Robotics: Science and Systems 2026
Sydney, Australia, July 13–17, 2026

R2RGen: Real-to-Real 3D Data Generation for Spatially Generalized Manipulation

Xiuwei Xu[123*], Angyuan Ma[123*], Hankun Li[123], Bingyao Yu[123✉], Zheng Zhu[4], Jie Zhou[123], Jiwen Lu[123✉]

[1]Department of Automation, Tsinghua University, [2]Beijing Key Laboratory of Embodied Intelligence Systems
[3]Institute for Embodied Intelligence and Robotics, Tsinghua University, [4]GigaAI, *Equal contribution

r2rgen.github.io

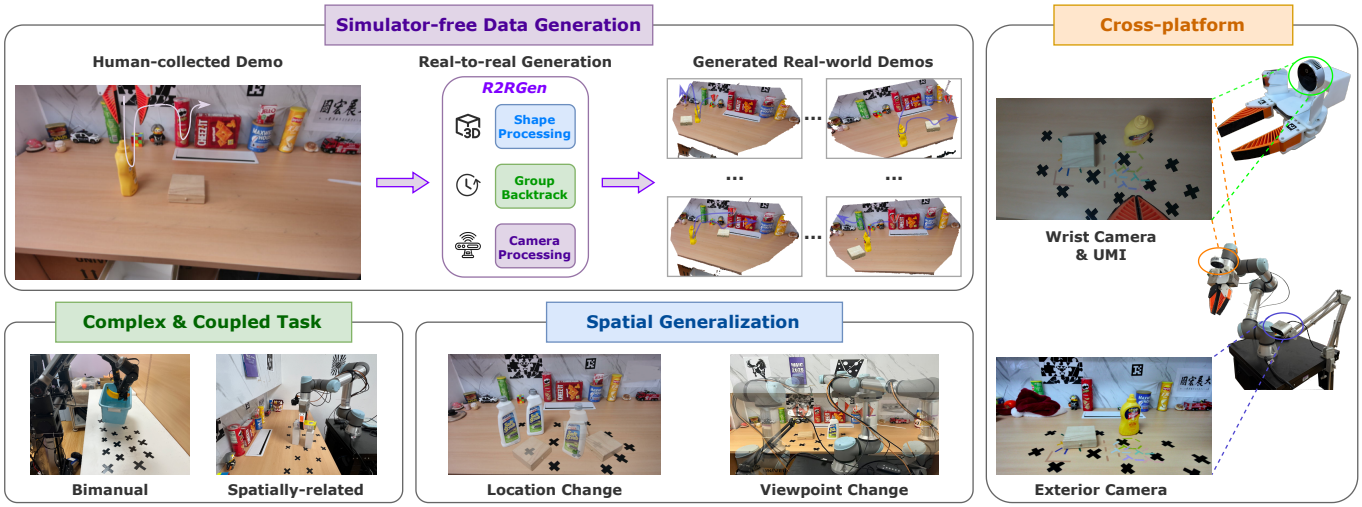

Fig. 1: R2RGen is a simulator-free data generation framework. Given one human-collected demonstration, R2RGen directly parses and edits both pointcloud observations and action trajectories in a shared 3D space. R2RGen achieves strong spatial generalization on diverse complex tasks and different data collection setups.

***Abstract*—Towards the aim of generalized robotic manipulation, spatial generalization is the most fundamental capability that requires the policy to work robustly under different spatial distribution of objects, environment and agent itself. To achieve this, substantial human demonstrations need to be collected to cover different spatial configurations for training a generalized visuomotor policy via imitation learning. Prior works explore a promising direction that leverages data generation to acquire abundant spatially diverse data from minimal source demonstrations. However, most approaches face significant sim-to-real gap and are often limited to constrained settings, such as fixed-base scenarios and predefined camera viewpoints. In this paper, we propose a real-to-real 3D data generation framework (R2RGen) that directly augments the pointcloud observation-action pairs to generate real-world data. R2RGen is simulator- and rendering-free, thus being efficient and plug-and-play. Specifically, we propose a unified three-stage framework, which (1) pre-processes source demonstrations under different camera setups in a shared 3D space with scene / trajectory parsing; (2) augments objects and robot's position with a group-wise backtracking strategy; (3) aligns the distribution of generated data with real-world 3D sensor using camera-aware post-processing. Empirically, R2RGen substantially enhances data efficiency on extensive experiments and demonstrates strong potential for scaling and application on mobile manipulation.**

I. Introduction

Robotic manipulation with visuomotor policy [8, 48, 13] has achieved great progress in recent years, while the reliance on large amount of human-collected data during imitation learning becomes the main bottlenecks for application and further scaling up [29]. Unlike most prior work focused on fixed-tabletop arms, this paper studies a more general manipulation setting involving mobile manipulators. Since the mobile base may be located at arbitrary positions, the resulting viewpoint variation further increases the policy's reliance on extensive training data [34].

Spatial generalization is the primary factor driving the substantial data demand during visuomotor policy learning. As pointed out by [14], control difficulty is not uniformly distributed among the human-collected trajectory. Note a trajectory can be divided into two categories of segments: contact-rich segments involving the interaction between robotic arm and objects, and other segments simply indicating the movement of robotic arm in free space, which are also known as *skill* and *motion* segments respectively. Skill segments are generally more challenging, whereas motion segments can often be handled effectively through motion planning. However, even thought skill segments are more informative, the majority of human demonstration effort is typically devoted to teaching motion behaviors. For instance, in a task such as "put apple on plate"—even with identical apple, plate, and pick and place skill—hundreds of demonstrations may be needed to cover varying objects' spatial arrangements and robot's base positions

to learn a generalized policy. Therefore, spatial generalization remains a fundamental bottleneck in data efficiency.

To reduce redundant human effort on ensuring spatial generalization, MimicGen [30] and follow-up works [17, 14, 19] replace the tedious relocate-and-recollect data collection procedure with automatic demonstration generation. These methods only require a few human-collected data, based on which they augment object configurations and apply transformation and interpolation to generate diverse trajectories with different motion patterns. Though achieving satisfactory performance in simulation, these methods require on-robot rollouts to collect real-world observation-action pair, which takes much more time and relies on human supervision. Recently, DemoGen [41] introduces a 3D-based data generation method that builds on point-cloud input policy [43]. By operating directly in the 3D domain, the approach augments object point clouds to synthesize varied trajectories along with their corresponding visual observations. This pipeline is simulator- and rendering-free, thus being very efficient and avoiding sim-to-real gap. However, DemoGen suffers from several critical limitations that restrict its broader applicability: (1) it is restricted to a fixed base and exterior camera, thus unable to handle viewpoint changes or wrist-camera setup; (2) it imposes strong assumption on input data, where the pointcloud of environment should be cropped, limited number of objects are supported, and each skill must involve only one target object; (3) it suffers from visual mismatch problem, i.e., large augmentation leads to incomplete pointcloud observation. Due to these constraints, DemoGen does not fully achieve practical real-to-real generation and remains limited in handling diverse task settings.

