DexArt: Benchmarking Generalizable Dexterous Manipulation with Articulated Objects

Chen Bao[1*] Helin Xu[2*] Yuzhe Qin[3] Xiaolong Wang[3]
[1]Shanghai Jiao Tong University [2]Tsinghua University [3]UC San Diego

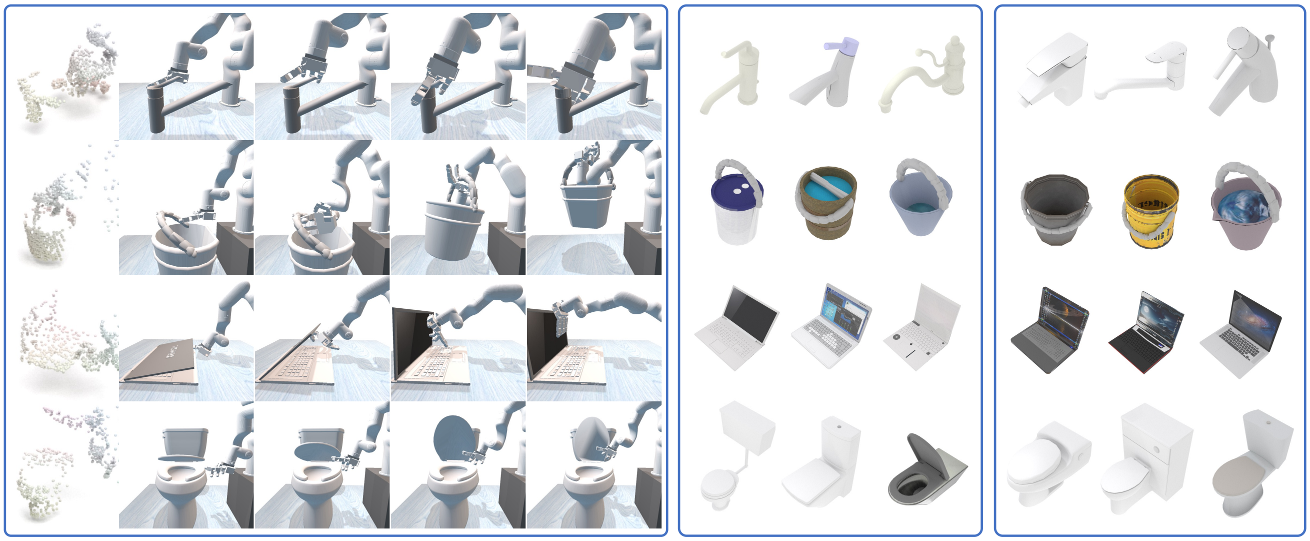
(a) DexArt Task Suite (b) Seen Objects (c) Unseen Objects

Figure 1. **Overview.** (a) We propose **DexArt**, a task suite of **Dex**terous manipulation with **Art**iculated object using point cloud observation. (b) We experiment with extensive benchmark methods that learn category-level manipulation policy on seen objects. (c) We evaluate the policies' generalizability on a collection of unseen objects, as well as their robustness to camera viewpoint change.

Abstract

To enable general-purpose robots, we will require the robot to operate daily articulated objects as humans do. Current robot manipulation has heavily relied on using a parallel gripper, which restricts the robot to a limited set of objects. On the other hand, operating with a multi-finger robot hand will allow better approximation to human behavior and enable the robot to operate on diverse articulated objects. To this end, we propose a new benchmark called DexArt, which involves Dexterous manipulation with Articulated objects in a physical simulator. In our benchmark, we define multiple complex manipulation tasks, and the robot hand will need to manipulate diverse articulated objects within each task. Our main focus is to evaluate the generalizability of the learned policy on unseen articulated objects. This is very challenging given the high degrees of freedom of both hands and objects. We use Reinforcement Learning with 3D representation learning to achieve generalization. Through extensive studies, we provide new insights into how 3D representation learning affects decision making in RL with 3D point cloud inputs. More details can be found at https://www.chenbao.tech/dexart/.

1. Introduction

Most tools and objects humans interact with are articulated objects. To allow household robots to facilitate our daily life, we will need to enable them to manipulate diverse articulated objects with multi-finger hands as humans do. However, learning dexterous manipulation remains a challenging task given the high Degree-of-Freedom (DoF) joints of the robot hands. While recent work has shown encouraging progress in using Reinforcement Learning (RL) [1, 8, 29, 69] for dexterous manipulation, most research focuses on manipulating a single rigid object. The manipulation of diverse articulated objects not only adds

[*] Equal contributions. Work done while interning at UC San Diego.

additional complexity with joint DoF, but also brings new challenges in generalizing to unseen objects in test time, which has been a major bottleneck for RL. This requires efforts on integrating 3D visual understanding and robot learning on a novel benchmark.

Recent proposed robotic manipulation benchmarks [7, 12, 34, 65] play important roles in robot learning algorithm development. For example, the MetaWorld [65] benchmark provides more than 50 tasks for evaluating RL algorithms. However, each proposed MetaWorld task only focuses on one single object without considering generalization across object instances. To enable generalizability for the robots, the ManiSkill [19, 41] benchmark is proposed with diverse manipulation tasks and a large number of objects to manipulate within each task. While this is encouraging, the use of a parallel gripper has limited the tasks the robot can perform, and the ways how the robot can operate. For example, it is very challenging for a parallel gripper to pick up a bucket using the handle.

In this paper, we propose a new benchmark for **Dex**terous manipulation with diverse **Art**iculated objects (**DexArt**). We introduce multiple tasks with a dexterous hand (the Allegro Hand) manipulating the articulated objects in the simulation. For each task, instead of operating with a particular object, we provide a training set of diverse articulated objects and the goal is to generalize the policy to a different test set of articulated objects. To achieve such a generalization, we incorporate RL with generalizable visual representation learning: we adopt 3D point clouds as our observations and use a PointNet encoder [44] to extract visual representations for decision making. The generalizability of the policy depends on the 3D structure understanding modeled by the PointNet encoder. We experiment and benchmark with different methods and settings, and provide four key observations as follows:

(i) Training with more objects leads to better generalization. For each task, we trained policies using varying numbers of objects for each task and tested them on the same set of unseen objects. We find training with more objects consistently achieves better success rates. Similar findings have been reported in studies on manipulation with parallel grippers (Generalist-Specialist Learning [24], ManiSkill [41]). While this might not be surprising from the perception perspective, it does present more challenges for a single RL policy to work with different objects simultaneously. It highlights the importance of learning generalizable visual representations for RL.

(ii) Encoder with a larger capacity does not necessarily help. We experiment with different sizes of PointNet encoders, and we observe the simplest one with the least parameters achieves the best sample efficiency and success rate, whether the network is pre-trained or not. This is surprising from the vision perspective, but it is consistent with previous literature which shows RL optimization becomes much more challenging with large encoders [41].

