RoboCasa365: A Large-Scale Simulation Framework for Training and Benchmarking Generalist Robots

Anonymous authors
Paper under double-blind review

Abstract

Recent advances in robot learning have accelerated progress toward generalist robots that can operate across diverse tasks and environments. Yet despite this momentum, it remains difficult to gauge how close we are to this goal, as the field lacks a reproducible, large-scale benchmark for systematic evaluation. To address this gap, we present RoboCasa365, a comprehensive simulation benchmark for everyday household robotics. Built on the RoboCasa platform, RoboCasa365 introduces 365 everyday tasks across 2,500 diverse kitchen environments, over 600 hours of human demonstration data and over 1600 hours of synthetically generated demonstration data, making it one of the most diverse and large-scale resources for studying generalist policies. We design the benchmark to support evaluation across key settings, including multi-task learning, robot foundation model training, and lifelong learning. Using RoboCasa365, we conduct extensive experiments comparing state-of-the-art approaches and analyze how task diversity, dataset scale, and environment variation influence generalization. Our results provide new insights into what factors most strongly affect the performance of generalist robots and help inform strategies for future progress in the field.

1 Introduction

Figure 1: **Overview of RoboCasa365.** RoboCasa365 is a large-scale simulation framework for training and benchmarking generalist robots. RoboCasa365 includes 365 everyday tasks, 2,500 diverse kitchen scenes, over 600 hours of human demonstration data and over 1600 hours of synthetically generated demonstration data, and systematic benchmarks to train and evaluate generalist robot models.

Recent advances in robot learning have enabled significant progress toward generalist robots capable of performing a wide range of tasks across diverse environments. Several efforts have focused on collecting large-scale robot datasets in the real world and training high-capacity models capable of performing complex behaviors (Black et al., 2024; Team et al., 2025a; NVIDIA et al., 2025; Intelligence et al., 2025). These models have demonstrated promising signs of generalization to novel objects, environments, and tasks, suggesting that training generalist policies is within reach.

Despite these advances, two major challenges remain. First, training generalist robots requires vast amounts of robot interaction data. Although recent datasets have grown substantially, they remain limited in scale, diversity, and task coverage, which constrains the ability to train robust, generalist policies. Second, evaluating and benchmarking these systems in the real world is resource-intensive, time-consuming, and subject to noise, making reproducible and systematic comparisons across methods difficult.

Simulation provides a practical avenue for addressing these limitations. With simulation, we can generate large-scale interaction datasets, covering an effectively infinite variety of tasks and environments (Mandlekar et al., 2023; Jiang et al., 2025). Simulation also enables rapid experimentation, controlled evaluation, and reproducible benchmarking that would be infeasible in real-world robotics Saxena et al. (2025). Together, these capabilities make it possible to generate data, train policies, and systematically evaluate generalist robots at scale.

However, existing simulation frameworks fall short of this potential. Most current tools support only limited tasks and environments, often focusing on simple object manipulation or single-room scenarios Zhu et al. (2020); James et al. (2020); Wang et al. (2023). The datasets they generate are small relative to the diversity and complexity of real-world robotics challenges, and benchmarking is typically confined to these narrow conditions (Liu et al., 2023; Mandlekar et al., 2021). Consequently, it remains difficult to study how task diversity, environment variation, and dataset scale affect policy generalization.

To address these gaps, we introduce RoboCasa365, a comprehensive simulation benchmark for everyday household robotics. RoboCasa365 is built on top of the RoboCasa simulation framework by Nasiriany et al. (2024), and is structured around four core components:

Diverse environments: The benchmark includes 2,500 unique kitchen scenes modeled from real kitchens across the United States. These scenes capture a wide spectrum of layouts, object configurations, and visual variations, providing realistic contexts for a variety of everyday tasks.

Comprehensive tasks: RoboCasa365 defines 365 tasks spanning over 50 distinct kitchen activities, including manipulation, semantic reasoning, long-horizon planning, and memory-dependent tasks. This task diversity allows evaluation across multiple dimensions of generalist robot capability.

Large-scale data: The benchmark provides over 2,000 hours of robot interaction data. This includes 675 hours of human demonstration data and an additional 1615 hours of synthetic demonstration data using the MimicGen data generation tool (Mandlekar et al., 2023) to significantly expand the quantity of data.

Systematic benchmarking: RoboCasa365 supports rigorous evaluation across three learning settings: massively multi-task training, foundation model training, and lifelong learning. The benchmark is designed to facilitate reproducible, large-scale experiments and in-depth analysis of which data and environment factors most strongly influence generalization.

By integrating these elements, RoboCasa365 provides a large, diverse, and systematically structured resource for studying generalist robots in simulation. It enables researchers to scale training, run reproducible evaluations, and analyze the impact of task and environment diversity on policy generalization. Using RoboCasa365, we conduct extensive experiments to compare state-of-the-art methods, evaluate learning strategies, and investigate the factors that most strongly drive performance in generalist robot learning.

2 Related Work

Robot Simulation Frameworks. There is a long list of robot simulation frameworks (Zhu et al., 2020; Gu et al., 2023; Mittal et al., 2023; Tao et al., 2025; Szot et al., 2021; Kolve et al., 2017; Li et al., 2023; 2024; Liu et al., 2023; Deitke et al., 2022). Some are focused on tabletop settings (Zhu et al., 2020; Liu et al., 2023; Li et al., 2024; James et al., 2020). We focus on simulating entire room-scale scenes, similar to some other prior works (Li et al., 2023; Nasiriany et al., 2024; Szot et al., 2021; Kolve et al., 2017). Our work is unique in that it features hundreds of tasks across thousands of unique scenes, and features large-scale high quality demonstration datasets, and comes with a suite of benchmarks for training and evaluating generalist robot models. To our knowledge our work is the first simulation framework to satisfy all of three criteria.

Datasets and Benchmarks for Generalist Robots. There have been numerous efforts towards collecting large robot datasets in the real world (Brohan et al., 2022; Walke et al., 2023; Khazatsky et al., 2024; Collaboration et al., 2023). Evaluating and benchmarking policies trained on these datasets in the real world is challenging due to the resources needed to run large-scale systematic evaluations, despite several recent approaches towards this goal (Atreya et al., 2025; Zhou et al., 2023; Yenamandra et al., 2023; Zhou et al., 2025; Krotkov et al., 2016; Correll et al., 2018). Simulation allows the ability to run large-scale benchmarks at scale. However, most simulation benchmarks are confined to a very narrow distribution of tasks and environments (Mandlekar et al., 2021; Zhu et al., 2020; Liu et al., 2023; Team et al., 2025b), Li et al. (2023) bring forth diverse environments and tasks but lack accompanying large-scale datasets. Nasiriany et al. (2024) include 100k demonstrations spanning 30 tasks and 100 scenes. In contrast, our datasets comprise over 650k demonstrations spanning over 300 tasks, and span 2500 unique scenes. While prior work focuses on benchmarking specific methods such as multi-task training Team et al. (2025b); Nasiriany et al. (2024) and lifelong learning Liu et al. (2023), we provide a comprehensive suite of benchmarks to systematically study multi-task training, foundation model training, and lifelong learning.

Training Generalist Robots. There is a long body of work on learning generalist robot policies from large diverse robot datasets (Octo Model Team et al., 2024; Collaboration et al., 2023; NVIDIA et al., 2025; Brohan et al., 2023; Kim et al., 2024; Shukor et al., 2025; Wen et al., 2025). In our work we aim to be agnostic to the choice of model, and instead create benchmarks to systematically assess the capabilities of these models across distinct settings, including multi-task training, pretraining and post-training, and lifelong learning.

3 RoboCasa365: Large-Scale Simulation of 365 Everyday Tasks

We present RoboCasa365, a large-scale simulation framework for training and benchmarking generalist robots. We use the existing RoboCasa simulation framework (Nasiriany et al., 2024) as the starting ground for RoboCasa365 and make significant contributions to scale up the assets, environments, tasks, and datasets. We also establish a rigorous benchmark to study state-of-the-art policy learning methods, which we will outline in Section 4. In the following sections, we will outline the individual components of this simulation framework, including assets, scenes, tasks, and datasets.

3.1 Expanding the Scope of Assets

RoboCasa features a diverse arrangement of objects and interactable fixtures and appliances, with a focus for use in kitchen environments. We use the existing library of 2,509 objects from Nasiriany et al. (2024), spanning 153 object categories. In addition to these, we source an additional collection of high-quality 3D assets spanning 57 object categories. These are high-quality 3D assets sourced from artists and edited to preserve strict quality standards. We use these new objects to support new tasks and to populate various areas of kitchen scenes generally. A complete inventory is provided in Appendix B.1.

In addition to the 3D assets, we significantly expand the scope of interactable fixtures and appliances in the kitchen environment. RoboCasa (Nasiriany et al., 2024) includes a total of 20 interactable fixtures and appliances spanning 4 categories: sinks, coffee machines, stoves, and microwaves. We significantly expand the scope of these assets to 456 instances spanning 12 categories. We include new categories of appliances, such as toasters, toaster ovens, stand mixers, blenders, and electric kettles. All of these appliances are articulated, including fridges, ovens, dishwashers, which were not previously articulated under RoboCasa. We model these assets using the same format as RoboCasa, as MJCF objects with annotations of the regions. For each category, we include between 20 to 50 instances, in order to ensure that there is sufficient diversity to support generalization to novel instances. We provide a complete inventory of our fixtures and appliances in Appendix B.2.

