FROM ONE HAND TO MULTIPLE HANDS: IMITATION LEARNING FOR DEXTEROUS MANIPULATION FROM SINGLE-CAMERA TELEOPERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce a novel single-camera teleoperation system for learning dexterous manipulation. Our system allows human operators to collect 3D demonstrations efficiently with only an iPad and a computer. These demonstrations are then used for imitation learning on complex multi-finger robot hand manipulation tasks. One key contribution of our system is that we construct a customized robot hand for each user in the physical simulator, which is a manipulator resembling the same kinematics structure and shape of the operator's hand. This not only avoids unstable human-robot hand retargetting during data collection, but also provides a more intuitive and personalized interface for different users to operate on. Once the data collection is done, the customized robot hand trajectories can be converted to different specified robot hands (models that are manufactured and commercialized) to generate training demonstrations. Using the data collected on the customized hand, our imitation learning results show large improvement over pure RL on multiple specified robot hands.

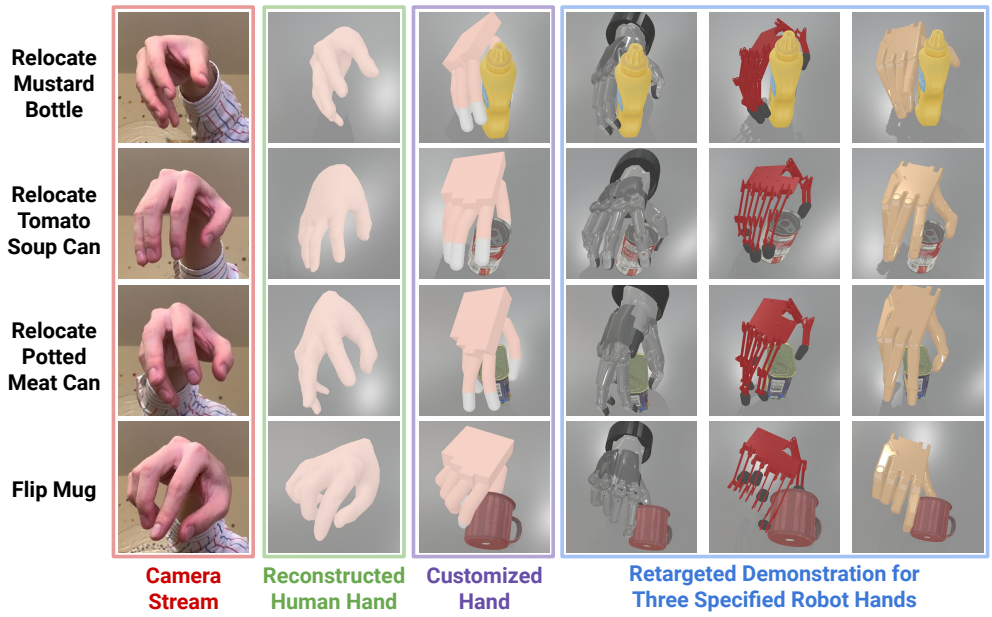


Figure 1: We introduce a teleoperation system which utilizes a single camera on an iPad to record a human hand, estimates the 3D hand pose and shape, converts it to a customized robot hand in a physical simulator for dexterous manipulation. Once the manipulation trajectories are collected, we convert them to different specified robot hands to generate demonstrations, and use them for imitation learning.

1 INTRODUCTION

Dexterous manipulation with multi-finger hand is of vital importance in robot manipulation study. The complex contact pattern between dexterous hand and manipulated objects is very hard to model.

It is very challenging to design a controller manually that can solve contact-rich tasks in unstructured environment. Recent research shows possibilities to learn dexterous manipulation skills with Reinforcement Learning (RL) (OpenAI et al., 2018; 2019; Rajeswaran et al., 2017). The closed-loop learning from interaction philosophy is appealing to robotics researchers. However, the high Degree-of-Freedom(DoF) joints and discontinuous contact increase the *sample complexity* to train an RL policy. It often takes a large amount of time to collect interaction data. Besides, black-box optimization with RL rewards can also lead to *unexpected and unsafe* behaviors.

Learning from human demonstration collected by the teleoperation system is a natural solution for dexterous manipulation given the similar morphology. Most current teleoperation systems require Virtual Reality (VR) (Rajeswaran et al., 2017; Kumar & Todorov, 2015; Hedayati et al., 2018; Pan et al., 2021; Zhang et al., 2018) devices or wired gloves to capture human hands. While providing accurate data collection, it also limits the flexibility and scalability of the process. On the other hand, vision-based teleoperation frees the human operator from wearing special devices and thus reduces the cost and is more scalable.

However, vision-based teleoperation introduces another challenge in the process. Almost all existing methods need to convert the collected human hand motion into robot hand motion to command the robot, which is called motion retargeting (Handa et al., 2020; Li et al., 2019; Antotsiou et al., 2018). A human operator needs to choose the movement based on the imagination of the future robot hand gesture and configuration. But humans are only familiar with their own hand. The human operators may find it hard to control the robot if the retargeting mapping is not intuitive, and it will take much time to calibrate their own hands with the robot hands. Moreover, the demonstrations collected with a specific robot hand can only be used for training imitation learning algorithm on the same robot.

