DROID: A Large-Scale In-The-Wild Robot Manipulation Dataset

https://droid-dataset.github.io

Alexander Khazatsky[*,1], Karl Pertsch[*,1,2], Suraj Nair[1,3], Ashwin Balakrishna[3], Sudeep Dasari[4], Siddharth Karamcheti[1], Soroush Nasiriany[5], Mohan Kumar Srirama[4], Lawrence Yunliang Chen[2], Kirsty Ellis[6], Peter David Fagan[7], Joey Hejna[1], Masha Itkina[3], Marion Lepert[1], Jason Ma[14], Patrick Tree Miller[3], Jimmy Wu[8], Suneel Belkhale[1], Shivin Dass[5], Huy Ha[1], Abraham Lee[2], Youngwoon Lee[2,16], Arhan Jain[9], Marius Memmel[9], Sungjae Park[10], Ilija Radosavovic[2], Kaiyuan Wang[11], Albert Zhan[6], Kevin Black[2], Cheng Chi[1], Kyle Hatch[3], Shan Lin[11], Jingpei Lu[11], Abdul Rehman[7], Pannag R Sanketi[12], Archit Sharma[1], Cody Simpson[3], Quan Vuong[12], Homer Walke[2], Blake Wulfe[3], Ted Xiao[12], Jonathan Yang[1], Arefeh Yavary[13], Tony Z. Zhao[1], Christopher Agia[1], Rohan Baijal[9], Mateo Guaman Castro[9], Daphne Chen[9], Qiuyu Chen[9], Trinity Chung[2], Jaimyn Drake[2], Ethan Paul Foster[1], Jensen Gao[1], David Antonio Herrera[1], Minho Heo[10], Kyle Hsu[1], Jiaheng Hu[5], Donovon Jackson[3], Charlotte Le[2], Yunshuang Li[14], Kevin Lin[1], Roy Lin[2], Zehan Ma[2], Abhiram Maddukuri[5], Suvir Mirchandani[1], Daniel Morton[1], Tony Nguyen[3], Abby O'Neill[2], Rosario Scalise[9], Derick Seale[3], Victor Son[1], Stephen Tian[1], Andrew Wang[2], Yilin Wu[4], Annie Xie[1], Jingyun Yang[1], Patrick Yin[9], Yunchu Zhang[9], Osbert Bastani[14], Glen Berseth[6], Jeannette Bohg[1], Ken Goldberg[2], Abhinav Gupta[4], Abhishek Gupta[9], Dinesh Jayaraman[14], Joseph J. Lim[10], Jitendra Malik[2], Roberto Martín-Martín[5], Subramanian Ramamoorthy[7], Dorsa Sadigh[1], Shuran Song[1,15], Jiajun Wu[1], Yuke Zhu[5], Thomas Kollar[3], Sergey Levine[2], Chelsea Finn[1]

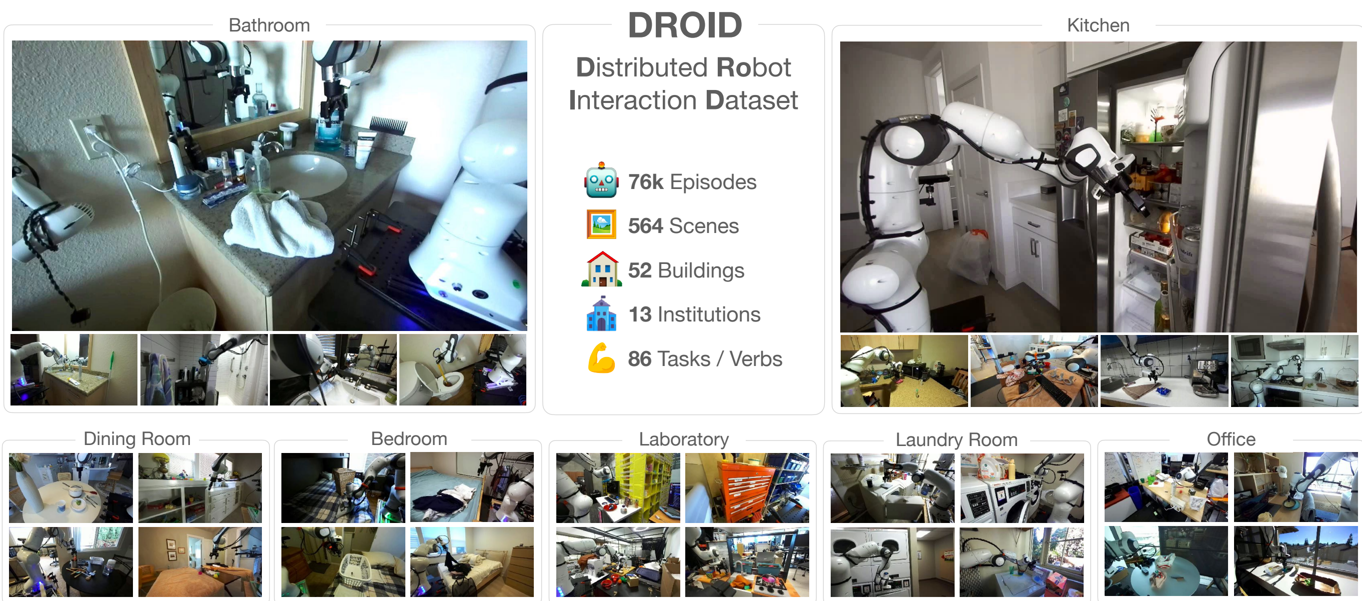

Fig. 1: We introduce DROID (**D**istributed **Ro**bot **I**nteraction **D**ataset), an "in-the-wild" robot manipulation dataset with 76k trajectories or 350 hours of interaction data, collected across 564 scenes, 86 tasks, and 52 buildings over the course of 12 months. Each DROID episode contains three synchronized RGB camera streams, camera calibration, depth information, and natural language instructions. We demonstrate that training with DROID leads to policies with higher performance, greater robustness, and improved generalization ability. We open source the full dataset, pre-trained model checkpoints, and a detailed guide for reproducing our robot setup.

***Abstract*— The creation of large, diverse, high-quality robot manipulation datasets is an important stepping stone on the path toward more capable and robust robotic manipulation policies. However, creating such datasets is challenging: collecting robot manipulation data in diverse environments poses logistical and safety challenges and requires substantial investments in hardware and human labour. As a result, even the most general robot manipulation policies today are mostly trained on data collected in a small number of environments with limited scene and task diversity. In this work, we introduce DROID (Distributed Robot Interaction Dataset), a diverse robot manipulation dataset with**

*Project Co-leads, correspondence to alexkhaz@stanford.edu, pertsch@berkeley.edu

76k demonstration trajectories or 350 hours of interaction data, collected across 564 scenes and 86 tasks by 50 data collectors in North America, Asia, and Europe over the course of 12 months. We demonstrate that training with DROID leads to policies with higher performance and improved generalization ability. We open source the full dataset, policy learning code, and a detailed guide for reproducing our robot hardware setup.

I. Introduction

A key feature of robot manipulation policies is their ability to generalize, i.e., their ability to perform a desired manipulation task under new lighting conditions, in new environments, or with new objects. Training policies that are robust to such variations is a crucial step towards the deployment of robots in everyday environments and may bring us closer to every roboticist's dream: robot models that can be downloaded and "just work" when tested on a new robot setup. A central ingredient for training such generalizable policies is diverse training data: in computer vision and natural language processing, training on large and diverse datasets scraped from the internet yields models that work in a wide range of new tasks. Similarly, in robot manipulation, a number of recent works have demonstrated that larger, more diverse robot training datasets enable us to push the envelope on policy generalization, including positive transfer to new objects, instructions, scenes, and embodiments [1, 2, 13, 30, 32, 33, 39, 50]. This suggests that an important stepping stone on the path toward more capable and robust robotic manipulation policies is the creation of large, diverse, high-quality robot manipulation datasets.

However, creating such datasets is challenging: in contrast to vision or language data, training manipulation policies typically requires robot manipulation data with recorded observations and actions, which cannot be easily scraped from the internet. Collecting robot manipulation data in diverse environments poses logistical and safety challenges when moving robots outside of controlled lab environments. Additionally, collecting data at scale requires substantial investments in hardware and human labour for supervision, particularly for collecting demonstration data. As a result, even the most general robot manipulation policies today are mostly trained on data collected in controlled, lab-like environments with limited scene and task diversity. To enable the next level of generalizable robot manipulation policy learning, the robot manipulation community needs more diverse datasets, collected across a wide range of environments and tasks.

In this work, we introduce DROID (**D**istributed **Ro**bot **I**nteraction **D**ataset), a robot manipulation dataset of unprecedented diversity (see Fig. 1). DROID consist of 76k demonstration trajectories or 350 hours of interaction data, collected across 564 scenes, 52 buildings and 86 tasks. DROID was collected by 18 research labs in North America, Asia, and Europe over the course of 12 months. To streamline distributed data collection and ensure applicability of the final dataset to a wide range of research settings, all data is collected on the same robot hardware stack based on the popular Franka Panda robot arm. Each episode contains three camera views, depth information, camera calibration, and language annotations.

In experiments across 6 tasks and 4 locations, from labs to offices and real households, we find that DROID boosts policy performance, robustness and generalizability by 20% on average over state-of-the-art approaches that leverage existing large-scale robot manipulation datasets [33]. We open-source the full DROID dataset under CC-BY 4.0 license, code for training policies using the dataset, and a detailed guide for reproducing our complete robot software and hardware setup.

II. Related Work

a) Large datasets in machine learning: The rapid progress in machine learning has been closely tied to the construction of large and diverse datasets. Examples include ImageNet [10], Kitti [16], Ego4D [17] and LAION [38] in computer vision, Common Crawl [36] and The Pile [15] in natural language processing, and ShapeNet [5] and Objaverse [8, 9] in 3D modeling. Key to their impact is their size and diversity: by enabling training on larger and more diverse data, they push the capabilities and robustness of machine learning models. With DROID we aim to continue this trend for robot manipulation and provide a large and diverse robot manipulation dataset to spur progress on generalizable policy learning.

b) Robot learning datasets: A number of prior works introduce datasets for robot learning of various sizes and diversity levels (see Table I). Broadly, these can be categorized into datasets collected autonomously via scripted and semi-random behaviors or learned agents [3, 7, 18, 24, 25, 27, 34], and datasets collected via human teleoperation [1, 2, 12, 13, 22, 30, 42, 49]. Multiple works focus on increasing dataset diversity: RH20T [13] collects data across 33 tasks in 7 table-top scenes and BridgeV2 [49] collects data in 24 scenes.[1] While these datasets increase diversity, most of their data is collected in a small number of scenes in a single research lab or building.

More recently, there has been a larger effort on pooling existing robot datasets into a coherent format, the Open X-Embodiment dataset (OXE) [33]. Albeit larger in scale than prior robot datasets, the OXE dataset still consists of individual datasets with few scenes, thus totalling around 300 scenes at the time of writing. Our goal with the DROID dataset is to significantly increase the scene diversity as well

Affiliations: [1]Stanford University; [2]University of California, Berkeley; [3]Toyota Research Institute; [4]Carnegie Mellon University; [5]University of Texas, Austin; [6]University of Montreal; [7]University of Edinburgh; [8]Princeton University; [9]University of Washington; [10]Korea Advanced Institute of Science & Technology (KAIST); [11]University of California, San Diego; [12]Google DeepMind; [13]University of California, Davis; [14]University of Pennsylvania; [15]Columbia University; [16]Yonsei University

[1]Note that prior works use various definitions for what constitutes a "task" and what constitutes a "scene". In this work, we use the number of unique *verbs* extracted from the language instructions to represent the number of tasks, which is more scalable than manually defining tasks [49] yet often more reflective of the behavior diversity than e.g., counting the number of verb-object combinations [2] (see Fig. 3 for DROID's verb distribution as an example). For scenes, we only count a scene as new if there is a substantial change of the robot's workspace, e.g., if it gets transported to a new corner of the kitchen or a new room altogether, but not if only the arrangement of objects in front of the robot or the table cloth changes.