3.2 Diverse Kitchen Scenes

Achieving generalization in robot learning requires exposure to a wide range of training environments; we address this need by providing thousands of diverse kitchen scenes spanning a broad spectrum of household settings. We categorize these scenes into *pretraining* and *post-training* splits. Our goal is to use the pretraining kitchen scenes for large-scale data collection and synthetic data

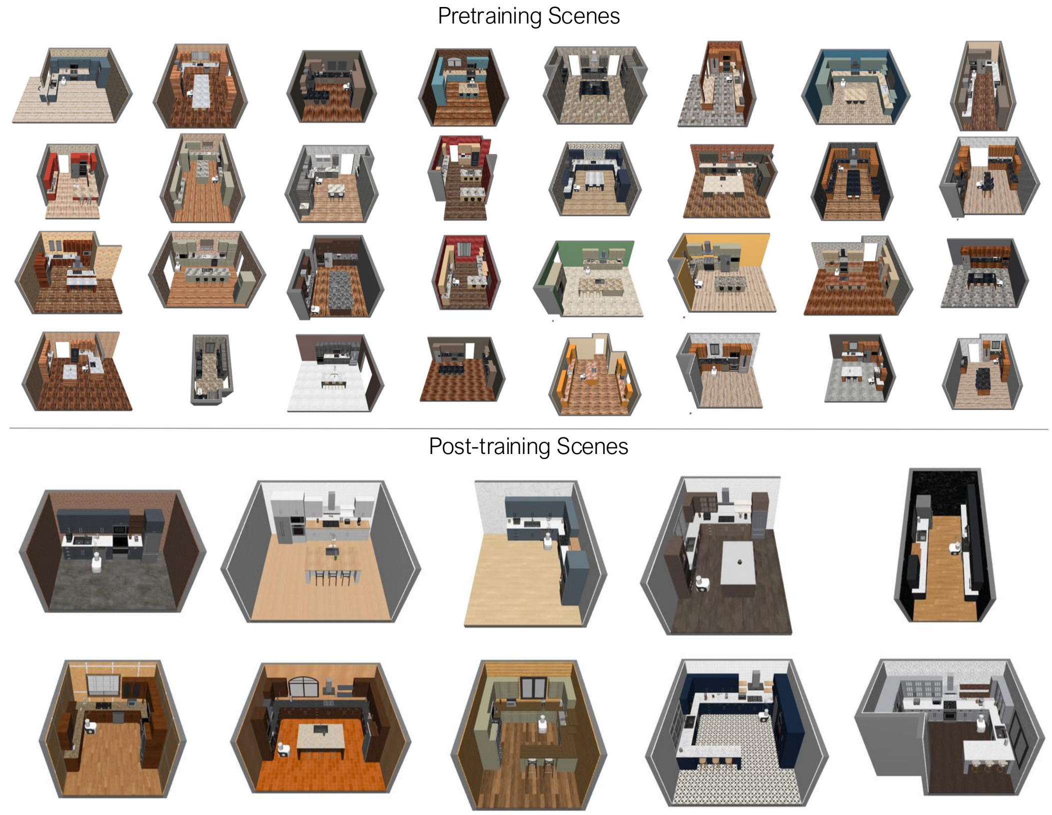


Figure 2: **Kitchen Scenes.** Our simulation framework features 2500 distinct kitchen scenes for pretraining (top, representative samples shown), and 10 distinct kitchen scenes for post-training (bottom, all scenes shown).

generation pipelines; we use the post-training kitchen scenes for post-training data collection and to run most of our experiment evaluations. Using the etymology from Nasiriany et al. (2024), we define each kitchen scene as a combination of layout and style, where the layout defines the floor plan, and the style defines the specific selection of fixtures, appliances, and textures used in the kitchen. We can configure each kitchen scene to use any combination of layout and scene.

For our post-training kitchens, we use the 10 layouts and 10 styles defined by Nasiriany et al. (2024) in RoboCasa, where each layout is matched with a specific style, for a total of 10 kitchen scenes. For our pretraining kitchen scenes, we create 50 distinct new layouts. In order to capture the distribution of diverse scenes, we source our kitchens from 50 real-world homes with active listings on Zillow.com, a real estate marketplace. These homes span diverse geographic locations across the United States. We build digital cousin (Dai et al., 2024) replicas for each of these environments, making sure to match the floor plan as closely as possible. In addition to these layouts, we create 50 distinct styles. We ensure that the pretraining and post-training styles do not overlap in the selection of the fixtures, appliances, or environment textures used. Together, we have a total combination of 50 layouts $\times$ 50 styles, for a total of 2,500 kitchen scenes. We provide an overview of the pretraining and post-training kitchen scenes in Figure 2.

3.3 Suite of 365 Everyday Tasks

We aim to provide a diverse set of tasks to support sharing knowledge across tasks and generalizing to new tasks. Nasiriany et al. (2024) define two broad categories of tasks: *atomic tasks*, which represent the execution of a single skill, and *composite tasks*, which involve executing a sequence of skills. Nasiriany et al. (2024) define eight foundational skills: (1) pick-and-place, (2) opening and closing doors, (3) opening and closing drawers, (4) turning levers, (5) turning knobs, (6) pressing buttons, (7) insertion, and (8) navigation. We adopt these skills as the basis for our atomic tasks. In addition to the 25 atomic tasks in RoboCasa, we create an additional set of 40 new atomic tasks to

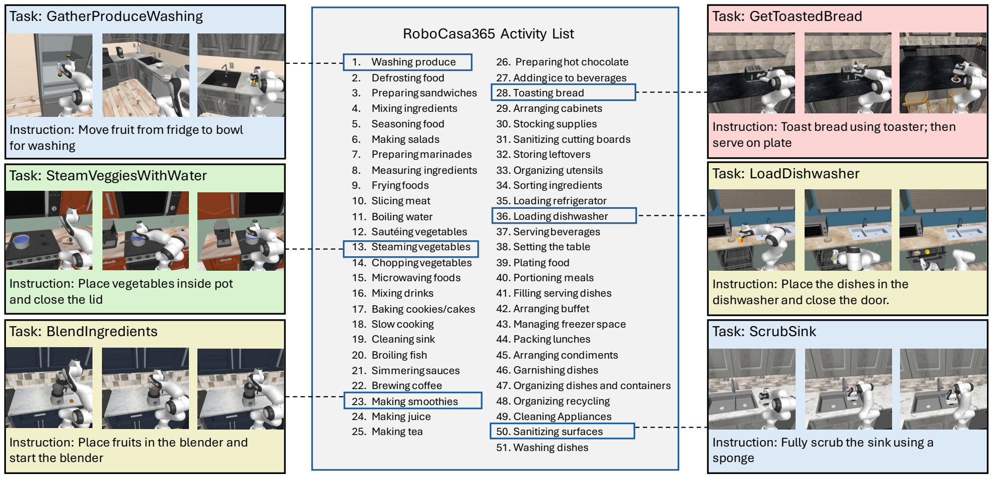


Figure 3: **Composite Tasks.** RoboCasa365 features 300 composite tasks that involve a sequence of skills. We use large language models to generate a set of high-level activities, and for each activity, a set of task blueprints. We highlight representative tasks for selected activities here.

support various new appliances and new behaviors afforded by our simulator. We provide the entire list of 65 atomic tasks in Appendix D.1.

For our composite tasks, we follow the framework established by Nasiriany et al. (2024), where we use large language models to solicit task blueprints. The process follows two stages. First, we prompt LLMs to give a list of *activities* representing high-level groups of tasks in kitchen environments. We retrieve a list of the top 50 activities, such as boiling water, toasting bread, brewing coffee, washing dishes, and storing leftovers, to name a few. For each activity, we then prompt the LLM to provide task blueprints, which consist of the name of the task, a high-level description of the task, the objects and fixtures involved, and the sequences of skills needed to solve the task. We then proceed to write code for the tasks based on these blueprints. We use 83 of the existing composite tasks from RoboCasa and generate an additional set of 217 new composite tasks, for a total of 300 composite tasks. We outline the full list of activities and representative composite tasks in Figure 3. In total, our benchmark includes 365 everyday tasks: 65 atomic tasks and 300 composite tasks. Out of these, 247 require mobile manipulation, while 118 can be performed without mobility.

3.4 Datasets

We provide a large collection of robot datasets covering all of our tasks. Broadly, our datasets are divided into two categories: *pretraining datasets* for data from the pretraining scenes, and *post-training datasets* from the post-training scenes.

3.4.1 Pretraining datasets

Out of the 365 total tasks outlined in Section 3.3, our pretraining data covers 300 tasks, with 65 atomic tasks and 235 composite tasks. For each of these 300 tasks, we collect 100 human demonstrations per task via robot teleoperation. This results in 30k human demonstrations total for pretraining. For our data collection, we use the Franka Panda Emika robot, equipped with an Omron mobile base (Haviland et al., 2022), and in principle, our simulation framework can support data collection with other mobile manipulators and humanoid platforms.

We also use the MimicGen generation system (Mandlekar et al., 2023) to generate large-scale synthetic data across 60 atomic tasks. For each task, we use the 100 human demonstrations previously collected as seed demonstrations, and generate 10k demonstrations, effectively scaling data for these tasks by 100×.

