KARL: Kalman-Filter Assisted Reinforcement Learner for Dynamic Object Tracking and Grasping

Kowndinya Boyalakuntla Abdeslam Boularias Jingjin Yu

***Abstract*— We present Kalman-filter Assisted Reinforcement Learner (KARL) for dynamic object tracking and grasping over eye-on-hand (EoH) systems, significantly expanding such systems' capabilities in challenging, realistic environments. In comparison to the previous state-of-the-art, KARL (1) incorporates a novel six-stage RL curriculum that doubles the system's motion range, thereby greatly enhancing the system's grasping performance, (2) integrates a robust Kalman filter layer between the perception and reinforcement learning (RL) control modules, enabling the system to maintain an uncertain but continuous 6D pose estimate even when the target object temporarily exits the camera's field-of-view or undergoes rapid, unpredictable motion, and (3) introduces mechanisms to allow retries to gracefully recover from unavoidable policy execution failures. Extensive evaluations conducted in both simulation and real-world experiments qualitatively and quantitatively corroborate KARL's advantage over earlier systems, achieving higher grasp success rates and faster robot execution speed. Source code and supplementary materials for KARL will be made available at: https://github.com/arc-l/karl.**

I. Introduction

Humans, and animals in general, interact with the physical world through observing and handling everyday objects [1], which makes object tracking and manipulation arguably the *most fundamental skill* for physical intelligence. In robotics, autonomous grasping in *stationary* settings has been extensively studied [2], [3], typically using decoupled vision and manipulation sub-systems where the camera does not move with the manipulator. While effective for static tasks, this approach struggles in dynamic scenarios where objects move or become occluded. Real-world interactions, such as handovers, require continuous tracking and adaptive grasping, highlighting the need for more integrated solutions.

Through coupling vision and manipulation sub-systems, eye-on-hand (EoH) and eye-in-hand (EiH)[1] systems [4]–[7] can have both the camera and robot's hand follow the target object, avoiding the above-mentioned pitfalls of decoupled systems. EoH systems hold the promise of revolutionizing robotic manipulation across a diverse array of real-world applications. For example, in cluttered and unstructured environments [5], EoH systems make it possible to autonomously track and grasp moving objects with precision, transforming operations in industrial automation and advanced service robotics. EoH systems' ability to provide real-time visual feedback also makes them particularly suitable for teleoperation, shared autonomy [4], and prosthetic hand control [6]. Recently, *imitation learning* (IL) systems have largely converged to dual-arm setups where each arm's end-effector has a fixed camera sensor, e.g., [8], [9].

