IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2023, London, UK. Cite as RA-L paper.

Multifingered Grasping Based on Multimodal Reinforcement Learning

Hongzhuo Liang[1], Lin Cong[1], Norman Hendrich[1], Shuang Li[1], Fuchun Sun[2], and Jianwei Zhang[1]

***Abstract*—In this work, we tackle the challenging problem of grasping novel objects using a high-DoF anthropomorphic hand-arm system. Combining fingertip tactile sensing, joint torques and proprioception, a multimodal agent is trained in simulation to learn the finger motions and to determine when to lift an object. Binary contact information and level-based joint torques simplify transferring the learned model to the real robot. To reduce the exploration space, we first generate postural synergies by collecting a dataset covering various grasp types and using principal component analysis. Curriculum learning is further applied to adjust and randomize the initial object pose based on the training performance. Simulation and real robot experiments with dedicated initial grasping poses show that our method outperforms two baseline models in the grasp success rate both for seen and unseen objects. This learning approach further serves as a fundamental technology for complex in-hand manipulations based on multi-sensory the system.**

***Index Terms*—Grasping, multifingered hands, reinforcement learning.**

I. Introduction

EVEN though the two-fingered grasping problem has been widely studied and reaches a satisfying success rate, multifingered grasping is still far from solved. The fact that the robotic community is beginning to expect robots to approach the manipulation capabilities of humans makes it important to solve this problem. Even with carefully planned trajectories or a dedicated mechanism design [1], [2], two-fingered grippers can only be used to execute some simple object interactions and manipulate some specific object categories. Therefore, to endow robots with the same dexterity as human hands, which can effectively grasp different objects and utilize various tools like spraying or in-hand rotating a cube [3], anthropomorphic hands have become a promising solution and have gained much attention over the past years of research.

When grasping an unknown object in daily life, humans usually first use visual perception to estimate the object's physical properties, *i.e.*, size, weight and center of mass, and surface smoothness and stiffness, based on prior experience. A rough grasp pose can be estimated from visual information. During the process of approaching and contacting with the object, humans begin to use comprehensive tactile and force feedback to explore the object further and finally choose a stable posture to grasp it.

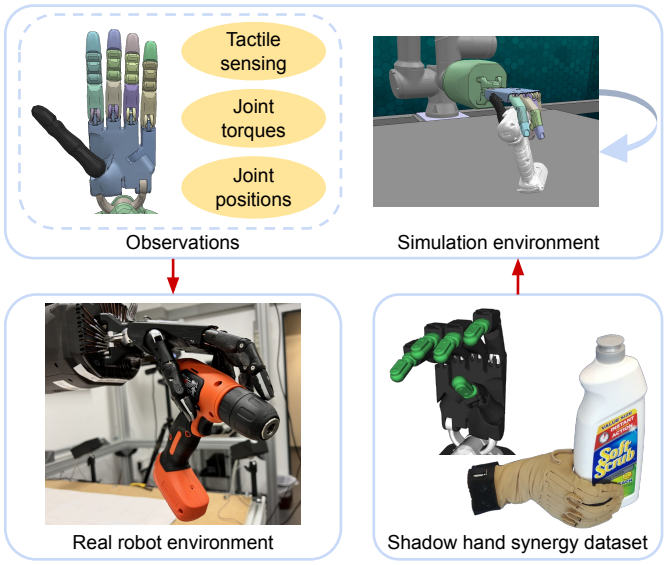


Fig. 1. An illustration of our proposed multifingered hand-arm grasping system. The top block presents our simulated training system. In the simulation, we use multimodal information as observations to train a multifingered grasping agent. To simplify the control of the hand and to make the hand move like a human, in the right bottom block, we first collect a Shadow hand pose dataset using a Cyberglove. We use this dataset to perform a principal component analysis to reduce the number of inputs needed to control the robot hand. The bottom left figure illustrates that the grasping agent purely trained in simulation has proved to work well in the real world by several grasping experiments.

Inspired by how humans grasp objects, we develop a robust robotic grasping strategy by merging multiple sensing modalities with anthropomorphic dexterous hands. The fusion of different data from tactile fingertips, torque sensors, and robot proprioception (joint positions) promises a more robust and intelligent method to teach the robot multifingered grasping.

Therefore, we propose a hand-arm system for multifingered grasping. We start with a point cloud as input and run the grasp evaluation network PointNetGPD [4] to generate a candidate two-fingered grasp, then map this as the pre-grasp pose for the dexterous hand. To control the dexterous hand to execute the grasping action, instead of using a fixed grasping trajectory, we first use hand synergies [5] to make a dimensional reduction. Then we use the reduced dimension information as target

Manuscript received: August 4, 2021; Revised: November 4, 2021; Accepted: December 6, 2021.

This paper was recommended for publication by Editor Hong Liu upon evaluation of the Associate Editor and Reviewers' comments.

This research was funded by the German Research Foundation (DFG) and the National Science Foundation of China (NSFC) in the project Crossmodal Learning, DFG TRR-169/NSFC 61621136008, partially supported by the China Scholarship Council (CSC) and European project H2020 Ultracept (778602).

[1]H. Liang, L. Cong, N. Hendrich, S. Li and J. Zhang are with Group TAMS, Dept. of Informatics, Universität Hamburg.

[2]F. Sun is with the Dept. of Computer Science and Technology, Tsinghua University.

Corresponding author: Lin Cong; Email cong@informatik.uni-hamburg.de

Digital Object Identifier (DOI): see top of this page.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2023, London, UK. Cite as RA-L paper.

action space to train a multimodal reinforcement learning (RL) based agent. With this agent, the dexterous robotic hand can close its fingers and grasp the object successfully.

