MoMaGen: Generating Demonstrations under Soft and Hard Constraints for Multi-Step Bimanual Mobile Manipulation

Chengshu Li[*, 1], Mengdi Xu[*, 1], Arpit Bahety[*, 2], Hang Yin[1], Yunfan Jiang[1], Huang Huang[1], Josiah Wong[1], Sujay Garlanka[1], Cem Gokmen[1], Ruohan Zhang[1], Weiyu Liu[1], Jiajun Wu[1], Roberto Martín-Martín[2], Li Fei-Fei[1]
[*] Equal Contribution, [1] Stanford University, [1] The University of Texas at Austin

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 [1–3] 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. More details are on our project page: `https://momagen-rss.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 [4–8]. 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 [1–3, 9, 10]. Compared to offline data augmentation techniques such as those based on image augmentation [11–15], 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 [16] through new generated experiences validated in simulation to ensure quality. Notably, the X-Gen family of techniques [1–3, 9, 10] leverages simulation and augmentation based

Robotics: Science and Systems (RSS) 2025 Workshop on Mobile Manipulation: Emerging Opportunities Contemporary Challenges.

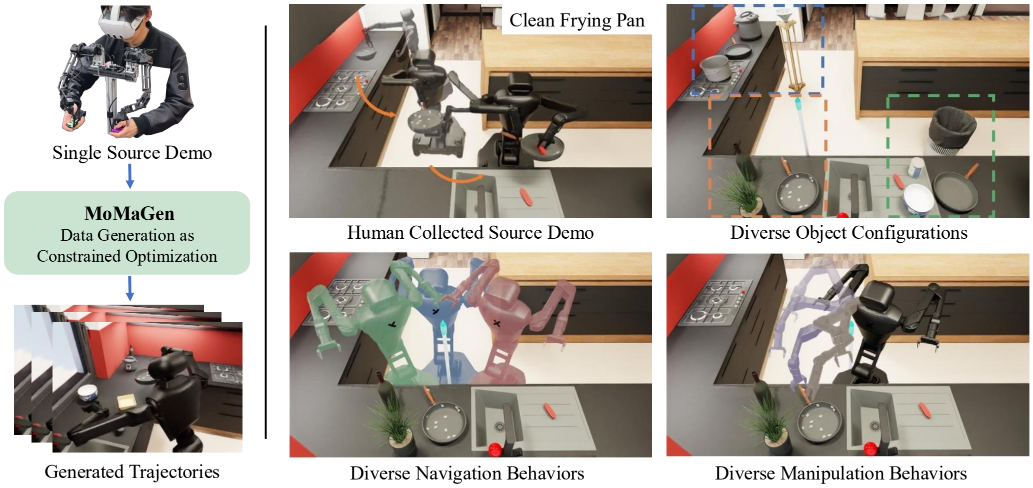


Figure 1: (left) MoMaGen uses a single human-collected demonstration to generate a large set of demonstrations, formulating data generation as a constrained optimization problem. (top-left) shows a human-collected demo for cleaning a pan with a scrub. (top-right) shows three novel object configurations 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.

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.

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 [17–19]. 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 [19–21] (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 [2] or replay [3, 22] 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 right before manipulation) and soft constraints (e.g., visibility while navigation). Significantly, we realized that previous methods of the X-Gen family can be interpreted as using different (but insufficient) hard and soft constraints for data generation, providing a unified framework. With data generated by MoMaGen, we evaluate on four multi-step bimanual mobile manipulation tasks. We 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 with a single source demonstration. MoMaGen generalizes across most existing automated data-generation approaches and offers a principled foundation for developing future methods.

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

Methods	Bimanual	Mobile	Obstacles	Base Random.	Active Perception	Hard Constraints	Soft Constraints
MimicGen [1]	✗	✓	✗	✗	✗	Succ	N/A
SkillMimicGen [2]	✗	✗	✓	✗	✗	Succ, Kin, C-Free	N/A
DexMimicGen [3]	✓	✗	✗	✗	✗	Succ, Temp	N/A
DemoGen [9]	✗	✗	✓	✗	✗	Kin, C-Free	N/A
PhysicsGen [10]	✓	✗	✗	✗	✗	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.

operators over extended periods of time [4–6] or needs to rely on crowdsourced teleoperation systems such as RoboTurk [23]. Offline data augmentation techniques can boost data quantity and diversity by perturbing existing trajectories [1, 2], or leveraging image augmentation techniques and generative models [11–15]. 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 [24–27] and pre-programmed (e.g., scripted) experts [28–31], which are yet to be proven effective for complex tasks without additional components such as planning [32]. X-Gen [1–3, 9, 10] represents a hybrid approach that leverages simulation and augmentation to build upon a handful of human demonstrations as seeds to generate many new variations, augmenting the data by a factor of $25\times$ to $350\times$ [1, 3], while ensuring synthesized data is valid. 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 chain navigation and manipulation, move through an environment to position itself, and then perform manipulation. Collecting teleoperation data for mobile manipulation is significantly more costly: operators must simultaneously control the robot base and arms [17, 19, 20, 33, 34], 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 [19, 35–42] and large pretrained models [43–49].

3 Problem Formulation: 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 incorporates 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 length and reduced jerkiness). Generating valid data requires strictly satisfying hard constraints while minimizing the costs associated with soft constraints. We formally define the problem as follows.

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 $\mathcal{D}_{src} = \{d^j = (s_0^j, a_0^j, \ldots, s_{T_{src}}^j)\}_{j=0}^{N_{src}}$, 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 $\mathcal{D} = \{d\}^{N_{gen}}$ given the source demonstrations $\mathcal{D}_{src}$ and a set of constraints $\{\mathcal{G}_i\}$. With the generated demonstrations, we can train Behavioral Cloning [50] policies π_θ using $\arg\min_\theta \mathbb{E}_{(s,a)\sim\mathcal{D}}[-\log \pi_\theta(a|s)]$.

Following prior work [1–3], 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, 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. picking, placing, wiping). The relative poses between the end-effector frame and the 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:

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

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 rechability 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, besides the commonly used hard and soft constraints mentioned in the previous section, we highlight a few key technical innovations that are essential for generating high-quality bimanual mobile manipulation demonstrations.

