

EARL: Eye-on-Hand Reinforcement Learner for Dynamic Grasping with Active Pose Estimation

Huang, Baichuan; Yu, Jingjin; Jain, Siddarth

TR2023-118 September 13, 2023

Abstract

In this paper, we explore the dynamic grasping of moving objects through active pose tracking and reinforcement learning for hand-eye coordination systems. Most existing vision-based robotic grasping methods implicitly assume target objects are stationary or moving predictably. Performing grasping of unpredictably moving objects presents a unique set of challenges. For example, a pre-computed robust grasp can become unreachable or unstable as the target object moves, and motion planning must also be adaptive. In this work, we present a new approach, Eye-on-Hand Reinforcement Learner (EARL), for enabling coupled Eye-on-Hand (EoH) robotic manipulation systems to perform real-time active pose tracking and dynamic grasping of novel objects without explicit motion prediction. EARL readily addresses many thorny issues in automated hand-eye coordination, including fast-tracking of 6D object pose from vision, learning control policy for a robotic arm to track a moving object while keeping the object in the camera's field of view, and performing dynamic grasping. We demonstrate the effectiveness of our approach in extensive experiments validated on multiple commercial robotic arms in both simulations and complex real-world tasks.

2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)

EARL: Eye-on-Hand Reinforcement Learner for Dynamic Grasping with Active Pose Estimation

Baichuan Huang, Jingjin Yu, and Siddarth Jain .

Abstract—In this paper, we explore the dynamic grasping of moving objects through active pose tracking and reinforcement learning for hand-eye coordination systems. Most existing vision-based robotic grasping methods implicitly assume target objects are stationary or moving predictably. Performing grasping of unpredictably moving objects presents a unique set of challenges. For example, a pre-computed robust grasp can become unreachable or unstable as the target object moves, and motion planning must also be adaptive. In this work, we present a new approach, **Eye-on-hAnd Reinforcement Learner (EARL)**, for enabling coupled Eye-on-Hand (EoH) robotic manipulation systems to perform real-time active pose tracking and dynamic grasping of novel objects without explicit motion prediction. EARL readily addresses many thorny issues in automated hand-eye coordination, including fast-tracking of 6D object pose from vision, learning control policy for a robotic arm to track a moving object while keeping the object in the camera’s field of view, and performing dynamic grasping. We demonstrate the effectiveness of our approach in extensive experiments validated on multiple commercial robotic arms in both simulations and complex real-world tasks.

I. INTRODUCTION

Robotic manipulation of everyday objects in dynamic environments constitutes a *fundamental skill* for enabling the next generation of advanced robotic systems. Providing robots with 6 degrees of freedom (DoF) pose tracking and grasping capability in unstructured and dynamic environments beyond static tabletop scenarios can benefit many automation applications. For example, a human handing over an object to the robot, assembly of industrial parts, etc.

Autonomous grasping and manipulation of objects in stationary settings have been studied extensively [1]. Dynamic environments bring many challenges for performing grasping. First, the target object might move with an unknown motion, which requires understanding and predicting the object’s motion or continuous tracking and active following by the manipulator. Second, computed motion plans can become obsolete, and thus dynamic environments require online or fast replanning. Additionally, the approach direction for grasp planning changes with the object’s motion, and therefore, a stable grasp can become unreachable and unstable as the target object moves. Lastly, active perception [2] is required for dynamic grasping with Eye-on-Hand (EoH) systems, as the robot can lose track of the target object because of the robot’s motion or as the target moves away from the field of view (FoV) of the camera.

Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA 02139, USA baichuan.huang@rutgers.edu, jingjin.yu@rutgers.edu, sjain@merl.com

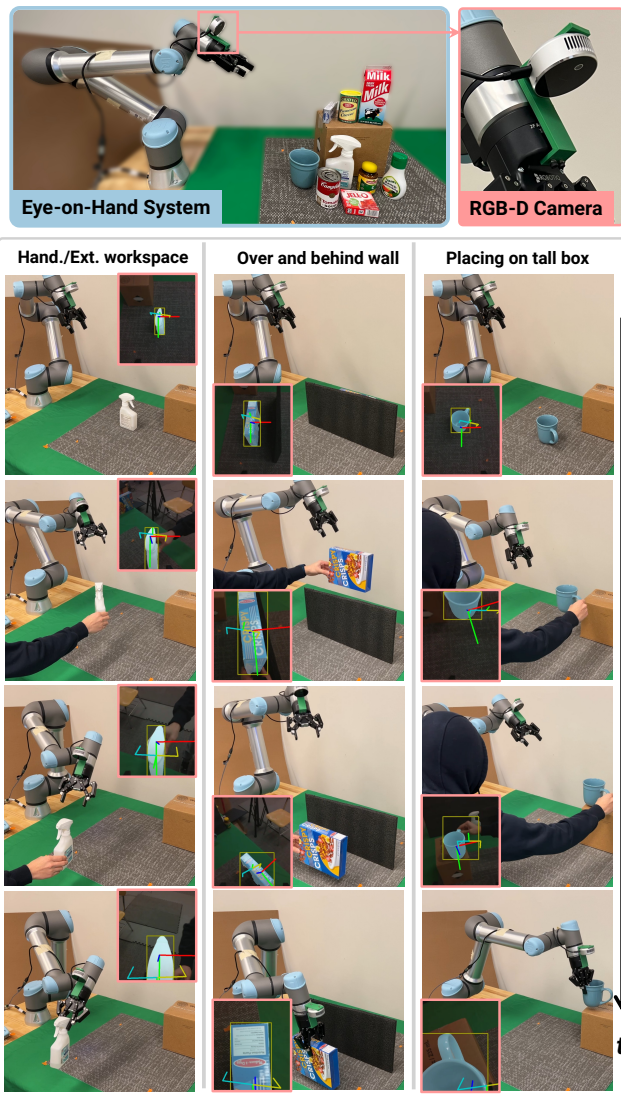


Fig. 1: *Top*: Eye-on-Hand (EoH) system with a wrist-mounted RGB-D camera and a robotic manipulator. *Bottom*: Sample of experimental tasks used to evaluate EARL in challenging setups. *Hand./ext. workspace* shows that the system works outside the trained workspace and can perform human-robot handover. *Over and behind wall* shows the system can overcome occlusions typical external camera setup faces by tracking across a barrier. *Placing on tall box* shows the system can handle significant elevation changes.

