LEARNING GENERALIZABLE DEXTEROUS MANIPULATION FROM HUMAN GRASP AFFORDANCE

Anonymous authors
Paper under double-blind review

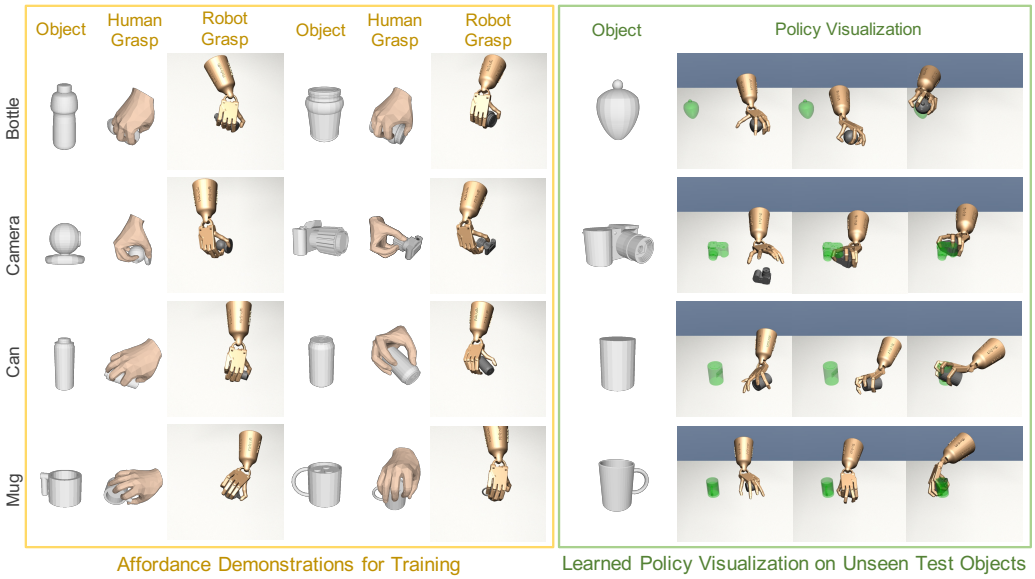

Figure 1: Examples of our affordance demonstrations and learned policies. **Left**: We visualize two groups of demonstrations for each object category. In each group, we visualize the given instance, generated human grasp, and the robot grasp for demonstrations. **Right**: We train a policy using imitation learning with the demonstrations, and visualize the learned policy relocating the unseen object.

ABSTRACT

Dexterous manipulation with a multi-finger hand is one of the most challenging problems in robotics. While recent progress in imitation learning has largely improved the sample efficiency compared to Reinforcement Learning, the learned policy can hardly generalize to manipulate novel objects, given limited expert demonstrations. In this paper, we propose to learn dexterous manipulation using large-scale demonstrations with diverse 3D objects in a category, which are generated from a human grasp affordance model. This generalizes the policy to novel object instances within the same category. To train the policy, we propose a novel imitation learning objective jointly with a geometric representation learning objective using our demonstrations. By experimenting with relocating diverse objects in simulation, we show that our approach outperforms baselines with a large margin when manipulating novel objects. We also ablate the importance on 3D object representation learning for manipulation.

1 INTRODUCTION

Human hands provide the primary means for our daily life interactions with the physical world. Our hands exhibit tremendous flexibility in operating objects around us. To enable the robot the same flexibility in assisting humans in daily life, dexterous manipulation with multi-finger robot hands

has been one of the core problems in robotics. At the same time, it is one of the most challenging problems in robotics given its high Degree-of-Freedom joints (e.g., 24 to 30 DoF). While recent progress in Reinforcement Learning (RL) has shown encouraging results on complex dexterous manipulation OpenAI et al. (2018; 2019); Zhu et al. (2019), it is still limited by the requirement of a large number of samples in training, and the trained policy can hardly generalize to novel objects during deployment.

To improve the sample efficiency in training, one promising direction is to perform imitation learning from human demonstrations Gupta et al. (2016); Rajeswaran et al. (2018); Schmeckpeper et al. (2020); Radosavovic et al. (2021). The expert demonstrations for dexterous manipulation can be collected by a human from teleoperation in a Virtual Reality (VR) system Rajeswaran et al. (2018) and using Mocap Handa et al. (2020). Guided by human demonstrations, it not only reduces sample complexity in learning but also helps robot hands perform human-like and safe behaviors. However, the current setup on data collection largely limits the number and diversity of the demonstrations. For example, data collection with VR in Rajeswaran et al. (2018) only leads to 25 demonstrations per task with one single object instance. With limited data, the learned policy can hardly generalize and transfer to unseen objects in test time.

To achieve generalization, we seek helps from recent studies on hand-object interactions and affordance reasoning Brahmbhatt et al. (2019); Taheri et al. (2020); Karunratanakul et al. (2020); Jiang et al. (2021). Instead of collecting the whole demonstration trajectories from a human at small scale, we can learn from the key interactions on how humans grasp and contact diverse 3D objects at much larger scale from existing affordance model Jiang et al. (2021).

