IEEE 19th International Conference on Automation Science and Engineering (CASE), Auckland, New Zealand, August 2023.

Deep Reinforcement Learning of Dexterous Pre-grasp Manipulation for Human-like Functional Categorical Grasping

Dmytro Pavlichenko and Sven Behnke

***Abstract*— Many objects such as tools and household items can be used only if grasped in a very specific way—grasped functionally. Often, a direct functional grasp is not possible, though. We propose a method for learning a dexterous pre-grasp manipulation policy to achieve human-like functional grasps using deep reinforcement learning. We introduce a dense multi-component reward function that enables learning a single policy, capable of dexterous pre-grasp manipulation of novel instances of several known object categories with an anthropomorphic hand. The policy is learned purely by means of reinforcement learning from scratch, without any expert demonstrations, and implicitly learns to reposition and reorient objects of complex shapes to achieve given functional grasps. Learning is done on a single GPU in less than three hours.**

I. Introduction

Grasping is a fundamental skill that manipulation robots need for interacting with their environment. Many objects are made for human hands and require a specific grasp for use. For example, a drill requires a power grasp with the index finger on the trigger. We refer to such grasps as functional. Often, a functional grasp cannot be achieved directly because the object is in the wrong pose. This can be addressed with pre-grasp manipulation: repositioning and reorienting the object until the desired functional grasp is achieved. However, robustly performing interactive functional grasping with a dexterous multi-finger hand is challenging. Solving this problem will allow robots to use tools and functional objects designed for humans.

Inspired by our previous work on functional re-grasping [1], in this paper we propose a methodology that replaces several complex classical components with a single data-driven approach. Deep Reinforcement Learning (DRL) has been applied to several complex dynamic robotic domains [2], [3], [4], [5]. In this work, we use a highly efficient GPU-based simulation [6] together with DRL to learn a policy for dexterous pre-grasp manipulation. Many approaches focus on learning the policies directly from low-level sensory inputs, such as camera images and point clouds [7], [8]. However, we argue that most of the data points in these inputs, such as background pixels in an image, are irrelevant to the manipulation policy most of the time. Therefore, we assume that perception is performed by an external method, and our approach is provided with high-level semantic information about the scene. This speeds up the learning process, since the policy can be represented by a model with fewer parameters, and the simulation does not need to perform expensive image rendering. By considering multiple object instances within the same category, we further reduce the inputs to the policy, as category-specific features of object geometry and dynamics are implicitly learned. Finally, we eliminate the need for expert demonstrations by introducing a dense multi-component reward function. This reward function naturally encourages the use of a dexterous anthropomorphic hand for object manipulation.

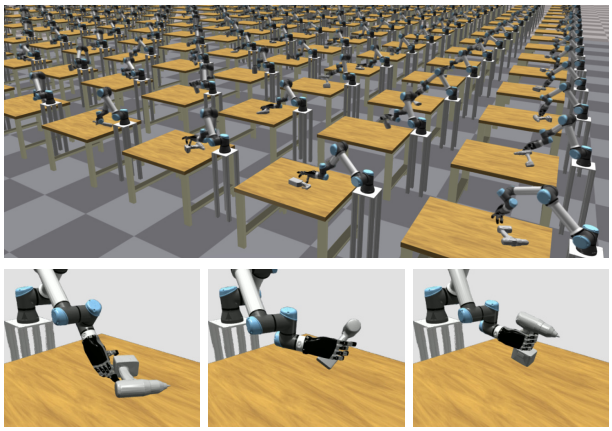

Fig. 1: Top: Learning human-like functional grasping in multiple parallel environments. Bottom: The learned policy performs pre-grasp manipulation of a novel object instance of a known category.

To evaluate the proposed method, we learn a single policy in simulation on three conceptually distinct rigid object categories: drills, spray bottles and mugs. Using dense multi-component reward, the policy learns to perform dexterous pre-grasp manipulation on previously unseen object instances of known categories (Fig. 1) with a high success rate. The learning is performed in simulation in less than three hours on a single GPU. The main contributions of this work are:

- a multi-component dense reward formulation that quickly yields policies capable of dexterous pre-grasp manipulation of novel objects using a multi-finger hand,
- a high-level state formulation and generic two-stage curriculum that facilitates implicit category-specific object geometry learning, and
- a functional grasping 3D mesh dataset with three object categories.

II. Related Work

Dexterous pre-grasp manipulation has been an active area of research for decades. Multiple classical model-based

This work was funded by the German Ministry of Education and Research (BMBF), grant no. 01IS21080, project "Learn2Grasp: Learning Human-like Interactive Grasping based on Visual and Haptic Feedback".

Both authors are with the Autonomous Intelligent Systems (AIS) Group, Computer Science Institute VI, University of Bonn, Germany. pavlichenko@ais.uni-bonn.de

approaches have been proposed [9], [10], [11], [12], [13]. They work for known objects with exact models, but require carefully hand-crafted task-dependent algorithms and suffer from uncertainty inherent in dexterous and highly dynamic manipulation.

Specifically, in our previous work [1], we address functional grasping of novel object instances of known categories by means of re-grasping with a dual-arm robot. Although showing good results, the approach consists of several highly complex components and lacks an ability to swiftly react to unforeseen changes of the object pose.

To alleviate these drawbacks, data-driven approaches have been proposed. In particular, Deep Reinforcement Learning (DRL) and Imitation Learning (IL) using Artificial Neural Networks (ANN) to represent policies for dexterous manipulation have gained much popularity in recent years [14], [15], [16]. By learning purely from observed experiences and/or provided demonstrations, these methods yield highly reactive policies capable of dexterous multi-finger manipulation.

Zhou et al. [17] address pre-grasp manipulation of objects in ungraspable configurations through extrinsic dexterity. Their method uses model-free RL to learn pushing objects against a wall in order to achieve a graspable pose. The method uses minimalistic object representation, similar to our approach. However, it has difficulties generalizing to objects with complex non-convex shapes. In our approach this issue is resolved by learned implicit category-specific geometry knowledge. Similarly, Sun et al. [18] use model-free RL to obtain a policy for a dual-arm robot which pushes an object next to a wall and turns it in order to grasp it with the other hand. Both works use parallel grippers, which make the manipulation less dexterous.