Reachability as Hard Constraint. One of the key distinctions between mobile and stationary manipulation is that, in the former, the robot must actively control its mobile base to position itself appropriately for effective downstream manipulation. While prior works [3, 22] have demonstrated mobile manipulation tasks, such as placing a pan on a stove or inserting a plate into a dishwasher, the navigation trajectories in these approaches are copied directly from source demonstrations without any adaptation. Such methods fail when object randomization places targets beyond the reachable workspace of the robot arm, making manipulation infeasible from the original base pose. To address this, we impose reachability as a hard constraint during data generation. Specifically, we ensure that the sampled base pose allows all required end-effector trajectories for downstream manipulation to remain within the robot's reachable workspace.

Object Visibility during Manipulation as Hard Constraint. A valid robot base pose must also satisfy a hard visibility constraint: the task-relevant objects must be within the field of view of the robot. This requirement is critical because the generated data is intended to train visuomotor policies, which rely on consistent visual access to task-relevant objects during manipulation. To enforce this, we ensure that each sampled base pose allows the robot's head camera to observe the task-relevant objects without occlusion, leveraging additional camera or torso articulation when necessary.

Object Visibility during Navigation as Soft Constraint. While maintaining task-relevant object visibility during navigation toward the target base pose is desirable, it is not strictly required and is therefore treated as a soft constraint. To encourage this behavior, we introduce an additional visibility cost that biases the robot's head camera to remain oriented toward the target object during navigation.

Retraction as Soft Constraint. After manipulation, it is usually beneficial for the robot to retract its torso and arms into a compact, tucked configuration, before the next phase of navigation. This reduces the robot's footprint and makes the future base motion generation easier and 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 our framework that leverages the novel constraints introduced above to efficiently generate diverse demonstration data.

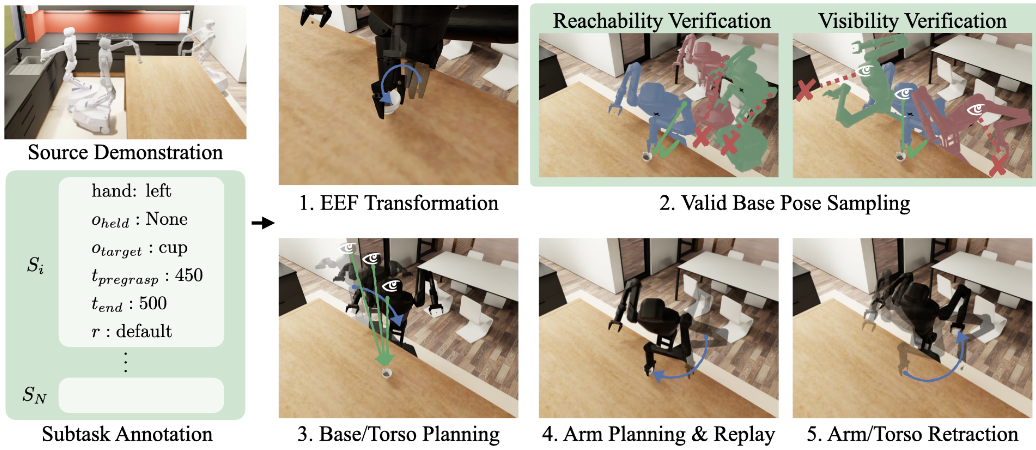


Figure 2: MoMaGen method. Given a single source demonstration, as well as annotations for object-centric subtasks for each end-effector, MoMaGen first randomizes scene configuration, and transforms the end-effector poses from the source demo to the new objects' frame of reference. For each subtask, it 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

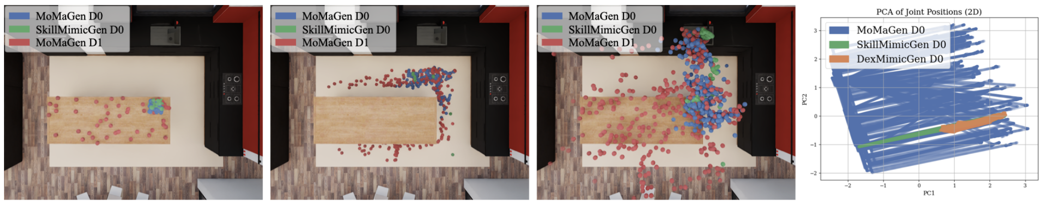
Input: original demo, new initial state s_0
Output: generated demo
1: **for** each segment **do**
2: Get current $\mathbf{T}^{\text{base}}$, $\mathbf{T}^{\text{cam}}$, q^{torso}, q^{arm}
3: **if** held object not in hand **then abort**
4: Compute transformed end-effector pose $\mathbf{T}^{\text{eef}}$ using new target object pose
5: Check visibility of target object with $\mathbf{T}^{\text{cam}}$
6: Solve IK for arm trajectory $\{q_t^{\text{arm}}\}$ with current $\mathbf{T}^{\text{base}}$, $\mathbf{T}^{\text{cam}}$
7: **while** not visible or no IK exists **do**
8: Sample new base pose $\mathbf{T}^{\text{base}}$
9: Sample new camera pose $\mathbf{T}^{\text{cam}}$
10: Solve IK for arm $\{q_t^{\text{arm}}\}$ and torso $\{q_t^{\text{torso}}\}$ with sampled $\mathbf{T}^{\text{base}}$, $\mathbf{T}^{\text{cam}}$
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
12: Plan motion for $\{q_t^{\text{arm}}\}$ from previous $\mathbf{T}^{\text{eef}}$ to pregrasp $\mathbf{T}^{\text{eef}}$
13: Control end-effector in task space to follow transformed $\mathbf{T}^{\text{eef}}$
14: Attempt retraction

Source Demonstration Annotation. For each demonstration, we first segment it into temporally ordered subtasks that include single arm uncoordinated motion or bimanual coordinated motion that require synchronization of the robot's left and right arms at subtask boundaries. Each subtask is annotated with the target object o_{target}, the object held by the gripper o_{held}, the timestep immediately preceding contact $t_{pregrasp}$, the motion's end timestep t_{end}, and the retraction type r to execute after the motion. Figure 2 illustrates an annotated subtask of grasping a cup using a single arm. To stress test our method and minimize human effort, we collect and annotate a single source demo (N_{src}=1).

