques, can be added to compensate for missing fine-grained detail. We leave as future work more extensive evaluations of this test-time compute to reconstruction error trade-off.

9.9 Statistical Comparison Between Teleoperation and R2R2R Data Efficacy

To evaluate whether R2R2R-generated data yields performance comparable to human teleoperation, we apply the Two One-Sided Tests (TOST) procedure across all tasks and policies. Unlike traditional significance tests that ask whether two conditions differ, TOST tests whether the difference between them is small enough to be considered practically negligible. Specifically, we test whether the absolute difference in success rates falls within a $\pm 5\%$ margin—chosen to reflect a practically insignificant difference for robot policy success rates.

As shown in Table 10, no individual task satisfies both conditions required for statistical equivalence (i.e., both p-values below 0.05). However, the results consistently show no strong evidence that either R2R2R or teleoperation outperforms the other. In particular, the global test across all tasks yields one p-value below 0.05 and one above, suggesting performance is similar but not provably equivalent under the chosen threshold. These results support the interpretation that R2R2R can match the effectiveness of teleoperation across the evaluated tasks, while offering a significantly more scalable method for data generation.

Task	**Policy**	**TOST lower p**	**TOST upper p**
Pick up the toy tiger	Diffusion Policy	0.2656	0.5359
	π_0-FAST (Finetuned)	0.6891	0.1429
Put the mug on the coffee maker	Diffusion Policy	0.5349	0.2712
	π_0-FAST (Finetuned)	0.6891	0.1429
Turn the faucet off	Diffusion Policy	0.6891	0.1429
	π_0-FAST (Finetuned)	0.3729	0.3729
Open the Drawer	Diffusion Policy	0.2656	0.5359
	π_0-FAST (Finetuned)	0.8051	0.0806
Pick up the package with both hands	Diffusion Policy	0.1429	0.6891
	π_0-FAST (Finetuned)	0.3955	0.3955
Overall (All Tasks)	–	0.4271	0.0497

Table 10: Equivalence testing (TOST) between human teleoperation (150 trajectories) and R2R2R-generated data (1,000 trajectories). We report the p-values from Two One-Sided Tests (TOST) applied to each task and policy, using a $\pm 5\%$ success rate margin as the equivalence threshold. The "lower p" tests whether R2R2R performs *no worse* than teleoperation by more than 5%, while the "upper p" tests whether teleoperation performs *no worse* than R2R2R. Statistical equivalence is only confirmed when *both* p-values fall below 0.05.

References

[1] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE transactions on Robotics and Automation*, 12(4):566–580, 1996.

[2] S. M. LaValle and J. J. Kuffner. Rapidly-exploring random trees: Progress and prospects: Steven m. lavalle, iowa state university, a james j. kuffner, jr., university of tokyo, tokyo, japan. *Algorithmic and computational robotics*, pages 303–307, 2001.

[3] Y. Tassa, T. Erez, and E. Todorov. Synthesis and stabilization of complex behaviors through online trajectory optimization. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4906–4913. IEEE, 2012.

[4] M. Posa, S. Kuindersma, and R. Tedrake. Optimization and stabilization of trajectories for constrained dynamical systems. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1366–1373. IEEE, 2016.

[5] C. Mastalli, R. Budhiraja, W. Merkt, G. Saurel, B. Hammoud, M. Naveau, J. Carpentier, L. Righetti, S. Vijayakumar, and N. Mansard. Crocoddyl: An efficient and versatile framework for multi-contact optimal control. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2536–2542. IEEE, 2020.

[6] A. Kumar, Z. Fu, D. Pathak, and J. Malik. Rma: Rapid motor adaptation for legged robots. *RSS*, 2021.

[7] I. Radosavovic, T. Xiao, B. Zhang, T. Darrell, J. Malik, and K. Sreenath. Real-world humanoid locomotion with reinforcement learning. *Science Robotics*, 9(89), 2024.

[8] I. Lenz, H. Lee, and A. Saxena. Deep learning for detecting robotic grasps. *The International Journal of Robotics Research*, 34(4-5):705–724, 2015.

[9] L. Pinto and A. Gupta. Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 3406–3413. IEEE, 2016.

[10] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg. Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. *arXiv preprint arXiv:1703.09312*, 2017.

[11] J. Mahler, M. Matl, V. Satish, M. Danielczuk, B. DeRose, S. McKinley, and K. Goldberg. Learning ambidextrous robot grasping policies. *Science Robotics*, 4(26):eaau4984, 2019.