In this paper, we propose to leverage the human grasp affordance model for generalizing dexterous manipulation to novel object instances in the same category. Specifically, we will first generate large-scale demonstrations on human hands interacting with diverse objects within the same category from affordance reasoning (left columns in Fig. 1). We then use imitation learning to train a policy by augmenting RL with these demonstrations and test on unseen objects (right columns in Fig. 1). Our policy takes the object point cloud and the robot hand state as inputs for decision making. We tackle generalization by jointly learning: (i) skill generalization with a new imitation learning objective; and (ii) geometric representation generalization with a behavior cloning objective. We illustrate different components of our approach as follows.

Demonstration Generation. Given each 3D object instance, we can first generate a hand grasp pose and a way to contact by leveraging the human grasp affordance model Jiang et al. (2021). Note there can be multiple possibilities of grasps that we can sample from the affordance model. We utilize motion planning to generate a trajectory that moves the robot hand from a start state to the target grasp. This trajectory provides a demonstration of how the robot hand can reach and stably grasp the object like humans do, preparing for the downstream tasks. Instead of exhaustively collecting demonstrations for a full task, we generate large-scale *partial demonstrations* across multiple diverse object instances in the same category.

Imitation Learning Objective. To learn the policy, we augment RL with our demonstrations for imitation learning. Previous approaches weighted all demonstrations equally during learning Rajeswaran et al. (2018); Radosavovic et al. (2021). Under diverse and large-scale demonstrations, we propose a novel ranking function to encourage the policy to learn from trajectories that it is less likely to reproduce. In addition, we estimate advantage values for state-action pairs from demonstrations with a growing weight so that the policy can still benefit from the given demonstrations at late training phases.

Geometric Representation Learning. The policy needs to understand the object shape given the point cloud inputs to manipulation it accordingly. We utilize PointNet Qi et al. (2017) to encode the input object and pre-train the representation with a behavior cloning task using our large-scale demonstrations. Besides pre-training, as the policy interacts with the environment during imitation learning, we can collect new data to continue fine-tuning the PointNet with behavior cloning. Our training pipeline jointly optimizes with the imitation learning objective for skill generalization and the behavior cloning objective for representation generalization.

We perform experiments in simulation with five different object categories. We train one policy for each object category on the *relocate* task, which requires the multi-finger robot hand to relocate an object instance from an initial position to a goal location. During the evaluation, we focus on

the metric on generalization to relocate novel objects that are not seen in training time. We not only observe significant improvement over RL and state-of-the-art imitation learning approaches but also ablate the effectiveness of our novel imitation learning objective and geometric representation learning using our demonstrations.

We highlight our main contributions as follows:

- A novel approach on generating large-scale dexterous manipulation demonstrations on diverse objects.
- A novel imitation learning objective and 3D geometric representation learning approach for generalizing dexterous manipulation.
- State-of-the-art performance on dexterous manipulation on novel objects, which has been rarely explored to our knowledge.

2 Related Work

Dexterous Manipulation. Dexterous manipulation with a multi-finger hand has been one of the core robotics problems Rus (1999); Okamura et al. (2000); Dogar & Srinivasa (2010); Andrews & Kry (2013); Bai & Liu (2014). While recent success has been shown in using Reinforcement Learning for solving complex dexterous manipulation tasks OpenAI et al. (2018; 2019), training with RL still suffers from high sample complexity and it might also need unexpected and unsafe behaviors given the high-dimensional action and state space. Recent efforts have been made on using affordance and contact reasoning to design an auxiliary loss to guide human-like dexterous grasping Mandikal & Grauman (2021). However, the small scale of contact examples limits the generalization ability of the policy. In this paper, we propose a novel method to generate large-scale demonstrations from grasp affordance, and a novel imitation learning algorithm to learn a generalizable policy.

Imitation learning. Imitation learning aims at recovering the expert policy that generates the given demonstrations. Beyond behavior cloning Torabi et al. (2018b); Reddy et al. (2019); Kelly et al. (2019), the definition of imitation learning also includes approaches that incorporate RL objectives, such as Inverse Reinforcement Learning (IRL) Ng et al. (2000); Abbeel & Ng (2004); Ho & Ermon (2016); Fu et al. (2017); Stadie et al. (2017); Aytar et al. (2018); Torabi et al. (2018a); Liu et al. (2020). For example, Ho et al. Ho & Ermon (2016) introduce to learn expert policy by matching occupancy measure Syed et al. (2008) between the agent policy and the expert policy using adversarial learning. Another line of imitation learning is to augment the expert demonstrations to the online collected data for Reinforcement Learning Peters & Schaal (2008); Duan et al. (2016); Večerík et al. (2017); Rajeswaran et al. (2018); Radosavovic et al. (2021). For example, Rajeswaran et al. Rajeswaran et al. (2018) propose to take maximum likelihood with demonstrations as an auxiliary term during RL training. However, they utilize perfect demonstrations collected via VR at a small scale. On the other hand, we propose to utilize imperfect partial demonstrations at a large scale for better generalization. In this spirit, our work is inspired by imitation learning from imperfect demonstrations Hester et al. (2018); Oh et al. (2018); Tangkaratt et al. (2019); Wu et al. (2019); Cao & Sadigh (2021), where demonstrations are not directly from an optimal policy.