(iii) Object part reasoning is essential. With multi-finger hand interacting with different object parts, our intuition is that object part recognition and reasoning can be essential for manipulation. To validate our intuition, we pre-train the PointNet encoder with object part segmentation tasks. We show the object part pre-training can significantly improve sample efficiency and success rate compared to approaches without pre-training and with other pre-training methods.

(iv) Geometric representation learning brings robust policy. We evaluate the robustness of the policy under unseen camera poses. We find that the policy trained with partial point cloud is surprisingly resilient to variations in camera poses, which aligns with the previous studies that use complete point clouds in policies [32]. The accuracy remains consistent even with large viewpoint variation. This is particularly useful for real robot applications as it is challenging to align the camera between sim and real.

With the proposed baselines and detailed analysis among them, we hope DexArt benchmark provides a platform to not only study generalizable dexterous manipulation skill itself, but also study how visual perception can be improved to aim for better decision making. We believe the unification of perception and action, and studying them under DexArt can create a lot of research opportunities.

2. Related Work

Dexterous Manipulation. Dexterous manipulation with multi-fingered robotic hands has been a long standing problem in robotics. Previous methods formulate dexterous manipulation as a planning problem [3, 5, 13, 21, 43, 51] and solve it with trajectory optimization [28, 39, 58]. These methods require well-tuned dynamics model for the robot and the manipulated object, which limits their generalizability. On the other hand, data-driven-based methods do not assume a pre-built model. The policies are learned either from demonstrations using imitation learning [10, 20, 26, 48–50, 63, 68] or from interaction data using reinforcement learning [1, 8, 23, 29, 69]. However, most methods focus on tasks with single-body objects like grasping or in-hand manipulation. Dexterous manipulation on articulated objects remains a challenging problem. In this paper, we propose a new benchmark on learning generalizable manipulation policy on articulated objects with point cloud observations.

Articulated Object Manipulation. The ability to perceive and manipulate articulated objects is of vital significance for domestic robot. There have been a lot of recent advancement on perception of articulated objects such as pose estimation and tracking [30, 33, 57], joint parameter prediction [25, 40, 56, 67], part segmentation [15, 38, 64], and dynamics property estimation [22]. On the robotics side, previous works [11, 52] also explore model-based control and planning for articulated object. A natural extension is to combine both lines of research by first estimating the articulated object model with perception algorithm and then manipulating it with model-based control [35, 59]. Another line of research bypasses the state and model estimation by directly learning the actionable information from raw sensory input [36, 62]. However, these approaches define a single-step action representation and execute it with pre-defined controllers in an open-loop manner. Different from these approaches, we formulate articulated object manipulation as a sequential decision making process where visual feedback is used in closed-loop control. During policy learning, we also study how 3D articulated object representation learning can help decision making.

Learning from Point Clouds. Point cloud learning has been a long-last research topic in 3D vision. The pioneer architectures for point cloud, e.g. PointNet [44, 45], SSCNs [18] have been widely used for geometric representation learning in part segmentation [15, 38, 64] and 3D reconstruction [15, 25] tasks. In robotics, the learned point cloud representation also facilities down-stream manipulation tasks, e.g. grasp proposal [6, 31, 46, 55], manipulation affordance [27, 36, 37], and key points [16]. Recently, researchers have explored to use point cloud as the direct input observation for RL policy [8, 23, 41, 60]. Inspired by these works, our DexArt benchmark introduces new tasks using a multi-finger hand to operate articulated objects. It is more challenging compared to previous environments given the high DoF for both the manipulator and the object. To tackle these tasks, we perform extensive experiments on how geometric representation learning (e.g., part reasoning) can affect decision making, which has not been thoroughly studied before.

3. DexArt Benchmark

We propose the DexArt benchmark which contains tasks with different levels of difficulty. It can be used to evaluate the sample efficiency and generalizability of different policy learning methods. In this work, we provide four dexterous manipulation tasks, Faucet, Bucket, Laptop and Toilet, each with a number of seen and unseen objects (see Table 1).

3.1. Task Description

Faucet. As shown in the first row of Figure 1, a robot is required to turn on a faucet with a revolute joint. The robot hand needs to firmly grasp the handle and then rotate it by around 90 degrees. This task evaluates the coordination between the motion of both dexterous hand and arm. While a 2-jaw parallel gripper can potentially perform this task, it heavily relies on precise arm motion due to its low DoF end-effector. The evaluation criteria are based on the rotated angle of the handle.

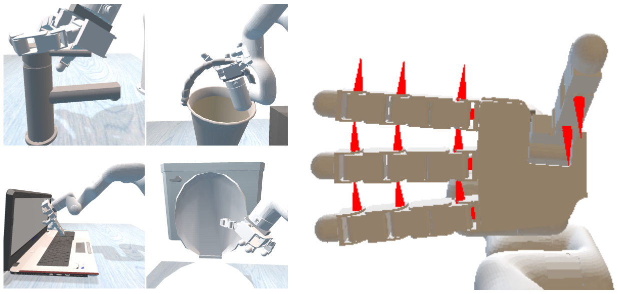

Figure 2. **Tasks and Dexterous Hand.** Left: visualization for all four tasks in DexArt Benchmark. Right: visualization of Allegro hand where red arrows indicate the revolute joint positions.

Task	objects		
	All	Seen	Unseen
Faucet	18	11	7
Bucket	19	11	8
Laptop	17	11	6
Toilet	28	17	11

Table 1. **Task Statistics.**

Bucket. As shown in the second row of Figure 1, this task requires the robot to lift a bucket up. To ensure stable lifting behavior, the robot should stretch out its hand under the bucket handle and hold it to construct a form closure [4]. On the contrary, a single parallel gripper can only grasp it with force closure [2], which can hardly achieve success without sufficiently large friction. In evaluation, this task is considered a success if the bucket is lifted to a given height.

Laptop. As shown in the third row of Figure 1, in this task, a robot should grasp the middle of the screen and then open the laptop lid. This task also fits dexterous hand well. A parallel gripper can do this by precisely plugging the lid between its jaws. However, this constraint increases the difficulty for arm motion and requires a larger workspace to open the lid. This task is evaluated based on the changed angle of laptop lid.

Toilet. As shown in the fourth row of Figure 1, the task is similar to the Laptop task, where the robot needs to open a larger toilet lid. The task is harder as the geometry of the lid is more irregular and diverse. The task is successfully solved if the toilet lid is opened at a threshold degree.

3.2. Environment Setup

In our benchmark, we implement our tasks in SAPIEN physical simulator [61] using a XArm6 robot arm (6 DoF) with an anthropomorphic hand, Allegro Hand (16 DoF).

Preliminaries. We model our control problem with dexterous hand as a Markov Decision Process (MDP), $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{T}, \rho_0, \gamma\}$, where $\mathcal{S} \in \mathbb{R}^n$, $\mathcal{A} \in \mathbb{R}^m$ stand for state and actions respectively. $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the reward function that measures the task progress, where human knowledge is often incorporated to guide the accomplishment of challenging tasks. $\mathcal{T} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ is the transition dynamics. ρ_0 is the initial probability distribution and $\gamma \in [0, 1)$ is the discount factor.