Dataset	# Traj.	# Verbs	# Scenes	Lang. Instruct.	Cam. Calibration	Public Robot	Collection
MIME [42]	8.3k	20	1	✗	✗	✓	human teleop
RoboTurk [30]	2.1k	2	1	✗	✗	✓	human teleop
RoboNet [7]	162k	n/a	10	✗	✗	✓	scripted
MT-Opt [24, 25]	800k	2	1	✗	✗	✓	scripted & learned
BridgeData [12]	7.2k	4	12	✓	✗	✓	human teleop
BC-Z [22]	26k	3	1	✓	✗	✗	human teleop
RT-1 [2]	130k	2	2	✓	✗	✗	human teleop
RH20T [13]	13k[2]	33	7	✓	✓	✓	human teleop
RoboSet [1]	98.5k	9	11	✓	✗	✓	30% human / 70% scripted
BridgeData V2 [49]	60.1k	82	24	✓	✗	✓	85% human / 15% scripted
DobbE [39]*	5.6k	6	216	✓	n/a	(✓)	human tool-based
Open X-Embodiment [33]†	1.4M	217	311	(✓)	✗	(✓)	dataset aggregation
DROID (ours)	76k	86	**564**	✓	✓	✓	human teleop

Table I: Comparison to existing datasets for robot manipulation. "# Scenes" refers to the number of unique robot work spaces, e.g., different kitchens count as different scenes, but rearrangement of objects does not. See Section II for a detailed discussion of the definition of "Tasks" and "Scenes". DROID offers high diversity in both, the number of verbs and scenes. *non-robot, tool-based data collection, †not a dataset in itself, but *aggregation* of existing datasets, including most previous rows in this table.

as scene realism by collecting data across a wide array of real world buildings in a diverse set of geographic locations. As a result, DROID contains data from 564 scenes across 52 buildings, a substantial increase compared to any existing robot manipulation dataset.

Collecting such data "in-the-wild" is more common for robot navigation and autonomous driving [4, 16, 26, 40, 41, 46, 47, 51] and enables training of policies that generalize zero-shot to new environments and even embodiments [40, 41]. With DROID, we take a step towards enabling similar generalization for robotic *manipulation* policies. Finally, there are some works that leverage cheap, off-the-shelf tools, such as reacher-grabber tools, for data collection, equipping robots with the same tools to allow for zero-shot transfer to the robot [39, 44, 50]. While this simplifies the data collection process, it limits the data to wrist camera viewpoints and may suffer from morphology differences when transferring from human-arm-collected data to robot arm execution. Additionally, DROID has larger scene and task diversity than prior tool-based collection datasets [39].

c) Scalable robot policy learning: Learning robot policies from increasingly large and diverse datasets has been the focus of numerous efforts over the last few years. Initially, these efforts focused in large part on learning from scripted or autonomously collected data [7, 11, 18, 24, 27, 34]. The success of transformer models [48] in natural language processing and computer vision motivated a number of recent works that collected large-scale demonstration datasets and trained transformer-based policies on them [2, 14, 23, 32, 33, 35, 41, 43, 52, 53]. Additionally, recent works suggest that diffusion denoising models [20] are a powerful parametrization for multi-modal action output distributions that combine expressivity with scalability [6, 14, 19, 32, 45]. Our focus with DROID is on introducing a new dataset, not a new policy learning algorithm. As such, we build on *existing* state-of-the-art diffusion policies [6] for all of our policy learning experiments.

[2]Fang et al. [13] report 110k trajectories for RH20T, but count each camera stream separately – here we report the number of unique multi-view trajectories, to compare fairly to all other datasets.

III. DROID Data Collection Setup

In this work, we introduce DROID (**D**istributed **Ro**bot **I**nteraction **D**ataset), an open-source robot manipulation dataset that provides for very high diversity and variability of scenes, tasks, and objects (see Table I). Diverse and high-quality data is a key ingredient for training generalizable policies, and DROID is designed to deliver both quantity and quality: it contains 76k robot demonstration trajectories, spanning 86 tasks and 564 scenes. It was collected over the course of 12 months in a large, cross-institutional effort with 18 robots and 50 data collectors across 13 institutions. All data is collected on a shared, open-source robot platform.

We are releasing all resources to enable researchers to build upon DROID at https://droid-dataset.github.io. This includes the full dataset under CC-BY 4.0 license, an interactive dataset visualizer, code for training generalizable policies on DROID, pre-trained policy checkpoints, and a detailed guide for reproducing our robot hardware setup and control stack. In this section, we introduce our hardware setup and the data collection protocol.

A. DROID Robot Platform

A crucial component of building the DROID dataset was distributed data collection at 13 institutions around the world: it is what enabled us to collect manipulation data across a large diversity of scenes and tasks. A key challenge in this distributed setup is robot hardware: how can we ensure *consistent and reproducible* robot control across so many setups, locations and time zones? To streamline the distributed data collection process we designed the DROID robot platform (see Fig. 2), a hardware platform for data collection that is shared between all institutions, allowing us to quickly set up new data collection units and roll out updates across the whole data collection fleet. It is designed to support easy transportation between scenes and quick adjustment to new scenes and tasks.

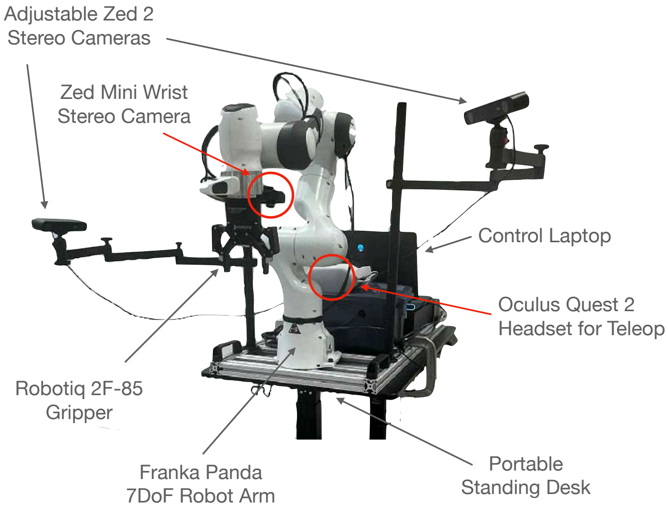

Fig. 2: The DROID robot platform. We use the same hardware setup across all 13 institutions to streamline data collection while maximizing portability and flexibility. The setup consists of a Franka Panda 7DoF robot arm, two adjustable Zed 2 stereo cameras, a wrist-mounted Zed Mini stereo camera, and an Oculus Quest 2 headset with controllers for teleoperation. Everything is mounted on a portable, height-adjustable desk for quick scene changes.

We chose the Franka Emika Panda 7 DoF robot arm as the base of our setup since it is widely adopted in the robot research community, reliable, relatively affordable and was available at most participating institutions. The robot arm is equipped with a Robotiq 2F-85 gripper and is mounted on a height-adjustable standing desk with wheels so it can easily move between scenes and buildings. We record image observations with three synchronized stereo camera streams: two exterior Zed 2 cameras, table-mounted on adjustable tripods to quickly adapt to a new scene layout, and a wrist-mounted Zed-Mini camera. We use the Polymetis controller [28] and record actions both in robot joint space and in end-effector space at a control frequency of 15Hz. The setup is completed with the Franka robot control box, a NUC that hosts the Polymetis server and an Alienware laptop that runs our data collection GUI (see Section III-B). Everything is powered with a single power cable to further simplify changes in location.

For teleoperation, we use the controllers of a Meta Quest 2 headset to control the pose of the arm in 6D space as well as the gripper in continuous space. Over the course of this project we have replicated this setup 18 times across various locations in North America, Asia, and Europe. We provide a thoroughly tested guide to replicate the hardware and software of our setup. We found that the setup is well-suited for data collection and policy learning across a wide range of scenes and tasks.

B. Data Collection Protocol

Our dataset is collected by 50 data collectors across various research institutions. A shared data collection protocol helps streamline data collection, particularly for inexperienced data collectors. When designing the collection protocol for DROID, we focused on the following objectives: (1) preventing common data collection mistakes like “camera cannot see robot” or “teleoperator in camera view”, (2) encouraging collection of diverse data, (3) allowing data collectors to creatively choose scenes and tasks.

Every data collection session starts with moving the robot to a new scene. Data collectors were encouraged to choose scenes that include multiple interesting tasks, numerous interaction objects, and a healthy amount of clutter (see example scenes in Fig. 12). After setting up the robot in the new scene, the data collector chooses views for the 3rd person cameras that can capture a wide range of interesting behaviors in the scene. Then they perform extrinsic camera calibration using a checkerboard and the OpenCV calibration algorithm. Next, the data collector will enter all potential tasks for the current scene into a data collection GUI on the laptop attached to the robot, either by selecting from a list of task options or by typing in free-from task instructions (see Fig. 11 for screenshots of the GUI). During data collection the GUI will prompt the data collector with a *randomly* sampled task from this list for each new episode. This way we ensure that there is high coverage of diverse tasks and collection is not biased to easier tasks or closer objects. Additionally, the GUI periodically prompts the data collector to perform randomly sampled “scene augmentations” like nudges to the mobile base, moving and re-calibrating the 3rd person cameras, changing the room lighting, and adding or removing items within the scene. For each trajectory, we record the output of all RGB cameras, relevant low level state information from the robot, equivalent robot control commands from various popular action spaces, a data collector ID, and the metadata entered in the GUI (see Section B for a detailed list of all features we record). The data collector also marks whether the collected sequence was a success, which we log as part of the metadata. DROID consists of 76k successful episodes; roughly 16k trajectories in our data collection were labeled as “not successful”, which we include in our dataset release but *do not* count towards the size of DROID. A data collector will typically collect up to 100 trajectories or about 20 minutes of interaction data per scene before moving on to a new scene.

During post-processing, we label each episode with natural language commands using crowdsourcing via the tasq.ai data labeling platform. We provide up to three independently labeled instructions per episode from different crowd workers to ensure diversity of annotations.

IV. DROID Dataset Analysis

While we have so far referred to DROID and other large-scale robot manipulation datasets as “diverse,” there is nuance in what constitutes a diverse robot dataset. Different axes of data diversity will affect the generalization abilities of models trained on the data differently: scene diversity may facilitate generalization to new scenes, while task or camera viewpoint diversity allows for greater generalization to new instructions and camera angles. We will analyze DROID along multiple

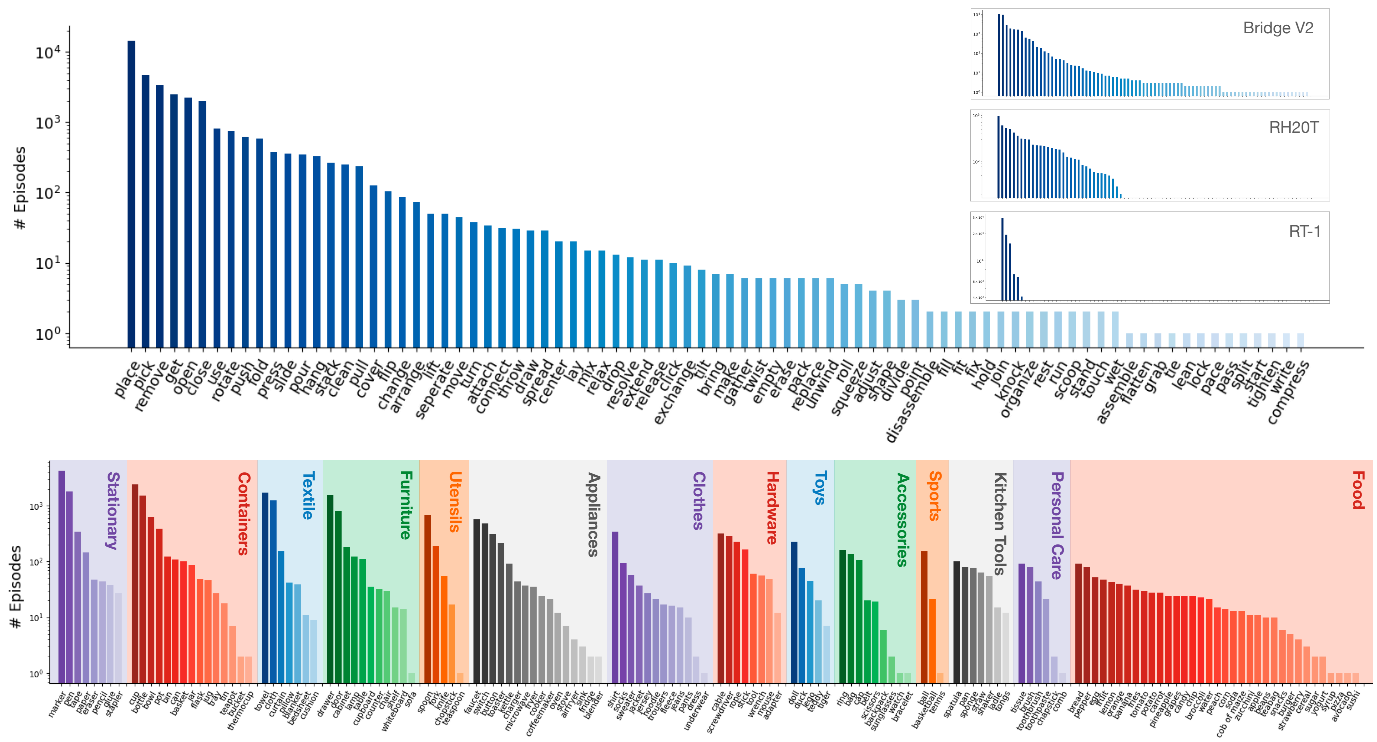

Fig. 3: Distribution of verbs and objects in DROID. **Top**: Distribution of verbs after de-duplication with GPT-4. DROID has a long tail of diverse tasks that span a wide range of behaviors. We also visualize the verb distributions for existing large manipulation datasets and find that only Bridge V2 [49] has a comparable long tail of skills (for a detailed view of verb distributions for all datasets, see Appendix, Fig. 13). **Bottom**: Distribution of objects the robot interacts with in DROID, sorted by category (best viewed zoomed in; for a detailed view, see Fig. 14).

