

# 000 001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 MoMAGEN: GENERATING DEMONSTRATIONS UNDER SOFT AND HARD CONSTRAINTS FOR MULTI-STEP BIMANUAL MOBILE MANIPULATION

Anonymous authors

Paper under double-blind review

## ABSTRACT

Imitation learning from large-scale, diverse human demonstrations has been shown to be effective for training robots, but collecting such data is costly and time-consuming. This challenge intensifies for multi-step bimanual mobile manipulation, where humans must teleoperate both the mobile base and two high-DoF arms. Prior X-Gen (Mandlekar et al., 2023b; Garrett et al., 2024; Jiang et al., 2025b) works have developed automated data generation frameworks for static (bimanual) manipulation tasks, augmenting a few human demos in simulation with novel scene configurations to synthesize large-scale datasets. However, prior works fall short for bimanual mobile manipulation tasks for two major reasons: 1) a mobile base introduces the problem of how to place the robot base to enable downstream manipulation (reachability) and 2) an active camera introduces the problem of how to position the camera to generate data for a visuomotor policy (visibility). To address these challenges, MoMAGEN formulates data generation as a constrained optimization problem that satisfies hard constraints (e.g., reachability) while balancing soft constraints (e.g., visibility while navigation). This formulation generalizes across most existing automated data generation approaches and offers a principled foundation for developing future methods. We evaluate on four multi-step bimanual mobile manipulation tasks and find that MoMAGEN enables the generation of much more diverse datasets than previous methods. As a result of the dataset diversity, we also show that the data generated by MoMAGEN can be used to train successful imitation learning policies using a single source demo. Furthermore, the trained policy can be fine-tuned with a very small amount of real-world data (40 demos) to be successfully deployed on real robotic hardware. More details are on our project page: [momagen-iclr2026.github.io](https://momagen-iclr2026.github.io).

## 1 INTRODUCTION

Learning from human demonstrations is a powerful paradigm for teaching robots complex manipulation skills. A common approach for collecting such data is teleoperation, where a human directly controls the robot to demonstrate desired behaviors. When scaled up, teleoperated data has enabled training visuomotor policies with impressive generalization and success in challenging manipulation tasks (Brohan et al., 2022b; O’Neill et al., 2024; Khazatsky et al., 2024; Black et al., 2024; Intelligence et al., 2025). However, this data collection process remains expensive and time-consuming, especially for tasks that require high-quality demonstrations to ensure effective policy learning.

Recently, collecting a small amount of human teleoperation data and then synthesizing additional data in simulation has become a popular approach to scale up data collection (Mandlekar et al., 2023b; Garrett et al., 2024; Jiang et al., 2025b; Xue et al., 2025; Yang et al., 2025). Compared to offline data augmentation techniques such as those based on image augmentation (Laskin et al., 2020; Yarats et al., 2021; Pitis et al., 2020; Yu et al., 2023; Chen et al., 2023), this approach can autonomously generate new behaviorally diverse data for the same task. This process enlarges the support and convergence region of the policies and reduces the teacher-student distribution mismatch (Ross et al., 2011) through new generated experiences validated in simulation to ensure quality. Notably, the X-Gen family of techniques (Mandlekar et al., 2023b; Garrett et al., 2024; Jiang et al., 2025b; Xue et al., 2025; Yang et al., 2025) leverages simulation and augmentation based on a small number of human demonstrations that are used as seeds to generate multiple new variations automatically. While they have shown success in simple table-top manipulation tasks, a significant standing challenge for X-Gen methods is to extend the benefits of generalizable data generation to real-world tasks with more complex robot embodiments such as mobile manipulators.

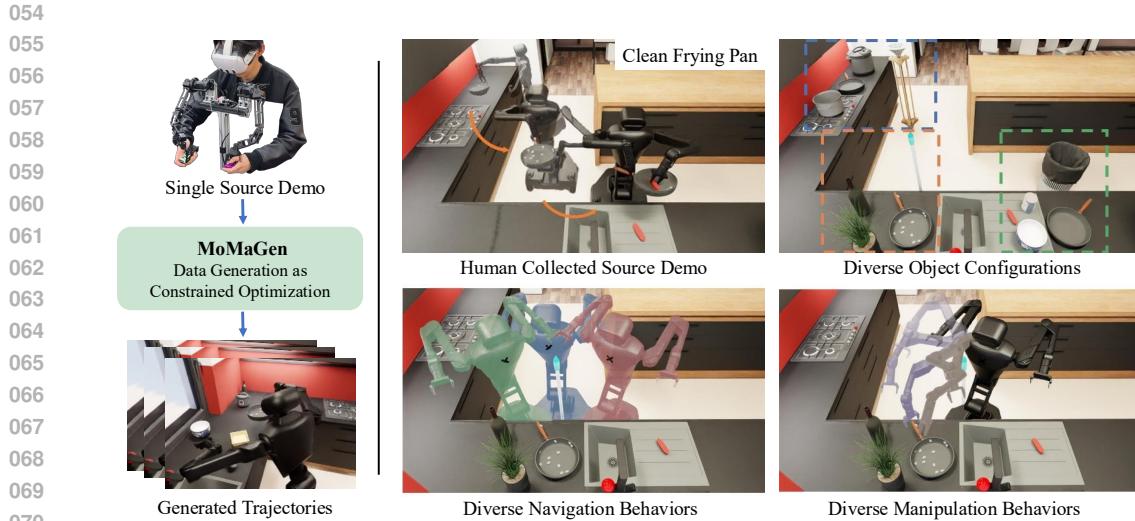


Figure 1: (left) MoMAGEN uses a single human-collected demonstration to generate a large set of demonstrations, formulating data generation as a constrained optimization problem. (top-left) shows a human-collected demo for cleaning a pan with a scrub. (top-right) shows three novel object configurations with aggressive object pose randomization and additional distractors/obstacles. MoMAGEN can generate novel trajectories in these diverse scenarios. (bottom-left) shows three robot base poses and (bottom-right) shows two arm trajectories for picking up the pan.

Solving real-world tasks, such as everyday household activities, often requires a mobile manipulator with whole-body control capabilities to coordinate stable and accurate navigation with end-effector manipulability, often with two arms (Jiang et al., 2025a; Li et al., 2020; Fu et al., 2024b). Teleoperation data collection becomes significantly challenging for high-degrees-of-freedom whole-body control since controlling the base and two arms is a severe overload on the human operators (Dass et al., 2024; Ishiguro et al., 2020; Fu et al., 2024b) (see Fig. 1, left). Augmenting a few (expensive) demonstrations becomes thus critical, but previous methods of the X-Gen family fall short in this domain due to two major reasons: First, mobile manipulation introduces the problem of **object reachability**. For novel object arrangements, naive replay of the navigation segments of the human collected demonstrations easily leads to robot configurations that render subsequent manipulation infeasible. Second, having a mobile base, and thus a movable camera, exacerbates the problems of partial observability. Concretely, when training visuomotor policies for mobile manipulation, a naive augmentation of demonstrations leads to severe problems in **object visibility**: the task-relevant objects may move out of the field of view, making it hard for the policy to make optimal decisions based on the images from onboard sensors. Naive motion planning (Garrett et al., 2024) or replay (Mandlekar et al., 2023a; Jiang et al., 2025b) is insufficient to ensure either reachability or visibility of the task-relevant objects.

To address these challenges, we propose MoMAGEN, a general data generation method for bimanual mobile manipulation. It formulates data generation as a constrained optimization problem with hard constraints (e.g., reachability, visibility during manipulation) and soft constraints (e.g., visibility while navigation). Significantly, we realized that previous methods of the X-Gen family can be interpreted with the same unified framework, except using different (and insufficient) hard and soft constraints for data generation. We evaluate MoMAGEN on four multi-step bimanual mobile manipulation tasks and find that it generates substantially more diverse datasets than prior methods. Leveraging this diversity, we show that the synthetic data can train effective imitation learning policies from just a single demonstration. Moreover, these policies can be fine-tuned with a small amount of real-world data (40 demonstrations) to achieve successful deployment on physical robotic hardware.

## 2 RELATED WORKS

**Data Acquisition for Robot Learning.** Collecting large-scale human teleoperation data for robot learning incurs considerable costs. Scaling up data collection requires a large number of human operators over extended periods of time (Brohan et al., 2022b; O’Neill et al., 2024; Khazatsky et al., 2024). Offline data augmentation techniques can boost data quantity and diversity by perturbing existing trajectories (Mandlekar et al., 2023b; Garrett et al., 2024), or leveraging image augmentation techniques and generative models (Laskin et al., 2020; Yarats et al., 2021; Pitis et al., 2020; Yu

Methods	Bimanual	Mobile	Obstacles	Base Random.	Active Perception	Hard Constraints	Soft Constraints
MimicGen (Mandlekar et al., 2023b)	✗	✓	✗	✗	✗	Succ	N/A
SkillMimicGen (Garrett et al., 2024)	✗	✗	✓	✗	✗	Succ, Kin, C-Free	N/A
DexMimicGen (Jiang et al., 2025b)	✓	✗	✗	✗	✗	Succ, Temp	N/A
DemoGen (Xue et al., 2025)	✓	✗	✓	✗	✗	Kin, C-Free	N/A
PhysicsGen (Yang et al., 2025)	✓	✗	✗	✗	✗	Kin, C-Free, Dyn	Trac
MoMaGen (Ours)	✓	✓	✓	✓	✓	Succ, Kin, C-Free, Temp, Vis	Vis, Ret

Table 1: Comparison of different automated data generation methods and the constraints they enforce. “Succ”: task success; “Kin”: kinematic feasibility; “C-Free”: collision-free execution; “Temp”: temporal constraints for bimanual coordination; “Dyn”: system dynamics; “Trac”: target trajectory tracking; “Vis”: visibility of task-relevant objects in the robot’s camera view; “Ret”: retraction of robot torso and arm to a compact configuration before navigation.

et al., 2023; Chen et al., 2023). However, the augmented data may not always be executable by real robots. A promising alternative is to leverage automated data generation and validation in simulation. Fully automated approaches include trial-and-error (Levine et al., 2018; Pinto & Gupta, 2016; Yu et al., 2016; Dasari et al., 2019) and pre-programmed (e.g., scripted) experts (James et al., 2020; Gu et al., 2023; Jiang et al., 2022; Wang et al., 2023), which are yet to be proven effective for multi-step complex tasks with contact-rich interaction. X-Gen (Mandlekar et al., 2023b; Garrett et al., 2024; Jiang et al., 2025b; Xue et al., 2025; Yang et al., 2025) represents a hybrid approach that uses a handful of human demonstrations as seeds to generate many new variations, augmenting the data by a factor of  $25\times$  to  $350\times$  (Mandlekar et al., 2023b; Jiang et al., 2025b), while ensuring synthesized data is valid in simulation. A comparison between prior X-Gen works and our work can be found in Table 1. Our work signifies an important step toward a more generalizable data generation framework for challenging mobile manipulation tasks, which have never been tackled before.