Observation Space. The observation consists of two parts. First, the proprioceptive data S_r includes the current joint position of the whole robot, linear velocity, angular velocity, position and pose of the end-effector palm. Second, the partial point cloud P_o captured by a depth camera includes the articulated object and the robot. The observed point cloud is first cropped within the robot workspace and then down-sampled uniformly. We also concatenate the observed point cloud P_o with an imaged robot point cloud P_i (see Section 4.1). All these observations are easily accessible for real-world robots and no oracle information is used.

Action Space. The action is a 22-dimensional vector that consists of two parts, 6-DoF for arm and 16-DoF for hand. We use an operational space control for robot arm where the first 6-D vector is the target linear and angular velocity of the palm. For Allegro hand, we use a joint position controller to command the position target of 16 joints. Both controllers are implemented by PD control.

3.3. Reward Design

The reward design for all dexterous manipulation tasks follows three principles. (i) To ensure each task is solvable in a reasonable amount of time, a dense reward is required. (ii) To eliminate unexpected behavior, the reward should regulate the behavior of policy to be natural (human-like) and safe. (iii) The reward structure should be general and standardized across all tasks. We decompose our tasks into three stages: reaching the functional part, constructing contact between the hand and manipulated objects, and executing task-specific actions to move the manipulated parts.

Reaching and Grasping Stage. We design a reach reward for the first two stages to encourage the robot hand to get close to the manipulated object as follows:

$$r_{\text{reach}} = \mathbf{1}(\text{stage} == 1) \min(-\|\mathbf{x}_{\text{palm}} - \mathbf{x}_{\text{object}}\|, \lambda), \quad (1)$$

where $\mathbf{1}()$ is an indicator function, $\mathbf{x}_{\text{palm}}$ and $\mathbf{x}_{\text{object}}$ is the 3D position of palm and object in the world frame, λ is a regularization term to prevent sudden surge in reward. Equation 3.3 only considers the Cartesian distance between hand and object, which may cause unexpected behavior, e.g., opening the laptop lid with a clenched motion rather pushing the side of the lid with hand. In the real world, such motion may cause damage to the manipulated object and robot itself. Inspired by [47], we add a contact term to encourage better contact between fingers and object:

$$\begin{aligned} r_{\text{contact}} &= \mathbf{1}(\text{stage} \geq 2)\ \text{IsContact}\ (\text{ palm, object })\\ &\textbf{AND} \left(\sum_{\text{finger}} \text{IsContact}\ (\text{ finger, object }) \geq 2 \right), \end{aligned} \quad (2)$$

where IsContact is a boolean function that performs collision detection to check whether two links are in contact. We believe a good contact relationship is constructed if both the palm and at least two fingers touch the object.

Part Manipulation Stage. In the last stage, the robot needs to manipulate the specific part of an articulated object to move it to the given pose. The reward of this stage is designed as follows:

$$r_{\text{progress}} = \mathbf{1}(\text{stage} == 3)\text{Progress(task)}, \quad (3)$$

where Progress is a task-specific evaluation function for the current task progress. For example, in the Faucet environment, we use the change of handle joint angle to indicate task progress.

To eliminate jerky and unstable robot motion, we add a penalty term r_{penalty}, which includes a L2 norm of the action and a task-specific term. The overall reward is the weighted sum of four reward terms. More details on the reward design can be found in our supplementary material.

3.4. Asset Selection and Annotation

We use the articulated object models from the PartNet-Mobility [61] dataset. We manually select object models for each task to avoid bad modeling and to ensure a consistent kinematics structure. We further annotate the scale and initial positions object-by-object to make sure they have reasonable sizes and don't initially intersect with the robot. We further apply randomness to the object initial position, *i.e.* for each task, we perform reasonable rotation and translation from the annotated position while making sure the goal is still achievable, and the object is randomized from the training set during policy learning.

4. Method

Solving dexterous manipulation tasks with RL methods suffers from high sample complexity due to high-dimensional action space. Tasks with *articulated objects* and *point cloud observation* increase the complexity further. In this section, we will discuss several methods for improving policy learning performance. In Section 4.1, we will talk about the policy learning architecture. Section 4.2 will describe details on how to generate the data for visual pre-training. Finally, we will discuss the pre-training methods evaluated in our benchmark in Section 4.3.

4.1. Policy Overview

Policy Learning. To achieve category-level generalization across diverse objects, we adopt 3D point cloud as our observation and use Proximal Policy Optimization (PPO) [53] as our RL algorithm. In the architecture design, the value and policy networks share the same feature extracted from the point cloud and robot proprioception, as shown in the right part of Figure 3. We use PointNet [44] as the point cloud feature extractor. It is worth noting that

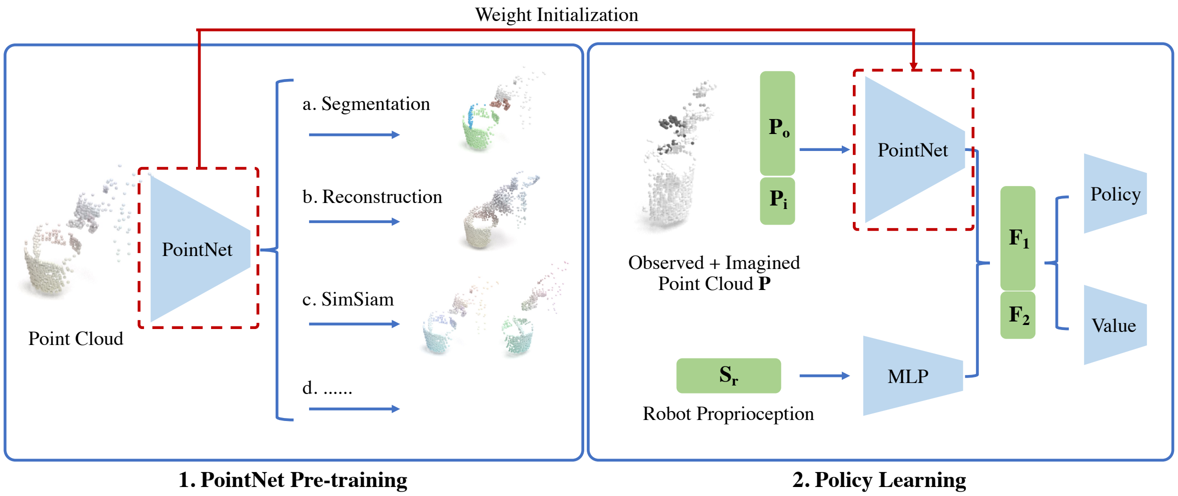

Figure 3. **Overview.** We adopt PPO algorithm with PointNet backbone to learn dexterous manipulation on articulated objects. We use pre-training to facilitate the policy learning process. (1) The PointNet is pre-trained on perception-only tasks, which includes segmentation, reconstruction, SimSiam, etc. (2) The pre-trained PointNet weight is then used to initialize the visual backbone in PPO before RL training.

we employ a simple version of PointNet. The local MLP has one hidden layer with a GELU activation function, followed by a max pooling that directly produces the output feature F_1. Meanwhile, an MLP is used to extract output feature F_2 from the robot proprioception vector S_r. The output feature F_1 and F_2 are then concatenated and passed through the value MLP and policy MLP. We show in experiments that increasing the volume of the vision extractor actually harms policy learning.