In this paper, we introduce a single-camera teleoperation system with a scalable setup and an intuitive control interface that can collect demonstrations for multiple robot hands. In particular, our system only requires an iPad or one of another mobile device as the capturing device and *DOES NOT need to perform motion retargeting online during teleoperation*. At the beginning of data collection, our system will first perform initialization and automatic calibration: estimate the hand geometry of a specific human operator and construct the mapping from the operator frame to the frame in simulated environment. The key of our system is to generate a customized robot hand on the fly in the physical simulator. The customized robot hand will resemble the same kinematics structure of the operator's hand in both geometry (e.g., shape and size) and morphology. The system will generate different robot hands for different human operators. In this way, we can avoid motion retargeting and provide a more intuitive way for manipulation at the same time. The operator can then control the customized hand that he or she is most familiar with to perform dexterous manipulation tasks. Despite the lack of tactile feedback, our system can collect demonstration efficiently at around 60 trajectories per hour for the Relocate and Flip task in subsection 4.1.

After all the data is collected with the teleoperation system, we perform motion retargeting via optimization *offline*. We convert the trajectory of a *customized robot hand* to actual *specified robot hands* (i.e., the corresponding models are manufactured and commercialized in the real world). We experiment with 3 types of robot hands including the Schunk Robot Hand (Schunk), the Adroit Robot Hand (Kumar et al., 2013), and the AR10 Robot Hand (Robots). We only need to collect the trajectories once to generate imitation data for all these specified robot hands. We can then use the demonstrations for imitation learning on the corresponding manipulation task. We apply the imitation learning algorithm by augmenting the RL objective with the collected demonstrations (Rajeswaran et al., 2017).

We experiment with two types of challenging dexterous manipulation tasks: Relocate and Flip. In the Relocate task, the robot needs to pick up and place an object to match a goal position, which is randomized in each trial. The Flip task requires the robot to revert a fallen mug to its normal pose. Our results show that human demonstrations collected in our teleoperation system significantly improve dexterous hand manipulation on three different specified robot hands, compared to pure RL method. During data collection, multiple users with different hand sizes can use our system which generates different customized hands to perform teleoperation. We hope our teleoperation system can reduce the burden of demonstration collections and *generalizable to training policies for different robot hands*.

We highlight our main contributions as follows:

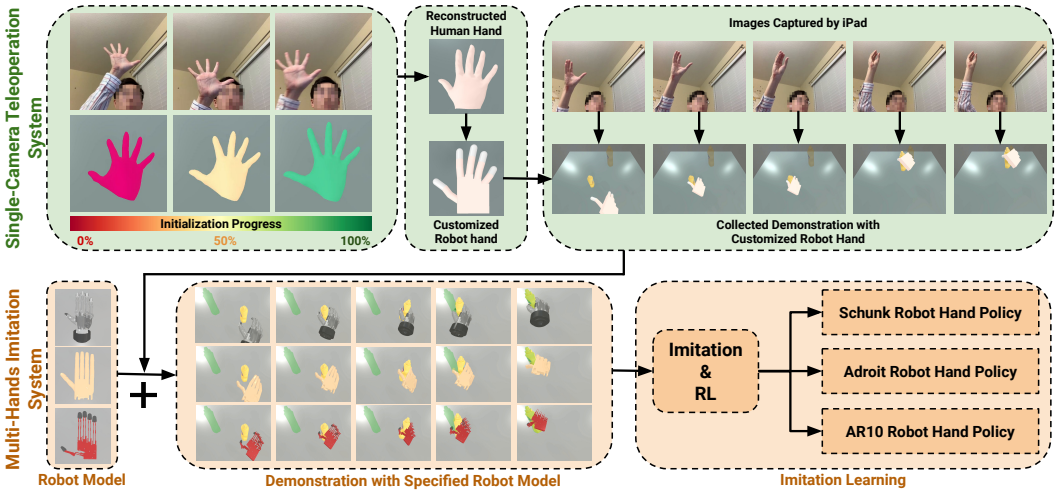


Figure 2: **Overview of our Teleoperation and Imitation System** We develop a teleoperation system with a paired imitation learning system. In the single-camera teleoperation system, we collect human demonstrations on dexterous tasks in simulator. In the imitation learning system, we perform motion retargeting to obtain demonstrations for multiple robot hands.

- We develop a single-camera teleoperation system to collect demonstrations and a paired imitation system to learn dexterous skills for multiple robot hands. To the best of our knowledge, it is the first teleportation system for dexterous manipulation with a single camera.
- We propose a novel customized robot hand, which is constructed on the fly based on the hand shape of a specific human operator.
- We show that the collected demonstration can effectively improve the performance of the policies trained with multiple specified robot hands.