[12] OpenAI. GPT-4 technical report. *CoRR*, abs/2303.08774, 2023. URL https://doi.org/10.48550/arXiv.2303.08774.

[13] G. Team. Gemini: A family of highly capable multimodal models, 2024. URL https://arxiv.org/abs/2312.11805.

[14] J. Bai, S. Bai, S. Yang, S. Wang, S. Tan, P. Wang, J. Lin, C. Zhou, and J. Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 2023. URL https://doi.org/10.48550/arXiv.2308.12966.

[15] H. Liu, C. Li, Q. Wu, and Y. J. Lee. Visual instruction tuning. In *NeurIPS*, 2023.

[16] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, et al. $\pi 0$: A vision-language-action flow model for general robot control, 2024. *URL https://arxiv. org/abs/2410.24164*.

[17] J. Bjorck, F. Castañeda, N. Cherniadev, X. Da, R. Ding, L. Fan, Y. Fang, D. Fox, F. Hu, S. Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

[18] G. R. Team, S. Abeyruwan, J. Ainslie, J.-B. Alayrac, M. G. Arenas, T. Armstrong, A. Balakrishna, R. Baruch, M. Bauza, M. Blokzijl, et al. Gemini robotics: Bringing ai into the physical world. *arXiv preprint arXiv:2503.20020*, 2025.

[19] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. P. Foster, P. R. Sanketi, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang, and C. Finn. Openvla: An open-source vision-language-action model. In P. Agrawal, O. Kroemer, and W. Burgard, editors, *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pages 2679–2713. PMLR, 06–09 Nov 2025. URL https://proceedings.mlr.press/v270/kim25c.html.

[20] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023.

[21] Octo Model Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, C. Xu, J. Luo, T. Kreiman, Y. Tan, L. Y. Chen, P. Sanketi, Q. Vuong, T. Xiao, D. Sadigh, C. Finn, and S. Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024.

[22] H. Huang, F. Liu, L. Fu, T. Wu, M. Mukadam, J. Malik, K. Goldberg, and P. Abbeel. Otter: A vision-language-action model with text-aware feature extraciton. *arXiv preprint arXiv:2503.03734*, 2025.

[23] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *RSS*, 2023.

[24] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

[25] L. Fu, H. Huang, G. Datta, L. Y. Chen, W. C.-H. Panitch, F. Liu, H. Li, and K. Goldberg. In-context imitation learning via next-token prediction. *arXiv preprint arXiv:2408.15980*, 2024.

[26] K. Goldberg. Igniting the real robot revolution requires closing the "data gap". Invited Talk at NVIDIA GPU Technology Conference (GTC), March 2025. URL https://www.nvidia.com/gtc/. Talk ID: S74739.

[27] S. Mirchandani, S. Belkhale, J. Hejna, E. Choi, M. S. Islam, and D. Sadigh. So you think you can scale up autonomous robot data collection? In *Proceedings of the 8th Conference on Robot Learning (CoRL), November 2024*, 2024.

[28] B. Chen, Z. Xu, S. Kirmani, B. Ichter, D. Sadigh, L. Guibas, and F. Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14455–14465, June 2024.

[29] Z. Teed and J. Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020.

[30] S. Wang, V. Leroy, Y. Cabon, B. Chidlovskii, and J. Revaud. Dust3r: Geometric 3d vision made easy. In *CVPR*, 2024.

[31] J. Zhang, C. Herrmann, J. Hur, V. Jampani, T. Darrell, F. Cole, D. Sun, and M.-H. Yang. Monst3r: A simple approach for estimating geometry in the presence of motion. *arXiv preprint arXiv:2410.03825*, 2024.

[32] R. Liu, R. Wu, B. Van Hoorick, P. Tokmakov, S. Zakharov, and C. Vondrick. Zero-1-to-3: Zero-shot One Image to 3D Object, Mar. 2023. URL http://arxiv.org/abs/2303.11328. arXiv:2303.11328 [cs].

[33] V. Guizilini, M. Z. Irshad, D. Chen, G. Shakhnarovich, and R. Ambrus. Zero-shot novel view and depth synthesis with multi-view geometric diffusion, 2025. URL https://arxiv.org/abs/2501.18804.