Feature Extractor Pre-training. We investigate how 3D representation learning helps with 3D policy learning. We benchmark vision pre-training with five different 3D representation learning methods, including both self-supervised learning and supervised learning, which will be discussed in Section 4.3. For all methods, we pre-train a visual model with PointNet backbone on perception-only tasks, and then use it to initialize the feature extractor for RL. The pre-training pipeline is illustrated in Figure 3.

Point Cloud Imagination. The point cloud RL has two challenges. First, the hand-object interaction will cause several occlusions. Second, the RL training can only handle low-resolution point cloud due to the memory limitation. Thus only few points in the observation come from the hand fingers, which is essential information for decision making. Inspired by [47], we leverage the robot model to compute the finger geometry via forward kinematics. We can then sample points P_i, called imagined point cloud, from the computed geometry. As shown in the right part of Figure 3, our point cloud feature extractor takes as input both observed points P_o and the imagined points P_i (deep-colored points in Figure 3). This way, we provide the missing details of the robot in point cloud observation. Note that P_i is accessible even for real robot.

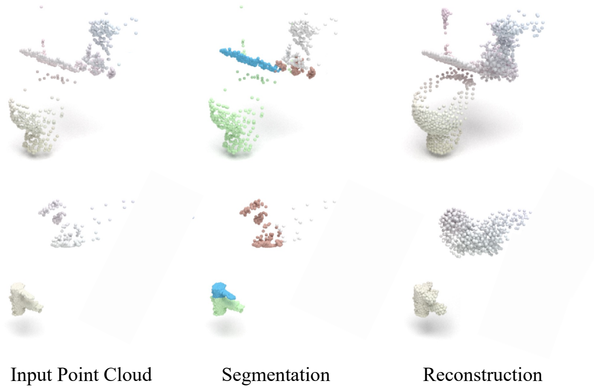

Figure 4. **Pre-training Visualizations.** We visualize the segmentation and reconstruction pre-training results on Toilet (top row) and Faucet (bottom row).

4.2. Pre-training Datasets

DexArt Manipulation Dataset (DAM). We render the point cloud observations with the setting as manipulation tasks. The dataset contains 6k point clouds for each object, including observed and imagined points, where the state of robot and articulated object are sampled randomly, as shown in the left column of Figure 4. For segmentation pre-training, we label the point cloud into 4 groups: the functional part of the object, the rest of the object, the robot hand, and the robot arm, as shown in the middle column of Figure 4.

PartNet-Mobility Manipulation Dataset (PMM). Different from DAM, PMM is directly rendered from PartNet-Mobility [61] without task information, e.g. robot. PMM contains 46 object categories and 1k point clouds for each category. The state of the object and the camera viewpoint are sampled randomly. For classification, each object in the same category shares the same label. For segmentation, we follow the procedure in [17] to generate ground truth seg-

Task	Faucet		Bucket		Laptop		Toilet	
Split	Seen	Unseen	Seen	Unseen	Seen	Unseen	Seen	Unseen
No Pre-train	0.30 ± 0.22	0.28 ± 0.21	0.51 ± 0.12	0.56 ± 0.08	0.81 ± 0.01	0.41 ± 0.09	0.71 ± 0.05	0.46 ± 0.02
Segmentation on PMM	0.27 ± 0.12	0.17 ± 0.09	0.35 ± 0.25	0.34 ± 0.24	0.85 ± 0.09	0.55 ± 0.09	0.66 ± 0.08	0.44 ± 0.02
Classification on PMM	0.20 ± 0.12	0.18 ± 0.14	0.56 ± 0.06	0.58 ± 0.12	0.80 ± 0.20	0.41 ± 0.14	0.69 ± 0.08	0.38 ± 0.03
Reconstruction on DAM	0.35 ± 0.02	0.21 ± 0.03	0.51 ± 0.08	0.50 ± 0.05	0.85 ± 0.04	0.54 ± 0.08	0.76 ± 0.03	0.52 ± 0.03
SimSiam on DAM	0.60 ± 0.15	0.45 ± 0.12	0.41 ± 0.30	0.38 ± 0.31	0.84 ± 0.04	0.49 ± 0.13	0.82 ± 0.02	0.50 ± 0.06
Segmentation on DAM	$\mathbf{0.79 \pm 0.02}$	$\mathbf{0.58 \pm 0.07}$	$\mathbf{0.75 \pm 0.04}$	$\mathbf{0.76 \pm 0.07}$	$\mathbf{0.92 \pm 0.02}$	$\mathbf{0.60 \pm 0.07}$	$\mathbf{0.85 \pm 0.01}$	$\mathbf{0.55 \pm 0.01}$

Table 2. **Success Rate of Different Pre-training Methods.** We report the success rate (mean ± std) on four tasks, for both seen and unseen objects. DAM = DexArt Manipulation Dataset, PMM = PartNet-Mobility Manipulation Dataset, as described in section 4.2.

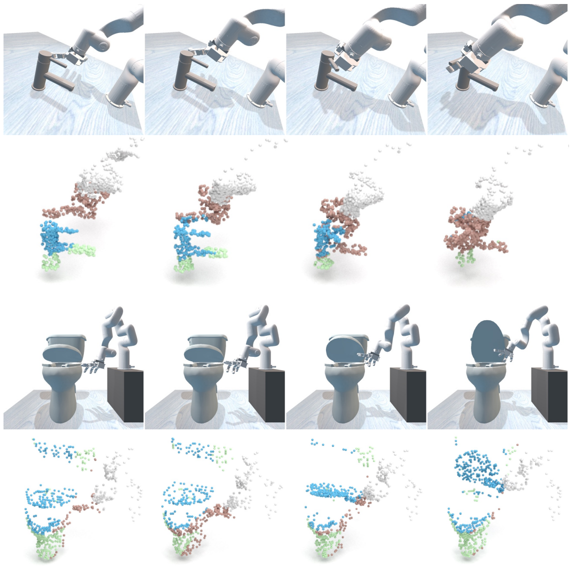

Figure 5. **Segmentation After RL Tuning.** We visualize the segmentation results after the PointNet is tuned during policy learning. The weight from PPO point cloud feature extractor can be directly applied back to the segmentation network to perform segmentation prediction.

mentation masks for functional parts on the articulated objects.