**Imitation Learning for Mobile Manipulation.** Early successes in robot imitation learning mostly focused on fixed-based arms, but many real-world tasks require a mobile manipulator that can both navigate and manipulate. Such robots need to effectively navigate in an environment to position themselves for downstream manipulation. Collecting teleoperation data for mobile manipulation is significantly more costly: operators must simultaneously control the robot base and arms (Wong et al., 2022; Fu et al., 2024b; Shaw et al., 2024; Dass et al., 2024; Jiang et al., 2025a), calling for automated data generation methods for better scaling. On the algorithmic side, imitation learning methods started handle the complexities of mobile manipulation tasks, employing behavior cloning (Brohan et al., 2022a; Fu et al., 2024b;a; Cheng et al., 2024; Wu et al., 2024a; Yang et al., 2024; Li et al., 2024b; Ze et al., 2024; He et al., 2024) and large pretrained models (Ichter et al., 2022; Wu et al., 2023; Xu et al., 2023; Stone et al., 2023; Jiang et al., 2024b; Shah et al., 2024; Wu et al., 2024b).

### 3 PROBLEM FORMULATION: AUTOMATED DATA GENERATION AS CONSTRAINED OPTIMIZATION

We formulate automated demonstration data generation as a constrained optimization problem, and provide a unified framework that incorporates existing approaches (see Table 1). This optimization problem includes both hard and soft constraints. The former must be strictly satisfied (e.g., task success, convergence to target end-effector poses at key frames, and collision avoidance). The latter capture desirable properties (e.g., shorter trajectory and reduced jerkiness). Generating valid data requires strictly satisfying hard constraints while minimizing the costs associated with soft constraints.

Each task is modeled as a Markov Decision Process (MDP) with state space  $\mathcal{S}$  and action space  $\mathcal{A}$ . Given a set of source demonstrations  $\mathcal{D}_{src} = \{d^j = (s_0^j, a_0^j, \dots, s_{T_{src}}^j)\}_{j=0}^{N_{src}}$ , where  $N_{src}$  is the number of source demonstrations,  $T_{src}$  is the trajectory length, and  $s_0 \sim D$  is the initial state from distribution  $D$ . We aim to generate a new set of successful demonstrations  $\mathcal{D} = \{d\}_{j=0}^{N_{gen}}$  given the source demonstrations  $\mathcal{D}_{src}$  and a set of constraints  $\{\mathcal{G}_i\}$ . With the generated demonstrations, we can train Behavioral Cloning (Pomerleau, 1988) policies  $\pi_\theta$  using  $\arg \min_\theta \mathbb{E}_{(s,a) \sim \mathcal{D}} [-\log \pi_\theta(a|s)]$ .

Following prior work (Mandlekar et al., 2023b; Garrett et al., 2024; Jiang et al., 2025b), each source demonstration  $d$  can be decomposed to  $N$  subtasks. Each subtask contains an object trajectory  $S_i(o_i)$ ,  $i \in [N]$ , where  $o_i$  is the object of interest (i.e. target object), and an end-effector trajectory  $\tau_i = \{\mathbf{T}_W^{E_k}\}_{k=0}^{K_i}$ ,  $i \in [N]$ ,  $k \in [K_i]$ , where  $\mathbf{T}_W^{E_k}$  is the pose of the end-effector frame  $E$  with respect to the world reference frame  $W$  at time  $k$ , and  $K_i$  is the number of steps for the subtask  $i$ . Each subtask can further be labeled as either a *free-space* subtask or a *contact-rich* subtask. In a free-

space subtask, the goal is to move the robot base or arms in free space (e.g. to a pregrasp pose), where feasible trajectories can be sampled using motion planning subject to kinematic and collision constraints. In a contact-rich subtask, the goal is to manipulate the relevant objects through contacts (e.g. grasping, wiping). The relative poses between the end-effector frame and the target object frame for contact-rich subtasks are preserved from the source to the generated demonstrations.

Generating a demonstration can be viewed as solving the following constrained optimization problem:

$$\arg \min_{a_t \in [T]} \mathcal{L}(\cdot) \quad \text{s.t.} \quad \begin{cases} s_{t+1} = f(s_t, a_t), & \forall t \in [T] \\ \mathcal{G}_{\text{kin}}(s_t, a_t) \leq 0, & \forall t \in [T] \\ \mathcal{G}_{\text{coll}}(s_t, a_t) \geq 0, & \forall t \in [T] \\ \mathcal{G}_{\text{vis}}(s_t, a_t, o_{i(t)}) \leq 0, & \forall t \in [T] \\ \mathbf{T}_W^{E_k} = \mathbf{T}_W^{o_i} (\mathbf{T}_W^{o_{i,src}})^{-1} \mathbf{T}_W^{E_k}, & \forall \text{contact } \tau_i, \forall k \in [K_i] \\ s_t \in D_{\text{success}} & \exists t \in [T] \text{ (task success)} \end{cases} \quad (1)$$

Here,  $\mathcal{L}(\cdot)$  contains user-specified soft constraints. The function  $f(s_t, a_t)$  denotes the system dynamics. The constraints  $\mathcal{G}_{\text{kin}}$  encode kinematic feasibility (e.g. joint limits),  $\mathcal{G}_{\text{coll}}$  encode collision avoidance, and  $\mathcal{G}_{\text{vis}}$  encode visibility constraints (e.g. during manipulation).

## 4 MOMAGEN

Following the proposed problem formulation in Section 3, we develop MOMAGEN that solves a constrained optimization problem to generate demonstrations for bimanual mobile manipulation tasks. We first introduce the reachability and visibility constraints that are essential for bimanual mobile manipulation in Section 4.1. We then detail the data generation method in Section 4.2.

### 4.1 CONSTRAINTS FOR BIMANUAL MOBILE MANIPULATION

In our instantiation of MOMAGEN, beyond the commonly used constraints discussed earlier, we introduce several key technical innovations that are crucial for generating high-quality bimanual mobile manipulation demonstrations, regarding reachability, visibility and retraction.

**Reachability as Hard Constraint.** A key distinction of mobile manipulation is the need to control the robot base for effective downstream manipulation. Prior works (Mandlekar et al., 2023a; Jiang et al., 2025b) reuse base trajectories from source demos without adaptation, which fails when randomized target objects lie outside the arm’s workspace. To address this, we impose reachability as a hard constraint, ensuring sampled base poses keep all required end-effector trajectories within reach.

**Object Visibility during Manipulation as Hard Constraint.** A valid base pose must also satisfy a hard visibility constraint: task-relevant objects must remain in view. Since the generated data is meant to train visuomotor policies, we ensure each sampled pose allows the head camera to observe these objects without occlusion, using the additional camera or torso articulation when needed.

**Object Visibility during Navigation as Soft Constraint.** Maintaining task-relevant object visibility during navigation is desirable but not required, so we treat it as a soft constraint by adding a visibility cost that biases the head camera toward the target object during navigation.

**Retraction as Soft Constraint.** After manipulation, the robot retracts its torso and arms into a compact configuration, reducing its footprint and making subsequent navigation safer.

As illustrated above, the choice of constraints, particularly soft constraints, is highly dependent on the specific application or domain. In our case, we selected the aforementioned constraints because we believe they promote the generation of high-quality bimanual manipulation demonstrations that closely approximate human-level optimality for visuomotor policy training.

### 4.2 AUTOMATED DEMONSTRATION GENERATION FOR BIMANUAL MOBILE MANIPULATION

In this section, we discuss in detail our framework that leverages the novel constraints introduced above to efficiently generate diverse demonstration data for bimanual mobile manipulation.

**Source Demonstration Annotation.** Each demonstration is segmented into temporally ordered subtasks, that are either single-arm motions or bimanual motions requiring synchronization at subtask boundaries. For each subtask, we annotate the target object  $o_{target}$ , gripper-held object  $o_{held}$ , timestep immediately preceding contact  $t_{pregrasp}$ , end timestep  $t_{end}$ , and retraction type  $r$  to execute after manipulation. Figure 2 illustrates an annotated subtask of grasping a cup with one arm. To stress-test our method and minimize human effort, we collect and annotate only a single source demo ( $N_{src}=1$ ).

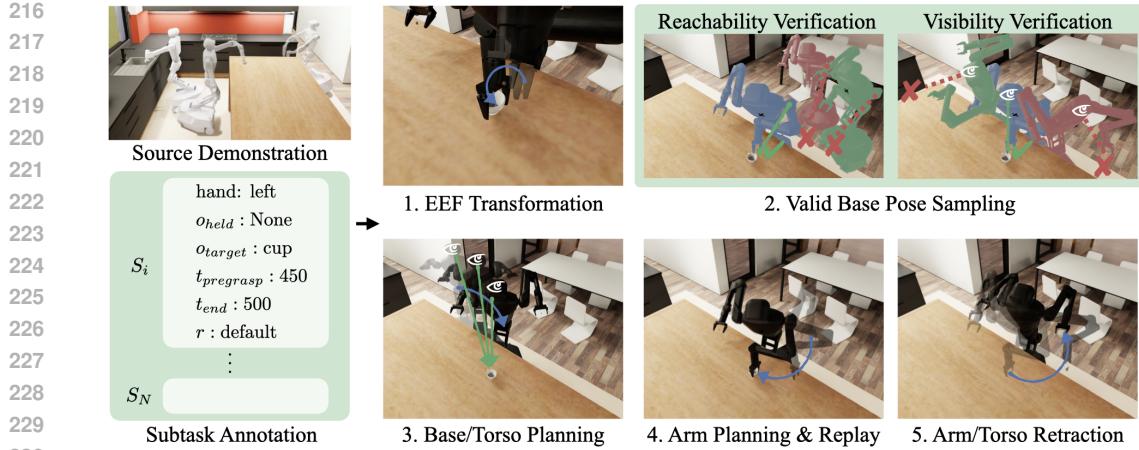


Figure 2: MOMAGEN method. Given a single source demonstration, as well as annotations for object-centric subtasks for each end-effector, MOMAGEN first randomizes scene configuration, and transforms the end-effector poses from the source demo to the new objects’ frame of reference. For each subtask, it tries to sample a valid base pose that satisfies reachability and visibility constraints. Once found, it plans a base and torso trajectory to reach the desired base and head camera pose while trying to look at the target object during navigation. Once arrived, it plans an arm trajectory to the pregrasp pose and uses task space control for replay, before retracting back to a tucked, neutral pose.