3.4.2 Post-training datasets

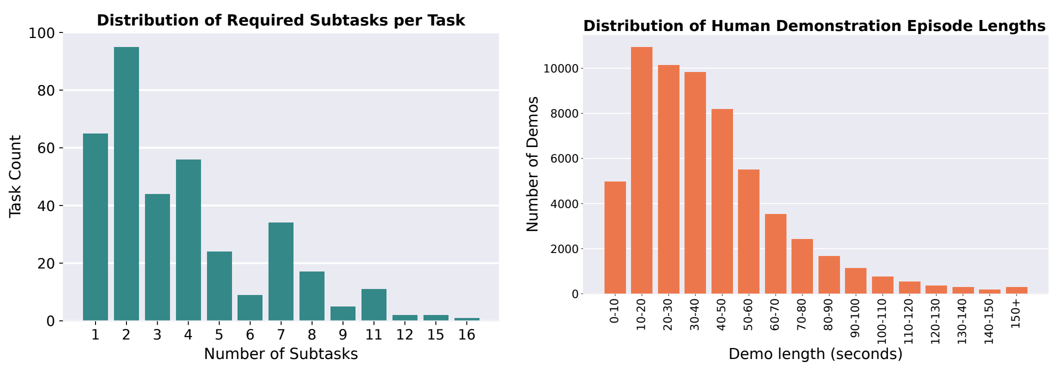
For our post-training data, we choose a set of 50 representative tasks out of the suite of 365 tasks, grouped into three splits:

- `Atomic` (18 tasks): We include 18 representative tasks out 65 total atomic tasks in the benchmark.
- `Composite-Seen` (16 tasks): We choose 16 representative composite tasks spanning 16 activities. These include a mix of short and long-horizon tasks, with some involving 2 subtasks and the longest task involving 15 subtasks.
- `Composite-Unseen` (16 tasks): To test the effect of our pretraining data, we also choose 16 composite tasks that are unseen in the pretraining data. These tasks are of similar difficulty to the composite seen tasks, but focus on another distinct set of 16 activities.

We list the entire set of 50 post-training tasks in Appendix D.2. For each of these tasks, we collect 500 human demonstrations via robot teleportation, for a total of 25k demonstrations.

3.4.3 Dataset statistics

We provide a high-level overview of our datasets in Appendix E. Our pretraining synthetic demonstration dataset spans the highest amount of data, with 1615 total hours, followed by human pretraining data (482 hours), and then human post-training data (193 hours). In Figure 4a we report the distribution over the number of subtasks required for each of our 365 tasks. Most tasks require one or two subtasks, but there are a few tasks that require 15 or more subtasks to complete. In Figure 4b, we report the distribution of episode lengths across all pretraining and post-training human data (55k episodes). The majority of episodes range from 10 to 60 seconds, with a long tail end for longer horizon episodes, some going beyond 3 minutes.



(a) Distribution of required subtasks per task.

(b) Distribution of demonstration lengths.

Figure 4: Distribution of task lengths (by number of subtasks) and dataset episode lengths (by number of seconds). We observe a long tail of tasks and data representing long-horizon behaviors.

4 Experiments

In our experiments, we conduct a systematic study to understand the key factors that influence the training of generalist robots. To this end, we design a comprehensive suite of benchmarks aimed at answering the following questions:

1. How well do generalist robots perform when trained on large multi-task datasets?
2. What role does pretraining data play, and to what extent can it improve learning of downstream tasks?
3. How effectively do agents learn in lifelong learning settings?
4. How does the scope and composition of pretraining data impact performance on downstream tasks?

4.1 Multi-task training

We begin by investigating how state-of-the-art methods perform when trained on massively multi-task datasets. This evaluation is a critical step toward developing generalist robots that can not only master a wide range of behaviors but also adapt to entirely novel tasks beyond their training data.

We train language-conditioned vision-based policies on the mixture of 300 pretraining human datasets outlined in Section 3.4.1. Each task has 100 human demonstrations, for a total of 30k demonstrations. Our experiments feature three state-of-the-art methods: **Diffusion policy** (Chi et al., 2023), $\boldsymbol{\pi_0}$ (Black et al., 2024), and **GR00T N1.5** (NVIDIA et al., 2025).

We train a multi-task language-conditioned policy for each method. We use the pretrained checkpoints released publicly for π_0 and GR00T N1.5 as the base model for training our models. We provide details on the training protocols for each method in Appendix F.

We evaluate on the 50 tasks outlined in Section 3.4.2: `Atomic`, `Composite-Seen`, and `Composite-Unseen`. Note that the `Composite-Unseen` tasks represent unseen tasks in the pretraining data, and thus our evaluation for these tasks are zero-shot and are aimed at understanding generalization to novel tasks. We run evaluations in the *pretraining* kitchen scenes for each task and report average task completion success rates across all methods. See Appendix F for details about the evaluation protocol.

Task Split	**Diffusion Policy**	π_0	**GR00T N1.5**
`Atomic`	15.7	36.3	**43.0**
`Composite-Seen`	0.2	5.2	**9.6**
`Composite-Unseen`	1.25	0.7	**4.4**
Average	6.1	15.0	**20.0**

Table 1: **Multi-task training results.** We compare state-of-the-art policy learning approaches on our human pretraining data across 300 tasks, and report task success rates (%) across seen and unseen tasks. We see that learning longer-horizon composite tasks is more challenging, and that performance suffers when evaluating on unseen tasks.

We report results in Table 1. Overall, we see that across all methods, learning Atomic tasks is the easiest, followed by learning Composite-Seen tasks, and Composite-Unseen tasks. This is reasonable, as the Atomic tasks are shorter horizon tasks which present fewer learning challenges for imitation learning (Ross et al., 2011), and the lower performance on Composite-Unseen tasks is due to the fact that the model has never been trained on these tasks. Overall, GR00T N1.5 performs the best among all three methods. It shows non-zero success rates on Composite-Unseen tasks, a sign of its generalization abilities. Diffusion Policy performs the worst, highlighting how high-capacity vision-language-action models can better fit large, diverse multi-task robot datasets. While our multi-task learning experiments show that GR00T outperforms the Diffusion Policy and π_0 baselines, we do not claim that GR00T is definitively the superior method. Performance can be influenced by many factors, including the amount of compute used (e.g., batch size), hyperparameters such as learning rate, and whether the visual or language backbones are fine-tuned. Overall, we see a significant opportunity for future methods to improve upon these results.

4.2 Foundation Model Training

In our next experiment, we are interested in studying foundation model training, i.e., training with our pretraining datasets, followed by fine-tuning on our post-training datasets. This learning paradigm has been established by numerous prior works in robotics (Black et al., 2024; NVIDIA et al., 2025), with evidence that pretraining can aid learning downstream tasks in a more robust and data-efficient manner. In our experiments, our pretraining data includes human datasets across 300 tasks (482 hours), and synthetic data across 60 atomic tasks (1615), while our post-training data include human datasets across 50 tasks (193 hours). Out of the 50 post-training tasks, 34 are are also represented in the human pretraining data (Atomic and Composite-Seen tasks), and the post-training data includes an additional 16 composite tasks that are not seen in the pretraining data (Composite-Unseen). We first train on all of our pretraining datasets (see Section 3.4.1), followed by post-training on three separate post-training split datasets (Atomic, Composite-Seen, Composite-Unseen; see Section 3.4.2). We compare post-training on different amounts of post-training data, with 50, 150, and 500 demos per task, representing 10%, 30%, and 100% of the total post-training data.

Unless otherwise noted, we use GR00T N1.5 as the model for these experiments and all subsequent experiments, due to its superior performance from the experiments in the preceding section. We compare pretraining only, post-training only, and pretraining followed by post-training. After

Task Type	Pretraining Only	Post-training Only			Pretraining + Post-training		
		10%	30%	100%	10%	30%	100%
`Atomic`	41.9	38.7	50.6	60.6	56.9	59.1	**68.5**
`Composite-Seen`	0.0	11.0	22.7	35.0	25.4	34.6	**40.6**
`Composite-Unseen`	0.2	11.2	27.5	33.3	22.7	30.8	**42.1**
Average	15.1	21.0	34.3	43.7	35.9	42.2	**51.1**

Table 2: **Foundation model training results**. Comparing the impact of pretraining and post-training on learning downstream tasks.

training, we evaluate the model across the 50 post-training tasks in the post-training kitchens. See Appendix F for a detailed discussion of the training and evaluation protocols. We report experiment results in Table 2.

We see that with pretraining alone, the model performs above 40% on the atomic tasks but performs very poorly on the composite tasks. For post-training only, we see more capable policies but they require a high amount of data to be performant. By using pretraining, we see a significant improvement in model performance. These gains are especially pronounced for the Composite-Unseen tasks (see Table 2). We visualize the improvement in performance in Figure 5, visualizing the average task success rates from Table 2. We observe a roughly $3\times$ improvement in data efficiency, i.e., pretraining helps achieve roughly the same performance as post-training only with 3x additional post-training data. In Appendix G.2 we run a rigorous set of evaluations on the robustness of these learned models, identifying how the model is sensitive to different factors.