Demonstration Generation. MoMaGen generates new robot demonstrations for novel initial states by following the high-level procedure described in Algorithm 1. For each subtask of a source demo, we first verify the robot is holding the required object and abort early if not (line 3), likely due to the previous grasping failures. We then compute end-effector poses for contact-rich motions by applying the appropriate transforms from the original demonstration (line 4). Next, we check reachability and visibility constraints for the current base and head camera configuration (lines 5-6); if these constraints are satisfied, the robot proceeds directly to manipulation (lines 12–13). Otherwise, we enter a sampling loop (lines 7–11) to find a feasible base and camera pose, repeatedly sampling

Pick Cup Tidy Table Put Dishes Away Clean Frying Pan

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.



(a) Object Diversity (b) Robot Base Diversity (c) Robot End-Effector Diversity (d) Robot Joint Position Diversity

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.

and checking configurations until a valid one is found. Throughout this process, heuristics-based sampling and inverse kinematics are used to guarantee both reachability and visibility (lines 8-10). Once a valid base configuration is found and reached (line 11), the robot executes the manipulation phase, bringing the end effector(s) to the pregrasp pose wiht motion planning and replaying the contact-rich segment using task-space control (lines 12–13). Finally, the robot attempts to retract its arms and torso to a canonical or previous configuration (line 14). This process repeats for each annotated subtask. We use cuRobo [51] for all motion planning and inverse kinematics checking.

Key Novelties. We highlight several key novelties of our approach for generating mobile manipulation demonstrations. **(1) Full-body Motion**: Unlike prior work that primarily focuses on the end-effector pose $\mathbf{T}^{eef}$, our method simultaneously considers the end-effector pose $\mathbf{T}^{eef}$, head camera pose $\mathbf{T}^{cam}$, and base pose $\mathbf{T}^{base}$. **(2) Visibility Guarantee**: We explicitly ensure that the target object α is visible before manipulation (lines 5 and 9), and further incorporate a soft visibility constraint to encourage the robot to keep the object in view during navigation (line 11). **(3) Expanded Workspace**: To fully leverage the robot's mobility, we actively sample base poses near the target object (line 8) and plan base motions to transition efficiently between target objects throughout the entire room (line 11). **(4) Efficient Generation**: To improve data generation efficiency, we prioritize inverse kinematics checks, which are significantly faster than full motion planning for preemptive filtering. We also decompose the robot's configuration into subspaces for the torso and arms, enabling more efficient conditional sampling in a manner similar to integrated task and motion planning methods [52].

5 Experiments and Results

We aim to investigate whether MoMaGen can effectively generate demonstrations for multi-step bimanual mobile manipulation tasks. We evaluate MoMaGen in four household tasks (Section 5.1) and with three different object/scene randomization schemes (Section 5.2). We compare MoMaGen with two data generation baselines [2, 3] and show that MoMaGen generates data with larger data diversity, higher data generation success rates for complex tasks, and substantially higher task-relevant object visibility which is critical for visuomotor policies in Section 5.3. We further analyze how MoMaGen's generated demonstrations help train imitation learning policies in Section 5.4.

5.1 Task Setup

We evaluate MoMaGen on four household tasks that require mobile manipulation, illustrated in Figure 3. These tasks are inspired by BEHAVIOR-1K benchmark and are implemented in OmniGibson [53, 54]. **Pick Cup** involves single arm mobile manipulation, where the robot must navigate to the table to grasp the cup, and lift it. Success is defined as raising the cup to a specified height above the table surface. **Tidy Table** involves longer range of mobile manipulation, where the robot must navigate to the countertop to pick up the cup, and then place it in the sink. The task is successful when the teacup is placed inside the sink. **Put Dishes Away** involves uncoordinated bimanual mobile manipulation. There is one plate initially on the shelf, and two are on the countertop.

The robot must pick up all countertop plates and stack them on the shelf. The task is successful if all plates form a stable stack on the shelf. **Clean Frying Pan** involves coordinated bimanual mobile manipulation, where the robot must use the brush to scrub the pan to remove the dust while holding the pan. Success is measured by the percentage of dust removed from the pan's surface. For each task we collect one human demonstration through teleoperation in OmniGibson. Each human expert demonstration lasts approximately 1–3 minutes. These tasks involve substantial mobile base movement, which accounts for approximately 45% of the total demonstration duration.

5.2 Domain Randomization Schemes

For each task, we have three levels of domain randomization with increasing difficulty. D0 randomizes task-relevant objects on a furniture with position and orientation ranges of ± 15 cm and ± 15 degrees, respectively. D1 randomizes task-relevant objects on the original furniture without constraints. For example, the position of the cup is randomly sampled on the entire kitchen island and the orientation is randomly sampled between $[-\pi, \pi]$ (Figure 4 (a)). D2 performs D1 randomization for task-relevant objects and introduces additional objects on furniture (obstacles for manipulation) and objects on the floor (obstacles for navigation) as illustrated in Figure 1 (upper right) . Note that this randomization scheme is much more aggressive than previous methods [1–3], due to MoMaGen 's capability to generate novel robot base motions. More detailed visualizations are in Appendix.

5.3 Data Generation Comparison

In this section, we evaluate the data generation performance of MoMaGen for bimanual mobile manipulation tasks under different randomization ranges. We compare MoMaGen with two data generation baselines: (1) SkillMimicGen [2], which generates collision-free trajectories for single-arm manipulation tasks using motion planners and task-space control for contact-rich motion, and (2) DexMimicGen [3], which focuses on dexterous bimanual manipulation data generation. Since all evaluated tasks involve substantial mobile base movement, we additionally extend SkillMimicGen and DexMimicGen to support mobile manipulation by incorporating base trajectory replay from the source demo, following a similar approach to MimicGen [22]. We compare different data generation methods using three categories of metrics: (1) **data diversity**, measured in terms of object pose variation, and robot action diversity; (2) **data generation success rate**, which reflects the method's ability to handle complex bimanual mobile manipulation tasks; and (3) **object visibility ratio**, quantifying how often target objects remain visible during robot navigation.

How diverse are the demonstrations generated by MoMaGen? Figure 4 (a) shows the variation in task-relevant object poses in the data generated by MoMaGen for D0 and D1 and SkillMimicGen for D0. The baseline can hardly generate trajectories for D1 due to their lack of ability to generate novel base motions. MoMaGen can generate data for much larger object pose coverage as compared to the baselines (where the object is clustered on the top-right of the table). Furthermore, data generated by MoMaGen has much more diverse scenes with novel obstacles and distractor objects (see Appendix). We also evaluate the diversity in the generated actions. Figure 4 (b) and (c) show that MoMaGen D1 has much larger coverage of base and end-effector actions as compared to MoMaGen D0 and the baseline. In Figure 4 (d), we visualize the PCA 2D projections of the robot arm and torso joints in the data generated by the two baselines and MoMaGen , and confirm that MoMaGen has much larger action coverage.