A supermajority of vision-based manipulation systems typically utilize cameras fixated above the workspace [3], [4]. Thus, the perception subsystem (e.g., RGB-D cameras) and the manipulation system (e.g., robot arms) are *decoupled*. Such settings implicitly assume ideal viewing distances and angles for focusing on target objects. This fixation may require large clearances above/around the workspace, rendering the overall system inflexible and unsuitable for some

applications, especially when occlusions are unavoidable or spaces are confined, e.g., retrieving a condiment jar in a cabinet or performing an inspection in a pipe using a snake-like robot [5] and robot dog. Such decoupling can cause occlusions or loss of tracking in many scenarios, especially when the target object is moving. Thus, mounting a camera on the wrist of a manipulator can be an alternative [6]. In some cases, the vision sensor is mounted on the robot arm near the end-effector [7], [8]; however, pose tracking has been carried out afar with another static workspace camera, rendering the perception process decoupled from the object manipulation operation.

In this work, we relax some of the assumptions and address the challenges mentioned above by tackling robotic grasping with an “Eye-on-Hand” (EoH) system, where the sensory perception system is coupled with the manipulator (see Fig. 1, *Top*). The binding constraint on the “eye” and the “hand” subsystems, dynamic range changes for perception, generalization to different objects, and no prior knowledge of the object’s motion profile bring many challenges. We propose a novel solution based on *active pose tracking and reinforcement learning*, which enables EoH robotic systems to execute dynamic grasping of moving objects in real-time. Our proposed method, called **Eye-on-hAnd Reinforcement Learner (EARL)**, approaches the target with the constraint of keeping the target object in the FoV, despite drastically changing distances and view angles between the camera and the target object. We propose a reinforcement learner that maps from the moving object pose differentials and adaptive changes of grasp pose to desirable robot arm joint velocities for dynamic grasping. We propose to perform active pose tracking to encode visual observations and perform the training through a carefully designed curriculum to overcome the challenges. Through extensive evaluation, we demonstrate that EARL robustly and effectively tracks and approaches the moving target object until successful grasping is realized.

In summary, our work makes the following contributions:

- It introduces a high-performance manipulation framework for Eye-on-Hand (EoH) robotic systems consisting of proposed active pose tracking with a moving camera and a curriculum-trained reinforcement learning method EARL with sim-to-real generalization capability for full 6-DoF dynamic grasping of novel objects with no prior knowledge of the object’s motion profiles.
- It extensively evaluates our proposed framework on a set of novel objects with different motion types and complex tasks in various settings for dynamic grasping in simulation and the real world, including multiple robotic arms as EoH systems.

We explore related work in Sec. II. System setup and task description are described in Sec. III. In Sec. IV, we present the overall framework EARL and our proposed method. Experimental details are described in Sec. VI. We discuss and conclude in Sec. VII.

II. RELATED WORK

Vision-based robotic grasping solutions [9] can categorize along several dimensions. Model-based approaches rely upon knowledge about the target object [10], e.g., a CAD model. Model-free methods directly propose grasp candidates and aim for generalization to novel objects. Analytic or geometric methods analyze the shape of a target object for grasp pose identification [11]. Data-driven grasp synthesis uses learning [1] and has significantly progressed due to better learning methods and data availability. Some approaches sample and rank grasp candidates using deep neural networks [12]. Reinforcement Learning (RL) approaches also find applicability for grasp synthesis [13], [14], where suitable candidates are learned and validated with repeated interactions in a simulation or real environment.

Another essential distinction is open or closed-loop grasping. Approaches performing open-loop grasp execution are sensitive to calibration errors, failing to handle dynamic environments. Some methods tackle closed-loop grasping using continuous visual observations [15] with visual servoing control [16] or RL. Some methods require vast data for training; for example, QT-Opt [17] collected data within several weeks across seven robots. Many methods characterize by constrained state-action spaces, for example, focusing on top-down grasping [15], [17], [18] in 4-DoF, and thus are limited in task scope. Only a handful of methods target grasp synthesis for 6-DoF [19] to grasp objects from a broader range of unstructured settings beyond the tabletop scenario.

Most existing methods focus on static environments, typically utilizing cameras fixed in the workspace to perform grasp synthesis. Such settings can suffer occlusions in many scenarios, especially when the robot is approaching the target. Employing multiple cameras can help mitigate occlusion in some scenarios. In real-world tasks, interacting with dynamic objects or manipulating objects while a robot is in motion can benefit many applications. For such dynamic grasping tasks, mounting a camera on a robot’s wrist can significantly reduce occlusion’s impact and enable the robot to perform grasp synthesis in various scenarios.

Grasping in dynamic environments presents additional challenges and requires the robot’s motion and grasp planning to be adaptive and real-time. Morrison et al. [15] present a generative grasping convolutional neural network (GG-CNN) for fast closed-loop grasping in slightly moving scenes. Their method generates quality measure for every pixel in depth images. Similar to our work, they use a wrist-mounted camera but only evaluate their approach for 4-DoF top-down grasping. Moreover, GG-CNN has high requirements for data annotation. Marturi et al. [7] dynamically plan trajectories to grasp a moving object based on visual information. However, their approach uses multiple cameras. Song et al. [19] propose an end-to-end RL method for closed-loop grasping where a DNN models a value function that maps the images from the wrist-mounted camera to the expected rewards in that state. The approach can handle 6-DoF grasps for slightly moving scenes but only operates on

a discrete action set. One common approach for dynamic grasping involves motion prediction for the target object to improve success. The motion prediction ability can be helpful in both planning a grasp and approaching the object. Akinola et al. [20] introduce a reachability and motion awareness solution. They implement a recurrent neural network (RNN) for modeling and predicting object motion that works well for linear, sinusoidal, and circle trajectories. However, it could be of poor generalization in unseen and complex trajectories, e.g., random movements, and their solution uses a position-fixed camera. In this work, we propose a model-free RL-based method with a wrist-mounted camera for picking a moving object with continuous actions while keeping the object in the FoV of the robot. Our approach does not limit the grasp synthesis to a top-down direction and generalizes well for unseen and complex trajectories.