important axes of diversity and compare it to existing large robot manipulation datasets.

When deciding which axes of generalization to inspect for robot manipulation datasets, it is important to consider which aspects of the problem may change between the training and downstream usage scenarios, i.e., which axes we want manipulation policies to generalize over. This may involve aspects of the scene, task, and robot setup. We identify the following important axes of diversity for closer analysis: task diversity, object diversity, scene diversity, viewpoint diversity, and interaction location diversity. The latter refers to the diversity of 3D locations relative to the robot's base at which interactions with objects occur, an important factor when generalizing to new scene layouts where interactions often need to generalize to new table heights or new parts of the robot's workspace.

We analyze DROID along these axes and compare it to existing large-scale robot manipulation datasets [2, 13, 49]. For each dataset, we run our analysis using one randomly sampled third-person camera frame per episode and the provided language instruction annotations. We find that results are consistent across randomly sampled frames.

We visualize the results of our analysis in Figs. 3 to 6. **Overall, we find that DROID significantly increases diversity in tasks, objects, scenes, viewpoints and interaction locations over existing large scale robot manipulation datasets**. A key reason is DROID's data collection protocol (see Section III-B): by collecting data with 50 data collectors in 52 buildings across three continents, switching scenes approximately every 20 minutes during collection and giving collectors the freedom to freely choose scene-appropriate tasks, we can substantially increase the diversity of scenes, tasks, and objects featured in the dataset. Next, we will describe our analysis for each category in more detail.

a) Task diversity: As explained in Section II, we use the distribution of de-duplicated verbs in a dataset's instructions as a scalable indicator for behavioral diversity. We use a semantic parsing algorithm [21] to extract verbs and referenced objects from the language instructions. We then use GPT4 to de-duplicate the verbs, i.e., remove synonyms and typos. We plot the distribution of verbs for DROID in Fig. 3, top. DROID features a wide variety of verbs with a long-tailed distribution. We use a logarithmic scale for these visualizations, since diversity is about covering a wide range of tasks, rather than having a high concentration of many episodes on only a handful of tasks – that is, it is less important if a task has 1000 vs. 2000 trajectories than whether it has 0 vs. 10. We also visualize the corresponding verb distributions for existing large manipulation datasets [2, 13, 49], and find that only Bridge V2 [49] has a comparable long tail of verb classes, although in a more restricted set of scenes (see scene diversity analysis below). Fig. 13 shows a detailed view of the verb

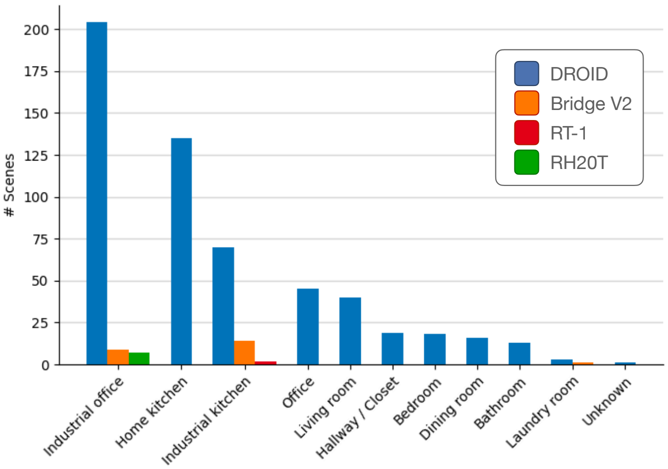

Fig. 4: Number of scenes per scene type. DROID has an order of magnitude more scenes than other large robot manipulation datasets, spanning a much wider range of scene types. We manually verified or confirmed with the authors that scene count and type for prior datasets is accurately reported.

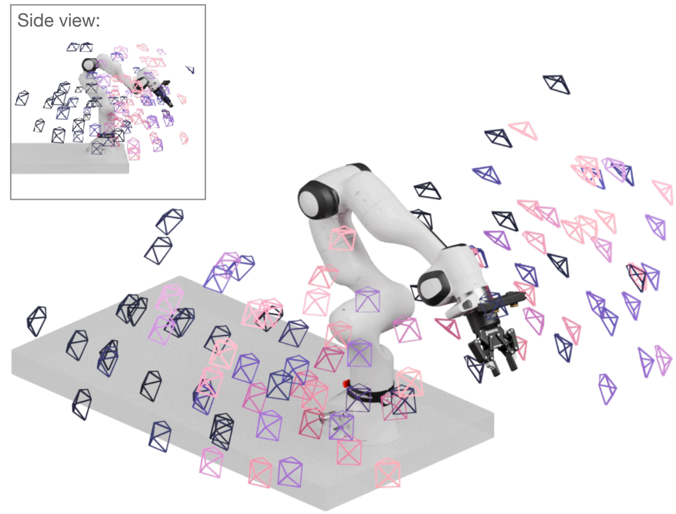

Fig. 5: Third-person camera viewpoints in DROID (subsampled). DROID episodes cover a total of 1417 camera viewpoints along with intrinsic and extrinsic stereo camera calibration. Brighter colors indicate regions of higher viewpoint density.

distributions for all datasets.

b) Object diversity: A dataset that includes manipulations for a large variety of objects facilitates generalization to new objects downstream. We analyze the objects the robot manipulates for each episode in DROID from the language instruction labels using the same semantic parsing pipeline [21] and show the distribution in Fig. 3, bottom (best viewed zoomed in, or see Fig. 14 for an enlarged version). DROID contains interactions with a wide range of everyday objects, spanning a diverse set of categories. We also plot the *joint* distribution of the most common verbs and interacted objects in Fig. 15. It shows that DROID not only contains diverse objects, but also a diverse range of interactions with most objects.

c) Scene diversity: We define 10 scene types (see Fig. 4) and use GPT-4V to determine the scene type for a given episode in DROID (see Appendix C for the used prompt). We find that this leads to high-quality scene type annotations (see Fig. 12 for example scenes and their categorization). For existing robot datasets we manually determine the scene types for each scene due to the small number of total scenes. DROID contains 564 unique scenes, an order of magnitude more than existing large robot manipulation datasets. The scenes cover a wide spectrum of scene types, from office environments to households. Qualitatively, the scenes in DROID reflect realistic real world scenarios with naturally occuring objects and backgrounds. We highly encourage the reader to inspect qualitative examples of scenes in Fig. 12 and the supplementary videos.

d) Viewpoint diversity: Existing large-scale robot learning datasets often only contain a limited set of camera viewpoints because the cameras are mounted in a fixed location relative to the scene or robot. In contrast, DROID varies camera viewpoints significantly during data collection and thus has a broad coverage of viewpoints (see Fig. 5) with 1417 unique view points in the dataset.

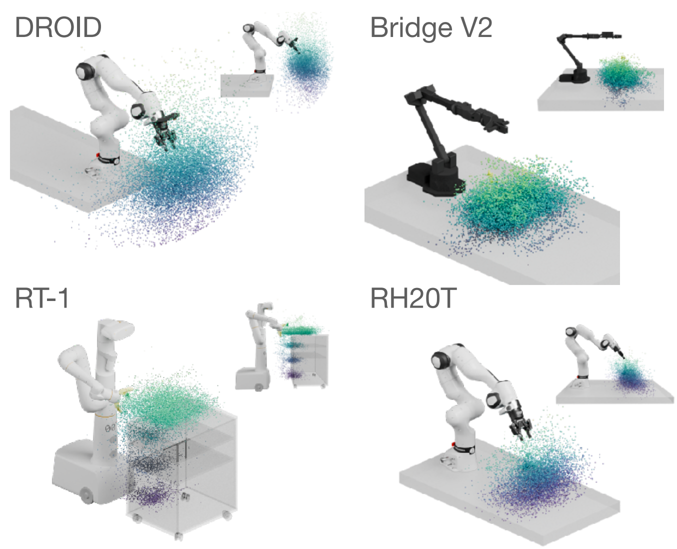

Fig. 6: Visualization of 3D interaction points relative to the robot base. We visualize the 3D location at which the gripper first closes in each trajectory, since closing the gripper often indicates meaningful object interactions. DROID's interactions cover a larger part of the robot's workspace, since the robot is moved freely between collection sessions instead of being placed in front of repetitive tabletop scenes.

e) Interaction location diversity: Another subtle yet important aspect of robot datasets is the diversity of interaction locations: are tasks always executed in the same narrow slice of the workspace, e.g., at the same table height, or does the data cover interactions across a large fraction of the work space? We use the point of first gripper closing in every episode as a proxy for interactions in the dataset and visualize the 3D location of these interaction points for different datasets in Fig. 6. DROID features interactions in a wider range of the workspace than existing robot manipulation datasets that

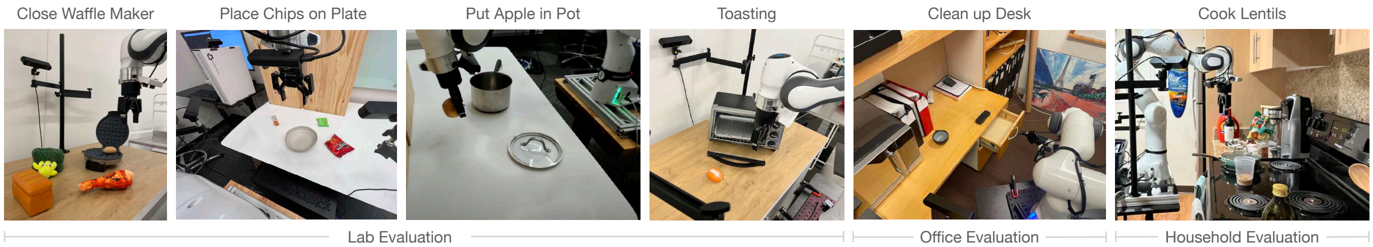

Fig. 7: Robot setups for policy evaluation. We cover a wide range of tasks and scenes, from lab evaluations to offices and real households, to reflect the diversity of use cases in real robot research. Depending on the task we collect between 50 and 150 demonstrations. We describe each task with out-of-distribution evaluation modifications in parenthesis, left to right: **Close Waffle Maker**: The robot needs to close a waffle maker (distractor objects). **Place Chips on Plate**: The robot needs to pick up the chips bag and place it on the provided plate (unseen chips bag and distractor objects). **Put Apple in Pot**: The robot needs to pick up the apple, place it in the pot, and close the lid (unseen distractor object). **Toasting**: The robot needs to pick up the object, place it in the toaster oven, and close the oven (toast a novel object). **Clean up Desk**: The robot needs to open the drawer, place the eraser that is on top of the desk inside the drawer, and close it (distractor objects on desk and in drawer). **Cook Lentils**: The robot needs to remove the pan lid, pick up and pour lentils into the pan, and turn on the stove(add distractor objects).

typically focus interactions on a table surface in front of the robot.