Yuzhe et al. [8] train a dexterous manipulation policy for an Allegro hand to grasp novel objects of a known category. Their approach uses point clouds as input to provide information about object geometry. The difference to our work is that we specifically address functional grasping, while in their work the grasps are arbitrary. Mandikal et al. [7] propose to learn a policy with object-centric affordances to dexterously grasp objects. Notably, the policy is learned with a prior derived from observing manipulation videos, which requires tedious annotation of human grasp regions in the observed images.

A wide range of works are based on real-world expert demonstrations [19], [20], [21]. These approaches have to deal with challenges of mapping human motion to the kinematics of the robot arm and hand. Hence, the direct applicability to different robotic setups is not straightforward. Chen et al. [22] address this issue by bootstrapping a small dataset of human demonstrations with a larger dataset including novel objects and grasps. The objects are deformed and dynamically consistent grasps are generated. The policy is then trained in a supervised manner in simulation, followed by a direct transfer to the real world. This method struggles with objects of complex shapes. In contrast, in our work we avoid using explicit demonstrations and instead rely on a general and dense reward function to guide the policy towards dexterous manipulation.

A completely different approach was proposed by Dasari et al. [23]: a combination of trajectory-centric formulation with a pre-grasp based exploration primitive. The pre-grasp based approaches were also introduced in [24], [25], [26] and our approach shares the high-level idea with them. The policy learns to perform a wide variety of tasks in simulation without any per-task engineering. The key difference to our work is that learned behaviors directly depend on supplied exemplar trajectories. In addition, the manipulation is performed by a freely floating hand, which relaxes multiple constraints introduced by the kinematics of the robotic arm in combination with object poses on the edge of the workspace.

To the best of our knowledge, there are no recent works which address the problem of categorical pre-grasp manipulation of novel objects in a context of functional grasping.

III. Method

The objective of this work is to learn a policy π which achieves the desired behavior: pre-grasp manipulation of novel object instances with aim of reaching a functional grasp. The policy π_θ is represented by a deep neural network and is parameterized by weights θ, learned with DRL. The problem is modeled as a Markov Decision Process (MDP): $\{S, A, P, r\}$ with state space $S \in \mathbb{R}^n$, action space $A \in \mathbb{R}^m$, state transition function $P\colon S \times A \mapsto S$, and reward function $r\colon S \times A \mapsto \mathbb{R}$. Since the problem has continuous state and action space, the policy $\pi_\theta(\boldsymbol{a}|\boldsymbol{s})$ represents an action probability distribution when observing a state $\boldsymbol{s}(t)$ at timestep t.
The objective of DRL is to maximize the expected reward:

$$J(\pi_\theta) = \sum_{t=0}^{T} \mathbb{E}[\gamma^t r(\boldsymbol{s}(t), \boldsymbol{a}(t))], \tag{1}$$

where $\gamma \in [0, 1]$ is the discounting factor.

The policy is provided with a target functional pre-grasp to reach, defined as a 6D hand pose in object frame plus hand joint positions. This pre-grasp is very close to the desired functional grasp, so that closing the hand will guarantee a successful grasp. The advantage of a pre-grasp is that it can be reached more freely and slight inaccuracies in its definition are negligible, as discussed by Dasari et al. [23]. We assume that there is a single object in front of the robot hand on a flat surface.

A. Action Space

The policy produces actions $\boldsymbol{a}(t)$ with a frequency of 30 Hz. An action represents a relative displacement in 3D hand position, hand rotation changes, and hand joint position increments. With this action definition, hand joint targets are straightforward to obtain. The arm joint targets are calculated via Inverse Kinematics (IK). Finally, the joints are controlled with PD controllers. In this work we apply the proposed method to a 6 DoF UR5e robotic arm with an attached 11 DoF Schunk SIH hand. The joints of the hand are coupled, leaving five controllable DoF. Thus, in this work an action is a 11-element vector: three elements define a displacement

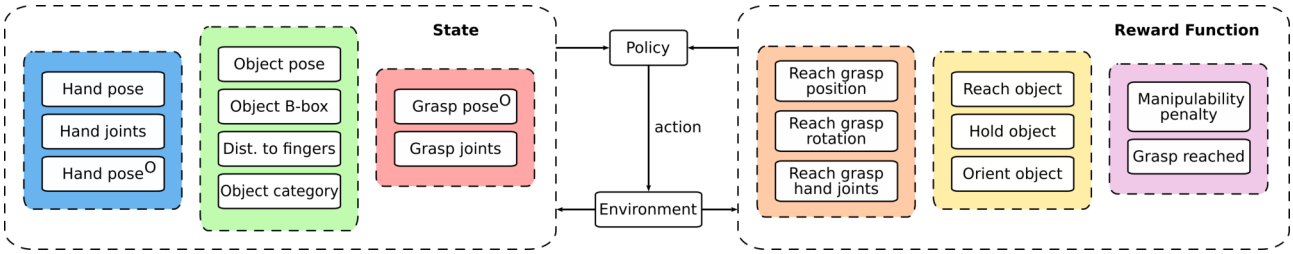

Fig. 2: Composition of the state representation and the reward function. The state consists of information about the hand, the object, and the target functional grasp. "O" denotes object frame of reference. The reward function consists of a term encouraging reaching the target grasp, a term encouraging pre-grasp manipulation, and a low manipulability score penalty.

of hand position, three elements define a displacement of hand rotation as Euler angles and five elements define a displacement of hand joint positions. We further assume a five-fingered hand with five controllable DoF, however it is straightforward to apply our approach to a hand with arbitrary number of fingers and DoF.