III. SYSTEM SETUP AND PROBLEM DESCRIPTION

This work endows Eye-on-Hand (EoH) systems with the capability to track and grasp moving target objects. The EoH system comprises a high-DOF robotic manipulator, an end-effector, and a wrist-mounted sensory perception system that is fixated on the manipulator near the end-effector. The coupling between the perception and manipulation subsystems means that such an EoH system can operate without workspace constraints and occlusions faced by most existing systems [21] using an external camera.

We instantiate experiments and demonstrations on two robotic arms as such EoH systems to showcase our methods' generality. These systems use Universal UR-5e (6-DOF) and Kinova Gen3 (7-DOF), respectively, both commanded using *joint velocity control*. The vision system includes an Intel RealSense L515 RGB-D camera for perception, secured on the gripper using a custom 3D-printed mount. We use a Robotiq 2F-85 two-finger gripper as the end-effector.

Problem Definition: We focus on enabling coupled EoH systems to perform dynamic grasping in 6-DoF (i.e., $SE(3)$) of a moving object with *a priori* unknown motion. We make no assumptions about the shape or identity of target objects other than that they are rigid bodies and graspable by the end-effector. Additionally, the target object can move freely in the robot's reachable workspace (approachable by the robot). Completing this task with the EoH system requires another sub-task of tracking the moving object, following it with the robot's motion such that it keeps the target object in the FoV of the EoH camera while approaching the target. The task is successful if the robot can grasp and pick up the object.

IV. EARL FRAMEWORK

In this section, we describe the various components of our framework (illustrated in Figure 2). First, we present our framework overview. We then describe our visual processing unit that tracks target object poses in real-time. We then describe the grasp planning pipeline that continuously adapts grasp selection. Finally, we present our arm motion generation method based on curriculum-trained RL.

A. Framework Overview

Grasping in a dynamic environment presents many challenges. A stable and reachable grasp can become unreachable and unstable as the target object moves. The EoH system can also lose track of the dynamic target object. We employ RL for EoH systems, where we learn a control policy (EARL) in simulations for dynamic grasping of arbitrary objects, and we propose to utilize the object's active pose information as visual feedback. Our proposed method enables dynamic grasping with tracking of novel objects without explicit motion prediction requirements. First, we perform pose estimation of the target with the moving EoH camera for providing visual feedback. We select a design that continuously tracks the target object in 2D using learned features [22], [23] and subsequently performs pose estimation using a combination of 2D features [24] and depth information [25] to speed up the processing and ensure sufficient tracking accuracy. The control policy is a Proximal Policy Optimization (PPO) based [26] RL framework which maps object pose differentials to desirable robot arm joint velocities. Since we work with an EoH system, the robot arm must continuously adjust the pose of the RGB-D camera to keep the target object in the FoV. Simultaneously, the control policy must guide the end-effector to approach the object and grasp it. Achieving all these requirements naturally and smoothly in real-time is only possible through a finely-tuned, multi-stage control policy, as described in Sec. V. We perform grasp planning with multiple grasp proposals on the target object. The best grasp pose is dynamically updated, which may happen as the target object continuously moves in 6D. We dynamically select a suitable candidate for biasing the control policy actions. The overall framework is shown in Fig. 2, and the steps are detailed in the following sections.

B. Active Object Pose Tracking

Active perception [2] implies computer vision implemented with a movable camera, which can intelligently alter the viewpoint to improve the system's performance. In this work, we consider active tracking with an Eye-on-Hand camera. Many methods focus on top-down grasping [15], [17], where they constraint the target object in the 2D workspace. This case can be relatively simple. In this work, we target grasp synthesis in 6-DoF, which requires the robot to be aware of the target object's 3D position and 3D orientation. We perform 6D pose estimation with the EoH system for tracking and computing an encoding of the target object from visual observations. Typically, 6D object pose estimation methods [9] assume known object models and can categorize into correspondence-based, template-based, and voting-based. We make no assumptions about the object's model and motion profile and continuously track the object's pose with the moving camera.

Given the first frame RGB-D image I_0 containing the target object O , we aim to continuously track O 's 6D pose relative to the camera at any time t in image I_t . We realize this by a correspondence-based approach and leverage BundleTrack [25] method for maintaining a keyframe