V. Experiments

The analysis in the previous section highlighted the diversity of tasks, objects, scenes, and viewpoints in the DROID dataset. In this section, we investigate whether this diverse data resource can be used to boost policy performance and robustness across a wide spectrum of robot manipulation tasks and environments. To this end, we train policies across 6 tasks in 4 different locations including lab, office, and household settings, to reflect the diversity of real world robotic research use cases (see Fig. 7). All experiments use representative, state of the art robot policy learning approaches [6]. Across the board, we find that DROID improves policy success rate while increasing robustness to scene changes like distractors or novel object instances.

A. Experimental Setup

a) Tasks: As illustrated in Fig. 7, we choose 6 tasks in 4 locations that span a representative range of real robot learning use cases: from simple pick-place tasks to multi-stage cooking tasks; from clean lab settings to real households. All experiments use the DROID hardware stack for policy evaluations. Concretely, we evaluate on the following 6 tasks, each with their own out-of-distribution variants:

Closing Waffle Maker: A short horizon task in a lab setting (70 demonstrations), where the task is to close a waffle maker. The waffle maker position is randomized between episodes. The out of distribution variant consists of adding several distractor objects on the table.

Place Chips on Plate: A short horizon task in a lab setting (50 demonstrations), where the task is to pick and place a bag of Doritos chips onto a plate, with two distractor objects on the table. All objects and the plate position are randomized between episodes on the table. The out of distribution variant consists of (a) changing the type of chips or (b) adding more distractor objects to the table.

Put Apple in Pot: A medium horizon task in a lab setting (60 demonstrations), where the task is to pick and place an apple into a pot and then put a lid on the pot. The apple, pot, and lid position are randomized between episodes on the table. The out of distribution variant involves placing a distractor plate on the table.

Toasting: A medium horizon task in a lab setting (150 demonstrations), where the task is to put an object on a toaster oven tray, then close the toaster oven. The object and toaster position are randomized between episodes on the table. The out of distribution variant consists of toasting novel objects.

Clean up Desk: A long horizon task in an office setting (50 demonstrations), where the task is to open a drawer, pick and place an eraser into the drawer, and then close the drawer. The eraser position is fixed. The out of distribution variant consists of adding distractor objects on the desk and in the drawer.

Cook Lentils: A long horizon task in a kitchen setting (50 demonstrations), where the task is to remove the lid off a pan, pour lentils into the pan, and turn on the stove. The object positions are fixed. The out of distribution variant consists of adding several distractor objects and a camera shift.

Additional details about each evaluation task can be found in Appendix E. All data is collected using the DROID teleoperation setup and training uses the same standardized policy learning backbone.

b) Policy training: The goal of this work is to introduce a new robot manipulation dataset, *not* to introduce a new policy learning method. Thus, during experimental evaluations we aim to leverage a well-adopted, state-of-the-art policy learning pipeline. To this end, we use diffusion policies [6, 14, 19, 32, 45], which leverage denoising diffusion models for action prediction and have recently demonstrated strong performance across a range of applications. We build on the implementation of diffusion policies in Robomimic [31], which provides high quality open-source implementations of a number of different imitation learning and offline RL algorithms. Concretely, all of our policies are conditioned on a language instruction, use the RGB camera streams from the two external cameras and the robot proprioception as input, and produce absolute robot end-effector translation, rotation, and

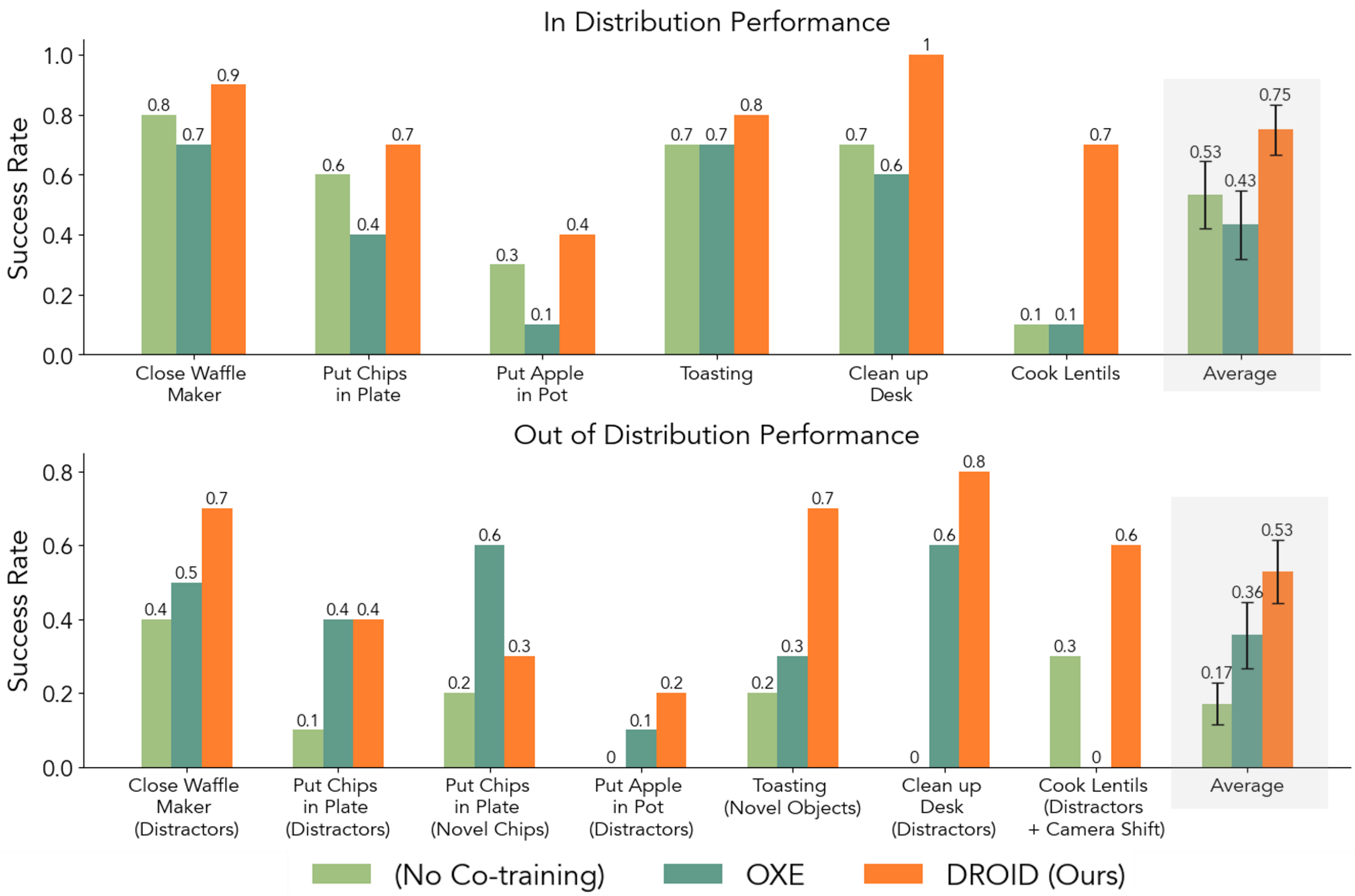

Fig. 8: **Does DROID Improve Policy Performance and Robustness?** We find that across all our evaluation tasks, co-training with DROID significantly improves both in distribution and OOD performance over both no co-training and co-training with the Open-X dataset. We compare success rate averaged across all tasks with standard error, and find DROID outperforms the next best method **by 22%** absolute success rate in-distribution and **by 17%** out of distribution.

gripper actions. We first downsample the camera observations to a resolution of 128×128 and use a ResNet-50 visual encoder pre-trained on ImageNet [10] to encode both visual inputs. We then concatenate these visual embeddings with a frozen DistilBERT [37] language embedding and the robots proprioceptive state. These concatenated features are then fed through an MLP and passed to a U-Net diffusion head which generates action trajectories. In line with prior work [6], we train the diffusion policy to generate 16-step action sequences, and during rollouts, step 8 actions open loop before re-running policy inference. For leveraging DROID during policy training, we simply mix training batches at a 50/50 ratio between the small in-domain dataset and the complete DROID dataset but excluding trajectories marked as "not successful", which we find to work well in practice. Additional details about the policy training can be found in Appendix F.

B. Does DROID Improve Policy Performance and Robustness?

To study if co-training with DROID can enable improved policy learning, we train separate policies for each evaluation task and compare all policies head-to-head in A/B evaluations using 10 rollouts for each task setting and method. To test how DROID and existing datasets affect policy robustness, we evaluate each task and method in two settings: "in-distribution," which reflects the distribution of tasks in the in-domain demonstrations with noise added to the initial robot and object positions, and "out-of-distribution" (OOD), which tests policy robustness e.g., by introducing distractor objects or switching the manipulated object. We evaluate the following approaches:

- **No Co-training**: Trains a diffusion policy [6] using the in-domain demonstrations only
- **DROID (Ours)**: Trains a diffusion policy, but mixes batches 50/50 between in-domain demonstrations and DROID demonstrations
- **OXE [33]**: Trains a diffusion policy, but mixes batches 50/50 between in-domain demonstrations and trajectories from the Open X-Embodiment dataset [33] (OXE). OXE contains most of the existing large robot manipulation datasets we compared DROID to in Section IV, as well as a large number of other robot datasets, spanning 22 robot

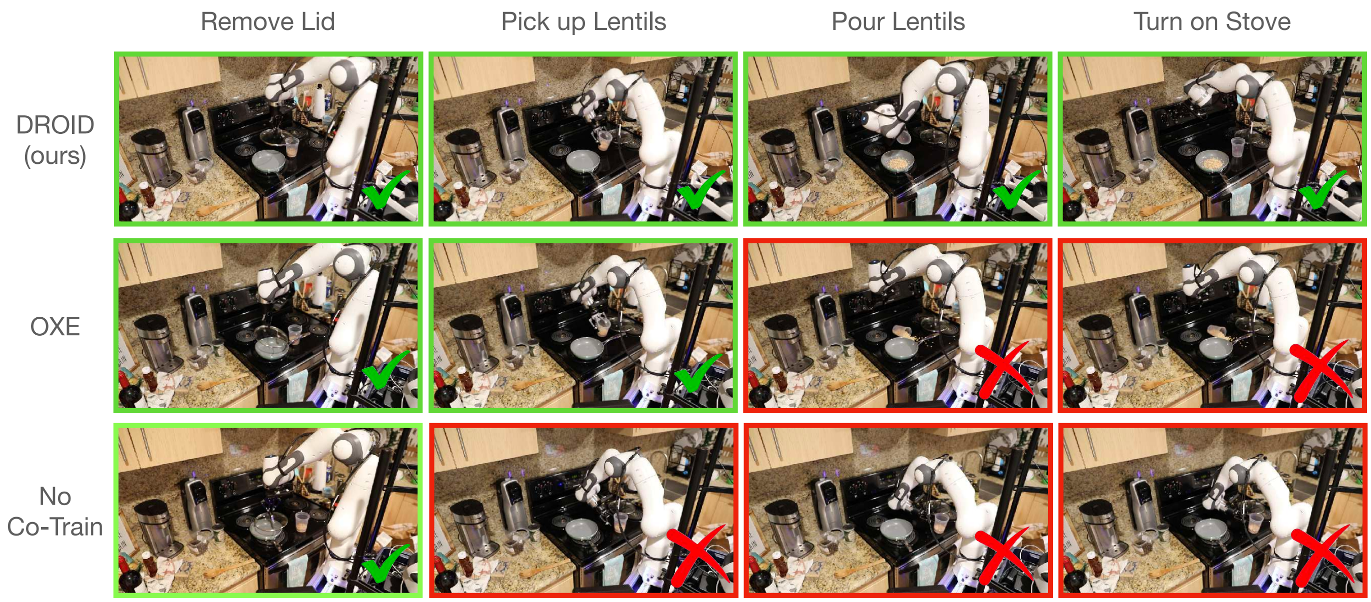

Fig. 9: Representative policy rollout examples on the most challenging "Cook Lentils" task in the kitchen of one of the authors. Qualitatively, we find that policies co-trained with DROID perform smoother, more precise motions, allowing them to solve long-horizon tasks like lentil cooking even in the presence of unseen distractor objects. In contrast, policies trained only with in-domain demonstrations or co-trained with Open-X data [33] struggle with long-horizon tasks and out-of-distribution evaluation settings. See https://droid-dataset.github.io for rollout videos.

embodiments and approximately 300 scenes total.[3]

We present the results of our policy evaluations in Fig. 8. Across all tasks, we find that DROID substantially improves policy performance compared to the diffusion policy trained on in-domain data only. Policies co-trained with DROID also perform better than policies that leverage diverse, existing robot datasets in Open X-Embodiment (OXE). Notably, when testing out of distribution performance, the No Co-training baseline performs quite poorly while the co-trained policies are much more effective. This difference is especially notable when co-training with DROID, which has the strongest overall performance.

