The supplementary material of Learning Cross-hand Policies of High-DOF Reaching and Grasping

Qijin She[1], Shishun Zhang[1], Yunfan Ye[3], Ruizhen Hu[2], and Kai Xu[1⋆]

[1] National University of Defense Technology, China
[2] Shenzhen University, China
[3] Hunan University, China

1 Method Details

1.1 Termination conditions

Our termination conditions consist of two requirements. Firstly, in each decision step, a random value is sampled from the range $[0, 1]$. If this value is less than the predicted stop value a_s within the range of $[-1, 1]$, the first requirement is met. The other requirement necessitates that at least two gripper components are sufficiently close to the object in current step. To determine whether a gripper component g_i is close enough to the object, we ensure that there are enough points on the gripper component that are close enough to the object. We count the number m_i of IBS points with the distance to the object smaller than a given threshold $\delta_d = 0.5cm$. If there are enough closing points ($m_i \geq 3$), we consider the gripper components close enough to the object.

1.2 Action ranges

In the policy model, we limit the predicted displacement of each point $\{\Delta p_k^i\}$ in $[-0.4cm, 0.4cm]^3$. The change of global translation and rotation change is constrained to be within Δp in $[-2.5cm, 2.5cm]^3$ and Δr in $[-0.025, 0.025]^3$ in radian. It is important to note that any displacement of the middle phalanx will be transferred to the fingertip on the same finger, resulting in real position changes of the fingertips within $[-0.8cm, 0.8cm]^3$. Regarding the adaptation model, We restrict the joint changes in $[-0.25, 0.025]$ in radian.

1.3 Training details

Reward function We adopt the reward function and training strategy provided by [9]. The reward function comprises two components: "Task Reward" and "Reaching Reward". Grasp reward evaluates the final grasp from two perspectives. First, we use the success signal S obtained from the simulator when

⋆ Corresponding author. Email: kevin.kai.xu@gmail.com

performing the final grasp. We consider a grasp to be successful if and only if the object can be lifted up by more than $0.2m$ when the gripper is vertically raised by $0.6m$. Second, the generalized Q_1 measure [6] is integrated to analyze the grasp stability in the geometric aspect.

Given that the two aforementioned metrics are calculated only upon task termination, a negative reward is assigned for each intermediate step to promote fast convergence. Consequently, the grasp reward function is formulated as follows:

$$R_g = \begin{cases} \omega_s S + \omega_q Q_1, & \text{if the task terminates;} \\ r_f + w_p a_s, & \text{otherwise.} \end{cases}$$

where we set $\omega_s = 180$, $\omega_q = 1200$, and $r_f = -3$ and $w_p = -0.25$ are constants and a_s in $[-1, 1]$ is the stop value predicted by the policy network.

The reaching reward mechanism is intended to prevent collisions between the gripper and the object, while also promoting contact between them. The collision means the shape of the gripper penetrates the object, which can be approximately judged by IBS points. This is defined as:

$$R_c^i = \begin{cases} -20, & \text{collide with the scene} \\ R^i_{\text{contact}}, & \text{otherwise.} \end{cases}$$

$$R^i_{\text{contact}} = \begin{cases} \lambda \min\{\eta, m_i\}, & m_i \geq \delta_m (\text{have enough contact}); \\ 0, & \text{otherwise.} \end{cases}$$

In all our experiments, we set $\lambda = 0.25$, $\delta_n = \delta_m = 3$ and $\eta = 10$.

Training algorithm The training of the policy model is based on Soft Actor-Critic [3] with vectorization enhancements. SAC has two networks, i.e., the policy(actor) network and the Q(critic)-network. Given an input state s, the policy network outputs a Gaussian distribution $\pi(*|s;\theta)$ for action sampling, where θ are the parameters of the network. Q-network outputs the evaluation value $Q(s, a; \Phi)$ for given part state s and action a, where Φ are the parameters of the network. SAC uses an additional target Q-network to calculate target value for temporal difference (TD) update, whose parameters are denoted by Φ'. A transition is denoted as a tuple $\{s, a, R, s', d\}$, where R is the reward and d indicates whether state s' is a terminated state. All the transitions will be stored in a replay buffer D and those networks are trained by the data sampled from it.

The output of the Q-network are modified from a single scale value to a vector. The policy network ouput a vector of $(Q_g, Q_c^0, \ldots, Q_c^5)$ to estimate both R_g and R_c^i Accordingly, the reward R in each experience is a vector of $(R_g, R_c^0, \ldots, R_c^5)$. Note that only R_g is accumulated while R_c^i is computed for every single step as immediate constraints.