[34] J. L. Schönberger and J.-M. Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[35] P. Katara, Z. Xian, and K. Fragkiadaki. Gen2sim: Scaling up robot learning in simulation with generative models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6672–6679. IEEE, 2024.

[36] Y. Wang, Z. Xian, F. Chen, T.-H. Wang, Y. Wang, K. Fragkiadaki, Z. Erickson, D. Held, and C. Gan. Robogen: Towards unleashing infinite data for automated robot learning via generative simulation. *arXiv preprint arXiv:2311.01455*, 2023.

[37] W. Ye, F. Liu, Z. Ding, Y. Gao, O. Rybkin, and P. Abbeel. Video2policy: Scaling up manipulation tasks in simulation through internet videos. *arXiv preprint arXiv:2502.09886*, 2025.

[38] H. Geng, F. Wang, S. Wei, Y. Li, B. Wang, B. An, C. T. Cheng, H. Lou, P. Li, Y.-J. Wang, Y. Liang, D. Goetting, C. Xu, H. Chen, Y. Qian, Y. Geng, J. Mao, W. Wan, M. Zhang, J. Lyu, S. Zhao, J. Zhang, J. Zhang, C. Zhao, H. Lu, Y. Ding, R. Gong, Y. Wang, Y. Kuang, R. Wu, B. Jia, C. Sferrazza, H. Dong, S. Huang, K. Sreenath, Y. Wang, J. Malik, and P. Abbeel. Roboverse: Towards a unified platform, dataset and benchmark for scalable and generalizable robot learning, 2025. URL https://roboverseorg.github.io/.

[39] M. Torne, A. Jain, J. Yuan, V. Macha, L. Ankile, A. Simeonov, P. Agrawal, and A. Gupta. Robot learning with super-linear scaling. *arXiv preprint arXiv:2412.01770*, 2024.

[40] M. Li, Z. Ferguson, T. Schneider, T. R. Langlois, D. Zorin, D. Panozzo, C. Jiang, and D. M. Kaufman. Incremental potential contact: intersection-and inversion-free, large-deformation dynamics. *ACM Trans. Graph.*, 39(4):49, 2020.

[41] C. M. Kim, M. Danielczuk, I. Huang, and K. Goldberg. Ipc-graspsim: Reducing the sim2real gap for parallel-jaw grasping with the incremental potential contact model. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6180–6187. IEEE, 2022.

[42] M. Mittal, C. Yu, Q. Yu, J. Liu, N. Rudin, D. Hoeller, J. L. Yuan, R. Singh, Y. Guo, H. Mazhar, A. Mandlekar, B. Babich, G. State, M. Hutter, and A. Garg. Orbit: A unified simulation framework for interactive robot learning environments. *IEEE Robotics and Automation Letters*, 8(6):3740–3747, 2023. doi:10.1109/LRA.2023.3270034.

[43] C. Mitash, F. Wang, S. Lu, V. Terhuja, T. Garaas, F. Polido, and M. Nambi. Armbench: An object-centric benchmark dataset for robotic manipulation. 2023. URL https://www.amazon.science/publications/armbench-an-object-centric-benchmark-dataset-for-robotic-manipulation.

[44] A. Sohn, A. Nagabandi, C. Florensa, D. Adelberg, D. Wu, H. Farooq, I. Clavera, J. Welborn, J. Chen, N. Mishra, P. Chen, P. Qian, P. Abbeel, R. Duan, V. Vijay, and Y. Liu. Introducing rfm-1: Giving robots human-like reasoning capabilities, Mar. 2024. URL https://covariant.ai/insights/introducing-rfm-1-giving-robots-human-like-reasoning-capabilities/. Accessed: 2025-04-29.

[45] V. Satish, J. Mahler, and K. Goldberg. Prime-1: Scaling large robot data for industrial reliability, Jan. 2025. URL https://www.ambirobotics.com/blog/prime-1-scaling-large-robot-data-for-industrial-reliability/. Accessed: 2025-04-29.

[46] Z. Teed and J. Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *Advances in neural information processing systems*, 34:16558–16569, 2021.

[47] E. Collaboration. Open x-embodiment: Robotic learning datasets and rt-x models, 2024. URL https://arxiv.org/abs/2310.08864.

[48] F. Ebert, Y. Yang, K. Schmeckpeper, B. Bucher, G. Georgakis, K. Daniilidis, C. Finn, and S. Levine. Bridge data: Boosting generalization of robotic skills with cross-domain datasets. *arXiv preprint arXiv:2109.13396*, 2021.