Qualitatively, we find that policies that leverage DROID during training are notably smoother and precise than other comparisons, particularly in the more challenging out-of-distribution evaluations. For instance, in the OOD setting of the Waffle Closing task, DROID is the only method that consistently reaches for the waffle maker, while the other methods get confused about the task. Similarly, in the multi-step Cook Lentils task, baselines tend to fail after two or sometimes just one step, while co-training with DROID is the only method able to consistently finish all three steps See Fig. 9 for examples of qualitative task rollouts.

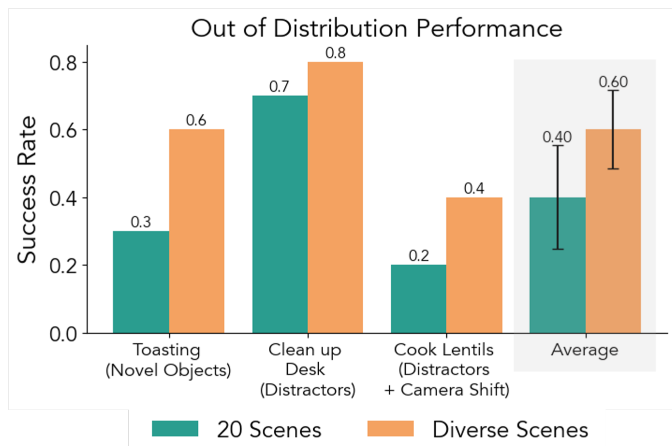

Fig. 10: **How important is the scene diversity in DROID?** We find that co-training on a subset of DROID with diverse scenes has higher OOD performance than co-training on a subset of DROID with only 20 scenes, suggesting the scene diversity of DROID is one of the driving factors behind strong policy performance when co-training.

C. *How important is the scene diversity in DROID?*

One of the unique benefits of DROID compared to existing robot datasets is its amount of *scene diversity*. Indeed we see in Figure 4 that DROID contains far more scene diversity than the next most diverse robot manipulation dataset. While we've

[3]We use a curated split of OXE based on Octo Model Team et al. [32], which has been shown to work well for policy learning in prior work [32]. We remove the Language Table dataset [29], equivalent to 5% of the Octo training mix, due to its repetitive scene layouts and tasks, and its raw size, which proved challenging to handle for our training infrastructure.

seen the benefits of co-training with DROID, *can we quantify how much of a role scene diversity plays in improved policy robustness?*

To test this, we design an experiment that uses the challenging OOD versions of the evaluation tasks from Section V-A, but compares:

- **DROID (7k, 20 Scenes)**: Selects for the 20 scenes from DROID with the most demonstrations each, resulting in 7362 trajectories with comparatively little scene diversity.
- **DROID (7k, Diverse Scenes)**: Uniform random sample of 7362 successful demonstrations from the DROID dataset, which matches dataset size to the previous method while retaining high scene diversity.

These comparisons use the same 50/50 co-training paradigm with individual task data used in the previous experiment. Hence, this helps establish whether the scene diversity of DROID results in better policy performance than just using 20 scenes while controlling for dataset size.

In Figure 10 we observe that using the split of the dataset with more diverse scenes yields better performance in the OOD evaluation setting. By comparing Figure 10's individual task performances with the corresponding tasks in Figure 8, we also see that the performance of co-training with the full DROID dataset matches or outperforms the performance with the subsampled dataset on all three tasks. These results suggest that the strength of DROID lies in its size and especially in its *diversity*.

VI. Discussion

In this work, we introduced DROID (**D**istributed **Ro**bot **I**nteraction **D**ataset), a new robot manipulation dataset with a large diversity of scenes, tasks, objects and viewpoints. Our dataset analysis in Section IV showed that DROID has an order of magnitude larger scene diversity than existing large robot manipulation datasets, a wide range of tasks, many interaction objects, and diverse viewpoints. Our policy learning evaluations show that DROID is a valuable data resource for improving policy performance and robustness, even in comparison to existing large robot data sources like the Open X-Embodiment dataset [33].

We hope that DROID can be a catalyst for research on general-purpose robot manipulation policies that are able to generalize to a broad range of tasks and scenes. In this work, we showed one example for leveraging DROID to boost policy performance, but there are many open questions about how to best make use of such diverse data: how should we combine DROID with existing large-scale robot datasets and how can we train policies that perform tasks in new scenes without *any* in-domain data? Can the diverse interaction data in DROID be used to learn better visual representations for robotic control? And in what situations is it helpful to train on the full dataset vs. slices of the data? We hope that DROID can accelerate research on these questions and are excited for how the community will leverage the dataset! We also hope that our open-sourced hardware platform, which already exists in 18 labs around the globe and is easy to reproduce, can improve reproducibility of robot learning research and facilitate future additions to the DROID dataset.

Acknowledgment

We thank the Toyota Research Institute (TRI) for their support in various aspects of this project, from data collection to compute for policy training. This work was supported by the Google TPU Research Cloud. We further acknowledge the following funding sources: Chelsea Finn's group was supported by TRI and ONR grants N00014-20-1-2675 and N00014-22-1-2621; Sergey Levine's group was supported by TRI, NSF FRR IIS-2150826, and ONR N00014-20-1-2383; Ram Ramamoorthy's group was supported by the United Kingdom Research and Innovation through grant EP/S023208/1 to the EPSRC Centre for Doctoral Training in Robotics and Autonomous Systems (RAS) and grant EP/V026607/1 to the UKRI Research Node on Trustworthy Autonomous Systems Governance and Regulation; Dorsa Sadigh's group was supported by TRI and ONR grant N00014-22-1-2293; Glen Berseth's group acknowledges funding support from NSERC and CIFAR and compute support from Digital Research Alliance of Canada, Mila IDT and NVidia; Jeannette Bohg's group was supported by TRI, Intrinsic, Toshiba and the National Science Foundation under Grant 2327974; Joseph Lim's group was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grants (No.2019-0-00075, Artificial Intelligence Graduate School Program, KAIST; No.2022-0-00077, AI Technology Development for Commonsense Extraction, Reasoning, and Inference from Heterogeneous Data), and a National Research Foundation of Korea (NRF) grant (NRF-2021H1D3A2A03103683) funded by the Korean government (MSIT).

References

[1] Homanga Bharadhwaj, Jay Vakil, Mohit Sharma, Abhinav Gupta, Shubham Tulsiani, and Vikash Kumar. Roboagent: Generalization and efficiency in robot manipulation via semantic augmentations and action chunking. *arXiv preprint arXiv:2309.01918*, 2023.

[2] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

[3] Serkan Cabi, Sergio Gomez Colmenarejo, Alexander Novikov, Ksenia Konyushkova, Scott Reed, Rae Jeong, Konrad Zolna, Yusuf Aytar, David Budden, Mel Vecerik, Oleg Sushkov, David Barker, Jonathan Scholz, Misha Denil, Nando de Freitas, and Ziyu Wang. Scaling data-driven robotics with reward sketching and batch reinforcement learning. *RSS*, 2019.

[4] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *preprint arXiv:1903.11027*, 2019.

[5] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[6] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[7] Sudeep Dasari, Frederik Ebert, Stephen Tian, Suraj Nair, Bernadette Bucher, Karl Schmeckpeper, Siddharth Singh, Sergey Levine, and Chelsea Finn. Robonet: Large-scale multi-robot learning. *CoRL*, 2019.

[8] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, et al. Objaverse-xl: A universe of 10m+ 3d objects. *arXiv preprint arXiv:2307.05663*, 2023.

[9] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13142–13153, 2023.

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

[11] Frederik Ebert, Chelsea Finn, Sudeep Dasari, Annie Xie, Alex Lee, and Sergey Levine. Visual foresight: Model-based deep reinforcement learning for vision-based robotic control. *arXiv:1812.00568*, 2018.

[12] Frederik Ebert, Yanlai Yang, Karl Schmeckpeper, Bernadette Bucher, Georgios Georgakis, Kostas Daniilidis, Chelsea Finn, and Sergey Levine. Bridge data: Boosting generalization of robotic skills with cross-domain datasets. *arXiv preprint arXiv:2109.13396*, 2021.

[13] Hao-Shu Fang, Hongjie Fang, Zhenyu Tang, Jirong Liu, Chenxi Wang, Junbo Wang, Haoyi Zhu, and Cewu Lu. Rh20t: A comprehensive robotic dataset for learning diverse skills in one-shot. *Towards Generalist Robots: Learning Paradigms for Scalable Skill Acquisition@ CoRL2023*, 3:5, 2023.

[14] 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.

[15] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

[16] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012.

[17] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19012, 2022.

[18] Abhinav Gupta, Adithyavairavan Murali, Dhiraj Prakashchand Gandhi, and Lerrel Pinto. Robot learning in homes: Improving generalization and reducing dataset bias. *Advances in neural information processing systems*, 31, 2018.

[19] Huy Ha, Pete Florence, and Shuran Song. Scaling up and distilling down: Language-guided robot skill acquisition. In *Conference on Robot Learning*, pages 3766–3777. PMLR, 2023.

[20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[21] Matthew Honnibal and Ines Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear, 2017.

[22] Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. Bc-z: Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learning*, pages 991–1002. PMLR, 2022.

[23] Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. VIMA: Robot manipulation with multimodal prompts. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 14975–15022. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/jiang23b.html.

[24] Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. QT-Opt: Scalable deep reinforcement learning for vision-based robotic manipulation. *arXiv preprint arXiv:1806.10293*, 2018.

[25] Dmitry Kalashnkov, Jake Varley, Yevgen Chebotar, Ben Swanson, Rico Jonschkowski, Chelsea Finn, Sergey Levine, and Karol Hausman. Mt-opt: Continuous multi-task robotic reinforcement learning at scale. *arXiv*, 2021.

[26] Haresh Karnan, Anirudh Nair, Xuesu Xiao, Garrett Warnell, Sören Pirk, Alexander Toshev, Justin Hart, Joydeep Biswas, and Peter Stone. Socially compliant navigation dataset (scand): A large-scale dataset of demonstrations for social navigation. *IEEE Robotics and Automation Letters*, 7(4):11807–11814, 2022.

[27] Sergey Levine, Peter Pastor, Alex Krizhevsky, and Deirdre

Quillen. Learning hand-eye coordination for robotic grasping with large-scale data collection. In *International Symposium on Experimental Robotics*. Springer, 2016.

[28] Yixin Lin, Austin S. Wang, Giovanni Sutanto, Akshara Rai, and Franziska Meier. Polymetis. https://facebookresearch.github.io/fairo/polymetis/, 2021.

[29] Corey Lynch, Ayzaan Wahid, Jonathan Tompson, Tianli Ding, James Betker, Robert Baruch, Travis Armstrong, and Pete Florence. Interactive language: Talking to robots in real time. *IEEE Robotics and Automation Letters*, 2023.

[30] Ajay Mandlekar, Yuke Zhu, Animesh Garg, Jonathan Booher, Max Spero, Albert Tung, Julian Gao, John Emmons, Anchit Gupta, Emre Orbay, et al. Roboturk: A crowdsourcing platform for robotic skill learning through imitation. In *Conference on Robot Learning*, pages 879–893. PMLR, 2018.

[31] Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In *arXiv preprint arXiv:2108.03298*, 2021.