B. State Space

The left part of Fig. 2 illustrates the vector $\boldsymbol{s}(t)$ representing the state, which consists of three distinctive parts: information about the hand $\boldsymbol{h}$, information about the object $\boldsymbol{o}$, and information about the target functional grasp $\boldsymbol{g}$:

$$\boldsymbol{s} = [\boldsymbol{h}, \boldsymbol{o}, \boldsymbol{g}]. \tag{2}$$

Information about the hand is a column vector:

$$\boldsymbol{h} = [\boldsymbol{h}_p, \boldsymbol{h}_r, \boldsymbol{h}_j, \boldsymbol{h}_p^O, \boldsymbol{h}_r^O], \tag{3}$$

where $\boldsymbol{h}_p$ is a 3D hand position vector, $\boldsymbol{h}_r$ is a 4-element hand rotation vector represented by a quaternion, and $\boldsymbol{h}_j$ is a 5-element hand joint position vector; $\boldsymbol{h}_p^O$ and $\boldsymbol{h}_r^O$ are hand position and rotation in object frame of reference O. Thus, information about the hand $\boldsymbol{h}$ is a 19-element vector.
Information about the object is a column vector:

$$\boldsymbol{o} = [\boldsymbol{o}_p, \boldsymbol{o}_r, \boldsymbol{o}_{bb}, \boldsymbol{o}_s, \boldsymbol{o}_c], \tag{4}$$

where $\boldsymbol{o}_p$ is 3D object position, $\boldsymbol{o}_r$ is a 4-element object rotation vector represented by a quaternion, $\boldsymbol{o}_{bb}$ is a 6-element vector representing object bounding box by two 3D positions of diagonally opposing bounding box corners, $\boldsymbol{o}_s$ is a 10-element vector of signed distances between fingertips and middles of the fingers to the object surface, $\boldsymbol{o}_c$ is a C-element one-hot vector representing object category. Distances from fingers to object surface are efficiently calculated from a precomputed object Signed Distance Field (SDF). We adopted this approach from [27]. Thus, information about the object is a (23+C)-element vector. The representation is compact; however, the general geometric features of the object categories are learned implicitly from the experience. The desired functional grasp is provided as column vector:

$$\boldsymbol{g} = [\boldsymbol{g}_p^O, \boldsymbol{g}_r^O, \boldsymbol{g}_j], \tag{5}$$

where $\boldsymbol{g}_p^O$ is 3D hand position in object frame of reference, $\boldsymbol{g}_r^O$ is a 4-element hand rotation vector represented by a quaternion, and $\boldsymbol{g}_j$ is a 5-element hand joint position vector. The target functional grasp is represented by a 12-element vector. In practice, the functional grasps can be provided by methods such as [28], [29].

Overall, for $C = 3$ the state is a 57-element vector. It resembles a high-level semantic representation of the scene. This compact state can be computed fast on a GPU and thus facilitates quick learning. Moreover, compared to DRL models which learn directly from raw visual inputs, smaller models with fewer parameters can be used.

C. Reward Function

The right part of Fig. 2 illustrates the composition of the reward function $r(t)$ that is defined as:

$$r(t) = r_{\text{grasp}}(t) + r_{\text{man}}(t) + r_{\text{MP}}(t) + r_{\text{T}}(t), \tag{6}$$

where r_{grasp} encourages movement towards target grasp $\boldsymbol{g}$, r_{man} encourages pre-grasp manipulation of an object, r_{MP} penalizes being in configurations with low manipulability, and r_{T} rewards reaching the target functional grasp $\boldsymbol{g}$. Each reward component is defined to be in $[-1, 1]$ and described in detail below. For brevity, we omit specifying dependency on time t, unless necessary.

First, we define the distance function ϕ between two quaternions $\boldsymbol{q}$ and $\boldsymbol{q}'$ as the rotation between them:

$$\phi(\boldsymbol{q}, \boldsymbol{q}') = 2 \arccos((\boldsymbol{q} \cdot \boldsymbol{q}'^{-1})_4). \tag{7}$$

The grasp reward r_{grasp} is defined as:

$$r_{\text{grasp}} = r_{h_p} + r_{h_r} + \lambda r_{h_j}, \tag{8}$$

where r_{h_p} encourages moving the hand position towards the target 3D grasp position, r_{h_r} encourages moving the hand rotation towards the target grasp rotation, and r_{h_j} encourages moving hand joints positions towards target grasp joint positions. $\lambda \in [0, 1]$ is the grasp joint reward importance factor. Overall, the r_{grasp} reward encourages to align hand pose and joint positions with the target grasp pose and joint positions. The hand position reward r_{h_p} is defined as:

$$r_{h_p}(t) = \frac{\Delta h_p(t-1) - \Delta h_p(t)}{\Delta h_p^{\max}}, \quad \Delta h_p = ||\boldsymbol{h}_p^O - \boldsymbol{g}_p^O||, \tag{9}$$

where Δh_p is the Euclidean distance from the hand position $\boldsymbol{h}_p^O$ to the target grasp hand position $\boldsymbol{g}_p^O$. $\Delta h_p^{\max}$ is a maximal hand position change during the step duration Δt: $\Delta h_p^{\max} = v_{h_p}^{\max} \Delta t$ with $v_{h_p}^{\max}$ being the maximal linear velocity of the hand. In case of the UR5e hand in this work $v_{h_p}^{\max} = 1\,\text{m/s}$ and $\Delta t = 0.0333\,\text{s}$.