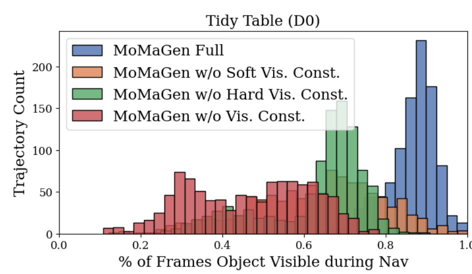


Figure 5: Object visibility analysis for MoMaGen and ablations. The x-axis is the % of frames where the target object is visible during navigation, and the y-axis is the trajectory count (out of 1000). MoMaGen siginificantly outperforms ablations thanks to both hard and soft visibility constraints.

Can MoMaGen achieve high-throughput data generation? Table 2 shows MoMaGen achieves an average data generation success rate of 63% for D0. Our method can generate data for all tasks, at all levels of randomization, although we note that the data throughput decreases as the randomization range increases. While the baselines perform well on simpler tasks like Pick Cup, the success rate drops for harder tasks like Clean Frying Pan due to a stronger need for adapting base motions, and cannot handle D1 and D2 randomization at all.

Can MoMaGen generate demonstrations with high object visibility? Object visibility plays a critical role in visuomotor policy learning (Sec. 5.4), motivating our evaluation of visibility ratios

	Methods	Pick Cup	Tidy Table	Put Dishes Away	Clean Frying Pan
D0	MoMaGen	0.86	**0.80**	0.38	**0.51**
	SkillMimicGen	**1.00**	0.69	0.38	0.40
	DexMimicGen	**1.00**	0.72	0.38	0.35
	MoMaGen w/o soft vis. const.	0.88	0.78	**0.50**	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	**0.64**	**0.34**	**0.20**
	MoMaGen w/o vis. const.	**0.66**	0.48	0.23	0.13
D2	MoMaGen	0.47	**0.22**	**0.07**	**0.16**
	MoMaGen w/o vis. const.	**0.50**	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.

	Methods	Pick Cup	Tidy Table	Put Dishes Away	Clean Frying Pan
D0	MoMaGen	**1.00**	**0.86**	**0.79**	**0.69**
	SkillMimicGen	**1.00**	0.40	0.71	0.65
	DexMimicGen	**1.00**	0.39	0.71	0.67
	MoMaGen w/o soft vis. const.	**1.00**	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	**0.93**	**0.89**	**0.78**	**0.80**
	MoMaGen w/o vis. const.	0.71	0.46	0.40	0.43
D2	MoMaGen	**0.94**	**0.79**	**0.75**	**0.81**
	MoMaGen w/o vis. const.	0.73	0.48	0.40	0.44

Table 3: 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.

presented in Table 3 and Figure 5. We compare the object visibility ratios of MoMaGen against baselines and three ablations that systematically ablate the hard and soft constraints introduced in MoMaGen . MoMaGen achieves significantly higher task-relevant object visibility (oftentimes doubled) as compared to the baselines and the ablations. Figure 5 compares task-relevant object visibility of MoMaGen to the ablations for the Tidy Table task, and shows that both hard and soft visibility constraints are crucial for high object visibility in the generated data.

5.4 Policy Learning with Generated Demonstrations

Although MoMaGen can synthesize successful trajectories to solve the tasks, it assumes privileged information such as ground truth object pose and geometry. We still need to train visuomotor policies from onboard sensor inputs (e.g., RGB images). In this section, we investigate whether the demos generated by MoMaGen help train imitation learning-based policies compared with other data generation methods, show how data with different object visibility ratios influences policy training, and whether MoMaGen generated data benefits different types of imitation learning methods.

Policy Learning Setup. We experiment with two imitation learning based methods, WB-VIMA [17] and π_0 [7]. 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 state. For WB-VIMA, we further fuse and post-process the three RGB images (with grountruth 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 π_0, we finetune a pre-trained π_0 model with a LoRA rank of 32. More implementation details are in the Appendix.

How do different data generation methods impact policy performance? We generate 1000 successful demonstrations using MoMaGen for Pick Cup (D0), Pick Cup (D1), and Tidy Table (D0). For comparison, we also generate 1000 demonstrations using SkillMimicGen and DexMimicGen with replayed navigation for Pick Cup (D0) and Tidy Table (D0). WB-VIMA policies are trained on data from MoMaGen and the baselines. As shown in Figure 6 (a), for Pick Cup (D0) with a small randomization range (0.3m × 0.3m), MoMaGen performs on par with the baselines, likely because learning the replayed navigation is sufficient for D0 object coverage. However, for Tidy Table (D0), MoMaGen clearly outperforms the baselines. The gap comes from overfitting long, nonsmooth navigation trajectories replayed from a single human demonstration. For the more challenging Pick Cup (D1) task (1.3m × 0.8m randomization range), only MoMaGen enables WB-VIMA to achieve a 0.25 success rate; baselines trained on D0 data fail entirely (Figure 6 (b)). The diverse base motions in MoMaGen (D0) data further support intermediate successes like touching the cup.

How does object visibility ratio affect policy performance? In MoMaGen, the eye camera is constrained to focus on the object of interest, aiding visual servoing during navigation and improving object visibility during manipulation (see Section 5.3). We investigate whether these visibility constraints, which increase the proportion of time the object remains in view, enhance imitation learning performance. We conduct an ablation study using WB-VIMA on Pick Cup (D0) and Tidy Table (D0), comparing the full MoMaGen with three variants: (1) without soft visibility constraints (camera not encouraged to look at the task-relevant object during navigation); (2) without hard visibility constraints (camera not enforced to look at the task-relevant object during manipulation); (3) without any visibility constraints. As shown in Figure 6 (d), for Pick Cup (D0), ablated variants achieve success rates between 0.45 and 0.65, below the 0.75 achieved by MoMaGen. For Tidy Table (D0), the gap is larger: ablations peak at 0.05, while MoMaGen reaches 0.40. These results suggest that enforcing visibility constraints during data generation significantly improves policy performance, particularly when the policy depends on short history inputs.