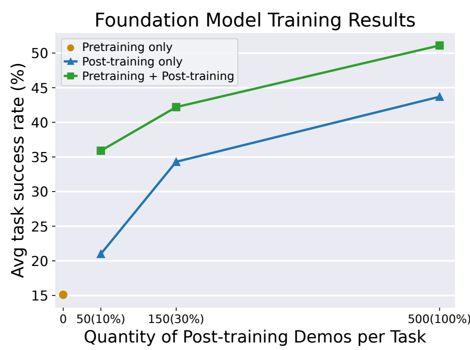


Figure 5: **Foundation Model Training Results.** Pre-training enables learning downstream tasks with significant gains in data efficiency.

4.3 Lifelong Learning

In contrast to the conventional two-stage paradigm of pretraining followed by post-training, robots in the real world are often required to acquire new skills continuously. This setting, known as lifelong learning, involves learning tasks over a sequence of phases. The central challenge lies in using prior knowledge to learn new tasks effectively, while still retaining how to perform previously learned tasks. We design a lifelong learning benchmark designed to assess these capabilities. In our experiments, the robot is tasked with learning a series of tasks over four phases. Each phase involves learning progressively longer horizon tasks. Phase 1 involves learning 65 atomic tasks, phase 2 involves learning 20 new composite tasks with 2 to 3 stages, phase 3 involves learning 20 new composite tasks with 4 or 5 stages, and phase 4 involves learning 20 new composite tasks with 6 or more stages. We define "stage" as the invocation of one of the robot skills defined by Nasiriany et al. (2024), such as pick-and-place, turning knobs, and navigation. We use pretraining datasets for these phases; phase 1 includes all human and MimicGen datasets for atomic tasks, while phases 2, 3, and 4 feature human datasets.

Phase	Atomic Tasks	2-3 Stage Tasks	4–5 Stage Tasks	6+ Stage Tasks
Phase1	41.5	-	-	-
Phase2	13.9	24.5	-	-
Phase3	13.9	4.8	11.3	-
Phase4	10.6	1.7	2.7	4.3

Table 3: **Lifelong learning results.** We train across four phases with progressively longer horizon tasks. After each phase, we report task success rates (%) across all tasks seen in the current and previous phases.

For each phase N, we take the model previously trained from phase $N-1$, and fine-tune it for data pertaining to the tasks in phase N. After training completes for phase N, we run evaluations for tasks from phase 1 through phase N in the pretraining kitchens and report results. We report results in Table 3. We make two distinct observations. First, we see that the success rates steadily drop as we learn progressively longer horizon tasks in each new phase (see the diagonal entries of the table). This is intuitively the case, as learning longer horizon tasks can demand higher data requirements. Second, we see that the performance on previously learned tasks steadily drops with each new phase. This highlights the catastrophic forgetting problem, i.e., performance degrades on prior tasks if the agent does not continue to train on them in subsequent phases. Overall, this experiment highlights the current challenges with lifelong learning, and is a useful testbed for improving upon these results.

4.4 Pretraining Data Composition Study

In Section 4.2, we showed that pretraining brings forth significant improvements in data efficiency for learning downstream post-training tasks. In this section, we run experiments to further understand how the composition of pretraining data affects downstream performance. In our foundation model training experiments, we used all of the available pretraining data, comprising human data from 300 tasks and MimicGen data across 60 tasks (`Human300+MG60`). We compare to a variant that does not include MimicGen data and only includes the human data (`Human300`). To better understand the role of task diversity in the pretraining data, we compare two variants that include human data from 50 tasks (`Human50`). These 50 tasks include the Atomic and Composite-Seen tasks, and an additional randomly selected set of tasks. Finally, we compare with the case with no pretraining data. Our pretraining and post-training protocol is identical to the process in Section 4.2. We specifically run two separate sets of post-training experiments, one for the low-data regime with 10% of post-training data, and one for the high-data regime with 100% of the post-training data.

Post-training Data	**Pretraining Data**			
	No Pretraining	Human50	Human300	Human300+MG60
`Atomic (10%)`	38.7	52.0	**57.0**	56.9
`Composite-Seen (10%)`	11.0	26.2	**28.7**	25.4
`Composite-Unseen (10%)`	11.2	23.8	**32.3**	22.7
Average (10%)	21.0	34.7	**40.0**	35.9
`Atomic (100%)`	60.6	68.1	**70.0**	68.5
`Composite-Seen (100%)`	35.0	41.0	**41.2**	40.6
`Composite-Unseen (100%)`	33.3	38.5	**44.0**	42.1
Average (100%)	43.7	50.0	**52.5**	51.1

Table 4: **Pretraining task diversity results.** We compare the downstream effects of training on different mixtures of pretraining data.

We report evaluations in post-training kitchens in Table 4. Compared to training on all pretraining data (Human300+MG60), we find that training on just the human data (Human300) yields better downstream learning results. Although MimicGen enables the large-scale generation of synthetic trajectories, we find that the resulting demonstrations vary in quality. Developing methods that can more effectively leverage such large, mixed-quality datasets is an important direction for future work. Comparing the Human50 and Human300 settings, we see that increasing the number of pretraining tasks can enable a significant improvement in downstream post-training, especially for the low-data regime post-training setting. Notably, the biggest gains are seen for the Composite-Unseen tasks, suggesting that increasing the scope of task diversity is especially beneficial for learning novel tasks.

In addition to task diversity, we study the effects of scene diversity in pretraining on downstream performance. We report these results in Appendix G.1.

4.5 Real-World Experiments

We conduct an additional set of experiments to examine the utility of our benchmark for downstream real world applications. Our real world setup uses the DROID Panda arm (Khazatsky et al., 2024) with three cameras.

We examine four tasks in a real kitchen:

- `CloseElectricKettleLid`: close the electric kettle lid
- `PnPToasterOvenToCounter`: place the item from the toaster oven to the counter
- `PnPCounterToCabinet`: place the object form the counter to the cabinet

- `PlaceOnDishRack`: a longer horizon task, involving place two items from the sink onto the dish rack.

We collect 30 demonstrations for each of the first three tasks, and 50 demonstrations of the last task, for a total of 140 real world demonstrations. We compare the following settings:

- Real-Only: we train GR00T on the real world demonstrations (140 demonstrations)
- Sim-and-Real (ours): we first pre-train the GR00T model on our simulation tasks (we use data from the highest performing tasks in simulation), and then co-fine-tune the model on the real world demonstrations and analogue data for the four real world tasks in simulation.

Figure 6: **Real robot setup.** Our real world setup features a Panda robot arm in a real world kitchen.

To facilitate better transfer, we re-render our simulation datasets to use similar camera views as our real-world setup, following cues from prior work (Maddukuri et al., 2025).

After training, we evaluate each model in the real world, where we conduct 20 trials per task. We report task success rates in Table 5.

	CloseElectric KettleLid	PnPToasterOven ToCounter	PnPCounter ToCabinet	PlaceOn DishRack	Avg
Real-Only	**70**	70	52	55	61.8
Sim-and-Real (Ours)	**70**	**100**	**84**	**65**	**79.8**

Table 5: **Real-world evaluations.** Across four real-world tasks, we compare training on real-world data only versus training on a mixture of our simulation and real world data. By additionally using simulation data, we outperform training on real-world data only by an average of 18.1%.

Overall, the Real-Only model achieves an average success rate of 61.8%, while Sim-and-Real training achieves a success rate of 79.8%, a significant improvement. This shows the promise of our simulation benchmark, not only to study and benchmark algorithms in simulation, but also to aid policy learning in the real world.

5 Conclusion

We presented RoboCasa365, a large-scale simulation framework for training and benchmarking generalist robot models. RoboCasa365 provides 2,500 realistic kitchen environments, 365 everyday tasks spanning over 50 activity categories, and over 2,000 hours of robot interaction data, making it one of the most diverse simulation resources to date.

Using this benchmark, we conducted a systematic study along three axes: multi-task learning at scale, foundation model pretraining, and lifelong learning. Our experiments show that (i) generalist policies trained on large multi-task datasets can acquire broad competence but still face challenges with long-horizon tasks, (ii) pretraining data significantly improves downstream learning, with both scale and task diversity playing key roles, and (iii) lifelong learning remains an open challenge, with substantial trade-offs between acquiring new tasks and retaining prior knowledge.

RoboCasa365 opens several avenues for future work. First, the benchmark is currently limited to kitchen environments, raising the question of how well findings transfer to other household settings or broader domains. Second, while the dataset is large, it does not capture the full sensory and physical complexity of the real world, and bridging the gap between simulation and real-world deployment remains a significant challenge. Addressing these limitations will be an important direction for future research.

REFERENCES

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

Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, Thomas Unterthiner, Daniel Keysers, Skanda Koppula, Fangyu Liu, Adam Grycner, Alexey Gritsenko, Neil Houlsby, ManojKumar, Keran Rong, Julian Eisenschlos, Rishabh Kabra, Matthias Bauer, Matko Bošnjak, Xi Chen, Matthias Minderer, Paul Voigtlaender, Ioana Bica, Ivana Balažević, Joan Puigcerver, Pinelopi Papalampidi, Olivier Henaff, Xi Xiong, Radu Soricut, Jeremiah Harmsen, and XiaohuaZhai. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint*, 2024. URL https://arxiv.org/abs/2407.07726.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. _0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164v1*, 2024. URL https://arxiv.org/abs/2410.24164v1.