Our contribution can be summed up as follows:

- We introduce a multifingered grasping agent that fuses multimodal sensor data (fingertip tactile sensing, joint torques, and hand proprioception) based on the reinforcement learning algorithm. The agent is easy to transfer from simulation to the real world platform with the help of binary tactile information and level-based torque information. Our robot experiments prove that our agent trained in simulation works well on the real robot system and outperforms the baseline methods;
- We collect a dataset about common human hand motions and map it to the Shadow robot hand successfully. Because the corresponding motion data of the Shadow hand joints are too complicated and take up a lot of computing memory, we calculate postural hand synergies using principal component analysis (PCA) to reduce the dimension of controlling the hand;
- Through comparative experiments in simulation, we find that the fusion of multimodal sensing increases the performance of the agent, and the policy with the Recurrent Neural Network structure (GRU in our work) is better than that with the Multi Layer Perception structure. We also find that the most suitable dimension reduction value for PCA is 5. Real robot experiments are performed with different models. Through the comparison of model with different modalities and different baselines, we verified the effectiveness of our proposed algorithm.

II. Related Work

A. Multifingered grasping

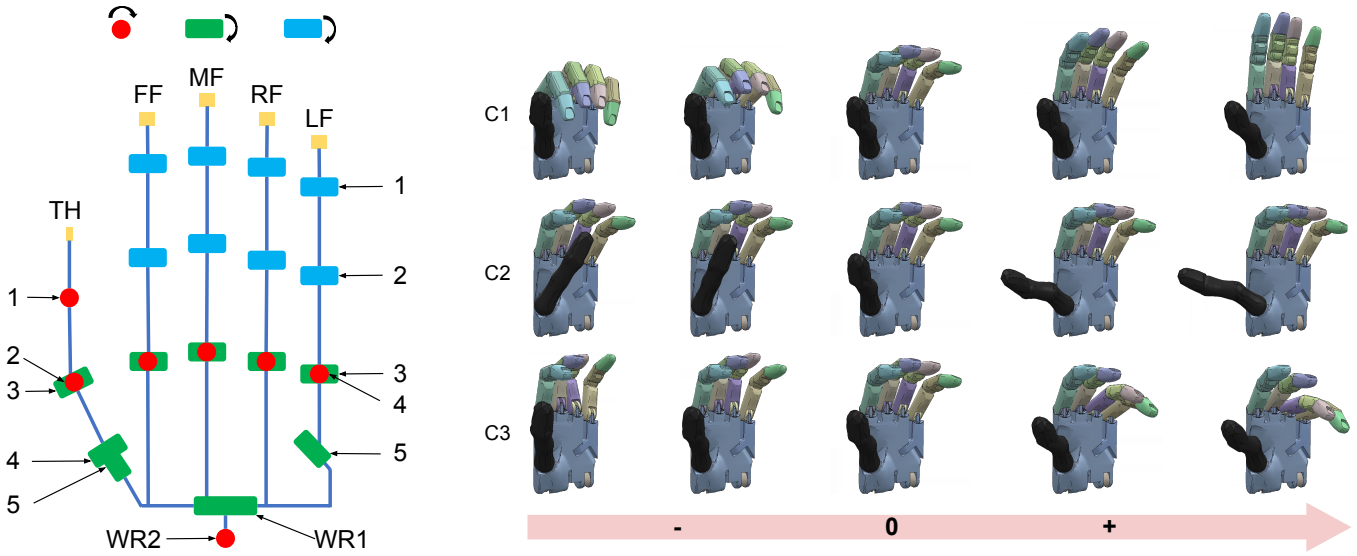
Due to the high DoF (degrees of freedom, 24 for the Shadow hand), it is very difficult to control a dexterous hand flexibly. Even without considering object dynamics, a perfect grasping action based on grasp synthesis for a high DoF dexterous hand is still a challenging task [6]. A lot of previous work has discussed the research on multifingered grasping. Li *et al*. [7] proposed a probabilistic model to address robust dexterous grasping under shape uncertainty, and Fan *et al*. [8] proposed a finger splitting strategy to plan precision grasps for multifingered hands from parallel grasps. However, both examples require prior knowledge of the object model. Shao *et al*. [9] proposed a deep-learning-based method to generate grasp points for multifingered robots without prior knowledge of the object model, but it is an open loop grasping method. In comparison, our algorithm adjusts finger motions in closed loop according to tactile and torque sensing as vision compensation. Brahmbhatt *et al*. [10] presented a novel framework for grasp synthesis based on the surface shape and contact map of the target object. They used a human-demonstrated contact map as a constraint to optimize and refine the grasp candidates from GraspIt! [11]. However, in real robot applications, ideal contact maps are hard to generate for unknown objects. Kumar *et al*. [12] used human hand motion demonstration to initialize and reduce the search space of their multifingered robot hand. However, this requires pose estimation for the objects to get an object bounding box. Ficuciello *et al*. [13] proposed a synergy-based strategy and achieved stable grasps, but the grasping performance critically depends on the quality of hand preshaping. Furthermore, they proposed a hand-arm grasp system based on this work [14], but they only demonstrated a single object scenario and have limited grasp objects. And the training process is in the real world, which makes the whole grasp pipeline time-consuming. Wu *et al*. [15] discretized the finger action space and increased the resolution of the finger joint movement during the training process. At each timestep, the robot decides whether or not to close each finger further in the binary form. However, the robot hand they used is not an anthropomorphic hand, making the agent easier to train. In [16], the authors trained an agent to regrasp based on an initial grasping pose to improve the success rate rather than to grasp the target directly, which also places strong restrictions on the prerequisite of the experiment setup. Object pose and visual feedback are assumed to be unknown in the above work. In [17] both object pose and contact forces were taken as a part of the state in the Markov Decision Process (MDP), and they concluded that contact forces can improve the grasping robustness specifically under pose uncertainty. However, transferring the contact sensing to a real world environment is needed to verify their theory on real hardware.