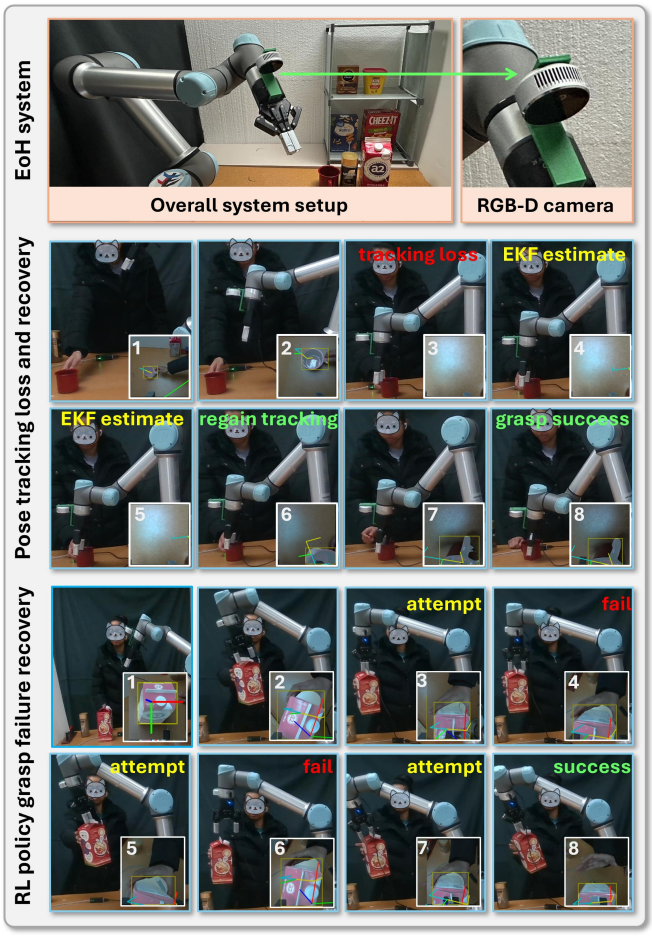

Fig. 1: [top row] The EoH system setup used for validating KARL's performance. [middle row] When tracking loss occurs due to an object temporarily going out-of-view or getting occluded, KARL maintains a EKF estimate of the object's pose to allow pose tracking recovery. [bottom row] KARL can detect and gracefully handle grasping failures and make multiple retries until success.

Fully leveraging EoH systems presents significant algorithmic and computational challenges, as rapid sensor feedback must be continuously processed for real-time control. While end-to-end IL systems [8], [9] perform well in stationary scenes, they require extensive data collection and computing power and struggle with dynamic object motion. Recently,

The authors are with the Department of Computer Science, Rutgers University, 08854 New Brunswick, USA. This work was supported in part by NSF awards IIS-1845888, IIS-2132972, and CCF-2309866.

[1]The difference between EoH and EiH systems are generally minor; in this work, we use eye-on-hand (EoH) to refer to both.

Huang et al. developed the EARL [7] reinforcement learning (RL) framework specifically for EoH systems. EARL can track/follow moving objects in real-time and find the right moment to grasp them. EARL achieves this by tightly integrating a perception sub-routine for 6D pose tracking and an RL sub-routine for robot/end-effector control. The pipeline allows EARL to handle a variety of challenging object motion patterns.

Whereas EARL shows promise in tackling dynamic manipulation settings, due to the need to handle conflicting constraints of EoH systems, it also comes with severe limitations (to be detailed shortly) that prevent it from being practical. To better leverage the capability of EoH systems and render them more practical, in this work, we propose a *Kalman-filter Assisted Reinforcement Learner* (KARL) framework that significantly expands EARL's capabilities. In particular, KARL brings the following key contributions:

- KARL introduces a Kalman filter layer that sandwiches between the perception and the RL control modules, allowing it to gracefully handle the case where rapid object movement causes the target object to disappear from the camera's field-of-view by maintaining an uncertain object pose estimate. EARL completely fails in such cases.
- EoH systems' strong coupling of camera and robot end-effector poses makes training RL policies that fully utilize the robot arm's motion ranges challenging. This forces EARL to have a limited workspace. KARL addresses this issue by introducing a sophisticated 6-step RL curriculum design to gradually expand the EoH system's reachability. Simultaneously, the majorly updated training schedule boosts the EoH system's speed by over 20.00%.
- Also by making changes to how the RL policy is trained and executed, KARL possesses the capability to recover from grasp failures and retry multiple times, as long as the target object can be continuously tracked. In contrast, EARL can only make a single grasp attempt.

In addition to the above-mentioned major upgrades, we also made cross-the-board updates to the original EARL's (perception and control) components. Notably, we replaced the object tracking module from BundleTrack [10] to FoundationPose [11] where applicable, leading to a few percentage points of success rate increase over the EARL baseline.

II. Related Work

Approaches to Robotic Grasping: Robotic grasping methods fall into two categories: analytic/geometric and data-driven. Analytic approaches rely on known object models or shape primitives, using geometry or physics-based analyses to plan stable grasps, often leveraging CAD models [12]–[14]. In contrast, learning-based methods generalize to novel objects by training on large grasp datasets [2]. These include deep networks that sample and rank grasp candidates from visual input [15] and reinforcement learning (RL) policies for grasp synthesis [16], [17]. Advances in algorithms and data, such as grasping datasets and simulations, have significantly improved learning-based grasping success.

