DemoGen: Synthetic Demonstration Generation for Data-Efficient Visuomotor Policy Learning

Zhengrong Xue[123*], Shuying Deng[1*], Zhenyang Chen[2], Yixuan Wang[1], Zhecheng Yuan[123], Huazhe Xu[123]
[1]Tsinghua University, [2]Shanghai Qi Zhi Institute, [3]Shanghai AI Lab, [*]Equal contribution

demo-generation.github.io

***Abstract*—Visuomotor policies have shown great promise in robotic manipulation but often require substantial human-collected data for effective performance. A key factor driving the high data demands is their limited spatial generalization capability, which necessitates extensive data collection across different object configurations. In this work, we present *DemoGen*, a low-cost, fully synthetic approach for automatic demonstration generation. Using only one human-collected demonstration per task, *DemoGen* generates spatially augmented demonstrations by adapting the demonstrated action trajectory to novel object configurations. Visual observations are synthesized by leveraging 3D point clouds as the modality and rearranging the subjects in the scene via 3D editing. Empirically, *DemoGen* significantly enhances policy performance across a diverse range of real-world manipulation tasks, showing its applicability even in challenging scenarios involving deformable objects, dexterous hand end-effectors, and bimanual platforms. Furthermore, *DemoGen* can be extended to enable additional out-of-distribution capabilities, including disturbance resistance and obstacle avoidance.**

I. Introduction

Visuomotor policy learning has demonstrated remarkable competence for robotic manipulation tasks [5, 59, 14, 57], yet it typically demands large volumes of human-collected data. State-of-the-art approaches often require tens to hundreds of demonstrations to achieve moderate success on complex tasks, such as spreading sauce on pizza [5] or making rollups with a dexterous hand [57]. More intricate, long-horizon tasks may necessitate thousands of demonstrations [60].

One key factor contributing to the data-intensive nature of these methods is their limited **spatial generalization** [41, 44, 39, 16] ability. Our empirical study suggests that visuomotor policies [5], even when coupled with pre-trained or 3D visual encoders [32, 38, 33, 57], exhibit limited spatial capacity, typically confined to regions adjacent to the demonstrated object configurations. Such limitation necessitates repeated data collection with repositioned objects until the demonstrated configurations sufficiently cover the full tabletop workspace. This creates a paradox: while the critical actions enabling dexterous manipulation are concentrated in a small subset of contact-rich segments, a substantial portion of human effort is spent teaching robots to approach objects in free space.

A potential solution to reduce redundant human effort is to replace the tedious relocate-and-recollect procedure with automatic demonstration generation. Recent advances such as MimicGen [31] and its subsequent extensions [19, 17, 21] have proposed to generate demonstrations by segmenting the demonstrated trajectories based on object interactions. These object-centric segments are then transformed and interpolated into execution plans that fit desired spatially augmented object configurations. The resulting plans are then executed through open-loop rollouts on the robot, termed *on-robot rollouts*, to verify their correctness and simultaneously capture the visual observations needed for policy training.

Despite their success in simulation, applying MimicGen-style strategies to real-world environments is hindered by the high costs of on-robot rollouts, which are nearly as expensive as collecting raw demonstrations. An alternative is to deploy via sim-to-real transfer [35, 45, 55], though bridging the sim-to-real gap remains a significant challenge in robotics.

In this work, we introduce *DemoGen*, a data generation system that can be seamlessly plugged into the policy learning workflow in both simulated and physical worlds. Recognizing the high cost of on-robot rollouts represents a major barrier to practical deployment, *DemoGen* adopts a **fully synthetic** pipeline that efficiently concretizes the generated plans into spatially augmented demonstrations ready for policy training.

For action generation, *DemoGen* develops the MimicGen strategy by incorporating techniques from Task and Motion Planning (TAMP) [8, 3, 30], similar to the practice in the recently released SkillMimicGen [17]. For visual observations, *DemoGen* selects point clouds as the modality and synthesizes the augmented visual observations through 3D editing. The key insight is that point clouds, which inherently live in the 3D space, can be easily manipulated to reflect the desired spatial augmentations. Generating augmented point cloud observations is reduced to identifying clusters of points corresponding to the objects or robot end-effectors and then applying the same spatial transformations used in the generated action plans. Notably, this strategy also applies to contact-rich skill segments, as parts in contact are treated as cohesive clusters that undergo uniform transformations. Furthermore, the artificially applied transformations on point clouds accurately reflect the underlying physical processes, thereby minimizing the visual gap between real and synthetic observations.

Empirically, we manifest the effectiveness of *DemoGen* by evaluating the performance of visuomotor policies trained on *DemoGen*-generated datasets from **only one** human collected demonstration per task. To assess the impact of *DemoGen* on spatial generalization, we adhere to a rigorous evaluation protocol in which the objects are placed across the entire tabletop workspace within the end-effectors' reach.

We conduct extensive real-world experiments, showing that *DemoGen* can be successfully deployed on both single-arm and bi-manual platforms, using parallel-gripper and dexterous-hand end-effectors, from both third-person and egocentric observation viewpoints, and with a range of rigid-body and deformable/fluid objects. Meanwhile, the cost of generating one demonstration trajectory with *DemoGen* is merely **0.01** seconds of computation. With such minimal cost, *DemoGen* significantly enhances policy performance, generalizing to un-demonstrated configurations and achieving an average of **74.6**% across **8** real-world tasks. Additionally, we demonstrate that simple extensions under the *DemoGen* framework can further equip imitation learning with acquired out-of-distribution (O.O.D.) generalization capabilities such as disturbance resistance and obstacle avoidance.

Please refer to the supplementary material for robot videos.

II. *DemoGen* Methods

A. Problem Formulation

A visuomotor policy $\pi : \mathcal{O} \mapsto \mathcal{A}$ directly maps the visual observations $o \in \mathcal{O}$ to the predicted actions $a \in \mathcal{A}$. To train such a policy, a dataset $\mathcal{D}$ of demonstrations must be prepared. We define a source demonstration $D_{s_0} \subseteq \mathcal{D}$ as a trajectory of paired observations and actions conditioned on an initial object configuration: $D_{s_0} = (d_0, d_1, \ldots, d_{L-1} | s_0)$, where each $d_t = (o_t, a_t)$ represents an observation-action pair, s_0 denotes the initial configuration, and L is the trajectory length. *DemoGen* is designed to augment a human-collected source demonstration by generating a new demonstration conditioned on a different initial object configuration:

$$\hat{D}_{s_0'} = (\hat{d}_0, \hat{d}_1, \ldots, \hat{d}_{L-1} | s_0').$$

Specifically, assuming the task involves the sequential manipulation of K objects $\{O_1, O_2, \ldots, O_K\}$, the initial object configuration s_0 is defined as the set of initial poses of these objects: $s_0 = \{\mathbf{T}_0^{O_1}, \mathbf{T}_0^{O_2}, \ldots, \mathbf{T}_0^{O_K}\}$, where $\mathbf{T}_t^O$ denotes the $\mathrm{SE}(3)$ transformation from the world frame to an object O at time step t. The action a_t consists of the robot arm and robot hand commands, represented as $a_t = (a_t^{\text{arm}}, a_t^{\text{hand}})$, where $a_t^{\text{arm}} \triangleq \mathbf{A}_t^{\text{EE}}$ is the target $\mathrm{SE}(3)$ end-effector pose in the world frame, and a_t^{hand} can either be a binary signal for a parallel gripper's open/close action or a higher-dimensional vector for controlling the joints of a dexterous hand. The observation o_t includes both the point cloud data and the proprioceptive feedback from the robot: $o_t = (o_t^{\text{pcd}}, o_t^{\text{arm}}, o_t^{\text{hand}})$, where o_t^{arm} and o_t^{hand} reflect the current state of the end-effector, with the same dimensionality as the corresponding actions.

B. Pre-processing the Source Demonstration

Segmented point cloud observations. To improve the practical applicability in real-world scenarios, we utilize a single-view RGBD camera for point cloud acquisition. The raw point cloud observations are first preprocessed by cropping the redundant points from the background and table surface. We assume the retained points are associated with either the manipulated object(s) or the robot's end-effector. A clustering operation [12] is then applied to filter out the outlier points in noisy real-world observations. Subsequently, the point cloud is downsampled to a fixed number of points (e.g., 512 or 1024) using farthest point sampling to facilitate policy learning [37].

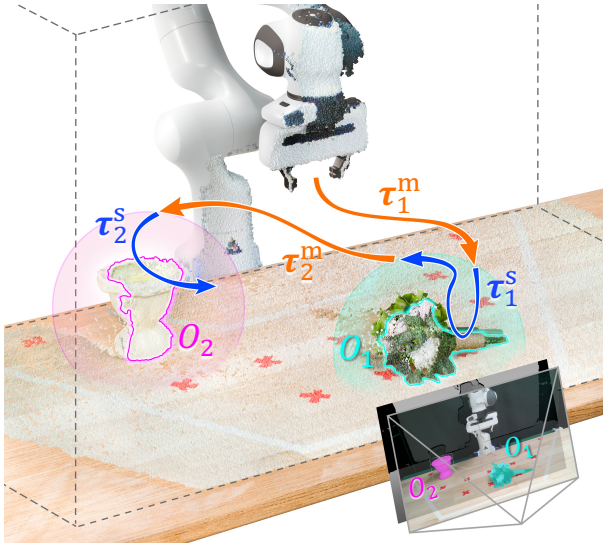

Fig. 1: **Pre-processing the source demonstration.** The raw point cloud observations are processed by cropping, clustering, and down-sampling. The source action trajectory is parsed into motion and skill segments by referring to the semantic masks of manipulated objects.

For the first frame of the trajectory, we employ Grounded SAM [40] to obtain the segmentation masks for the manipulated objects from the RGB image. These masks are then applied to the pixel-aligned depth image and projected onto the 3D point cloud, as shown in Fig. 1.

Parsing the source trajectory. Following previous work [31, 17], we assume that the execution trajectory can be parsed into a sequence of object-centric segments. Noticing that the robot must initially approach the object in free space before engaging in on-object manipulation through contact, each object-centric segment can be further subdivided into two stages: motion and skill. For example, in the task illustrated in Fig. 1, the trajectory is divided into four stages: 1) *move* to the flower, 2) *pick* up the flower, 3) *transfer* the flower to the vase, and 4) *insert* the flower into the vase.

We can easily identify the skill segments associated with a given object by checking whether the distance between the geometric center of the object's point cloud and the robot's end-effector falls within a predefined threshold, as illustrated by the spheres in Fig. 1. The intermediate trajectories between two skill segments are classified as motion segments.