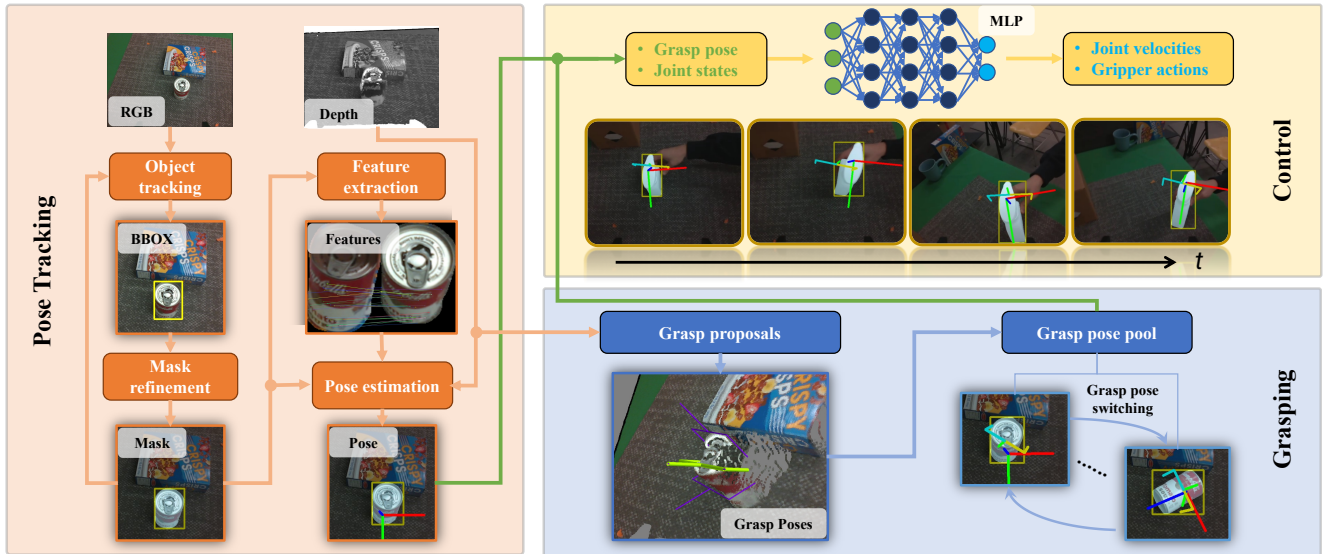


Fig. 2: Illustration of the proposed framework. RGB-D image input initializes object tracking and mask refinement for pose estimation components. The pose tracking module provides a 6D pose of the target object. A grasp pose pool is built from a grasp proposal network. The grasp planning module computes a suitable grasp pose on the target. The control policy commands the EoH system to track and finally grasp the target using the robot arm.

memory pool and performing online pose graph optimization that takes both feature correspondences and dense pixel-wise correspondences from a depth image. To accomplish fast and active pose tracking of novel objects in motion using only a sequence of RGB-D images, we introduce several augmentations to enhance the BundleTrack.

At time t , the pose tracking process starts with obtaining a bounding box B_t of the target object, employing a transformer-based 2D object tracker [22], capable of tracking an object through multiple frames with a series of RGB images. We then enhance the fit of the bounding box B_t with Alpha-Refine [23] and simultaneously obtain an object mask M_t . Alpha-Refine [23] is a boosting method that extracts and maintains detailed spatial information. These augmentations support fast tracking and work with novel objects without requiring additional training or object models. We utilize the mask M_t to provide more accurate RGB and depth information as input to the BundleTrack, leading to direct performance gains for fast-tracking the 6D pose. A key component for correspondence-based pose estimation is the feature extractor. We employ R2D2 [24] feature detector, which predicts a set of sparse locations as object features in the input image that are repeatable and reliable for local feature matching and thus outperforms the LF-Net [27] detector utilized in the BundleTrack. We note that the color image input to R2D2 contains only the target object cropped based on M_t . Our method enables more robust active pose tracking in real-time robot control.

C. Grasp Planning

We use the first RGB-D image to initialize the pose tracking (see Sec. IV-B) and compute grasp proposals. Selecting a good set of candidate grasp pose is important for two reasons. First, computing grasp poses on the fly is time-consuming, making it difficult to achieve real-time performance. Second, it is required to compute a desired output from a control policy towards a graspable pose on the moving object that

drives the end-effector close to the target object. For grasp planning, we compute a set of grasp poses using Contact-GraspNet [28], which utilizes a raw depth image and object mask as input to generate a set of 6-DoF grasp proposals. This approach works on novel objects and environments without the need for fine-tuning. Next, we compute a set of transformations from grasp poses to the 6D pose of the target object and store them in a grasp memory pool. All proposed grasp poses are relative to the target object’s pose, which helps us avoid doing computationally expensive real-time grasp synthesis. As both the EoH and the object move, the system dynamically chooses which grasp pose to track from the grasp memory pool at a given time. The selection of the grasp pose is driven mainly by *angular distance*. We use quaternion representation to store the orientation information. At each control iteration, the orientation of gripper q_g is compared with that of all grasp poses $q_i \in G$ via

$$\Delta q_i = q_g * q_i^{-1}, d_i = 2 \arcsin(\min(\|\Delta q_i\|, 1)).$$

The grasp pose with the smallest d_i is set to be the current target pose to be tracked.

D. Coupled Control of EoH System

Given active pose tracking and grasp planning for adjusting the target grasp for a moving object, next, the EoH system must track and follow the target and successfully perform dynamic grasping. Two constraints must be jointly satisfied by a policy to control the robot’s approach and grasp the target object: maintaining the object in the camera’s FoV and avoiding collisions, including self-collision.

As a first solution, we develop a standard PD controller driven by an inverse kinematics (IK) solver for the control policy. We treat this solution as a baseline approach for performing dynamic grasping with the EoH system.

Baseline control policy: Given an end-effector goal as a grasp pose target to the EoH system at every frame, using the IK (damped least squares as described in [29]) we develop a servoing method which generates desired joint velocity

commands using a PD control formulation. At a higher level, the servoing-based baseline repeatedly computes the pose of the next goal for the end-effector following a hand-designed trajectory optimization, shown in Fig. 3.

The EoH system navigates the gripper (with pose P_r) “above” the object to a pose at some distance from the object, from where a straight-line movement of the gripper can lead to the desired grasp pose P_t . In the figure, this stage is achieved as P_r ’s z -axis aligns with the object pose’s z -axis. To realize the design, a pose waypoint for the gripper P_r' can be computed as position $P_{r,p}'$ and rotation $P_{r,q}'$. The next position of pose $P_{r,p}$ needs to move the gripper toward the approaching direction (z -axis of P_t) while maintaining a fixed distance,

$$r = |P_{r,p} - P_{t,p}| = |P_{r,p}' - P_{t,p}|.$$

We realize this by moving on a sphere with a radius of r as shown in Fig. 3. If the z -axis between P_r and P_t are well aligned (smaller than a threshold θ_{align} , set to 0.175rad), then, r is decreased, otherwise, r is increased without exceeding r_{max} (set to 30cm).