The hand rotation reward is defined as:

$$r_{h_r}(t) = \frac{\Delta h_r(t-1) - \Delta h_r(t)}{\Delta h_r^{\max}}, \quad \Delta h_r = \phi(\boldsymbol{h}_r^O, \boldsymbol{g}_r^O), \quad (10)$$

where Δh_r is a distance from the hand rotation $\boldsymbol{h}_r^O$ to the target grasp hand rotation $\boldsymbol{g}_r^O$, calculated according to Eq. 7. $\Delta h_r^{\max}$ is a maximal hand rotation change during time Δt. It is defined analogously to $\Delta h_p^{\max}$. We use $v_{h_r}^{\max} = \pi$ rad/s. Finally, the hand joint reward is defined as:

$$r_{h_j}(t) = \frac{\Delta h_j(t-1) - \Delta h_j(t)}{\Delta h_j^{\max}}, \Delta h_j = \frac{1}{N}\sum_{i=0}^{N} |h_{j_i} - g_{j_i}|, \quad (11)$$

where N is the number of controllable hand joints, Δh_j is an average per-joint distance to the target grasp joint positions, and $\Delta h_j^{\max}$ is a maximal joint position displacement during time Δt. It is defined similarly to the maximal position and rotation displacements, through maximal joint velocity. We use $v_{h_j}^{\max} = \pi$ rad/s.
The hand joint importance factor λ is defined as:

$$\lambda = \left(1 - \frac{\min(h_p^{\text{prox}}, \Delta h_p)}{h_p^{\text{prox}}}\right)\left(1 - \frac{\min(h_r^{\text{prox}}, \Delta h_r)}{h_r^{\text{prox}}}\right), \quad (12)$$

where h_p^{prox} is a predefined constant, representing a proximity distance between the hand position and the target grasp position, from which the hand joint position reward becomes active. We set it to the length of the hand. Similarly, h_r^{prox} is a rotation proximity distance; we use $h_r^{\text{prox}} = 1$ rad. Overall, λ leads to ignoring the hand joint reward when the hand is far from the target grasp pose, and to use of fingers for manipulation rather than pursuing yet distant target positions.
The manipulation reward r_{man} is defined as:

$$r_{\text{man}} = r_{\text{reach}} + r_{\text{hold}} + r_{\text{orient}}, \quad (13)$$

where r_{reach} encourages moving the hand towards the object, r_{hold} encourages holding the object in the hand, and r_{orient} encourages orienting the object towards a nominal rotation, where the target grasp is more likely to be reachable. Thus, the manipulation reward term encourages a canonical *reach* $\rightarrow$ *hold* $\rightarrow$ *orient* behavior for pre-grasp object manipulation. In this reward, all terms are only positive.
The hand reach reward is defined as:

$$r_{\text{reach}}(t) = \frac{\sum_{k=1}^{K}\left(d(H_{p_k}(t-1)) - d(H_{p_k}(t))\right)}{\Delta h_p^{\max}}, \quad (14)$$

where d is a function, taking a set of 3D points and returning signed distances from the points to the object surface, utilizing the precomputed object SDF. $\Delta h_p^{\max}$ is a maximal position displacement, defined in Eq. 9. H_p is a set of K points positioned between the thumb and the other fingers, described in detail below. In the context of this reward, these points guide the hand towards a position where the object is between the thumb and the other fingers, which is advantageous for manipulation.
The object hold reward is defined as:

$$r_{\text{hold}} = \frac{1}{K}\sum_{k=1}^{K} \frac{d(H_{p_k}) - \rho}{d_k^{\max}}, \quad (15)$$

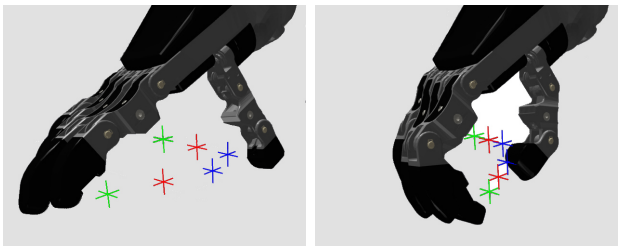

Fig. 3: Equidistant points between thumb tip and middle finger tip & center, used to query distances to the object. Red: central point, green and blue: distal points. When the hand closes, the points move closer together, yielding higher reward in case of a grasped object.

where ρ is a predefined constant radius of spheres with points H_p as centers and $d_k^{\max}$ is a per-point maximum possible distance from the point to the closest finger surface. The set of hold-detect points H_p is positioned between the tip of the thumb and between the tip and middle of the other fingers. Thus, points between fingertips represent positions where objects can be pinch-grasped; and points between the thumb tip and middles of the fingers represent positions where objects are grasped more securely. Each direction tip-tip or tip-middle of a finger has three equidistant points. This ensures a positive response when an object is positioned between the thumb and other fingers imperfectly. When the hand closes, the equidistant points come closer to each other, which promotes closing the hand around an object. Note, that the maximum r_{hold} is achieved when fingers evenly embrace the object, which naturally resembles a grasp. For simplicity, we use only the thumb to middle finger line in this work, which yields six points (Fig. 3). In practice, we observed that this is sufficient to learn a grasping behavior.
The object orient reward is defined as:

$$r_{\text{orient}}(t) = \frac{\Delta o_r(t-1) - \Delta o_r(t)}{\pi}, \Delta o_r = \phi(\boldsymbol{o}_r, \boldsymbol{o}_r^{\text{nominal}}), \quad (16)$$

where Δo_r is a distance from the object rotation to the nominal object rotation $\boldsymbol{o}_r^{\text{nominal}}$. A nominal rotation resembles a natural object orientation as intended for functional use: the object z-axis points upwards and the object x-axis (the direction of the tool tip) points away from the hand. Although there are many other feasible object orientations to perform a functional grasp, we find that such definition is generic and unbiased. In practice, it provides good guidance on how to reorient an object when it is in a state where a direct functional grasp is not possible.
The manipulability penalty reward is defined as:

$$r_{\text{MP}} = 1 - 2\Big/\left(1 + \left(\frac{\min(|J|, |J|_{\max})}{|J|_{\max}}\right)^3\right), \quad (17)$$

where $|J|$ is a determinant of the end-effector Jacobian J and $|J|_{\max}$ is a maximum determinant value which is penalized. We define $|J|_{\max}$ to be 15% of maximal observed $|J|$ for a specific arm. This reward penalizes coming close to singularities and leads to learning more intuitive motions.