3 Method

We propose to learn a policy using imitation learning with demonstrations generated from the human grasp affordance model. Our policy takes the object point clouds together with the hand joint states as inputs for decision making. We introduce our approach **I**mitation **L**earning from **A**ffordance **D**emonstration (ILAD) as a 2-stage pipeline:

(i) **Affordance Demonstration Generation**. We leverage a state-of-the-art affordance model GraspCVAE Jiang et al. (2021) to generate diverse grasps on diverse objects within the same category. With the generated grasps, we utilize motion planning to obtain trajectories to reach these grasps. While these trajectories do not show how to perform a particular task, they can serve as partial demonstrations for guiding our policy to achieve the right contacts in grasping.

(ii) **Imitation Learning with Representation Learning**. We propose a novel imitation learning objective to learn the policy with the affordance demonstrations. As we utilize a PointNet Qi et al.

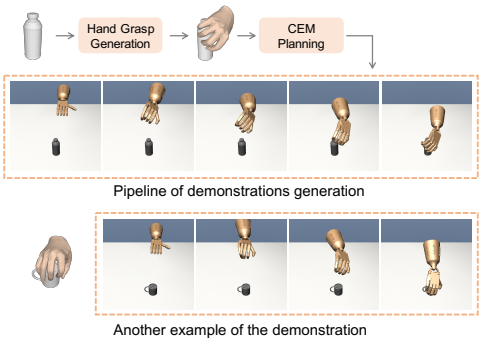

Figure 2: Affordance demonstration generation. We first generate the hand grasp on a given object and then use CEM for planning to reach the target grasp position. We also provide another example of the demonstration.

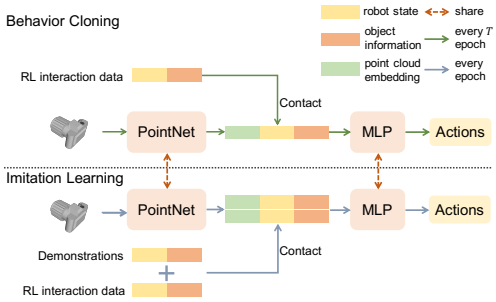

Figure 3: Training pipeline for the proposed behavior cloning and imitation learning. Every T epoch, we finetune the PointNet θ_{pc} using the objective (Eq. 5) with respect to θ_{pc}. During imitation learning, we update the MLP θ_p by estimating the gradient (Eq. 6) with the given demonstrations and the RL interaction data.

(2017) encoder to extract the 3D object shape information, we propose a 3D geometric representation learning approach jointly with imitation learning.

3.1 Affordance Demonstration Generation

We propose to generate demonstrations from human grasp affordance. This procedure includes two steps as shown in Fig. 2: grasp generation and motion planning for grasp trajectory.

Grasp generation. Given diverse 3D objects, we adopt the GraspCVAE model proposed in Jiang et al. (2021) to generate diverse human grasps for each object. Specifically, the GraspCVAE will take the object point cloud observation as inputs, and outputs the grasp pose represented by the MANO Romero et al. (2017) model parameterized by the shape parameter β as well as the pose parameter α. With these parameters, we can compute the human hand joints using the forward kinematics function $j^h = \mathcal{J}_h(\beta, \alpha)$. We will use these joint positions to guide the robot to reach the objects plausibly.

Motion planning for grasp trajectory. Given the target grasp hand joints j^h, our goal is to find an robot hand action sequence $a_1, ..., a_K$ which generates a robot hand joint sequence $j_0^r, ..., j_K^r$ so that the last robot hand joint positions j_K^r reaches j^h. Note the initial robot hand joints j_0^r are given. The objective for motion planning is,

$$\min_{a_1,...,a_K} \|j_K^r - j^h\|^2 + \lambda \|p_K - p_1\|^2,$$

where p_1 and p_K are object poses at time step 1 and K, and constant $\lambda = 10$. The first term of the objective encourages the robot hand to reach the human grasp, and the second term indicates the object should not be moving during the process. We use cross-entropy method (CEM) Rubinstein (1999) for motion planning given this objective.

3.2 Imitation Learning Objective

We perform imitation learning using demonstrations generated from our planning algorithm. Instead of pure behavior cloning, our imitation learning considers a setting where a reward function for Reinforcement Learning and demonstrations are given at the same time. In this way, we can perform training even with imperfect demonstrations since the RL objective needs to be achieved. Meanwhile, a large-sale and diverse demonstrations can provide effective guidance for exploration during RL training.

Preliminaries. We consider a standard Markov Decision Process (MDP). It is represented by a tuple $\langle S, A, P, R, \gamma \rangle$, where S and A are state and action space, $P(s_{t+1}|s_t, a_t)$ is the transition density of state s_{t+1} at step $t+1$ given action a_t made under state s_t, $R(s, a)$ is the reward function, and γ is the discount factor. The goal of RL is to maximize the expected reward with a stochastic policy $\pi(a|s)$.