V. TRAINING EARL

We propose an Eye-on-Hand reinforcement learner (EARL) as a control policy that directly maps the desired grasp pose and joint states to the robot’s desired joint velocity and gripper actions. Training an RL model with many continuous DoFs is challenging due to long policy horizons and sparse reward signals. In our case, the challenge is compounded by the strong coupling between vision and manipulation subsystems. To effectively train EARL in simulations, we formulate the task as a single-agent Markov Game. In each episode, a target object (possibly among other objects) is randomly placed in the workspace. The target object may randomly translate and rotate in the 3D workspace (gravity is disabled so that the object can move in the air). The robot is tasked to grasp the target object using a specific grasp pose. The environment resets itself upon a grasp success/failure or if object tracking fails (out-of-view).

Choice and Design of Learning Framework. We train EARL policy with a PPO [26] agent, which receives the state quantities of the target object and the EoH system and decides which action will be taken to execute dynamic grasping. We employ Isaac Gym [30] for efficient RL and data acquisition, capitalizing on PPO’s inherent support for parallel training. We perform simultaneous simulations of thousands of independent EoH systems in Isaac Gym for model training. The overview of training EARL is shown in Fig. 4. The MLP for PPO has three hidden layers of size [256, 128, 64].

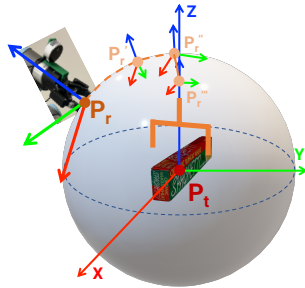


Fig. 3: Baseline control policy. Target grasp pose on the object is shown in orange. The robot arm should follow the trajectory of P_r', P_r'', P_r''' to reach the target object and finally grasp it at P_t .

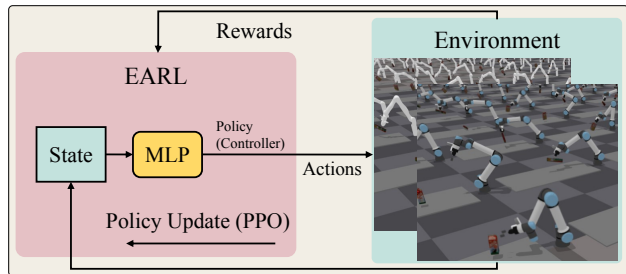


Fig. 4: Overview of training EARL.

For input, the reinforcement learner needs information on the target object, which we provide as an encoding represented by the object’s pose (translations and orientation), in addition to the state of the robotic arm. We describe a unique pose representation for the input. Because Euler angles have singularities and quaternions are non-unique, training the RL agent with such parameterizations can be difficult in [31]. Borrowing from [32], [33], we use four *keypoints* to represent the translation and orientation of a rigid body. These keypoints can be represented in many ways, and in our settings, we describe them using the corners of a fixed size square where the square is selected to coincide with the XY plane of the coordinate system of the given object’s pose, with its center coinciding with the origin of the pose coordinates. We represent the input as,

$$S = [K_r, K_{rg}, K_{gg'}, J_p, J_v, c],$$

where K_r are the keypoints of the gripper. Let K_g be the target (grasp) keypoints, then $K_{rg} = K_r - K_g$ is the error that should be minimized (reach 0) for a successful grasp. $K_{gg'} = K_g - K_{g'}$ is the change of the target grasp keypoints between the current and previous frames. J_p and J_v are the robot arm’s joint position and velocity vectors, respectively. J_p is scaled to $[0, 1]$ depending on the rotation limits. c , a Boolean variable, indicates whether the gripper is closed or open. Policy actions, as the output of the MLP, are the velocity target for each robot’s joint and a boolean command for the gripper. We choose joint velocities to control the robot arm because it results in comparatively smoother motions than position-based control.

Reward Shaping. The learning process can be much faster and successful through a suitable reward schedule with diverse guidance. We carefully perform reward shaping to encourage the robot to grasp the target object fast and securely. On a high level, the problem can be seen as tracking and grasping. The tracking portion moves the end-effector closer to the desired grasp pose while maintaining the camera’s view of the object and simultaneously avoiding collisions. The grasping component needs to determine an opportune moment for closing the gripper. We consider the case where the motion of the target object can be unpredictable, so the best strategy for the robot arm is to continuously track the target and grasp it at an opportune moment. We design the reward to have multiple components,

$$R = P_{dist} + P_{view} + P_{coll} + P_{over} + P_{align} + P_{move} + R_{grasp}.$$

P_{dist} , a dense penalty, is the distance between the gripper pose and grasp pose, which encourages the robot to minimize K_{rg} . P_{view} is a penalty given when the target pose is out-

of-view. This penalty helps the active pose estimation with the moving camera by keeping the object in FoV. P_{coll} is a penalty for any collision before grasping. It is easy for the robot to keep the object in view and avoid collision by staying far from the target object, a behavior that must be avoided. For that, P_{over} penalizes the agent when the distance between the gripper and grasp pose is larger than a threshold. P_{align} is a helper to regularize arm motion and help with learning. It contains three parts, the z-axis alignment of the gripper and grasp pose, the y-axis alignment of the gripper and grasp pose, and the centering of the object in the camera view. The z-axis and y-axis alignments guide the robot’s approach to the target. The closer the gripper is to the grasp pose, the higher probability the target object may collide with the robot or move out of the FoV, as the target object could randomly move. A P_{move} penalty is introduced to keep the robot’s gripper θ_d (set to 20cm) away from the object if the speed of the object is faster than ν_o (set to 4cm/s). R_{grasp} is the reward for closing the gripper when the target object has been successfully grasped. We use the contact force of two fingers to determine whether the gripper has grasped the target object in the simulation.