2 Related Work

Dexterous Manipulation. Manipulation with dexterous robot hands has been long studied in robotics and it remains to be one of the most challenging control task (Rus, 1999; Okamura et al., 2000; Andrews & Kry, 2013; Bai & Liu, 2014). Recently, we have witnessed Reinforcement Learning (RL) approaches (OpenAI et al., 2018; 2019) delivering promising results on complex in-hand manipulation tasks. While these results are encouraging, RL suffers from poor sample efficiency in training. Under a high degree of freedom (more than 20 in most hands), the RL policy can easily explore unexpected behaviors without well-designed rewards and external constraints.

Imitation Learning from Demonstrations. Learning from human demonstrations can not only provide external constraint for the robot to explore the expected human-like behaviors but also largely reduces sample efficiency. Beyond behavior cloning (Pomerleau, 1989; Young et al., 2020), imitation learning has been widely studied in the form of Inverse Reinforcement Learning (Ng et al., 2000; Abbeel & Ng, 2004; Ho & Ermon, 2016; Fu et al., 2017; Torabi et al., 2018; Aytar et al., 2018) and incorporating expert demonstrations into the RL objectives (Peters & Schaal, 2008; Duan et al., 2016; Večerík et al., 2017; Rajeswaran et al., 2018; Radosavovic et al., 2021). Our work is highly inspired by Rajeswaran (Rajeswaran et al., 2018), where a VR setup is proposed to collect demonstrations for dexterous manipulation and an algorithm named Demo Augmented Policy Gradient (DAPG) is introduced for imitation learning. However, data collection with VR requires a lot of human effort and is not scalable. We propose to collect data via a single-camera teleoperation system, which makes the process scalable and accessible for different users Our work is also related to imitation learning from human videos (Schmeckpeper et al., 2020; Shao et al., 2020; Song et al., 2020; Young et al., 2020). However, most of these works focus on a 2-jaw parallel gripper and relatively simple tasks, where 3D information is not necessary. Our teleoperation system provides critical 3D hand-object pose information for guiding dexterous manipulation.

Vision-based Teleoperation. Vision-based teleoperation frees the operator from wearing data capture devices, which is commonly used in the game industry(Zhang, 2012). This technique has been used for robot teleoperation (Kofman et al., 2007; Du et al., 2012; 2010; Almetwally & Mallem, 2013) on manipulation tasks, e.g. pick and place with a parallel gripper. DexPilot (Handa et al., 2020) is a pioneering work to extend the vision-based teleoperation to manipulation with an Allegro Hand (Robotics). To capture the hand pose, a black-clothed table with four calibrated RealSense cameras is used in their system. Our work only requires a single iPad camera for teleoperation without the controlled environment. Our novel customized robot hand also provides a more intuitive way for data collection and allows generalization for learning with multiple specified robot hands, which has not been shown before.

Motion Imitation. Once the demonstrations are generated, we perform imitation learning for a policy that can be generalized to different object configurations. We stress this is different from the works which train a policy to follow one expert demonstration (Peng et al., 2018; Liu et al., 2018; Pathak et al., 2018; Sharma et al., 2018; Garcia-Hernando et al., 2020; Sieb et al., 2020; Xiong et al., 2021).

3 OVERVIEW

We propose a novel framework for imitation learning from single-camera teleoperation. As shown in Figure 2, our framework is composed of a single-camera teleoperation system and an imitation learning system.

(i) Single-Camera Teleoperation System to collect demonstrations for dexterous manipulation tasks. It only requires video streaming from an iPad. A key innovation of the system is constructing a customized robot hand on the fly based on the estimated shape of the operator's own hand. The human operators can then control the customized robot hand in a physical simulation environment to perform dexterous manipulation tasks. The demonstrations can be efficiently collected with around *60 demonstrations per hour*.

(ii) Imitation Learning System, which can learn dexterous manipulation skills for multiple robot hands from one set of collected demonstrations. Given the collected trajectories from the teleoperation system, we perform motion retargeting to convert the demonstrations on customized hands to the specified robot hands. These new trajectories are then used to train policies on the same task with the corresponding robot hand. Our system can efficiently learn dexterous skills on complex tasks which are hard to solve by RL alone.

4 SINGLE-CAMERA TELEOPERATION SYSTEM

The hardware of our teleoperation system consists of an iPad and a laptop as shown in Figure 3. We use the front camera of an iPad to stream the RGB-D video of the human operator at 25 fps. The teleoperation system consists of three components, a physical simulator, a hand detector to capture human motion, and a GUI to visualize the current simulation environment for the human operator. We use a laptop with an RTX 2070 GPU. The processing time for each RGB-D frame is less than 30ms, which means the whole teleoperation system can run at 25 fps, the same as camera frequency.

4.1 TASK DESCRIPTION

We adopt SAPIEN(Xiang et al., 2020) as our physical simulator and design multiple dexterous manipulation tasks there. The environments are used for both demonstration collection and policy learning. We develop 2 types of manipulation tasks with different objects. The objects are select from YCB dataset (Calli et al., 2015) with reasonable sizes for human to manipulate.