Formally, we represent an interval of time stamps as $\boldsymbol{\tau}$:

$$\boldsymbol{\tau} = (t_{\text{start}},\ t_{\text{start}}+1,\ \ldots,\ t_{\text{end}}-1,\ t_{\text{end}}) \subseteq (0, 1, \ldots, L-1),$$

which can be used as an *index sequence* for the extraction of the corresponding segments from a sequence of demonstrations, actions, or observations. For instance, $d[\boldsymbol{\tau}] = (d_{t_{\text{start}}}, d_{t_{\text{start}}+1}, \ldots, d_{t_{\text{end}}-1}, d_{t_{\text{end}}})$ represents the extracted

subset of source demonstration indexed by $\boldsymbol{\tau}$. Using this notation, we parse the source demonstration into alternating motion and skill segments according to the index sequence $(\boldsymbol{\tau}_1^{\mathrm{m}}, \boldsymbol{\tau}_1^{\mathrm{s}}, \ldots, \boldsymbol{\tau}_K^{\mathrm{m}}, \boldsymbol{\tau}_K^{\mathrm{s}})$:

$$D_{s_0} = (d[\boldsymbol{\tau}_1^{\mathrm{m}}], d[\boldsymbol{\tau}_1^{\mathrm{s}}], \ldots, d[\boldsymbol{\tau}_K^{\mathrm{m}}], d[\boldsymbol{\tau}_K^{\mathrm{s}}] | s_0).$$

C. TAMP-based Action Generation

Adapting actions to the new configuration. The generation process begins by selecting a target initial configuration $s_0' = \{\mathbf{T}_0^{O_1\prime}, \mathbf{T}_0^{O_2\prime}, ..., \mathbf{T}_0^{O_K\prime}\}$. Under the 4×4 homogeneous matrix representation, the spatial transformation between the target and source configurations is computed as:

$$\Delta s_0 = \{(\mathbf{T}_0^{O_1})^{-1} \cdot \mathbf{T}_0^{O_1\prime}, \ldots, (\mathbf{T}_0^{O_K})^{-1} \cdot \mathbf{T}_0^{O_K\prime}\}.$$

Recall that the actions consist of both robot arm and robot hand commands. The robot hand commands define the interactive actions *on* the object, e.g., holding the flower with the gripper, or rolling up the dough with the dexterous hand. Since they are *invariant* of the spatial transformation, a_t^{hand} should remain unchanged regardless of the object configuration:

$$\hat{a}_t^{\mathrm{hand}} = a_t^{\mathrm{hand}}, \quad \forall\, t, s_o, s_0'.$$

In contrast, the robot arm commands should be spatially *equivariant* to the object movements in order to adjust the trajectory according to the altered configuration. Specifically, for the motion and skill segments involving the k-th object, we adapt the robot arm commands $\mathbf{A}^{\mathrm{EE}}[\boldsymbol{\tau}_k^{\mathrm{m}}], \mathbf{A}^{\mathrm{EE}}[\boldsymbol{\tau}_k^{\mathrm{s}}]$ following a TAMP-based procedure, illustrated by Fig. 2.

For the skill segments with dexterous on-object behaviors, the spatial relations between end-effectors and objects must remain relatively static. Thus, the entire skill segments are transformed following the corresponding objects:

$$\hat{\mathbf{A}}^{\mathrm{EE}}[\boldsymbol{\tau}_k^{\mathrm{s}}] = \mathbf{A}^{\mathrm{EE}}[\boldsymbol{\tau}_k^{\mathrm{s}}] \cdot (\mathbf{T}_0^{O_k})^{-1} \cdot \mathbf{T}_0^{O_k\prime}.$$

For the motion segments moving in free space, the goal is to chain adjacent skill segments. Therefore, we plan the robot arm commands in the motion stage via motion planning:

$$\hat{\mathbf{A}}^{\mathrm{EE}}[\boldsymbol{\tau}_k^{\mathrm{m}}] = \texttt{MotionPlan}(\hat{\mathbf{A}}^{\mathrm{EE}}[\boldsymbol{\tau}_{k-1}^{\mathrm{s}}][-1],\ \hat{\mathbf{A}}^{\mathrm{EE}}[\boldsymbol{\tau}_k^{\mathrm{s}}][0]),$$

where the starting pose for motion planning is taken from the last frame of the previous skill segment, and the ending pose is from the first frame of the current skill segment. For simple uncluttered workspaces, linear interpolation suffices. For complex environments requiring obstacle avoidance, an off-the-shelf motion planning method [25] is employed.

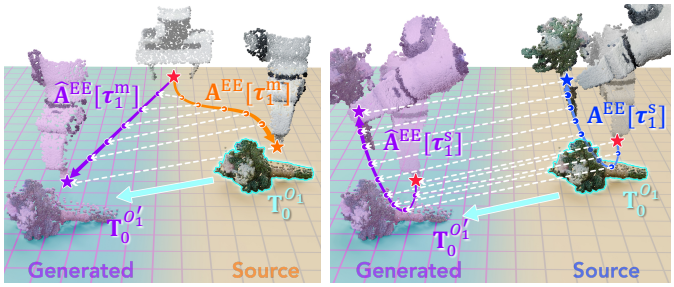

Fig. 2: **Illustrations for action generation.** (Left) Actions in the motion stage are planned to connect the neighboring skill segments. (Right) Actions in the skill stage undergo a uniform transformation.

D. Fully Synthetic Observation Generation

Adapting proprioceptive states. The observations consist of point cloud data and proprioceptive states. Since the proprioceptive states share the same semantics with the actions, they should undergo the same transformation:

$$\hat{o}_t^{\mathrm{hand}} = o_t^{\mathrm{hand}}, \quad \forall\, t, s_o, s_0';$$
$$\hat{o}_t^{\mathrm{arm}} = o_t^{\mathrm{arm}} \cdot (\mathbf{A}_t^{\mathrm{EE}})^{-1} \cdot \hat{\mathbf{A}}_t^{\mathrm{EE}}.$$

It is noteworthy that we found directly replacing the current state with the next target pose action (i.e., $\hat{o}_t^{\mathrm{arm}} \leftarrow \hat{a}_{t+1}^{\mathrm{arm}}$) may impair performance, as the IK controllers may not always achieve the exact target pose.

Synthesizing point cloud observations. To synthesize the spatially augmented point clouds for the robot and objects, we employ a simple segment-and-transform strategy. Apart from the target transformations, the only required information for synthesis is the segmentation masks for the K objects on the first frame of the source demonstration, obtained in Sec. II-B.

For each object, we define 3 stages. In the *to-do* stage, the object is static and unaffected by the robot, and its point cloud is transformed according to the initial object configuration $(\mathbf{T}_0^{O_k})^{-1} \cdot \mathbf{T}_0^{O_k\prime}$. In the *doing* stage, the object is in contact with the robot, and its point cloud is merged with the end-effector's point cloud. In the *done* stage, the object remains in its final state. These stages are easily identified by referencing the trajectory-level motion and skill segments.

For the robot's end-effector, its point cloud undergoes the same transformation as indicated by the proprioceptive states $(\mathbf{A}_t^{\mathrm{EE}})^{-1} \cdot \hat{\mathbf{A}}_t^{\mathrm{EE}}$. Given the assumption of a cropped workspace, the point clouds for the robot and the objects in the *doing* stage can be separated by subtracting the object point clouds in the *to-do* and *done* stages from the scene point cloud.

Examples of the synthetic trajectories in real-world experiments can be found in Fig. 21 and Fig. 22 in the appendix.

III. Experiments: O.O.D. Spatial Generalization

We assess the spatial generalization capability of visuomotor policies enhanced by *DemoGen* across 8 real-world tasks

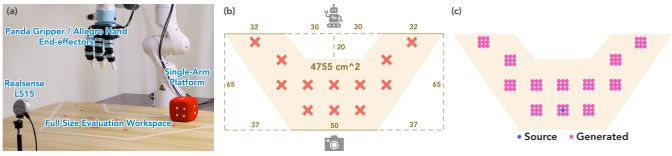

Fig. 3: **Protocol for evaluating spatial generalization.** (a) Setups on the single-arm platform. (b) Illustration for the full-size evaluation workspace. (c) Illustration for the generation strategy targeting the evaluated configurations along with small-range perturbations.

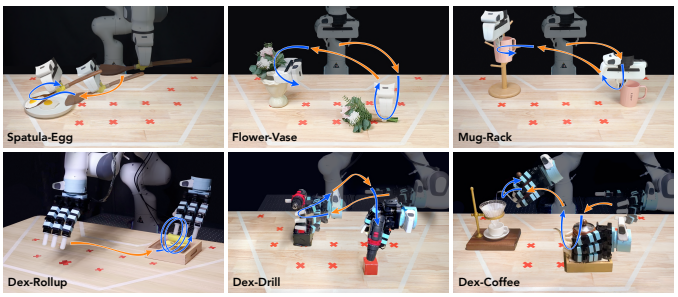

Fig. 4: **Tasks for real-world evaluation on spatial generalization.** Spatula-Egg and Dex-Rollup are one-stage tasks involving contact-rich behaviors. Flower-Vase, Mug-Rack, Dex-Drill, and Dex-Coffee are two-stage tasks requiring precise manipulation.

TABLE I: **A summary of real-world tasks for spatial generalization evaluation.** ActD: action dimension. #Obj: number of manipulated objects. #Eval: number of evaluated configurations. #GDemo: number of *DemoGen*-generated demonstrations.

Task	Platform	ActD	#Obj	#Eval	#GDemo
Spatula-Egg	Gripper	6	1	10	270
Flower-Vase	Gripper	7	2	4×4	432
Mug-Rack	Gripper	7	2	4×4	432
Dex-Cube	Dex. Hand	22	1	10	270
Dex-Rollup	Dex. Hand	22	1	12	324
Dex-Drill	Dex. Hand	22	2	3×3	243
Dex-Coffee	Dex. Hand	22	2	3×3	243
Fruit-Basket	Bimanual	14	2	4×6	72

deployed on 3 different platforms. A task summary is provided in Tab. I. Detailed generation strategies are supplied in Sec. E.

A. Single-Arm Platforms

Tasks. On the Franka Panda single-arm platform, we design 3 tasks using the original Panda gripper and 4 tasks using an Allegro dexterous hand as the end-effector. The motion and skill trajectories of these tasks are visualized in Fig. 4 and the task descriptions are provided in Appendix M. For all tasks, a single Intel Realsense L515 camera is adopted to capture point cloud observations, as depicted in Fig. 3(a).

Evaluation protocol. To evaluate spatial generalization, we define a large planar evaluation workspace, the size of which corresponds to the maximum reach of the robot arm. We uniformly sample 12 points within this irregularly-shaped workspace as the coordinates for potential object configurations, with a 15cm spacing between neighboring coordinates, as illustrated in Fig. 3(b). To determine the actual evaluated configurations for each task, we perform manual trials using kinematic teaching to confirm the feasibility of each configuration. We conduct trials on all feasible configurations and repeat the evaluations 5 times per configuration to ensure the reliability of the results.

Results analysis. The performance of visuomotor policies [57] trained on 3 source demonstrations and *DemoGen*-generated demonstrations are reported in Tab. III. Agents trained solely on source demonstrations exhibit severe overfitting behaviors, blindly replicating the demonstrated trajectory. In Appendix H, we evaluate the policy performance trained on datasets containing additional human-collected demonstrations. We found the spatial effective range of the trained policies is approximately the sum of demonstrated configurations, aligned with the empirical study in Sec. A. More details are listed in Sec. I.

TABLE II: **The time cost for generating real-world demonstrations.** The computational cost of *DemoGen* is measured on a single-process procedure. Since the synthetic generation process is highly parallelizable, it can be further accelerated using multi-processing.