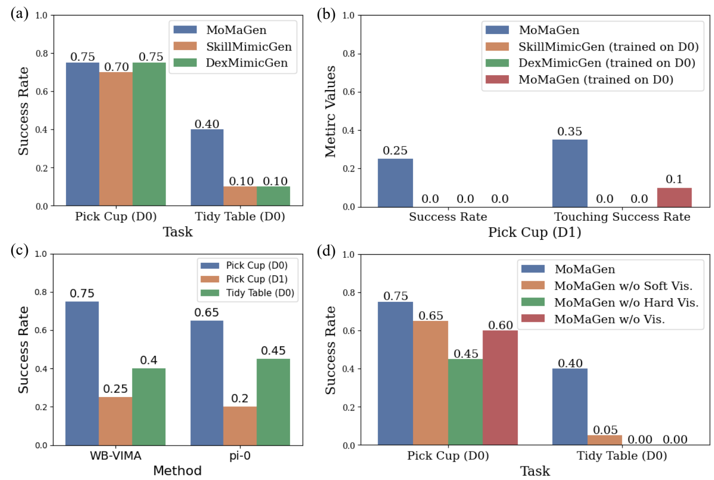


Figure 6: Comparison between MoMaGen and other data generation methods on WB-VIMA's performances in (a) and (b), performances of WB-VIMA and π_0 trained with MoMaGen data in (c) and visibility ablations in (d). The success rate is averaged over 20 unseen evaluation episodes. Policies trained on MoMaGen data consistently perform better than those trained on others' data.

Does MoMaGen generated data benefit various imitation learning methods? We fine-tune π_0 on MoMaGen data for Pick Cup (D0), Pick Cup (D1), and Tidy Table (D0) with 1000 generated demonstrations. Figure 6 (c) shows that the fine-tuned π_0 achieves success rates that are comparable to WB-VIMA across all three tasks. These results demonstrate that MoMaGen generated data effectively enhance the performance of diverse imitation learning methods.

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 visibilty 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.

References

[1] Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations. In Jie Tan, Marc Toussaint, and Kourosh Darvish, editors, *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pages 1820–1864. PMLR, 06–09 Nov 2023.

[2] Caelan Reed Garrett, Ajay Mandlekar, Bowen Wen, and Dieter Fox. Skillmimicgen: Automated demonstration generation for efficient skill learning and deployment. In *8th Annual Conference on Robot Learning*, 2024.

[3] 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. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

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

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

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

[7] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. pi_0 : A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

[8] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. pi_0.5: A vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

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

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

[11] Misha Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel, and Aravind Srinivas. Reinforcement learning with augmented data. *Advances in neural information processing systems*, 33:19884–19895, 2020.

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

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

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

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

[16] 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*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

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

[18] Chengshu Li, Fei Xia, Roberto Martin-Martin, and Silvio Savarese. Hrl4in: Hierarchical reinforcement learning for interactive navigation with mobile manipulators. In *Conference on Robot Learning*, pages 603–616. PMLR, 2020.

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

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

[21] Yasuhiro Ishiguro, Tasuku Makabe, Yuya Nagamatsu, Yuta Kojio, Kunio Kojima, Fumihito Sugai, Yohei Kakiuchi, Kei Okada, and Masayuki Inaba. Bilateral humanoid teleoperation system using whole-body exoskeleton cockpit tablis. *IEEE Robotics and Automation Letters*, 5(4):6419–6426, 2020.

[22] 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.

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

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

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

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

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

[28] 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.

[29] 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.

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

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

[32] Murtaza Dalal, Ajay Mandlekar, Caelan Reed Garrett, Ankur Handa, Ruslan Salakhutdinov, and Dieter Fox. Imitating task and motion planning with visuomotor transformers. In *Conference on Robot Learning*, pages 2565–2593. PMLR, 2023.

[33] Josiah Wong, Albert Tung, Andrey Kurenkov, Ajay Mandlekar, Li Fei-Fei, Silvio Savarese, and Roberto Martín-Martín. Error-aware imitation learning from teleoperation data for mobile manipulation. In *Conference on Robot Learning*, pages 1367–1378. PMLR, 2022.

[34] Kenneth Shaw, Yulong Li, Jiahui Yang, Mohan Kumar Srirama, Ray Liu, Haoyu Xiong, Russell Mendonca, and Deepak Pathak. Bimanual dexterity for complex tasks. In *8th Annual Conference on Robot Learning*, 2024.

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

[36] Zipeng Fu, Qingqing Zhao, Qi Wu, Gordon Wetzstein, and Chelsea Finn. Humanplus: Humanoid shadowing and imitation from humans. *arXiv preprint arXiv: 2406.10454*, 2024.

[37] Xuxin Cheng, Jialong Li, Shiqi Yang, Ge Yang, and Xiaolong Wang. Open-television: Teleoperation with immersive active visual feedback. In *8th Annual Conference on Robot Learning*, 2024.

[38] Jimmy Wu, William Chong, Robert Holmberg, Aaditya Prasad, Yihuai Gao, Oussama Khatib, Shuran Song, Szymon Rusinkiewicz, and Jeannette Bohg. Tidybot++: An open-source holonomic mobile manipulator for robot learning. *arXiv preprint arXiv: 2412.10447*, 2024.

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

[40] Jinhan Li, Yifeng Zhu, Yuqi Xie, Zhenyu Jiang, Mingyo Seo, Georgios Pavlakos, and Yuke Zhu. OKAMI: Teaching humanoid robots manipulation skills through single video imitation. In *8th Annual Conference on Robot Learning*, 2024.

[41] Yanjie Ze, Zixuan Chen, Wenhao Wang, Tianyi Chen, Xialin He, Ying Yuan, Xue Bin Peng, and Jiajun Wu. Generalizable humanoid manipulation with improved 3d diffusion policies. *arXiv preprint arXiv:2410.10803*, 2024.

[42] Tairan He, Zhengyi Luo, Xialin He, Wenli Xiao, Chong Zhang, Weinan Zhang, Kris M. Kitani, Changliu Liu, and Guanya Shi. Omnih2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning. In *8th Annual Conference on Robot Learning*, 2024.

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

[44] Jimmy Wu, Rika Antonova, Adam Kan, Marion Lepert, Andy Zeng, Shuran Song, Jeannette Bohg, Szymon Rusinkiewicz, and Thomas Funkhouser. Tidybot: personalized robot assistance with large language models. *Autonomous Robots*, 47:1087–1102, 2023.