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. In *arXiv preprint arXiv:2212.06817*, 2022.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control, 2023. URL https://arxiv.org/abs/2307.15818.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023.

Open X-Embodiment Collaboration et al. Open X-Embodiment: Robotic learning datasets and RT-X models. https://arxiv.org/abs/2310.08864, 2023.

Nikolaus Correll, Kostas E. Bekris, Dmitry Berenson, Oliver Brock, Albert Causo, Kris Hauser, Kei Okada, Alberto Rodriguez, Joseph M. Romano, and Peter R. Wurman. Analysis and observations from the first amazon picking challenge. *IEEE Transactions on Automation Science and Engineering*, 15(1):172–188, 2018. doi: 10.1109/TASE.2016.2600527. URL https://doi.org/10.1109/TASE.2016.2600527.

Tianyuan Dai, Josiah Wong, Yunfan Jiang, Chen Wang, Cem Gokmen, Ruohan Zhang, Jiajun Wu, and Li Fei-Fei. Automated creation of digital cousins for robust policy learning. *arXiv preprint arXiv:2410.07408*, 2024.

Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Jordi Salvador, Kiana Ehsani, Winson Han, Eric Kolve, Ali Farhadi, Aniruddha Kembhavi, and Roozbeh Mottaghi. Procthor: Large-scale embodied ai using procedural generation, 2022. URL https://arxiv.org/abs/2206.06994.

Jiayuan Gu, Fanbo Xiang, Xuanlin Li, Zhan Ling, Xiqiang Liu, Tongzhou Mu, Yihe Tang, Stone Tao, Xinyue Wei, Yunchao Yao, et al. Maniskill2: A unified benchmark for generalizable manipulation skills. *arXiv preprint arXiv:2302.04659*, 2023.

Jesse Haviland, Niko Sünderhauf, and Peter Corke. A holistic approach to reactive mobile manipulation. *IEEE Robotics and Automation Letters*, 7(2):3122–3129, 2022.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *CoRR*, abs/2504.16054, April 2025. doi: 10.48550/arXiv.2504.16054. URL https://arxiv.org/abs/2504.16054.

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

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, 2025. URL https://arxiv.org/abs/2410.24185.

Oussama Khatib. Inertial properties in robotic manipulation: An object-level framework. *International Journal of Robotics Research*, 1995.

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, 2024.

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

Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, et al. AI2-THOR: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*, 2017.

Eric P. Krotkov, Douglas Hackett, Larry Jackel, Michael Perschbacher, James Pippine, Jesse Strauss, Gill Pratt, and Christopher Orlowski. The darpa robotics challenge finals: Results and perspectives. *Journal of Field Robotics*, 34(2):229–240, 2016. doi: 10.1002/rob.21683. URL https://onlinelibrary.wiley.com/doi/10.1002/rob.21683.

Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Michael Lingelbach, Jiankai Sun, et al. Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation. In *Conference on Robot Learning*, pp. 80–93. PMLR, 2023.

Xuanlin Li, Kyle Hsu, Jiayuan Gu, Karl Pertsch, Oier Mees, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, Sergey Levine, Jiajun Wu, Chelsea Finn, Hao Su, Quan Vuong, and Ted Xiao. Evaluating real-world robot manipulation policies in simulation. *arXiv preprint arXiv:2405.05941*, 2024.

Zhiqi Li, Guo Chen, Shilong Liu, Shihao Wang, V.S. Vibashan, Yishen Ji, Shiyi Lan, Hao Zhang, Yilin Zhao, Subhashree Radhakrishnan, Nadine Chang, Karan Sapra, Amala Sanjay Deshmukh, Tuomas Rintamaki, Matthieu Le, Ilia Karmanov, Lukas Voegtle, Philipp Fischer, De-An Huang, Timo Roman, Tong Lu, Jose M. Alvarez, Bryan Catanzaro, Jan Kautz, Andrew Tao, Guilin Liu, and Zhiding Yu. Eagle 2: Building post-training data strategies from scratch for frontier vision-language models. *arXiv preprint*, arXiv:2501.14818, 2025.

Yaron Lipman, Ricky T.Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023. URL `https://openreview.net/forum?id=KZy4-0etZgZ`.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *arXiv preprint arXiv:2306.03310*, 2023.

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

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 *Conference on Robot Learning*, 2021.

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. *arXiv preprint arXiv:2310.17596*, 2023.

Mayank Mittal, Calvin Yu, Qinxi Yu, Jingzhou Liu, Nikita Rudin, David Hoeller, Jia Lin Yuan, Ritvik Singh, Yunrong Guo, Hammad Mazhar, Ajay Mandlekar, Buck Babich, Gavriel State, Marco Hutter, and Animesh Garg. Orbit: A unified simulation framework for interactive robot learning environments. *IEEE Robotics and Automation Letters*, 8(6):3740–3747, 2023. doi: 10.1109/LRA.2023.3270034.

Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots. In *Robotics: Science and Systems (RSS)*, 2024.

NVIDIA, Nikita Cherniadev Johan Bjorck andFernando Castañeda, Xingye Da, Runyu Ding, Linxi ”Jim” Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. GR00T N1: An open foundation model for generalist humanoid robots. In *ArXiv Preprint*, March 2025.

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

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*, volume 32, pp. 3942–3951, 2018. doi: 10.1609/aaai.v32i1.11671. URL `https://aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17253`.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763, 2021. URL `https://proceedings.mlr.press/v139/radford21a.html`.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635, 2011.

Vaibhav Saxena, Matthew Bronars, Nadun Ranawaka Arachchige, Kuancheng Wang, Woo Chul Shin, Soroush Nasiriany, Ajay Mandlekar, and Danfei Xu. What matters in learning from large-scale datasets for robot manipulation, 2025. URL https://arxiv.org/abs/2506.13536.

Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, Simon Alibert, Matthieu Cord, Thomas Wolf, and Remi Cadene. Smolvla: A vision-language-action model for affordable and efficient robotics, 2025. URL https://arxiv.org/abs/2506.01844.

Andrew Szot, Alexander Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Singh Chaplot, Oleksandr Maksymets, et al. Habitat 2.0: Training home assistants to rearrange their habitat. *Advances in Neural Information Processing Systems*, 34:251–266, 2021.

Stone Tao, Fanbo Xiang, Arth Shukla, Yuzhe Qin, Xander Hinrichsen, Xiaodi Yuan, Chen Bao, Xinsong Lin, Yulin Liu, Tse kai Chan, Yuan Gao, Xuanlin Li, Tongzhou Mu, Nan Xiao, Arnav Gurha, Viswesh Nagaswamy Rajesh, Yong Woo Choi, Yen-Ru Chen, Zhiao Huang, Roberto Calandra, Rui Chen, Shan Luo, and Hao Su. Maniskill3: Gpu parallelized robotics simulation and rendering for generalizable embodied ai. *Robotics: Science and Systems*, 2025.

Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis Armstrong, Ashwin Balakrishna, Robert Baruch, Maria Bauzá, Michiel Blokzijl, Steven Bohez, Konstantinos Bousmalis, Anthony Brohan, Thomas Buschmann, Arunkumar Byravan, Serkan Cabi, Ken Caluwaerts, Federico Casarini, Oscar Chang, José Enrique Chen, Xi Chen, Hao-Tien Lewis Chiang, Krzysztof Choromanski, Davide D'Ambrosio, Sudeep Dasari, Todor Davchev, Coline Devin, Norman Di Palo, Tianli Ding, Adil Dostmohamed, Danny Driess, Yilun Du, Debidatta Dwibedi, Michael Elabd, Claudio Fantacci, Cody Fong, Erik Frey, Chuyuan Fu, Marissa Giustina, Keerthana Gopalakrishnan, Laura Graesser, Leonard Hasenclever, Nicolas Heess, Brandon Hernaez, Alexander Herzog, R. Hofer, Tsang-Wei Edward Lee, Jacky Liang, Yixin Lin, Sharath Maddineni, Anirudha Majumdar, Assaf Hurwitz Michaely, Robert Moreno, Michael Neunert, Francesco Nori, Carolina Parada, Emilio Parisotto, Peter Pastor, Acorn Pooley, Kanishka Rao, Krista Reymann, Dorsa Sadigh, Stefano Saliceti, Pannag Sanketi, Pierre Sermanet, Dhruv Shah, Mohit Sharma, Kathryn Shea, Charles Shu, Vikas Sindhwani, Sumeet Singh, Radu Soricut, Jost Tobias Springenberg, Rachel Sterneck, Razvan Surdulescu, Jie Tan, Jonathan Tompson, Vincent Vanhoucke, Jake Varley, Grace Vesom, Giulia Vezzani, Oriol Vinyals, Ayzaan Wahid, and Stefan Welker. Gemini robotics: Bringing ai into the physical world. *CoRR*, abs/2503.20020, March 2025a. doi: 10.48550/arXiv.2503.20020. URL https://arxiv.org/abs/2503.20020.