---

**Algorithm 1** MOMAGEN

---

**Input:** original demo, new initial state  $s_0$

**Output:** generated demo

```

241: 1: for each segment do
242: 2:   Get current  $\mathbf{T}^{\text{base}}$ ,  $\mathbf{T}^{\text{cam}}$ ,  $q^{\text{torso}}$ ,  $q^{\text{arm}}$ 
243: 3:   if held object not in hand then abort
244: 4:   Compute transformed end-effector pose  $\mathbf{T}^{\text{eef}}$  using new target object pose
245: 5:   Check visibility of target object with  $\mathbf{T}^{\text{cam}}$ 
246: 6:   Solve IK for arm trajectory  $\{q_t^{\text{arm}}\}$  with current  $\mathbf{T}^{\text{base}}$ ,  $\mathbf{T}^{\text{cam}}$ 
247: 7:   while not visible or no IK exists do
248: 8:     Sample new base pose  $\mathbf{T}^{\text{base}}$ 
249: 9:     Sample new camera pose  $\mathbf{T}^{\text{cam}}$ 
250: 10:    Solve IK for arm  $\{q_t^{\text{arm}}\}$  and torso  $\{q_t^{\text{torso}}\}$  with sampled  $\mathbf{T}^{\text{base}}$ ,  $\mathbf{T}^{\text{cam}}$ 
251: 11:    Plan motion for  $\{q_t^{\text{torso}}\}$  from current  $\mathbf{T}^{\text{base}}$  to sampled  $\mathbf{T}^{\text{base}}$ ,  $\mathbf{T}^{\text{cam}}$  w/ soft visibility
252: 12:    Plan motion for  $\{q_t^{\text{arm}}\}$  from previous  $\mathbf{T}^{\text{eef}}$  to pregrasp  $\mathbf{T}^{\text{eef}}$ 
253: 13:    Control end-effector in task space to follow transformed  $\mathbf{T}^{\text{eef}}$ 
254: 14:    Attempt retraction

```

---

**Demonstration Generation.** MOMAGEN generates new demonstrations for novel initial states by following Algorithm 1. For each subtask, we first verify that the robot holds the required object, aborting early if not (line 3), typically due to a failed grasp. We then compute end-effector poses for contact-rich motions by applying the transforms from the original demo (line 4). Next, we check reachability and visibility constraints for the current base and head camera configuration (lines 5-6); if satisfied, the robot proceeds to manipulation (lines 12-13). Otherwise, we sample base and head camera configurations until a valid one is found (lines 7-11), using heuristics and inverse kinematics to ensure reachability and visibility (lines 8-10). Once a valid one is reached via motion planning with soft visibility constraint (line 11), the robot executes the manipulation, moving to the pregrasp pose via motion planning and replaying the contact-rich segment in task space (lines 12-13). Finally, the robot tries to retract to a canonical or prior joint configuration (line 14). This process repeats for each subtask, with motion planning and IK handled by cuRobo (Sundaralingam et al., 2023).

**Key Novelties.** Our approach introduces four main innovations for generating mobile manipulation demonstrations. **(1) Full-body Motion:** Unlike prior work focused only on end-effector pose  $\mathbf{T}^{\text{eef}}$ , we jointly consider  $\mathbf{T}^{\text{eef}}$ , head camera pose  $\mathbf{T}^{\text{cam}}$ , and base pose  $\mathbf{T}^{\text{base}}$ . **(2) Visibility Guarantee:** We enforce object visibility before manipulation (lines 5, 9) and add a soft constraint to maintain



Figure 3: Task visualization. Our multi-step tasks include long-range navigation, sequential and coordinated bimanual manipulation, requiring pick-and-place and contact-rich motion.

visibility during navigation (line 11). **(3) Expanded Workspace:** We sample base poses near target objects (line 8) and plan base motions across the room to fully exploit mobility (line 11). **(4) Efficient Generation:** We accelerate generation by prioritizing fast IK checks over full motion planning whenever possible for preemptive filtering, and by decomposing robot’s configuration into torso/arm subspaces for efficient conditional sampling, akin to task–motion planning (Garrett et al., 2021).

## 5 EXPERIMENTS AND RESULTS

We evaluate MoMAGEN on four household tasks (Section 5.1) under three object/scene randomization schemes (Section 5.2). Compared with two baselines (Jiang et al., 2025b; Garrett et al., 2024), MoMAGEN produces much more diverse data, higher generation success rates, and substantially greater object visibility, which is crucial for visuomotor policies (Section 5.3). We further assess how these demonstrations help train imitation learning policies, with sim-to-real potentiality (Section 5.4).

### 5.1 TASK SETUP

We evaluate MoMAGEN on four household tasks that require mobile manipulation and involve various manipulation skills, including pick-and-place, contact-rich interactions and bimanual coordination, illustrated in Figure 3. Inspired by BEHAVIOR-1K, all tasks are implemented in OmniGibson (Li et al., 2023; 2024a). **Pick Cup:** navigate to a table and lift a cup. **Tidy Table:** move a cup from the countertop to the sink, requiring long-range mobile manipulation. **Put Dishes Away:** stack two plates from the countertop onto a shelf using two arms independently, requiring bimanual uncoordinated motion. **Clean Frying Pan:** use both arms to scrub a dusty pan with a brush, requiring contact-rich bimanual coordinated motion. For each task, we collect a single 1–3 minute human teleoperated demonstration, with base motion accounting for 45% of the duration.

### 5.2 DOMAIN RANDOMIZATION SCHEMES

For each task, we define three domain randomization levels of increasing difficulty. **D0:** task-relevant objects randomized within  $\pm 15$  cm and  $\pm 15^\circ$  on the same furniture. **D1:** task-relevant objects placed anywhere on the furniture with unrestricted orientation (Figure 4a). **D2:** D1 plus additional objects on furniture (manipulation obstacles) and on the floor (navigation obstacles) (Figure 1, upper right). This scheme is significantly more aggressive than prior work (Mandlekar et al., 2023b; Garrett et al., 2024; Jiang et al., 2025b), enabled by MoMAGEN’s unique ability to generate novel base motions. More details and visualizations can be found in Sec. B.3 and Fig. 12 in the Appendix.

### 5.3 DATA GENERATION COMPARISON

We evaluate MoMAGEN for bimanual mobile manipulation under varying randomization schemes, comparing it to two baselines: SkillMimicGen (Garrett et al., 2024), which generates single-arm trajectories via motion planning and task-space control, and DexMimicGen (Jiang et al., 2025b), which targets dexterous bimanual data. Since all the evaluated tasks require substantial base movement, both baselines are extended with base trajectory replay from source demos, similar to MimicGen (Mandlekar et al., 2023a). We assess methods using three metrics: (1) data diversity (object pose and action variation), (2) generation success rate, and (3) object visibility ratio during navigation.

**How diverse are the demonstrations generated by MoMAGEN?** Figure 4a shows the variation in task-relevant object poses for the Tidy Table task under MoMAGEN (D0/D1) and SkillMimicGen (D0). The baselines succeed only in D0, as they cannot generate novel base motions. In contrast, MoMAGEN (D1) covers a much wider range of object poses, spanning the entire counter rather than a small corner. Figures 4b–c further demonstrate that MoMAGEN D1 yields substantially broader coverage of base and end-effector poses compared to D0 and the baselines. Figure 4d shows PCA projections of arm and torso joints under D0. Despite identical object pose distributions, randomizing feasible base placements allows MoMAGEN to achieve far greater action diversity. Moreover, with its motion planner, MoMAGEN can generate demonstrations in cluttered scenes with distractors or

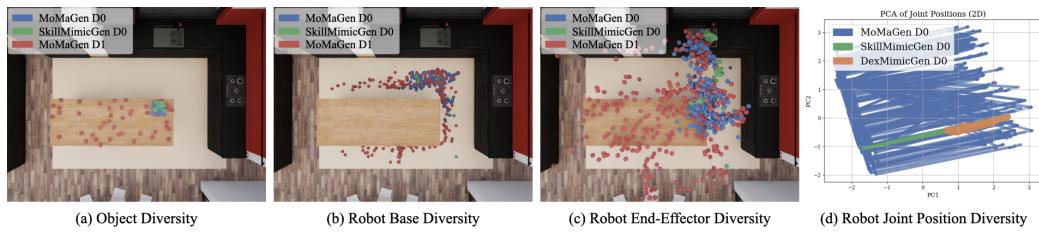


Figure 4: Generated data diversity analysis for Tidy Table task (50 trajectories, subsampled). Given the same object randomization (D0) (a), compared to SkillMimicGen, MoMAGEN samples diverse base poses (b), and as a result, diverse end-effector poses (c) and joint positions (d). MoMAGEN is also the only method that can generate data for D1 randomization (red) for even greater diversity.

	Methods	Pick Cup	Tidy Table	Put Dishes Away	Clean Frying Pan
D0	MoMAGEN	0.86	<b>0.80</b>	0.38	<b>0.51</b>
	SkillMimicGen	<b>1.00</b>	0.69	0.38	0.40
	DexMimicGen	<b>1.00</b>	0.72	0.38	0.35
	MoMAGEN w/o soft vis. const.	0.88	0.78	<b>0.50</b>	0.46
	MoMAGEN w/o hard vis. const.	0.97	0.59	0.29	0.24
	MoMAGEN w/o vis. const.	0.97	0.74	0.29	0.36
D1	MoMAGEN	0.60	<b>0.64</b>	<b>0.34</b>	<b>0.20</b>
	MoMAGEN w/o vis. const.	<b>0.66</b>	0.48	0.23	0.13
D2	MoMAGEN	0.47	<b>0.22</b>	<b>0.07</b>	<b>0.16</b>
	MoMAGEN w/o vis. const.	<b>0.50</b>	0.16	0.05	0.12

Table 2: Data generation success rates comparison. For simpler tasks (Pick Cup), ablations and baselines achieve higher data gen success rates because of fewer constraints and less motion planning stochasticity. However, for more complex tasks (the other three), enforcing hard visibility constraints helps position the robot torso to a suitable configuration that facilitates downstream manipulation, leading to higher success rates. The baselines suffer from zero success rates and hence are omitted for D1/D2 because the objects are beyond the reachability of replayed base poses from the source demo.

obstacles (Figure 12), further enhancing the diversity of both robot actions and visual observations. Additional results for other tasks are shown in Figures 13, 14, and 15 in the Appendix.

**Can MoMAGEN achieve high-throughput data generation?** Table 2 shows that MoMAGEN achieves a 63% average data generation success rate for D0 and can generate data for all tasks across all randomization levels, though throughput decreases with higher difficulty. In contrast, the baselines perform well on simple tasks like Pick Cup but fail on harder ones such as Clean Frying Pan, where adapting base motions is critical, and cannot handle D1 or D2 randomization at all.

**Can MoMAGEN generate demonstrations with high object visibility?** Because task-relevant object visibility is critical for visuomotor policy learning (Sec. 5.4), we evaluate whether MoMAGEN indeed produces data with high object visibility. We compare it against baselines and three ablations that remove hard and/or soft visibility constraints. As shown in Table 3, MoMAGEN achieves substantially higher task-relevant object visibility (often double) relative to both baselines and ablations, especially for multi-step tasks. In the Tidy Table task (Figure 5), results further show that both hard and soft constraints are essential for maintaining high visibility during navigation.

**Can MoMAGEN generate demos with a new robot embodiment?** We evaluate cross-embodiment data generation by using a single source demo from a Galexea R1 robot to generate Pick Cup demos on a TIAGO robot. MoMAGEN successfully transfers across platforms by planning/replaying dense end-effector trajectories in the task space, which is largely agnostic to robot-specific kinematics, demonstrating robustness and flexibility of MoMAGEN. More discussions are found in Appendix B.2.

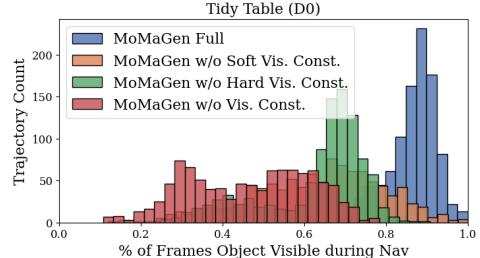


Figure 5: Object visibility analysis for MoMAGEN and ablations. The x-axis is the % of frames where the target object is visible during navigation, and the y-axis is the trajectory count (out of 1000). MoMAGEN significantly outperforms ablations thanks to both hard and soft visibility constraints.