Throughout the training process, the Q-network and policy network are updated in an alternating manner. The loss function used for training the Q-

network is established through the temporal difference (TD) update method:

$$L_Q(\Phi) = [(Q_g(s, a; \Phi) - y_g(R_g, s', d))^2 + \sum_{i=0}^{5}(Q_c^i(s, a; \Phi) - R_c^i)^2)] \quad (1)$$

and target value y_g for R_g defined as:

$$y_g(R_g, s', d) = R_g + \gamma(1 - d)\ [Q_g(s', \widetilde{a}'; \Phi')\ - \alpha \log \pi(\widetilde{a}|s'; \theta)], \quad (2)$$

where $\widetilde{a}' \sim \pi(*|s'; \theta))$ is the sampled action. $\gamma = 0.99$ is the discount factor, and α is the temperature parameter that can be adjusted automatically to match an entropy target in expectation, to balance exploring the environment and maximizing reward. In the case of the policy network, the loss function $L_Q(\theta)$ is defined as:

$$L_Q(\theta) = [Q_g(s, \widetilde{a}(s; \theta)) + (\sum_{i=0}^{5} Q_c^i(s, \widetilde{a}(s; \theta)) - \alpha \log \pi(\widetilde{a}|s; \theta)], \quad (3)$$

where $\widetilde{a} \sim \pi(*|s; \theta))$ is the sampled action based on the current state and network parameters.

2 Experiment Details

2.1 Experiment platform

We run our method on a home computer equipped with 64GB RAM, RTX 3090TI, and Intel Core 9700K. The joint training stage required approximately 2.5 days for 800k updates, while the transfer training stage took around 20 minutes per 10k updates.

2.2 Grippers details

Table 1 illustrates the Degrees of Freedom (DoF) for each finger across various robot models. The finger DoF is listed in sequence from the pinky finger to the thumb finger. The gripper models utilized originate from diverse sources. The Shadow Hand and Rutgers Hand are sourced from Graspit [7]. The Schunk SVH Hand and Allegro Hand have been adapted from their official libraries. The actuated Mano Hand has been customized from an open-source library that built an URDF file for the Mano hand [8].

Gripper	Finger DoFs
ShadowHand	(4, 3, 3, 3, 5)
SchunkHand	(4, 4, 3, 4, 4)
ManoHand	(4, 4, 4, 4, 4)
RutgersHand	(4, 4, 4, 4, 3)
AllegroHand	(4, 4, 4, 4)

Table 1: Degree of Freedom (DoF) of different robot models.

2.3 Gripper Pose Initalization

For each object, we use its center to establish a sphere with a radius $r = 20cm$. We then select sample points on the upper hemisphere to serve as the local coordinate system origin for the gripper, with the palm's center as the reference point. The gripper is oriented to align its palm with the object's center and position its thumb upwards. During testing, we maintain a consistent set of initial configurations for each object. We partition the hemisphere into an 8x3 grid based on longitude and latitude divisions. Subsequently, we utilize these grid points as the initial positions for the hand.

2.4 Object dataset and result details

We use the three object dataset of Liu [6] while removing objects with volume $v > 1.5dm^3$: KIT [5]: 74 objects; GD [4]: 251 objects; YCB [2]: 48 objects. The first two sets of objects are used for training purposes, while the remaining set is allocated for testing.

We further narrowed down the similar objects in YCB and left 25 objects, resulting in a final selection of 25 objects. To enhance the assessment of our method, we incorporated the ContactPose dataset [1] for testing purposes, our evaluation utilized the same set of 25 objects as outlined in their paper.

We present the test results of our method on the aforementioned two object datasets using various dexterous hands in Table 2 and Table 3. The success rate denotes the average success rate across different initial positions.

Table 2: The success rate for each object in the YCB dataset

Shape	Shadow	Schunk	Mano	Rutgers	Allegro
ycb_002_master_chef_can	0.67	0.69	0.77	0.73	0.81
ycb_004_sugar_box	0.77	0.75	0.79	0.58	0.58
ycb_008_pudding_box	0.88	0.62	0.67	0.69	0.67
ycb_010_potted_meat_can	0.73	0.83	0.79	0.83	0.85
ycb_011_banana	0.77	0.67	0.67	0.56	0.60
ycb_013_apple	0.88	0.83	0.73	0.73	0.65
ycb_014_lemon	0.83	0.83	0.77	0.44	0.44
ycb_021_bleach_cleanser	0.71	0.46	0.38	0.60	0.58
ycb_033_spatula	0.65	0.62	0.46	0.46	0.40
ycb_037_scissors	0.46	0.42	0.48	0.31	0.12
ycb_038_padlock	0.58	0.50	0.71	0.42	0.52
ycb_044_flat_screwdriver	0.48	0.50	0.50	0.29	0.56
ycb_048_hammer	0.77	0.60	0.67	0.50	0.67
ycb_050_medium_clamp	0.69	0.46	0.69	0.29	0.27
ycb_052_extra_large_clampd	0.46	0.52	0.60	0.50	0.44
ycb_056_tennis_ball	0.79	0.85	0.75	0.52	0.62