	Single o-a Pair	A Trajectory	Whole Dataset
MimicGen	2.1 s	2.1 min	83.7 h
DemoGen	**0.00015** s	**0.010** s	**22.0** s

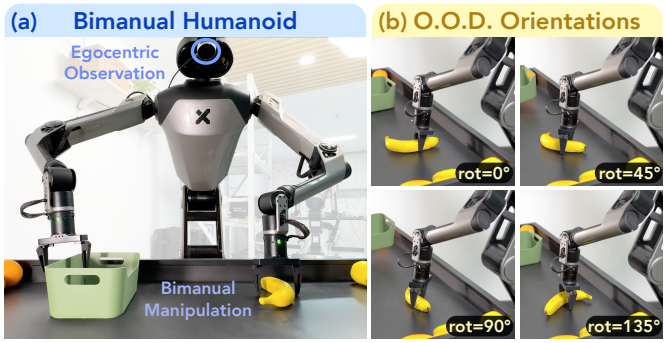

Fig. 5: **Bimanual humanoid platform.** (a) Egocentric observations and bimanual manipulation. (b) The Fruit-Basket task involves the out-of-distribution orientations during evaluation.

Generation cost. We compare the time cost of real-world demonstration generation between MimicGen [31] and *DemoGen*. We estimate MimicGen's time cost by multiplying the duration of replaying a source trajectory by the number of generated demonstrations and adding an additional 20 seconds per trajectory for human operators to reset the object configurations. It is important to note that MimicGen involves continuous human intervention, while the time cost of *DemoGen* is purely computational, without the involvement of either the robot or human operators.

B. Bimanual Humanoid Platform

Task. In addition to the tasks on the single-arm platform, we also designed a Fruit-Basket task on a Galaxea R1 robot, illustrated in Fig. 5. The Fruit-Basket task is distinguished from the previous tasks by three key features: *1) Bimanual manipulation.* The robot simultaneously grasps the basket with one arm and the banana with the other. *2) Egocentric observation.* The camera is mounted on the robot's head with a first-person view [56]. *3) O.O.D. orientations.* Still given a single demonstration, the banana is placed with orientational offsets (i.e., 45°, 90°, and 135°) during evaluation.

Results analysis. The success rates for both the source and generated datasets are compared in Tab. III, and the spatial heatmap is shown in Fig. 19. The high success rate of 90.8% demonstrates the effectiveness of *DemoGen* on bimanual hu-

TABLE III: **Real-world evaluation of *DemoGen* for spatial generalization.** For reliable evaluation, a total of 530 policy rollouts are conducted on the 8 tasks. The success rate for each task is averaged on 5 repetitions for each evaluated configuration. The evaluated configurations for each task are counted in Tab. I, and visualized in Fig. 19.

	Spatula-Egg	Flower-Vase	Mug-Rack	Dex-Cube	Dex-Rollup	Dex-Drill	Dex-Coffee	Fruit-Basket	Averaged
Source	10.0	6.3	6.3	10.0	8.3	11.1	11.1	25.0	11.0
DemoGen	**88.0**	**82.5**	**85.0**	**78.0**	**76.7**	**55.6**	**40.0**	**90.8**	**74.6**

Fig. 6: **Tasks with articulated objects.** (Left) Drawer-Opening and (Right) Door-Opening.

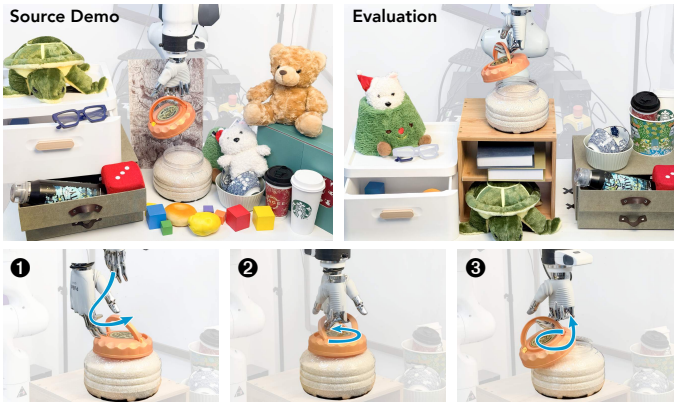

Fig. 7: **Task with cluttered scene.** The Lid-Opening task is demonstrated in a cluttered scene and tested in another scene where the jar is placed on a shelf.

manoid platforms and its ability to help policies generalize to out-of-distribution orientations. A more detailed analysis is presented in Appendix J.

IV. Experiments: More Complex Scenarios

A. Articulated Objects

In the tasks we have considered, the manipulation of the target objects is freewheeling, i.e., 6-DoF movement for rigid-body objects and infinite-DoF shaping for deformable objects. An unstudied class of representative tasks involves manipulating articulated objects, e.g., 1-DoF sliding or revolving.

Empirically, we conduct two supplementary tasks: 1) **Drawer-Opening** by pulling along the direction of its sliding rail; 2) **Door-Opening** by rotating around its hinge axis. Both the drawer and the door are randomized in a range of $20\text{cm} \times 10\text{cm}$. 30 evaluation rollouts are conducted for each task, and the averaged success rate of *DemoGen*-enabled policies is $\underline{\mathbf{88}}\%$: $27/30$ for the drawer; $26/30$ for the door.

B. Cluttered Scene

To highlight the effects of spatial generalization, we assume a clear tabletop workspace in Sec. III to maximize the diversity in the initial configurations of the manipulated objects. When the scene becomes even more complex, e.g., clutter, *DemoGen* does not necessarily work well. This is because the object point clouds may be occluded by the clutter, thus hindering the 3D point cloud editing process for generating synthetic visual observations. Yet, as long as the entire object(s) can be segmented from the scene, the *DemoGen* pipeline holds, unaffected by the presence of the surrounding distractors.

Empirically, we conduct a challenging **Lid-Opening** task, where the dexterous hand needs to stretch its fingers into the handle on the jar's lid, grasp it tight, twist a half circle, and finally lift to open the lid. The jar is placed in a cluttered scene in the source demonstration and on the top of a shelf, giving an O.O.D. height during evaluation. For *DemoGen* generation, we artificially lift the whole trajectory by 30cm to mimic the presence of the shelf. The success rate is $\underline{\mathbf{85}}\%$ $(17/20)$.

V. Experiments: Extended O.O.D. Capabilities

A. O.O.D. Disturbance Resistance

One critical advantage of visuomotor policies is their ability to perform closed-loop corrections under disturbances. We investigate whether a *DemoGen*-generated dataset, derived from one human-collected and two replayed source demonstrations, can train visuomotor policies equipped with such capability.

Task. We consider a Sauce-Spreading task (Fig. 8(a)) adapted from DP [5]. Initially, the pizza crust contains a small amount of sauce at its center (Fig. 8(b)). The gripper maneuvers the spoon in hand to approach the sauce center and periodically spread it to cover the pizza crust in a spiral pattern (Fig. 8(c)).

Evaluation protocol. During the sauce-spreading process, disturbances are introduced by shifting the pizza crust twice to the neighboring spots within the workspace. We consider 5 neighboring spots (Fig. 8(d)) and conduct 5 trials per spot,

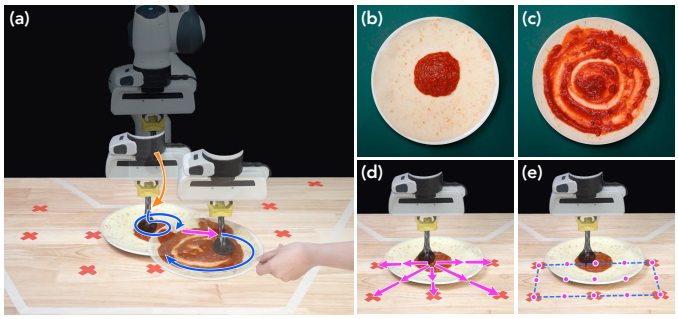

Fig. 8: ***DemoGen* for disturbance resistance.** (a-c) Illustration, initial, and ending states of the Sauce-Spreading task. (d) Disturbance applied for quantitative evaluation. (e) Standard generation strategy.

TABLE IV: **Real-world evaluation of *DemoGen* for disturbance resistance.** Raw evaluation results and detailed definitions for the metrics are presented in Appendix K.

	Sauce Coverage	Normalized Score
Regular *DemoGen*	34.2	40.4
***DemoGen* w/ ADR**	**61.2**	**92.3**
Initial State	13.2	0
Human Expert	65.2	100

resulting in 25 trials. For quantitative evaluation, we measure the sauce coverage on the pizza crust. Additionally, we report a normalized sauce coverage score, where 0 represents no operation taken, and 100 corresponds to human expert performance. Detailed calculations are provided in Appendix K.

Generation strategies. A standard generation strategy selects 15 intermediate spots (Fig. 8(e)) observed during the disturbance process as the initial object configurations for a standard *DemoGen* data generation procedure.

To specifically enhance disturbance resistance, we propose a specialized strategy named Augmentation for Disturbance Resistance (ADR), illustrated in Fig. 9. In ADR, the pizza crust is artificially displaced to nearby positions at certain time steps to simulate the disturbance. The robot's end-effector, holding the spoon, initially remains static and subsequently interpolates its motion to re-approach the displaced crust before continuing the periodic spreading motion.

Results analysis. Tab. IV presents the sauce coverage and normalized scores for both the standard *DemoGen* and the ADR-enhanced *DemoGen* strategies. The ADR strategy significantly outperforms the standard *DemoGen*, achieving performance comparable to human experts. In the video, we showcase the ADR-enhanced policy is still robust under up to 5 successive disturbances. These findings underscore the critical role of the demonstration data in enabling policy capabilities. The ability to resist disturbances does not emerge naturally but is acquired through targeted disturbance-involved demonstrations.

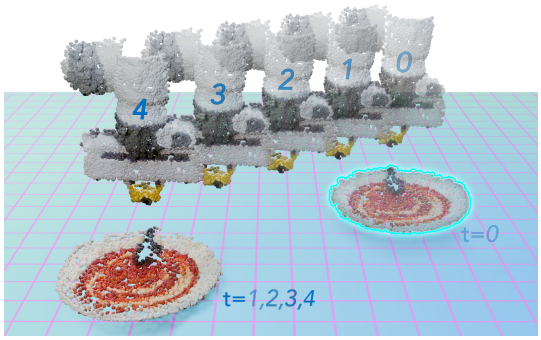

Fig. 9: **Illustration for the ADR strategy.** Asynchronous transformations are applied to the disturbed object and the robot end-effector to simulate the disturbance resistance process.

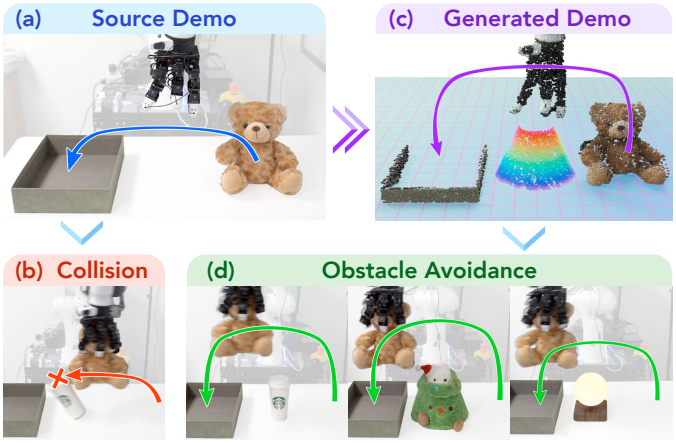

Fig. 10: ***DemoGen* for obstacle avoidance.** (ab) Policy trained on the source demonstration collides with the unseen obstacle. (cd) Policy trained on the generated dataset could avoid diverse-shaped obstacles.