We build our approach upon an imitation learning baseline algorithm called Demo Augmented Policy Gradient (DAPG) Rajeswaran et al. (2017). It combines learning from demonstration and policy optimization. The learning objective function at epoch k can be represented as,

$$g_{aug} = \sum_{(s,a)\in D_{\pi_\theta}} \nabla_\theta \ln \pi_\theta(a|s) A^{\pi_\theta}(s,a) + \sum_{(s,a)\in D_{\pi_\mathrm{E}}} \nabla_\theta \ln \pi_\theta(a|s) \lambda_0 \lambda_1^k \max_{(s,a)\in D_{\pi_\theta}} A^{\pi_\theta}(s,a),$$

where A^{π_θ} is the advantage function Baird III (1993) that is used to estimate the difference of the discounted reward sum starting from (s,a) and s according to policy π_θ, D_{π_E} are state-action pairs from the expert demonstrations, $D_{\pi_{\pi_\theta}}$ are state-action pairs collected with policy π_θ, and λ_0 and λ_1 hyper-parameters bounded by 0 and 1. In the implementation of Rajeswaran et al. (2017), $\max_{(s,a)\in\rho_{\pi_\theta}} A^{\pi_\theta}(s,a)$ is set to 1 for stability. Therefore, the objective is reduced to

$$g_{aug} = \sum_{(s,a)\in D_{\pi_\theta}} \nabla_\theta \ln \pi_\theta(a|s) A^{\pi_\theta}(s,a) + \sum_{(s,a)\in D_{\pi_\mathrm{E}}} \nabla_\theta \ln \pi_\theta(a|s) \lambda_0 \lambda_1^k, \tag{1}$$

which suggests that the given demonstrations are considered equally during training and the importance of the demonstration term is decreasing along with training time in order to reduce objective bias.

Learning from partial demonstrations with multiple objects. For generalizing dexterous manipulation to multiple objects, where there are easier and more challenging shapes, the demonstrations should not be taken equally during training. We propose to adaptively rank the demonstrations based on the difficulties of objects and the learning progress of the policy. Specifically, we will first define the objective below, and then explain the terms in this objective,

$$\begin{aligned} g_{ILAD} = &\sum_{(s,a)\in D_{\pi_\theta}} \nabla_\theta \ln \pi_\theta(a|s) A^{\pi_\theta}(s,a) + \sum_{(s,a)\in D_{\pi_\mathrm{E}}} \nabla_\theta \ln \pi_\theta(a|s) \lambda_0 \lambda_1^k w_k(s,a) + \\ &\sum_{(s,a)\in D_{\pi_\mathrm{E}}} \nabla_\theta \ln \pi_\theta(a|s) \lambda_0' (1-\lambda_1^k) A_\phi^{\pi_\theta}(s,a), \end{aligned} \tag{2}$$

where $w_k(s,a)$ in the second term is computed as the negative of a normalized log likelihood to encourage the policy to learn from trajectories that it is hard to reproduce by the current policy. $A_\phi^{\pi_\theta}(s,a)$ in the third term is an advantage function estimated with a model parameterized by ϕ for state-action pairs of the demonstrations. Formally, $w_k(s,a)$ can be represented as a normalized value (scaled between 0 and 1) of the negative of the log likelihood,

$$w_k(s,a) = \frac{l_k(\tau_{s,a}) - \min_\tau l_k(\tau)}{\max_\tau l_k(\tau) - \min_\tau l_k(\tau)}, \tag{3}$$

where $\tau_{s,a}$ is a trajectory from the demonstrations that contains station-action pair (s,a), and the negative of the log likelihood for a trajectory $l_k(\tau)$ is defined as,

$$l_k(\tau) = -\frac{1}{|\tau|} \sum_{(s,a)\in\tau} \log \Pr(s,a|\pi_\theta). \tag{4}$$

With these definitions, we will explain our key innovations on the second term and the third term in Eq. 2.

Normalized likelihood weights in the second term in Eq. 2. In our experiments, we find that the policy could easily learn to manipulate a certain kind of object while neglecting the others. To encourage the policy to generalize on diverse objects, we will dynamically assign larger weights $w_k(s,a)$ for demonstrations that have a smaller likelihood in the current epoch, which encourages the policy to focus training on them more. Specifically, $w_k(s,a)$ is a weight at epoch k for state-action pair (s,a) from an expert demonstration $\tau_{s,a}$. To compute this weight, we first compute the negative of log likelihood $l_k(\tau)$ for a trajectory τ as Eq. 4. $l_k(\tau)$ will be large if the current policy does not fit the trajectory τ well, which means the training should pay more attention to this trajectory. The weight $w_k(s,a)$ is a normalized version of the negative of log likelihood using Eq. 3, so that it will be scaled between 0 and 1.

Advantage approximation for demonstrations in the third term in Eq. 2 is designed to further elevate the utilization of demonstrations. While in the previous approach Rajeswaran et al. (2017)

the advantage function for demonstrations is taken as 1 in Eq. 1, we propose to approximate the true advantage for a more accurate estimation of the gradients. Since we do not have the rewards provided in demonstrations to compute $A^{\pi_\theta}(s,a)$ as with online data in the first term, we train a neural network $A^{\pi_\theta}_{\phi}(s,a)$ parameterized by ϕ to predict the advantage function directly. This new advantage function is trained with the online data collected by RL and applied to the partial demonstrations.