4.3. Pre-training Methods

Supervised Pre-training. We experiment with two supervised pre-training methods including semantic segmentation and classification. For classification, we train a PointNet on PMM data to predict the label for 46 object categories. Compared to simpler tasks like grasping, articulated object manipulation requires more understanding on 3D parts. The policy needs to locate the functional part and reason how to interact with it. Thus, we also investigate how pre-training on part segmentation can help policy learning. We train segmentation on both DAM and PMM.

Self-supervised Pre-training. We also experiment with two self-supervised pre-training methods, including point cloud reconstruction and SimSiam [9]. Following OcCo [54], we use an encoder-decoder architecture for point cloud reconstruction on DAM dataset. The encoder is a PointNet that extracts global embedding and the decoder is a PCN [66] which reconstructs the original point cloud from global embedding. The reconstruction is trained via Chamfer loss [14]. The reconstruction results are visualized in the right column of Figure 4. After pre-training, we use the PointNet encoder to initialize the PointNet in PPO.

We follow SimSiam and design a siamese network with PointNet. In SimSiam training, the network takes two augmented views of the same point cloud, and forwards them into the same PointNet encoder. An MLP is connected on one side to predict the similarity while the gradient is stopped on the other side. The method is trained to maximize the similarity between both sides. We pre-train the PointNet encoder inside SimSiam on the DAM dataset.

5. Experiment

We conduct experiments on the proposed tasks including Faucet, Bucket, Laptop, and Toilet defined in Section 3.1. We perform experiments on three aspects: (i) We benchmark different pre-training methods by evaluating both seen and unseen articulated objects for all tasks. We test the success rate during and after training. (ii) We ablate how the number of seen objects and the architecture size of visual backbone can affect policy learning. (iii) We study the robustness to camera viewpoint change for different methods, where we evaluate the task success rate when the input point cloud is captured by cameras at novel poses. Overall, we evaluate the methods by success rate and episodic returns on both seen objects and unseen objects. We train RL policy with 3 different random seeds for each experiment.

5.1. Main Results

We provide the success rates of all benchmark methods in Table 2. We compare the RL policy trained from scratch (1st row) with five different pre-training methods (the following rows). The results show that proper visual pre-training can benefit the policy learning. We highlight our findings as follows. (i) Part segmentation boosts the policy learning on all tasks. It performs the best on all tasks. With segmentation pre-training, the PointNet can better distinguish and locate the functional parts, which is critical for ar-

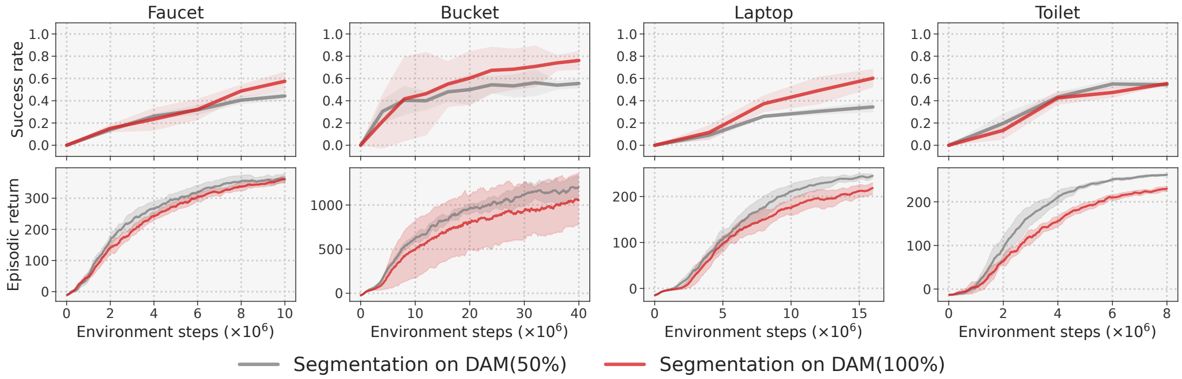

Figure 6. **Training Process with Different Number of Seen Objects.** The x-axis is the environment steps. The y-axis of the upper row is the success rate on unseen objects, evaluated with 3 random seeds, and the shaded area indicates standard deviation. The y-axis of the bottom row is the episodic return, where the shaded area represents the standard error. The grey curves show methods with segmentation pre-training on around 50% of the seen DexArt objects within each category, compared with the red curves that are pre-trained with 100%.

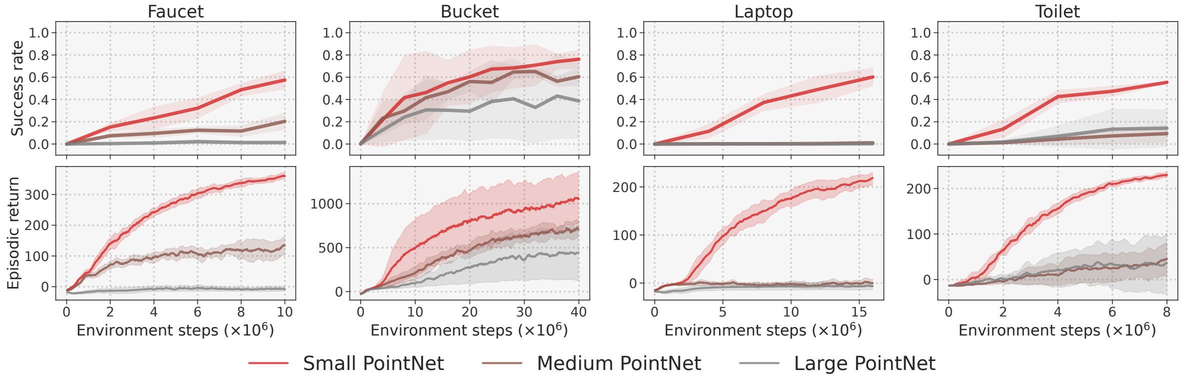

Figure 7. **Training with Different PointNet Sizes.** The axes mean the same as in Figure 6. The experiments with tree curves are segmentation pre-trained on DAM(100%), with small/medium/large PointNet described in Section 5.2.

ticulated object manipulation. (ii) Other pre-training methods to learn a representative global embedding, *i.e.*, classification, reconstruction, and SimSiam, also improve the policy learning in some cases, especially on Laptop task. (iii) Tasks that involve manipulating small functional parts, *e.g.* faucet handles, benefit more from segmentation pre-training. As shown in Figure 8, the segmentation results can predict the label of small faucet handles correctly, while reconstruction focuses more on the global shape completeness and ignores small part details. Thus, part segmentation is a more effective pre-training method for more delicate manipulation tasks.

5.2. Ablation Study

We ablate how the number of objects used in training, the size of the vision extractor, and different visual representation learning methods influence the generalizability.