B. O.O.D. Obstacle Avoidance

Task. The ability to avoid obstacles is also imparted through demonstrations containing obstacle-avoidance behaviors. To investigate such capability, we introduce obstacles to a Teddy-Box task, where the dexterous hand grasps the teddy bear and transfers it into the box on the left (Fig. 10(a)). Trained on the source demonstrations without obstacles, the visuomotor policy fails to account for potential collisions, e.g., it might knock over the coffee cup placed in the middle (Fig. 10(b)).

Generation strategy. To generate obstacle-involved demonstrations, we augment the real-world point cloud observations by sampling points from simple geometries, such as boxes and cones, and fusing these points into the original scene (Fig. 10(c)). Obstacle-avoiding trajectories are generated by a motion planning tool [25], ensuring collision-free actions.

Evaluation and results analysis. For evaluation, we position 5 everyday objects with diverse shapes in the middle of the workspace (Fig. 10(d)) and conduct 5 trials per object, resulting in a total of 25 trials. The agent trained on the augmented dataset successfully bypasses obstacles in 22 out of 25 trials. Notably, in scenarios without obstacles, the agent follows the lower trajectory observed in the source demonstrations, indicating its responsiveness to environmental variations.

VI. Live-streaming: Efficiency and Effectiveness

DemoGen is simple yet effective. To highlight its strength, we **live-streamed** its full pipeline on social media, including: 1) collecting one source demonstration, 2) generating synthetic demonstrations with *DemoGen*, 3) visuomotor policy training, and 4) testing the trained policy on the real robot. The pipeline was finished in around **30 minutes**, and the trained policy achieved a **100**% success rate with **O.O.D.** configurations on the challenging Jar-Opening task.

VII. Conclusion

In this work, we introduce *DemoGen*, a fully synthetic data generation system designed to help visuomotor policy

learning generalize to various O.O.D. scenarios. We discuss the limitations and future works in Sec. O.

REFERENCES

[1] Kaylee Burns, Zach Witzel, Jubayer Ibn Hamid, Tianhe Yu, Chelsea Finn, and Karol Hausman. What makes pre-trained visual representations successful for robust manipulation? *arXiv preprint arXiv:2312.12444*, 2023.

[2] Tao Chen, Jie Xu, and Pulkit Agrawal. A system for general in-hand object re-orientation. In *CoRL*, 2022.

[3] Shuo Cheng, Caelan Reed Garrett, Ajay Mandlekar, and Danfei Xu. Nod-tamp: Multi-step manipulation planning with neural object descriptors. In *CoRL 2023 Workshop on Learning Effective Abstractions for Planning (LEAP)*, 2023.

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

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

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

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

[8] 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.

[9] Norman Di Palo and Edward Johns. Learning multi-stage tasks with one demonstration via self-replay. In *Conference on Robot Learning*, pages 1180–1189. PMLR, 2022.

[10] Norman Di Palo and Edward Johns. Dinobot: Robot manipulation via retrieval and alignment with vision foundation models. *arXiv preprint arXiv:2402.13181*, 2024.

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

[12] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996.

[13] Pete Florence, Corey Lynch, Andy Zeng, Oscar A Ramirez, Ayzaan Wahid, Laura Downs, Adrian Wong, Johnny Lee, Igor Mordatch, and Jonathan Tompson. Implicit behavioral cloning. In *CoRL*, 2022.

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

[15] Chongkai Gao, Zhengrong Xue, Shuying Deng, Tianhai Liang, Siqi Yang, Lin Shao, and Huazhe Xu. Riemann: Near real-time se (3)-equivariant robot manipulation without point cloud segmentation. *arXiv preprint arXiv:2403.19460*, 2024.

[16] George Jiayuan Gao, Tianyu Li, and Nadia Figueroa. Out-of-distribution recovery with object-centric keypoint inverse policy for visuomotor imitation learning. *arXiv preprint arXiv:2411.03294*, 2024.

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

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[19] Ryan Hoque, Ajay Mandlekar, Caelan Garrett, Ken Goldberg, and Dieter Fox. Intervengen: Interventional data generation for robust and data-efficient robot imitation learning. *arXiv preprint arXiv:2405.01472*, 2024.

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

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

[22] Edward Johns. Coarse-to-fine imitation learning: Robot manipulation from a single demonstration. In *2021 IEEE international conference on robotics and automation (ICRA)*, pages 4613–4619. IEEE, 2021.

[23] Tsung-Wei Ke, Nikolaos Gkanatsios, and Katerina Fragkiadaki. 3d diffuser actor: Policy diffusion with 3d scene representations. *arXiv preprint arXiv:2402.10885*, 2024.

[24] 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.

[25] James J Kuffner and Steven M LaValle. Rrt-connect: An efficient approach to single-query path planning. In *Proceedings 2000 ICRA. Millennium Conference. IEEE*

International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065), volume 2, pages 995–1001. IEEE, 2000.

[26] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17(39):1–40, 2016.

[27] 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.

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

[29] I Loshchilov. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[30] Ajay Mandlekar, Caelan Reed Garrett, Danfei Xu, and Dieter Fox. Human-in-the-loop task and motion planning for imitation learning. In *Conference on Robot Learning*, pages 3030–3060. PMLR, 2023.

[31] 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.

[32] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. In *Conference on Robot Learning*, pages 892–909. PMLR, 2023.

[33] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[34] 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.

[35] Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 3803–3810. IEEE, 2018.

[36] Aaditya Prasad, Kevin Lin, Jimmy Wu, Linqi Zhou, and Jeannette Bohg. Consistency policy: Accelerated visuomotor policies via consistency distillation. In *Robotics: Science and Systems*, 2024.

[37] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017.

[38] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[39] Alfredo Reichlin, Giovanni Luca Marchetti, Hang Yin, Ali Ghadirzadeh, and Danica Kragic. Back to the manifold: Recovering from out-of-distribution states. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8660–8666. IEEE, 2022.

[40] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024.

[41] Vaibhav Saxena, Matthew Bronars, Nadun Ranawaka Arachchige, Kuancheng Wang, Woo Chul Shin, Soroush Nasiriany, Ajay Mandlekar, and Danfei Xu. What matters in learning from large-scale datasets for robot manipulation. In *CoRL 2024 Workshop on Mastering Robot Manipulation in a World of Abundant Data*, 2024. URL https://openreview.net/forum?id=oY0PQdLA6c.

[42] Anthony Simeonov, Yilun Du, Andrea Tagliasacchi, Joshua B Tenenbaum, Alberto Rodriguez, Pulkit Agrawal, and Vincent Sitzmann. Neural descriptor fields: Se (3)-equivariant object representations for manipulation. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6394–6400. IEEE, 2022.

[43] Anthony Simeonov, Yilun Du, Lin Yen-Chen, , Alberto Rodriguez, , Leslie P. Kaelbling, Tomas L. Perez, and Pulkit Agrawal. Se(3)-equivariant relational rearrangement with neural descriptor fields. In *Conference on Robot Learning (CoRL)*. PMLR, 2022.

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

[45] Marcel Torne, Anthony Simeonov, Zechu Li, April Chan, Tao Chen, Abhishek Gupta, and Pulkit Agrawal. Reconciling reality through simulation: A real-to-sim-to-real approach for robust manipulation. *arXiv preprint arXiv:2403.03949*, 2024.

[46] Eugene Valassakis, Georgios Papagiannis, Norman Di Palo, and Edward Johns. Demonstrate once, imitate immediately (dome): Learning visual servoing for one-shot imitation learning. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8614–8621. IEEE, 2022.

[47] Vitalis Vosylius and Edward Johns. Instant policy: In-context imitation learning via graph diffusion. *arXiv preprint arXiv:2411.12633*, 2024.

[48] Dian Wang, Stephen Hart, David Surovik, Tarik Kelestemur, Haojie Huang, Haibo Zhao, Mark Yeatman,

Jiuguang Wang, Robin Walters, and Robert Platt. Equivariant diffusion policy. *arXiv preprint arXiv:2407.01812*, 2024.

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

[50] 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.

[51] Zhendong Wang, Zhaoshuo Li, Ajay Mandlekar, Zhenjia Xu, Jiaojiao Fan, Yashraj Narang, Linxi Fan, Yuke Zhu, Yogesh Balaji, Mingyuan Zhou, et al. One-step diffusion policy: Fast visuomotor policies via diffusion distillation. *arXiv preprint arXiv:2410.21257*, 2024.

[52] Bowen Wen, Wenzhao Lian, Kostas Bekris, and Stefan Schaal. You only demonstrate once: Category-level manipulation from single visual demonstration. *Robotics: Science and Systems 2022*, 2022.

[53] Zhengrong Xue, Zhecheng Yuan, Jiashun Wang, Xueqian Wang, Yang Gao, and Huazhe Xu. Useek: Unsupervised se (3)-equivariant 3d keypoints for generalizable manipulation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1715–1722. IEEE, 2023.

[54] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *CoRL*, 2020.

[55] Zhecheng Yuan, Tianming Wei, Shuiqi Cheng, Gu Zhang, Yuanpei Chen, and Huazhe Xu. Learning to manipulate anywhere: A visual generalizable framework for reinforcement learning. *arXiv preprint arXiv:2407.15815*, 2024.

[56] 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.

[57] Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy. *arXiv preprint arXiv:2403.03954*, 2024.

[58] Andy Zeng, Pete Florence, Jonathan Tompson, Stefan Welker, Jonathan Chien, Maria Attarian, Travis Armstrong, Ivan Krasin, Dan Duong, Vikas Sindhwani, et al. Transporter networks: Rearranging the visual world for robotic manipulation. In *Conference on Robot Learning*, pages 726–747. PMLR, 2021.

[59] Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

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

[61] Haoyi Zhu, Yating Wang, Di Huang, Weicai Ye, Wanli Ouyang, and Tong He. Point cloud matters: Rethinking the impact of different observation spaces on robot learning. *arXiv preprint arXiv:2402.02500*, 2024.

APPENDIX

A. Empirical Study: Spatial Generalization of Visuomotor Policies

In this section, we present an empirical study examining the spatial generalization capability of visuomotor policies. We demonstrate how the lack of such generalization contributes to the data-intensive nature of learning visuomotor policies.

1) Visualization of Spatial Effective Range: Spatial generalization refers to the ability of a policy to perform tasks involving objects placed in configurations that were not seen during training. To gain an intuitive understanding of spatial generalization, we visualize the relationship between the spatial effective range of visuomotor policies and the spatial distribution of demonstration data.

Tasks. We evaluate a Button-Large task adapted from the MetaWorld [54] benchmark, where the robot approaches a button and presses it down. The object randomization range is modified to a $30\,\mathrm{cm} \times 40\,\mathrm{cm} = 1200\,\mathrm{cm}^2$ area on the tabletop workspace, covering most of the end-effector's reachable space. Noticing the large size of the button makes it pressed down even if the press motion does not precisely hit the center, we also examine a more precision-demanding variant, Button-Small, where the button size is reduced by a factor of 4.

