Supplementary Material for 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

1. Environment Details

1.1. Task Stage

As described in the Section 3.3 of the main paper, our reward is dependent on the stage of the environment. Stage is a hidden variable in the environment, which describes the overall progress of the task.

Faucet, Laptop and Toilet. For the Faucet, Laptop, and Toilet environments, we decompose the task progress into three stages. In the first stage, *i.e.* reaching, the palm of the robot should move close to the target functional part (handle of the faucet, lid of the laptop or toilet). The agent enters the second stage, *i.e.* grasping, if and only if the distance is close enough. The second stage encourages the contact of the palm and the target functional part, so we design a contact reward to ensure stable and natural grasp. We enter the final stage, *i.e.* manipulating, when the robot hand fully grasps the functional part. In the final stage, the agent receives the reward of task-specific progress, which is defined as the relative rotation of the joint in these three environments.

Bucket. The bucket environment can also be decomposed into three stages. In the first stage, the Euclidean distance between the palm and the bucket handle is computed to encourage the palm to reach the handle. When the distance is below a given threshold, the agent enters the second stage. The goal of the second stage is to grasp the handle and lift the handle to the vertical state in joint space. To produce a stable grasp, we penalize the linear and angular velocity of the base of each bucket as the object should not be moved during the stage. When the vertical degree of the joint state on the handle is over the threshold and the handle is fully grasped, the environment switches to the final stage, where the robot is required to lift the bucket. In the final stage, the height of the bucket relative to its rest plane as well as the grasp between the palm and the handle is rewarded to encourage the lifting behavior. Since we find the palm can produce a more stable support force, we reward the signed angle between the normal vector of the palm and the horizon plane to encourage the palm to face up.

1.2. Reward

As shown in the Section 3.3 of the main paper, our overall reward contains four terms:

$$\begin{aligned}\mathcal{R} =& w_{\text{reach}}\, r_{\text{reach}} + w_{\text{contact}}\, r_{\text{contact}} + \\ & w_{\text{progress}} r_{\text{progress}} + w_{\text{penalty}}\, r_{\text{penalty}}\,,\end{aligned} \tag{1}$$

$$r_{\text{reach}} = \mathbf{1}(\text{stage} == 1)\, \min(-\|\mathbf{x}_{\text{palm}} - \mathbf{x}_{\text{object}}\|, \lambda), \tag{2}$$

$$\begin{aligned}r_{\text{contact}} =& \mathbf{1}(\text{stage} \geq 2)\ \text{IsContact}\ (\ \text{palm, object}\ ) \\ & \textbf{AND}\left(\sum_{\text{finger}} \text{IsContact}\ (\ \text{finger, object}\ ) \geq 2\right),\end{aligned} \tag{3}$$

$$r_{\text{progress}} = \mathbf{1}(\text{stage} == 3)\text{Progress(task)}, \tag{4}$$

where $\lambda = 0.2$ in our inplementation. We define the progress function for each task as follows:

$$\text{Progress(Faucet)} = \frac{\pi}{2}\text{Normalize}(\theta), \tag{5}$$

$$\begin{aligned}&\text{Progress(Bucket)} = 0.5\text{Normalize}(\theta) + \\ &3(h_{\text{o\_current}} - h_{\text{o\_init}}) + 100(h_{\text{p\_current}} - h_{\text{p\_init}}),\end{aligned} \tag{6}$$

$$\text{Progress(Laptop)} = \text{Normalize}(\theta), \tag{7}$$

$$\text{Progress(Toilet)} = \text{Normalize}(\theta), \tag{8}$$

where θ is the relative rotation of the target joint in these environments. Normalize is a function in which values are shifted and re-scaled so that they end up ranging between 0 and 1. $h_{\text{o\_init}}$ and $h_{\text{o\_current}}$ are the initial and current height of the object and $h_{\text{p\_init}}$ and $h_{\text{p\_current}}$ are the initial and current height of the palm. For Faucet, Laptop and Toilet environments, the action penalty reward $r_{\text{penalty}} = -\|a\|_2^2$. For Bucket environment, the penalty reward is:

$$\begin{aligned}r_{\text{penalty}} =& w_{p1}\|a\|_2^2 + w_{p2} D + \\ & w_{p3}\mathbf{1}(\text{stage} == 2)(\|v\|_2^2 + \|\omega\|_2^2),\end{aligned} \tag{9}$$

Parameter	Value
Mini-Batch Size	500
Learning Rate	1e−4
Horizon	250
Clip Range	0.2

Table 1. **PPO Parameters.**

Module	Architecture	Output Dim
Small PointNet	PointNet Local Channel:(64, 256)	256
Medium PointNet	PointNet Local Channel:(64, 64, 128, 256)	256
Large PointNet	PointNet Local Channel:(64, 64, 128, 128, 256, 256)	256
State Feature Extractor	MLP: (64, 64)	64
Actor	MLP:(64, 64)	64
Critic	MLP:(64, 64)	64

Table 2. **Policy Learning Architecture.**

where $w_{p1} = -1$, $w_{p2} = 20$, $w_{p3} = -1$, D is the signed angle between the normal vector of the palm and the plane horizon, v and ω are the linear and angular velocity of the object.