	Methods	Pick Cup	Tidy Table	Put Dishes Away	Clean Frying Pan
D0	MoMAGEN	<b>1.00</b>	<b>0.86</b>	<b>0.79</b>	<b>0.69</b>
	SkillMimicGen	<b>1.00</b>	0.40	0.71	0.65
	DexMimicGen	<b>1.00</b>	0.39	0.71	0.67
	MoMAGEN w/o soft vis. const.	<b>1.00</b>	0.63	0.62	0.56
	MoMAGEN w/o hard vis. const.	0.98	0.63	0.68	0.55
	MoMAGEN w/o vis. const.	0.90	0.46	0.40	0.35
D1	MoMAGEN	<b>0.93</b>	<b>0.89</b>	<b>0.78</b>	<b>0.80</b>
	MoMAGEN w/o vis. const.	0.71	0.46	0.40	0.43
D2	MoMAGEN	<b>0.94</b>	<b>0.79</b>	<b>0.75</b>	<b>0.81</b>
	MoMAGEN w/o vis. const.	0.73	0.48	0.40	0.44

Table 3: Task-relevant object visibility comparison. Our hard and soft visibility constraints are exceedingly effective in keeping the object in view during navigation, achieving over 75% visibility even for aggressively randomized object pose (D1) and obstacles/occluders (D2). We omit baselines for D1/D2 due to zero data generation success rates.

#### 5.4 POLICY LEARNING WITH GENERATED DEMONSTRATIONS

While MoMAGEN can synthesize successful trajectories for the benchmark tasks, it relies on privileged information such as ground-truth object poses and geometry. To enable real-world deployment, visuomotor policies must still be trained from onboard sensory inputs (e.g., RGB images). In this section, we investigate: (1) whether MoMAGEN generated demonstrations improve imitation learning performance over other data generation methods; (2) whether these benefits hold across different imitation learning algorithms; (3) how task-relevant object visibility influences policy training; (4) how performance scales with the number of generated demonstrations; and (5) whether a model trained on MoMAGEN data can be successfully deployed on real hardware.

**Policy Learning Setup.** We experiment with two imitation learning based methods, WB-VIMA (Jiang et al., 2025a) and  $\pi_0$  (Black et al., 2024). Both methods take as input proprioceptive info and RGB images from the head camera and two wrist-mounted cameras, and outputs the target robot joint position. For WB-VIMA, we further fuse and post-process the three RGB images (with groundtruth depth from the simulation) into egocentric colored point cloud, before feeding into the policy network. For WB-VIMA, we train individual single-task policies from scratch, whereas for  $\pi_0$ , we finetune a pre-trained  $\pi_0$  model with a LoRA rank of 32. More implementation details are in Appendix C.

**How do different data generation methods impact policy performance?** We generate 1000 demonstrations with MoMAGEN for Pick Cup (D0/D1), and Tidy Table (D0), and compare against SkillMimicGen and DexMimicGen augmented with base trajectory replay. As shown in Figure 6a, MoMAGEN matches the baselines on Pick Cup (D0), where the small randomization range (0.3m  $\times$  0.3m) makes replayed navigation sufficient. On Tidy Table (D0), however, MoMAGEN significantly outperforms the baselines, which overfit to the long, nonsmooth replayed trajectories. For the more challenging Pick Cup (D1) task (1.3m  $\times$  0.8m randomization), only MoMAGEN enables WB-VIMA to achieve a 0.25 success rate, while baselines trained on D0 data fail completely (Figure 6b). The diverse base motions in MoMAGEN data also improve intermediate success like touching the cup.

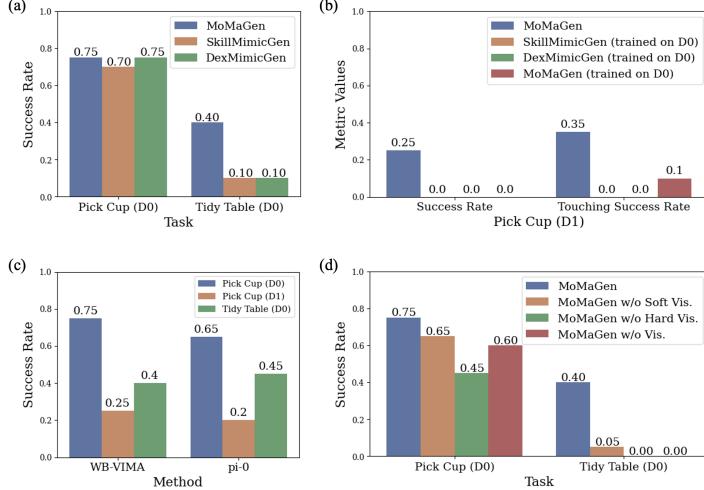


Figure 6: Comparison between MoMAGEN and other data generation methods on WB-VIMA’s performances in (a) and (b), performances of WB-VIMA and  $\pi_0$  trained with MoMAGEN data in (c) and visibility ablations in (d). The success rate is averaged over 20 unseen evaluation episodes. Policies trained on MoMAGEN data consistently perform better than those trained on others’ data.

432 **Does MOMAGEN generated data benefit different imitation learning methods?** We fine-tune  
 433  $\pi_0$  on MOMAGEN data for Pick Cup (D0/D1) and Tidy Table (D0) using 1000 demonstrations. As  
 434 shown in Figure 6c, the fine-tuned  $\pi_0$  achieves success rates comparable to WB-VIMA across all three  
 435 tasks. These results indicate that MOMAGEN data can effectively improve the performance of diverse  
 436 imitation learning approaches (either trained from scratch, or pretrained and LoRA fine-tuned).

437 **How does object visibility ratio affect policy performance?** In MOMAGEN, the head camera  
 438 is constrained to focus on the task-relevant object, aiding visual servoing during navigation and  
 439 improving visibility during manipulation (Sec. 5.3). To assess the impact of these constraints on  
 440 imitation learning, we conduct an ablation study with WB-VIMA on Pick Cup (D0) and Tidy Table  
 441 (D0), comparing the full method with three variants: (1) without soft visibility (camera not encouraged  
 442 to track the object during navigation); (2) without hard visibility (camera not enforced to observe  
 443 the object during manipulation); and (3) without any visibility constraints. As shown in Figure 6d,  
 444 ablations on Pick Cup (D0) achieve success rates of 0.45 to 0.65, below the 0.75 of MOMAGEN. The  
 445 gap is even larger on Tidy Table (D0), where ablations peak at 0.05 while MOMAGEN reaches 0.40.  
 446 These results demonstrate that enforcing visibility constraints during data generation substantially  
 447 improves policy performance, particularly when policies rely on short history inputs.

448 **How is the model performance scaled with the number  
 449 of MOMAGEN generated demonstrations?** We extend the  
 450  $\pi_0$  experiments to examine the effect of scaling MOMAGEN-  
 451 generated data. Models were fine-tuned with 500, 1000, and  
 452 2000 demonstrations for 50K steps and evaluated across  
 453 tasks and randomization levels (Figure 7). Results show  
 454 promising scaling trends, particularly under D1 randomiza-  
 455 tion, where larger datasets improve coverage of both state  
 456 and action spaces. These findings indicate that increasing  
 457 synthetic data yields meaningful performance gains.

458 **Can policies trained with MOMAGEN’ generated data  
 459 be deployed in the real world?** To evaluate the real-world  
 460 utility of MOMAGEN, we tested the Pick Cup task using the same robot platform and a setup similar  
 461 to our simulation (white table, green cup). Because zero-shot sim-to-real transfer is challenging,  
 462 we collected 40 real-world demos and assessed the benefit of pretraining on 1,000 synthetic demos  
 463 generated by MOMAGEN. For WB-VIMA, pretraining on synthetic data followed by fine-tuning  
 464 on real data achieved a 10% success rate, while training on real data alone yielded 0%. Although  
 465 the overall success was low, the pretrained model consistently exhibited meaningful behavior (e.g.,  
 466 reaching the cup), whereas the baseline failed to progress. For  $\pi_0$ , the effect was stronger: the  
 467 pretrained model achieved 60% success compared to 0% for the baseline. In this case, the baseline  
 468 attempted grasps but failed due to poor precision, with the cup slipping from the gripper, underscoring  
 469 the difficulty of learning robust policies from limited real data alone, even when starting from a strong  
 470 pretrained foundation (i.e.  $\pi_0$  pretrained weights). Despite the sim-to-real gap, these results show that  
 471 diverse synthetic data provide a valuable prior, enabling efficient policy learning in low-data regimes.  
 472 This highlights the practical utility of MOMAGEN for real-world deployment (see Appendix A.1).

## 472 5.5 MOMAGEN FAILURE AND COMPUTATIONAL COST ANALYSIS

473 **Failure Analysis.** We analyze the causes of unsuccessful data  
 474 generation episodes by categorizing failures into base sampling,  
 475 base-level planning, arm-level planning, and simulation instabilities. *Base Sampling* failures occur when no base pose is found that  
 476 satisfies reachability and visibility constraints. *Base MP IK* and  
 477 *Base MP TrajOpt* correspond to inverse-kinematics and trajectory-  
 478 optimization failures during base motion planning. Similarly,  
 479 *Arm MP IK* and *Arm MP TrajOpt* capture IK and trajectory-  
 480 optimization failures during arm motion planning, while *Arm MP*  
 481 *Other* includes additional arm-planning issues such as invalid  
 482 start or goal states. *Simulation Instabilities* arise from controller  
 483 inaccuracies and stochastic effects in the simulator. Fig. 8 shows  
 484 that across all three domain randomizations, simulation instabilities  
 485 account for a substantial share of failures (35%), which can

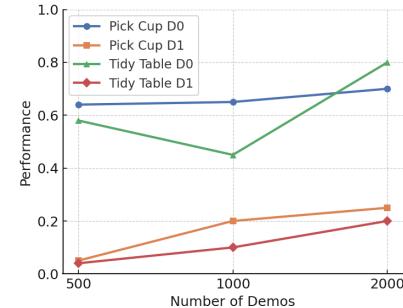
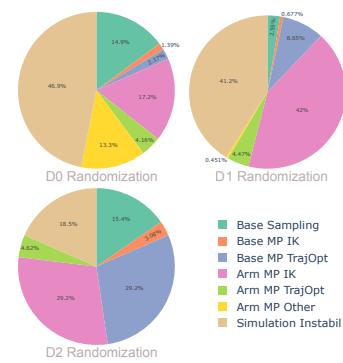
Figure 7: Data Scaling of  $\pi_0$ 

Figure 8: Failure Analysis

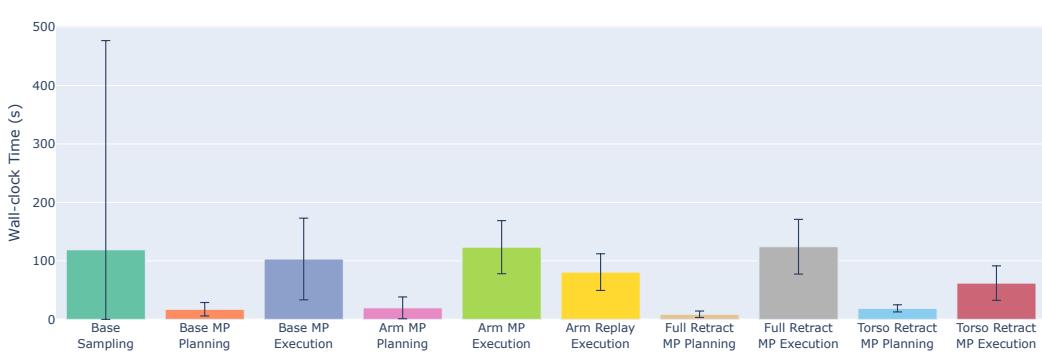


Figure 9: Compute costs (wall-clock time) for MoMAGEN data generation components.

be mitigated by improvements in controller robustness and simulation fidelity. Arm-level motion planning contributes the largest proportion of planner-related failures (40% on average), exceeding base-level planning failures (26% on average). Notably, in D2 randomization, navigation-related failures (base sampling, base IK, and base TrajOpt) increase significantly due to floor obstacles in an already tight navigation space. We also analyze the distribution of failures across task-level steps within each multi-step task (Fig. 16 in Appendix).