TRI LBM Team, Jose Barreiros, Andrew Beaulieu, Aditya Bhat, Rick Cory, Eric Cousineau, Hongkai Dai, Ching-Hsin Fang, Kunimatsu Hashimoto, Muhammad Zubair Irshad, Masha Itkina, Naveen Kuppuswamy, Kuan-Hui Lee, Katherine Liu, Dale McConachie, Ian McMahon, Haruki Nishimura, Calder Phillips-Grafflin, Charles Richter, Paarth Shah, Krishnan Srinivasan, Blake Wulfe, Chen Xu, Mengchao Zhang, Alex Alspach, Maya Angeles, Kushal Arora, Vitor Campagnolo Guizilini, Alejandro Castro, Dian Chen, Ting-Sheng Chu, Sam Creasey, Sean Curtis, Richard Denitto, Emma Dixon, Eric Dusel, Matthew Ferreira, Aimee Goncalves, Grant Gould, Damrong Guoy, Swati Gupta, Xuchen Han, Kyle Hatch, Brendan Hathaway, Allison Henry, Hillel Hochsztein, Phoebe Horgan, Shun Iwase, Donovon Jackson, Siddharth Karamcheti, Sedrick Keh, Joseph Masterjohn, Jean Mercat, Patrick Miller, Paul Mitiguy, Tony Nguyen, Jeremy Nimmer, Yuki Noguchi, Reko Ong, Aykut Onol, Owen Pfannenstiehl, Richard Poyner, Leticia Priebe Mendes Rocha, Gordon Richardson, Christopher Rodriguez, Derick Seale, Michael Sherman, Mariah Smith-Jones, David Tago, Pavel Tokmakov, Matthew Tran, Basile Van Hoorick, Igor Vasiljevic, Sergey Zakharov, Mark Zolotas, Rares Ambrus, Kerri Fetzer-Borelli, Benjamin Burchfiel, Hadas Kress-Gazit, Siyuan Feng, Stacie Ford, and Russ Tedrake. A careful examination of large behavior models for multitask dexterous manipulation. 2025b. URL https://arxiv.org/abs/2507.05331.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033, 2012.

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. In *Conference on Robot Learning (CoRL)*, 2023.

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. In *Arxiv*, 2023.

Junjie Wen, Yichen Zhu, Jinming Li, Minjie Zhu, Kun Wu, Zhiyuan Xu, Ning Liu, Ran Cheng, Chaomin Shen, Yaxin Peng, Feifei Feng, and Jian Tang. Tinyvla: Towards fast, data-efficient vision-language-action models for robotic manipulation, 2025. URL `https://arxiv.org/abs/2409.12514`.

Sriram Yenamandra, Arun Ramachandran, Karmesh Yadav, Austin S. Wang, Mukul Khanna, Théophile Gervet, Tsung-Yen Yang, Vidhi Jain, Alexander William Clegg, John M. Turner, ZsoltKira, Manolis Savva, Angel X. Chang, Devendra Singh Chaplot, Dhruv Batra, Roozbeh Mottaghi, Yonatan Bisk, and Chris Paxton. Homerobot: Open-vocabulary mobile manipulation. In *Proceedings of the 7th Conference on Robot Learning (CoRL)*, volume 229 of *Proceedings of Machine Learning Research*, pp. 1975–2011. PMLR, Nov 2023. URL `https://proceedings.mlr.press/v229/yenamandra23a.html`.

Gaoyue Zhou, Victoria Dean, Mohan Kumar Srirama, Aravind Rajeswaran, Jyothish Pari, Kyle Hatch, Aryan Jain, Tianhe Yu, Pieter Abbeel, Lerrel Pinto, Chelsea Finn, and Abhinav Gupta. Train offline, test online: A real robot learning benchmark. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9197–9203. IEEE, 2023. doi: 10.1109/ICRA48891.2023.10160594. URL `https://doi.org/10.1109/ICRA48891.2023.10160594`.

Zhiyuan Zhou, Pranav Atreya, You Liang Tan, Karl Pertsch, and Sergey Levine. Autoeval: Autonomous evaluation of generalist robot manipulation policies in the real world. In *Proceedings of the 9th Conference on Robot Learning (CoRL)*, volume 305 of *Proceedings of Machine Learning Research*, pp. 1997–2017. PMLR, Sep 2025. URL `https://proceedings.mlr.press/v305/zhou25a.html`.

Yuke Zhu, Josiah Wong, Ajay Mandlekar, and Roberto Martín-Martín. robosuite: A modular simulation framework and benchmark for robot learning. In *arXiv preprint arXiv:2009.12293*, 2020.

A Use of Large Language Models

We use the aid of large language models to create activity labels and task blueprints, using the process outlined by Nasiriany et al. (2024). We also use large language models for soliciting writing feedback for parts of this manuscript.

B Simulation Assets

B.1 3D objects

We have 57 new object categories: aluminum foil, basket, blender jug, cheese grater, chicken drumstick, cinnamon, colander, cookie dough ball, cream cheese stick, digital scale, dish brush, flour bag, glass cup, honey bottle, hotdog bun, ice cube, ice cube tray, jar, juice, kebab skewer, lemon wedge, lettuce, marshmallow, mayonnaise, measuring cup, mustard, non electric kettle, oil and vinegar bottle, oven tray, pancake, paprika, peeler, pickle slice, pitcher, pizza, pizza cutter, placemat, pot, reamer, salt and pepper shaker, sandwich bread, saucepan, saucepan lid, shrimp, soap dispenser, spray, strainer, straw, sugar cube, syrup bottle, tomato slice, tongs, tupperware, turkey slice, turmeric, whisk, and wooden spoon.

B.2 Interactive fixtures and appliances

We report an inventory of all fixtures and appliances in table 6.

Category	**Unique models**
Blender	22
Coffee machine	48
Dishwasher	25
Electric kettle	25
Fridge	50
Microwave	50
Oven	21
Sink	49
Stand mixer	25
Stove	50
Toaster	44
Toaster oven	47
Total	**456**

Table 6: Inventory of fixtures and appliances

C SCENES

We build 50 kitchen layouts modeled after 50 homes on sale on Zillow.com. These homes span locations in the Bay Area (California), Austin (Texas), Denver (Colorado), Boston (Massachusetts), and Atlanta (Georgia).

D TASKS

D.1 ATOMIC TASKS

We have 65 atomic tasks: `AdjustToasterOvenTemperature`, `AdjustWaterTemperature`, `CheesyBread`, `CloseBlenderLid`, `CloseCabinet`, `CloseDishwasher`, `CloseDrawer`, `CloseElectricKettleLid`, `CloseFridge`, `CloseFridgeDrawer`, `CloseMicrowave`, `CloseOven`, `CloseStandMixerHead`, `CloseToasterOvenDoor`, `CoffeeServeMug`, `CoffeeSetupMug`, `LowerHeat`, `MakeIcedCoffee`, `NavigateKitchen`, `OpenBlenderLid`, `OpenCabinet`, `OpenDishwasher`, `OpenDrawer`, `OpenElectricKettleLid`, `OpenFridge`, `OpenFridgeDrawer`, `OpenMicrowave`, `OpenOven`, `OpenStandMixerHead`, `OpenToasterOvenDoor`, `OrganizeMugsByHandle`, `PnPCabinetToCounter`, `PnPCounterToBlender`, `PnPCounterToCabinet`, `PnPCounterToDrawer`, `PnPCounterToMicrowave`, `PnPCounterToOven`, `PnPCounterToSink`, `PnPCounterToStandMixer`, `PnPCounterToStove`, `PnPCounterToToasterOven`, `PnPDrawerToCounter`, `PnPFridgeDrawerToShelf`, `PnPFridgeShelfToDrawer`, `PnPMicrowaveToCounter`, `PnPSinkToCounter`, `PnPStoveToCounter`, `PnPToasterOvenToCounter`, `PnPToasterToCounter`, `PreheatOven`, `SlideDishwasherRack`, `SlideOvenRack`, `SlideToasterOvenRack`, `StartCoffeeMachine`, `TurnOffMicrowave`, `TurnOffSinkFaucet`, `TurnOffStove`, `TurnOnBlender`, `TurnOnElectricKettle`, `TurnOnMicrowave`, `TurnOnSinkFaucet`, `TurnOnStove`, `TurnOnToaster`, `TurnOnToasterOven`, `TurnSinkSpout`.

D.2 POST-TRAINING TASKS

We provide an overview for the post-training tasks across Tables 7, 8, and 9.

Activity	Task	# Sub-tasks	MoMa req.	Description
Atomic	`CloseBlenderLid`	1	No	Close the lid blender by securely placing the lid on top.
Atomic	`CloseFridge`	1	No	Close the fridge door(s).
Atomic	`CloseToasterOvenDoor`	1	No	Close the toaster oven door.
Atomic	`CoffeeSetupMug`	1	No	Pick the mug from the counter and place it under the coffee machine dispenser.
Atomic	`NavigateKitchen`	1	Yes	Navigate to the [*kitchen location*].
Atomic	`OpenCabinet`	1	No	Open the cabinet door(s).
Atomic	`OpenDrawer`	1	No	Open the [*left/right*] drawer.
Atomic	`OpenStandMixerHead`	1	No	Open the stand mixer head.
Atomic	`PnPCounterToCabinet`	1	No	Pick the item from the counter and place it in the cabinet.
Atomic	`PnPCounterToStove`	1	No	Pick the item from the plate and place it in the pan.
Atomic	`PnPDrawerToCounter`	1	No	Pick the item from the drawer and place it on the counter.
Atomic	`PnPSinkToCounter`	1	No	Pick the item from the sink and place it on the container located on the counter.
Atomic	`PnPToasterToCounter`	1	No	Place the toasted item on a plate.
Atomic	`SlideDishwasherRack`	1	No	Fully slide the top dishwasher rack [*in/out*].
Atomic	`TurnOffStove`	1	No	Turn off the [*burner location*] burner of the stove.
Atomic	`TurnOnElectricKettle`	1	No	Press down the lever to turn on the electric kettle.
Atomic	`TurnOnMicrowave`	1	No	Press the start button on the microwave.
Atomic	`TurnOnSinkFaucet`	1	No	Turn on the sink faucet.