[49] H. Walke, K. Black, A. Lee, M. J. Kim, M. Du, C. Zheng, T. Zhao, P. Hansen-Estruch, Q. Vuong, A. He, V. Myers, K. Fang, C. Finn, and S. Levine. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning (CoRL)*, 2023.

[50] H.-S. Fang, H. Fang, Z. Tang, J. Liu, J. Wang, H. Zhu, and C. Lu. Rh20t: A robotic dataset for learning diverse skills in one-shot. In *RSS 2023 Workshop on Learning for Task and Motion Planning*, 2023.

[51] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, P. Florence, C. Fu, M. G. Arenas, K. Gopalakrishnan, K. Han, K. Hausman, A. Herzog, J. Hsu, B. Ichter, A. Irpan, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, L. Lee, T.-W. E. Lee, S. Levine, Y. Lu, H. Michalewski, I. Mordatch, K. Pertsch, K. Rao, K. Reymann, M. Ryoo, G. Salazar, P. Sanketi, P. Sermanet, J. Singh, A. Singh, R. Soricut, H. Tran, V. Vanhoucke, Q. Vuong, A. Wahid, S. Welker, P. Wohlhart, J. Wu, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *arXiv preprint arXiv:2307.15818*, 2023.

[52] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International journal of robotics research*, 37(4-5):421–436, 2018.

[53] J. Kerr, H. Huang, A. Wilcox, R. Hoque, J. Ichnowski, R. Calandra, and K. Goldberg. Self-supervised visuo-tactile pretraining to locate and follow garment features. *arXiv preprint arXiv:2209.13042*, 2022.

[54] A. Wilcox, J. Kerr, B. Thananjeyan, J. Ichnowski, M. Hwang, S. Paradis, D. Fer, and K. Goldberg. Learning to localize, grasp, and hand over unmodified surgical needles. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 9637–9643. IEEE, 2022.

[55] L. Y. Chen, H. Huang, E. Novoseller, D. Seita, J. Ichnowski, M. Laskey, R. Cheng, T. Kollar, and K. Goldberg. Efficiently learning single-arm fling motions to smooth garments. In *The International Symposium of Robotics Research*, pages 36–51. Springer, 2022.

[56] L. Fu, H. Huang, L. Berscheid, H. Li, K. Goldberg, and S. Chitta. Safe self-supervised learning in real of visuo-tactile feedback policies for industrial insertion. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10380–10386. IEEE, 2023.

[57] I. Radosavovic, B. Shi, L. Fu, K. Goldberg, T. Darrell, and J. Malik. Robot learning with sensorimotor pre-training. *arXiv preprint arXiv:2306.10007*, 2023.

[58] A. Mandlekar, S. Nasiriany, B. Wen, I. Akinola, Y. Narang, L. Fan, Y. Zhu, and D. Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations. In *7th Annual Conference on Robot Learning*, 2023.

[59] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 23–30. IEEE, 2017.

[60] A. Maddukuri, Z. Jiang, L. Y. Chen, S. Nasiriany, Y. Xie, Y. Fang, W. Huang, Z. Wang, Z. Xu, N. Chernyadev, S. Reed, K. Goldberg, A. Mandlekar, L. Fan, and Y. Zhu. Sim-and-real co-training: A simple recipe for vision-based robotic manipulation. In *Proceedings of Robotics: Science and Systems (RSS)*, Los Angeles, CA, USA, 2025.

[61] X. Li, J. Li, Z. Zhang, R. Zhang, F. Jia, T. Wang, H. Fan, K.-K. Tseng, and R. Wang. Robogsim: A real2sim2real robotic gaussian splatting simulator. *arXiv preprint arXiv:2411.11839*, 2024.

[62] L. Y. Chen, C. Xu, K. Dharmarajan, M. Z. Irshad, R. Cheng, K. Keutzer, M. Tomizuka, Q. Vuong, and K. Goldberg. Rovi-aug: Robot and viewpoint augmentation for cross-embodiment robot learning. *arXiv preprint arXiv:2409.03403*, 2024.

[63] Z. Jiang, Y. Xie, K. Lin, Z. Xu, W. Wan, A. Mandlekar, L. Fan, and Y. Zhu. Dexmimicgen: Automated data generation for bimanual dexterous manipulation via imitation learning. *arXiv preprint arXiv:2410.24185*, 2024.