The weights for the four terms in the overall reward are: $w_{\text{reach}} = 0.1$, $w_{\text{contact}} = 0.2$, $w_{\text{progress}} = 1$, $w_{\text{penalty}} = 0.01$.

2. Learning Details

RL Training. As mentioned in the Section 4.1 of the main paper, we use PPO as our on-policy RL algorithm to train our manipulation policy based on the point cloud. We give the hyper-parameters in Table 1.

Policy Learning Architecture As shown in Figure 3 of the main paper, We use PointNet as the point cloud feature extractor. We concatenate the point feature and the proprioception feature from the MLP, which is then shared by both value network and policy network to predict value and action. We show the details for the network architecture in Table 2. Note that in the Section 5.2 of the main paper, we use three PointNet with different sizes, i.e. small PointNet, medium PointNet and large PointNet. All other experiments are based on the small PointNet as the vision extractor.

3. Figures and Videos

We provide extra visualizations for our tasks and objects. In Figure 1 and 2, we visualize all the training (seen) and testing (unseen) objects. We also provide a video where we render how the PointNet is understanding the part segmentation during the RL training to show that part reasoning exists during the RL training, and we also test our policy under observations from novel camera viewpoints to show that our policy is robust to viewpoint changes.

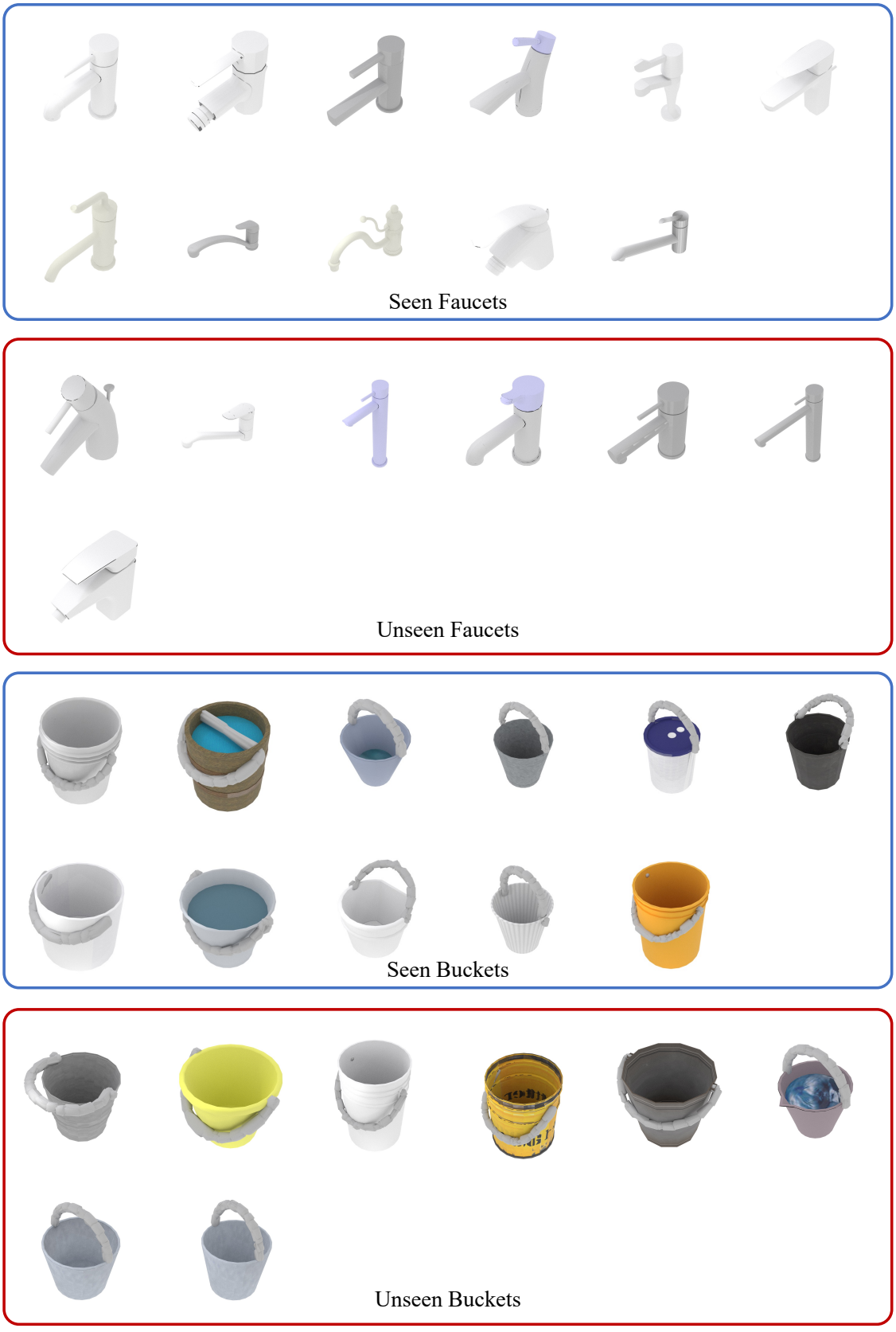

Figure 1. **Seen and Unseen Objects (1).**

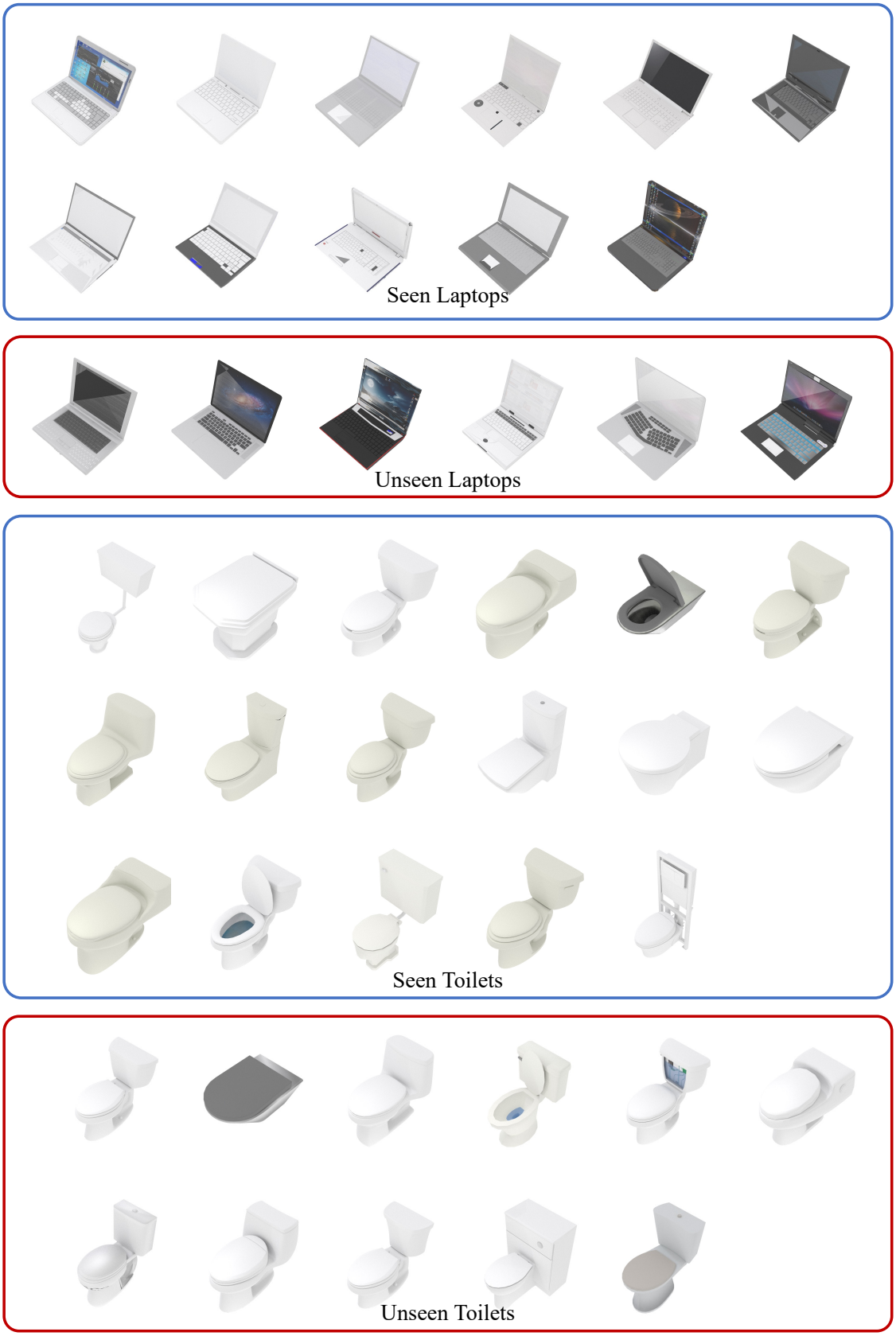

Figure 2. **Seen and Unseen Objects (2).**