[32] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Charles Xu, Jianlan Luo, Tobias Kreiman, You Liang Tan, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. https://octo-models.github.io, 2023.

[33] Open X-Embodiment Collaboration, Abhishek Padalkar, Acorn Pooley, Ajinkya Jain, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anikait Singh, Anthony Brohan, Antonin Raffin, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Brian Ichter, Cewu Lu, Charles Xu, Chelsea Finn, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Chuer Pan, Chuyuan Fu, Coline Devin, Danny Driess, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Federico Ceola, Fei Xia, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Giulio Schiavi, Hao Su, Hao-Shu Fang, Haochen Shi, Heni Ben Amor, Henrik I Christensen, Hiroki Furuta, Homer Walke, Hongjie Fang, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jaehyung Kim, Jan Schneider, Jasmine Hsu, Jeannette Bohg, Jeffrey Bingham, Jiajun Wu, Jialin Wu, Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jitendra Malik, Jonathan Tompson, Jonathan Yang, Joseph J. Lim, João Silvério, Junhyek Han, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Zhang, Keyvan Majd, Krishan Rana, Krishnan Srinivasan, Lawrence Yunliang Chen, Lerrel Pinto, Liam Tan, Lionel Ott, Lisa Lee, Masayoshi Tomizuka, Maximilian Du, Michael Ahn, Mingtong Zhang, Mingyu Ding, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J Joshi, Niko Suenderhauf, Norman Di Palo, Nur Muhammad Mahi Shafiullah, Oier Mees, Oliver Kroemer, Pannag R Sanketi, Paul Wohlhart, Peng Xu, Pierre Sermanet, Priya Sundaresan, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto Martín-Martín, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Sherry Moore, Shikhar Bahl, Shivin Dass, Shuran Song, Sichun Xu, Siddhant Haldar, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Sudeep Dasari, Suneel Belkhale, Takayuki Osa, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Vidhi Jain, Vincent Vanhoucke, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiaolong Wang, Xinghao Zhu, Xuanlin Li, Yao Lu, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yueh hua Wu, Yujin Tang, Yuke Zhu, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zhuo Xu, and Zichen Jeff Cui. Open X-Embodiment: Robotic learning datasets and RT-X models. https://arxiv.org/abs/2310.08864, 2023.

[34] Lerrel Pinto and Abhinav 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.

[35] Ilija Radosavovic, Baifeng Shi, Letian Fu, Ken Goldberg, Trevor Darrell, and Jitendra Malik. Robot learning with sensorimotor pre-training. In *Conference on Robot Learning*, 2023.

[36] Ahad Rana. Common crawl – building an open web-scale crawl using hadoop, 2010. URL https://www.slideshare.net/hadoopusergroup/common-crawlpresentation.

[37] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108, 2019. URL https://api.semanticscholar.org/CorpusID:203626972.

[38] 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. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022.

[39] Nur Muhammad Mahi Shafiullah, Anant Rai, Haritheja Etukuru, Yiqian Liu, Ishan Misra, Soumith Chintala, and Lerrel Pinto. On bringing robots home, 2023.

[40] Dhruv Shah, Ajay Sridhar, Arjun Bhorkar, Noriaki Hirose, and Sergey Levine. Gnm: A general navigation model to drive any robot. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7226–7233. IEEE, 2023.

[41] Dhruv Shah, Ajay Sridhar, Nitish Dashora, Kyle Stachowicz, Kevin Black, Noriaki Hirose, and Sergey Levine. ViNT: A foundation model for visual navigation. In *7th Annual Conference on Robot Learning*, 2023. URL https://arxiv.org/abs/2306.14846.

[42] Pratyusha Sharma, Lekha Mohan, Lerrel Pinto, and Abhinav Gupta. Multiple interactions made easy (mime): Large scale demonstrations data for imitation. In *Conference on robot learning*, pages 906–915. PMLR, 2018.

[43] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Proceedings of the 6th Conference on Robot Learning (CoRL)*, 2022.

[44] Shuran Song, Andy Zeng, Johnny Lee, and Thomas Funkhouser. Grasping in the wild: Learning 6dof closed-loop grasping from low-cost demonstrations. *IEEE Robotics and Automation Letters*, 5(3):4978–4985, 2020.

[45] Ajay Sridhar, Dhruv Shah, Catherine Glossop, and Sergey Levine. Nomad: Goal masked diffusion policies for navigation and exploration. *arXiv preprint arXiv:2310.07896*, 2023.

[46] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset, 2019.

[47] Samuel Triest, Matthew Sivaprakasam, Sean J Wang, Wenshan Wang, Aaron M Johnson, and Sebastian Scherer. Tartandrive: A large-scale dataset for learning off-road dynamics models. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2546–2552. IEEE, 2022.

[48] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[49] Homer Walke, Kevin Black, Abraham Lee, Moo Jin Kim, Max Du, Chongyi Zheng, Tony Zhao, Philippe Hansen-Estruch, Quan Vuong, Andre He, Vivek Myers, Kuan Fang, Chelsea Finn, and Sergey Levine. Bridgedata v2: A dataset for robot learning at scale, 2023.

[50] Sarah Young, Dhiraj Gandhi, Shubham Tulsiani, Abhinav Gupta, Pieter Abbeel, and Lerrel Pinto. Visual imitation made easy, 2020.

[51] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2636–2645, 2020.

[52] 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.

[53] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *7th Annual Conference on Robot Learning*, 2023.

Appendix A
Contributions

Project Leads: Alexander Khazatsky, Karl Pertsch

Research Leads (contributed significantly to development of data collection setup, data post-processing and policy training): Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama

Engineers (helped implement data collection, postprocessing and policy learning infrastructure): Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, Peter David Fagan, Masha Itkina, Marion Lepert, Jason Ma, Patrick Tree Miller, Jimmy Wu, Huy Ha, Youngwoon Lee, Kaiyuan Wang, Kevin Black, Cheng Chi, Kyle Hatch, Shan Lin, Jingpei Lu, Abdul Rehman, Pannag R Sanketi, Cody Simpson, Quan Vuong, Blake Wulfe, Ted Xiao, Jonathan Yang, Arefeh Yavary, Tony Z. Zhao

Lab Leads (coordinated data collection in their respective labs): Suraj Nair, Ashwin Balakrishna, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, Peter David Fagan, Joey Hejna, Masha Itkina, Marion Lepert, Jason Ma, Patrick Tree Miller, Jimmy Wu, Suneel Belkhale, Shivin Dass, Huy Ha, Abraham Lee, Youngwoon Lee, Arhan Jain, Marius Memmel, Sungjae Park, Ilija Radosavovic, Kaiyuan Wang, Albert Zhan, Archit Sharma, Homer Walke

Policy Evaluators (ran robot evaluations for policy learning experiments): Alexander Khazatsky, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Mohan Kumar Srirama, Joey Hejna, Donovon Jackson, Tony Nguyen, Derick Seale

Data Collectors: Alexander Khazatsky, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, Peter David Fagan, Masha Itkina, Marion Lepert, Jason Ma, Patrick Tree Miller, Jimmy Wu, Suneel Belkhale, Shivin Dass, Abraham Lee, Arhan Jain, Marius Memmel, Sungjae Park, Ilija Radosavovic, Albert Zhan, Christopher Agia, Rohan Baijal, Mateo Guaman Castro, Daphne Chen, Qiuyu Chen, Trinity Chung, Jaimyn Drake, Ethan Paul Foster, Jensen Gao, David Antonio Herrera, Minho Heo, Kyle Hsu, Jiaheng Hu, Donovon Jackson, Charlotte Le, Yunshuang Li, Kevin Lin, Roy Lin, Zehan Ma, Abhiram Maddukuri, Suvir Mirchandani, Daniel Morton, Tony Nguyen, Abby O'Neill, Rosario Scalise, Derick Seale, Victor Son, Stephen Tian, Andrew Wang, Yilin Wu, Annie Xie, Jingyun Yang, Patrick Yin, Yunchu Zhang

Lead Advisor: Chelsea Finn

Advisors: Osbert Bastani, Glen Berseth, Jeannette Bohg, Ken Goldberg, Abhinav Gupta, Abhishek Gupta, Dinesh Jayaraman, Joseph J. Lim, Jitendra Malik, Roberto Martín-Martín, Subramanian Ramamoorthy, Dorsa Sadigh, Shuran Song, Jiajun Wu, Yuke Zhu, Thomas Kollar, Sergey Levine

Appendix B
DROID Data Features

All DROID data is recorded at 15Hz. Each DROID trajectory contains the following elements:

- 3 stereo RGB camera streams at 1280x720 resolution
- robot joint positions and velocities (7D)
- robot end-effector pose and velocity in robot base frame (6D)
- robot gripper position and velocity (1D)

Additionally each trajectory has the following metadata:

- 1-3 natural language instructions describing the task performed in the trajectory, collected via crowdsourcing
- extrinsic camera calibration matrices for both exterior cameras
- building name and data collector user ID
- scene type, as classified by GPT4V (see Section C)

Appendix C
Scene Type Classification

We labeled scene types in an automated fashion using the GPT4V API. For each scene, we sampled a random episode from that scene and a random image from that episode. That image along with the prompt shown in Listing C.1 was sent for labeling. We then reviewed samples assigned "Other" to confirm that we were not missing any major categories, and then reassigned those labels manually.

Appendix D
Indentifying Unique Scenes

As mentioned in the main body of the paper, we define a unique scene as a substantial change to the robot's workspace. For example, a home kitchen may have multiple unique scenes associated with it in which the robot is placed in front of the refrigerator, sink, stove top, or different sections of the counter. We do not consider changes in the objects being interacted with or changes to the poses of external cameras sufficient to constitute a unique scene.

We label unique scenes as follows. During data collection, a scene ID number is generated each time the user indicates that robot or external cameras are moved. In total, there are 2,080 unique scene IDs in the dataset. Many of these scene IDs correspond to the same scene based on the definition provided above since users mislabel scene changes or move the robot back to the same scene after moving it somewhere else.

In order to identify these duplicates, we collect scenes into groups that share the same robot serial number, name of the lab collecting the data, and building name. Within each group, we order the scenes by timestamp. We then go through the scenes sequentially, identifying cases where the scene did not change sufficiently to constitute a unique scene. Finally, we search across the remaining set of scenes within each group to identify cases where a robot was placed at the same scene