B. Dimensional reduction for multifingered hands

The human hand is a very high-dimensional, complex, and versatile end-effector system. Such a system, in theory, requires a vast amount of computational time for its control. However, the human nervous system can control hands extremely fast. The result from [5] indicates that human hand control takes place in a subspace of much lower dimension than the original hand's DoF. [18]–[20] further implemented this on a dexterous robot hand. [13] built a reinforcement learning algorithm to confer autonomous grasping to anthropomorphic hands. However, the hand they used for testing is a Schunk dexterous hand with nine motors that actuate 20 joints. Thus, the action space of this hand has been mechanically constrained. Instead, we concentrate on the Shadow Hand, which has 18 motors to control 22 finger joints and 2 motors for two wrist joints. It is difficult to control this hand by traditional methods.

III. Grasp Synergies Dataset

Similar to the human hand, the Shadow hand has five fingers with 24 joints, including 2 active joints on the wrist, 18 active joints on the fingers and 4 coupled joints, as shown in Fig. 2(a). However, it is quite challenging to train an RL agent in such a high-dimensional action space. High-dimensional action equals huge exploration space, which will cause a low learning efficiency and generate weird hand poses without constraints. To ensure that generated hand poses are more human-like and the exploration space can also be greatly reduced, we simplify this problem by creating a motion subspace.



(a) (b)

Fig. 2. (a) Joint mechanics of the Shadow hand. $\theta_w \in \{\text{WR1}, \text{WR2}\}$ refers to wrist joints 1 and 2. And $\theta_f \in \{\text{LF}, \text{RF}, \text{MF}, \text{FF}, \text{TH}\}$ refers to little finger, ring finger, middle finger, first finger, and thumb (22 joints). Joints 1 and 2, marked blue in each finger, are coupled (these two joints are controlled by one motor). (b) C1, C2, and C3 rows show example grasp postures of the Shadow hand controlled by the three first principal components of our dataset.

In the research field of human hand grasping, it is popular to apply the definition of grasp synergies [5] to describe and simplify human grasps by Principal Component Analysis (PCA). Therefore, a Shadow hand pose dataset is necessary to calculate the eigenvectors and eigenvalues representing the most correlated directions in the joint space.

To this end, we collect a Shadow hand pose dataset, using the 33 grasp types of humans described in [21], such as large-diameter grasp and tripod grasp. The first portion of the hand pose dataset is from our previous work [19]. Three human subjects teleoperate an air-muscle version of the Shadow hand with a Cyberglove. Eight precision grasp taxonomies are used to grasp twelve primitive objects and 442 joint samples are recorded. To improve the collection efficiency and the object diversity of the dataset, we further collect 3000 samples by controlling the hand by the Cyberglove in simulation. Furthermore, we extend the hand pose dataset based on the YCB-Affordance dataset [22]. The YCB-Affordance dataset contains manually annotated human grasps covering all 33 grasp taxonomies on the 58 objects of the YCB benchmark set. We use a BioIK [23] solver to calculate a corresponding Shadow hand mapping via the labeled human hand keypoints. After removing some unreachable grasps for the Shadow hand, we finally get more than 200 robot hand postures.

Based on the collected dataset, we consequently perform a principal component analysis to reduce the joint space dimensions used to control the multifingered hand. The n eigenvectors corresponding with the largest n eigenvalues are selected to form a transformation matrix. The optimal number of synergies n is determined by our comparative experiments in section V-A. Fig. 2(b) illustrates the hand motion corresponding to the first three components, which are called the first, second, and third grasp synergies (C1, C2, C3).

IV. Multimodal Grasping Policy

A complete grasp could be defined as consisting of four parts 1) the grasping point describing the end-effector position, 2) the approach vector in which the robot hand approaches the grasping point, 3) the wrist orientation and 4) an initial finger configuration [25]. To generate such a grasp for the Shadow hand, we have to focus on controlling the finger and wrist joints during the RL agent training process. Instead of generating a fixed grasp candidate each time, we propose a two-stage dynamic grasping method where the robot continuously adjusts its motions until a lift-up decision. In the first stage, the hand is trying to do a closing motion from the initial setup according to a human-like hand closing trajectory T until a contact is detected between any of the five fingertips and the object. After that, the hand closing motion is stopped and the robot goes into the second stage. This stage is a closed-loop control process, during which the hand obtains a set of observations from proprioception, binary tactile values of the fingertips, and finger joint torques as elaborated in section IV-B. No visual perception or object model is provided in the simulation environment. The robot adjusts all the joint angles continuously to grasp the object better until it lifts the object. Fig. 3 gives an overview of our multimodal reinforcement learning.

The whole multifingered grasping process is modeled as a finite-horizon discounted MDP. At each timestep t, the agent perceives an observation $o_t \in O$ from the environment, executes an action $a_t \in A$ and obtains a reward $r_t \in R$; details on actions are presented in section IV-C and on the reward function in section IV-D. The agent executes an action according to a stochastic policy $\pi(a_t|O_t)$, which is a distribution over actions conditioned on several recent observations. The goal is to find a policy π that maximizes the expected sum of