**Compute Cost Analysis.** To better understand the efficiency of our data generation pipeline, we analyze the computational cost contribution of each component. Fig. 9 shows the wall-clock time distributions for base sampling, motion planning, and simulation execution. Simulation execution dominates total computation time, greatly exceeding the corresponding planning durations. For instance, base motion planning averages 18 seconds, whereas executing the resulting motion in simulation takes 100 seconds. Base sampling also exhibits high variance due to our current strategy of random sampling within a ring-shaped region around the target: when only a few poses are feasible, the search becomes slow, whereas abundant feasible poses lead to quick success. Future work may reduce this variance through more informed sampling methods—for example, biasing toward regions with greater free space. Fig. 10 reports the average compute cost of each component as a percentage of the total episode duration, computed only over successful episodes to ensure a fair comparison across all components. Notably, base sampling becomes increasingly expensive from D0 to D2—reflecting the rise in scene complexity, where fewer feasible base poses exist and longer searches are required.

## 6 CONCLUSIONS AND LIMITATIONS

In this work, we present MoMAGEN, a general data generation method for multi-step bimanual mobile manipulation using a single human-collected demonstration. MoMAGEN formulates data generation as a constrained optimization problem that satisfies hard constraints while balancing soft constraints. We propose key novelties that involve reachability and visibility constraints, and evaluate our method on four challenging bimanual mobile manipulation tasks. We showcase superior diversity and task-relevant object visibility of MoMAGEN-generated data compared to those generated by baselines and ablations, which further translates to better policy learning results.

**Limitations.** We currently assume access to full scene knowledge during demonstration generation. While this is straightforward in simulation, it poses challenges in real-world scenarios. A possible solution is to incorporate vision models such as SAM2 to estimate object poses relative to the robot. Additionally, we only show data generation results with alternating phases of navigation and manipulation, although our framework is easily extensible to whole-body manipulation (e.g. opening doors) and we leave it for future work. Lastly, our approach depends on sizable GPU resources to run GPU-accelerated motion generators, which can be computationally intensive during data generation.

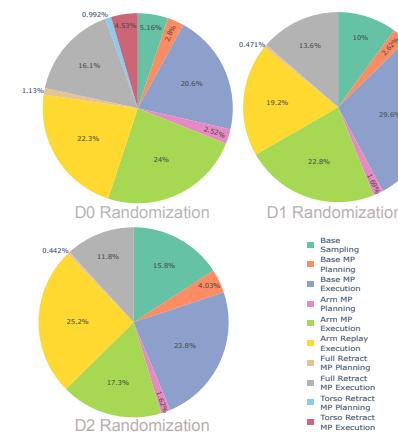


Figure 10: Compute cost of each component as a percentage of the episode

540 REPRODUCIBILITY STATEMENT  
541

542 We provide all the source code for simulation task setups, data generation, policy training, and detailed  
543 instructions for reproducing all data generation and policy learning results in our code submission  
544 (included in the supplementary materials). All training hyperparameters and implementation details  
545 are documented in Appendix C to facilitate full reproducibility.

546  
547 REFERENCES  
548

549 Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz,  
550 Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, Thomas  
551 Unterthiner, Daniel Keysers, Skanda Koppula, Fangyu Liu, Adam Grycner, Alexey Gritsenko,  
552 Neil Houlsby, Manoj Kumar, Keran Rong, Julian Eisenschlos, Rishabh Kabra, Matthias Bauer,  
553 Matko Bošnjak, Xi Chen, Matthias Minderer, Paul Voigtlaender, Ioana Bica, Ivana Balazevic, Joan  
554 Puigcerver, Pinelopi Papalampidi, Olivier Henaff, Xi Xiong, Radu Soricu, Jeremiah Harmsen, and  
555 Xiaohua Zhai. Paligemma: A versatile 3b vlm for transfer, 2024. URL <https://arxiv.org/abs/2407.07726>.

556 Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai,  
557 Lachy Groom, Karol Hausman, Brian Ichter, et al. pi\_0 : A vision-language-action flow model for  
558 general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

559 Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn,  
560 K. Gopalakrishnan, Karol Hausman, Alexander Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter,  
561 A. Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J. Joshi, Ryan C. Julian, Dmitry Kalashnikov,  
562 Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, S. Levine, Yao Lu, U. Malla, D. Manjunath, Igor  
563 Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell  
564 Quiambao, Kanishka Rao, M. Ryoo, Grecia Salazar, Pannag R. Sanketi, Kevin Sayed, Jaspia  
565 Singh, S. Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega,  
566 Q. Vuong, F. Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-1:  
567 Robotics transformer for real-world control at scale. *Robotics: Science and Systems*, 2022a. doi:  
568 10.48550/arXiv.2212.06817. URL <https://arxiv.org/abs/2212.06817v2>.

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

572 Zoey Chen, Sho Kiami, Abhishek Gupta, and Vikash Kumar. Genaug: Retargeting behaviors to  
573 unseen situations via generative augmentation. *arXiv preprint arXiv:2302.06671*, 2023.

574 Xuxin Cheng, Jialong Li, Shiqi Yang, Ge Yang, and Xiaolong Wang. Open-television: Teleoperation  
575 with immersive active visual feedback. In *8th Annual Conference on Robot Learning*, 2024. URL  
576 <https://openreview.net/forum?id=Yce2jeILGt>.

577 Sudeep Dasari, Frederik Ebert, Stephen Tian, Suraj Nair, Bernadette Bucher, Karl Schmeckpeper,  
578 Siddharth Singh, Sergey Levine, and Chelsea Finn. Robonet: Large-scale multi-robot learning.  
579 *arXiv preprint arXiv:1910.11215*, 2019.

580 Shivin Dass, Wensi Ai, Yuqian Jiang, Samik Singh, Jiaheng Hu, Ruohan Zhang, Peter Stone, Ben  
581 Abbatematteo, and Roberto Martín-Martín. Telemoma: A modular and versatile teleoperation  
582 system for mobile manipulation. *arXiv preprint arXiv:2403.07869*, 2024.

583 Zipeng Fu, Qingqing Zhao, Qi Wu, Gordon Wetzstein, and Chelsea Finn. Humanplus: Humanoid  
584 shadowing and imitation from humans. *arXiv preprint arXiv: 2406.10454*, 2024a. URL <https://arxiv.org/abs/2406.10454v1>.

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

587 Caelan Reed Garrett, Rohan Chitnis, Rachel Holladay, Beomjoon Kim, Tom Silver, Leslie Pack  
588 Kaelbling, and Tomás Lozano-Pérez. Integrated task and motion planning. *Annual review of  
589 control, robotics, and autonomous systems*, 4(1):265–293, 2021.

594 Caelan Reed Garrett, Ajay Mandlekar, Bowen Wen, and Dieter Fox. Skillmimicgen: Automated  
 595 demonstration generation for efficient skill learning and deployment. In *8th Annual Conference on*  
 596 *Robot Learning*, 2024. URL <https://openreview.net/forum?id=YOFrRTDC6d>.

597  
 598 Jiayuan Gu, Fanbo Xiang, Xuanlin Li, Zhan Ling, Xiqiang Liu, Tongzhou Mu, Yihe Tang, Stone Tao,  
 599 Xinyue Wei, Yunchao Yao, et al. Maniskill2: A unified benchmark for generalizable manipulation  
 600 skills. *arXiv preprint arXiv:2302.04659*, 2023.

601 Tairan He, Zhengyi Luo, Xialin He, Wenli Xiao, Chong Zhang, Weinan Zhang, Kris M. Kitani,  
 602 Changliu Liu, and Guanya Shi. Omnih2o: Universal and dexterous human-to-humanoid whole-  
 603 body teleoperation and learning. In *8th Annual Conference on Robot Learning*, 2024. URL  
 604 <https://openreview.net/forum?id=oL1WEZQa18>.

605 Brian Ichter, Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog,  
 606 Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, Dmitry Kalashnikov, Sergey Levine,  
 607 Yao Lu, Carolina Parada, Kanishka Rao, Pierre Sermanet, Alexander Toshev, Vincent Vanhoucke,  
 608 Fei Xia, Ted Xiao, Peng Xu, Mengyuan Yan, Noah Brown, Michael Ahn, Omar Cortes, Nicolas  
 609 Sievers, Clayton Tan, Sichun Xu, Diego Reyes, Jarek Rettinghouse, Jornell Quiambao, Peter  
 610 Pastor, Linda Luu, Kuang-Huei Lee, Yuheng Kuang, Sally Jesmonth, Nikhil J. Joshi, Kyle Jeffrey,  
 611 Rosario Jauregui Ruano, Jasmine Hsu, Keerthana Gopalakrishnan, Byron David, Andy Zeng, and  
 612 Chuyuan Kelly Fu. Do as I can, not as I say: Grounding language in robotic affordances. In  
 613 Karen Liu, Dana Kulic, and Jeffrey Ichniowski (eds.), *Conference on Robot Learning, CoRL 2022*,  
 614 *14-18 December 2022, Auckland, New Zealand*, volume 205 of *Proceedings of Machine Learning*  
 615 *Research*, pp. 287–318. PMLR, 2022. URL <https://proceedings.mlr.press/v205/ichter23a.html>.

616  
 617 Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess,  
 618 Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. pi\_0.5: A vision-language-action  
 619 model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

620 Yasuhiro Ishiguro, Tasuku Makabe, Yuya Nagamatsu, Yuta Kojio, Kunio Kojima, Fumihiro Sugai,  
 621 Yohei Kakiuchi, Kei Okada, and Masayuki Inaba. Bilateral humanoid teleoperation system using  
 622 whole-body exoskeleton cockpit tablis. *IEEE Robotics and Automation Letters*, 5(4):6419–6426,  
 623 2020. doi: 10.1109/LRA.2020.3013863.

624 Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. Rlbench: The robot  
 625 learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–  
 626 3026, 2020.

627 Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei,  
 628 Anima Anandkumar, Yuke Zhu, and Linxi Fan. Vima: General robot manipulation with multimodal  
 629 prompts. *arXiv preprint arXiv:2210.03094*, 2(3):6, 2022.

630 Yunfan Jiang, Chen Wang, Ruohan Zhang, Jiajun Wu, and Li Fei-Fei. Transic: Sim-to-real policy  
 631 transfer by learning from online correction. *arXiv preprint arXiv: 2405.10315*, 2024a. URL  
 632 <https://arxiv.org/abs/2405.10315v3>.