Relocate. The robot needs to pick up an object on the table and to a target position. It requires the agent to manipulate the object so that it can match the given goal. The first three rows of Figure 1 illustrate the relocate task with three objects of different geometry. The goal pose is visualized using the transparent object. It is a goal-conditioned task where we *randomize both the initial pose and the goal pose* for each trial. The task is evaluated based on the distance between the final object position and the goal position.

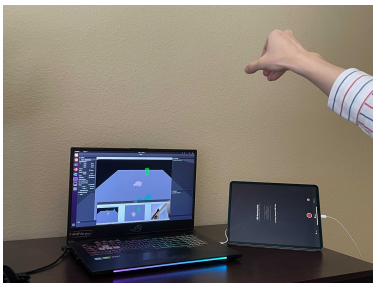


Figure 3: Hardware setup with an iPad and a computer.

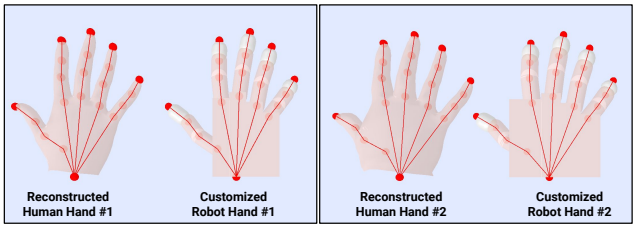


Figure 4: Illustration of different customized robot hands generated from different human hands. The hand on left and right comes from different human. The red dots are the joints and the red lines visualize the kinematics tree.

Flip. As shown in row 4 of Figure 1, it requires the robot to flip a mug on the table. The robot needs to rotate the object 90 degrees carefully to avoid pushing the object away. This task evaluates robot's ability to exert force in a certain direction. We *randomize the position and the horizontal rotation* of the mug for each trial. This task is evaluated based on the final orientation of the object.

4.2 Hand Detector

Our hand detector takes as input the RGB-D frames and outputs the wrist pose (6-d), hand pose parameters (45-d), and hand shape parameters (10-d). It is implemented based upon MediaPipe (Zhang et al., 2020) and FrankMocap (Rong et al., 2020). First, we use MediaPipe hand tracker to detect the axis-aligned bounding-box and crop the image around the hand region. The cropped images are then fed into FrankMocap to estimate the pose and shape parameters. It will detect 21 3D key points on the human hand. We use SMPLX (Pavlakos et al., 2019) model to represent pose and shape parameters. It parameterizes the hand by shape parameters for the hand geometry and pose parameters for the deformation. Given the shape and pose parameters, we can reconstruct a hand in the canonical frame where the wrist is placed at the origin. Then we adopt the Perspective-n-Point (PnP) algorithm to match the key points in the canonical and the detected key points in the camera frame to solve the transformation of the wrist to the camera. The outputs of the hand detector to the downstream modules are wrist pose, hand pose parameters, and hand shape parameters.

4.3 Teleoperation Pipeline

As shown in the top row of Figure 2, our teleoperation pipeline includes two steps: (a) Initialization; and (b) Customized Robot Hand Construction.

Initialization The camera captures the RGB-D image of the human operator, detects the human hand and estimates the hand shape and pose. In teleoperation, human operators naturally interpret their motion commands to the robots with respect to an egocentric frame (Kozhevnikov & Hegarty, 2001; Stransky et al., 2010). It is common to build a global reference frame in the real world that correspond to a frame defined in the simulated environment (Vuong et al., 2021; Li et al., 2019). So another purpose of the initialization step is to define the global reference frame by the operator. During initialization, the operator is required to spread five fingers as shown in the first green rounded rectangle of Figure 2, which offers the best view of the human hand for the camera. This design ensures better reconstruction accuracy for the human hand and generates a customized robot hand that fits the human. We render a human hand turning from red to green to indicate the initialization progress. The whole process takes 5 seconds. See supplementary video for more details.

Customized Robot Hand Our system builds a customized robot hand based on the hand geometry of each user. Given the shape parameters from initialization, we can reconstruct a human hand at rest pose. We then build an articulated hand model in the physical simulator based on the reconstructed human hand. We extract the joint skeleton of the human hand (the red lines in Table 4) and create a robot model with the same kinematics structure. We choose primitive shapes, e.g. box for the palm and capsules for fingers, for efficient collision detection (Kockara et al., 2007) and stable simulation (Nvidia). To ensure that our customized hand can achieve any motion made by human,

we build three revolute joints for each joint on human hands. The customized hand has 45 (15*3) DoF, which matches the SMPLX model. In this way, we can directly rotate the joints of customized robot hand using detected pose parameters without motion retargeting. Table 4 shows different human hands and the corresponding customized hand. In this figure, the right human hand has a shorter thumb. This characteristic reflects in the customized robot hand.

We use a PD controller to control the joint angles of the customized robot hand. With each hand detection, we set the estimated pose passed by a low-pass filter as the position target. One challenge of visual teleoperation is perception error. When the human hand is vertical to the camera plane, it makes the detection results not so reliable and in turn generates an abrupt motion.