Finally, the target grasp reward is defined as:

$$r_{\mathrm{T}} = \begin{cases} 1 & \text{if } \Delta h_p < T_p \wedge \Delta h_r < T_r \wedge \Delta \boldsymbol{h}_j < T_j \\ 0 & \text{otherwise,} \end{cases} \tag{18}$$

where T_p, T_r, T_j are the distances thresholds for hand position, rotation, and hand joint positions to the target grasp that define the accuracy with which the target grasp is achieved. We use $T_p = 1\,\mathrm{cm}$, $T_p = 0.15\,\mathrm{rad}$, and $T_p = 0.1\,\mathrm{rad}$. The episode ends when reaching the target grasp.

A wide use of differential distances in our reward instead of directly using the velocities naturally avoids learning overshooting behaviors. Note that all reward terms are defined in a generic way, and can be easily configured for an arbitrary robotic arm and hand. All reward components are defined to be in the interval $[-1, 1]$. This allows to apply relative scaling easily. For best performance, we scale the rewards proportionally to frequency of their achievement: $r_{\mathrm{T}} \gg r_{\mathrm{orient}} \gg r_{\mathrm{hold}} \gg r_{\mathrm{reach}}$. We leave the other rewards unscaled. This reduces the probability that the policy gets stuck in local minima created by accumulating rewards granted for actions which can be achieved easier than the final goal.

To summarize, the multi-component reward function can be split into three terms:

1) the manipulability penalty reward r_{MP} penalizes being close to singularities and thus helps to avoid unintuitive behavior,
2) the grasp reward r_{grasp} encourages reaching the given functional grasp, and
3) the manipulation reward r_{man} encourages to reach, hold, and reorient the object.

Each component is a continuous dense reward. The components combine to effectively guide the policy towards learning a robust dexterous pre-grasp manipulation. Finally, a sparse component r_T rewards reaching the target grasp.

D. Curriculum

In this work, we avoid having any explicit expert demonstrations and focus on learning robust and natural policies for object pre-grasp manipulation through pure DRL with dense reward shaping. To facilitate faster and more stable learning, we propose a simple two-stage curriculum. In the first stage, we place the objects in poses where target functional grasps can be reached directly. The second stage then has full difficulty, taking advantage of the warm-start provided by the first stage. During the first stage, the objects are put on the table in their nominal poses 5 cm away from the inner side of the hand. The arm is set to a neutral configuration with high manipulability score. We disable the r_{man} reward term during the first stage, so that the policy can converge faster without being stuck in potential multiple local minima. Note, that this curriculum is agnostic to object-specific details. Thus, we keep the approach general while achieving faster policy convergence.

IV. Evaluation

To evaluate the proposed approach, we apply it to the 6 DoF UR5e robotic arm with attached 11 DoF Schunk SIH hand. The joints of this wire-driven hand are coupled, leaving 5 controllable DoF. With this evaluation we try to answer the following questions:

- Does our approach reliably produce robust manipulation policies, capable of dexterous pre-grasp manipulation of unseen object instances of a known category?
- Does the multi-component manipulation reward r_{man} lead to policies with higher success rates?
- Does the curriculum improve convergence stability?

A. Setup

We use Proximal Policy Optimization (PPO) [30] to train the policies. We employ the RL Games [31] high-performance implementation for GPU parallelization. We use findings of Mosbach et al. [27] as a base, keeping the learning algorithm hyper-parameters the same. The policy is represented by a three-layer fully-connected neural network. In our case, the input is a 57-element vector. The network is a multilayer perceptron and has the following structure:

$$57 \times 512 \rightarrow 512 \times 256 \rightarrow 256 \times 128 \rightarrow 128 \times 11.$$

In our experiments, we pursue the objective of learning a single functional grasping policy for three rigid objects categories: drills, spray bottles and mugs. To this end, we prepared a 3D mesh dataset of 39 objects: 13 of each category, where ten objects are for training and the remaining three objects are used for testing. The dataset was composed of meshes from [32] and of meshes available online[1]. We make the dataset available online[2].

We use the high-performance GPU physics simulator Isaac Gym [6]. The experiments are performed on a single NVIDIA RTX A6000 GPU with 48GB of VRAM.

B. Experiments

In this work, we assume that the objects are located on a flat surface in front of the robot. Thus, there are three possible natural poses in which drills, spray bottles or mugs can be: standing upright and laying on their left or right side. All other possible poses on a flat surface are unstable and transition quickly to the one of the described poses. Mugs can also be positioned upside down; however, we do not use this pose in our experiments to ensure that the results are consistent and comparable between object categories.

Actions are generated with a frequency of 30 Hz. An episode terminates when: (i) a target functional pre-grasp is reached, (ii) an object falls from the table, or (iii) a maximum number of steps is reached. We set the maximum number of steps to 200, which corresponds to ≈ 6.7 seconds. The objects are spawned on a table in front of the robot, such that at least 75% of their bounding box is in its manipulation workspace. Poses in which objects are lying on their sides are the most challenging for functional grasping because of the occlusion. For this reason, we focus on such poses and use the following object rotation distribution: 20% of the objects are upright, 40% are on their left side, and 40% are

[1] `https://free3d.com`, `https://3dsky.org`

[2] `https://github.com/AIS-Bonn/fun_cat_grasp_dataset`

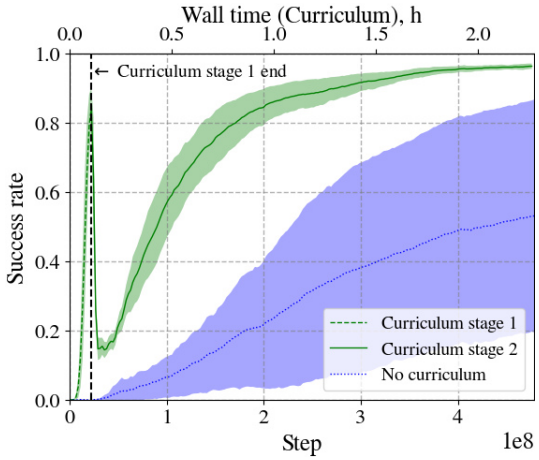

Fig. 4: Average success rates during the curriculum ablation experiment (mean and 95% confidence intervals). The two-stage curriculum significantly improves convergence stability, compared to the runs without curriculum.