633 Yunfan Jiang, Ruohan Zhang, Josiah Wong, Chen Wang, Yanjie Ze, Hang Yin, Cem Gokmen, Shuran  
 634 Song, Jiajun Wu, and Li Fei-Fei. Behavior robot suite: Streamlining real-world whole-body  
 635 manipulation for everyday household activities. *arXiv preprint arXiv:2503.05652*, 2025a.

636 Zhenyu Jiang, Yuqi Xie, Jinhan Li, Ye Yuan, Yifeng Zhu, and Yuke Zhu. Harmon: Whole-body  
 637 motion generation of humanoid robots from language descriptions. In *8th Annual Conference on*  
 638 *Robot Learning*, 2024b. URL <https://openreview.net/forum?id=UUZ4Yw3lt0>.

639  
 640 Zhenyu Jiang, Yuqi Xie, Kevin Lin, Zhenjia Xu, Weikang Wan, Ajay Mandlekar, Linxi Fan, and  
 641 Yuke Zhu. Dexmimicgen: Automated data generation for bimanual dexterous manipulation via  
 642 imitation learning. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*,  
 643 2025b.

644 Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth  
 645 Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis,  
 646 et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*,  
 647 2024.

648 Misha Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel, and Aravind Srinivas. Rein-  
 649forcement learning with augmented data. *Advances in neural information processing systems*, 33:  
 650 19884–19895, 2020.

651 Sergey Levine, Peter Pastor, Alex Krizhevsky, Julian Ibarz, and Deirdre Quillen. Learning hand-  
 652 eye coordination for robotic grasping with deep learning and large-scale data collection. *The*  
 653 *International journal of robotics research*, 37(4-5):421–436, 2018.

654 Chengshu Li, Fei Xia, Roberto Martin-Martin, and Silvio Savarese. Hrl4in: Hierarchical rein-  
 655forcement learning for interactive navigation with mobile manipulators. In *Conference on Robot*  
 656 *Learning*, pp. 603–616. PMLR, 2020.

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

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

665 Jinhan Li, Yifeng Zhu, Yuqi Xie, Zhenyu Jiang, Mingyo Seo, Georgios Pavlakos, and Yuke Zhu.  
 666 OKAMI: Teaching humanoid robots manipulation skills through single video imitation. In *8th*  
 667 *Annual Conference on Robot Learning*, 2024b. URL <https://openreview.net/forum?id=URj5TQTXAM>.

668 Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke  
 669 Zhu, and Dieter Fox. Mimicgen: A data generation system for scalable robot learning using human  
 670 demonstrations. *arXiv preprint arXiv: 2310.17596*, 2023a. URL <https://arxiv.org/abs/2310.17596v1>.

671 Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke  
 672 Zhu, and Dieter Fox. Mimicgen: A data generation system for scalable robot learning using human  
 673 demonstrations. In Jie Tan, Marc Toussaint, and Kourosh Darvish (eds.), *Proceedings of The 7th*  
 674 *Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp.  
 675 1820–1864. PMLR, 06–09 Nov 2023b. URL <https://proceedings.mlr.press/v229/mandlekar23a.html>.

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

680 Lerrel Pinto and Abhinav Gupta. Supersizing self-supervision: Learning to grasp from 50k tries and  
 681 700 robot hours. In *2016 IEEE international conference on robotics and automation (ICRA)*, pp.  
 682 3406–3413. IEEE, 2016.

683 Silviu Pitis, Elliot Creager, and Animesh Garg. Counterfactual data augmentation using locally  
 684 factored dynamics. *Advances in Neural Information Processing Systems*, 33:3976–3990, 2020.

685 Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural*  
 686 *information processing systems*, 1, 1988.

687 Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured  
 688 prediction to no-regret online learning. In *Proceedings of the fourteenth international conference*  
 689 *on artificial intelligence and statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings,  
 690 2011.

691 Rutav Shah, Albert Yu, Yifeng Zhu, Yuke Zhu\*, and Roberto Martín-Martín\*. Bumble: Unifying  
 692 reasoning and acting with vision-language models for building-wide mobile manipulation. *arXiv*  
 693 *preprint*, 2024.

702 Kenneth Shaw, Yulong Li, Jiahui Yang, Mohan Kumar Srirama, Ray Liu, Haoyu Xiong, Russell  
 703 Mendonca, and Deepak Pathak. Bimanual dexterity for complex tasks. In *8th Annual Conference  
 704 on Robot Learning*, 2024. URL <https://openreview.net/forum?id=55tYfHvanf>.

705 Austin Stone, Ted Xiao, Yao Lu, Keerthana Gopalakrishnan, Kuang-Huei Lee, Quan Vuong, Paul  
 706 Wohlhart, Sean Kirmani, Brianna Zitkovich, Fei Xia, Chelsea Finn, and Karol Hausman. Open-  
 707 world object manipulation using pre-trained vision-language models. In *7th Annual Conference on  
 708 Robot Learning*, 2023. URL <https://openreview.net/forum?id=9a16taqfTzr>.

709 Balakumar Sundaralingam, Siva Kumar Sastry Hari, Adam Fishman, Caelan Garrett, Karl Van Wyk,  
 710 Valts Blukis, Alexander Millane, Helen Oleynikova, Ankur Handa, Fabio Ramos, et al. Curobo:  
 711 Parallelized collision-free robot motion generation. In *2023 IEEE International Conference on  
 712 Robotics and Automation (ICRA)*, pp. 8112–8119. IEEE, 2023.

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

716 Josiah Wong, Albert Tung, Andrey Kurenkov, Ajay Mandlekar, Li Fei-Fei, Silvio Savarese, and  
 717 Roberto Martín-Martín. Error-aware imitation learning from teleoperation data for mobile manipu-  
 718 lation. In *Conference on Robot Learning*, pp. 1367–1378. PMLR, 2022.

719 Jimmy Wu, Rika Antonova, Adam Kan, Marion Lepert, Andy Zeng, Shuran Song, Jeannette  
 720 Bohg, Szymon Rusinkiewicz, and Thomas Funkhouser. Tidybot: personalized robot assis-  
 721 tance with large language models. *Autonomous Robots*, 47:1087–1102, 2023. doi: 10.  
 722 1007/s10514-023-10139-z. URL <https://link.springer.com/article/10.1007/s10514-023-10139-z/fulltext.html>.

723 Jimmy Wu, William Chong, Robert Holmberg, Aaditya Prasad, Yihuai Gao, Oussama Khatib,  
 724 Shuran Song, Szymon Rusinkiewicz, and Jeannette Bohg. Tidybot++: An open-source holonomic  
 725 mobile manipulator for robot learning. *arXiv preprint arXiv: 2412.10447*, 2024a. URL <https://arxiv.org/abs/2412.10447v1>.

726 Qi Wu, Zipeng Fu, Xuxin Cheng, Xiaolong Wang, and Chelsea Finn. Helpful doggybot: Open-world  
 727 object fetching using legged robots and vision-language models. In *arXiv*, 2024b.

728 Mengdi Xu, Peide Huang, Wenhao Yu, Shiqi Liu, Xilun Zhang, Yaru Niu, Tingnan Zhang, Fei Xia,  
 729 Jie Tan, and Ding Zhao. Creative robot tool use with large language models. *arXiv preprint arXiv:  
 730 2310.13065*, 2023.

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

734 Jingyun Yang, Zi ang Cao, Congyue Deng, Rika Antonova, Shuran Song, and Jeannette Bohg.  
 735 Equibot: Sim(3)-equivariant diffusion policy for generalizable and data efficient learning. *arXiv  
 736 preprint arXiv: 2407.01479*, 2024.

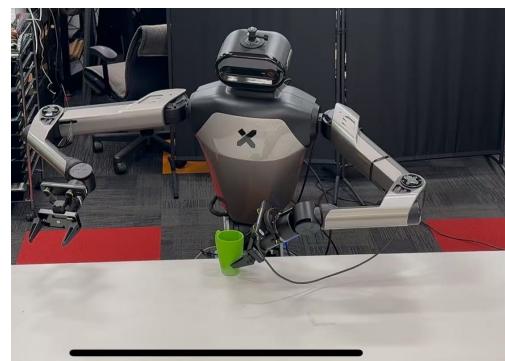
737 Lujie Yang, HJ Suh, Tong Zhao, Bernhard Paus Graesdal, Tarik Kelestemur, Jiuguang Wang, Tao  
 738 Pang, and Russ Tedrake. Physics-driven data generation for contact-rich manipulation via trajectory  
 739 optimization. *arXiv preprint arXiv:2502.20382*, 2025.

740 Denis Yarats, Ilya Kostrikov, and Rob Fergus. Image augmentation is all you need: Regularizing  
 741 deep reinforcement learning from pixels. In *International conference on learning representations*,  
 742 2021.

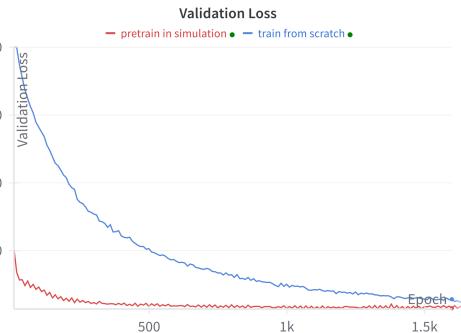
743 Kuan-Ting Yu, Maria Bauza, Nima Fazeli, and Alberto Rodriguez. More than a million ways to be  
 744 pushed. a high-fidelity experimental dataset of planar pushing. In *2016 IEEE/RSJ international  
 745 conference on intelligent robots and systems (IROS)*, pp. 30–37. IEEE, 2016.

746 Tianhe Yu, Ted Xiao, Austin Stone, Jonathan Tompson, Anthony Brohan, Su Wang, Jaspiar Singh,  
 747 Clayton Tan, Jodilyn Peralta, Brian Ichter, et al. Scaling robot learning with semantically imagined  
 748 experience. *arXiv preprint arXiv:2302.11550*, 2023.