Eye-on-Hand (EoH) vs. Static Camera Setups: Most vision-based manipulators use fixed cameras [18], [19], simplifying perception but introducing occlusion and coverage limitations. Multi-camera setups help but add complexity. Eye-on-Hand (EoH) architectures mount the camera on the robot's wrist or end-effector [4]–[7], ensuring continuous object tracking and eliminating blind spots. While beneficial in dynamic scenarios, the tight coupling introduces planning constraints that can limit system motion.

Dynamic Grasping: demands real-time adaptation to unpredictable object motion. Early approaches relied on feedback control or fast vision networks, such as Morrison et al.'s Generative Grasp CNN for rapid closed-loop grasping in mildly dynamic scenes [20]. Other methods continuously re-planned trajectories using multi-camera tracking [21] or motion prediction [22], which struggled with erratic movements and relied on fixed cameras, limiting real-time pursuit.

Visual Servoing (VS) based grasping using EoH Camera is a well-established manipulation framework that uses visual feedback to control a robot's motion and perform closed-loop, real-time grasping [23]. One of the main approaches within VS is Position-Based Visual Servoing (PBVS), which estimates the full 6D pose of an object and directly computes the control commands in task space to drive the robot to the desired configuration. PBVS has been used for dynamic grasping of moving objects [24] with unpredictable motion, although limited to slow object movement scenarios. *Deep RL* has emerged as a promising alternative, with Song et al. [25] training an RL policy that maps wrist-camera images to Q-values, enabling 6-DoF grasping in clutter. However, it was constrained to discrete actions and slower object motions. Many prior methods imposed constraints on grasp orientation or speed to ensure reliability [25], [26].

III. Preliminaries

A. Problem Formulation

We aim to endow EoH systems with the ability to dynamically track and grasp moving target objects. Our primary objective is to enable EoH systems to perform dynamic grasping in full six degrees of freedom (i.e., within the SE(3) space) for a moving object whose motion is a priori unknown. We make minimal assumptions about the target—namely, that it is a graspable rigid body. No assumptions are made regarding the object's shape, identity, or specific motion profile; the object may move freely within the robot's reachable workspace. Successfully solving this problem requires addressing two interrelated challenges:

- Object Tracking and Pursuit: The system must continuously estimate the 6D pose of the moving object and actively control the robot to keep the target within view.
- Grasp Execution: Simultaneously, the robot must effectively approach and execute a stable grasp on the target object despite its unpredictable motion.

A grasp is deemed successful if the system is able to securely seize and lift the target object.

Besides simulation-based evaluation, we evaluate an EoH system with the Universal UR-5e (a 6-DOF robot) with a