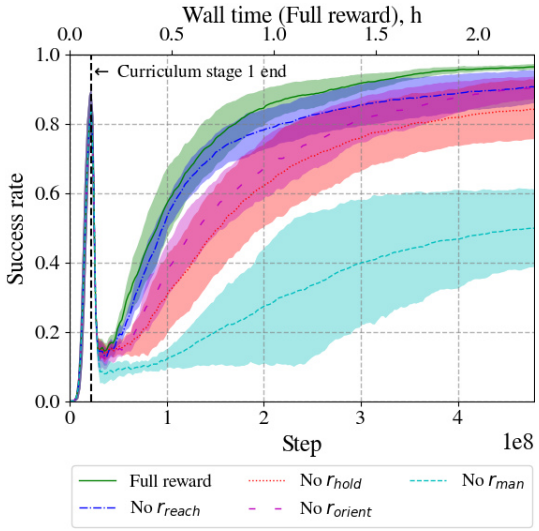

Fig. 5: Average success rates during the manipulation reward ablation experiment (mean and 95% confidence intervals). Disabling single reward components slightly deteriorated the convergence rate and stability. Disabling the whole manipulation reward component made the learning process significantly slower and less stable.

on their right side. The yaw angle and the object position are sampled uniformly. The hand starts at a random 6D pose above the table. Notably, objects laying on the right side require more complex pre-grasp manipulation for functional grasping with the right hand. Learning is performed on the training set of 30 objects. A target functional pre-grasp was defined for each object manually.

To make the simulation setup more realistic, Gaussian noise is applied to all observations supplied to the policy. For positions and distances, the zero-mean noise has $\sigma = 3\,\text{mm}$. For rotations, the zero-mean noise has $\sigma = 5°$. The only two observation which do not have noise are the object category and the target grasp. In each environment, an object is assigned a realistic random mass. The mass distribution in kg per category is represented by a Gaussian: $\mathcal{N}(1.4, 0.2)$ for drills, $\mathcal{N}(0.5, 0.15)$ for spray bottles, and $\mathcal{N}(0.3, 0.07)$ for mugs. Both noise and mass are limited to deviate from the mean for not more than 3σ. We scale the reward components: target grasp reward r_T by 5000, orienting reward r_{orient} by 500, and holding reward r_{hold} by 25. The other reward components are not scaled.

We train the policy on a single GPU with 16,384 parallel environments. Each policy in this evaluation is trained three times with three different seeds to assess convergence stability. First, we perform an ablation study of the two-stage curriculum proposed in Section III-D. During the first stage, the objects are spawned with a nominal rotation and close to the hand. Since at this stage the grasp is easily reachable, we disable the manipulation reward r_{man} to ensure quicker convergence. The first stage continues until at least 50% success rate is achieved for each object. During the second stage, the objects are spawned with rotations described above and the full reward is used. Fig. 4 shows average success rates during learning with and without curriculum. In addition, the wall time is shown for the curriculum runs. Wall time of other runs is similar ($\pm 10\,\text{min}$). The first stage of the curriculum is completed quickly. The second stage takes longer, but all three runs reliably converge to a success rate of 97% in under three hours with little variance. Without curriculum, the policy achieves only $\approx 50\%$ success rate and has a large variance within runs. Hence, the two-stage curriculum significantly improves convergence stability and success rate. We used a default discounting factor $\gamma = 0.95$ in all experiments. Lower values, such as $\gamma = 0.9$ decreased learning speed significantly, since only short time spans were represented in rewards, leaving longer-term consequences of complex manipulation unrepresented. Higher values, such as $\gamma = 0.975$ did not provide any significant improvement.

Next, we perform an ablation study of the proposed multi-component reward function. We train five policy variants: full reward, with disabled reward component encouraging moving the hand towards the object r_{reach}, with disabled reward component encouraging holding the object r_{hold}, with disabled reward component encouraging rotating the object towards the nominal rotation r_{orient}, and finally, with disabled whole manipulation component $r_{\text{man}} = r_{\text{reach}} + r_{\text{hold}} + r_{\text{orient}}$. Fig. 5 shows the of average success rates for this ablation study. One can observe that when a single component of the manipulation reward is disabled, the policy learns to achieve the goal slower, but still reliably makes progress towards high success rate. The most influence has the removal of the r_{hold} component. Without r_{hold}, the policy had the highest variance within runs and achieved the lowest success rate among single-component ablations. This shows that the reward component encouraging holding behavior is the most important one of the proposed manipulation reward. A deteriorated, but still reliable convergence without single reward terms suggests that although each component is important, the formulation is generic enough to not depend on the exact details. In contrast, disabling the whole manipulation reward r_{man} has a drastic negative effect on the performance of