Policy. We adopt 3D Diffusion Policy (DP3) [57] as the studied policy, as our benchmarking results indicate that 3D observations provide superior spatial generalization compared to 2D approaches. Training details are provided in Appendix F1.

Evaluation. To visualize the spatial effective range, we uniformly sample 21 points along each axis within the workspace, resulting in a total of 441 distinct button placements. Demonstrations are generated using a scripted policy, with 4 different spatial distributions ranging from `single` to `full`. The performance of each configuration is evaluated on the 441 placements, enabling a comprehensive assessment of spatial generalization. The visualization result is presented in Fig. 11.

Key findings. Overall, the spatial effective range of visuomotor policies is closely tied to the distribution of object configurations seen in the demonstrations. Specifically, the effective range can be approximated by the union of the areas surrounding the demonstrated object placements. Thus, to train a policy that generalizes well across the entire object randomization range, demonstrations must cover the full workspace, resulting in substantial data collection costs. Furthermore, as task precision requirements increase, the effective range shrinks to more localized areas, necessitating a greater number of demonstrations to adequately cover the workspace. A more detailed analysis is available in Appendix G1.

2) Benchmarking Spatial Generalization Capability: The practical manifestation of the spatial generalization is reflected in the number of demonstrations required for effective policy learning. In the following benchmarking, we explore the relationship between the number of demonstrations and policy performance to determine how many demonstrations are sufficient for effective training.

Tasks. To suppress the occurrence of inaccurate but successful policy rollouts, we design a Precise-Peg-Insertion task that enforces a strict fault tolerance of $1\,\mathrm{cm}$ during both the picking and insertion stages, asking for millimeter-level precision. The peg and socket are randomized within a $40\,\mathrm{cm} \times 20\,\mathrm{cm}$ area, yielding an effective workspace of $40\,\mathrm{cm} \times 40\,\mathrm{cm} = 1600\,\mathrm{cm}^2$. To examine the influence of object randomization, we also consider a `half` workspace, where the randomization range is halved for both objects, and a `fixed` setting, where object positions remain fixed. More details are listed in Appendix G3.

Policies. In addition to Diffusion Policy (DP) [5] and 3D Diffusion Policy (DP3) [57] trained from scratch, we explore the potential of pre-trained visual representations to enhance spatial generalization. Specifically, we replace the train-from-scratch ResNet [18] encoder in DP with pre-trained encoders including R3M [32], DINOv2 [33], and CLIP [38]. Detailed implementations are provided in Appendix F2.

Demonstrations. We vary the number of demonstrations from 25 to 400. The object configurations are randomly sampled from a slightly larger range than the evaluation workspace to avoid performance degradation near workspace boundaries. A visualization is provided in Fig. 16 in the appendix.

Evaluation. In the `full` workspace, both the peg and socket are placed on 45 uniformly sampled coordinates, resulting in

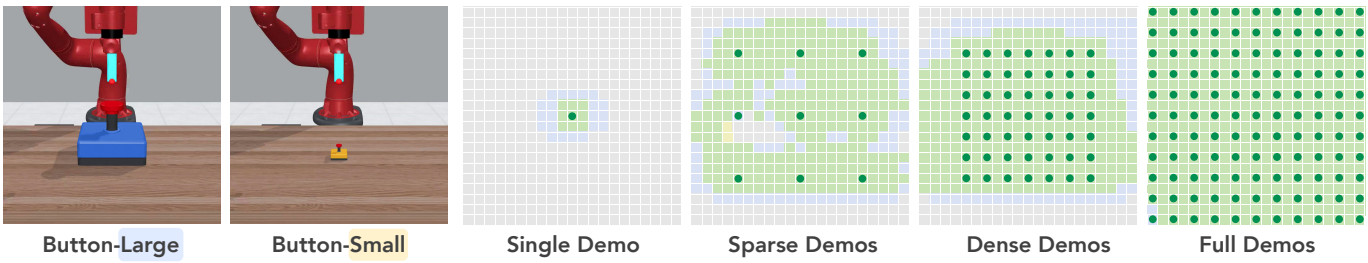

Fig. 11: **Qualitative visualization of the spatial effective range.** The grid maps display discretized tabletop workspaces from a bird's-eye view under different demonstration configurations. Dark green spots mark the locations where buttons are placed during the demonstrations. Each grid cell corresponds to a policy rollout with the button placed at that location. Blue, yellow, green, and gray grids denote successful executions for the Button-Large, Button-Small, both tasks, and no tasks, respectively.

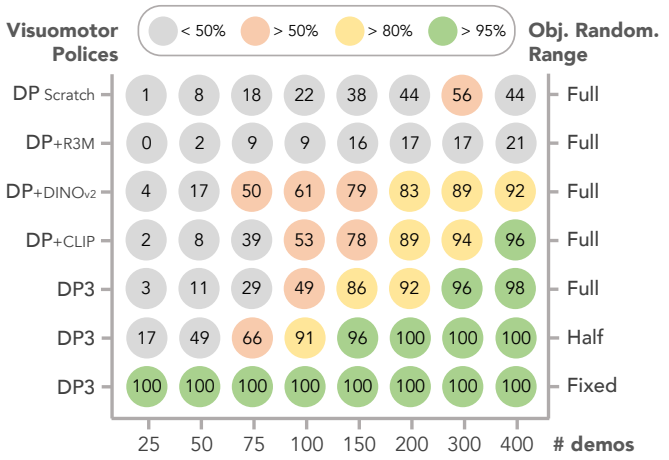

Fig. 12: **Quantitative benchmarking on the spatial generalization capacity.** We report the relationship between the agent's performance in success rates and the number of demonstrations used for training when different visuomotor policies and object randomization ranges are adopted. The results are averaged over 3 seeds.

2025 distinct configurations for evaluation. For the `half` and `fixed` settings, the number of evaluated configurations is 225 and 1, respectively. The results are presented in Fig. 12.

Key findings. The degree of object randomization significantly influences the required demonstrations. Therefore, an effective evaluation protocol for visuomotor policies must incorporate a sufficiently large workspace to provide enough object randomization. On the other hand, both 3D representations and pre-trained 2D visual encoders contribute to improved spatial generalization capabilities. However, none of these methods fundamentally resolve the spatial generalization problem. This indicates the agent's spatial capacity is not inherently derived from the policy itself but instead develops through extensive traversal of the workspace from the given demonstrations. A more detailed analysis is provided in Appendix G2.

B. Related Works

1) Visuomotor Policy Learning: Represented by Diffusion Policy [5] and its extensions [57, 23, 36, 51, 48], visuomotor policy learning refers to the imitation learning methods that learn to predict actions directly from visual observations in an end-to-end fashion [26]. The end-to-end learning objective is a two-edged sword. Its flexibility enables visuomotor policies to learn dexterous skills from human demonstrations, extending beyond rigid-body pick-and-place. However, the absence of structured skill primitives makes such policies intrinsically data-intensive.

The conflicts between the huge data demands and the great expense of robotic data collection have driven the growing attention to data-centric research. Such efforts include more efficient data collection systems [7, 4, 27], collaborative gathering of large-scale datasets [34, 24], and empirical studies on data scaling [60, 28]. Instead of scaling up via pure human labor, *DemoGen* aims to show that synthetic data generation can help save much of the human effort.

2) TAMP-Based Imitation Learning: Attempting to develop manipulation policies from only a handful of demonstrations, data-efficient imitation learning methods often build on the principles of Task and Motion Planning (TAMP) [8, 3, 30], while incorporating imitation learning to replace some components in the TAMP pipeline. A common approach is to learn the end-effector poses for picking and placing [58, 42, 52, 53, 15, 43]. The whole trajectories are generated using motion planning toolkits [25] and then executed in an open-loop manner. Some methods extend this idea to more complex scenarios by learning to estimate the states of manipulated objects in the environment and replaying demonstrated trajectory segments centered around the target objects [22, 46, 9, 10]. While these approaches are effective for simpler, Markovian-style tasks [47], as we will show in Sec. D, their reliance on open-loop execution limits their application to more dexterous tasks requiring closed-loop retrying and re-planning.

In contrast, *DemoGen* leverages the TAMP principles for synthetic data generation. Subsequently, the synthetic demonstrations are used to train closed-loop visuomotor policies for task resolution. In this way, *DemoGen* effectively combines the merits of both approaches.

3) Data Generation for Robotic Manipulation: Automated demonstration generation offers the opportunity to breed capable visuomotor policies with significantly reduced human efforts. A branch of recent works attempts to generate demonstrations by leveraging LLM for task decomposition and then using planning or reinforcement learning for subtask resolution [49, 20, 50]. While this paradigm enables data generation from the void, the resulting manipulation skills are often restricted by the capacity of either LLM, planning, or reinforcement learning.

An alternative line of research is exemplified by MimicGen [31] and its extensions [19, 17, 21]. Unlike generating demonstrations from the void, MimicGen adapts some human-collected source demonstrations to novel object configurations by synthesizing corresponding execution plans. This approach is theoretically applicable to a wide range of manipulation skills and object types. For example, DexMimicGen [21] extends MimicGen's strategy to support bi-manual platforms equipped with dexterous hand end-effectors. However, execution plans produced by the MimicGen framework are not ready-to-use demonstrations in the form of observation-action pairs. To bridge this gap, the MimicGen family [31, 19, 17, 21] relies on costly on-robot rollouts, which poses significant challenges for the deployment on physical robots.

Building upon MimicGen and its extensions, *DemoGen* incorporates their strategies for generating execution plans, but replaces the expensive on-robot rollouts with an efficient, fully synthetic generation process. This enables *DemoGen* to generate real-world demonstrations ready for policy training in a cost-effective manner.

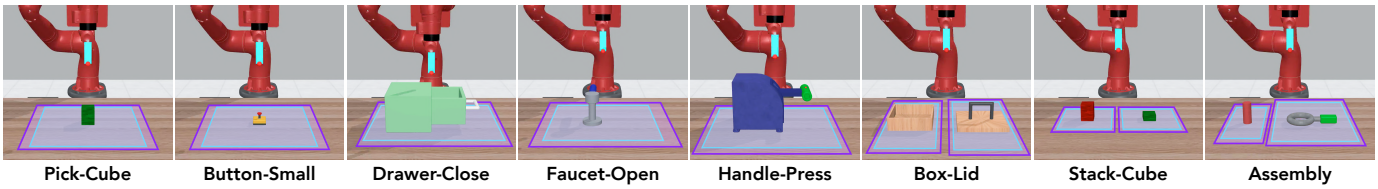

Fig. 13: **Tasks for simulated evaluation on spatial generalization.** Purple and sky-blue rectangles mark the workspaces for demonstration generation and evaluation, respectively. The detailed sizes of these workspaces are listed in Tab. VII in the appendix.

TABLE V: **Simulated evaluation of *DemoGen* for spatial generalization.** We report the maximum/averaged success rates over 3 seeds.

	Pick-Cube	Button-Small	Drawer-Close	Faucet-Open	Handle-Press	Box-Lid	Stack-Cube	Assembly	Averaged
1 Source	0/0	4/4	55/50	39/23	17/16	11/11	0/0	0/0	16/13
DemoGen	**76/73**	**92/84**	**100/100**	**95/92**	**100/100**	**100/95**	**79/77**	**86/83**	**91/88**
10 Source	29/29	54/52	100/100	90/89	100/99	94/89	44/38	47/45	70/68
25 Source	82/74	90/84	100/100	100/100	100/100	100/100	95/93	83/79	94/91

C. Experiments in the Simulator

1) Effectiveness: One-Shot Imitation: Before deploying *DemoGen* to the real world, we evaluate its effectiveness in the simulator by training visuomotor policies on datasets generated by *DemoGen* from only one source demonstration per task.