Continued on next page

Shape	Shadow	Schunk	Mano	Rutgers	Allegro
ycb_061_foam_brick	0.96	0.85	0.71	0.81	0.79
ycb_063-a_marbles	0.85	0.83	0.79	0.67	0.62
ycb_065-f_cups	0.96	0.88	0.98	0.73	0.60
ycb_070-a_colored_wood_blocks	0.62	0.48	0.29	0.33	0.54
ycb_072-b_toy_airplane	0.62	0.67	0.73	0.46	0.69
ycb_073-b_lego_duplo	0.67	0.62	0.48	0.46	0.17
ycb_073-c_lego_duplo	0.58	0.33	0.52	0.44	0.31
ycb_073-g_lego_duplo	0.69	0.67	0.54	0.67	0.60
ycb_077_rubiks_cube	0.85	0.81	0.83	0.69	0.62

Table 3: The success rate for each object in the ContactPose dataset

Shape	Shadow	Schunk	Mano	Rutgers	Allegro
apple	0.81	0.88	0.73	0.83	0.71
banana	0.50	0.52	0.69	0.52	0.62
binoculars	0.71	0.81	0.73	0.60	0.54
bowl	0.71	0.65	0.73	0.50	0.58
camera	0.73	0.75	0.73	0.69	0.67
cell_phone	0.65	0.42	0.58	0.48	0.46
cup	0.92	0.88	0.90	0.83	0.77
door_knob	0.92	0.96	0.94	0.75	0.62
eyeglasses	0.62	0.54	0.67	0.54	0.65
flashlight	0.56	0.48	0.88	0.31	0.29
hammer	0.54	0.31	0.44	0.19	0.33
headphones	0.44	0.69	0.52	0.42	0.60
knife	0.42	0.15	0.25	0.00	0.15
light_bulb	0.85	0.75	0.63	0.67	0.52
mouse	0.77	0.60	0.67	0.50	0.52
mug	0.92	0.85	0.88	0.81	0.65
pan	0.25	0.25	0.40	0.33	0.42
ps_controller	0.90	0.58	0.58	0.77	0.50
scissors	0.46	0.21	0.42	0.17	0.06
stapler	0.73	0.58	0.62	0.27	0.50
toothbrush	0.10	0.06	0.10	0.04	0.04
toothpaste	0.71	0.27	0.21	0.40	0.38
utah_teapot	0.79	0.69	0.50	0.60	0.60
water_bottle	0.88	0.94	0.88	0.75	0.65
wine_glass	0.88	0.73	0.73	0.69	0.58

2.5 Post-process operation

Note our method ends with a configuration of the gripper, and no force information is attached. In order to physically lift objects, we introduce a post-process operation. The post-process operation can be divided into two steps. First, we close the gripper fingers by moving the joints toward the palm. During the process, if one link contacts the scene. The corresponding joint of the contact link and its ancestral joints will be stopped. The contact link and its corresponding joint are recorded. We call them "contact" joints. The first step ends when all the joints stop or a fixed period of time $t = 0.5s$ has passed. Then, we make all the "contact" joints continue to move following the previous direction, squeezing the object to produce adaptive forces that can form a stable grasp.

References

1. Brahmbhatt, S., Tang, C., Twigg, C.D., Kemp, C.C., Hays, J.: Contactpose: A dataset of grasps with object contact and hand pose. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16. pp. 361–378. Springer (2020)
2. Calli, B., Singh, A., Bruce, J., Walsman, A., Konolige, K., Srinivasa, S., Abbeel, P., Dollar, A.M.: Yale-cmu-berkeley dataset for robotic manipulation research. The International Journal of Robotics Research **36**(3), 261–268 (2017)
3. Haarnoja, T., Zhou, A., Abbeel, P., Levine, S.: Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: International conference on machine learning. pp. 1861–1870. PMLR (2018)
4. Kappler, D., Bohg, J., Schaal, S.: Leveraging big data for grasp planning. In: 2015 IEEE International Conference on Robotics and Automation (ICRA). pp. 4304–4311. IEEE (2015)
5. Kasper, A., Xue, Z., Dillmann, R.: The kit object models database: An object model database for object recognition, localization and manipulation in service robotics. The International Journal of Robotics Research **31**(8), 927–934 (2012)
6. Liu, M., Pan, Z., Xu, K., Ganguly, K., Manocha, D.: Deep differentiable grasp planner for high-dof grippers. In: Proceedings of the Robotics: Science and Systems 2020 (2020)
7. Miller, A.T., Allen, P.K.: Graspit! a versatile simulator for robotic grasping. IEEE Robotics & Automation Magazine **11**(4), 110–122 (2004)
8. Romero, J., Tzionas, D., Black, M.J.: Embodied hands: Modeling and capturing hands and bodies together. arXiv preprint arXiv:2201.02610 (2022)
9. She, Q., Hu, R., Xu, J., Liu, M., Xu, K., Huang, H.: Learning high-dof reaching-and-grasping via dynamic representation of gripper-object interaction. ACM Transactions on Graphics (SIGGRAPH 2022) **41**(4) (2022)