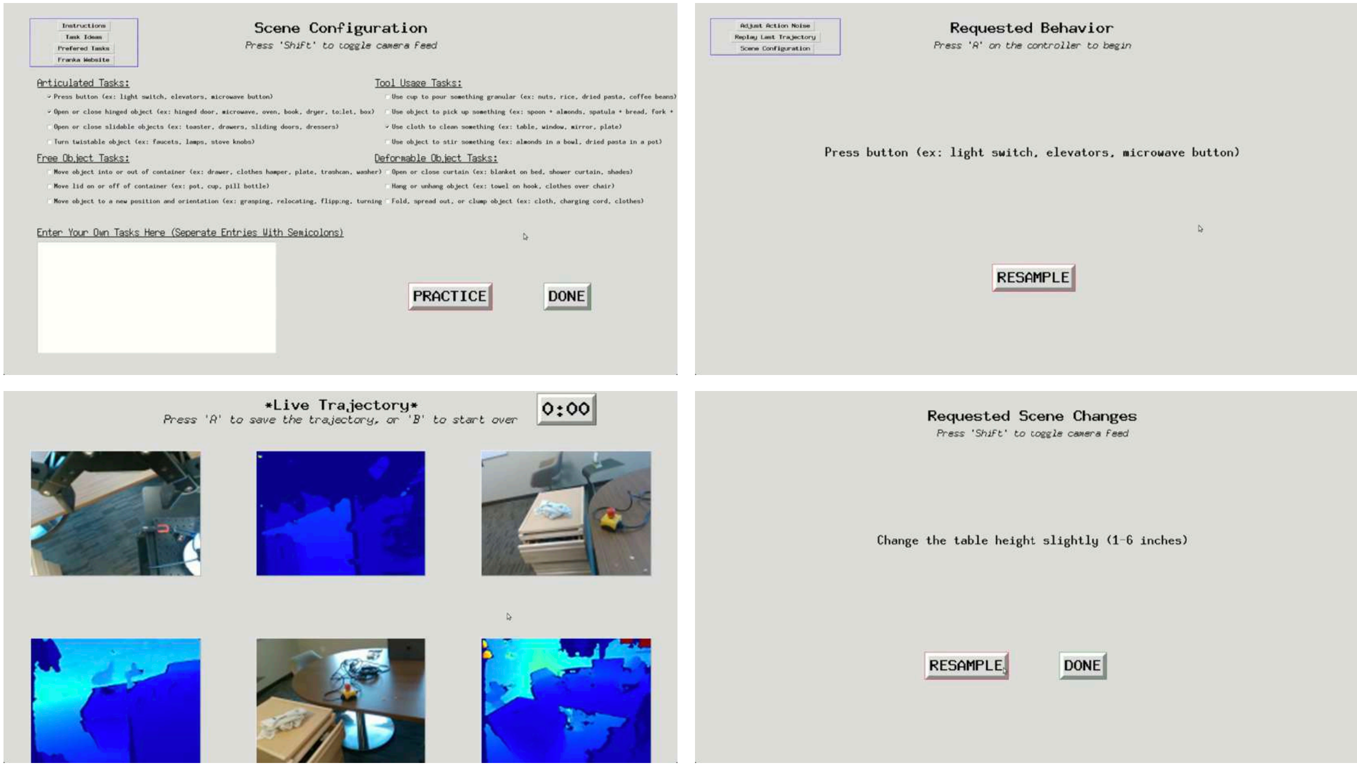

Fig. 11: DROID data collection GUI. **Top left**: Screen for entering feasible tasks for the current scene. Tasks can either be selected from a list of suggestions or typed as free-form instructions. **Top right**: Instruction screen – the GUI samples a task *at random* from the entered list of feasible tasks and instructs the data collector to record a demonstration trajectory. This ensures wide coverage of possible tasks in each scene and avoids bias towards easy or familiar tasks. **Bottom left**: Data collection screen – displays RGB and depth camera live streams. **Bottom right**: The GUI periodically suggests scene changes between demonstration collections to ensure high scene diversity.

twice (though not sequentially), and also remove these from the set of unique scenes.

This labeling approach has some limitations. For example, because we group scenes based on robot serial number and only identify duplicates within that group, if two different robots are placed at the same scene then that scene would be counted twice. Nevertheless, during labeling we were conservative in our estimate of what constituted a unique scene, and as a result believe that the number reported in the paper represents a conservative estimate.

Appendix E
Evaluation Procedure

We evaluate learned policies on the following 6 tasks, each with their own out of distribution variants. For each evaluation, we ensure that each of the policies see a similar initial distribution of object locations across trials.

Place Chips on Plate: A short horizon task in a lab setting, where the task is to pick and place a bag of Doritos chips onto a plate, with two distractor objects on the table. All objects and the plate position are randomized between episodes on the table. We collect 50 demonstrations, and mark success if the chips are in the plate. We also consider two out of distribution variants: (1) changing the type of chips to Sun Chips (different size and color) and (2) putting two additional distractor objects (an apple and an orange) on the table.

Put Apple in Pot: A medium horizon task in a lab setting, where the task is to pick and place an apple into a pot and then put a lid on the pot. The apple, pot, and lid position are randomized between episodes on the table. We collect 60 demonstrations, and mark success if the apple is in the pot and the lid is on the pot. The out of distribution variant involves placing an additional plate on the table as a distractor.

Toasting: A medium horizon task in a lab setting, where the task is to put an object on a toaster oven tray, then close the toaster oven. The object and toaster position are randomized between episodes on the table. We collect 150 demonstrations, and mark success if the object is in the toaster oven and the toaster oven is closed. The out of distribution variant consists of considering novel objects to toast.

Closing Waffle Maker: A short horizon task in a lab setting, where the task is to close a waffle maker. The waffle maker position is randomized between episodes. We collect 70 demonstrations, and mark success if the waffle maker is closed. The out of distribution variant consists of adding several distractor objects on the table.

Clean up Desk: A long horizon task in an office setting, where the task is to open a drawer, pick and place an eraser into the drawer, and then close the drawer. The eraser position is varied at the start of each episode at a set schedule of different positions and orientations. We collect 50 demonstrations, and

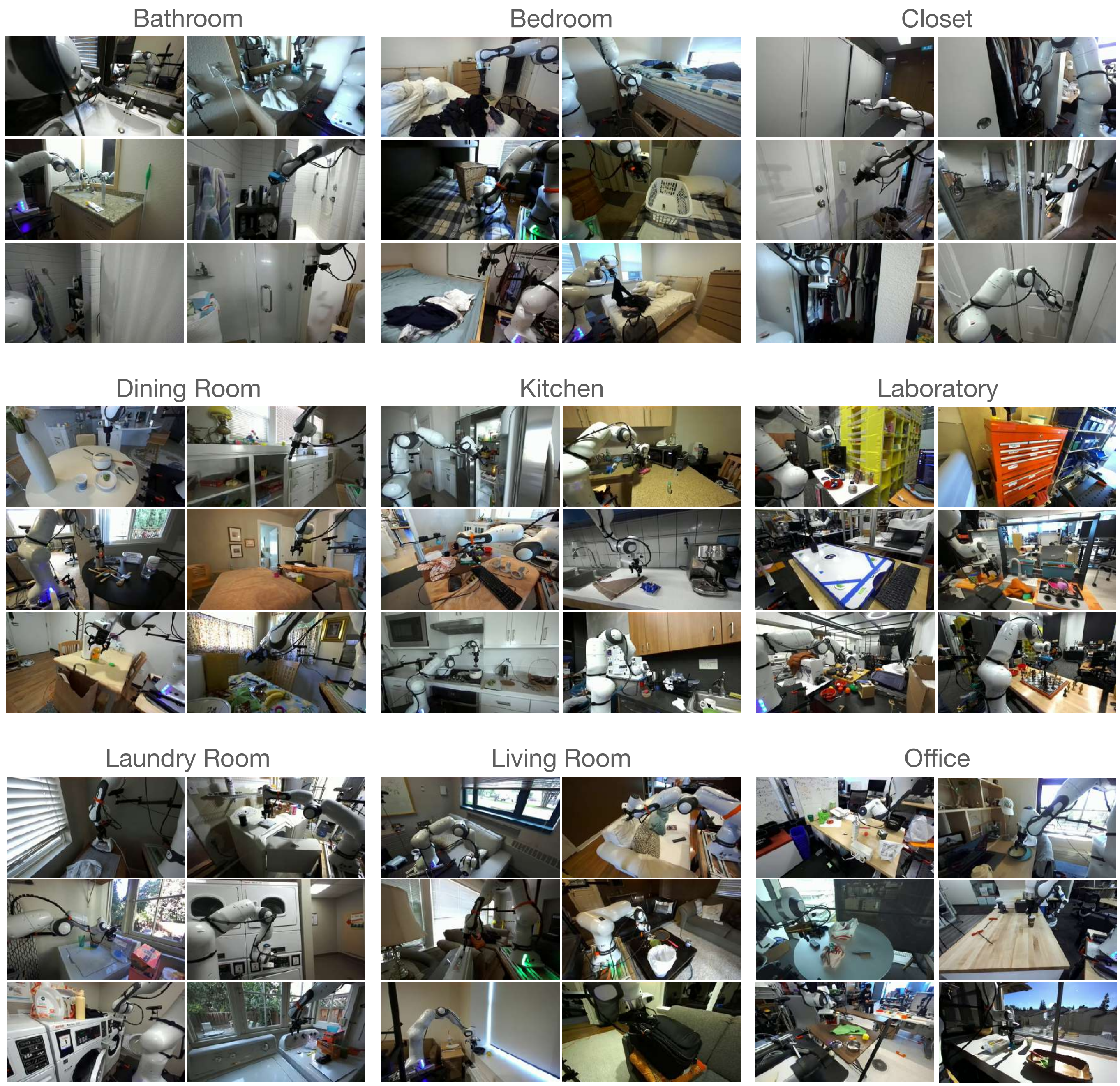

Fig. 12: Qualitative examples of scenes in DROID. We use GPT-4V to categorize scenes into 9 scene types. DROID contains robot manipulation demonstrations in a wide range of “in-the-wild” scenes across 52 buildings. Please check out the *interactive* dataset viewer included in the supplementary material to browse the dataset videos.

mark success if the drawer is closed with the eraser in it. The out of distribution variant consists of adding distractor objects on the desk, specifically a calculator, three whiteboard markers, and a clip. We found that adding distractor objects inside the desk caused all policies to fail.

Cook Lentils: A long horizon task in a kitchen setting, where the task is to remove the lid off a pan, pour lentils into the pan, and turn on the stove. The object positions are fixed. We collect 50 demonstrations, and mark success if all 3 stages of the task are successfully completed. The out of distribution variant consists of adding several distractor objects and a camera shift.

Appendix F
Diffusion Policy Details

In Section F-A we discuss the policy architecture and hyperparameters used for all policy learning experiments. Then in Section F-B, we describe how the various datasets in the paper are used to construct training batches for policy learning.

```
Please classify the image into one of the following categories.
Respond with just the category name (do not include the category number).
1. Industrial office: industrial office tables and chairs, conference rooms,
conference TVs
2. Industrial kitchen: industrial refrigerator, sink, coffee maker
3. Industrial dining room: industrial setting with dining tables
4. Home office: desk or desk chairs in a home setting
5. Home kitchen: refrigerator, kitchen sink, kitchen tabletop in a home setting
6. Home dining room: dining table, dining chairs, in a home setting
7. Bedroom: room with a bed
8. Bathroom: Showers, baths, toilets, bathroom sinks
9. Living room: places with couches, armchairs, coffee tables, tvs in a home
setting
10. Hallway / closet: areas between rooms, situations where the robot is
interacting with a door or objects in a closet
11. Other: any other location that does not fit into those categories
12: Unknown: a scene that's too hard to classify because the image is dark or too
close up
```

Listing C.1: The prompt provided to GPT4V in order to classify scene types.

A. *Diffusion Policy Architecture and Hyperparameters*

We build our diffusion policy [6] training pipeline on the Robomimic codebase [31], which provides high quality implementations of a number of different imitation learning and offline RL algorithms. Given camera observations and a language instruction for the task, within Robomimic, we define observation keys corresponding to each of the two external camera observations, a frozen DistilBERT [37] language embedding, the 3D cartesian position of the gripper, and the gripper state, which measures the degree to which the gripper is closed.

For each of the camera observations, we first downsample each image to a resolution of 128×128 and apply color jitter and random cropping as a form of data augmentation. We then use a ResNet-50 visual encoder pre-trained on ImageNet [10] to produce embeddings for each of the visual inputs. These embeddings are directly concatenated with all of the other observation keys. These concatenated features are then fed through an Observation Processing MLP with layers defined in Table II. The output of this MLP is then passed to a U-Net diffusion head which generates action trajectories. We use an observation horizon of 2 to condition the diffusion head, diffuse out 16-step action sequences, and step the first 8 actions open loop before re-running policy inference. All relevant hyperparameters are defined in Table II. In line with prior work [6], we use DDIM to diffuse out action trajectories for improved efficiency.

All experiments use the training hyperparameters in Table II with one exception: for the Cook Lentils task OOD experiment, we train all policies with 50000 training steps due to the increased complexity of the task.

Table II: **Training Hyperparameters**

Hyperparameter	**Value**
Batch Size	128
Optimizer	Adam
Learning Rate	1e-4
Learning Rate Scheduler	Linear
Train Steps	25000
Observation Processing MLP	[1024, 512, 512]
Image Resolution	(128, 128)
Crop Height	116
Crop Width	116
Diffusion Method	DDIM
EMA Power	0.75
U-Net Hidden Layer Sizes	[256, 512, 1024]
Observation Horizon	2
Prediction Horizon	16
Action Horizon	8

B. *Training Batch Construction*

For each evaluation task, we train policies with 3 different methods of constructing training batches:

- **No Co-training**: trains a state of the art diffusion policy [6] using samples from the in-domain demonstrations only.
- **DROID (Ours)**: Trains a diffusion policy, but mixes batches 50/50 between in-domain demonstrations and DROID trajectories. For this experiment, we consider the first 40K successful trajectories in DROID for which language annotations were available at the time of policy training. For the *scene diversity* experiments, we use a 7362 trajectory subset of these 40K trajectories.
- **OXE [33]**: Trains a diffusion policy, but mixes batches 50/50 between in-domain demonstrations and trajectories from a curated subset of the Open X-Embodiment

dataset [33] (OXE) used in Octo Model Team et al. [32]. We also omitted data from the language table split of OXE to bring down the number of trajectories to a manageable scale (400K trajectories).