Curriculum Design. Learning dynamic grasping can be challenging for EoH systems, and training such a network from scratch with many continuous DoFs is challenging. We explore a curriculum design to train the EoH system efficiently. In particular, we use a three-stage curriculum [34], gradually increasing the task’s difficulty and dynamically changing the rewards scalar for efficient learning. In the first stage, the object is randomly placed in the workspace, and the robot’s gripper is randomly sampled closer toward the grasp pose on the object. The object randomly moves at a low speed ($\leq \nu_o$ cm/s). In addition, an episode is not terminated when the object is out-of-view. These relaxations significantly limit the initial search space to help training. Once the success rate of the initial policy is over a threshold, in the second stage of training, the environment will reset once the object is out-of-view, denoting task failure. The penalty for P_{view} , P_{coll} , P_{over} , P_{move} are increased, and they will continue to increase to reinforce that the robot should avoid these unwanted scenarios. Low penalty causes the policy to be aggressive, which leads to high collision rates or not keeping the object in view. On the other hand, a high penalty discourages the robot from approaching the object. We dynamically change penalty scales to encourage different behaviors and achieve a high success rate. In the last stage, we let the object move faster (the max ν_o is set to 8.5cm/s), and the robot gripper is fixed far from the target object at the start of each episode, working in the full workspace.

The RL training parameters are identical for UR-5e (6-Dof) and Kinova-Gen3 robots (7-Dof). The only difference is that the penalty scales of P_{view} and P_{coll} are set to be lower for Kinova due to the robot’s kinematics, which has long links between joints, making it comparatively harder to train. Once trained, our policy EARL runs in real-time on new objects and the training time amounts to just ~ 4 hours of the learning experience.



Fig. 5: *Top Left:* Training set objects. *Bottom Left:* Validation set objects used in simulation. *Right:* Real test set objects. Note that the real objects are significantly more challenging.

Enabling Direct Sim-to-Real Adaption. A goal of EARL is to have RL agents trained in a simulator directly applicable to real-world EoH systems. We developed an effective technique of independent interest based on an observation that the sim-to-real gap does not strongly correlate with a control policy, and we encode visual observations as high-level representations. Instead, the gap is mainly caused by parameter differences between the simulation and the real system. With this observation, we *decouple* the sim-to-real gap reduction from training EARL. Using only the baseline PD-based controller (Sec. IV-D), running in both simulation and real-world EoH systems, we fine-tuned the necessary parameters to reduce the sim-to-real gap. For example, because real robot arms are torque-controlled at a lower level, damping parameters must be appropriately adjusted to realize accurate joint velocity control.

VI. EXPERIMENTS & RESULTS

We evaluated the proposed methods both in simulation (Issac Gym [30], [33]) and on real robots (Universal UR-5e 6-DoF (R1) and Kinova Gen3 7-DoF (R2)) as EoH systems. Both robots are equipped with a two-finger Robotiq 2F-85 gripper and an Intel RealSense L515 camera. The workspace for training is a cubic region of $40 \times 40 \times 40$ cm³. We evaluated our system on machines with a single GPU (Nvidia 3090, uses 8 GB memory). Our method can handle novel objects for which grasp pose can be reasonably tracked, allowing us only to use a few objects (four) for training. We tested in simulation and the real world with four unseen sets of objects each. Objects are selected from the YCB and HOPE datasets [35], [36]; and some household items. Fig. 5 shows these objects. The pipeline runs at 15 FPS.

A. Simulation Experiments

Task and Object Motion. The main objective of simulation experiments is to evaluate the control policies BASELINE and EARL for the dynamic grasping problem. In the simulation experiments, similar to training, the simulator gives the object’s pose, and we use one grasp per trial. To evaluate the policy, we created test cases that have four different motions as shown in Fig. 6.

- *Static (S).* Objects are randomly sampled in the workspace.
- *Linear (L).* Line paths in XYZ space are randomly sampled from boundary to boundary.

- *Oval* (O). Oval paths in the XY plane are randomly sampled, and the objects stop moving at a random time step. The z -axis is also randomly sampled beforehand.
- *Random* (R). The object randomly moves for some time, then stops moving for five seconds and repeats.

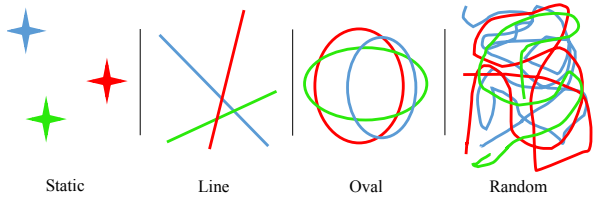


Fig. 6: Four motions are considered in the simulation evaluation. The figure illustrates these motions in 2D for visualization purposes, which happen in 3D and 6D. The RL control policy is trained only for random (6D) motion.

The object randomly moves with speed along any dimension set to a max. 5cm/s for translation, and 0.1rad/s for rotation. In the simulation, the target object never stops moving, even when slowed down considerably at random; it will “vibrate” around the center of mass to simulate uncertainty.

Metrics. We evaluate our method using the success rate, where the robot successfully grasped the target object in a given time (timeout). Three failure conditions are recorded: (1) collision, (2) losing track of the target object, and (3) the maximum episode length is reached. Since active pose estimation is not performed in the simulation experiments, checking that an object is out of view is done by checking the object’s pose center is beyond a certain threshold ($\sim 20^\circ$ in our experiments) away from the simulated camera’s principle axis. The maximum time per trial is 35 seconds (timeout). The robot is required to approach the target object and execute a swift grasp when appropriate.