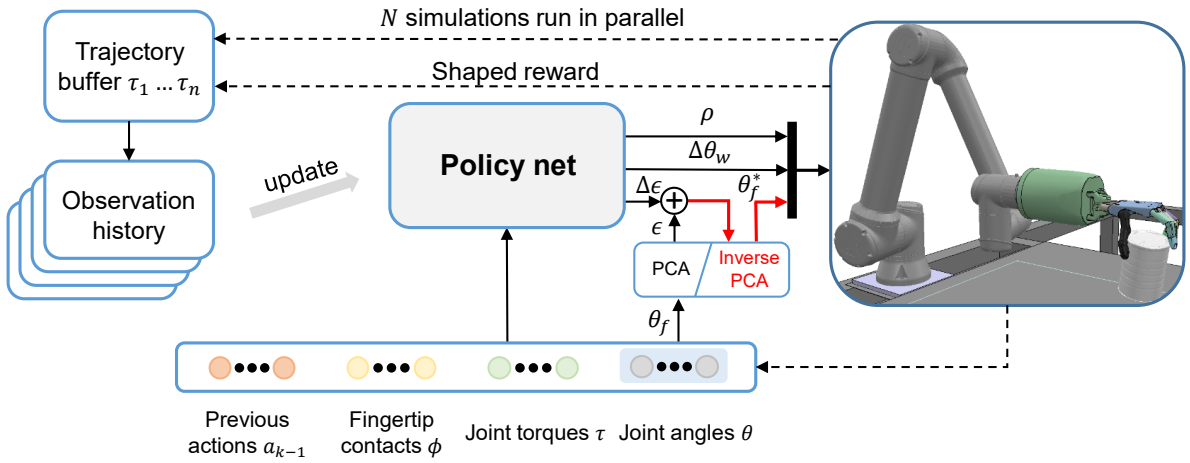


Fig. 3. An overview of our multimodal reinforcement learning structure. At each timestep, four different types of input information are captured from the environment and concatenated as one vector, representing the agent's current state. This state vector is then stacked with several history states as the input and goes into the policy net. Three types of actions come from the policy net: the lifting decision ρ, the wrist rotation increment $\Delta\theta_w$, and the PCA value increment $\Delta\epsilon$ that controls finger joints θ_f^* separately. The observation history buffer keeps track of several previous state transitions. All trials go into the trajectory buffer as training data to update the policy.

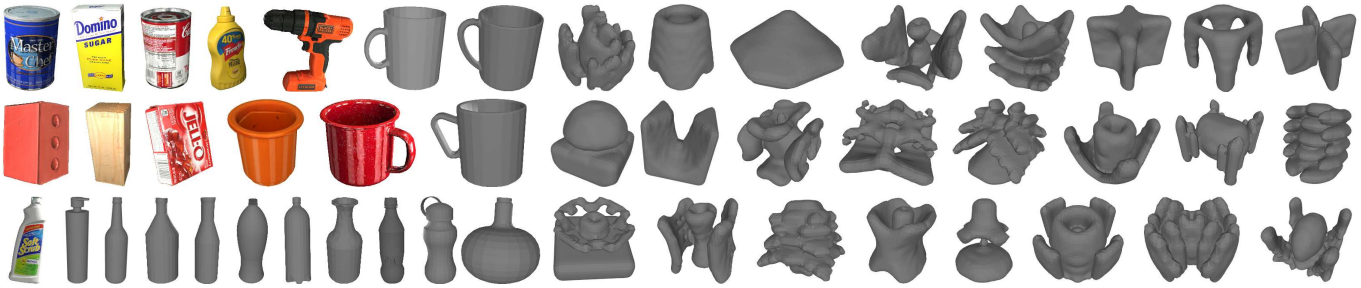


Fig. 4. Objects used to train the RL agent. This includes 11 objects from the YCB object dataset, 14 objects from the ShapeNet object dataset, 5 primitive shapes (4 boxes and 1 cylinder), and 31 (only 24 shown) objects from the EGAD dataset [24].

discounted rewards over a finite trajectory T. The action value function is defined as:

$$\pi^* = \underset{\pi}{\operatorname{argmax}}\, \mathbb{E}_{\tau(\pi)}\left[\sum_{t=0}^{T} \gamma^t r_t\right] \tag{1}$$

where $\gamma \in (0, 1)$ is the discount factor, τ is the trajectory distribution under policy π.

We apply PPO [26], an on-policy gradient algorithm to solve the specified policy optimization problem. Default network parameters are used in our work. To improve the efficiency of the trajectory sampling process, we run 20 simulation environments in parallel, generating 2 million state transitions in 8 hours.

A. Simulation environment

As seen in Fig. 1, the simulation and real robot setup are kept the same. This setup consists of a UR10e robot arm and a Shadow dexterous hand (left hand version).

In the simulation, we use 61 objects for training and testing, including 11 objects from the YCB object set [27], 19 objects from ShapeNet [28], and 31 objects from the EGAD datasets [24], as listed in Fig. 4. Since the object pose is known in simulation, we set the initial grasp pose by adding an offset along the z-axis and a slight disturbance of the x- and y-axis.

B. Observations

Inspired by human hands using multiple modalities to grasp objects, we also introduce tactile sensing, torques and joint angles in the observation space of the Shadow hand agent. To transfer the training model directly from the simulation to a real platform without any further training, observations of the agent should be as similar to a real robot as possible. As accurate contact force values and joint torque values are notoriously hard to get in simulation environments and these continuous values are difficult to map to a real robot, we use the binary contact information of the fingertip denoted as $\phi \in \{0, 1\}$ and level-based joint torque denoted as $\tau \in \{0, ..., 5\}$ in the model to minimize the gap between simulation and real scene. The details of the mapping from raw values to the abstracted values are described in section V-C. The whole observation at $t = t_k$ is defined as $o_k = \langle a_{k-1}, \phi, \tau, \theta \rangle$, where a_{k-1} is the previous target action, τ are the joint torques, and $\theta = \{\theta_f, \theta_w\}$ are finger joint and wrist joint angles of the hand as shown in Fig. 2(a). We find that the historical observations of several previous timesteps are very meaningful during the training process. This may be because these observations sequences can characterize the surface shape of the object, which helps the agent's exploring process. Finally, the input observation values at timestep t can be expressed as stacked vector $O_t = \langle o_{t-(h-1)}, ..., o_{t-1}, o_t \rangle$; h is the timestep length used in the network ($h = 3$ in our work).