In this paper, we propose R2RGen, a real-to-real 3D data generation framework which is applicable for mobile robot [13], compatible for different camera setups, works on raw pointcloud observation and supports any number of objects and interaction modes. Given a source demonstration, previous methods apply spatial transformation on each object individually in camera frame. This object-centric paradigm can only handle skills relevant to only one target object and assumes a static camera. To overcome this, we propose to convert pointcloud observations and actions to a shared 3D space and apply group-wise data augmentation, which links each skill to a group of objects rather than a single target to maintain necessary object combination for complicated skills. It also leverages a backtracking mechanism to augment the 3D observation without disturbing the causal order of each operation. Moreover, since pointcloud from 3D sensor (e.g. RGB-D camera) is incomplete, large transformation will make the augmented 3D observation unreasonable. E.g., points that should be observed are missing, while points that should be occluded exist. To this end, we further present camera-aware 3D post-processing to ensure the 3D observation after augmentation obey the distribution of 3D sensor. Through extensive evaluation on different camera setups and real-world platforms, we validate R2RGen's effectiveness on one-shot imitation learning and its strong scalability with additional demonstrations, different vision foundation models and task settings.

II. Related Work

Imitation learning for robotic manipulation: With the development of robotic data collection system and model architecture, using imitation learning to train visuomotor policies [48, 8, 32, 35, 38, 43, 44], which end-to-end predict actions from visual observation, becomes a promising way to learn dexterous manipulation skills from human demonstrations. Inspired by the success of large language models (LLM) and vision language models (VLM), there are multiple recent works exploring scaling up the model size and data amount to train generalist robot policies with imitation learning. One promising approach for training such generalist are vision-language-action models (VLA) [2, 24, 1, 3, 46, 5, 27], which finetunes VLM pre-trained on internet-scale data for robot control. Though being flexible, imitation learning methods are data-intensive due to the lack of skill priors. In fact, to achieve strong generalization ability, visuomotor policy requires large amount of data for training / finetuning. Since largest embodied datasets [31, 22] are still much smaller than the counterparts in vision and language fields [10, 33], current works manage to solve this problem with advanced data collection systems like UMI [9, 15] and VR [6, 11], or empirical studies on data scaling [49, 29, 45] and data selection [16, 47] techniques.

Data generation for visuomotor policy: In order to train generalized manipulation visuomotor policy with less human labor, automatic data generation has been paid increased attention in recent years. A branch of works [18, 36, 37, 20] utilize the common knowledge from LLM / VLM and privileged information from simulatorfor zero-shot task and motion planning. To improve data quality, some works generate robotic data from human demonstration video [12, 26, 42], which fully exploits structured skill primitives from human to acquire reasonable data. However, these methods still rely on vision foundation models [23, 21] to estimate and track poses of hand and object / part, which may not be accurate enough. Different from generating robotic data in robot-free manner, MimicGen [30] and its follow-up works [17, 14, 19] expand real-world demonstrations acquired from teleoperation by synthesizing different execution plans in simulator. These methods work well for simulation, but suffer from time-consuming on-robot rollouts to acquire real-world observation-action pairs. More recently, DemoGen [41] proposes to apply augmentation on real-world 3D visual input as well as the trajectory. By using 3D policy, DemoGen directly takes the augmented 3D pointcloud as input, thus being simulator- and rendering-free. The real-to-real generation paradigm is efficient and plug-and-play without simulator setup, but currently DemoGen struggles on strong input assumption and visual mismatch problems which severely hinder its application.

III. Approach

A. Problem Statement

Visuomotor policy learning: Robotic manipulation task can be modeled as a Partially Observable Markow Decision

Process (POMDP) with visuomotor policy $\pi : \mathcal{O} \mapsto \mathcal{A}$, which defines a function that maps current RGB-D observation $o_t \in \mathcal{O}$ to the robot's action $a_t \in \mathcal{A}$. $o_t = (I_t, P_t)$, where I_t is RGB observation and P_t is the pointcloud in camera coordinate system lifted from depth observation and camera intrinsics. To train this policy, a large dataset of demonstrations $\mathcal{D} = \{o_1^i, a_1^i, ..., o_{H_i}^i, a_{H_i}^i\}_{i=1}^N$ should be collected. To reduce human labor, we aim to improve data efficiency by generating spatially diverse data $\mathcal{D}'$ with only one human-collected source demonstration $D_s \in \mathcal{D}$. Formally written as:

$$\mathcal{D}' = \{D_s, D_g^1, D_g^2, ..., D_g^N\}, \quad \{D_g^i\}_{i=1}^N = \text{R2RGen}(D_s) \tag{1}$$

It is expected that we can train a spatially generalized visuomotor policy purely from $\mathcal{D}'$. As a real-to-real framework, we directly augment the 3D observation as well as the action trajectory to generate diverse observation-action pair. Therefore, π should be a 3D policy that directly takes in pointcloud as visual input. We opt for iDP3 [44] as our policy, which consumes the egocentric pointcloud P_t without requirement on camera pose.

Assumptions: We make the following assumptions: (1) The visuomotor policy only predicts the actions of robotic arm. Although we support manipulation with different base positions, the mobile base remains fixed during each individual task execution. (2) Similar to [30], the actions in a demonstration can be treated as a sequence of continuous end-effector pose and discrete gripper state. Formally, $a_t = (\mathbf{A}_t^{ee}, a_t^{grip})$, where $\mathbf{A}_t^{ee}$ is SE(3) end-effector pose.

Definition of coordinate system: The original end-effector actions are defined in the robot's base coordinate system (BCS), with the arm base $\mathbf{B}_a$ as its origin. We introduce a world coordinate system (WCS) for the scene and redefine actions $\mathbf{A}_t^{ee}$ by transforming them into this unified frame. The camera frame used for observation is designated as the robot-observation coordinate system (RoCS), which may differ between data collection (RoCS_{DC}) and deployment (RoCS_{DP}).

We support three camera setups: (1) data collection and deployment using an exterior camera which is fixed along with the robot's base; (2) data collection with an exterior camera, while deploying using a wrist camera mounted on the end-effector; (3) data collection and deployment using a wrist camera. Note that the SE(3) end-effector action $\mathbf{A}_t^{ee}$ naturally defines the transformation from WCS to the wrist camera frame, since the wrist camera is rigidly attached to the end-effector.

Overview of R2RGen: R2RGen employs a three-stage data generation pipeline to extend the source demonstration:

$$\text{RoCS}_{DC} \xrightarrow{SP} \text{WCS} \xrightarrow{Aug} \text{WCS} \xrightarrow{CAP} \text{RoCS}_{DP} \tag{2}$$

where the source demonstration is first converted from RoCS_{DC} to WCS with source pre-processing (SP). Then pointcloud observations and actions are jointly augmented (Aug) in WCS. Finally R2RGen applies camera-aware processing (CAP) to convert the pointcloud observation back to RoCS_{DP}.