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

[46] Austin Stone, Ted Xiao, Yao Lu, Keerthana Gopalakrishnan, Kuang-Huei Lee, Quan Vuong, Paul Wohlhart, Sean Kirmani, Brianna Zitkovich, Fei Xia, Chelsea Finn, and Karol Hausman. Open-world object manipulation using pre-trained vision-language models. In *7th Annual Conference on Robot Learning*, 2023.

[47] Zhenyu Jiang, Yuqi Xie, Jinhan Li, Ye Yuan, Yifeng Zhu, and Yuke Zhu. Harmon: Whole-body motion generation of humanoid robots from language descriptions. In *8th Annual Conference on Robot Learning*, 2024.

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

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

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

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

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

[53] 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*, pages 80–93. PMLR, 2023.

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

[55] 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, Manoj Kumar, Keran Rong, Julian Eisenschlos, Rishabh Kabra, Matthias Bauer, Matko Bošnjak, Xi Chen, Matthias Minderer, Paul Voigtlaender, Ioana Bica, Ivana Balazevic, Joan Puigcerver, Pinelopi Papalampidi, Olivier Henaff, Xi Xiong, Radu Soricut, Jeremiah Harmsen, and Xiaohua Zhai. Paligemma: A versatile 3b vlm for transfer, 2024.

[56] Yunfan Jiang, Chen Wang, Ruohan Zhang, Jiajun Wu, and Li Fei-Fei. Transic: Sim-to-real policy transfer by learning from online correction. *arXiv preprint arXiv: 2405.10315*, 2024.

A Additional Data Generation Details

A.1 Visualizations of Domain Randomization Schemes

In Figure 7, we visualize the domain randomization schemes (D0 / D1 / D2) across all four tasks. Thanks to the robot base sampling mechanism, MoMaGen can generate successful demonstrations across significantly more aggressive domain randomization than prior works (D1). With motion planners in-the-loop, it can also generate diverse robot trajectories while avoiding obstacles for both the base and the arm (D2).

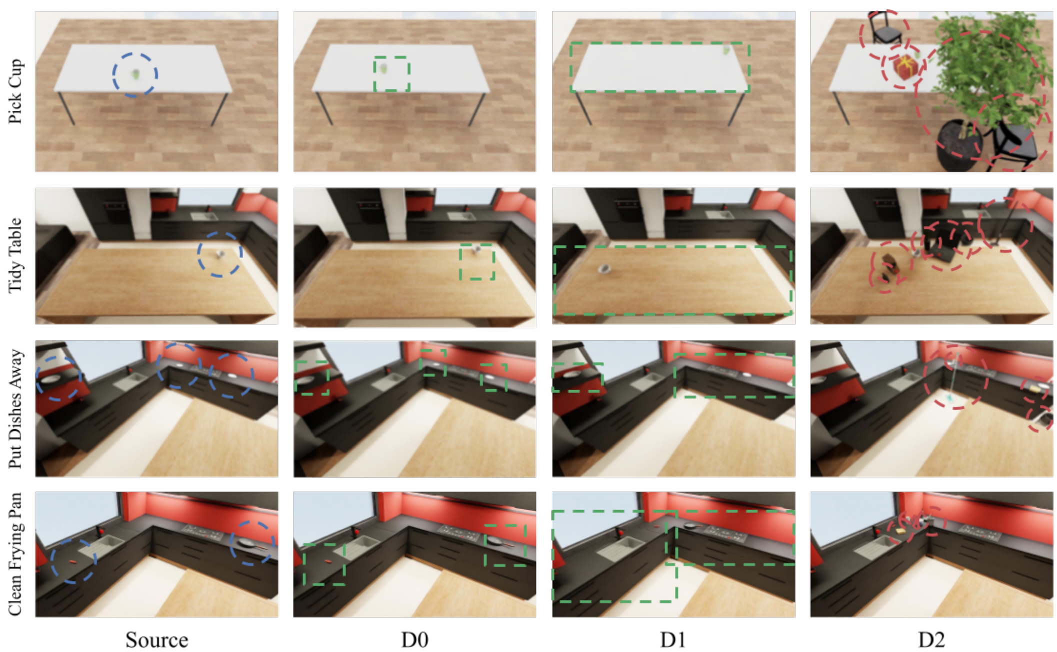
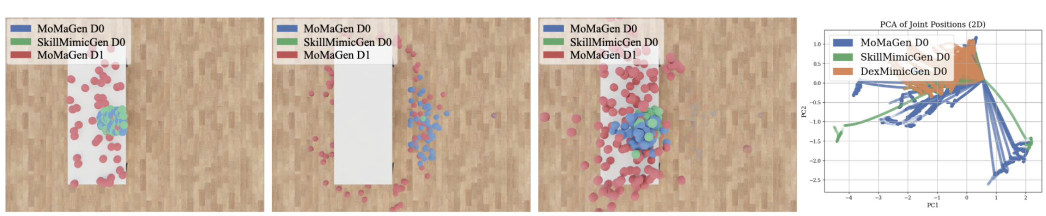


Figure 7: Visualization of Domain Randomization. Blue represents the task-relevant objects. Green represents the randomization range for these objects. Red represents the obstacles/distractor objects.

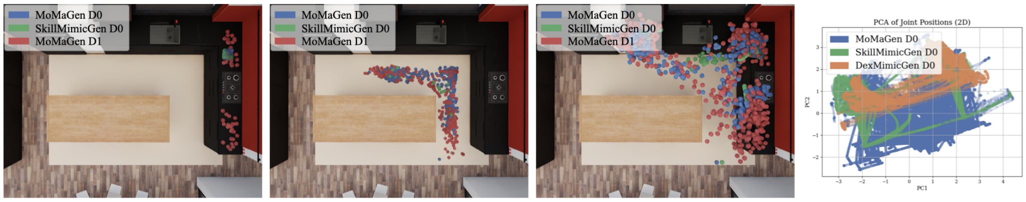
A.2 Visualizations of Data Diversity

Similar to Figure 4, we also include the visualization of data diversity for the other three tasks in Figure 8, 9, and 10, comparing MoMaGen to baselines (SkillMimicGen and DexMimicGen). As we expect, the object diversity of D1 (shown in subfigures (a)) induces significantly more diverse robot base and end-effector trajectories (shown in subfigures (b) and (c)), as well as joint positions. This leads to better state space and action space coverage in both task space and joint space.