3.3 Policy Training with Geometric Representation

Our policy takes both the point cloud of the object, object 6D pose, robot hand joint states as inputs, and predicts the actions for the robot hand. Specifically, to represent the object shape, we utilize the PointNet Huang et al. (2021) encoder θ_{pc} for the point cloud inputs. Given the point cloud embedding, we concatenate it with the object 6D pose parameters and hand joint states together and forward them together to a 3-layer MLP θ_p network for decision making. Thus the policy network is parameterized by $\theta = \{\theta_{pc}, \theta_p\}$.

To perform training, besides optimizing towards the imitation learning objective, we design a geometric representation learning objective for training the PointNet jointly. Our overall model architecture and training pipeline are visualized in Fig. 3. We will first explain the representation learning objective, and then the joint training approach in the following.

Behavior cloning for geometric representation learning. We utilize behavior cloning to provide an objective to train our PointNet encoder. We obtain the training data directly from the examples D_{π_θ} collected during the interaction with the environment in policy learning. Specifically, the behavior cloning objective can be represented as,

$$\mathcal{L}_{bc} = \frac{1}{|D_{\pi_\theta}|} \sum_{(s,a)\in D_{\pi_\theta}} \|\pi_\theta(s) - a\|^2, \tag{5}$$

where we still utilize the whole network $\theta = \{\theta_{pc}, \theta_p\}$ including the decision making MLP θ_p to compute the loss, we only optimize the PointNet parameters θ_{pc} through backpropagation. This part of training corresponds to the upper part of Fig. 3.

Pre-training. The same objective Eq. 5 can also be used to pre-train the PointNet encoder and policy network before policy learning. To perform pre-training, we utilize our collected demonstrations D_{π_E} instead of D_{π_θ}.

Joint learning with both objectives. We train our policy jointly with both the imitation learning objective and the behavior cloning objective as shown in Fig. 3. Empirically, we find training the PointNet with policy gradient in RL (Eq. 2) makes the representation unstable for decision making. Unlike supervised learning, the variance of gradients is much larger in RL and it is very challenging to learn an encoder with high-dimensional inputs directly Srinivas et al. (2020); Stooke et al. (2021). On the other hand, behavior cloning provides a supervised objective Eq. 5 to stably train the PointNet representation. Thus we propose to share the network parameters $\theta = \{\theta_{pc}, \theta_p\}$ for both objectives, but use policy gradient to optimize the decision making MLP θ_p and behavior cloning to optimize the PointNet encoder θ_{pc}. We can re-write the objective Eq. 2 by replacing θ with θ_p in red as,

$$\begin{aligned} g_{ILAD} = &\sum_{(s,a)\in D_{\pi_\theta}} \nabla_{\theta_p} \ln \pi_\theta(a|s) A^{\pi_\theta}(s,a) + \sum_{(s,a)\in D_{\pi_\text{E}}} \nabla_{\theta_p} \ln \pi_\theta(a|s) \lambda_0 \lambda_1^k w_k(s,a) + \\ &\sum_{(s,a)\in D_{\pi_\text{E}}} \nabla_{\theta_p} \ln \pi_\theta(a|s) \lambda_0' (1-\lambda_1^k) A^{\pi_\theta}_{\phi}(s,a). \end{aligned} \tag{6}$$

To further stabilize the training, we propose to perform slower updates on the PointNet encoder so that the decision-making can be based on similar representations over time. Specifically, while we update the MLP θ_p for every epoch using policy gradients, we only perform an update on the PointNet encoder θ_{pc} every T epochs using behavior cloning. We will ablate the parameters T in our experiments. In Fig. 3, we visualize the arrows with different colors to represent different update strategies. We summarize our learning procedure in Algorithm 1.

Algorithm 1 ILAD Pre-Training and Joint Learning

Input: partial demonstrations D_{π_E}, PointNet θ_{pc}, policy network θ_p
Pre-train θ_{pc} and θ_p, according to Eq. 5
for $t = 0, 1, 2, ...$ **do**
    Sample trajectories $D_{\pi_\theta} = \{(s_i, a_i)\}_{i=1}^{n}$
    if $t \equiv 0 \pmod{T}$ **then**
        Update θ_{pc}, according to Eq. 5
    end if
    Update θ_p, according to Eq. 6
end for