Each of the above settings defer only in the data used to construct each training batch: otherwise all policies have identical architectures and are trained with the same training parameters specified in Section F-A.

The in-domain demonstrations used for policy training consist of only the demonstrations collected for each evaluation task in Section E with one exception: for the Toasting and Close Waffle Maker Tasks, one multi-task policy is trained on the combination of their demonstrations. Thus, in this case, the **No Co-training** policy defined above trains one diffusion policy on the *combined* in-domain demonstrations, while the Co-training experiments sample batches via a 50/50 split of data from these combined in-domain demonstrations and data from either DROID or **OXE [33]**.

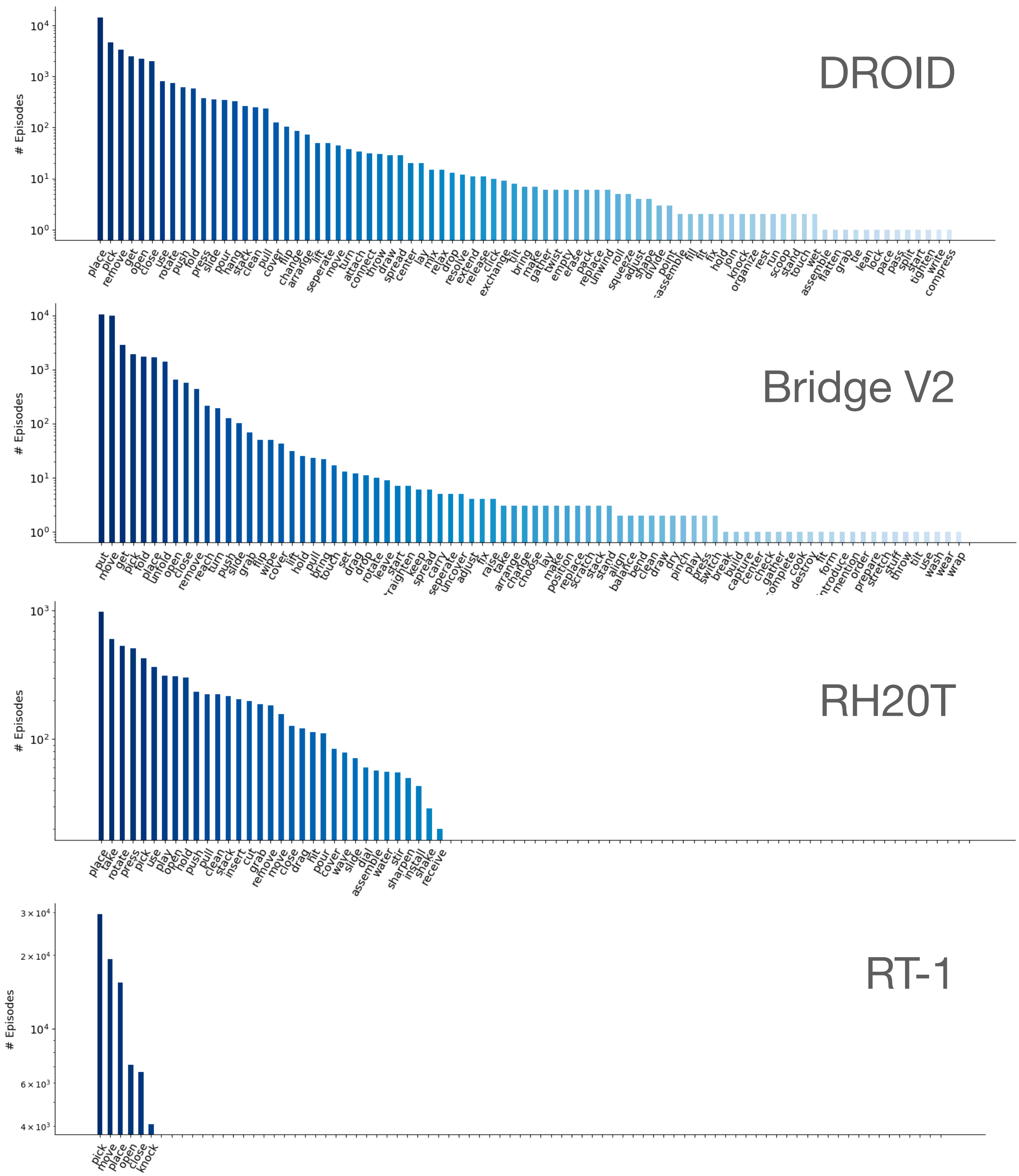

Fig. 13: Distribution of skills, i.e., verbs, for DROID and existing large robot manipulation datasets. **Top to bottom**: DROID, Bridge V2 [49], RH20T [13], RT-1 [2]. DROID features a long tail of diverse verb classes that is only matched by Bridge V2, while the RH20T and RT-1 datasets have a more constrained set of skills.

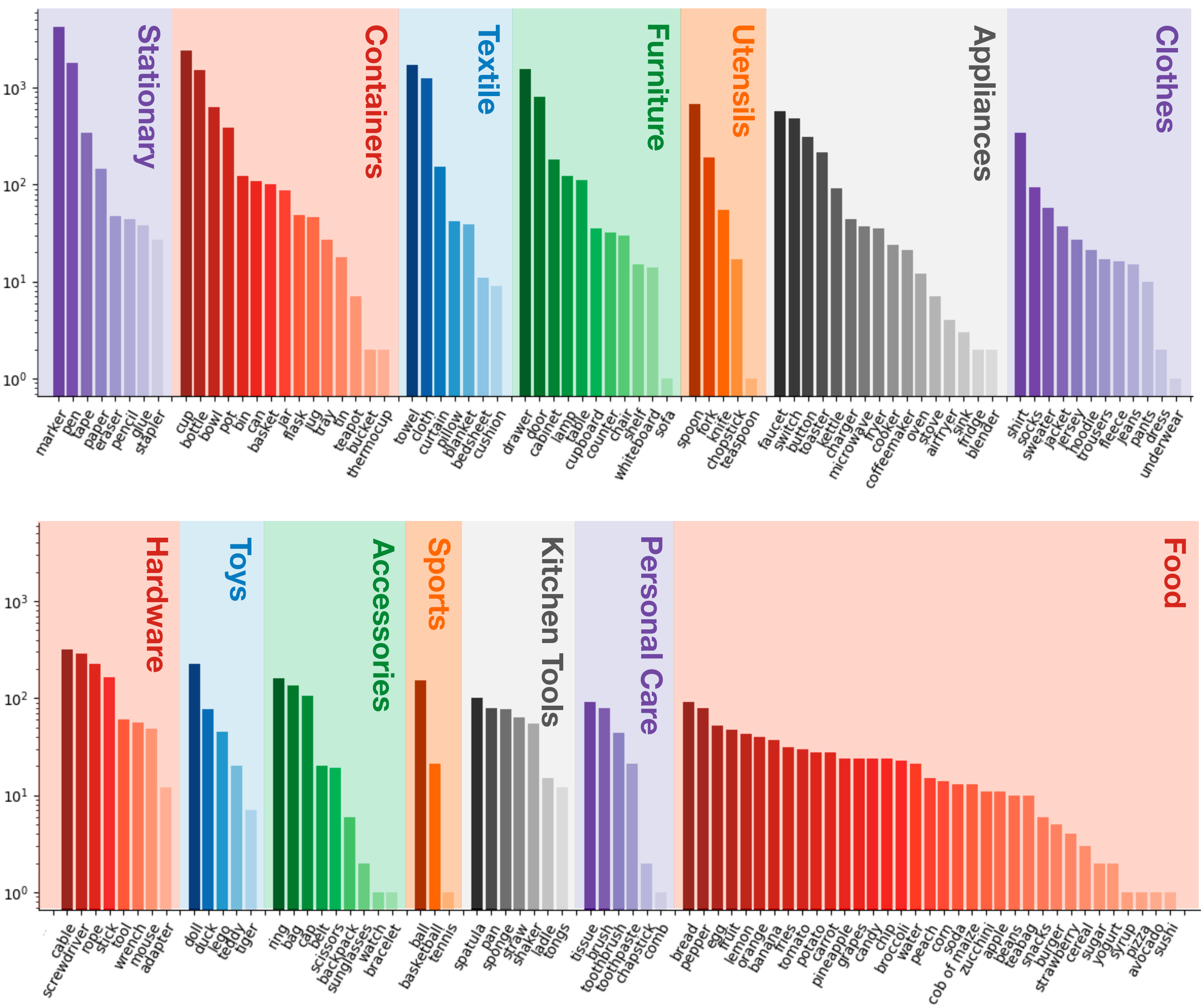

Fig. 14: Distribution of interacted objects in DROID, grouped by category. The robot interacts with a wide range of everyday objects.

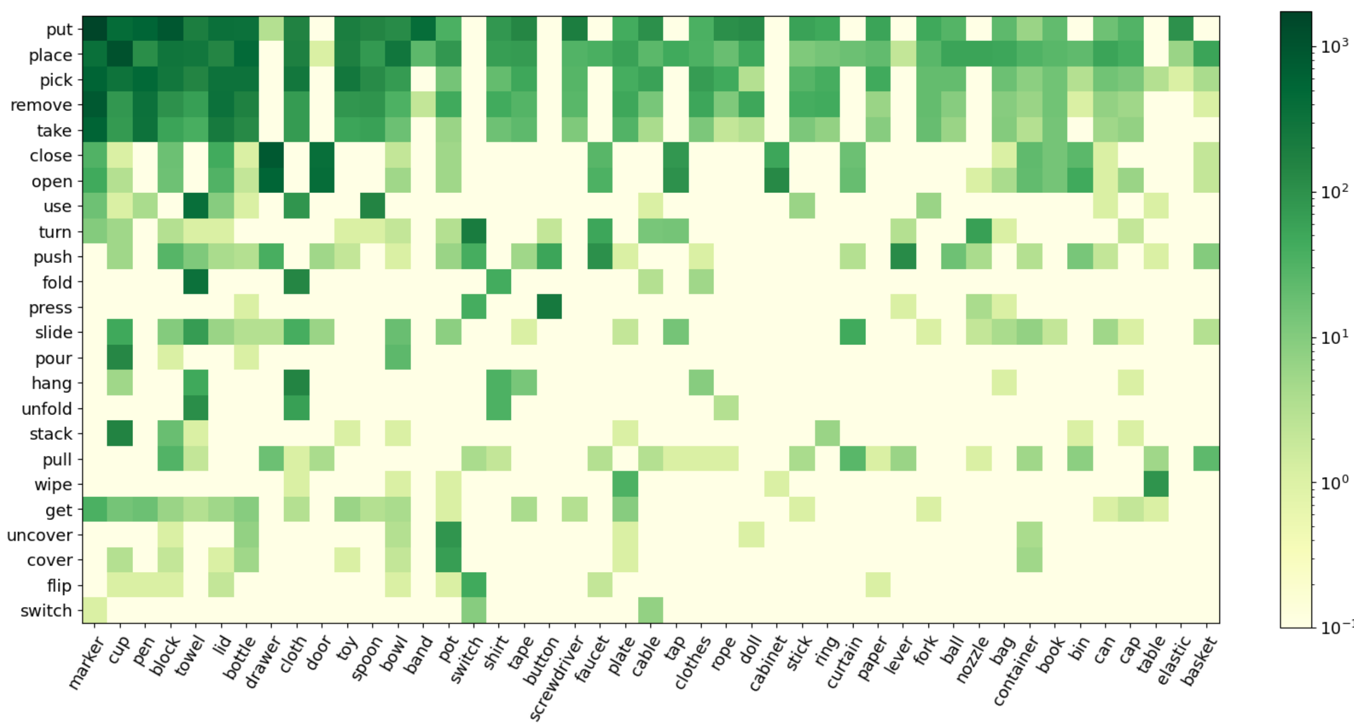

Fig. 15: Joint distribution of verbs and interacted objects in DROID. Most objects have a diverse range of interactions that are performed on them.