Policy. Both in the simulator and real world, we select DP3 [57] as the visuomotor policy, which predicts actions by consuming point cloud and proprioception observations. For a fair comparison, we fix the total training steps counted by observation-action pairs for all evaluated settings, resulting in an equal training cost regardless of the dataset size. The training details are listed in Appendix F1.

Tasks. We design 8 tasks adapted from the MetaWorld [54] benchmark, illustrated in Fig. 13. To strengthen the significance of spatial generalization, we modify these tasks to have enlarged object randomization ranges, as listed in Appendix L.

Generation and evaluation. We write scripted policies for these tasks and prepare only 1 source demonstration per task for demonstration generation. We also produce 10 and 25 source demonstrations per task using the scripted policy as a reference for human-collected datasets. Based on the one source demonstration, we leverage *DemoGen* to generate 100 spatially augmented demonstrations for the tasks containing the spatial randomization of one object. Since the tasks concerning two objects have a more diverse range of object configurations, 200 demonstrations are generated.

Results analysis. The evaluation results for the simulated tasks are presented in Tab. V. *DemoGen* significantly enhances the policy performance compared with the source demonstration baseline. The policies trained on *DemoGen*-generated datasets also outperform those trained on 10 source demonstrations and get close to 25 source demonstrations. This indicates *DemoGen* has the potential to maintain the policy performance with over 20× reduced human effort for data collection.

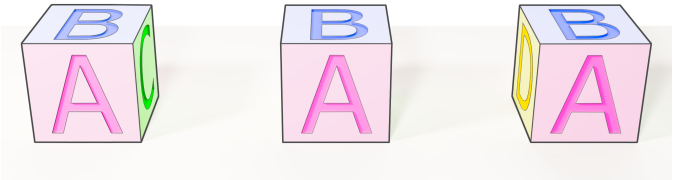

Fig. 14: **Illustration for the visual mismatch problem.** As objects move through 3D space, their appearance changes due to variations in perspective. Under the constraint of a single-view observation, synthetic demonstrations consistently reflect a fixed side of the object's appearance seen in the source demonstration. This discrepancy causes a visual mismatch between the synthetic and real-captured data.

2) Limitation: The Visual Mismatch Problem: While the one-shot imitation experiment verifies the effectiveness of *DemoGen*, it also reveals its limitation: synthetic demonstrations generated from one source demonstration are not as effective as the same number of human-collected demonstrations. We attribute the performance gap to the visual mismatch between the synthetic point clouds and those captured in the real world, under the constraint of a single-view observation perspective. An illustration is provided in Fig. 14.

Performance saturation. A notable consequence of the visual mismatch problem is the phenomenon of performance saturation. An empirical analysis is conducted on the Pick-Cube task. In Fig. 15(a), we fix the spatial density of target object configurations in the synthetic demonstrations and increase their spatial coverage by adding more synthetic demonstrations. The curve indicates that the performance improvement plateaus once the spatial coverage exceeds a certain threshold. This saturation occurs because the visual mismatch intensifies as the distance between the source and synthetic object configurations increases, making additional synthetic demonstrations ineffective. In Fig. 15(b), a similar performance saturation effect is observed when we increase the density while keeping the spatial coverage fixed. This indicates excessive demonstrations are unnecessary once they sufficiently cover the workspace.

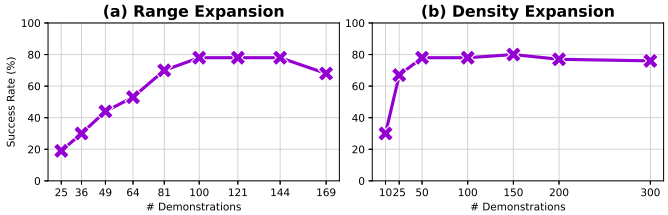

Fig. 15: **Performance Saturation.** We report the policy performance boost w.r.t. the increase of synthetic demonstrations over 3 seeds.

D. Discussion: DemoGen v.s. TAMP-Based Imitation

The correctness of *DemoGen*-generated demonstrations is built upon the effectiveness of TAMP-based methods designed for one-shot imitation learning [22, 46, 9, 10], which manipulate by aligning the demonstrated trajectory with the target object in the novel scene and then replaying the transformed trajectory. A natural question arises: *If TAMP is effective by itself, what is the meaning and the necessity of the more complicated DemoGen pipeline?*

This question asks for a very fundamental comparison of the pros and cons between TAMP and visuomotor policies. In principle, TAMP-based methods directly manipulate objects by planning and then executing actions in an ***open-loop*** manner. In contrast, *DemoGen* adopts TAMP as a tool for data generation, and the generated demonstrations are used to train ***closed-loop*** visuomotor policies.

We acknowledge that advanced TAMP-based methods are well effective for many tasks, as long as the task is represented in a ***static*** scene, e.g., rigid-body pick-and-place. However, we believe closed-loop visuomotor policies have irreplaceable advantages in more dexterous tasks, where the capability for ***real-time responsiveness*** is often indispensable.

Notable examples in our paper: 1) **Dex-Rollup** where the policy dynamically adjusts the wrapping motions (e.g., the numbers of wrapping, when to stop wrapping) in response to the distinct plasticity of every hand-molded piece of plasticine; 2) **Sauce-Spreading** where the policy dynamically replans the spreading motions in response to external disturbance.

Empirically, we compare the performance of *DemoGen*-enabled policies against DINOBot [10], an advanced TAMP-based method. The results indicate *DemoGen*-enabled closed-loop policies are more responsive agents than TAMP, especially under external disturbances.

TABLE VI: **Comparison between open-loop and closed-loop policies in terms of responsiveness.** In line with the aforementioned evaluation metrics, we report the success rate for Dex-Rollup and normalized coverage score for Sauce-Spreading.

	Dex-Rollup	Sauce-Spreading
DINOBot	55.0	24.0
DemoGen	**76.7**	**92.3**

E. Detailed Generation Strategies

1) Single-Arm Platforms: As in the simulated environments, we collect only one source demonstration for each task. However, real-world point cloud observations are often noisy, with issues such as flickering holes in the point clouds or projective smearing around object outlines. Even after performing clustering and downsampling during the point cloud preprocessing stage (Sec. II-B), the imitation learning policy can overfit to these irregularities if only one demonstration is provided.

To mitigate this issue, we replay the source demonstration twice and capture the corresponding point cloud observations. The altogether 3 point cloud trajectories enrich the diversity in visual degradations and help alleviate the overfitting problem. Since replaying twice is low-cost, we consider this approach a beneficial tradeoff between efficiency and effectiveness.

For each task, we set the generated object configurations to correspond to the evaluated configurations. However, human operators cannot always place objects with perfect precision in the real world, yet we found visuomotor policies are sensitive to even small deviations. Thus, we further augment the generated object configurations by adding small-range perturbations. Specifically, for each target configuration, we generate 9 demonstrations with $(\pm 1.5\text{cm}) \times (\pm 1.5\text{cm})$ perturbation to mimic slight placement variations in the real world. The final generated configurations are shown in Fig. 3(c).

In summary, the total number of generated demonstrations is calculated as $3 \times (\#\text{Eval}) \times 9$, which represents the 3 source demonstrations, multiplied by the number of evaluated configurations, and further multiplied by the 9 perturbations. The detailed counts for each task are listed in Tab. I.

2) Bimanual Humanoid Platform: The generation procedure follows a similar approach as that used for the single-arm platform. Specifically, the human-collected demonstration is replayed twice, yielding 3 source demonstrations in total. *DemoGen* generates synthetic demonstrations by independently adapting the actions of both arms to the respective transformations of the objects. Small-range perturbations are omitted in this task due to the relatively lower precision requirements.

A challenge in synthesizing point cloud observations with orientational offsets lies in the limited view provided by the single camera, which only captures the objects' front-facing appearance. To address this limitation, the humanoid robot adopts a stooping posture, enabling a near bird's-eye view perspective. This adjustment allows for more effective point cloud editing to simulate full-directional yaw rotations.

F. Policy Training and Implementation Details

We select 3D Diffusion Policy (DP3) [57] as the visuomotor policy used for real-world and simulated experiments. We compare its performance against 2D Diffusion Policy (DP) [5] in the empirical study in Sec. A. We list the training and implementation details as follows.

1) Details for Policy Training: For a fair comparison, we fix the total training steps counted by observation-action pairs to be 2M for all evaluated settings, resulting in an equal training cost regardless of the dataset size. To stabilize the training

process, we use AdamW [29] optimizer and set the learning rate to be $1\mathrm{e}^{-4}$ with a 500 step warmup.

In real-world experiments, we use the DBSCAN [12] clustering algorithm to discard the outlier points and downsample the number of points in the point cloud observations to 1024. In the simulator, we skip the clustering stage and downsample the point clouds to 512 points.

We follow the notation in the Diffusion Policy [5] paper, where T_{o} denotes the observation horizon, T_{p} as the action prediction horizon, and T_{a} denotes the action execution horizon. In real-world experiments, we set $T_{\mathrm{o}} = 2$, $T_{\mathrm{p}} = 8$, $T_{\mathrm{a}} = 5$. We run the visuomotor policy at 10Hz. Since T_{a} indicates the steps of actions executed on the robot without re-planning, our horizon settings result in a closed-loop re-planning latency of 0.5 seconds, responsive enough for conducting dexterous retrying behaviors and disturbance resistance. In the simulator, since the tasks are simpler, we set $T_{\mathrm{o}} = 2$, $T_{\mathrm{p}} = 4$, $T_{\mathrm{a}} = 3$.

2) Pre-Trained Encoders for Diffusion Policies: To replace the train-from-scratch ResNet18 [18] visual encoder in the original Diffusion Policy architecture, we consider 3 representative pre-trained encoders: R3M [32], DINOv2 [33], and CLIP [38]. R3M utilizes a ResNet [18] architecture and is pre-trained on robotics-specific tasks. DINOv2 and CLIP employ ViT [11] architectures and are pre-trained on open-world vision tasks. These encoders are widely used in previous works [6, 28] to enhance policy performance.

G. Spatial Generalization Empirical Study Details

In Sec. A, we conducted an empirical study on the spatial generalization capability of visuomotor policies. In this section, we provide more detailed analysis of the study's results.

1) More Analysis on the Visualization Results in Fig. 11: The results suggest that visuomotor policies exhibit some degree of spatial interpolation capability. Specifically, the green-colored effective range in the `sparse` setting with 9 demonstrations is significantly larger than 9 times the effective range in the `single` setting. However, increased precision requirements would make it harder to interpolate, as indicated by comparing the two task variants under the `sparse` setting.

Meanwhile, extrapolation proves to be more challenging. Although the number of demonstrations in the `dense` setting is much larger than in the `sparse` setting, the contours of the effective range remain similar across both cases. This suggests more demonstrations near the center of the workspace do not significantly extend the effective range to more distant areas.