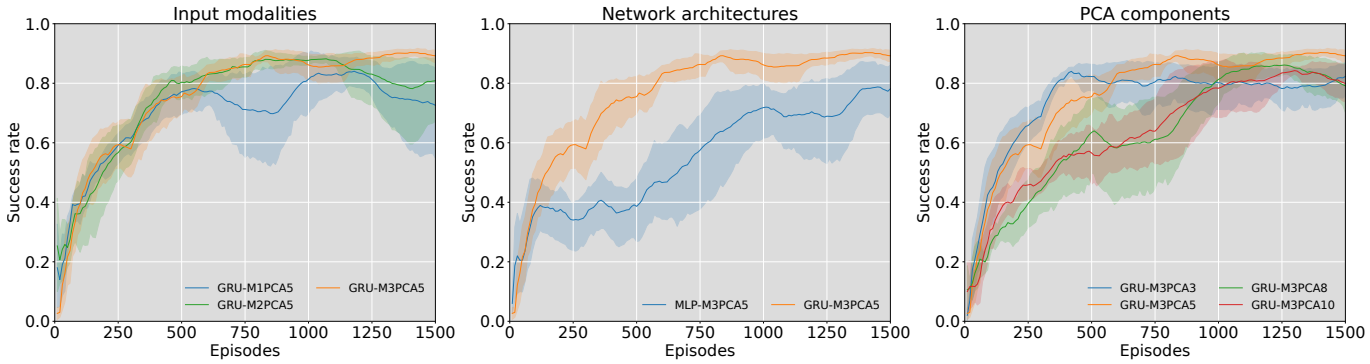


Fig. 5. Network evaluation with different parameters while training our RL agent. The curves are labeled according to the network architecture (recursive GRU or feed-forward MLP), the number of input modalities, and the PCA dimension number. (left) Training result for different input modalities. M1 means the input has only joint angles. M2 means the input modalities are joint angles and fingertip tactile sensing, M3 means the inputs have one more modality: joint torques. (middle) Training result on GRU and MLP network architectures. (right) Training result on different dimension reduction values for the Shadow hand. The reduced dimension values tested here are 3, 5, 8 and 10.

C. Actions

The action generated from the policy comprises two continuous parts. 1) principal component value increments $\Delta\epsilon$, representing the increments of the first n principal components ($n \leq 10$ in our model). 2) wrist joint angle increments $\Delta\theta_w$, and a discrete part 3) lifting decision ρ, which decides whether or not to lift the hand for a pick-up attempt. The combined action output from the policy is represented as $a_t = \{\Delta\epsilon, \Delta\theta_w, \rho\}$.

In our method we do action planning and optimization in the subspace after applying PCA dimension reduction to the 22 finger joints. As is illustrated in Fig. 3, the output action $\Delta\epsilon$ from policy net is an increment based on the current principal components $\epsilon = \mathcal{P}(\theta_f)$, in which $\mathcal{P}$ is the function mapping from joint space to the subspace of planning. Target joint angles of the current timestep can be denoted as:

$$\theta_f^{target} = \mathcal{P}^{-1}(\Delta\epsilon + \mathcal{P}(\theta_f)) \tag{2}$$

Given that the initial palm pose may not always be a proper grasping pose when the objects move because of external interference, we also consider wrist motion. The robot can adjust its wrist joint angles to a better palm pose like humans do while grasping objects. The Shadow hand has two wrist joints; $\Delta\theta_w$ is a 2-dimensional vector representing the rotation angles around the two wrist axes.

The robot's decision to lift its arm for a pick-up or not is also determined by policy output ρ. The robot arm keeps the original pose if $\rho = 0$ else lifts to a fixed height above the object. During lifting all hand joints stay invariant and the episode terminates after the lifting attempt. The log action probability can be denoted as:

$$\begin{aligned} \log\pi(a_t|O_t) = \log\pi(\rho|O_t) + \\ (1-\rho)\big[(\log\pi(\Delta\epsilon|O_t) + \log\pi(\Delta\theta_w|O_t)\big] \end{aligned} \tag{3}$$

To ensure state-space exploration, the PPO learning algorithm used for our agent internally represents the policy as a set of Gaussian functions, whose mean μ and variance σ are updated during learning. The stochastic action output from the policy net $\mathcal{F}(O_t)$ at timestep t is therefore a tuple of two Gaussian samples (namely the change of principal component values for the fingers, $\Delta\epsilon_w \sim \mathcal{N}(\mu_\epsilon(O_t), \sigma_\epsilon(O_t))$, and the update of hand wrist angles $\Delta\theta_w \sim \mathcal{N}(\mu_\theta(O_t), \sigma_\theta(O_t))$) and one discrete value taken from a Bernoulli distribution, $\rho \sim Bern(sigmoid(\beta_\rho(O_t)))$.

D. Reward

To apply RL to a specific robotic task, a carefully designed reward function is usually very useful. In our multifingered grasping task, a training episode is terminated after the lifting attempt and then a binary reward $r_b \in \{0, 1\}$ representing whether the robot picked up the object successfully is returned. Our concrete reward function is defined as:

$$R = \begin{cases} r_b & t = t_{final} \\ 0.03 r_c & t \in [1, t_{final} - 1] \end{cases} \tag{4}$$

The hand closing reward r_c is to guide the agent towards closing the hand. We assume that the combination of positive increments in several key joints: J^* = {FF3, MF3, RF3, LF3, TH5} will lead to a finger closing hand motion. Therefore we denote a mask matrix $M = [m_1, m_2 ... m_n]$, $m_i \in \{0, 1\}$ representing joint J_i in J^* or not.