To tackle this issue, we utilize the hand shape estimation results as a confidence score and use it in PD control. Since the shape parameters are estimated from the best view during initialization, we can use them as the ground-truth hand shape. Our intuition is that when the camera view is not reliable, both shape and pose estimation results will suffer from errors. Thus we can compute the error of shape parameters by comparing with the ground-truth and use it as a confidence score for pose accuracy. This problem can be formulated using normal distribution: $s_t \sim \mathbf{N}(s_0, \Sigma)$, where s_t and s_o are the shape parameters from t frame and initialization, respectively. The covariance Σ is set to be a diagonal matrix. We compute the normalized probability density $p(t)$ as the confidence score. The confidence-based PD position control is,

$$u(t) = p(t)K_p e(t) + k_d \frac{de(t)}{dt} \tag{1}$$

, where $u(t)$ is the joint torque and k_p and k_d are PD parameters. When the perception error is large, we reduce the stiffness of controller. This design eliminates the undesired abrupt motion caused by perception error.

5 Imitation Learning System

5.1 Motion Retargeting

Table 2 shows the DoF of each finger for different robot models. The finger DoF is given in the following order: Thumb, Index, Middle, Ring, Pinky. Due to the discrepancy of DoF and kinematics tree, we need to convert the demonstration from the customized robot hand to a specified robot model, namely motion retargeting. With our customized robot hand design, we can skip the motion retargeting during teleportation, which is often computational heavy to run on the fly. And after data collection, we use optimization based motion retargeting to process the demonstration and use it for imitation learning. The optimization objective is defined based on the keypoints on the robot hand as,

$$\begin{aligned} \min_{q_t^R} \sum_{i=0}^{N} ||f_i^C(q_t^C) - f_i^R(q_t^R)||^2 + \alpha ||q_t^R - q_{t-1}^R||^2 \\ \text{s.t.} \quad q_{lower}^R \leq q_t^R \leq q_{upper}^R \end{aligned} \tag{2}$$

, where q_t^C is joint angles at step t for customized robot and q_t^R is joint angles at step t for a specific robot, e.g. Schunk Robot Hand. We use f_i^C and f_i^R to represent the forward kinematics function $i-th$ keypoints on the two robots. To improve the temporal consistency, we add a normalization term to penalize the joint angle change and initialize q_t^C using the value from q_{t-1}^C.

5.2 Imitation Learning Method

Given the retargeted demonstration, we perform imitation learning to solve the dexterous tasks defined in subsection 4.1. Note that naive behavior cloning may be hard to work with randomized initial and target pose. Instead, we adopt imitation learning algorithms which incorporate the demonstration into RL. Specifically, we use Demo Augmented Policy Gradient (DAPG) (Rajeswaran et al., 2017) as our imitation algorithm. DAPG combines the experiences collected in demonstration and interaction into a augmented policy gradient objective,

$$g_{aug} = \sum_{(s,a)\in\rho_{\pi_\theta}} \nabla \ln \pi(a|s) A^\pi(s,a) + \sum_{(s,a)\in\rho_{\pi_{\text{demo}}}} \nabla \ln \pi_\theta(a|s) \lambda_0 \lambda_1^k \max_{(s',a')\in\rho_\pi} A^\pi(s',a')$$

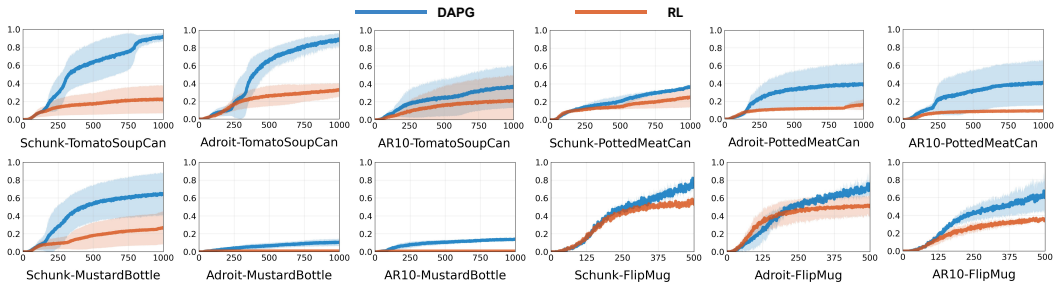


Figure 5: Learning curves of the RL and DAPG on the four tasks with three different robot hands. The x-axis is training iterations and y-axis is the normalized reward. The shaded area indicates the standard deviation for three individual random seeds for all curves.