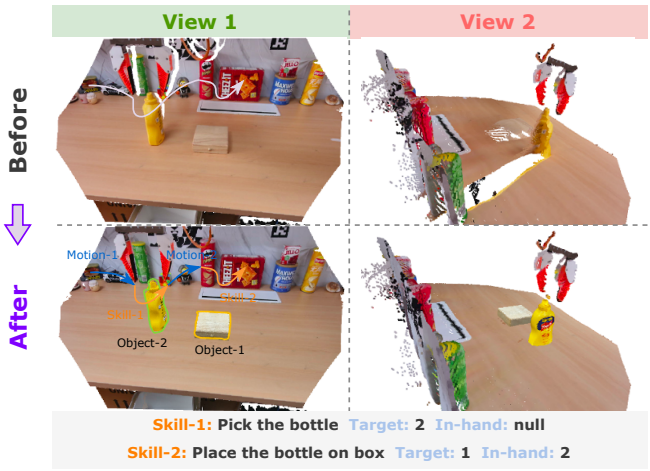

Fig. 2: Pre-processing results. The 3D scene is parsed into complete objects, environment and robot's arm. The trajectory is parsed into interleaved motion and skill segments.

B. Source Demonstration Pre-processing

Given the source demonstration $D_s = \{o_1, a_1, ..., o_{H_s}, a_{H_s}\}$, we need to fully parse the 3D observations $\{P_t\}$ into editable composition of objects, and parse the action trajectory $\{a_t\}$ into motion and skill segments, which facilitates further data generation. Formally, the goal of this pre-processing stage is: (1) **Scene parsing**. Assume the task of D_s involves K relevant objects. So for each timestamp t, the pointcloud observation o_t should be segmented into K object pointclouds $\{P_t^1, ..., P_t^K\}$, one environment pointcloud P_t^e and the pointcloud P_t^a of robot's arm. (2) **Trajectory parsing**. Same with previous definition [14, 41] of motion and skill, $\{a_t\}$ should be segmented into sequence of interleaved motion / skill segments $\{a_1, ..., a_{m_1}\}$ (motion-1), $\{a_{m_1+1}, ..., a_{s_1}\}$ (skill-1), $\{a_{s_1+1}, ..., a_{m_2}\}$ (motion-2), $\{a_{m_2+1}, ..., a_{s_2}\}$ (skill-2), etc. The objects relevant to each skill should also be known.

Scene parsing: We can segment the K objects in the first frame I_1 and track through the whole video. The 2D masks can be projected into 3D to obtain pointcloud of each object. However, only segmenting each object from $\{P_t\}$ is insufficient due to the incompleteness of RGB-D observation. For instance, the pointcloud of a cup may only represent the side facing the camera, leaving the occluded side unobserved. As a result, the observation will be incomplete when generating demonstrations from a novel viewpoint relative to the object. To this end, in addition to segmentation, we further complete the object pointclouds $\widetilde{P}_t^i = \mathcal{C}(P_t^i)$. We adopt a template-based 3D object tracking system[1] [39] to achieve this, which generates complete object pointclouds $\{\widetilde{P}_t^1, ..., \widetilde{P}_t^K\}_{t=1}^{H_s}$ for all frames given K object masks in the first frame I_1. Note that the object pointclouds should be converted to WCS. If the data is collected with an exterior camera, we simply use a fixed transformation $\mathbf{T}_e$ from exterior camera to WCS. When a wrist

[1]We discuss utilization of other vision foundation models for tracking and completion in Section IV-E as an extension.

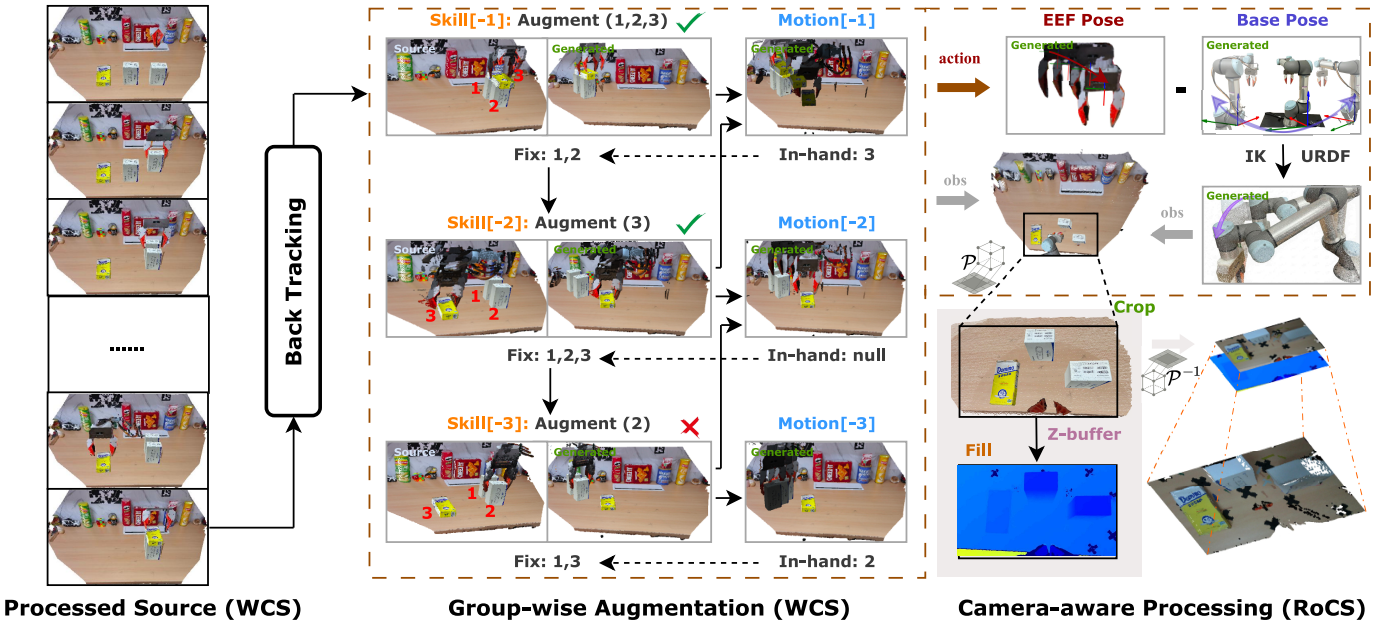

Fig. 3: The pipeline of *R2RGen*. Given processed source demonstration, we backtrack skills and apply group-wise augmentation to maintain the spatial relationships among target objects, where a fixed object set is maintained to judge whether the augmentation is applicable. Then motion planning is performed to generate trajectories that connect adjacent skills. After augmentation, we perform camera-aware processing to make the pointclouds follow distribution of RGB-D camera. The solid arrows indicate the processing flow, while the dashed arrows indicate the updating of fixed object set.

camera is used, we apply $(\mathbf{A}_t^{ee})^{-1}$ to convert the pointcloud of corresponding timestamp from wrist frame to WCS.

We also need to acquire the complete environment pointcloud $\widetilde{P}^e$. Since $\widetilde{P}^e$ is static regardless of time, we can simply remove the K objects and scan the environment in WCS before we collect D_s. The pointcloud of robot's arm is ignored at this stage. We just measure $\mathbf{B}_a$ of BCS, including the base position and orientation, which will be used to generate complete arm pointcloud according to robot dynamics in the following stage.