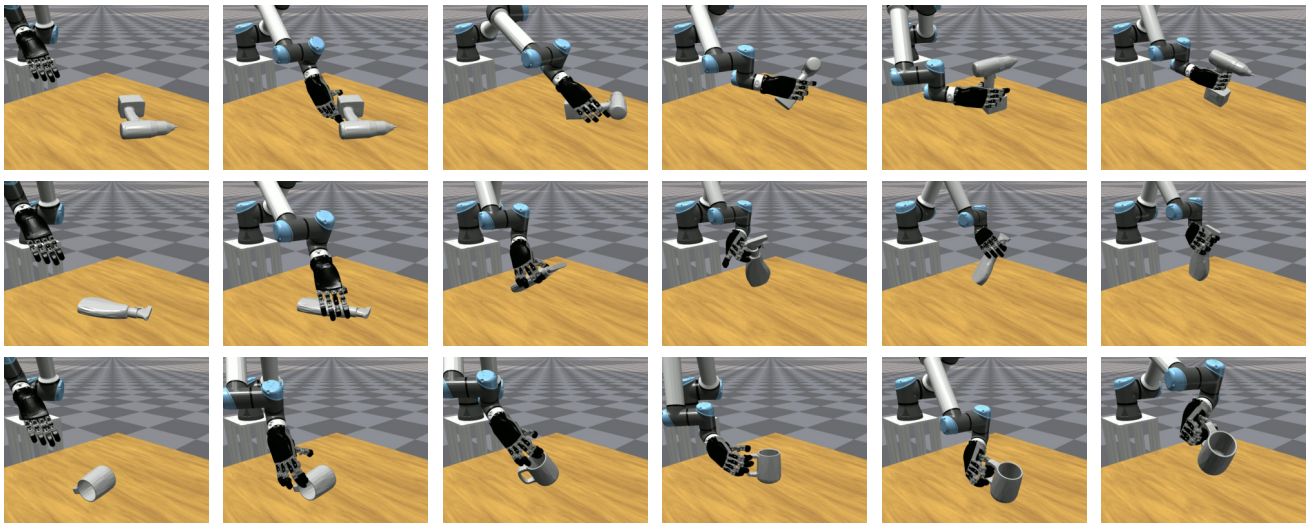

Fig. 6: Rollouts of policy manipulating unseen objects of known categories, which are positioned in a way that a direct functional grasp is impossible. Top to bottom: drill, spray bottle, and mug. Note the functional grasps achieved in the end.

the policy. Although it achieved a success rate of 50%, it struggled to learn a robust behavior for objects in difficult configurations. Overall, this ablation study demonstrated that the proposed manipulation reward component significantly speeds-up learning of dexterous pre-grasp manipulation.

To evaluate the generalization capability of the learned policy, we measure the success rate of the policy learned with curriculum and full reward on the training set and on the test set. The training set consists of 30 objects, ten for each of the three categories. The test set consists of nine novel objects of the known categories. We perform 100 grasping attempts for each object. This results in 3000 attempts for the training set and 900 attempts for the test set. Object initial rotations are sampled as during learning: 20% upright, 40% on the left side and 40% on the right side. Once the target pre-grasp is reached, the success is tested by closing the hand. If the object stays in the hand and the key condition of a functional grasp (such as an index finger on the trigger) is satisfied, an attempt is considered successful. We allocate 300 steps or 10 s per episode. Fig. 6 shows example rollouts for three test set objects. One can observe that dexterous interactive pre-grasp manipulation has been learned that leads to functional grasps for all three object categories. The observed success rates for all object categories are reported in Table I. On the training set, the learned policy shows a high success rate of 97.7%. As expected, on the test set the success rate is with 94.1% lower, but still high. The highest success rates were achieved on mugs. This is because they are relatively easy to flip over from the side position and have a simple geometry. The hardest object category was the spray bottles. This is because spray bottles are narrow, have a high center of mass, and can be easily dropped. Videos of the learned interactive functional grasping behavior are available online[3]. One can observe that complex pre-grasping strategies such as repositioning the object, reorienting and up-righting the object, and regrasping have been learned. The policy learned to reattempt the subtasks in case of failures.

TABLE I: Average success rate per category in %.

Category	Training set	Test set
Drills	96.0 ± 1.2	94.3 ± 2.6
Spray bottles	97.7 ± 0.9	92.3 ± 3.0
Mugs	99.3 ± 0.5	95.6 ± 2.3
Σ	97.7 ± 0.5	94.1 ± 1.5

*Mean ± 95% confidence interval is shown.

C. Discussion

The evaluation showed that the proposed approach is capable of consistently learning robust dexterous manipulation policies for functional grasping in simulation. The main strength of the approach is the generality of the proposed reward function and two-stage curriculum, which do not require any category or instance-specific engineering. At the same time, the multi-component reward function provides dense continuous rewards which quickly guide the policy towards general and robust behavior without a need for human demonstrations. In combination with a high-performance GPU simulation, complex pre-grasping strategies are learned in under three hours. The state and the reward are formulated in a way that is agnostic of the robotic arm joint number and can be easily adapted to a hand with an arbitrary number of fingers and controlled DoFs.

The main limitation of the proposed approach is its reliance on frequent and accurate estimation of the target object pose. In the real world in presence of the robotic hand, 6D object pose estimation is challenging [33]. Transferring our approach to a real robot is a prominent future work. Although the learned policies showed a robust behavior, it is likely that additional real-world learning will have to be performed in order to close the sim2real gap. For this, we envision an additional, smaller network on top of our policy.

[3]https://www.ais.uni-bonn.de/videos/CASE_2023_Pavlichenko

Such a corrective policy could be learned online on the real robot [34], using the proposed dense multi-component reward for faster convergence.

V. Conclusion

In this paper, we presented a deep reinforcement learning approach for dexterous categorical pre-grasp manipulation for functional grasping with an anthropomorphic hand. We introduced a dense multi-component reward function and a two-stage curriculum to quickly learn a single policy for dexterous manipulation of complex objects of three categories.

Our experiments demonstrated that learning with our approach reliably converges and produces policies with a high success rate, even for previously unseen object instances of known categories. Complex pre-grasping strategies such as repositioning the object, reorienting and up-righting the object, and regrasping have been learned.

Ablation studies confirmed the importance of the proposed multi-component reward function and the curriculum. Our approach utilizes a high-performance GPU-based simulation, and the policy was learned on a single GPU in less than three hours. Our policy achieved 94.1% success rate for functional grasping of novel object instances.

References

[1] D. Pavlichenko, D. Rodriguez, C. Lenz, M. Schwarz, and S. Behnke, "Autonomous bimanual functional regrasping of novel object class instances," *IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, pp. 351–358, 2019.

[2] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, "Learning agile and dynamic motor skills for legged robots," *Science Robotics*, vol. 4, no. 26, pp. 58–72, 2019.

[3] D. Rodriguez and S. Behnke, "DeepWalk: Omnidirectional bipedal gait by deep reinforcement learning," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 3033–3039.

[4] D. Chen, Q. Qi, Z. Zhuang, J. Wang, J. Liao, and Z. Han, "Mean field deep reinforcement learning for fair and efficient UAV control," *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 813–828, 2021.