Number of Seen Objects. Different from the previous experiment in Section 5.2, we train our model and policy using only 50% of the seen objects. We report the learning curve and the success rate on novel objects during training for methods using 50% and 100% of the objects. In Figure 6, while the convergence speed (represented by episodic return) with 100% seen objects is slower compared with the 50% one due to more diverse object geometry, the success rate (top row) of the 100% training on unseen objects remains higher during the whole training process and for all tasks. It demonstrates more training objects are crucial for better policy generalizability.

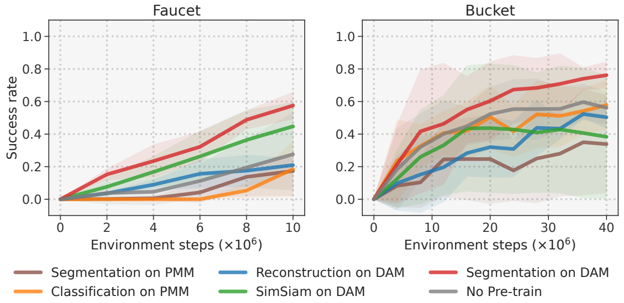

Figure 8. **Different Pre-train Methods.** Evaluation success rate of different methods in Faucet and Bucket tasks. The shaded area indicates the standard deviation.

Size of the Vision Extractor. We experiment with three different sizes for PointNet: (i) The small PointNet with one hidden layer. (ii) The medium PointNet with three hidden layers. (iii) The large PointNet with five hidden layers. All other components for these three PointNet are the same. Surprisingly, we find that the smallest PointNet achieves the best performance for both success rate and episodic return, as shown in Figure 7. Different from our common understanding from vision perspective, the smaller network not only trains faster but also generalizes better.

Non-3D Representation. We compare our PointNet pre-trained on DAM segmentation in Laptop task with following 2D pre-training representation : R3M [42]. In R3M, the Ego4D human video dataset was used to pre-train a ResNet-18 with time-contrastive learning and video-language alignment. Table 3 shows the results of the experiment. The results indicate that 3D visual representation learning with PointNet is better at manipulating objects. Compared to Non-3D representation learning, 3D policies can achieve better manipulation performance on both seen and unseen objects.

Encoder	Seen	Unseen
PointNet	0.78 ± 0.04	0.41 ± 0.08
ResNet-18	0.64 ± 0.07	0.28 ± 0.05

Table 3. **Non-3D Representations.**

5.3. Robustness to Viewpoint Change

We experiment with the viewpoint change of camera in Laptop task to evaluate the robustness of policy based on PointNet and ResNet-18. The PointNet policy is pre-trained on DAM segmentation and the ResNet-18 policy is pre-trained on R3M. The viewpoint sampling procedure can be described as follow: (i) Determine a semi-sphere for camera pose sampling. We first compute the radius r of the semi-sphere using the distance from the initial camera position to the manipulated object. The center of this semi-sphere is defined by moving along the camera optical line with distance r. (ii) Sample a point on the semi-sphere as camera position. We uniformly sample the azimuthal angle in every 20° from $-60°$ to 60° and polar angle in every 20° from $-20°$ to 20°, relative to the training viewpoint. It results in $7 \times 5 = 35$ camera positions in total. (iii) Rotate the camera so that it points to the semi-sphere center. Using the procedure above, we sample 35 camera poses. We set these camera poses during the inference.

As shown in Figure 9, the trained PointNet policy shows great robustness against viewpoint change, even though we change the azimuthal angle by 60° and the polar angle by 20°. By contrast, the success rate of the ResNet-18 policy suffers dramatically drop when the difference between the training viewpoint and the novel evaluation viewpoint increases. It informs us that the robustness mainly comes from point cloud representation learning and PointNet architecture.

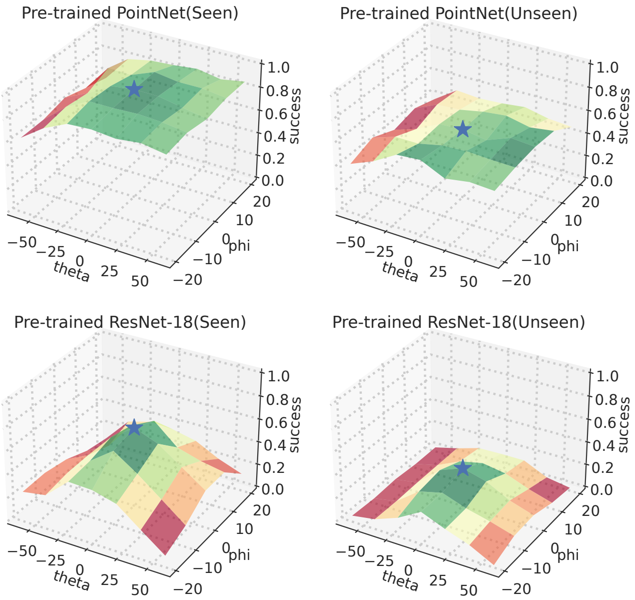

Figure 9. **Success Rate under Different Viewpoints.** The x-axis is the polar angle ϕ (relative to the training viewpoint) and the y-axis is the azimuthal angle θ on the semi-sphere centered at the object. The z-axis represents the success rate. The viewpoint during training is highlighted by a blue star.

6. Conclusion

We propose a new benchmark for dexterous manipulation with articulated objects, and study the generalizability of the RL policy. We experiment and benchmark with different methods to provide several insights: (i) RL with more diverse objects leads to better generalizability. We find that training with more objects leads to consistently better performance on unseen objects. (ii) Large encoders may not be necessary for RL training to perform dexterous manipulation tasks. We find that, in all environments, the simplest PointNet always leads better sample efficiency and best generalizability. (iii) 3D visual understanding helps policy learning. Part-segmentation facilities manipulation with small functional parts while tasks with larger functional parts benefit from all visual pre-training methods. (iv) Geometric representation learning with PointNet feature extractor brings strong robustness to the policy against camera viewpoint change. In conclusion, we hope DexArt can serve as a platform to study generalizable dexterous manipulation, and the joint improvement between perception and decision making.

Acknowledgements. We thank Hao Su for providing valuable feedback to the paper. This project was supported, in part, by the Industrial Technology Innovation Program (20018112, Development of autonomous manipulation and gripping technology using imitation learning based on visualtactile sensing) funded by the Ministry of Trade Industry and Energy of the Republic of Korea, Amazon Research Award and gifts from Qualcomm.