Model	Bottle	Remote	Mug	Can	Camera	**Average**
RL	0.00 ± 0.00	0.62 ± 0.24	0.01 ± 0.01	0.00 ± 0.00	0.15 ± 0.20	0.16 ± 0.13
DAPG	0.58 ± 0.17	0.54 ± 0.20	0.70 ± 0.23	0.58 ± 0.24	0.64 ± 0.16	0.61 ± 0.20
DAPG (large)	0.32 ± 0.44	0.81 ± 0.02	0.97 ± 0.02	0.68 ± 0.25	0.56 ± 0.07	0.67 ± 0.23
ILAD (T=50)	0.95 ± 0.03	0.91 ± 0.04	0.94 ± 0.05	0.67 ± 0.45	$\mathbf{0.99 \pm 0.01}$	0.89 ± 0.20
ILAD (T =10, large)	0.81 ± 0.16	$\mathbf{0.94 \pm 0.06}$	0.97 ± 0.01	0.93 ± 0.05	0.98 ± 0.00	0.93 ± 0.07
ILAD (T =20, large)	0.85 ± 0.05	0.93 ± 0.03	$\mathbf{0.99 \pm 0.01}$	$\mathbf{0.96 \pm 0.02}$	0.93 ± 0.02	0.93 ± 0.02
ILAD (T =50, large)	$\mathbf{0.99 \pm 0.01}$	0.93 ± 0.04	0.96 ± 0.03	0.91 ± 0.05	$\mathbf{0.99 \pm 0.01}$	$\mathbf{0.96 \pm 0.03}$

Table 1: The success rate of the evaluated methods on unseen objects. For better clarity, we use "large" to represent demonstration size of 1000 trajectories during training and demonstration size of 100 trajectories for others. T is the updating interval.

4 Experiments

4.1 Experiment and Comparison Settings.

We conduct experiments on *Relocate* task with five categories: bottle, remote, mug, can, and camera. In the task, an object is placed on a table with random orientation and position and the robot is required to grasp the object and move it to a random target position. For each category, we use 40 objects for training and use around 30 unseen objects (differ by category) for testing to evaluate the generalizability. The unseen objects did not appear during training but are within the same category as training objects. Both training and testing objects are from ShapeNet Chang et al. (2015). There are two settings of the demonstration size. One is 100 trajectories for each category and one is 1000 for each category. We compare our method with DAPG and RL and ablate the updating interval T using a large number of demonstrations. We ablate other proposed components and the demonstration quality using a small number of demonstrations.

We adopt TRPO Schulman et al. (2015) as our Reinforcement Learning baseline. We evaluate the proposed imitation learning algorithm using DAPG as a baseline. They incorporate the demonstrations with the same RL algorithm (TRPO) using the same hyper-parameters. DAPG shares the same PointNet encoder as ours to encode the point cloud. We parameterized the value function with two separate 2-layer MLPs.

4.2 Main Comparisons

Success rate. We compare ILAD with DAPG and RL using both small and large number of demonstrations. The results are presented in terms of *success rate* on unseen objects in Tab. 1. The performance is evaluated via 100 trials for three seeds. A trial is counted as a success when the final position of the object (after executing the policy for 200 steps) is within 0.1 unit length to the specified target. Note that both the initial object position and target position are randomized. Tab. 1 shows that ILAD outperforms RL and DAPG with a large margin. Relocating unseen objects is difficult for RL baseline and it only achieves an average success rate of 16%. While ILAD achieves an average success rate of 96% which is about 43% improvement over DAPG. At the same time, a large number of demonstrations could achieve about 8% improvement on average for both DAPG and ILAD. This is also a contribution of our demonstration generation method since it can automatically generate large-scale demonstrations.

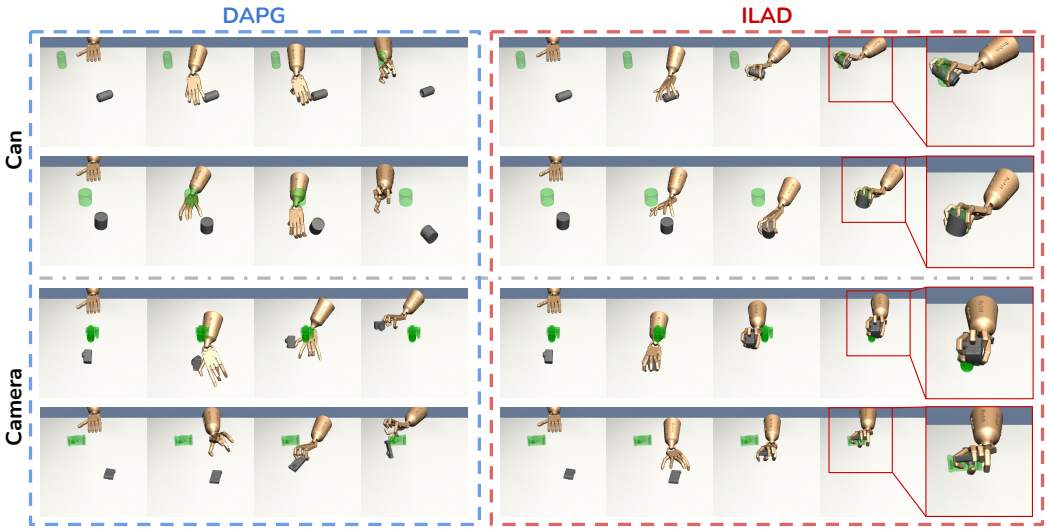

Figure 4: Comparison of the robustness on unseen can and camera objects. **Left**: policy learned by DAPG; **Right**: policy learned by ILAD. The environments in the same row share the same objects, initial position, and target position. We zoom in the last frame of our results.