Task	RL	DAPG
Toma.	45.3 ± 4.0	85.0 ± 12.3
Pott.	6.7 ± 6.3	41.0 ± 20.3
Must.	44.0 ± 20.0	75.3 ± 24.7
Mug	48.0 ± 4.3	77.3 ± 5.0

Schunk Robot

Task	RL	DAPG
Toma.	41.7 ± 30.3	95.0 ± 3.0
Pott.	0 ± 0	53.3 ± 37.7
Must.	0 ± 0	0 ± 0
Mug	28.7 ± 17.7	54.7 ± 15.3

Adroit Robot

Task	RL	DAPG
Toma.	28.3 ± 12.7	37.3 ± 10.2
Pott.	0 ± 0	6.3 ± 3.3
Must.	0 ± 0	0 ± 0
Mug	45.0 ± 20.8	76.7 ± 16.2

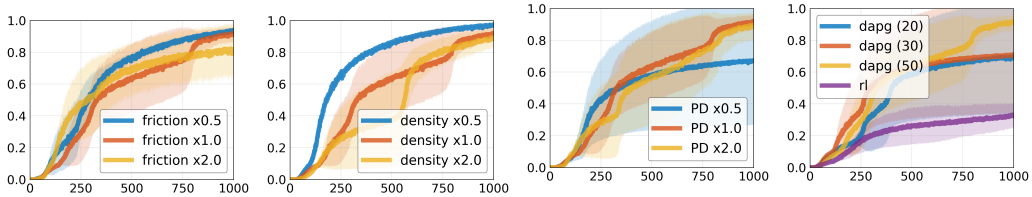
AR10 Robot

Table 1: Success rate of the evaluated methods on the relocate task and flip task. We use $\pm$ to represent mean and standard deviation over three random seeds. Relocate task (Toma., Pott., Must.) has three different objects: tomato soup can, potted meat can, and mustard bottle. Flip task (Mug) has one object: mug. The success of relocate is defined based on the distance between object and target. The success of flip is defined based on the orientation of the object at the end of episode.

, where the first line is the vanilla policy gradient objective and the second line is behavior cloning from demonstration. ρ_π is the occupancy measure under policy π, λ_0 and λ_1 are hyper-parameters, and k is the training iterations. The $A^\pi(s', a')$ is the advantage under policy π.

6 Experiment

We evaluate on the tasks of *Relocate* three different objects and *Flip* a mug. We use the processed demonstration to train policy to perform these tasks and compare them with the RL baseline. To train the policy, we adopt imitation learning algorithms which incorporate the demonstrations into RL. We ablate how friction, PD Controller Parameters, Object Density, and the number of demonstrations can affect the learning process. For the RL baseline, we use Trust Region Policy Optimization(TRPO) (Schulman et al., 2015) as the on-policy algorithm. Both policy and value function are 32×32 2-layer Multi-Layer Perceptrons (MLPs). The TRPO will use 200 trajectories for each step. The imitation learning algorithm is DAPG described in section 5. DAPG adopts TRPO using the same hyper-parameters with demonstrations. We collect 50 trajectories of demonstration for each task and retarget the motion from customized hand to the specified robot. We train policies



(a) Object Friction (b) Object Density (c) PD Controller Params (d) Num of Demonstrations

Figure 6: **Ablation Study**: Learning curves of DAPG on the *Relocate* task with tomato soup can using Schunk Robot Hand. We ablate: (a) friction parameter of the relocated object; (b) density of object; (c) PD controller parameters: stiffness and damping; (d) number of demonstrations used to train DAPG . The demonstrations are kept the same for all conditions.

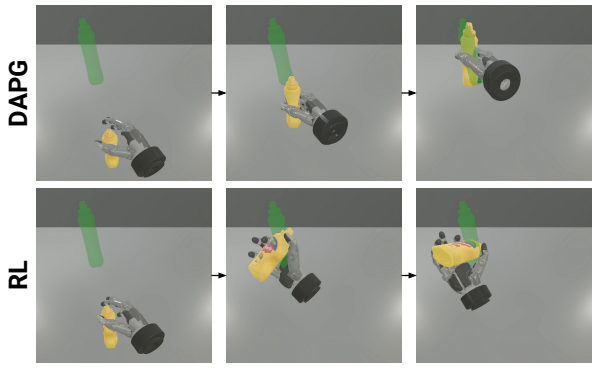


Figure 7: Comparison of the naturalness on *Relocate* with mustard bottle using Schunk robot hand. **Top Row**: policy learned by DAPG with demonstrations; **Bottom Row**: policy learned by RL without demonstrations.

Robot	Finger DoF
Schunk	$(4,4,3,4,5)$
Adroit	$(5,4,4,4,5)$
AR10	$(2,2,2,2,2)$
Customized	$(9,9,9,9,9)$

Table 2: DoF comparison for different robot models. Customized stands for the customized hand robot hand in subsection 4.3. The number in the table follows the order of: Thumb, Index Finger, Middle Finger, Ring Finger, and Little Finger. The customized hand has the same DoF for each finger.

with three different random seeds. For *relocate* task both initial position and target position are randomized for both training and evaluation. For *flip* task, the initial position is randomized.

State and Action Space contain the angles of robot joints, the velocity of hand palm, object position, and orientation. We also include target position for *relocate* task. The action space is composed of two parts: hand palm and finger joints. The motion of the palm is controlled by 6 velocity controllers (3 for translation, 3 for rotation). And the finger joints are actuated by position controllers. The dimension of action is $6+n_q$, where n_q is the DoF shown in Table 2.