$$r_c = \sum_{i \in joints} (\theta_f^{target} - \theta_f) \cdot M \tag{5}$$

E. Curriculum Learning

The initial horizontal position and rotation angle of the grasp object are both randomized in a variable range related to the learning process p, known as curriculum learning [29]. The aim is to choose environmental parameters that are neither too challenging nor too trivial for the agents. During the experiment, we found that the initial pose of the object is essential for the hand to generate an effective grasp. As the training process goes on, we increase the task difficulty by adding a randomized disturbance to the initial pose. The new position pos_{start} and yaw orientation rot_{start} of the object at the beginning of each episode changes according to the grasping success rate r_s, original position pos_o, and rot_o:

$$\begin{cases} pos_{start} = pos_o + \delta_{pos} & \delta_{pos} \in [-\delta_p, +\delta_p] \\ rot_{start} = rot_o + \delta_{rot} & \delta_{rot} \in [-\delta_o, +\delta_o] \end{cases} \tag{6}$$

in which $\delta_p = 1.2(r_s + 0.2)$ and $\delta_o = 0.1r_s$ are the variation range of orientation and position, respectively. δ_{pos} and δ_{rot} are both sampled from uniform distribution: $U(-\delta, \delta)$.

V. Experiment

We evaluate our grasping agent based on multimodal reinforcement learning both in simulation and on the real robot hardware. In the simulation, we train our agent with different parameters to choose the best performing model regarding grasp success rate and time to converge during training. For the real robotic experiments, the best-performing agent in the simulation is used to execute the multifingered grasping and is compared with the baseline.

A. Simulation Results

We test our algorithm with different parameters for comparison: 1) Comparing the network performance with different input modalities: using joint angle information only, using joint angle as well as tactile information, and use all the information: joint angle, tactile and joint torques; 2) Comparing networks that use Gated Recurrent Unit (GRU) and Multi-Layer Perception (MLP); 3) Comparing different PCA dimension reduction numbers. We use the below pattern for naming the models. MX_1, where $X_1 \in \{1, 2, 3\}$ means different numbers of input modalities. PCAX_2 where $X_2 \in \{3, 5, 8, 10\}$ means the dimension reduced from the original Shadow hand joint space. All models are trained with three timesteps of history observations as input. The experiment results in Fig. 5 show the grasp success rate of the above three experiments, respectively. Each curve is plotted using five individual runs trained using the same hyperparameters. All models are trained for 1500 episodes. The first experiment (Fig. 5 left) illustrates the grasping performance of different input modalities. The learning curves are similar in the first 500 episodes, after which the learning curve with only joint angles as input (M1) begins to drop, then increases back to 80% and drops again. This is because the task difficulty is changing all the time through curriculum learning (section IV-E). The agent with two modalities as input (M2) exhibits similar instability from episode 1000 for the same reason. As a comparison, the multimodal agent (M3) shows the best robustness to the increasing task difficulty. The second experiment (Fig. 5 middle) demonstrates that GRU outperforms the MLP architecture. The memory mechanism of GRU can understand the historic interaction information better, thus improving the grasping performance. For the last experiment (Fig. 5 right), we test the performance using different PCA dimension reduction values. A latent space with a higher dimension means a bigger action space to explore and more dexterous hand motions to learn. The best performance is from the agent with latent dimension space $X_2 = 5$. The model GRU-M3PCA3 learns faster but the curve stops growing after episode 500 and converges at a lower success rate of 80%. This signifies that the latent space is too small to learn dexterous hand motions to grasp all the objects correctly. When we increase the dimension value to 8 and 10, the agents learn more slowly and have a lower success rate than other models, which indicates that the action space is too large and makes learning how to grasp a challenge for the agent.

An interesting find is that the robot tends to use the little finger and the thumb to form a grasp. We guess the reason is that using these two fingers can make the hand cover more object surfaces due to the mechanical design of the hand, which approximates the human hand but is not perfect.

B. Initial grasp generation for real robot experiments

Since we do not know the exact object poses in the real world experiments, we use our previous work [4] to serve as an initial grasp pose generator. PointNetGPD is a two-fingered grasp evaluation network that takes the partial point cloud near the grasp object as input and outputs grasp quality of the grasp candidate. After motion planning and collision checking using MoveIt [30], the kinematically feasible grasps are chosen as our initial grasp proposals. As these grasps are initially intended for a two-fingered gripper, a proper mapping from a two-fingered gripper to a five-fingered hand is needed, as illustrated in Fig. 6.

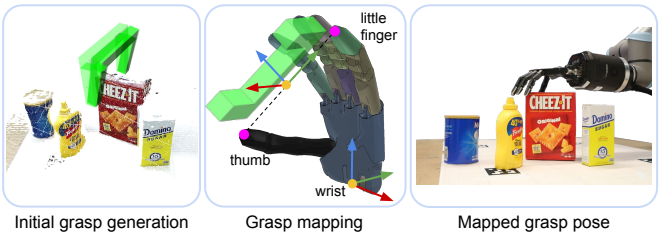


Fig. 6. Grasp mapping method used in this work. (left) The initial grasp generation using PointNetGPD. (middle) Mapping from two-fingered grasp to Shadow hand. (right) Initial grasp pose in the real robot system.

The grasp mapping proceeds as follows:

1) We move all fingers of the Shadow hand to a predefined pose where all the fingers make a "C" shape as shown in Fig. 6 (middle);
2) The grasp location is defined as the middle point between the fingertips of thumb and little finger;
3) The approach direction of the grasp (x-axis) is the palm norm inverse direction;
4) The y-axis of the grasp is chosen by the connection of the fingertips of thumb and little finger;
5) Then the z-axis can be defined as the cross product of the x- and y-axis.