Results. We generated one thousand test cases in simulations across the motion types and test objects and repeated them for five trials. EARL control policy outperforms the baseline method on both robot systems R1 and R2, as shown in Table I. The baseline method achieves an average success rate of 86.83%. EARL demonstrates an excellent average success rate of 95.73%. The primary failure mode of EARL is timeout due to the high penalty for collision and out-of-view, which encourages the robot to move conservatively toward the target object. Timeout failure for EARL is relatively low in random motion settings, likely due to the training being performed only using the case of random object motion. Nevertheless, EARL performs for other object motion types. In comparison, the performance of the baseline method goes down significantly as the difficulty of the task increases.

B. Real Robot Experiments

Tasks and Object Motion. The real experiments use actual perception and perform grasp planning with pose tracking by the EoH system. A human operator moves the object freely in the workspace based on the task scenario. We perform four motions (S/L/O/R) from simulation experiments to analyze the sim-to-real performance gap. The grasp pose pool is limited to one for a fair comparison. We also perform two more

A R	M	Success	Failure	Timeout	Collision	Out-of-View
B R1	S	94.90	5.10	0.20	4.90	0.00
	L	90.76	9.24	2.30	6.54	0.40
	O	86.54	13.46	1.14	5.30	7.02
	R	76.28	23.72	5.34	12.22	6.16
E R1	S	95.76	4.24	4.20	0.00	0.04
	L	95.00	5.00	4.74	0.00	0.26
	O	96.50	3.50	2.12	0.70	0.68
	R	97.80	2.20	0.60	0.60	1.00
B R2	S	95.90	4.10	2.32	1.78	0.00
	L	95.08	4.92	2.90	1.78	0.24
	O	83.24	16.76	7.32	3.86	5.58
	R	71.92	28.08	13.78	7.80	6.50
E R2	S	96.72	3.28	3.08	0.00	0.20
	L	92.98	7.02	5.56	0.50	0.96
	O	95.58	4.42	2.32	1.10	1.00
	R	95.48	4.52	0.64	3.18	0.70

TABLE I. Simulation experiment results (all values in %). A|R refers to the algorithm (baseline or EARL) and the robot 6-DoF (R1) or 7-DoF (R2). M refers to motion type (static (S), linear (L), oval (O), random (R)).

motion tasks for testing adaptive grasp switching: random and complex. The former executes random movements of the object, forcing a grasp pose switch to achieve success. The complex tasks involve a combination of motion types. These include three challenging tasks, as described in Fig. 1, which demonstrate generalization capability.

Results. We performed real robot trials using the UR-5e (6-DoF) as the EoH system for each method and motion type with each test object using random start and end positions (20 trials for each motion type per method). We ensure each trial is consistent across both methods. Table. II show these results. EARL outperforms the baseline method for all test cases achieving an average success rate of 91.25%, versus 73.75% with the baseline. Note that EARL policy is trained entirely in simulations. Comparing the simulation results demonstrates that we accomplish a small sim-to-real gap (less than 5%) on the motions (S/L/O/R) with our sim-to-real adaptation. Furthermore, EARL also performed well on the random motion tasks and complex tasks with adaptive grasp switching, achieving an average success rate of 90% and 80%, respectively. From our observation, the pose tracking module is bottleneck for achieving dynamic grasping without explicit modeling of motion prediction. We perform active pose tracking with object-level SLAM by considering texture, geometric information, and robust features; still, the RGB image suffers from motion blur and the depth image from noisy reflections and inaccurate measurements, affecting performance. Moreover, object tracking can suffer from complex backgrounds and deficiency of texture on the object itself. Despite that, EARL performs well in complex tasks such as human-robot handover and moving over and behind a barrier, overcoming object occlusion limitations that external camera setups typically face.

VII. CONCLUSIONS

We presented a framework for enabling EoH systems to perform 6-DoF dynamic grasping using visual sensory inputs with active pose tracking and a curriculum-trained RL method. Our method relates sensing to actuation and applies to complex robotic tasks utilizing visual feedback for

Object Motion Type	BASELINE Success	EARL Success
Static	85.00	100.00
Line	75.00	90.00
Oval	65.00	85.00
Random	70.00	90.00
Random (switch)	70.00	90.00
Complex (switch)	70.00	80.00

TABLE II. Real-world experiment results (all values in %). 20 trials were performed for each test case using the UR5e real robot as the EoH system. eye-in-hand control. We validated our contributions through extensive experiments in simulations and complex real-world tasks, attaining a high success rate on previously unseen objects. Our framework is generic concerning the task, but it does not consider explicit collision modeling, and the target cannot move faster than the robot. In the future, we will focus on mechanisms for recovering the target in case of tracking failures and handling more cluttered environments. Moreover, our method’s performance could be enhanced by leveraging recent advancements in pose estimation [37] and grasp detection methods.