6.1 Main Comparison

We evaluate both RL and DAPG on *relocate* and *flip* tasks. The training curves are shown in Figure 5. The y-axis is the normalized return and the x-axis is the number of steps. The success rate is summarized in Table 1. For *relocate*, the task is considered as successful when the object position is within 0.1 unit length to the target at the end of the episode. For *flip*, the robot will get success when the orientation of mug is flipped back, where the angle between the negative z-axis and the direction of gravity is less than 5 degree.

As shown in Figure 5 and Table 1, imitation learning method outperforms the RL baseline for most tasks except the mustard bottle and for all robots. The performance of DAPG using Schunk is much better than RL at all tasks. We find that for both RL and DAPG, *relocate* with a mustard bottle using Adroit and AR10 robot is very challenging. The reason is that the thumb and other four fingers can not form a tight shape closure even at the joint limit. While for the Schunk robot, the freedom of the thumb is large enough to grasp the object. On the other hand, Adroit achieves the best across all three robots on *relocate* with a tomato soup can. This indicates the existence of robot-specific skills. Different robot hands are designed to fit objects with different geometry and a single robot hand can hardly do best for all tasks. The results highlight the importance of using a customized hand to collect demonstration that can support all tasks by motion retargeting to multiple robot hands.

6.2 Ablation Study

To investigate the influence of different dynamics conditions and the number of demonstrations, we ablate the object friction, robot controller parameters, object density, and the number of demonstrations. We choose the *relocate* task with tomato soup can using Schunk robot as the task for ablation study. Figure 6 (a) shows that the learning curve is robust to friction change. To hold the object firmly, a two-finger parallel-jaw gripper usually needs to form an antipodal grasp (Chen & Burdick, 1993), which is sensitive to friction change. Different from parallel-gripper, the dexterous hand can form generate force closure with multiple contact points, thus can withstand smaller friction. Similar results can also be found in Figure 6 (b) where the density is changing. The final performance with $x2$ density is comparable to normal density. Figure 6 (c) illustrates the influence of controller parameters on the learning curve. With larger stiffness, the robot can move the object to the target sooner and get a larger reward. Figure 6 (d) shows more demonstrations can achieve better perfor-

mance. We also observe that when using only 20 or 30 demos, the variance of training curves is larger.

6.3 Visualization of Policy Behavior

In this experiment, we visualize the policy trained by DAPG and RL in Table 7. We observe that DAPG tries to grasp the mustard bottle in a natural behavior while RL policy lifts the object by rotating the wrist. These results highlight the value of demonstration to regulate the behavior of policy to be expected and safe. This phenomenon is especially important for training embodied agents when we want the agent to learn to interact with the environment in the desired way.

7 Conclusion

We propose a novel single-camera teleoperation system to collect human hand manipulation data for imitation learning. We introduce a novel customized robot hand, providing a more intuitive way for different human operators to collect data. We show the collected demonstrations can improve the learning of dexterous manipulation on multiple robots, when the data collection only needs to be conducted once.

References

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

Ismail Almetwally and Malik Mallem. Real-time tele-operation and tele-walking of humanoid robot nao using kinect depth camera. In *2013 10th IEEE International Conference on networking, sensing and control (ICNSC)*, pp. 463–466. IEEE, 2013.

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

Dafni Antotsiou, Guillermo Garcia-Hernando, and Tae-Kyun Kim. Task-oriented hand motion retargeting for dexterous manipulation imitation. In *ECCV Workshops*, 2018.

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

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

Berk Calli, Aaron Walsman, Arjun Singh, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. Benchmarking in manipulation research: The ycb object and model set and benchmarking protocols. *arXiv*, 2015.

I-Ming Chen and Joel W Burdick. Finding antipodal point grasps on irregularly shaped objects. *IEEE transactions on Robotics and Automation*, 9(4):507–512, 1993.

Guang-Long Du, Ping Zhang, Li-Ying Yang, and Yan-Bin Su. Robot teleoperation using a vision-based manipulation method. In *2010 International Conference on Audio, Language and Image Processing*, pp. 945–949. IEEE, 2010.

Guanglong Du, Ping Zhang, Jianhua Mai, and Zeling Li. Markerless kinect-based hand tracking for robot teleoperation. *International Journal of Advanced Robotic Systems*, 9(2):36, 2012.

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

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

Guillermo Garcia-Hernando, Edward Johns, and Tae-Kyun Kim. Physics-based dexterous manipulations with estimated hand poses and residual reinforcement learning. *arXiv*, 2020.

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.

Hooman Hedayati, Michael Walker, and Daniel Szafir. Improving collocated robot teleoperation with augmented reality. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pp. 78–86, 2018.

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

Sinan Kockara, Tansel Halic, Kamran Iqbal, Coskun Bayrak, and Richard Rowe. Collision detection: A survey. In *2007 IEEE International Conference on Systems, Man and Cybernetics*, pp. 4046–4051. IEEE, 2007.