Trajectory parsing: We introduce a lightweight annotation system. The interface plays the RGB video $\{I_1, ..., I_{H_s}\}$ and asks the annotator to label the start frame and end frame of each skill. The intermediate trajectories between two skill segments are classified as motion segments. Apart from annotating skill segments, the annotator also specifies the object IDs (ranging from 1 to K) associated with each skill. Specifically, for every skill, both target object IDs and in-hand object ID are provided: the in-hand object refers to the item being held by the gripper during the skill execution (if any), while the target objects denote the entity with which the gripper interacts. IDs may be null, a single object, or multiple objects, depending on the task structure. The entire process uses only the RGB video as input and requires less than 60 seconds per demonstration, making it efficient and minimally labor-intensive—particularly suitable for settings with very few source demonstrations.

Figure 2 illustrates the parsed results of a source demonstration. More details about the object parsing and trajectory parsing systems can be found in Appendix A1.

C. Group-wise Data Augmentation

To synthesize new demonstrations, we randomly augment the location and rotation of objects and environment to acquire new scene configurations, and generate the corresponding action trajectories. Previous pointcloud-based data generation methods [41] assume only one target object per skill and transform skill segments solely based on that object's transformation. Motion segments are then generated using a planner to connect adjacent skills into a complete trajectory. However, this approach fails when a skill involves multiple target objects whose spatial relationships must be preserved. For instance, for task "build a bridge" shown in Figure 3, placing the bridge deck (object-3) requires the two bridge piers (object-1 and object-2) to be positioned at a specific relative distance. Independently augmenting each pier would disrupt this relationship and prevent successful execution of the final skill. To support arbitrary interaction modes, we propose a group-wise augmentation strategy that maintains structural constraints among multiple objects during data generation.

Group-wise backtracking: Instead of modeling skills as object-centric, we assign each skill to a group of objects consisting of the annotated target and in-hand objects. All objects within the same group undergo identical geometric transformations (i.e., the same translation and rotation). Augmentations are performed in a backtracking manner to avoid causal conflicts among object states. Formally, we begin from the last skill (skill-n) $\{a_{m_n+1}, ..., a_{s_n}\}$. Denote $O_n = O_n^{tar} \cup O_n^{hand}$ is the ID set of target and in-hand objects of current skill, and $\overline{O}_n = \varnothing$ is the ID set of fixed objects (i.e.,

objects cannot be augmented). We decide whether to augment current group according to $\overline{O}_n \cap O_n$. If the intersection is $\varnothing$, we randomly sample a transformation matrix $\mathbf{T}_n \in \mathbb{R}^{4\times4}$ to apply XY-plane translation and Z-axis rotation on group O_n (XY plane is fitted through the tabletop point cloud). Otherwise this group is fixed at current time and we cannot augment it. After applying the group transformation, the fixed object set is updated as follows:

$$\overline{O}_{n-1} = (\overline{O}_n \cup O_n^{tar}) \setminus O_n^{hand} \tag{3}$$

where current group is appended into the fixed set to maintain spatial relationships, while in-hand object is released since its state before grasped is independent of current skill's constraints. We then proceed to skill-$(n-1)$ and repeat above operations until all skills are traversed.

Skill augmentation: For skill-i, if the corresponding group is not fixed, we apply transformation $\mathbf{T}_i$ to augment the end-effector's pose while remaining the gripper state:

$$\hat{\mathbf{A}}_t^{ee} = \mathbf{A}_t^{ee} \cdot \mathbf{T}_i, \quad \hat{a}_t^{grip} = a_t^{grip}, \quad \forall\, t \in [m_i+1, s_i] \tag{4}$$

Then the pointclouds of objects $\widetilde{P}_t^k$ $(k \in O_i)$ are transformed with $\mathbf{T}_i$ in the same way.

Motion augmentation: For motion-i, we apply motion planning to generate a trajectory that starts at the end of skill-$(i-1)$ and ends at the beginning of skill-i:

$$\hat{\mathbf{A}}_{t_1:t_2}^{ee} = \texttt{MotionPlan}(\hat{\mathbf{A}}_{s_{i-1}}, \hat{\mathbf{A}}_{m_i+1}), \quad \hat{a}_{t_1:t_2}^{grip} = a_{t_1:t_2}^{grip},$$
$$t_1 = s_{i-1}+1, \quad t_2 = m_i \tag{5}$$

Then the pointclouds of in-hand objects $\widetilde{P}_t^k$ $(k \in O_i^{hand})$ are transformed with relative pose transformation $(\mathbf{A}_t^{ee})^{-1} \cdot \hat{\mathbf{A}}_t^{ee}$.

Base augmentation: We augment the base position and orientation $\mathbf{B}_a$ to $\hat{\mathbf{B}}_a$ using transformation $\mathbf{T}_b$, which simulates the viewpoint change of robot. With the augmented end-effector pose and arm base in WCS, the joint angles of robot's arm can be solved using Inverse Kinematics in BCS. In this way, the complete arm pointcloud $\widetilde{P}_t^a$ can be acquired through:

$$\widetilde{P}_t^a = \mathcal{U}(\hat{\mathbf{B}}_a, \texttt{IK-Solver}(\hat{\mathbf{B}}_a, \hat{\mathbf{A}}_t^{ee})) \tag{6}$$

where $\mathcal{U}$ converts the robotic arm's URDF model into corresponding pointcloud in BCS based on joint angles solved by IK, and then transforms this pointcloud to WCS using $\hat{\mathbf{B}}_a$.

Finally, we obtain the augmented pointcloud $\hat{P}_t$ as the combination of object, environment and arm pointclouds. An algorithm diagram of the group-wise augmentation pipeline is detailed in Appendix A2. For bimanual manipulation, we additionally introduce constraints to ensure the generated demonstrations executable. Refer to Appendix A3 for more details on bimanual tasks.

D. Camera-aware 3D Post-processing

After training on generated demonstrations $\mathcal{D}'$, the 3D policy is deployed in real-world with RGB-D camera as input sensor. Therefore, the pointcloud observation in $\mathcal{D}'$ should be similar to the raw RGB-D observation. We first convert the augmented pointcloud $\hat{P}_t$ in WCS to $\hat{P}_t^{trans}$ in RoCS$_{DP}$. If the deployed camera is an exterior camera, as this camera is fixed along with base, we apply $(\mathbf{T}_e)^{-1} \cdot \mathbf{T}_b$ to transform the coordinates. When a wrist camera is adopted for deployment, we apply $\hat{\mathbf{A}}_t^{ee}$ to convert $\hat{P}_t$ to the wrist camera frame.