Figure 7: **Post-training Atomic Seen Tasks (18)**

Note: The ”MoMa req.” column indicates whether the task requires Mobile Manipulation or base movement.

Activity	`Task`	# Sub-tasks	MoMa req.	Description
Serving beverages	`DeliverStraw`	4	Yes	Take a straw from the drawer in front and place it inside the glass cup on the dining counter.
Toasting bread	`GetToastedBread`	4	Yes	Start the toaster. Once the lever pops up, take the bread to the plate on the dining counter.
Brewing	`KettleBoiling`	2	Yes	Pick the kettle from the counter and place it on a stove burner. Then turn the burner on.
Loading dishwasher	`LoadDishwasher`	3	No	Pick up the items from the counter, place them in the dishwasher, and close the dishwasher door.
Packing lunches	`PackIdentical Lunches`	15	Yes	Place two identical items of each object in each tupperware on the nearby counter, to pack two identical lunches.
Washing dishes	`PreSoakPan`	3	No	Pick the pan and sponge and place them into the sink. Then turn on the water.
Brewing	`PrepareCoffee`	2	No	Pick the mug from the cabinet, place it under the coffee machine dispenser, and press the start button.
Cleaning sink	`RinseSinkBasin`	2	No	Turn on the sink and manuever the spout to wash all locations of the sink basin.
Sanitizing cutting boards	`ScrubCutting Board`	2	Yes	Pick up the sponge from the counter and clean the cutting board by briefly scrubbing or pressing down on the cutting board. Once finished, release the sponge.
Frying	`SearingMeat`	3	Yes	Grab the pan from the cabinet and place it on the [*burner location*] burner on the stove. Then place the item on the stove and turn the burner on.
Slicing meat	`SetUpCutting Station`	2	Yes	Pick up the knife from the drawer and place it on the cutting board. Then place the meat from the plate to the cutting board.
Organizing dishes and containers	`StackBowls Cabinet`	2	Yes	Pick up the bowls on the counter and stack them on top of one another in the open cabinet. Place the smaller bowl on top of the larger bowl.
Steaming food	`SteamIn Microwave`	6	Yes	Pick the item from the sink and place it in the bowl. Then pick the bowl and place it in the microwave. Then close the microwave door and press the start button.
Sauteing vegetables	`StirVegetables`	4	Yes	Put the items in the pot. Retrieve the spatula and lightly stir the vegetables in the pot.
Storing leftovers	`StoreLeftovers InBowl`	5	Yes	Pick the chicken drumstick and item from their plates and place them in the bowl. Then put the bowl in the fridge.
Making salads	`WashLettuce`	2	No	Wash the lettuce in the sink by running water over it.

Figure 8: **Post-training Composite Seen Tasks (16)**

Activity	Task	# Sub-tasks	MoMa req.	Description
Setting the table	`ArrangeBreadBasket`	5	Yes	Open the cabinet, pick up the item from the cabinet and place it in the basket. Then move the basket to the dining counter.
Brewing	`ArrangeTea`	3	No	Pick the kettle from the counter and place it on the tray. Then pick the mug from the cabinet and place it on the tray. Then close the cabinet doors.
Making toast	`BreadSelection`	2	Yes	From the different types of pastries on the counter, select a croissant and place it on the cutting board. Then retrieve a jar of jam from the cabinet and place it alongside the croissant on the cutting board.
Arranging condiments	`CategorizeCondiments`	2	Yes	Put the shaker and condiment bottle from the counter next to their counterparts in the cabinet.
Chopping vegetables	`CuttingToolSelection`	1	Yes	Place the appropriate cutting tool for cutting the item skin on the cutting board.
Garnishing dishes	`GarnishPancake`	4	Yes	Take the strawberry from the fridge and place it on top of the pancake, located on the dining counter.
Arranging cabinets	`GatherTableware`	4	Yes	Gather all objects into one cabinet and sort the glasses and bowls to opposite sides.
Preparing sandwiches	`HeatKebabSandwich`	6	Yes	Pick up the kebab skewer and baguette bread, and place them inside the toaster oven. Close the toaster oven door and start by setting the timer.
Adding ice to beverages	`MakeIceLemonade`	5	Yes	Grab a lemon wedge from the fridge and one ice cube from the ice bowl, and put them in the glass of lemonade.
Serving food	`PanTransfer`	3	No	Pick up the pan and dump the vegetables in it onto the plate. Then return the pan to the stove.
Portioning meals	`PortionHotDogs`	4	Yes	Place one bun and one sausage from the bowl on each plate.
Organizing recycling	`RecycleBottlesByType`	3	Yes	Move the plastic bottles in the middle to the plastics group, and the glass bottles in the middle to the glass group.
Managing freezer space	`SeparateFreezerRack`	7	Yes	Take the meat container that has the meat item(s) and place it on the second highest rack of the freezer. Then take the vegetable container that has the vegetable(s) and place it on the highest rack of the freezer.
Reheating food	`WaffleReheat`	4	Yes	Open the microwave, place the bowl with waffle inside the microwave, then close the microwave door and turn it on.
Washing produce	`WashFruitColander`	4	No	Put the colander in the sink, put the item in the colander, and turn on the sink faucet and pour water over the colander.
Measuring ingredients	`WeighIngredients`	2	No	Pick the item and place it on the digital scale for weighing, and close the cabinet.

Figure 9: **Post-training Composite Unseen Tasks (16)**

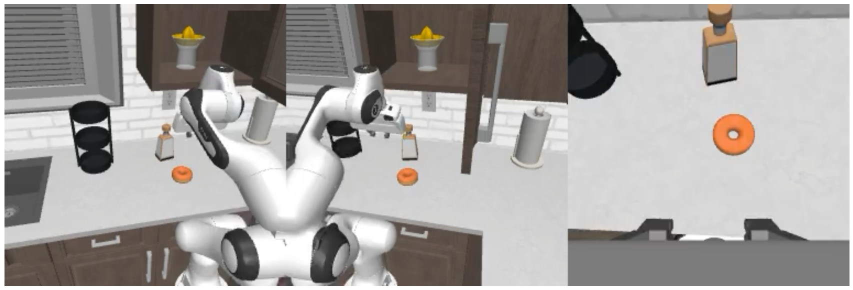


Figure 10: **Camera Images.** Camera images from three views (rendered at 256 × 256 resolution are fed into the model.)

E DATASETS

We present an overview of our datasets in Table 7.

Setting	Num Tasks	Num Scenes	Demos per Task	Dataset Size (hrs)
Pretraining (Human)	300	2500	100	482
Pretraining (MimicGen)	60	2500	10,000	1615
Post-training (Human)	50	10	500	193

Table 7: Dataset statistics across pretraining and post-training settings.

For each demonstration, we store the language instruction, proprioceptive information (robot base pose, robot end effector pose, gripper state information), images from three cameras (wrist camera, left third-person camera, right their-person camera), and the actions.

F POLICY LEARNING

F.1 MODEL ARCHITECTURES AND TRAINING PROTOCOL

Our experiments focus on training vision-based models. The model takes as input a combination of low-level proprioceptive information (base pose, end effector pose, gripper state), task instruction language, and camera images (one wrist camera image, and two third-person camera images). Each of the camera images is at 256 × 256 resolution, and we show examples of the camera views in Figure 10.

We experiment with three models:

Diffusion Policy. Diffusion Policy models trajectory generation as a conditional denoising process in action space, recovering actions from noisy expert trajectories to handle multi-modal robot behaviors. We use an open source diffusion policy codebase and add language conditioning by fusing CLIP-based language embeddings (Radford et al., 2021) with the ResNet visual encoder via FiLM conditioning layers Perez et al. (2018). We use the transformer diffusion variant, with a 12-layer transformer with embedding dimension of 512. We train the model with a batch size of 192 and train for 250k steps for the multi-task learning experiment.

π_0. π_0 is a vision-language-action model which uses PaLI Gemma (Beyer et al., 2024) as the underlying VLM and fuses an action expert to output robot actions via flow matching (Lipman et al., 2023). We use the official open source repository. We use the default full fine-tuning configuration, and we use a batch size of 64 (the highest batch size we can fit on a GH200 GPU). For the multi-task learning experiment, we train the model for 75k steps (48 hours of training time on a GH200 GPU).

GR00T N1.5. GR00T N1.5 is a vision-language-action model which uses a system1-system2 architecture, with the Eagle2 VLM (Li et al., 2025) serving as the high-level encoder (system2) and an action decoder to produce actions via flow matching (system1). We use the official open source repository. For all experiments, we freeze the vision encoder and language encoder (which are the default settings from the open source codebase), and we use a batch size of 128 (the highest batch size we can fit on a GH200 GPU). For the multi-task learning experiments, we train for 120k steps. For the foundation model training experiments, we pretrain for 80k steps and post-train for 60k steps. Finally, for lifelong learning experiments we train stage1 for 100k steps followed by 60k steps for all subsequent stages of training. Generally, we found these settings to be sufficient to allow for model convergence.