756 Yanjie Ze, Zixuan Chen, Wenhao Wang, Tianyi Chen, Xialin He, Ying Yuan, Xue Bin Peng, and  
757 Jiajun Wu. Generalizable humanoid manipulation with improved 3d diffusion policies. *arXiv*  
758 *preprint arXiv:2410.10803*, 2024.  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809

810 A ADDITIONAL EXPERIMENTAL RESULTS  
811812  
813 A.1 SIM-TO-REAL  
814815  
816 **Real robot setup.** We investigate whether MOMAGEN can support real-world deployment, using  
817 the Galaxeia R1 mobile manipulator, the same platform used for data collection and generation in  
818 simulation. Point clouds are captured with R1’s onboard ZED 2 stereo camera and an additional ZED  
819 Mini mounted above the left gripper. To mitigate depth noise, we apply the built-in AI model for  
820 point cloud refinement, followed by farthest point sampling to 4096 points, as in simulation. We  
821 evaluate on the Pick Cup (D0) task with a table of comparable height to the simulated setup and a  
822 3D-printed green cup. The real-world configuration is shown in Figure 11a.  
823824 **Experiment setup.** Zero-shot transfer is challenging due to domain gaps, particularly for vision-  
825 based policies (Jiang et al., 2024a). In our case, simulated and real RGB images differ substantially,  
826 and real-world depth estimates are noisy. To mitigate this, we collect 50 real-world demonstrations  
827 for the Pick Cup (D0) task and evaluate whether simulation pretraining improves real-world learning.  
828829 **Sim-to-real experiments demonstrate the benefit of simulation pretraining.** For WB-VIMA, we  
830 compare models initialized from a simulation-pretrained checkpoint (1.8M steps on 1,000 synthetic  
831 demos) against random initialization. Both are fine-tuned for 35k steps on 40 real demos, with 10  
832 held out for validation. As shown in Figure 11b, the pretrained model converges much faster, reaching  
833 a validation loss of 3.0 compared to 6.0 for the baseline. On real hardware, it achieves 10% success,  
834 while the baseline fails entirely. Despite low absolute success, the pretrained model consistently  
835 exhibits meaningful behaviors (e.g., reaching the cup), unlike the baseline, which shows no progress.  
836837 For  $\pi_0$ , we compare two variants initialized from pretrained weights: one further trained on simulation  
838 data before fine-tuning on 40 real demonstrations, and one fine-tuned only on real data. The  
839 simulation-pretrained model achieves 60% success, while the baseline remains at 0%. Although the  
840 baseline occasionally attempts grasps, it fails due to poor precision, highlighting the challenge of  
841 learning robust visuomotor policies from limited real data. This result shows that even with a strong  
842 foundation pretrained on 10k+ hours of robot data, it is beneficial to first adapt to the target robot  
843 embodiment and scene setup with simulation data before fine-tuning on scarce real-world data.  
844845 Overall, these results show that diverse simulation-generated data provide a strong prior, enabling  
846 efficient fine-tuning and effective sim-to-real transfer, particularly in low-data regimes. This finding  
847 demonstrates the practical utility of MOMAGEN-generated data for real-world robotics. More  
848 real-world policy rollouts can be found on our website.  
849

(a) Real World Setup



(b) Validation Loss

Figure 11: Real world setup for Pick Cup (a) and validation loss curve of WB-VIMA (b).

864 **B ADDITIONAL DATA GENERATION DETAILS**  
865866 **B.1 CAN MoMAGEN GENERALIZE ACROSS DIFFERENT FAMILIES OF MOBILE MANIPULATION**  
867 **TASKS?**  
868869 Beyond the tasks analyzed in the main paper (rigid-body pick-and-place, wiping, and stacking), we  
870 further investigate whether MoMAGEN extends to other common families of mobile manipulation  
871 behaviors. In particular, we consider three representative tasks that span different “families” of mobile  
872 manipulation: (i) articulated-object interaction, (ii) pouring and particle-transfer tasks, and (iii) whole-  
873 body control. These proof-of-concept demonstrations highlight the versatility of MoMAGEN in  
874 generating stable and collision-free trajectories across diverse tasks. Videos for all tasks are provided  
875 on our project website.  
876877 **Get Bottle from Fridge (Articulated Object Manipulation).** The robot navigates to a fridge,  
878 grasps and opens the door with one arm, and then retrieves a bottle inside. This task also demonstrates  
879 MoMAGEN’s ability to manage occlusion, as the bottle is initially hidden until the fridge is opened.  
880 For the initial subtasks, the fridge is treated as the task-relevant object; once the door is open, the  
881 target switches to the bottle.  
882883 **Pour Cat Food (Pouring and Particle Interaction).** The robot grasps a tin of cat food containing  
884 rigid particles, navigates to a bowl, and performs a controlled pouring motion until all particles  
885 are transferred. This task highlights MoMAGEN’s ability to generate trajectories that require  
886 coordinated container orientation, stable end-effector motion, and interaction with particle-like  
887 objects, representative of the broader family of pouring tasks.  
888889 **Push Chair (Whole-Body Control).** A chair is initially positioned outside a table. The robot  
890 navigates to the back of the chair, establishes contact, and pushes it into place underneath the  
891 table. This task requires coupling the robot’s base movement with the end-effector contact direction,  
892 demonstrating that MoMAGEN can generate whole-body coordination behaviors involving sustained  
893 contact and the manipulation of large objects.  
894895 **B.2 CAN MoMAGEN ACHIEVE CROSS-EMBODIMENT DATA GENERATION?**  
896897 We evaluate cross-embodiment data generation by using a single demonstration collected on a Galexea  
898 R1 robot to generate trajectories for a TIAGo robot. Although both platforms feature dual 7-DoF arms  
899 and holonomic bases, their torso designs and arm workspaces differ substantially. Our experiments  
900 show that MoMAGEN can successfully produce Pick Cup demonstrations on TIAGo using the R1  
901 source demo. This transfer is enabled by planning and replaying dense end-effector trajectories,  
902 rather than joint-space trajectories, making the process largely agnostic to robot-specific kinematics.  
903 These results highlight the robustness and flexibility of our framework across platforms. Videos of  
904 the generated demonstrations of this new robot embodiment are available on our website.  
905906 Cross-embodiment transfer, however, has limitations. Tasks requiring operation in confined spaces  
907 may fail due to differences in gripper size, leading to unavoidable collisions. Similarly, TIAGo’s  
908 bulkier arms can render some trajectories from R1 demos infeasible or even cause self-collisions.  
909 Thus, while MoMAGEN offers a proof of concept for cross-embodiment data generation, advancing  
910 more powerful approaches in this direction remains an exciting avenue for future work.  
911912 **B.3 VISUALIZATIONS OF DOMAIN RANDOMIZATION SCHEMES**  
913914 Figure 12 illustrates the domain randomization schemes (D0/D1/D2) across all four tasks. Lever-  
915 aging base sampling, MoMAGEN can generate successful demonstrations under more aggressive  
916 randomization (D1) than prior methods (D0). With motion planners in the loop, it further handles  
917 obstacles in both base and arm motion (D2), producing richer trajectories and more diverse visual  
918 observations.

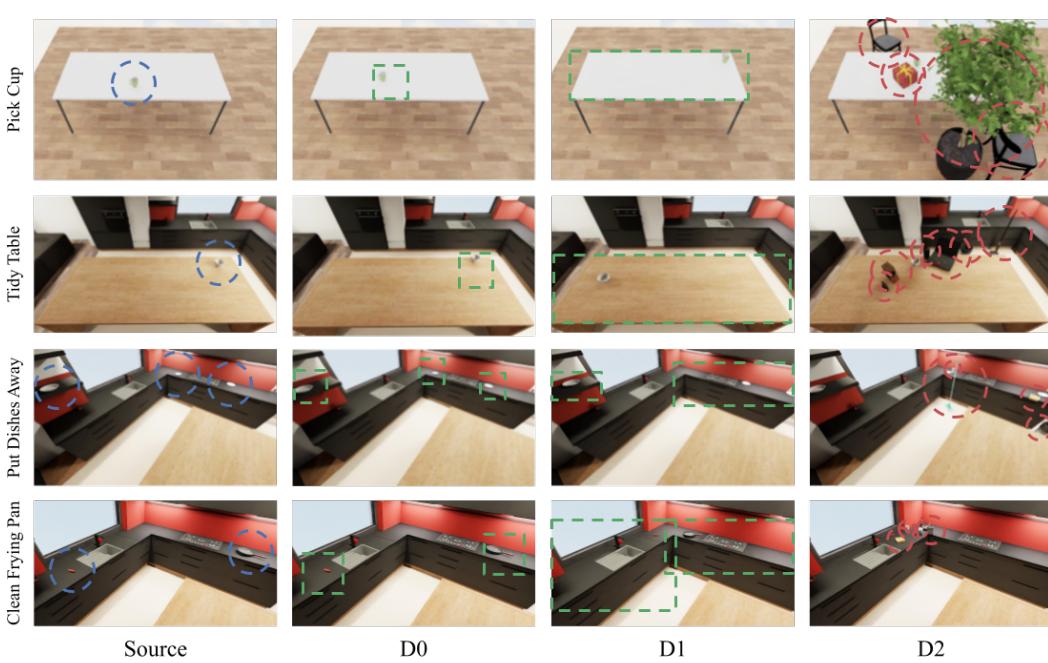


Figure 12: Visualization of Domain Randomization. Blue represents the task-relevant objects. Green represents the approximate randomization range for these objects (for D1 and D2, the range is anywhere on the same furniture). Red represents the obstacles/distractor objects.

#### B.4 VISUALIZATIONS OF DATA DIVERSITY

Figure 13, 14, and 15 extend the visualizations in Figure 4, comparing MoMAGen to SkillMimicGen and DexMimicGen across the other three tasks. As expected, the object diversity in D1 (subfigures a) induces substantially more varied robot base and end-effector trajectories (subfigures b–c) and joint configurations (subfigure d), resulting in broader state- and action-space coverage.

#### B.5 COMPUTE RESOURCES

The compute resource required for data generation scales with 1) the number of subtasks, 2) the length of each subtask (including the free-space subtask and contact-rich subtask), affecting motion execution time, and 3) the complexity of the scene (task-relevant objects, obstacles, etc), affecting motion planning time. It inversely scales with the data generation success rate. Each successful demonstration takes 0.1 to 1.3 GPU hours to generate, ranging from Pick Cup task to Put Dishes Away task. All data generation runs are conducted on a single NVIDIA TITAN RTX GPU.

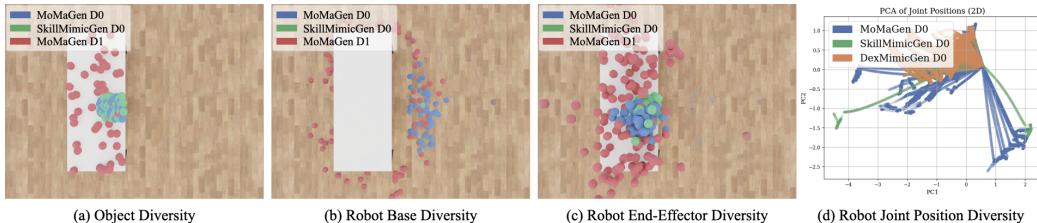


Figure 13: Generated data diversity analysis for Pick Cup task (50 trajectories, subsampled).

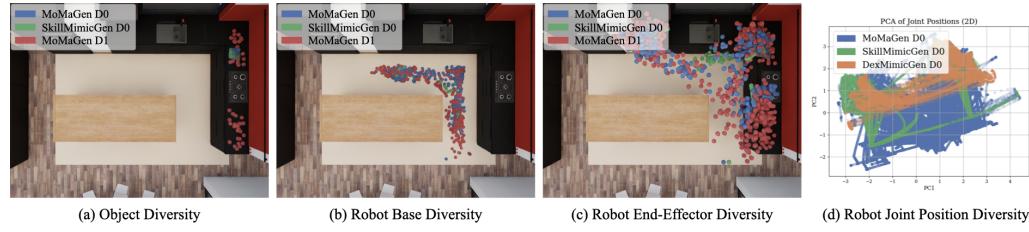


Figure 14: Generated data diversity analysis for Put Dishes Away task (50 trajectories, subsampled).

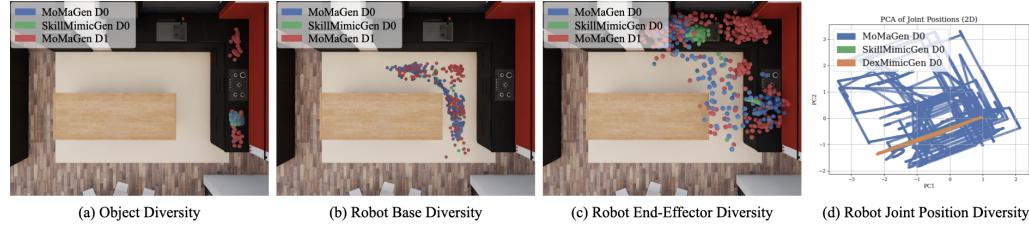


Figure 15: Generated data diversity analysis for Clean Frying Pan task (50 trajectories, subsampled).