[64] M. Lepert, J. Fang, and J. Bohg. Phantom: Training robots without robots using only human videos. *arXiv preprint arXiv:2503.00779*, 2025.

[65] Z. Xue, S. Deng, Z. Chen, Y. Wang, Z. Yuan, and H. Xu. Demogen: Synthetic demonstration generation for data-efficient visuomotor policy learning. *arXiv preprint arXiv:2502.16932*, 2025.

[66] J. Duan, Y. R. Wang, M. Shridhar, D. Fox, and R. Krishna. AR2-d2: Training a robot without a robot. In *7th Annual Conference on Robot Learning*, 2023. URL https://openreview.net/forum?id=JdpleC92J4.

[67] V. Lim, H. Huang, L. Y. Chen, J. Wang, J. Ichnowski, D. Seita, M. Laskey, and K. Goldberg. Planar robot casting with real2sim2real self-supervised learning, 2022. URL https://arxiv.org/abs/2111.04814.

[68] N. Pfaff, E. Fu, J. Binagia, P. Isola, and R. Tedrake. Scalable real2sim: Physics-aware asset generation via robotic pick-and-place setups. *arXiv preprint arXiv:2503.00370*, 2025.

[69] T. Dai, J. Wong, Y. Jiang, C. Wang, C. Gokmen, R. Zhang, J. Wu, and L. Fei-Fei. Automated creation of digital cousins for robust policy learning. *arXiv preprint arXiv:2410.07408*, 2024.

[70] M. Torne, A. Simeonov, Z. Li, A. Chan, T. Chen, A. Gupta, and P. Agrawal. Reconciling reality through simulation: A real-to-sim-to-real approach for robust manipulation. *arXiv preprint arXiv:2403.03949*, 2024.

[71] J. Kerr, C. M. Kim, M. Wu, B. Yi, Q. Wang, K. Goldberg, and A. Kanazawa. Robot see robot do: Imitating articulated object manipulation with monocular 4d reconstruction. In *8th Annual Conference on Robot Learning*, 2024. URL https://openreview.net/forum?id=2LLu3gavF1.

[72] J. Yu, K. Hari, K. El-Refai, A. Dalil, J. Kerr, C.-M. Kim, R. Cheng, M. Z. Irshad, and K. Goldberg. Persistent object gaussian splat (pogs) for tracking human and robot manipulation of irregularly shaped objects. *ICRA*, 2025.

[73] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

[74] C. M. Kim, M. Wu, J. Kerr, M. Tancik, K. Goldberg, and A. Kanazawa. Garfield: Group anything with radiance fields. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[75] A. Guédon and V. Lepetit. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. *CVPR*, 2024.

[76] G. Pavlakos, D. Shan, I. Radosavovic, A. Kanazawa, D. Fouhey, and J. Malik. Reconstructing hands in 3D with transformers. In *CVPR*, 2024.

[77] C. M. Kim*, B. Yi*, H. Choi, Y. Ma, K. Goldberg, and A. Kanazawa. Pyroki: A modular toolkit for robot kinematic optimization, 2025. URL https://arxiv.org/abs/2505.03728.

[78] K. Pertsch, K. Stachowicz, B. Ichter, D. Driess, S. Nair, Q. Vuong, O. Mees, C. Finn, and S. Levine. Fast: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025.

[79] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, and W. Chen. Lora: Low-rank adaptation of large language models. *CoRR*, abs/2106.09685, 2021. URL https://arxiv.org/abs/2106.09685.

[80] N. Pfaff, E. Fu, J. Binagia, P. Isola, and R. Tedrake. Scalable real2sim: Physics-aware asset generation via robotic pick-and-place setups. 2025. URL https://arxiv.org/abs/2503.00370.

[81] A. Wei, A. Agarwal, B. Chen, R. Bosworth, N. Pfaff, and R. Tedrake. Empirical analysis of sim-and-real cotraining of diffusion policies for planar pushing from pixels. *arXiv preprint arXiv:2503.22634*, 2025.

[82] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[83] I. Loshchilov and F. Hutter. Sgdr: Stochastic gradient descent with warm restarts. 2017.

[84] M. Chen, A. Radford, R. Child, J. Wu, H. Jun, D. Luan, and I. Sutskever. Generative pretraining from pixels. 2020.

[85] P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv:1706.02677*, 2017.