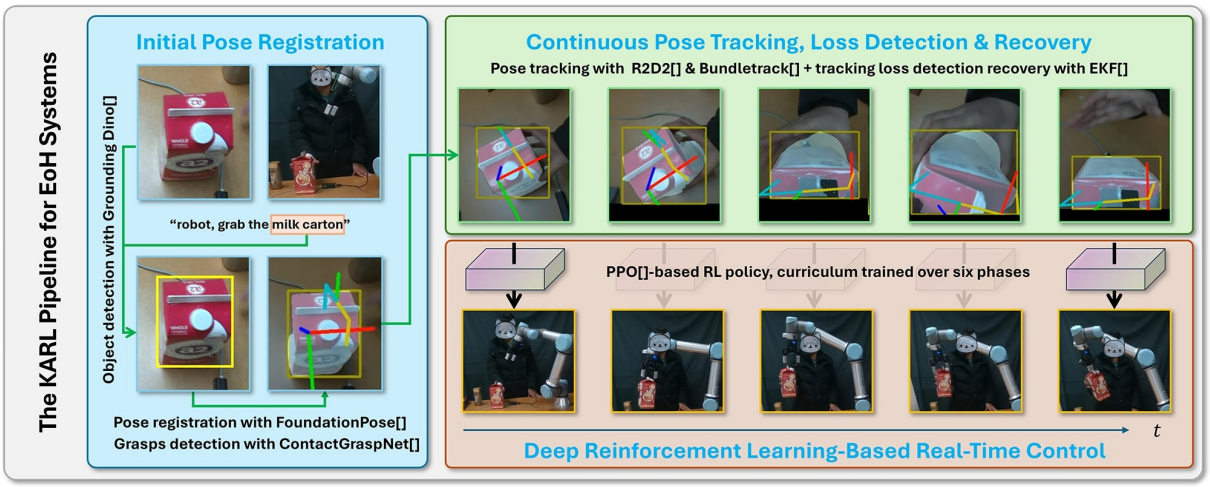

Fig. 2: The KARL framework for EoH systems. It has three interconnected modules for initial pose estimation, continuous pose tracking, and RL-based continuous robot control. In the beginning, based on user input KARL automatically detects the target object, estimates its initial pose, and computes a set of candidate grasps. It then enters the main pose tracking and control loop. In the process, KARL also keeps monitoring and recovering from potential tracking loss (due to sudden quick object movements or temporary occlusion) and possible grasping failure.

mounted Intel RealSense L515 RGB-D camera. A Robotiq 2F-85 two-finger gripper serves as the end-effector.

B. Eye-on-hand Reinforcement Learner (EARL) Overview

Eye-on-Hand Reinforcement Learner (EARL) [7] integrates perception and RL-based tracking and grasping for EoH systems, which eliminates workspace constraints and occlusion issues typical to external camera setups. In an EoH system, an RGB-D sensor mounted on the robotic arm can track objects in real time. In EARL, to register the 6D pose of the object, a few 3D points are selected manually on the point cloud of the scene, which is then fed to the tracking module that uses R2D2 [27] and BundleTrack [10] to maintain keyframe memory and performs 6D pose estimation updates in real-time. In EARL, a target object is manually selected by drawing a bounding box. It leverages ContactGraspNet [28] to generate multiple 6-DoF grasp candidates. These candidates are stored in a grasp memory pool, allowing EARL to select the most stable and reachable grasp based on the current pose of the object.

EARL trains a Proximal Policy Optimization (PPO) [29] RL policy to generate joint velocity commands and gripper actions, given object pose input from the perception subsystem and the robot's joint states. A three-step curriculum is designed to simultaneously track the object and drive the end-effector's motion. To achieve robust grasping, reward is carefully shaped to: (1) **maintain object tracking** by assigning a penalty whenever the target object moves out of view, (2) **avoid collision** through a penalty discouraging pre-grasp collisions while ensuring the gripper remains close, (3) **encourages proper gripper alignment and approach angles** along multiple axes (y,z) to maximize grasp stability, and (4) **execute successful grasps** by rewarding the agent for closing the gripper at the optimal moment.

Limitations. While EARL's three-stage RL curriculum progressively increases task difficulty, the initial stage imposes overly restrictive starting robot joint positions, restricting object pose diversity. In later stages, strong penalties are imposed to prevent object tracking loss and collision, further reducing the allowed object's motion range and speed. Quantitatively, EARL's effective workspace is confined to a $40 \times 40 \times 40\ cm^3$ volume and needs a longer time for its RL policy to run (e.g., > 15 seconds to reach 90% success rate due to slower supported robot speed. Moreover, EARL lacks mechanisms to recover from RL policy failures, leading to 100% failures if the object quickly moves out of the camera's view or a grasp attempt fails. To fully unlock the potential of EoH systems in dynamic manipulation settings, these severe limitations must be overcome.

IV. Proposed System

KARL has three main, interconnected modules (see Fig. 2 for an illustration). The perception sub-system of KARL contains two: an initial pose registration module and a continuous pose tracking/recovery module. The third module is the RL-based robot controller. Algorithm 1 outlines the method.