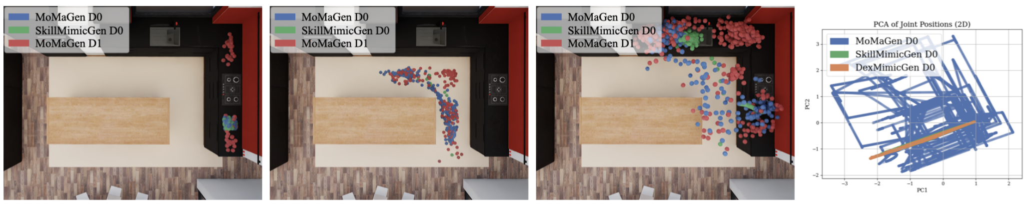
(a) Object Diversity (b) Robot Base Diversity (c) Robot End-Effector Diversity (d) Robot Joint Position Diversity

Figure 8: Generated data diversity analysis for Pick Cup task (50 trajectories, subsampled).



(a) Object Diversity (b) Robot Base Diversity (c) Robot End-Effector Diversity (d) Robot Joint Position Diversity

Figure 9: Generated data diversity analysis for Put Dishes Away task (50 trajectories, subsampled).



(a) Object Diversity (b) Robot Base Diversity (c) Robot End-Effector Diversity (d) Robot Joint Position Diversity

Figure 10: Generated data diversity analysis for Clean Frying Pan task (50 trajectories, subsampled).

A.3 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.

B Additional Policy Training Details

In this section, we first describe the data cleaning process in Section B.1 and then provide additional training details for WB-VIMA (Section B.2) and π_0 (Section B.3). For both methods, 90% of the demonstrations are used for training and the remaining 10% for validation.

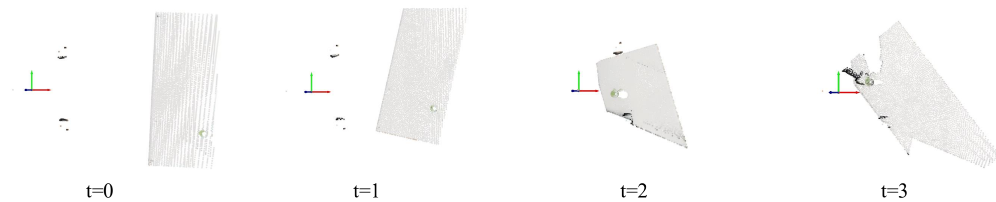
B.1 Data Cleaning

When teleoperating in simulation, it is difficult for human operators to accurately perceive depth, often leading to hesitation to better align the robot gripper, especially just before grasping or making contact with objects. As a result, the collected human demonstrations may contain short segments where the gripper remains nearly stationary. These "frozen" segments can negatively affect training, particularly for imitation learning methods with limited temporal context (e.g., WB-VIMA uses a 2-step history, and π_0 only uses the current state).

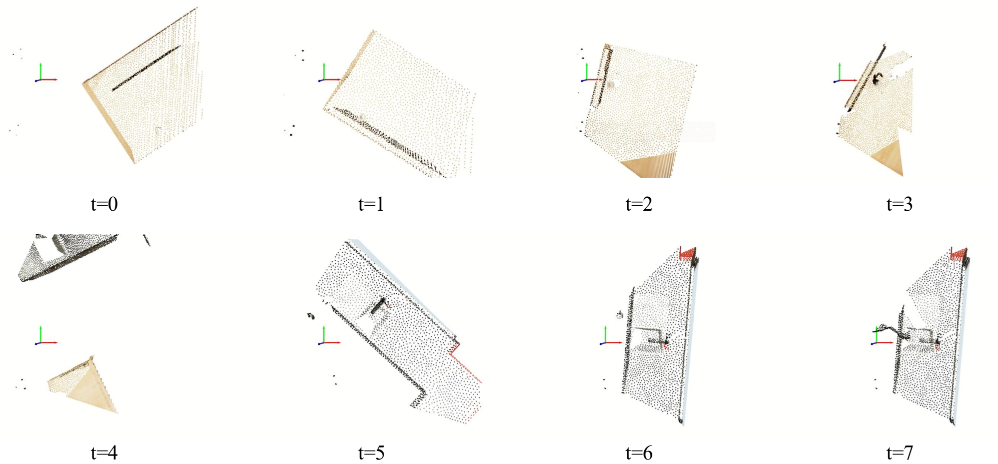
To mitigate this, we add a data preprocessing step to clean the hesitation segments. For a trajectory of length T, if at any timestep $i \in [0, T-5]$, the absolute difference in joint positions between step i and step $i+5$ is smaller than a threshold (1e-3) across all dimensions, we treat the segment from i to $i+5$ as frozen and remove it prior to policy training.

B.2 WB-VIMA Training Details

Policy Architecture. WB-VIMA [17] takes as input both proprioceptive observations and an egocentric colored point cloud. The proprioceptive inputs include the mobile base velocity $v^{\text{base}} \in \mathbb{R}^3$, torso joint position $q^{\text{torso}} \in \mathbb{R}^4$, left arm joint position $q^{\text{left}} \in \mathbb{R}^6$, left gripper width $q^{\text{grip-left}} \in \mathbb{R}^1$ as well as right arm position and gripper width $q^{\text{right}} \in \mathbb{R}^6, q^{\text{grip-right}} \in \mathbb{R}^1$. The point cloud is constructed by fusing RGB-D images from three cameras mounted on the robot: one eye-level camera and two wrist-mounted cameras. Examples of point clouds for Pick Cup and Tidy Table are



(a) Pick Cup point cloud visualization



(b) Tidy Table point cloud visualization

Figure 11: Visualization of ego-centric point cloud for Pick Cup and Tidy Table.

shown in Figure 11. It is then cropped using a robot-centric bounding box and downsampled to 4096 points using farthest point sampling. WB-VIMA takes a history input with 2 steps. Additional policy architecture details can be found in the original paper [17]. For each task, we train WB-VIMA from scratch, resulting in a single-task policy.

Hyperparameters. The training hyperparameters, covering the PointNet for point cloud processing, the diffusion head, transformer backbone, learning rates, and task-specific point cloud clipping ranges, are listed in Table 4. Note that the ego-centric point cloud clipping ranges are customized for each task to account for differences in scene layouts and the randomization range of the target objects. These hyperparameters are selected via grid search.