## B.6 FAILURE STEP ANALYSIS

Fig. 16 illustrates the distribution of failures across the individual steps of each multi-step task. For example, in the put dishes away task, failures occur more frequently during the navigation and placement steps compared to the plate-picking step.

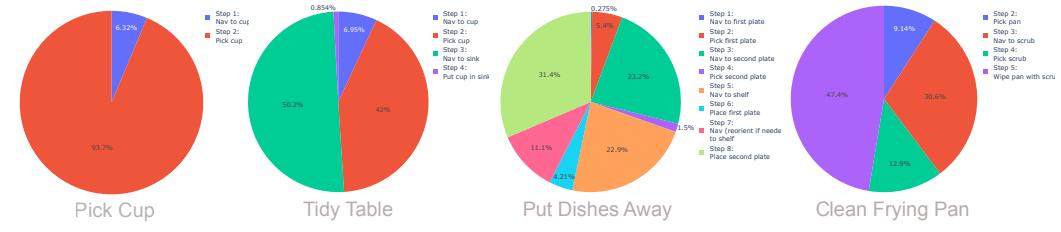


Figure 16: Distribution of failures across task-level steps within each multi-step task.

## C ADDITIONAL POLICY TRAINING DETAILS

In this section, we first describe the data cleaning process in Section C.1 and then provide additional training details for WB-VIMA (Section C.2) and  $\pi_0$  (Section C.3). For both methods, 90% of the demonstrations are used for training and the remaining 10% for validation.

### C.1 DATA CLEANING

Teleoperation in simulation makes depth perception difficult, often causing operators to hesitate and leave the gripper nearly stationary before grasping or making contact. As a result, some demonstrations contain short “frozen” segments, which can degrade training, especially for imitation learning methods with limited temporal context (e.g., WB-VIMA uses a 2-step history, and  $\pi_0$  relies only on the current state). To address this, we introduce a preprocessing step that removes frozen segments. Specifically, for a trajectory of length  $T$ , if at any timestep  $i \in [0, T - 5]$  the absolute joint position difference between step  $i$  and  $i + 5$  is below  $1 \times 10^{-3}$  in all dimensions, we treat the interval  $[i, i + 5]$  as frozen and discard it prior to policy learning.

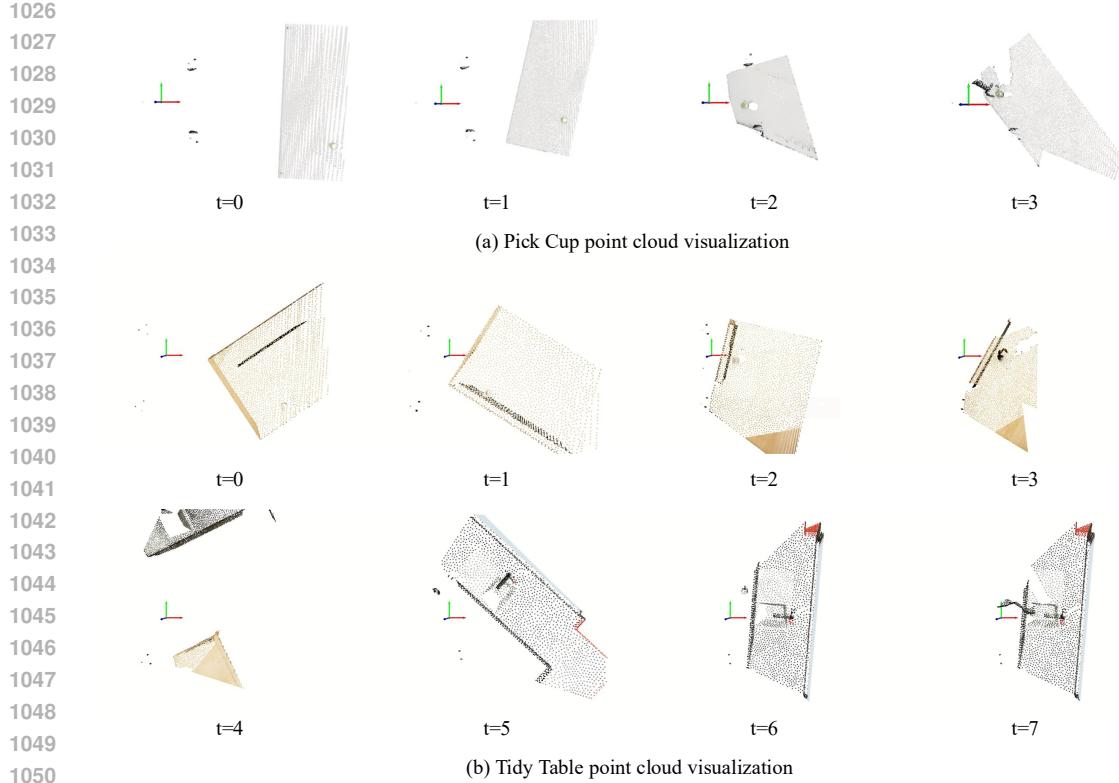


Figure 17: Visualization of ego-centric point cloud for Pick Cup and Tidy Table, used for WB-VIMA training. We fuse the colored point-cloud from three RGB-D cameras, crop with a robot-centric bounding box, and then [downsample](#) to 4096 points via farthest point sampling.

## C.2 WB-VIMA TRAINING DETAILS

**Policy Architecture.** WB-VIMA (Jiang et al., 2025a) takes as input both proprioceptive observations and an egocentric colored point cloud. The proprioceptive inputs include base velocity  $v^{\text{base}} \in \mathbb{R}^3$ , torso joints  $q^{\text{torso}} \in \mathbb{R}^4$ , left arm joints  $q^{\text{left}} \in \mathbb{R}^6$ , left gripper width  $q^{\text{grip-left}} \in \mathbb{R}^1$ , right arm joints  $q^{\text{right}} \in \mathbb{R}^6$ , and right gripper width  $q^{\text{grip-right}} \in \mathbb{R}^1$ . The point cloud is constructed by fusing RGB-D images from an eye-level camera and two wrist-mounted cameras; examples for Pick Cup and Tidy Table are shown in Figure 17. It is then cropped with a robot-centric bounding box and downsampled to 4096 points via farthest point sampling. WB-VIMA uses a 2-step history input. Additional architectural details are available in the original paper (Jiang et al., 2025a). For each task, we train WB-VIMA from scratch to obtain a single-task policy.

**Hyperparameters.** Training hyperparameters for the PointNet (point cloud processing), diffusion head, transformer backbone, learning rates, and task-specific point cloud clipping ranges are summarized in Table 4. The ego-centric clipping ranges are customized per task to reflect differences in scene layout and object randomization. All hyperparameters are chosen via grid search.

## C.3 $\pi_0$ TRAINING DETAILS

**Policy Architecture.** We fine-tune a pretrained  $\pi_0$  model using LoRA with rank 32 for 50k steps and a batch size of 64. The model takes as input the RGB images from the eye-level and two wrist-mounted cameras, along with the same proprioceptive signals used by WB-VIMA, and predicts the target joint position for the next 50 time steps.

**Hyperparameters.** All RGB inputs are resized to  $224 \times 224$ . Actions and proprioceptive signals are normalized using the 1st–99th quantile and zero-padded to match the 32-dimensional action

1080	Hyperparameter	Value
1081	Number of points in point cloud	4096
1082	PointNet hidden dim	256
1083	PointNet hidden depth	2
1084	PointNet output dim	256
1085	PointNet activation	GELU
1086	Proprioceptive MLP input dim	21
1087	Proprioceptive MLP hidden dim	2
1088	Proprioceptive MLP hidden depth	256
1089	Proprioceptive MLP output dim	256
1090	Proprioceptive MLP activation	ReLU
1091	Transformer embedding size	512
1092	Transformer layers	4
1093	Transformer heads	8
1094	Transformer dropout rate	0.1
1095	Transformer activation	GEGLU
1096	Action dim	21
1097	Unet down dims	[128, 256]
1098	Unet kernel size	5
1099	Unet number of groups	8
1100	Diffusion step embedding dim	256
1101	Diffusion noise scheduler	DDIM
1102	Number of training steps	100
1103	Beta schedule	squaredcos_cap_v2
1104	Number of denoise steps per inference	16
1105	Learning rate	1e-4
1106	Learning rate scheduler	Cosine decay
1107	Learning rate warmup steps	100000
1108	Learning rate cosine steps	1300000
1109	Optimizer	AdamW
1110	Batch size per GPU	128
1111	Number of GPUs in parallel	2
1112	Pick Cup (D0) pointcloud clip range	x: [0.0, 2.3], y: [-0.5, 0.5], z: [0.7, 2.0]
1113	Pick Cup (D1) pointcloud clip range	x: [0.0, 2.7], y: [-1.0, 1.0], z: [0.7, 2.0]
1114	Tidy Table (D0) pointcloud clip range	x: [0.0, 2.3], y: [-1.5, 1.5], z: [0.7, 1.5]
1115	Model size	37.1M

Table 4: Hyperparameters for WB-VIMA.

1134	Hyperparameter	Value
1135	Proprioceptive MLP input dim	32
1136	Proprioceptive MLP output dim	1024
1138	Flow Matching MLP input dim	32
1139	Flow Matching MLP hidden dim	2048
1140	Flow Matching MLP hidden depth	2
1141	Flow Matching MLP output dim	1024
1142	Flow Matching MLP activation	swish
1143	PaliGemma embedding size	2048
1144	PaliGemma number of layers	18
1145	PaliGemma number of heads	18
1146	PaliGemma heads dimension	256
1147	PaliGemma MLP dimension	16384
1148	Action expert embedding size	1024
1149	Action expert MLP dimension	4096
1150	Number of flow matching steps per inference	10
1151	Learning rate	2.5e-5
1152	Learning rate scheduler	Cosine decay
1153	Learning rate warmup steps	1000
1154	Learning rate cosine steps	30000
1155	Optimizer	AdamW
1156	Batch size	64
1157	Number of GPUs in parallel	4
1158	Model size	3.3B
1159		

Table 5: Hyperparameters for  $\pi_0$ .

space expected by  $\pi_0$ . The model uses a PaliGemma VLM backbone (Beyer et al., 2024) with a 300M-parameter action expert. Further architectural details are in (Black et al., 2024), and additional training hyperparameters are listed in Table 5.

#### C.4 COMPUTE RESOURCES

With a batch size of 128, WB-VIMA can be trained on two RTX 3090 GPUs (24GB each), taking approximately 40 hours to reach 1 million steps. With a batch size of 64,  $\pi_0$  can be trained on four H200 GPUs, taking approximately 7 hours to reach 50k steps.

#### D USE OF LARGE LANGUAGE MODELS

We used a large language model (ChatGPT) solely for writing refinement, including grammar correction and improving clarity and conciseness of the text. The model was not used for research ideation, experimental design, analysis, or content generation beyond language editing.