On the whole, the spatial generalization range of visuomotor policies can be roughly approximated by the union of the adjacent areas around the object configurations in the provided demonstrations. The extent of the adjacent regions is influenced by the fault tolerance level required for manipulation.

2) More Analysis on the Benchmarking Results in Fig. 12: On visuomotor policies, we find DP3 exhibits the highest spatial generalization capacity compared to all 2D-based counterparts. Additionally, models utilizing CLIP and DINOv2 representations achieve competitive results, significantly surpassing the train-from-scratch baseline. This highlights the value of pre-training on open-world vision tasks in enhancing spatial reasoning capabilities for robotic manipulation. Notably, while prior studies [7, 28, 1, 61] have emphasized the benefits of 3D and pre-trained encoders for visual generalization, our findings extend these insights to spatial generalization, offering a complementary perspective on encoder selection strategies.

On object randomization range, our experiments with `half` and `fixed` settings demonstrate that high precision requirements alone do not necessarily produce challenging tasks unless the object positions are fully randomized. This suggests that precision requirements and spatial randomization both contribute to the task difficulty.

On the number of demonstrations, while task performance generally improves with an increased number of demonstrations, the effect diminishes beyond a certain threshold. For instance, in the `full` workspace setting with DP3 as the policy, a 50 demonstrations increase from 100 to 150 enhances the performance by 37%, but the increase from 150 to 200 only improves the performance by 6%. This finding also highlights the inherent difficulty in achieving near-perfect success rates in robot learning systems.

3) The Precise-Peg-Insertion Task: We construct a T-shaped peg, whose upper end has a cross-section of $6\,\mathrm{cm} \times 6\,\mathrm{cm}$, and the bottom end has a cross-section of $3\,\mathrm{cm} \times 3\,\mathrm{cm}$. The hole in the green socket has a cross-section of $4\,\mathrm{cm} \times 4\,\mathrm{cm}$. This shape enforces a strict fault tolerance of 1 cm during both the picking and insertion stages, asking for millimeter-level precision. Both objects are randomized in a $40\,\mathrm{cm} \times 20\,\mathrm{cm}$ workspace in the `full` setting. The randomization range is halved into $20\,\mathrm{cm} \times 10\,\mathrm{cm}$ in the `half` setting.

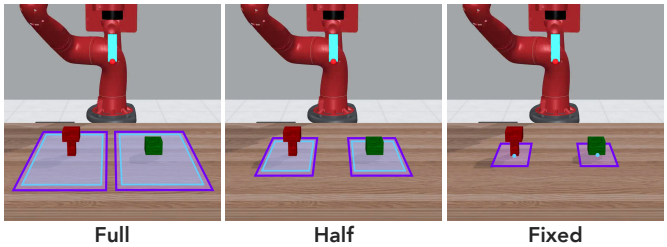

Fig. 16: **The Precise-Peg-Insertion task.** A total of 3 workspace sizes is considered. Purple and sky-blue rectangles mark the workspaces for demonstration and evaluation, respectively.

H. Increased Human-Collected Demonstrations

In Tab. III, we compare the *DemoGen*-generated dataset against 3 human-collected source demonstrations. In Fig. 18, we provide a reference on how the increase of source demonstrations leads to the enhancement of policy performance on the Dex-Cube task. To further understand the policy capacity enabled by human-collected demonstrations, we visualize the spatial heatmaps of human-collected datasets in Fig. 17. By comparing the demonstrated configurations and the spatial effective range of the resulting policies, we found the policy

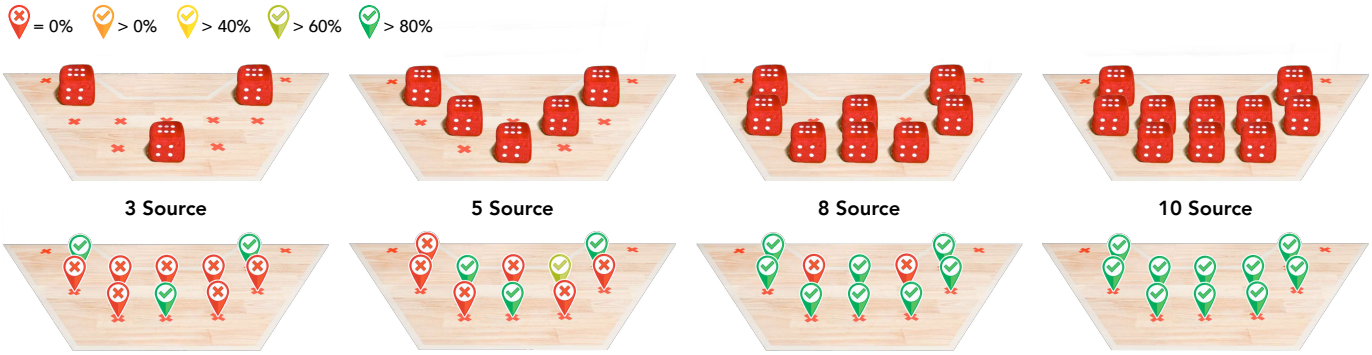

Fig. 17: **Visualization of the policy performance trained on human-collected datasets.** (Upper row) The demonstrated configurations. (Bottom row) The spatial heatmaps with success rates averaged on 5 trials.

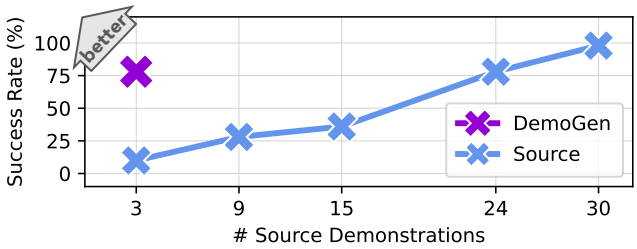

Fig. 18: **Real-world comparison between *DemoGen*-generated and human-collected datasets.** The *DemoGen*-generated dataset is based on 3 source demonstrations.

capacity is upper-bounded by the demonstrated configurations. This is in line with the findings in the empirical study.

I. Detailed Analysis of the Single-Arm Experiments

Similar to the effects of manually covering the workspace with human-collected demonstrations, *DemoGen*-generated datasets enable the agents to display a more adaptive response to diverse evaluated configurations, resulting in significantly higher success rates. *DemoGen* consistently enhances the performance across all the evaluated tasks. Although the performance gains are less pronounced in the Dex-Drill and Dex-Coffee tasks, we found the policies trained on the generated data still guide the dexterous hands to generally appropriate manipulation poses. The relatively lower performance is primarily due to stringent precision requirements.

To further investigate the generalization capabilities enabled by *DemoGen*, we visualize the spatial heatmaps for the evaluated configurations in Fig. 19. The heatmaps reveal high success rates on configurations close to the demonstrated ones, while the performance diminishes as the distance from the demonstrated configuration increases. We attribute this decline to the visual mismatch problem caused by single-view observations, as discussed in Sec. C2.

A notable observation arises in the Dex-Rollup task, where the policy trained on the *DemoGen*-generated dataset could dynamically adjust the number of wrapping motions ranging from 2 to 5 in response to the distinct plasticity of every hand-molded piece of plasticine. This suggests the usage of *DemoGen* is not in conflict with the resulting agent's closed-loop re-planning capability. The intrinsic strength of visuomotor policies is effectively preserved.

J. Detailed Analysis of the Bimanual Humanoid Experiment

The orientational augmentations share the same visual mismatch problem as translational augmentation. The policy performs as expected when the generated orientations are close to the orientation in the source demonstration. As the orientational difference increases, we observed the policy might react to the orientation in the current visual observation with actions for mismatched orientations.

Additionally, we found the spatial generalization problem persists in mobile manipulation scenarios. This is mainly due to the physical constraints of real-world environments, such as kitchen countertops or fruit stands, as demonstrated in our experiments, where terrain limitations prevent the base from approaching objects at arbitrary distances. Consequently, the base typically moves to a fixed point at a specific distance from the object, after which the robot conducts a standard non-mobile manipulation process at the fixed base position.

K. Disturbance Resistance Experiments Details

1) Evaluation Metrics: The sauce coverage score is computed as follows. First, we distinguish between green background and red sauce in the HSV color space. The identified background is set to black, the sauce is set to red, and the rest which should be the uncovered crust is set to white. Second, due to the highlights on the sauce liquid, some small fragmented points of the sauce may be identified as the crust. To address this, we apply smoothing filtering followed by dilation and erosion, where the kernel size is 9×9. Finally, the coverage is calculated as the ratio of red areas (sauce) over non-black areas (sauce + uncovered crust).

2) Raw Evaluation Results: For quantitative evaluation, we perform 5 repetitions for each of the 5 disturbance directions, resulting in 25 trials for both strategies.

L. Randomization Ranges for Simulated Tasks

In Fig. 13, we illustrated the simulated tasks for the evaluation on spatial generalization. To strengthen the sig-

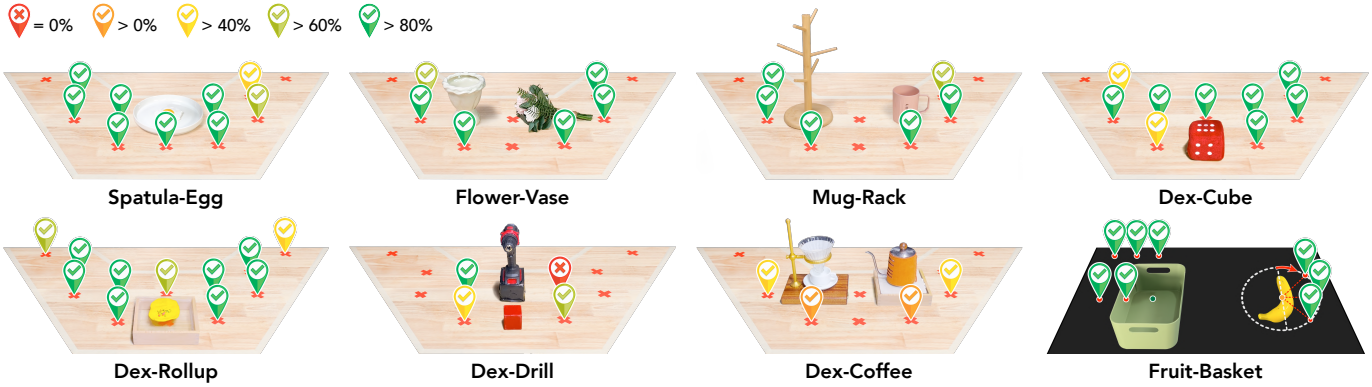

Fig. 19: **Spatial heatmaps for the real-world evaluation results.** The success rate for each coordinate is calculated as the average across all relevant trials. For example, each coordinate of the vase in the Flower-Vase task is in combination with 4 coordinates of the flower, including the one appearing in the source demonstration. This results in a total of 20 trials, given 5 repetitions per combination.

TABLE VII: **Object randomization ranges in simulated tasks.** All the reported sizes have the units in centimeters.