B.3 π_0 Training Details

Policy Architecture. For π_0, we finetune a pre-trained π_0 model with a lora rank of 32 for 50k steps with a batch size of 64. The model takes the current eye-level camera RGB image and the two wrist-mounted camera RGB images, the proprioceptive inputs as WB-VIMA, and outputs the target robot joint state for the next 50 time-steps.

Hyperparameters. All the RGB images are resized to the size of 224x224. Actions and proprioceptive inputs are normalized using the 1st and 99th quantile. Since π_0 takes an action dimension of 32, the actions and proprioceptive inputs are zero-padded to 32. The model consists of a PaliGemma VLM [55] backbone and a 300M action expert. More details on model architecture can be found in [7]. More training hyper-parameters are shown in Table 5.

Hyperparameter	Value
Number of points in point cloud	4096
PointNet hidden dim	256
PointNet hidden depth	2
PointNet output dim	256
PointNet activation	GELU
Proprioceptive MLP input dim	21
Proprioceptive MLP hidden dim	2
Proprioceptive MLP hidden depth	256
Proprioceptive MLP output dim	256
Proprioceptive MLP activation	ReLU
Transformer embedding size	512
Transformer layers	4
Transformer heads	8
Transformer dropout rate	0.1
Transformer activation	GEGLU
Action dim	21
Unet down dims	[128, 256]
Unet kernel size	5
Unet number of groups	8
Diffusion step embedding dim	256
Diffusion noise scheduler	DDIM
Number of training steps	100
Beta schedule	squaredcos_cap_v2
Number of denoise steps per inference	16
Learning rate	1e-4
Learning rate scheduler	Cosine decay
Learning rate warmup steps	100000
Learning rate cosine steps	1300000
Optimizer	AdamW
Batch size per GPU	128
Number of GPUs in parallel	2
Pick Cup (D0) pointcloud clip range	x: [0.0, 2.3], y: [-0.5, 0.5], z: [0.7, 2.0]
Pick Cup (D1) pointcloud clip range	x: [0.0, 2.7], y: [-1.0, 1.0], z: [0.7, 2.0]
Tidy Table (D0) pointcloud clip range	x: [0.0, 2.3], y: [-1.5, 1.5], z: [0.7, 1.5]
Model size	37.1M

Table 4: Hyperparameters for WB-VIMA.

B.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, π_0 can be trained on four H200 GPUs, taking approximately 7 hours to reach 50k steps.

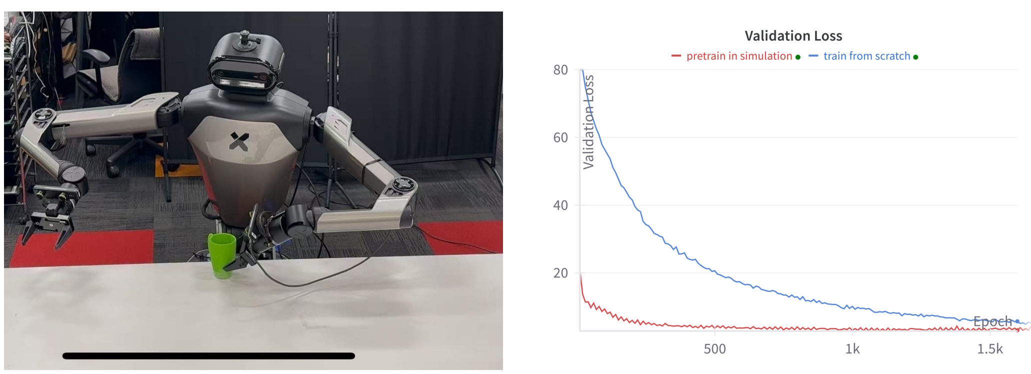
C Additional Experimental Results

C.1 Sim-to-real

We aim to investigate whether MoMaGen can support real-world deployment. For our experiments, we use the Galaxea R1 mobile manipulator, which is the same robot used for data collection and data generation in simulation. To obtain point clouds, we use R1's onboard eye camera (a ZED 2 stereo camera) and an additional ZED Mini camera mounted above the left gripper. Due to noise in depth estimation and point cloud reconstruction, we employ the built-in AI model to enhance the

Hyperparameter	Value
Proprioceptive MLP input dim	32
Proprioceptive MLP output dim	1024
Flow Matching MLP input dim	32
Flow Matching MLP hidden dim	2048
Flow Matching MLP hidden depth	2
Flow Matching MLP output dim	1024
Flow Matching MLP activation	swish
PaliGemma embedding size	2048
PaliGemma number of layers	18
PaliGemma number of heads	18
PaliGemma heads dimension	256
PaliGemma MLP dimension	16384
Action expert embedding size	1024
Action expert MLP dimension	4096
Number of flow matching steps per inference	10
Learning rate	2.5e-5
Learning rate scheduler	Cosine decay
Learning rate warmup steps	1000
Learning rate cosine steps	30000
Optimizer	AdamW
Batch size	64
Number of GPUs in parallel	4
Model size	3.3B

Table 5: Hyperparameters for π_0.



(a) Real World Setup

(b) Validation Loss

Figure 12: Real world setup and the validation loss curve.

point cloud quality. We test on the Pick Cup (D0) task, using a table of similar height to the one in simulation and a 3D-printed green cup. Similar to point cloud processing in simulation, we also use farthest point sampling to downsample the point clouds to 4096 points. The real-world setup is shown in Figure 12 (a).

Zero-shot transfer to the real world is notoriously challenging due to domain gaps, especially for vision-based policies [56]. In our case, significant differences exist between simulated and real RGB images, and the real-world depth estimates are inherently noisy. To address this, we collect 50 real-world demonstrations for the Pick Cup (D0) task and evaluate whether pretraining in simulation improves real-world learning. We compare two training setups: one initializes from a simulation-pretrained model checkpoint (trained for 1.8M steps), and the other trains from scratch. Both models are fine-tuned for 35k steps using 40 real demonstrations, with the remaining 10 used for validation. The results show that the pretrained model achieves a much faster drop in validation loss. The validation loss curve is in Figure 12 (b). After 35k steps, the pretrained model reaches a validation loss of approximately 3.0, while the model trained from scratch remains around 6.0. Additional rollout videos for both the simulation-pretrained policy and the policy trained from scratch are available on the project website: `https://momagen-rss.github.io/`.