F.2 Evaluation Protocol

After training, we evaluate the model at a specified checkpoint on a suite of evaluation tasks. For each evaluation task, we run 30 trials for a specified maximum number of timesteps (the maximum duration is task-dependent). If during this duration the agent achieves the task success condition (binary condition), the episode is counted as a success, otherwise a failure. We report the average success rate across tasks.

G Additional Experiments

G.1 Pretraining Scene Diversity

Our pretraining data spans 2500 kitchen scenes (50 layouts $\times$ 50 styles), and we compare to restricting pretraining data to 25 scenes (5 layouts $\times$ 5 styles), and 5 scenes (5 layouts $\times$ 1 style). To run a fair comparison across these settings, we use MimicGen to generate demonstrations for each setting, generating data across 17 atomic tasks in pretraining kitchens. We run zero-shot evaluations on the 10 fixed post-training kitchen scenes, and also try post-training on the atomic post-training data with 50 demonstrations per task. See Table 8 for results. For zero-shot evaluation, we observe notable performance gains as the number of pretraining scenes increases. These gains also hold in subsequent post-training, highlighting the need for diverse pretraining data.

	Pretraining Data		
	5 Scenes	25 Scenes	2500 Scenes
Zero-shot Evaluation	29.6	39.6	44.7
+ Post-training on Atomic (10%)	53.3	56.7	62.4

Table 8: **Pretraining scene diversity results.** Increasing the composition of scenes in pretraining data improves downstream task performance.

G.2 Robustness Evaluations

In order to examine the generalization capabilities endowed by pretraining on our data, we perform a set of robustness evaluations on the GR00T N1.5 model trained on the full pretraining and post-training mixture. We perturb an aspect of the model's input and evaluate it on our `Composite-Seen` and `Composite-Unseen` tasks. Specifically, we look at the following perturbations:

- **Novel Language**: We prompt an LLM for novel but semantically similar task instructions.
- **Novel Joint Angles**: We sample Gaussian noise and add it to the starting joint angles of the robot.
- **Novel Base Pose**: We sample Gaussian noise and add it to the starting position and yaw of the robot base.
- **Novel Camera Pose**: We sample Gaussian noise and add it to the default third-person and wrist camera poses.

We find that the model is robust to language variations, but can suffer with novel camera poses, joint angles, and base poses.

Table 9: Evaluation of robustness under different perturbations.

Task Split	No Perturbation	Novel Language	Camera Perturbations	Initial Joint Noise	Initial Base Pose Noise
`Composite-Seen`	40.6	38.3	28.8	27.9	31.2
`Composite-Unseen`	42.1	39.2	31.5	32.1	30.2

G.3 JOINT CO-TRAINING OF PRETRAINING AND POST-TRAINING DATA

As an extension to the foundation model training experiments, we examine a separate variant in which we train on all pre-training data and 100% of the post-training data jointly in one single phase. We trained the model for 120k steps and report the resulting task success rates in the post-training kitchens as follows:

- Atomic-Seen: 44.1%
- Composite-Seen: 9.0%
- Composite-Unseen: 11.7%
- Average: 22.5%

Compared to our two-stage learning framework (pretraining first followed by post-training), performance under this co-training regime is substantially lower. This result underscores the importance of a dedicated post-training phase for learning highly performant policies tailored to the post-training tasks.

G.4 LORA FINE-TUNING

For the multi-task learning experiments in section 4.2, we ran GR00T with LoRA fine-tuning, trained for the same number of steps, batch size, etc, as the full fine-tuning variant. The results are as follows:

Table 10: Policy success rates (%) comparing full vs LoRA fine-tuning for GR00T.

	Atomic-Seen	Composite-Seen	Composite-Unseen	Average
Full fine-tuning	43.0	9.6	4.4	20.0
LoRA fine-tuning	2.4	0.2	0.8	1.2

Full fine-tuning is critical to model performance.

H ADDITIONAL ANALYSIS

H.1 FOUNDATION MODEL TRAINING ANALYSIS

We break down the per-task task performance for the best performing variant, pretraining followed by post-training (on 100% of data), in Table 11. Among the Atomic-Seen tasks, the worst performing tasks are TurnOffStove and CloseBlenderLid, which involve high precision and dexterity. However, for Composite-Seen and Composite-Unseen tasks, the worst performing tasks span many diverse characteristics. We run a qualitative analysis, outlining common failure modes for the tasks where the model performs at 30% or less success rate:

- `SteamInMicrowave`: difficulty placing the bowl in the microwave, either placing on edge of microwave or dropping the bowl in the air right before placing in microwave
- `SearingMeat`: typically does not turn on stove burner correctly, or attempts to turn on the incorrect stove burner; or does not place the pan on a valid location on the stovetop
- `PackIdenticalLunches`: unreliable picking from fridge; not moving to the tupperware to place items; placing items in wrong tupperware
- `PrepareCoffee`: often does not place coffee mug correctly under the coffee machine

- `DeliverStraw`: range of failures: difficulty opening drawers, difficulty transporting straw (dropping it), difficulty placing straw into cup
- `GetToastedBread`: often does not press down on lever fully; sometimes presses down lever but then acts randomly
- `PortionHotDogs`: unreliable picks from crowded bowl; unreliable place by placing item on counter instead of plate; placing items on the wrong plate
- `PanTransfer`: generally picks up pan but does not reliably flip the contents of the pan into the plate. this is a dynamic task that's quite unique, does not have much overlap with other tasks in the benchmark
- `HeatKebabSandwich`: often fails to pull out the toaster rack; other times often after placing the first item on the rack inadvertently pushes the rack in by accident and does not place the second item in
- `CategorizeCondiments`: pick and place is not reliable, or does not place matching condiments next to each other
- `SeparateFreezerRack`: often fails to reliably place the tupperware into the freezer, as the freezer is a tight space
- `GatherTableware`: must locate the other mug from the kitchen, and bring it back. navigation ability here is not reliable. also sometimes does not place the mug inside the cabinet , drops it in the air without reaching far into the cabinet.

I SIMULATION INFRASTRUCTURE

I.1 PHYSICS AND RENDERING ENGINE

RoboCasa365 is built on top of RoboSuite (Zhu et al., 2020), which uses the MuJoCo physics engine (Todorov et al., 2012). While MuJoCo's core physics computations are CPU-based, we leverage GPU-based rendering. RoboCasa365 simulates at 20 Hz, with the simulation running approximately in real time—slightly faster or slower depending on scene complexity and hardware specifications. Multiple asynchronous environments can be run in parallel, allowing overall throughput to scale with the number of available CPU cores and GPUs.

I.2 ACTION SPACE

We adopt the underlying controller from RoboCasa (Nasiriany et al., 2024). Specifically, we use an Operational Space Controller (Khatib, 1995) running at 20 Hz that commands the arm through seven action dimensions: three for translation, three for rotation, and one for gripper opening and closing. In addition, we include five action dimensions for mobile base translation and rotation, torso height control, and an action mode that gates mobile base control.

Task	Success Rate (%)	Stages	MoMa Required
Atomic-Seen			
TurnOnElectricKettle	93	1	No
OpenStandMixerHead	90	1	No
CloseToasterOvenDoor	87	1	No
OpenCabinet	87	1	No
SlideDishwasherRack	87	1	No
PnPToasterToCounter	73	1	No
TurnOnMicrowave	70	1	No
OpenDrawer	70	1	No
PnPSinkToCounter	70	1	No
PnPCounterToStove	70	1	No
CloseFridge	67	1	No
TurnOnSinkFaucet	63	1	No
PnPCounterToCabinet	63	1	No
CoffeeSetupMug	60	1	No
NavigateKitchen	60	1	Yes
PnPDrawerToCounter	50	1	No
TurnOffStove	37	1	No
CloseBlenderLid	37	1	No
Composite-Seen			
StackBowlsCabinet	83	2	Yes
PreSoakPan	70	3	No
ScrubCuttingBoard	70	2	Yes
WashLettuce	67	2	No
RinseSinkBasin	60	2	No
KettleBoiling	53	2	Yes
LoadDishwasher	47	3	No
StoreLeftoversInBowl	43	5	Yes
SetUpCuttingStation	33	2	Yes
StirVegetables	33	4	Yes
SteamInMicrowave	30	6	Yes
SearingMeat	27	3	Yes
PackIdenticalLunches	17	15	Yes
PrepareCoffee	13	2	No
DeliverStraw	3	4	Yes
GetToastedBread	0	4	Yes
Composite-Unseen			
RecycleBottlesByType	87	3	Yes
WaffleReheat	83	4	Yes
ArrangeBreadBasket	77	5	Yes
WeighIngredients	67	2	No
BreadSelection	60	2	Yes
CuttingToolSelection	47	2	Yes
GarnishPancake	47	4	Yes
ArrangeTea	43	3	No
WashFruitColander	40	4	No
MakeIceLemonade	40	5	Yes
PortionHotDogs	23	4	Yes
PanTransfer	20	3	No
HeatKebabSandwich	13	6	Yes
CategorizeCondiments	10	2	Yes
SeparateFreezerRack	10	7	Yes
GatherTableware	7	4	Yes

Table 11: **Foundation Model Training Results.**