C. Sensor mapping

We need to process the raw data from the tactile and torque sensors to the abstracted sensor data that are used as RL inputs. For the tactile sensor, in the simulator, we detect whether there is a contact between fingertip and object to determine the mapping values. On the real robot, we press each fingertip sensor manually and record the raw reading to get the upper sensor range of each finger. The lower range of the tactile sensors is calculated by keeping the hand still, reading the sensor raw value ten times, and getting an average. Then we mark a tactile observation as 1 if the sensor reading is higher

than a threshold value of 0.3% of the total range, otherwise as 0. For torques, in the simulator we directly map the measured joint torques to the discrete levels. However, in the real robot, as the Shadow hand does not provide torque reading out of the box, we use the measured tendon force of each motor instead. We also found that the tendon reading τ_j will drift after some time. Therefore, we take the tendon reading when the hand is empty at the beginning of each grasp attempt and set it as the initial tendon value $\tau_{j,0}$ for each joint. For mapping the torque data, we use empirical thresholds $\in \{-200, -100, 0, 100, 200\}$ for the real robot and $\{-20, -10, 0, 10, 20\}$ for the simulator to map the reading $\tau_j - \tau_{j,0}$ to the interval $[0, 5]$ expected by the RL agent.

D. Real Robot Verification

For the real robot experiments, we selected 14 objects including some objects used during training (object number 1 to 9) and novel objects (object number 10 to 14), as can be seen in Fig. 7. A Kinect2 depth camera is used to get the object point cloud required for initial grasp pose generation in the actual robot experiments. For each object, we conduct ten grasp trials for each of the below four different agents. The first two agents are our baseline1 and baseline2, respectively, which use a hard-coded sequence to close the fingers for all the grasp trials. In baseline1, we set a torque limit for each joint and control the hand in position mode. All active joints are controlled to track the given trajectory until reaching the target positions or the tendon force limit. In baseline2, besides joint position and joint torque sensing, we add one more modality: tactile sensing. The first finger, middle finger, and ring finger will stop closing if the tactile sensor on the fingertip is triggered, which helps to prevent from over pushing the object beyond a proper grasping position. The third agent uses the GRU-M2PCA5 model. The fourth agent uses the GRU-M3PCA5 model.

To perform a fair comparison of the experiments to demonstrate the difference between agents when the grasp pose is the same, we first mark the target object in a fixed position on the table. Then we use a point cloud and PointNetGPD to generate ten initial grasp poses that are limited to top-down grasps. After this, we perform the finger motion with the above four different agents. The grasp success rates in Table I indicate that the RL method can outperform the baseline method in most objects, which establishes that the agent trained using RL has learned a robust grasping strategy. The RL agent trained using three modalities performs better than the RL agent using two modalities. We also see that the success rates for objects 4 and 5 are quite low. This is because these objects are smaller on the top, which needs more precise grasping actions. From the experiments we find out that the average episode length in simulation and real experiments is 5.8 and 6.2 for model GRU-M3PCA5. The action frequencies of the models are 1.8 Hz and 1.6 Hz in simulation and the real world. Besides the random grasp experiments, we also conduct a grasp experiment where we fix our object pose and initial grasp pose for all three agent conditions to show the difference between agents when the grasp pose is the same. The result is presented in the form of a supplemented video to demonstrate the motion difference clearly.

VI. Conclusion and Future Work

This paper proposes a novel hand-arm multifingered grasping system to solve the autonomous multifingered grasping problem. We first build a hand pose dataset to teach the dexterous Shadow hand how humans commonly move their hand during grasping by mapping human motions to the Shadow hand. A PCA-based hand synergy is then trained to reduce the dimension used to control the hand, which accelerates the training speed of a grasping agent. Then we build a simulation environment to train the RL agent. Detailed simulation trials with different parameters demonstrate that our agent works best with three modalities as input and a GRU network architecture. Real robot experiments show that the trained RL agent can be applied in the real world even if the model is trained purely in simulation. Our method outperforms the baseline method.

Training the robot to do object in-hand manipulation tasks such as tool use is our on-going work. Domain randomization plays an important role to bridge the gap between simulation and the real world. The dynamic parameters such as friction, object mass, and even moment inertia will be randomized in a reasonable range. We also plan to accelerate the simulation time with a differentiable physical simulator. Besides, adding vision to the agent to let the agent find the initial grasp by itself is also an exciting research direction.

References

[1] R. R. Ma, L. U. Odhner, and A. M. Dollar, "Dexterous manipulation with underactuated fingers: Flip-and-pinch task," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 3551–3552.

[2] L. Ke, J. Wang, T. Bhattacharjee, B. Boots, and S. Srinivasa, "Grasping with chopsticks: Combating covariate shift in model-free imitation learning for fine manipulation," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[3] O. M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, *et al.*, "Learning dexterous in-hand manipulation," *The International Journal of Robotics Research*, vol. 39, no. 1, pp. 3–20, 2020.

[4] H. Liang, X. Ma, S. Li, M. Görner, S. Tang, B. Fang, F. Sun, and J. Zhang, "PointNetGPD: Detecting grasp configurations from point sets," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019, pp. 3629–3635.

[5] M. Santello, M. Flanders, and J. F. Soechting, "Postural hand synergies for tool use," *Journal of neuroscience*, vol. 18, no. 23, pp. 10 105–10 115, 1998.

[6] M. A. Roa, M. J. Argus, D. Leidner, C. Borst, and G. Hirzinger, "Power grasp planning for anthropomorphic robot hands," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 563–569.

[7] M. Li, K. Hang, D. Kragic, and A. Billard, "Dexterous grasping under shape uncertainty," *Robotics and Autonomous Systems*, vol. 75, pp. 352–364, 2016.

[8] Y. Fan, T. Tang, H.-C. Lin, and M. Tomizuka, "Real-time grasp planning for multi-fingered hands by finger splitting," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 4045–4052.