After coordinate transformation, there are still two main differences between raw observation and $\hat{P}_t^{trans}$: (1) $\hat{P}_t^{trans}$ is over-complete. While for a given perspective of RGB-D camera, raw pointcloud converted from depth image is complete only at this viewpoint. (2) Due to the augmentation on robot's base, the spatial distribution of $\hat{P}_t^{trans}$ may be shifted. To solve this, we propose a camera-aware 3D post-processing to adjust the distribution of generated pointcloud observations:

$$\hat{P}_t^{adjust} = \mathcal{P}^{-1}(\texttt{Fill}(\texttt{Z-buffer}(\texttt{Crop}(\{(u_i, v_i, d_i)\})))),$$
$$\{(u_i, v_i, d_i)\} = \mathcal{P}(\hat{P}_t^{trans}) \tag{7}$$

where $\mathcal{P}$ projects 3D pointcloud to image plane with camera intrinsics. The `Crop` operation removes pixels $\{(u_i, v_i, d_i) | u_i < 0 \ or \ u_i \geq W \ or \ v_i < 0 \ or \ v_i \geq H\}$ which are out of image boundary. `Z-buffer` processes overlapped pixels and only keep one pixel with smallest depth value, which removes hidden points at current viewpoint. In practice, we notice the density of pointclouds may not be so high, making front surface unable to hide all points behind. Therefore, we propose a patch-wise Z-buffer operator, where each point with small depth value can hide deeper points in a r-radius neighborhood on image plane. Since the environment is augmented, pixels near the image boundary may be empty (i.e., no point is projected to these pixels). So we `Fill` the empty pixels by either shrinking the image size or expanding the environment pointcloud, which we detail in Section IV-D. Finally, after post-processing in the image plane, we project the pixels back to camera coordinate system. The adjusted pointcloud $\hat{P}_t^{adjust}$ well matches the distribution of RGB-D camera and can be directly fed into our 3D policy during training.

IV. Experiment

In this section, we evaluate R2RGen through extensive real-world experiments, demonstrating its effectiveness on one-shot imitation learning and how it scales up with more source demonstrations. We also conduct comprehensive ablation studies to explore the optimal design choices. Furthermore, we show R2RGen can be extended to facilitate appearance generalization and mobile manipulation.

A. Experimental Setup

Policy. We select iDP3 [44] as the visuomotor policy, which takes egocentric pointclouds (i.e., pointclouds in camera coordinate system) and proprioception state as inputs without requirement on camera pose and calibration. Details on training iDP3 on each task are described in Appendix A4.

Hardware. We utlize two robot platforms: single-arm and bimanual. The single-arm platform consists of a 7-DoF UR5 arm equipped with a parallel jaw gripper and a mobile base. An ORBBEC femto bolt camera can be either rigidly affixed via a mounting bracket to the mobile base as exterior camera,

TABLE I: Real-world evaluation of *R2RGen* for spatial generalization. Success rate is reported.

	Single-Arm Task						Dual-Arm Task		Averaged
	Open-Jar	Place-Bottle	Pot-Food	Hang-Cup	Stack-Brick	Build-Bridge	Grasp-Box	Store-Item	
1 Source	3.1	3.1	3.1	3.1	3.1	3.1	4.2	4.2	3.4
+DemoGen	18.8	15.6	–	–	–	–	16.7	16.7	–
+R2RGen	**50.0**	**50.0**	**37.5**	**34.4**	**43.8**	**34.4**	**41.7**	**33.3**	**40.3**
10 Source	56.3	34.3	9.4	15.6	9.4	9.4	25.0	20.8	22.5
25 Source	78.1	53.1	21.9	43.8	40.6	28.1	29.2	33.3	41.0
40 Source	87.5	68.8	28.1	43.8	50.0	43.8	37.5	41.7	50.2

TABLE II: Real-world evaluation of *R2RGen* on exterior-wrist and wrist-wrist setup. Success rate is reported.

	Exterior-Wrist				Wrist-Wrist			
	Pot-Food	Hang-Cup	Stack-Brick	Build-Bridge	Pot-Food	Hang-Cup	Stack-Brick	Build-Bridge
1 Source	–	–	–	–	3.1	3.1	3.1	3.1
+R2RGen	**50.0**	**37.5**	**37.5**	**31.3**	**40.6**	**37.5**	**37.5**	**28.1**
10 Source	–	–	–	–	9.4	9.4	6.3	9.4
25 Source	–	–	–	–	21.9	31.3	31.3	28.1
40 Source	–	–	–	–	50.0	43.8	37.5	41.7

or attached to the end-effector as wrist camera, which provides the RGB-D observations. The bimanual platform adheres to the MobileAloha architecture [13], employing dual AgileX PiPER arms integrated with HexFellow omnidirectional mobile base. A head-mounted RGB-D camera (exterior camera) provides egocentric perception. See Appendix B for further details.

Tasks and evaluation. We design 8 representative tasks for evaluation, including 2 simple tasks (Open-Jar, Place-Bottle), 4 complex tasks (Pot-Food, Hang-Cup, Stack-Brick, Build-Bridge) and 2 bimanual tasks (Grasp-Box, Store-Item). We compare with DemoGen [41] on the two simple single-arm tasks and two bimanual tasks. While for other tasks involving complex spatial relationships among objects, DemoGen fails to generate reasonable data. We evaluate different methods on diverse objects' locations and rotations as well as robot's viewpoints, including a portion of out-of-distribution samples not encountered during training. Refer to Appendix C for detailed task definition and evaluation protocol.

Camera settings. We conduct experiments on three camera setups mentioned in Section III-A, which we denote as exterior-exterior, exterior-wrist and wrist-wrist to specify the camera used during data collection and deployment, respectively. Since exterior-exterior is the mainstream setting in prior works, we compare R2RGen with baseline method on all 8 tasks for this setting. For exterior-wrist and wrist-wrist setting, we conduct experiments on the 4 complex single-arm tasks, where the source demonstration is collected using exterior/wrist camera, while policy is evaluated with wrist camera. By default, all experiments employ the exterior–exterior camera setup unless otherwise specified.

B. Results: One-shot Imitation Learning

For one-shot imitation learning, we only collect one human demonstration for R2RGen to generate new data. Similar to DemoGen, we replay the collected human demonstration twice to acquire diverse pointcloud observations, which significantly reduce the impact of sensor noise. The three pointcloud trajectories are all used for 3D data generation. Then we compare the policy trained on purely generated data with ones trained on different number of human demonstrations, as shown in Table I. When trained with only one human demonstration, the policy succeeds merely at the demonstrated pose but fails to generalize. Both DemoGen and R2RGen improve its performance. It is shown that R2RGen consistently outperforms DemoGen across all tasks, even though DemoGen crops pointclouds of background while we do not. This advantage primarily stems from our scene parsing and camera-aware processing techniques, which enable generating high-quality data under large variations in object location / rotation and robot viewpoint. In contrast, DemoGen suffers from significant visual mismatch under such challenging evaluation. R2RGen achieves performance comparable to policies trained with 25 human demonstrations, and even surpasses 40 demonstrations on several difficult tasks, which validates its effectiveness on spatial generalization.

According to Table II, R2RGen also works well on exterior-wrist and wrist-wrist settings. Naive cross-camera deployment—training on exterior views but testing on wrist views—leads to catastrophic failure, underscoring the severe domain gap. By leveraging 3D point cloud transformations between camera and world coordinates, R2RGen eliminates this discrepancy and achieves robust performance gains.