References

[1] OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020. 1, 2

[2] Antonio Bicchi. On the closure properties of robotic grasping. *The International Journal of Robotics Research*, 14(4):319–334, 1995. 3

[3] Antonio Bicchi and Vijay Kumar. Robotic grasping and contact: A review. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 348–353. IEEE, 2000. 2

[4] Antonio Bicchi and Vijay Kumar. Robotic grasping and contact: A review. In *Proceedings 2000 ICRA. Millennium conference. IEEE international conference on robotics and automation. Symposia proceedings (Cat. No. 00CH37065)*, volume 1, pages 348–353. IEEE, 2000. 3

[5] Jeannette Bohg, Antonio Morales, Tamim Asfour, and Danica Kragic. Data-driven grasp synthesis—a survey. *IEEE Transactions on robotics*, 30(2):289–309, 2013. 2

[6] Samarth Brahmbhatt, Ankur Handa, James Hays, and Dieter Fox. Contactgrasp: Functional multi-finger grasp synthesis from contact. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2386–2393. IEEE, 2019. 3

[7] Yu-Wei Chao, Chris Paxton, Yu Xiang, Wei Yang, Balakumar Sundaralingam, Tao Chen, Adithyavairavan Murali, Maya Cakmak, and Dieter Fox. Handoversim: A simulation framework and benchmark for human-to-robot object handovers. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6941–6947. IEEE, 2022. 2

[8] Tao Chen, Jie Xu, and Pulkit Agrawal. A system for general in-hand object re-orientation. In *Conference on Robot Learning*, pages 297–307. PMLR, 2022. 1, 2, 3

[9] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021. 6

[10] Zoey Qiuyu Chen, Karl Van Wyk, Yu-Wei Chao, Wei Yang, Arsalan Mousavian, Abhishek Gupta, and Dieter Fox. Dextransfer: Real world multi-fingered dexterous grasping with minimal human demonstrations. *arXiv preprint arXiv:2209.14284*, 2022. 2

[11] Sachin Chitta, Benjamin Cohen, and Maxim Likhachev. Planning for autonomous door opening with a mobile manipulator. In *2010 IEEE International Conference on Robotics and Automation*, pages 1799–1806. IEEE, 2010. 2

[12] Sudeep Dasari, Jianren Wang, Joyce Hong, Shikhar Bahl, Yixin Lin, Austin Wang, Abitha Thankaraj, Karanbir Chahal, Berk Calli, Saurabh Gupta, et al. Rb2: Robotic manipulation benchmarking with a twist. *arXiv preprint arXiv:2203.08098*, 2022. 2

[13] Mehmet R Dogar and Siddhartha S Srinivasa. Push-grasping with dexterous hands: Mechanics and a method. 2010. 2

[14] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017. 6

[15] Samir Yitzhak Gadre, Kiana Ehsani, and Shuran Song. Act the part: Learning interaction strategies for articulated object part discovery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15752–15761, 2021. 2, 3

[16] Wei Gao and Russ Tedrake. kpam-sc: Generalizable manipulation planning using keypoint affordance and shape completion. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6527–6533. IEEE, 2021. 3

[17] Haoran Geng, Helin Xu, Chengyang Zhao, Chao Xu, Li Yi, Siyuan Huang, and He Wang. Gapartnet: Cross-category domain-generalizable object perception and manipulation via generalizable and actionable parts, 2022. 5

[18] Benjamin Graham, Martin Engelcke, and Laurens Van Der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9224–9232, 2018. 3

[19] Jiayuan Gu, Fanbo Xiang, Xuanlin Li, Zhan Ling, Xiqiaing Liu, Tongzhou Mu, Yihe Tang, Stone Tao, Xinyue Wei, Yunchao Yao, Xiaodi Yuan, Pengwei Xie, Zhiao Huang, Rui Chen, and Hao Su. Maniskill2: A unified benchmark for generalizable manipulation skills. In *International Conference on Learning Representations*, 2023. 2

[20] Abhishek Gupta, Clemens Eppner, Sergey Levine, and Pieter Abbeel. Learning dexterous manipulation for a soft robotic hand from human demonstrations. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3786–3793. IEEE, 2016. 2

[21] Li Han and Jeffrey C Trinkle. Dexterous manipulation by rolling and finger gaiting. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 730–735. IEEE, 1998. 2

[22] Eric Heiden, Ziang Liu, Vibhav Vineet, Erwin Coumans, and Gaurav S Sukhatme. Inferring articulated rigid body dynamics from rgbd video. *arXiv preprint arXiv:2203.10488*, 2022. 2

[23] Wenlong Huang, Igor Mordatch, Pieter Abbeel, and Deepak Pathak. Generalization in dexterous manipulation via geometry-aware multi-task learning. *arXiv preprint arXiv:2111.03062*, 2021. 2, 3

[24] Zhiwei Jia, Xuanlin Li, Zhan Ling, Shuang Liu, Yiran Wu, and Hao Su. Improving policy optimization with generalist-specialist learning. In *International Conference on Machine Learning*, pages 10104–10119. PMLR, 2022. 2

[25] Zhenyu Jiang, Cheng-Chun Hsu, and Yuke Zhu. Ditto: Building digital twins of articulated objects from interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5616–5626, 2022. 2, 3

[26] Edward Johns. Coarse-to-fine imitation learning: Robot manipulation from a single demonstration. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4613–4619. IEEE, 2021. 2

[27] David Inkyu Kim and Gaurav S Sukhatme. Semantic labeling of 3d point clouds with object affordance for robot manipulation. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5578–5584. IEEE, 2014. 3

[28] Vikash Kumar, Emanuel Todorov, and Sergey Levine. Optimal control with learned local models: Application to dexterous manipulation. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 378–383. IEEE, 2016. 2

[29] Sergey Levine, Nolan Wagener, and Pieter Abbeel. Learning contact-rich manipulation skills with guided policy search. In *IEEE International Conference on Robotics and Automation, ICRA*, pages 156–163. IEEE, 2015. 1, 2

[30] Xiaolong Li, He Wang, Li Yi, Leonidas J Guibas, A Lynn Abbott, and Shuran Song. Category-level articulated object pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3706–3715, 2020. 2

[31] Hongzhuo Liang, Xiaojian Ma, Shuang Li, Michael Görner, Song Tang, Bin Fang, Fuchun Sun, and Jianwei Zhang. Pointnetgpd: Detecting grasp configurations from point sets. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3629–3635, 2019. 3

[32] Minghua Liu, Xuanlin Li, Zhan Ling, Yangyan Li, and Hao Su. Frame mining: a free lunch for learning robotic manipulation from 3d point clouds. *arXiv preprint arXiv:2210.07442*, 2022. 2

[33] Qihao Liu, Weichao Qiu, Weiyao Wang, Gregory D Hager, and Alan L Yuille. Nothing but geometric constraints: A model-free method for articulated object pose estimation. *arXiv preprint arXiv:2012.00088*, 2020. 2

[34] Ziyuan Liu, Wei Liu, Yuzhe Qin, Fanbo Xiang, Minghao Gou, Songyan Xin, Maximo A Roa, Berk Calli, Hao Su, Yu Sun, et al. Ocrtoc: A cloud-based competition and benchmark for robotic grasping and manipulation. *IEEE Robotics and Automation Letters*, 7(1):486–493, 2021. 2

[35] Mayank Mittal, David Hoeller, Farbod Farshidian, Marco Hutter, and Animesh Garg. Articulated object interaction in unknown scenes with whole-body mobile manipulation. *arXiv preprint arXiv:2103.10534*, 2021. 3