[9] L. Shao, F. Ferreira, M. Jorda, V. Nambiar, J. Luo, E. Solowjow, J. A. Ojea, O. Khatib, and J. Bohg, "Unigrasp: Learning a unified model to grasp with multifingered robotic hands," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2286–2293, 2020.

[10] S. Brahmbhatt, A. Handa, J. Hays, and D. Fox, "Contactgrasp: Functional multi-finger grasp synthesis from contact," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 2386–2393.

TABLE I
ROBOT EXPERIMENT RESULT

Object ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Baseline 1	80%	70%	40%	40%	20%	20%	50%	50%	80%	60%	60%	100%	50%	70%
Baseline 2	70%	80%	40%	40%	30%	20%	40%	60%	80%	70%	60%	100%	60%	**80%**
GRU-M2PCA5	60%	90%	**80%**	**60%**	**50%**	**80%**	60%	**70%**	90%	**100%**	90%	100%	90%	**80%**
GRU-M3PCA5	**100%**	**100%**	**80%**	50%	40%	**80%**	**70%**	**70%**	**100%**	**100%**	**100%**	100%	**100%**	70%

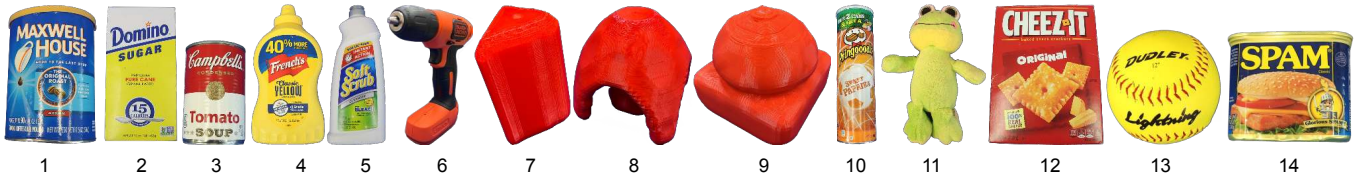


Fig. 7. Objects used in the real robot experiments. Objects with IDs from 1 to 9 are in the training dataset, and objects with IDs from 10 to 14 are unseen by the multifingered grasping agent. Objects 7, 8, and 9 are 3D printed models from EGAD.

Fig. 8. Grasp examples for each object. The agent used in these grasps is GRU-M3PCA5.

[11] M. Ciocarlie, C. Goldfeder, and P. Allen, "Dimensionality reduction for hand-independent dexterous robotic grasping," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2007, pp. 3270–3275.

[12] V. Kumar, T. Hermans, D. Fox, S. Birchfield, and J. Tremblay, "Contextual reinforcement learning of visuo-tactile multi-fingered grasping policies," in *NeurIPS Workshop on Robot Learning: Control and Interaction in the Real World*, 2019.

[13] F. Ficuciello, D. Zaccara, and B. Siciliano, "Synergy-based policy improvement with path integrals for anthropomorphic hands," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 1940–1945.

[14] F. Ficuciello, A. Migliozzi, G. Laudante, P. Falco, and B. Siciliano, "Vision-based grasp learning of an anthropomorphic hand-arm system in a synergy-based control framework," *Science Robotics*, vol. 4, no. 26, 2019.

[15] B. Wu, I. Akinola, J. Varley, and P. Allen, "Mat: Multi-fingered adaptive tactile grasping via deep reinforcement learning," in *3rd Conference on Robot Learning (CoRL)*, 2019.

[16] Y. Chebotar, K. Hausman, Z. Su, G. S. Sukhatme, and S. Schaal, "Self-supervised regrasping using spatio-temporal tactile features and reinforcement learning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 1960–1966.

[17] H. Merzić, M. Bogdanović, D. Kappler, L. Righetti, and J. Bohg, "Leveraging contact forces for learning to grasp," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019, pp. 3615–3621.

[18] M. T. Ciocarlie and P. K. Allen, "Hand posture subspaces for dexterous robotic grasping," *The International Journal of Robotics Research*, vol. 28, no. 7, pp. 851–867, 2009.

[19] A. Bernardino, M. Henriques, N. Hendrich, and J. Zhang, "Precision grasp synergies for dexterous robotic hands," in *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2013, pp. 62–67.

[20] Z. Deng, B. Fang, B. He, and J. Zhang, "An adaptive planning framework for dexterous robotic grasping with grasp type detection," *Robotics and Autonomous Systems*, vol. 140, p. 103727, 2021.

[21] T. Feix, J. Romero, H. B. Schmiedmayer, A. M. Dollar, and D. Kragic, "The GRASP taxonomy of human grasp types," *IEEE Transactions on Human-Machine Systems*, pp. 66–77, 2016.

[22] E. Corona, A. Pumarola, G. Alenya, F. Moreno-Noguer, and G. Rogez, "GanHand: Predicting human grasp affordances in multi-object scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 5031–5041.

[23] P. Ruppel, N. Hendrich, S. Starke, and J. Zhang, "Cost functions to specify full-body motion and multi-goal manipulation tasks," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 3152–3159.

[24] D. Morrison, P. Corke, and J. Leitner, "EGAD! an evolved grasping analysis dataset for diversity and reproducibility in robotic manipulation," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4368–4375, 2020.

[25] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesis—a survey," *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 289–309, 2013.

[26] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[27] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, "The ycb object and model set: Towards common benchmarks for manipulation research," in *IEEE International Conference on Advanced Robotics (ICAR)*, 2015.

[28] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, *et al.*, "ShapeNet: An information-rich 3D model repository," *arXiv preprint arXiv:1512.03012*, 2015.

[29] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Science robotics*, vol. 5, no. 47, 2020.

[30] D. Coleman, S. Chitta, and N. Correll, "Reducing the barrier to entry of complex robotic software: a MoveIt! case study," *Journal of Software Engineering for Robotics*, pp. 3–16, 2014.