C. Results: Performance-annotation Tradeoff

We further study how R2RGen scales up with more source demonstrations. For each task, we run R2RGen to generate data from 1, 2, 3 and 5 human demonstrations respectively. We then report the policy performance boost w.r.t. the increase of synthetic data under different numbers of source demonstrations. As shown in Figure 4, the success rate gradually saturates as

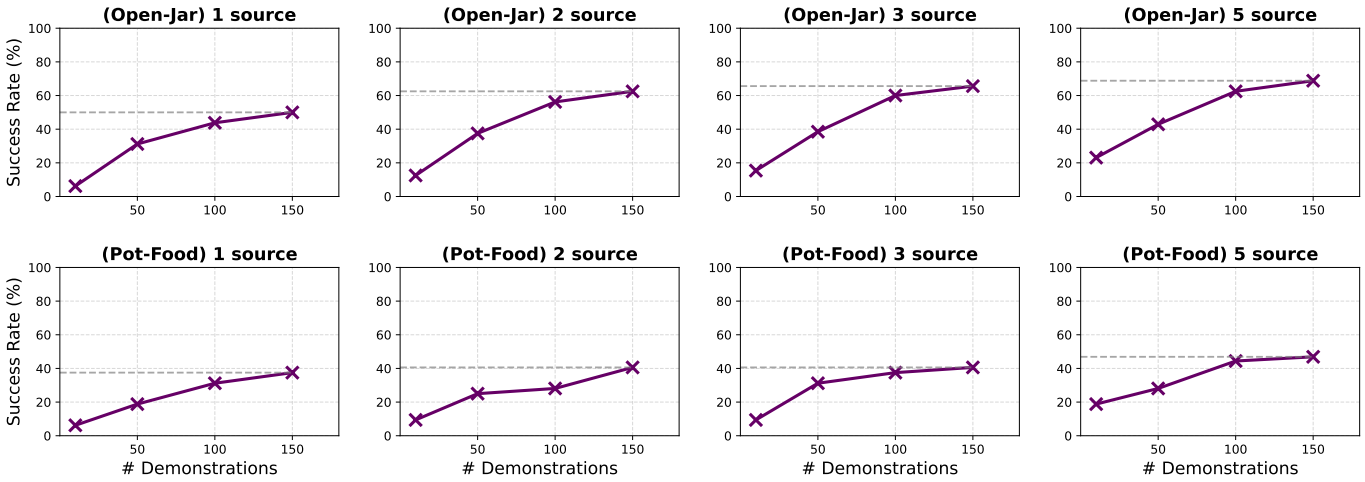

Fig. 4: Effects of the number of generated demonstrations and source demonstrations on the final performance of R2RGen.

TABLE III: Effects of the pointcloud processing.

Method	SR
Remove pointcloud completion	34.4
Remove environment pointcloud	37.5
Remove arm pointcloud	15.6
Remove base augmentation	21.9
R2RGen	**50.0**

TABLE IV: Effects of camera-aware processing.

Method	SR
Remove `Crop` operation	34.4
Remove `Z-buffer` operation	15.6
Remove `Fill` operation	28.1
R2RGen	**50.0**

the number of generated demonstrations increases—a trend also observed with human-collected data. This behavior is due to the limited capacity of the iDP3 policy, which uses a lightweight PointNet encoder. Further scaling beyond this plateau would require policies with larger 3D backbones. We also note that more source demonstrations lead to a higher saturation performance, demonstrating R2RGen's ability to effectively leverage additional data for improved results.

D. Ablation Study

We carefully design ablation experiments to study the effects of each design choice. The analysis is conducted on the Place-Bottle task.

Pointcloud processing. We first study how object and environment pointclouds affect the final performance, as shown in Table III. We notice removing pointcloud completion will lead to unrealistic data under large spatial augmentation, while removing operations on environment reduces the policy's robustness to viewpoint changes.

Camera-aware processing. We then ablate camera-aware processing by removing one of the key operations at each time. As shown in Table IV, these operations play a critical role in determining final performance. For the `Fill` operation, we further compare two design choices, please refer to Appendix A5 for details.

TABLE V: Extension on non-rigid objects using SAM 3D.

	T_1-Articulated (Store-Box)	T_2-Deformable (Cover-Object)
1 Source	3.1	3.1
+R2RGen	**31.3**	**53.1**
10 Source	18.8	15.6
25 Source	31.3	37.5
40 Source	37.5	46.9

E. Extension and Application

Extension: non-rigid object. Our R2RGen is a general framework that can also handle non-rigid objects. Since currently FoundationPose [39] only supports rigid objects, we can switch to other vision foundation models for non-rigid ones. For instance, ANCSH [28] for articulated objects and GarmentNets [7] for deformable objects. More recently, SAM 3D [4] achieves generalized 3D shape generation and pose estimation given monocular RGB observation with depth map, which can perfectly serve as a unified 3D object completion model regardless of rigidness. Equipped with this powerful tool, we design two tasks T_1 and T_2 to validate the universality of R2RGen on articulated objects and deformable objects respectively, as shown in Table V. More details on T_1 and T_2 can be found in Appendix C3.

Extension: appearance generalization. Beyond spatial generalization, robotic manipulation tasks involve other forms of generalization, such as appearance generalization (i.e., adapting to novel object instances and environments) and task generalization. Among these, spatial generalization serves as the fundamental prerequisite for other generalization capabilities. Since this work focuses on single-task visuomotor policy learning, we investigate whether the spatial generalization enabled