[36] Kaichun Mo, Leonidas J Guibas, Mustafa Mukadam, Abhinav Gupta, and Shubham Tulsiani. Where2act: From pixels to actions for articulated 3d objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6813–6823, 2021. 3

[37] Kaichun Mo, Yuzhe Qin, Fanbo Xiang, Hao Su, and Leonidas Guibas. O2o-afford: Annotation-free large-scale object-object affordance learning. In *Conference on Robot Learning*, pages 1666–1677. PMLR, 2022. 3

[38] Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 909–918, 2019. 2, 3

[39] Igor Mordatch, Zoran Popović, and Emanuel Todorov. Contact-invariant optimization for hand manipulation. In *Proceedings of the ACM SIGGRAPH/Eurographics symposium on computer animation*, pages 137–144, 2012. 2

[40] Jiteng Mu, Weichao Qiu, Adam Kortylewski, Alan Yuille, Nuno Vasconcelos, and Xiaolong Wang. A-sdf: Learning disentangled signed distance functions for articulated shape representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13001–13011, 2021. 2

[41] Tongzhou Mu, Zhan Ling, Fanbo Xiang, Derek Yang, Xuanlin Li, Stone Tao, Zhiao Huang, Zhiwei Jia, and Hao Su. Maniskill: Generalizable manipulation skill benchmark with large-scale demonstrations. *arXiv preprint arXiv:2107.14483*, 2021. 2, 3

[42] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation, 2022. 8

[43] Domenico Prattichizzo and Jeffrey C Trinkle. Grasping. In *Springer handbook of robotics*, pages 955–988. Springer, 2016. 2

[44] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 2, 3, 4

[45] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 3

[46] Yuzhe Qin, Rui Chen, Hao Zhu, Meng Song, Jing Xu, and Hao Su. S4g: Amodal single-view single-shot se(3) grasp detection in cluttered scenes. In *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pages 53–65. PMLR, 2020. 3

[47] Yuzhe Qin, Binghao Huang, Zhao-Heng Yin, Hao Su, and Xiaolong Wang. Dexpoint: Generalizable point cloud reinforcement learning for sim-to-real dexterous manipulation. 2022. 4, 5

[48] Yuzhe Qin, Hao Su, and Xiaolong Wang. From one hand to multiple hands: Imitation learning for dexterous manipulation from single-camera teleoperation. *IEEE Robotics and Automation Letters*, 7(4):10873–10881, 2022. 2

[49] Yuzhe Qin, Yueh-Hua Wu, Shaowei Liu, Hanwen Jiang, Ruihan Yang, Yang Fu, and Xiaolong Wang. Dexmv: Imitation learning for dexterous manipulation from human videos. *arXiv preprint arXiv:2108.05877*, 2021. 2

[50] Ilija Radosavovic, Xiaolong Wang, Lerrel Pinto, and Jitendra Malik. State-only imitation learning for dexterous manipulation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7865–7871. IEEE, 2020. 2

[51] Daniela Rus. In-hand dexterous manipulation of piecewise-smooth 3-d objects. *The International Journal of Robotics Research*, 1999. 2

[52] Andreas J Schmid, Nicolas Gorges, Dirk Goger, and Heinz Worn. Opening a door with a humanoid robot using multi-sensory tactile feedback. In *2008 IEEE International Conference on Robotics and Automation*, pages 285–291. IEEE, 2008. 2

[53] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 4

[54] Hanchen Wang, Qi Liu, Xiangyu Yue, Joan Lasenby, and Matt J Kusner. Unsupervised point cloud pre-training via occlusion completion. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9782–9792, 2021. 6

[55] Lirui Wang, Yu Xiang, Wei Yang, Arsalan Mousavian, and Dieter Fox. Goal-auxiliary actor-critic for 6d robotic grasping with point clouds. In *Conference on Robot Learning*, pages 70–80. PMLR, 2022. 3

[56] Xiaogang Wang, Bin Zhou, Yahao Shi, Xiaowu Chen, Qinping Zhao, and Kai Xu. Shape2motion: Joint analysis of motion parts and attributes from 3d shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8876–8884, 2019. 2

[57] Yijia Weng, He Wang, Qiang Zhou, Yuzhe Qin, Yueqi Duan, Qingnan Fan, Baoquan Chen, Hao Su, and Leonidas J Guibas. Captra: Category-level pose tracking for rigid and articulated objects from point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13209–13218, 2021. 2

[58] Albert Wu, Michelle Guo, and C Karen Liu. Learning diverse and physically feasible dexterous grasps with generative model and bilevel optimization. *arXiv preprint arXiv:2207.00195*, 2022. 2

[59] Ruihai Wu, Yan Zhao, Kaichun Mo, Zizheng Guo, Yian Wang, Tianhao Wu, Qingnan Fan, Xuelin Chen, Leonidas Guibas, and Hao Dong. Vat-mart: Learning visual action trajectory proposals for manipulating 3d articulated objects. *arXiv preprint arXiv:2106.14440*, 2021. 3

[60] Yueh-Hua Wu, Jiashun Wang, and Xiaolong Wang. Learning generalizable dexterous manipulation from human grasp affordance. *arXiv preprint arXiv:2204.02320*, 2022. 3

[61] Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, Li Yi, Angel X. Chang, Leonidas J. Guibas, and Hao Su. SAPIEN: A simulated part-based interactive environment. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 3, 4, 5

[62] Zhenjia Xu, Zhanpeng He, and Shuran Song. Universal manipulation policy network for articulated objects. *IEEE Robotics and Automation Letters*, 7(2):2447–2454, 2022. 3

[63] Jianglong Ye, Jiashun Wang, Binghao Huang, Yuzhe Qin, and Xiaolong Wang. Learning continuous grasping function with a dexterous hand from human demonstrations. *arXiv preprint arXiv:2207.05053*, 2022. 2

[64] Li Yi, Haibin Huang, Difan Liu, Evangelos Kalogerakis, Hao Su, and Leonidas Guibas. Deep part induction from articulated object pairs. *arXiv preprint arXiv:1809.07417*, 2018. 2, 3

[65] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Metaworld: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pages 1094–1100. PMLR, 2020. 2

[66] Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. Pcn: Point completion network. In *2018 International Conference on 3D Vision (3DV)*, pages 728–737, 2018. 6

[67] Vicky Zeng, Tabitha Edith Lee, Jacky Liang, and Oliver Kroemer. Visual identification of articulated object parts. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2443–2450. IEEE, 2021. 2

[68] Tianhao Zhang, Zoe McCarthy, Owen Jow, Dennis Lee, Xi Chen, Ken Goldberg, and Pieter Abbeel. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5628–5635, 2018. 2

[69] Henry Zhu, Abhishek Gupta, Aravind Rajeswaran, Sergey Levine, and Vikash Kumar. Dexterous manipulation with deep reinforcement learning: Efficient, general, and low-cost. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3651–3657. IEEE, 2019. 1, 2