Model	Bottle	Remote	Mug	Can	Camera	**Average**
DAPG (random embed.)	0.18 ± 0.22	0.09 ± 0.11	0.08 ± 0.12	0.01 ± 0.01	0.60 ± 0.23	0.19 ± 0.15
DAPG	0.58 ± 0.17	0.54 ± 0.20	0.70 ± 0.23	0.58 ± 0.24	0.64 ± 0.16	0.61 ± 0.20
ILAD (random embed.; w/o JT)	0.16 ± 0.17	0.06 ± 0.05	0.01 ± 0.00	0.31 ± 0.43	0.68 ± 0.19	0.24 ± 0.22
ILAD ($\lambda_0' = 0$; w/o JT)	0.61 ± 0.31	0.52 ± 0.05	0.72 ± 0.36	0.61 ± 0.32	0.69 ± 0.23	0.63 ± 0.28
ILAD (w/o joint learning)	0.65 ± 0.24	0.57 ± 0.26	0.76 ± 0.26	0.66 ± 0.33	0.70 ± 0.25	0.67 ± 0.27
ILAD	$\mathbf{0.95 \pm 0.03}$	$\mathbf{0.91 \pm 0.04}$	$\mathbf{0.94 \pm 0.05}$	$\mathbf{0.67 \pm 0.45}$	$\mathbf{0.99 \pm 0.01}$	$\mathbf{0.89 \pm 0.20}$

Table 2: The success rate of the ablative baselines on unseen objects. The performance is evaluated via 100 trials for three seeds. In this comparison, we set updating interval $T = 50$ for the joint training. For clarity, we use "JT" for joint learning.

Visualization on generalizability. We execute the policies on unseen objects and visualize the results in Fig. 4. For pair comparison, we fix the initial position of the object and the target position for both policies. In Fig. 4, we show that ILAD learns to hold the objects firmly even when they are not seen during training. Although DAPG achieves competitive performance in terms of average return during training, it is weak to generalize to unseen objects. It is especially challenging to grasp cylinder objects that require a specific angle and careful handling as suggested in the first row. In the fourth row, the policy is required to relocate a camera lying flat on the table. The proposed ILAD grasps the whole camera, which allows it to move the camera stably. On the other hand, DAPG only holds one side of the camera and the camera ends up being thrown away.

5 Conclusion

In this paper, we introduce ILAD, an imitation learning method that generalizes to manipulate novel objects. We design a demonstration generation pipeline with affordance reasoning. Further, we propose a novel imitation learning objective jointly with a geometric representation learning objective, which allows second-order policy gradient methods to train efficiently with high-dimensional geometric information. We try to bridge the gap between computer vision and robotics, which could benefit both communities. At the same time, we focus on the test of unseen objects, which is an under-explored field in previous studies. Our method could learn a more generalizable and robust policy, which is valuable for real robots in the real world.

REFERENCES

Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. 2004. 3

Sheldon Andrews and Paul G Kry. Goal directed multi-finger manipulation: Control policies and analysis. *Computers & Graphics*, 2013. 3

Yusuf Aytar, Tobias Pfaff, David Budden, Thomas Paine, Ziyu Wang, and Nando de Freitas. Playing hard exploration games by watching youtube. In *NeurIPS*, 2018. 3

Yunfei Bai and C Karen Liu. Dexterous manipulation using both palm and fingers. 2014. 3

Leemon C Baird III. Advantage updating. Technical report, 1993. 5

Samarth Brahmbhatt, Cusuh Ham, Charles C Kemp, and James Hays. Contactdb: Analyzing and predicting grasp contact via thermal imaging. In *CVPR*, 2019. 2

Zhangjie Cao and Dorsa Sadigh. Learning from imperfect demonstrations from agents with varying dynamics. *IEEE Robotics and Automation Letters*, 6(3):5231–5238, 2021. 3

Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 7

Mehmet R Dogar and Siddhartha S Srinivasa. Push-grasping with dexterous hands: Mechanics and a method. 2010. 3

Yan Duan, Xi Chen, Rein Houthooft, John Schulman, and Pieter Abbeel. Benchmarking deep reinforcement learning for continuous control. 2016. 3

Justin Fu, Katie Luo, and Sergey Levine. Learning robust rewards with adversarial inverse reinforcement learning. *arXiv*, 2017. 3

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

Ankur Handa, Karl Van Wyk, Wei Yang, Jacky Liang, Yu-Wei Chao, Qian Wan, Stan Birchfield, Nathan Ratliff, and Dieter Fox. Dexpilot: Vision-based teleoperation of dexterous robotic hand-arm system. In *ICRA*, 2020. 2

Todd Hester, Matej Vecerik, Olivier Pietquin, Marc Lanctot, Tom Schaul, Bilal Piot, Dan Horgan, John Quan, Andrew Sendonaris, Ian Osband, et al. Deep q-learning from demonstrations. In *Thirty-second AAAI conference on artificial intelligence*, 2018. 3