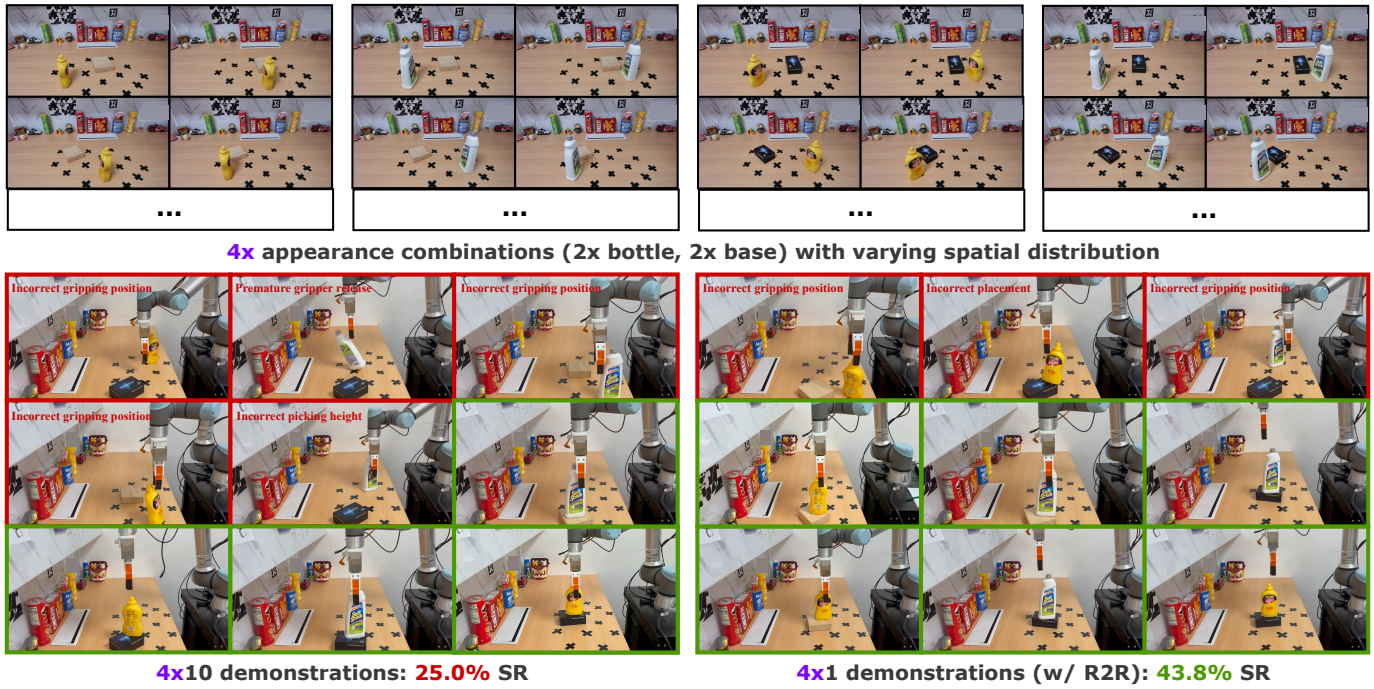

Fig. 5: Extension on appearance generalization. The spatial generalization of R2RGen can serve as a foundation to achieve other kinds of generalization with much less data.

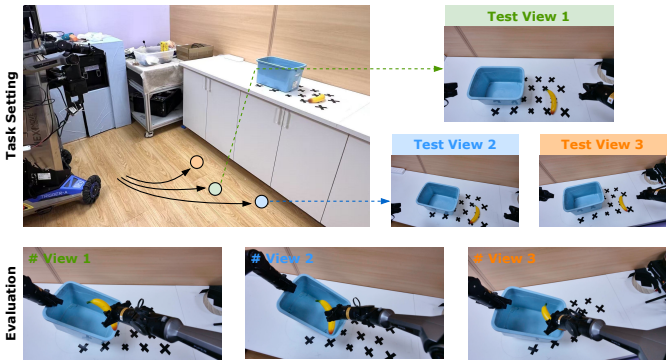

Fig. 6: Visualization of mobile manipulation results. The policy trained with R2RGen successfully generalizes to different camera views with only one human-collected demonstration.

by R2RGen can further facilitate appearance generalization. As shown in Figure 5, we design a more challenging Place-Bottle task with four distinct bottle-base appearance combinations (2 bottle types $\times$ 2 base types). We observe that achieving both appearance and spatial generalization significantly increases data demand. Even with 40 human demonstrations (10 per bottle-base pair), the policy only reaches a 25% success rate. In contrast, using R2RGen, only 1 demonstration per bottle-base pair (4 in total) is needed to achieve a success rate of 43.8%, demonstrating its efficiency in handling combined generalization challenges.

Application: mobile manipulation. R2RGen makes our 3D policy achieve strong spatial generalization across different viewpoints without camera calibration, so we can achieve mobile manipulation by simply combining a navigation system [40] and a manipulation policy trained with R2RGen. Since the termination condition of navigation is relatively loose, the robot may stop at different docking point around the manipulation area, which imposes great challenges on the manipulation policy. According to Figure 6, using iDP3 trained with R2RGen, the policy successfully generalizes to different docking points with maximum distance larger than 5cm. Different from DemoGen (DP3) which requires a careful calibration of the camera pose to crop environment pointclouds, our method directly applies on raw RGB-D observations during both data generation and policy training / inference stages, which is more practical in real-world applications.

V. Concluding Remark

R2RGen introduces a real-to-real 3D data generation framework that generalizes beyond prior pointcloud-based methods such as DemoGen [41]. Specifically, it supports mobile manipulators, raw sensor inputs, different camera setups, arbitrary numbers of objects and diverse interaction modes, overcoming key limitations of existing approaches. With only one human demonstration, our method directly augments the 3D observation as well as the action trajectories to generate large amount of pointcloud-action pairs, which are utilized to train a spatially generalized 3D policy. Extensive experiments on multiple real-world tasks validate the effectiveness of R2RGen. We further extend its application to different vision foundation models, appearance generalization and mobile manipulation scenarios, demonstrating its strong generalizability and scalability for

broad real-world deployment.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant 624B2076, Grant 62125603, and Grant 62321005, and in part by the Beijing Natural Science Foundation under Grant No. L247009.

References

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

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

[3] Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, et al. Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024.

[4] Xingyu Chen, Fu-Jen Chu, Pierre Gleize, Kevin J Liang, Alexander Sax, Hao Tang, Weiyao Wang, Michelle Guo, Thibaut Hardin, Xiang Li, et al. Sam 3d: 3dfy anything in images. *arXiv preprint arXiv:2511.16624*, 2025.

[5] An-Chieh Cheng, Yandong Ji, Zhaojing Yang, Zaitian Gongye, Xueyan Zou, Jan Kautz, Erdem Bıyık, Hongxu Yin, Sifei Liu, and Xiaolong Wang. Navila: Legged robot vision-language-action model for navigation. *arXiv preprint arXiv:2412.04453*, 2024.

[6] Xuxin Cheng, Jialong Li, Shiqi Yang, Ge Yang, and Xiaolong Wang. Open-television: Teleoperation with immersive active visual feedback. *arXiv preprint arXiv:2407.01512*, 2024.

[7] Cheng Chi and Shuran Song. Garmentnets: Category-level pose estimation for garments via canonical space shape completion. In *ICCV*, pages 3324–3333, 2021.

[8] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *IJRR*, 2023.

[9] Cheng Chi, Zhenjia Xu, Chuer Pan, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, Russ Tedrake, and Shuran Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. *arXiv preprint arXiv:2402.10329*, 2024.

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. Ieee, 2009.

[11] Runyu Ding, Yuzhe Qin, Jiyue Zhu, Chengzhe Jia, Shiqi Yang, Ruihan Yang, Xiaojuan Qi, and Xiaolong Wang. Bunny-visionpro: Real-time bimanual dexterous teleoperation for imitation learning. *arXiv preprint arXiv:2407.03162*, 2024.

[12] Jiafei Duan, Yi Ru Wang, Mohit Shridhar, Dieter Fox, and Ranjay Krishna. Ar2-d2: Training a robot without a robot. *arXiv preprint arXiv:2306.13818*, 2023.

[13] Zipeng Fu, Tony Z Zhao, and Chelsea Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. *arXiv preprint arXiv:2401.02117*, 2024.

[14] Caelan Garrett, Ajay Mandlekar, Bowen Wen, and Dieter Fox. Skillmimicgen: Automated demonstration generation for efficient skill learning and deployment. *arXiv preprint arXiv:2410.18907*, 2024.

[15] Huy Ha, Yihuai Gao, Zipeng Fu, Jie Tan, and Shuran Song. Umi on legs: Making manipulation policies mobile with manipulation-centric whole-body controllers. *arXiv preprint arXiv:2407.10353*, 2024.

[16] Joey Hejna, Chethan Bhateja, Yichen Jiang, Karl Pertsch, and Dorsa Sadigh. Re-mix: Optimizing data mixtures for large scale imitation learning. *arXiv preprint arXiv:2408.14037*, 2024.