	Pick-Cube	Button-Small	Drawer-Close	Faucet-Open	Handle-Press	Box-Lid	Stack-Cube	Assembly
Object(s)	Cube	Button	Drawer	Faucet	Toaster	Box × Lid	Red × Green	Pillar × Hole
Evaluation	40×40	40×40	15×15	30×30	20×30	$(2.5\times30)^2$	$(15\times15)^2$	$(10\times30)^2$
DemoGen	48×48	48×48	20×20	40×40	25×40	$(7.5\times40)^2$	$(20\times20)^2$	$(15\times40)^2$

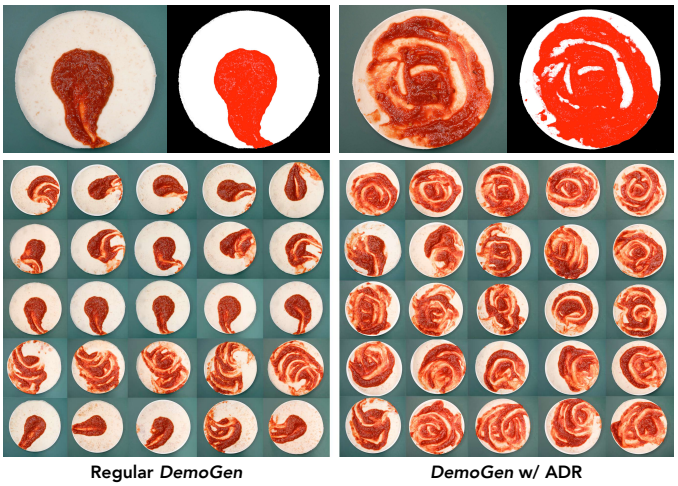

Fig. 20: **Raw evaluation results in the Sauce-Spreading task.** (Top) Examples of the processing results for metric calculation. (Bottom) Compared with the regular *DemoGen*, the policy trained with the ADR strategy better spreads the sauce to cover the crust under external disturbance.

nificance of spatial generalization, we enlarge the original object randomization ranges in the MetaWorld [54] tasks. For demonstration generation, we select a slightly larger range than the evaluation workspace to avoid performance degradation near the workspace boundaries. The detailed workspace sizes are listed in Tab. VII.

M. Task Descriptions for Real-World Tasks

In Fig. 4, we illustrated the real-world tasks for the evaluation on spatial generalization. We describe these tasks in the text as follows, where we mark the verbs for motion and skill actions in the corresponding colors.

1) **Spatula-Egg.** The gripper holds a spatula in hand. The robot maneuvers the spatula to first move toward the fried egg and then 1) slide beneath the egg, 2) lift the egg leveraging the contact with the plate's rim, 3) carry the egg and maintain stable suspension.
2) **Flower-Vase.** The gripper moves toward the flower, picks it up, reorients it in the air while transferring toward the vase, and finally inserts it into the vase.
3) **Mug-Rack.** The gripper moves toward the mug, picks it up, reorients it in the air while transferring toward the rack, and hangs it onto the rack.
4) **Dex-Cube.** The dexterous hand moves toward the cube and grasps up the cube.
5) **Dex-Rollup.** The dexterous hand moves toward a piece of plasticine and wraps it multiple times until it is fully coiled. The required times of the wrapping motion may vary due to the distinct plasticity of every hand-molded piece of plasticine.
6) **Dex-Drill.** The dexterous hand moves toward the drill, grasps it up, transfers it toward the cube, and finally touches the cube with the drill.
7) **Dex-Coffee.** The dexterous hand moves toward the kettle, grasps it up, transfers it toward the coffee filter, and finally pours water into the filter.

N. Visualization of DemoGen-Generated Trajectories

In Fig. 21, we gave a concrete example of the trajectory of synthetic visual observations. We provide more examples in Fig. 22 by showcasing the key frames of source and generated demonstrations.

O. Limitation and Future Works

Limitations. Although we have demonstrated the effectiveness of *DemoGen*, it has several limitations. First, the performance

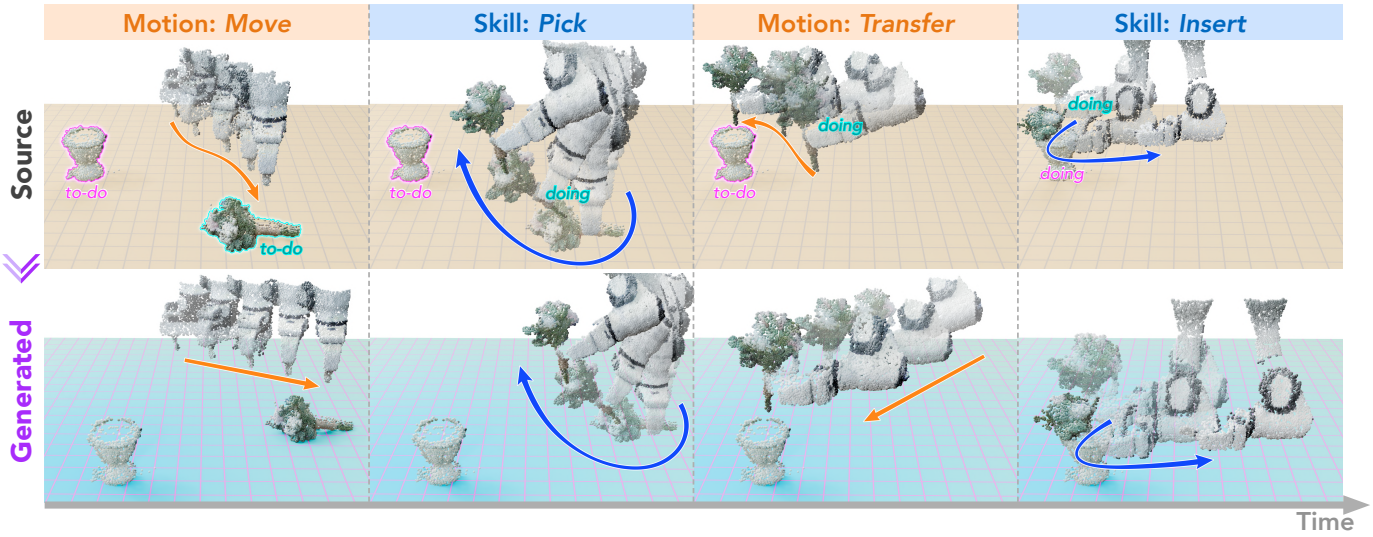

Fig. 21: **Illustrations for synthetic visual observation generation.** Objects in the *to-do* stage are segmented and transformed by the target object configurations. Objects in the *doing* stage are merged with the end-effector and transformed according to the proprioceptive states.

of *DemoGen* is affected by the visual mismatch problem, as discussed in Sec. C2. Second, *DemoGen* relies on the availability of segmented point clouds, which limits its applicability in highly cluttered or unstructured environments, as discussed in Sec. IV-B. Third, *DemoGen* is not suitable for tasks where spatial generalization is not involved, such as in-hand reorientation [2] or push-T [13, 5] with a fixed target pose. Fourth, while *DemoGen* improves the data-efficiency from an overall policy learning perspective, it does not enhance the data-efficiency of visuomotor policies. Fifth, even though *DemoGen* dramatically relieves the burden of data collection, training visuomotor policies may not be as efficient as direct TAMP-based imitation for tasks with a static scene, e.g., rigid-body pick-and-place tasks, as discussed in Sec. D.

Future works. Future works could explore mitigating the impact of visual mismatch, potentially by leveraging techniques such as contrastive learning or 3D generative models. Besides, the generation pipeline can be combined with finer-grained 3D modeling approaches to benefit the manipulation of articulated objects. Another avenue for future research is to use additional human-collected demonstrations as source data, aiming to identify the optimal balance between policy performance and the overall cost of data collection.

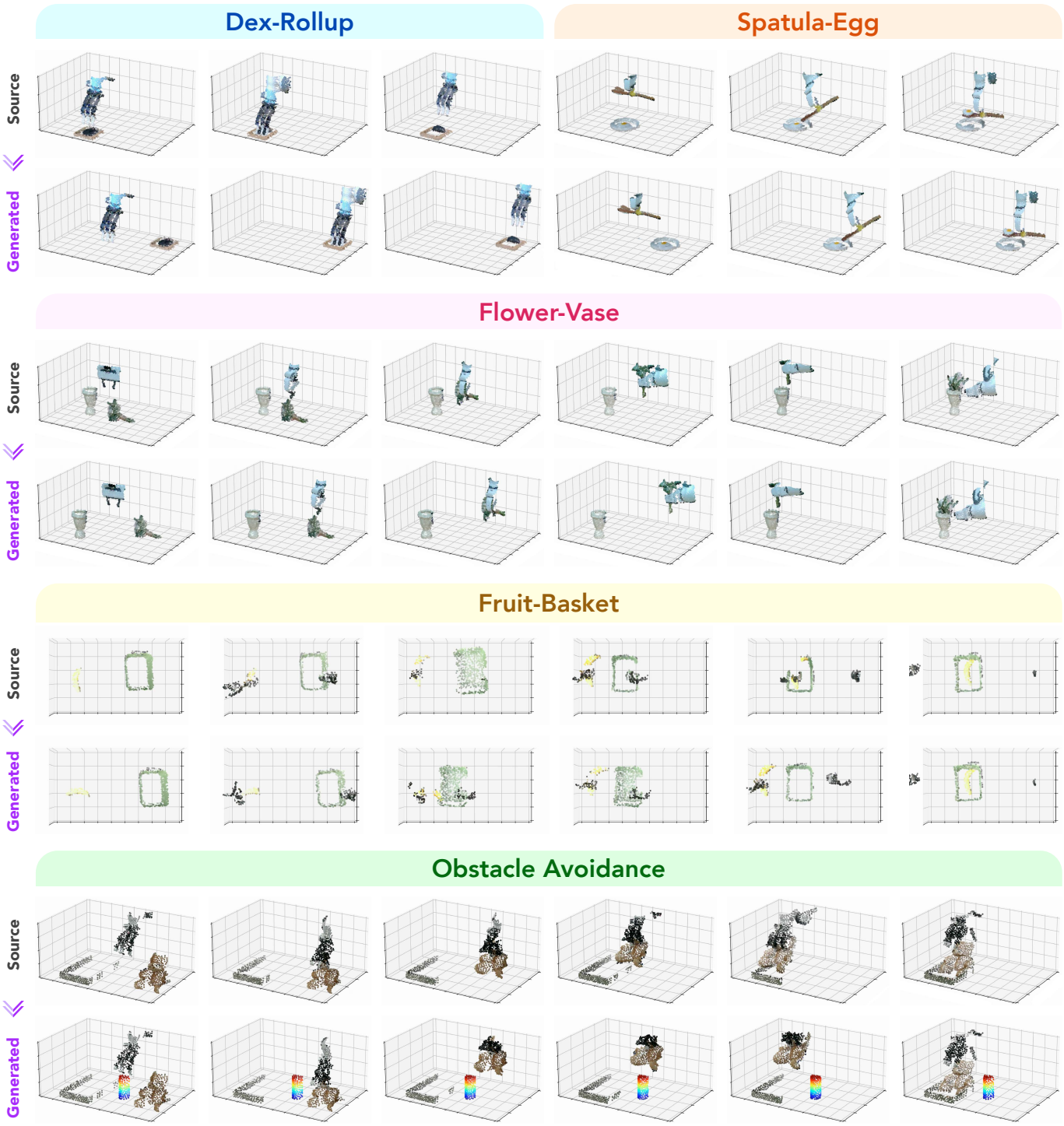

Fig. 22: **More examples of the trajectories consisting of synthetic visual observations.**