[5] Y. Pane, S. Nageshrao, J. Kober, and R. Babuska, "Reinforcement learning based compensation methods for robot manipulators," *Engineering Appl. of Artificial Intelligence*, vol. 78, pp. 236–247, 2019.

[6] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, and G. State, "Isaac Gym: High performance GPU based physics simulation for robot learning," in *Thirty-fifth Conf. on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[7] P. Mandikal and K. Grauman, "Learning dexterous grasping with object-centric visual affordances," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2020, pp. 6169–6176.

[8] Y. Qin, B. Huang, Z.-H. Yin, H. Su, and X. Wang, "Dexpoint: Generalizable point cloud reinforcement learning for sim-to-real dexterous manipulation," in *Conf. on Robot Learning (CoRL)*, 2022.

[9] L. Han and J. Trinkle, "Dextrous manipulation by rolling and finger gaiting," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, vol. 1, 1998, pp. 730–735.

[10] M. R. Dogar and S. S. Srinivasa, "Push-grasping with dexterous hands: Mechanics and a method," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2010, pp. 2123–2130.

[11] L. Y. Chang, S. S. Srinivasa, and N. S. Pollard, "Planning pre-grasp manipulation for transport tasks," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2010, pp. 2697–2704.

[12] Muhayyuddin, M. Moll, L. Kavraki, and J. Rosell, "Randomized physics-based motion planning for grasping in cluttered and uncertain environments," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 712–719, 2018.

[13] K. Hang, A. S. Morgan, and A. M. Dollar, "Pre-grasp sliding manipulation of thin objects using soft, compliant, or underactuated hands," *IEEE Robotics and Autom. Letters*, vol. 4, no. 2, pp. 662–669, 2019.

[14] H. Zhu, A. Gupta, A. Rajeswaran, S. Levine, and V. Kumar, "Dexterous manipulation with deep reinforcement learning: Efficient, general, and low-cost," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019, pp. 3651–3657.

[15] O. M. Andrychowicz, B. Baker, M. Chociej, R. Jzefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, J. Schneider, S. Sidor, J. Tobin, P. Welinder, L. Weng, and W. Zaremba, "Learning dexterous in-hand manipulation," *The International Journal of Robotics Research*, vol. 39, no. 1, pp. 3–20, 2020.

[16] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1334–1373, 2016.

[17] W. Zhou and D. Held, "Learning to grasp the ungraspable with emergent extrinsic dexterity," in *IEEE ICRA Workshop: Reinforcement Learning for Contact-Rich Manipulation*, 2022.

[18] Z. Sun, K. Yuan, W. Hu, C. Yang, and Z. Li, "Learning pregrasp manipulation of objects from ungraspable poses," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2020, pp. 9917–9923.

[19] A. Rajeswaran, V. Kumar, A. Gupta, G. Vezzani, J. Schulman, E. Todorov, and S. Levine, "Learning complex dexterous manipulation with deep reinforcement learning and demonstrations," in *RSS*, 2018.

[20] I. Radosavovic, X. Wang, L. Pinto, and J. Malik, "State-only imitation learning for dexterous manipulation," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2021, pp. 7865–7871.

[21] P. Mandikal and K. Grauman, "Dexvip: Learning dexterous grasping with human hand pose priors from video," in *Conf. on Robot Learning (CoRL)*, 2022.

[22] Z. Chen, K. V. Wyk, Y.-W. Chao, W. Yang, A. Mousavian, A. Gupta, and D. Fox, "Learning robust real-world dexterous grasping policies via implicit shape augmentation," in *Conf. on Robot Learning*, 2022.

[23] S. Dasari, A. Gupta, and V. Kumar, "Learning dexterous manipulation from exemplar object trajectories and pre-grasps," in *Conf. on Neural Information Processing Systems (NeurIPS)*, 2022.

[24] D. Kappler, L. Chang, M. Przybylski, N. Pollard, T. Asfour, and R. Dillmann, "Representation of pre-grasp strategies for object manipulation," in *IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, 2010, pp. 617–624.

[25] I. Baek, K. Shin, H. Kim, S. Hwang, E. Demeester, and M.-S. Kang, "Pre-grasp manipulation planning to secure space for power grasping," *IEEE Access*, vol. 9, pp. 157 715–157 726, 2021.

[26] D. Kappler, L. Y. Chang, N. S. Pollard, T. Asfour, and R. Dillmann, "Templates for pre-grasp sliding interactions," *Robotics and Autonomous Systems*, vol. 60, no. 3, pp. 411–423, 2012.

[27] M. Mosbach and S. Behnke, "Efficient representations of object geometry for reinforcement learning of interactive grasping policies," in *IEEE Int. Conf. on Robotic Computing (IRC)*, 2022, pp. 156–163.

[28] D. Rodriguez and S. Behnke, "Transferring category-based functional grasping skills by latent space non-rigid registration," *IEEE Robotics and Automation Letters*, vol. PP, pp. 1–8, 2018.

[29] T. Zhu, R. Wu, X. Lin, and Y. Sun, "Toward human-like grasp: Dexterous grasping via semantic representation of object-hand," in *IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, 2021.

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

[31] D. Makoviichuk and V. Makoviychuk, "rl-games: A high-performance framework for reinforcement learning," https://github.com/Denys88/rl_games, 2021.

[32] D. Rodriguez, A. Di Guardo, A. Frisoli, and S. Behnke, "Learning postural synergies for categorical grasping through shape space registration," in *IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, 2018, pp. 270–276.

[33] A. Amini, A. S. Periyasamy, and S. Behnke, "YOLOPose: Transformer-based multi-object 6D pose estimation using keypoint regression," in *17th International Conference on Intelligent Autonomous Systems (IAS)*, ser. LNCS, vol. 577. Springer, 2022, pp. 392–406.

[34] D. Pavlichenko and S. Behnke, "Real-robot deep reinforcement learning: Improving trajectory tracking of flexible-joint manipulator with reference correction," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2022, pp. 2671–2677.

Acknowledgment: We would like to thank Malte Mosbach for providing the source code of his work on object geometry representations for grasping [27] as a base for this research.