[17] Ryan Hoque, Ajay Mandlekar, Caelan Garrett, Ken Goldberg, and Dieter Fox. Intervengen: Interventional data generation for robust and data-efficient robot imitation learning. In *IROS*, pages 2840–2846. IEEE, 2024.

[18] Pu Hua, Minghuan Liu, Annabella Macaluso, Yunfeng Lin, Weinan Zhang, Huazhe Xu, and Lirui Wang. Gensim2: Scaling robot data generation with multi-modal and reasoning llms. *arXiv preprint arXiv:2410.03645*, 2024.

[19] Zhenyu Jiang, Yuqi Xie, Kevin Lin, Zhenjia Xu, Weikang Wan, Ajay Mandlekar, Linxi Fan, and Yuke Zhu. Dexmimicgen: Automated data generation for bimanual dexterous manipulation via imitation learning. *arXiv preprint arXiv:2410.24185*, 2024.

[20] Pushkal Katara, Zhou Xian, and Katerina Fragkiadaki. Gen2sim: Scaling up robot learning in simulation with generative models. In *ICRA*, pages 6672–6679. IEEE, 2024.

[21] Justin Kerr, Chung Min Kim, Mingxuan Wu, Brent Yi, Qianqian Wang, Ken Goldberg, and Angjoo Kanazawa. Robot see robot do: Imitating articulated object manipulation with monocular 4d reconstruction. *arXiv preprint arXiv:2409.18121*, 2024.

[22] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

[23] Chung Min Kim, Mingxuan Wu, Justin Kerr, Ken Goldberg, Matthew Tancik, and Angjoo Kanazawa. Garfield: Group anything with radiance fields. In *CVPR*, pages 21530–21539, 2024.

[24] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla:

An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[25] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *ICCV*, pages 4015–4026, 2023.

[26] Marion Lepert, Jiaying Fang, and Jeannette Bohg. Phantom: Training robots without robots using only human videos. *arXiv preprint arXiv:2503.00779*, 2025.

[27] Qixiu Li, Yaobo Liang, Zeyu Wang, Lin Luo, Xi Chen, Mozheng Liao, Fangyun Wei, Yu Deng, Sicheng Xu, Yizhong Zhang, et al. Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation. *arXiv preprint arXiv:2411.19650*, 2024.

[28] Xiaolong Li, He Wang, Li Yi, Leonidas J Guibas, A Lynn Abbott, and Shuran Song. Category-level articulated object pose estimation. In *CVPR*, pages 3706–3715, 2020.

[29] Fanqi Lin, Yingdong Hu, Pingyue Sheng, Chuan Wen, Jiacheng You, and Yang Gao. Data scaling laws in imitation learning for robotic manipulation. *arXiv preprint arXiv:2410.18647*, 2024.

[30] Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations. In *Conference on Robot Learning*, pages 1820–1864. PMLR, 2023.

[31] 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 *ICRA*, pages 6892–6903. IEEE, 2024.

[32] Aaditya Prasad, Kevin Lin, Jimmy Wu, Linqi Zhou, and Jeannette Bohg. Consistency policy: Accelerated visuomotor policies via consistency distillation. *arXiv preprint arXiv:2405.07503*, 2024.

[33] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *NeurIPS*, 35: 25278–25294, 2022.

[34] Hengkai Tan, Xuezhou Xu, Chengyang Ying, Xinyi Mao, Songming Liu, Xingxing Zhang, Hang Su, and Jun Zhu. Manibox: Enhancing spatial grasping generalization via scalable simulation data generation. *arXiv preprint arXiv:2411.01850*, 2024.

[35] Dian Wang, Stephen Hart, David Surovik, Tarik Kelestemur, Haojie Huang, Haibo Zhao, Mark Yeatman, Jiuguang Wang, Robin Walters, and Robert Platt. Equivariant diffusion policy. *arXiv preprint arXiv:2407.01812*, 2024.

[36] Lirui Wang, Yiyang Ling, Zhecheng Yuan, Mohit Shridhar, Chen Bao, Yuzhe Qin, Bailin Wang, Huazhe Xu, and Xiaolong Wang. Gensim: Generating robotic simulation tasks via large language models. *arXiv preprint arXiv:2310.01361*, 2023.

[37] Yufei Wang, Zhou Xian, Feng Chen, Tsun-Hsuan Wang, Yian Wang, Katerina Fragkiadaki, Zackory Erickson, David Held, and Chuang Gan. Robogen: Towards unleashing infinite data for automated robot learning via generative simulation. *arXiv preprint arXiv:2311.01455*, 2023.

[38] Zhendong Wang, Max Li, Ajay Mandlekar, Zhenjia Xu, Jiaojiao Fan, Yashraj Narang, Linxi Fan, Yuke Zhu, Yogesh Balaji, Mingyuan Zhou, et al. One-step diffusion policy: Fast visuomotor policies via diffusion distillation. In *International Conference on Machine Learning*, pages 63399–63416. PMLR, 2025.

[39] Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects. In *CVPR*, pages 17868–17879, 2024.

[40] Zhenyu Wu, Angyuan Ma, Xiuwei Xu, Hang Yin, Yinan Liang, Ziwei Wang, Jiwen Lu, and Haibin Yan. Moto: A zero-shot plug-in interaction-aware navigation for general mobile manipulation. In *CoRL*, 2025.

[41] Zhengrong Xue, Shuying Deng, Zhenyang Chen, Yixuan Wang, Zhecheng Yuan, and Huazhe Xu. Demogen: Synthetic demonstration generation for data-efficient visuomotor policy learning. *arXiv preprint arXiv:2502.16932*, 2025.

[42] 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. *arXiv preprint arXiv:2505.09601*, 2025.

[43] Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. *Robotics: Science and Systems XX*, 2024.

[44] Yanjie Ze, Zixuan Chen, Wenhao Wang, Tianyi Chen, Xialin He, Ying Yuan, Xue Bin Peng, and Jiajun Wu. Generalizable humanoid manipulation with 3d diffusion policies. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2873–2880. IEEE, 2025.

[45] Lihan Zha, Apurva Badithela, Michael Zhang, Justin Lidard, Jeremy Bao, Emily Zhou, David Snyder, Allen Z Ren, Dhruv Shah, and Anirudha Majumdar. Guiding data collection via factored scaling curves. *arXiv preprint arXiv:2505.07728*, 2025.

[46] Jiazhao Zhang, Kunyu Wang, Rongtao Xu, Gengze Zhou, Yicong Hong, Xiaomeng Fang, Qi Wu, Zhizheng Zhang, and He Wang. Navid: Video-based vlm plans the next step for vision-and-language navigation. *arXiv preprint arXiv:2402.15852*, 2024.

[47] Yu Zhang, Yuqi Xie, Huihan Liu, Rutav Shah, Michael Wan, Linxi Fan, and Yuke Zhu. Scizor: A self-supervised approach to data curation for large-scale imitation learning. *arXiv preprint arXiv:2505.22626*, 2025.

[48] Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with

low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

[49] Tony Z Zhao, Jonathan Tompson, Danny Driess, Pete Florence, Kamyar Ghasemipour, Chelsea Finn, and Ayzaan Wahid. Aloha unleashed: A simple recipe for robot dexterity. *arXiv preprint arXiv:2410.13126*, 2024.