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *NeurIPS*, 2016. 3

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. 6

Hanwen Jiang, Shaowei Liu, Jiashun Wang, and Xiaolong Wang. Hand-object contact consistency reasoning for human grasps generation. *ICCV*, 2021. 2, 3, 4

Korrawe Karunratanakul, Jinlong Yang, Yan Zhang, Michael J Black, Krikamol Muandet, and Siyu Tang. Grasping field: Learning implicit representations for human grasps. In *2020 International Conference on 3D Vision (3DV)*, pp. 333–344. IEEE, 2020. 2

Michael Kelly, Chelsea Sidrane, Katherine Driggs-Campbell, and Mykel J Kochenderfer. Hg-dagger: Interactive imitation learning with human experts. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 8077–8083. IEEE, 2019. 3

Fangchen Liu, Zhan Ling, Tongzhou Mu, and Hao Su. State alignment-based imitation learning. In *ICLR*, 2020. 3

Priyanka Mandikal and Kristen Grauman. Learning dexterous grasping with object-centric visual affordances. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6169–6176. IEEE, 2021. 3

Andrew Y Ng, Stuart J Russell, et al. Algorithms for inverse reinforcement learning. 2000. 3

Junhyuk Oh, Yijie Guo, Satinder Singh, and Honglak Lee. Self-imitation learning. In *International Conference on Machine Learning*, pp. 3878–3887. PMLR, 2018. 3

Allison M Okamura, Niels Smaby, and Mark R Cutkosky. An overview of dexterous manipulation. In *ICRA*, 2000. 3

OpenAI, Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafał Józefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, Jonas Schneider, Szymon Sidor, Josh Tobin, Peter Welinder, Lilian Weng, and Wojciech Zaremba. Learning dexterous in-hand manipulation. *arXiv*, 2018. 2, 3

OpenAI, Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, Jonas Schneider, Nikolas Tezak, Jerry Tworek, Peter Welinder, Lilian Weng, Qiming Yuan, Wojciech Zaremba, and Lei Zhang. Solving rubik's cube with a robot hand. *arXiv*, 2019. 2, 3

Jan Peters and Stefan Schaal. Reinforcement learning of motor skills with policy gradients. *Neural networks*, 2008. 3

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*, pp. 652–660, 2017. 2, 3

Ilija Radosavovic, Xiaolong Wang, Lerrel Pinto, and Jitendra Malik. State-only imitation learning for dexterous manipulation. *IROS*, 2021. 2, 3

Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *arXiv*, 2017. 5

Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. 2018. 2, 3

Siddharth Reddy, Anca D Dragan, and Sergey Levine. Sqil: Imitation learning via reinforcement learning with sparse rewards. *arXiv preprint arXiv:1905.11108*, 2019. 3

Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *ToG*, 2017. 4

Reuven Rubinstein. The cross-entropy method for combinatorial and continuous optimization. *Methodology and computing in applied probability*, 1(2):127–190, 1999. 4

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

Karl Schmeckpeper, Oleh Rybkin, Kostas Daniilidis, Sergey Levine, and Chelsea Finn. Reinforcement learning with videos: Combining offline observations with interaction. *arXiv*, 2020. 2

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *ICML*, 2015. 7

Aravind Srinivas, Michael Laskin, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. *arXiv preprint arXiv:2004.04136*, 2020. 6

Bradly C Stadie, Pieter Abbeel, and Ilya Sutskever. Third-person imitation learning. *arXiv preprint arXiv:1703.01703*, 2017. 3

Adam Stooke, Kimin Lee, Pieter Abbeel, and Michael Laskin. Decoupling representation learning from reinforcement learning. In *International Conference on Machine Learning*, pp. 9870–9879. PMLR, 2021. 6

Umar Syed, Michael Bowling, and Robert E Schapire. Apprenticeship learning using linear programming. In *Proceedings of the 25th international conference on Machine learning*, pp. 1032–1039, 2008. 3

Omid Taheri, Nima Ghorbani, Michael J Black, and Dimitrios Tzionas. Grab: A dataset of whole-body human grasping of objects. In *ECCV*, 2020. 2

Voot Tangkaratt, Bo Han, Mohammad Emtiyaz Khan, and Masashi Sugiyama. Vild: Variational imitation learning with diverse-quality demonstrations. *arXiv preprint arXiv:1909.06769*, 2019. 3

Faraz Torabi, Garrett Warnell, and Peter Stone. Generative adversarial imitation from observation. *arXiv*, 2018a. 3

Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. *arXiv preprint arXiv:1805.01954*, 2018b. 3

Matej Večerík, Todd Hester, Jonathan Scholz, Fumin Wang, Olivier Pietquin, Bilal Piot, Nicolas Heess, Thomas Rothörl, Thomas Lampe, and Martin Riedmiller. Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards. *arXiv*, 2017. 3

Yueh-Hua Wu, Nontawat Charoenphakdee, Han Bao, Voot Tangkaratt, and Masashi Sugiyama. Imitation learning from imperfect demonstration. In *International Conference on Machine Learning*, pp. 6818–6827. PMLR, 2019. 3

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)*, pp. 3651–3657. IEEE, 2019. 2