REFERENCES

- [1] K. Kleeberger, R. Bormann, W. Kraus, and M. F. Huber, “A survey on learning-based robotic grasping,” *Current Robotics Reports*, vol. 1, no. 4, pp. 239–249, 2020.
- [2] S. Chen, Y. Li, and N. M. Kwok, “Active vision in robotic systems: A survey of recent developments,” *The International Journal of Robotics Research*, vol. 30, no. 11, pp. 1343–1377, 2011.
- [3] B. Huang, A. Boularias, and J. Yu, “Parallel monte carlo tree search with batched rigid-body simulations for speeding up long-horizon episodic robot planning,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022.
- [4] M. Tuschler, J. Hörz, D. Driess, and M. Toussaint, “Deep 6-dof tracking of unknown objects for reactive grasping,” in *IEEE International Conference on Robotics and Automation*. IEEE, 2021.
- [5] H. Yamada, S. Takaoka, and S. Hirose, “A snake-like robot for real-world inspection applications (the design and control of a practical active cord mechanism),” *Advanced Robotics*, vol. 27, no. 1, pp. 47–60, 2013.
- [6] R. Feng, Y. Kim, G. Lee, E. K. Gordon, M. Schmittle, S. Kumar, T. Bhattacharjee, and S. S. Srinivasa, “Robot-assisted feeding: Generalizing skewering strategies across food items on a plate,” in *The International Symposium of Robotics Research*. Springer, 2019, pp. 427–442.
- [7] N. Marturi, M. Kopicki, A. Rastegarpanah, V. Rajasekaran, M. Adjigle, R. Stolkin, A. Leonardis, and Y. Bekiroglu, “Dynamic grasp and trajectory planning for moving objects,” *Autonomous Robots*, vol. 43, no. 5, pp. 1241–1256, 2019.
- [8] T. Wu, F. Zhong, Y. Geng, H. Wang, Y. Zhu, Y. Wang, and H. Dong, “Grasparl: Dynamic grasping via adversarial reinforcement learning,” *arXiv preprint arXiv:2203.02119*, 2022.
- [9] G. Du, K. Wang, S. Lian, and K. Zhao, “Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: a review,” *Artificial Intelligence Review*, vol. 54, no. 3, pp. 1677–1734, 2021.
- [10] W. Wan, K. Harada, and F. Kanehiro, “Planning grasps with suction cups and parallel grippers using superimposed segmentation of object meshes,” *IEEE Transactions on Robotics*, vol. 37:1, no. 1, pp. 166–184, 2020.
- [11] S. Jain and B. Argall, “Grasp detection for assistive robotic manipulation,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 2015–2021.
- [12] 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 *International Conference on Robotics and Automation*, 2019.
- [13] M. Q. Mohammed, K. L. Chung, and C. S. Chyi, “Review of deep reinforcement learning-based object grasping: Techniques, open challenges, and recommendations,” *IEEE Access*, vol. 8, pp. 178 450–178 481, 2020.
- [14] Y. Zhang, L. Ke, A. Deshpande, A. Gupta, and S. Srinivasa, “Cherry-picking with reinforcement learning,” *arXiv preprint arXiv:2303.05508*, vol. 15, 2023.
- [15] D. Morrison, P. Corke, and J. Leitner, “Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach,” *arXiv preprint arXiv:1804.05172*, 2018.
- [16] U. Viereck, A. Pas, K. Saenko, and R. Platt, “Learning a visuomotor controller for real world robotic grasping using simulated depth images,” in *Conference on robot learning*, 2017.
- [17] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, *et al.*, “Scalable deep reinforcement learning for vision-based robotic manipulation,” in *Conference on Robot Learning*, 2018.
- [18] B. Burgess-Limerick, C. Lehnert, J. Leitner, and P. Corke, “Eyes on the prize: Improved perception for robust dynamic grasping,” in *IROS*. IEEE, 2022.
- [19] S. Song, A. Zeng, J. Lee, and T. Funkhouser, “Grasping in the wild: Learning 6dof closed-loop grasping from low-cost demonstrations,” *IEEE Robotics and Automation Letters*, 2020.
- [20] I. Akinola, J. Xu, S. Song, and P. K. Allen, “Dynamic grasping with reachability and motion awareness,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2021.
- [21] B. Huang, T. Guo, A. Boularias, and J. Yu, “Interleaving monte carlo tree search and self-supervised learning for object retrieval in clutter,” in *International Conference on Robotics and Automation*, 2022.
- [22] C. Mayer, M. Danelljan, G. Bhat, M. Paul, D. P. Paudel, F. Yu, and L. Van Gool, “Transforming model prediction for tracking,” in *CVPR*, 2022, pp. 8731–8740.
- [23] B. Yan, X. Zhang, D. Wang, H. Lu, and X. Yang, “Alpha-refine: Boosting tracking performance by precise bounding box estimation,” in *CVPR*, 2021, pp. 5289–5298.
- [24] J. Revaud, P. Weinzaepfel, C. R. de Souza, and M. Humenberger, “R2D2: repeatable and reliable detector and descriptor,” in *NeurIPS*, 2019.
- [25] B. Wen and K. E. Bekris, “Bundletrack: 6d pose tracking for novel objects without instance or category-level 3d models,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2021.
- [26] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” *arXiv preprint arXiv:1707.06347*, 2017.
- [27] Y. Ono, E. Trulls, P. Fua, and K. M. Yi, “Lf-net: Learning local features from images,” *NeurIPS*, vol. 31, 2018.
- [28] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox, “Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes,” 2021.
- [29] S. R. Buss, “Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods,” *IEEE Journal of Robotics and Automation*, vol. 17, no. 1-19, p. 16, 2004.
- [30] 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,” 2021.
- [31] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, “On the continuity of rotation representations in neural networks,” in *CVPR*, 2019.
- [32] A. Allshire, M. Mittal, V. Lodaya, V. Makoviychuk, D. Makoviichuk, F. Widmaier, M. Wüthrich, S. Bauer, A. Handa, and A. Garg, “Transferring dexterous manipulation from gpu simulation to a remote real-world trifinger,” in *IROS*. IEEE, 2022.
- [33] Y. Narang, K. Storey, I. Akinola, M. Macklin, P. Reist, L. Wawrzyniak, Y. Guo, A. Moravszky, G. State, M. Lu, A. Handa, and D. Fox, “Factory: Fast contact for robotic assembly,” in *R:SS*, 2022.
- [34] S. Narvekar, B. Peng, M. Leonetti, J. Sinapov, M. E. Taylor, and P. Stone, “Curriculum learning for reinforcement learning domains: A framework and survey,” *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 7382–7431, 2020.
- [35] 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 *2015 international conference on advanced robotics (ICAR)*. IEEE, 2015.
- [36] S. Tyree, J. Tremblay, T. To, J. Cheng, T. Mosier, J. Smith, and S. Birchfield, “6-dof pose estimation of household objects for robotic manipulation: An accessible dataset and benchmark,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2022.
- [37] B. Wen, J. Tremblay, V. Blukis, S. Tyree, T. Müller, A. Evans, D. Fox, J. Kautz, and S. Birchfield, “Bundlesdf: Neural 6-dof tracking and 3d reconstruction of unknown objects,” in *CVPR*, 2023, pp. 606–617.