Jonathan Kofman, Siddharth Verma, and Xianghai Wu. Robot-manipulator teleoperation by markerless vision-based hand-arm tracking. *International Journal of Optomechatronics*, 1(3):331–357, 2007.

Maria Kozhevnikov and Mary Hegarty. A dissociation between object manipulation spatial ability and spatial orientation ability. *Memory & cognition*, 29(5):745–756, 2001.

Vikash Kumar and Emanuel Todorov. Mujoco haptix: A virtual reality system for hand manipulation. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pp. 657–663. IEEE, 2015.

Vikash Kumar, Zhe Xu, and Emanuel Todorov. Fast, strong and compliant pneumatic actuation for dexterous tendon-driven hands. In *ICRA*, 2013.

Shuang Li, Xiaojian Ma, Hongzhuo Liang, Michael Görner, Philipp Ruppel, Bin Fang, Fuchun Sun, and Jianwei Zhang. Vision-based teleoperation of shadow dexterous hand using end-to-end deep neural network. In *ICRA*, 2019.

YuXuan Liu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. Imitation from observation: Learning to imitate behaviors from raw video via context translation. In *ICRA*, 2018.

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

Nvidia. PhysX physics engine. `https://www.geforce.com/hardware/technology/physx`.

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

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.

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.

Yong Pan, Chengjun Chen, Dongnian Li, Zhengxu Zhao, and Jun Hong. Augmented reality-based robot teleoperation system using rgb-d imaging and attitude teaching device. *Robotics and Computer-Integrated Manufacturing*, 71:102167, 2021.

Deepak Pathak, Parsa Mahmoudieh, Guanghao Luo, Pulkit Agrawal, Dian Chen, Yide Shentu, Evan Shelhamer, Jitendra Malik, Alexei A. Efros, and Trevor Darrell. Zero-shot visual imitation. In *ICLR*, 2018.

Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Xue Bin Peng, Angjoo Kanazawa, Jitendra Malik, Pieter Abbeel, and Sergey Levine. Sfv: Reinforcement learning of physical skills from videos. *TOG*, 2018.

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

Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. In *NeurIPS*, 1989.

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

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.

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.

WONIK Robotics. Allegro robot hand. `https://www.wonikrobotics.com/research-robot-hand`.

Active Robots. Ar10 humanoid robotic hand. `https://www.active-robots.com/ar10-humanoid-robotic-hand.html`.

Yu Rong, Takaaki Shiratori, and Hanbyul Joo. Frankmocap: Fast monocular 3d hand and body motion capture by regression and integration. *arXiv preprint arXiv:2008.08324*, 2020.

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

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

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

Schunk. Schunk svh robot hand. `https://schunk.com/us_en/gripping-systems/highlights/svh`.

Lin Shao, Toki Migimatsu, Qiang Zhang, Karen Yang, and Jeannette Bohg. Concept2robot: Learning manipulation concepts from instructions and human demonstrations. In *RSS*, 2020.

Pratyusha Sharma, Lekha Mohan, Lerrel Pinto, and Abhinav Gupta. Multiple interactions made easy (mime): Large scale demonstrations data for imitation. *arXiv*, 2018.

Maximilian Sieb, Zhou Xian, Audrey Huang, Oliver Kroemer, and Katerina Fragkiadaki. Graph-structured visual imitation. In *CoRL*, 2020.

Shuran Song, Andy Zeng, Johnny Lee, and Thomas Funkhouser. Grasping in the wild: Learning 6dof closed-loop grasping from low-cost demonstrations. *Robotics and Automation Letters*, 2020.

Debi Stransky, Laurie M Wilcox, and Adam Dubrowski. Mental rotation: cross-task training and generalization. *Journal of Experimental Psychology: Applied*, 16(4):349, 2010.

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

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.

Quan Vuong, Yuzhe Qin, Runlin Guo, Xiaolong Wang, Hao Su, and Henrik Christensen. Single rgb-d camera teleoperation for general robotic manipulation. *arXiv preprint arXiv:2106.14396*, 2021.

Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, et al. Sapien: A simulated part-based interactive environment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11097–11107, 2020.

Haoyu Xiong, Quanzhou Li, Yun-Chun Chen, Homanga Bharadhwaj, Samarth Sinha, and Animesh Garg. Learning by watching: Physical imitation of manipulation skills from human videos. *arXiv*, 2021.

Sarah Young, Dhiraj Gandhi, Shubham Tulsiani, Abhinav Gupta, Pieter Abbeel, and Lerrel Pinto. Visual imitation made easy. *arXiv*, 2020.

Fan Zhang, Valentin Bazarevsky, Andrey Vakunov, Andrei Tkachenka, George Sung, Chuo-Ling Chang, and Matthias Grundmann. Mediapipe hands: On-device real-time hand tracking. *arXiv preprint arXiv:2006.10214*, 2020.

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)*, pp. 5628–5635. IEEE, 2018.

Zhengyou Zhang. Microsoft kinect sensor and its effect. *IEEE multimedia*, 19(2):4–10, 2012.