A. Initial Pose Registration

KARL takes in requests in the form of simple natural language inputs, e.g., "robot, grab that *milk carton*." Upon receiving a request, KARL prompts Grounding DINO [30] to automatically extract the target object from images taken from the hand-mounted RGB-D camera. At this point ($t = 0$), two additional tasks are performed: initial pose estimation/registration and candidate grasp generation. For the former, FoundationPose [11] is invoked to perform the pose estimation of the target object using RGB-D input, object mask and its textured mesh. If the CAD model of the object

Algorithm 1 Active Pose Estimation and Object Grasping

Initialization:
$t \leftarrow 0$, $\mathcal{I}_t, \mathcal{D}_t$: RGB, Depth images, $\mathcal{C}_t$: Camera pose
$\mathcal{G}_t^W, \mathcal{G}_t^C$: Gripper pose in world & camera frames
$\mathcal{O}_{\text{pool}}^W, \mathcal{O}_{\text{pool}}^C$: Grasp pose pool in world & camera frames
$\mathcal{O}_t^C$: Grasp pose in camera frame
$\mathcal{T}_t^W, \mathcal{T}_t^C$: Object pose in world & camera frames
Read initial RGB-D image: $(\mathcal{I}_0, \mathcal{D}_0)$
Read object name $\mathcal{L}$ (e.g., "coffee cup")
$\mathcal{B}_0 \leftarrow \text{GroundingDINO}(\mathcal{I}_0, \mathcal{D}_0, \mathcal{L})$
$\mathcal{M}_0 \leftarrow \text{2DTracker}_{\text{init}}(\mathcal{I}_0, \mathcal{B}_0)$
$\mathcal{K} \leftarrow \text{CAD\_Model}(L)$
$\mathcal{T}_0^C \leftarrow \text{FoundationPose}(\mathcal{I}_0, \mathcal{D}_0, \mathcal{M}_0, \mathcal{K})$
$\mathcal{O}_{\text{pool}}^W \leftarrow \text{ContactGraspNet}(\mathcal{I}_0, \mathcal{D}_0)$
$\mathcal{O}_{\text{pool}}^C \leftarrow \text{TransformToCameraFrame}(\mathcal{C}_0, \mathcal{O}_{\text{pool}}^W)$
$\mathcal{O}_0^C \leftarrow \text{SelectBestGrasp}(\mathcal{G}_0^C, \mathcal{O}_{\text{pool}}^C)$
$\mathcal{T}_{\text{obj\_grasp}}^C \leftarrow \text{ComputeTransform}(\mathcal{T}_0^C, \mathcal{O}_0^C)$
Initialize EKF with state $\mathbf{x}_0$, covariance matrix $\mathbf{P}_0$
use_foundationpose $\leftarrow$ **False**
handover $\leftarrow$ **False**
while handover = **False** and $t < t_{\max}$ **do**
  $t \leftarrow t + 1$
  Read new RGB-D frame: $(\mathcal{I}_t, \mathcal{D}_t)$
  $\mathcal{M}_t, \mathcal{B}_t \leftarrow \text{2DTracker}_{\text{track}}(\mathcal{I}_t)$
  if $\mathcal{M}_t \neq$ **None then**
    if use_foundationpose = **True then**
      $\mathcal{T}_t^C \leftarrow \text{FoundationPose}(\mathcal{I}_t, \mathcal{D}_t, \mathcal{M}_t, \mathcal{K}, \mathcal{T}_{t-1}^C)$
      use_foundationpose = **False**
    else
      $\mathcal{T}_t^C \leftarrow \text{BundleTrack\_R2D2}(\mathcal{I}_t, \mathcal{D}_t, \mathcal{M}_t, \mathcal{T}_{t-1}^C)$
    $\mathcal{T}_t^W \leftarrow (\mathcal{C}_t * \mathcal{T}_t^C)$
    $\text{EKF.update}(\mathcal{T}_t^W)$
  else
    use_foundationpose $\leftarrow$ **True**
    $\mathcal{T}_t^W \leftarrow \text{EKF.predict}()$
    $\mathcal{T}_t^C \leftarrow (inverse(\mathcal{C}_t) * \mathcal{T}_t^W)$
  $\mathcal{O}_t^C \leftarrow \mathcal{T}_{\text{obj\_grasp}}^C \cdot \mathcal{T}_t^C$
  $\mathcal{O}_t^C \leftarrow \text{SelectBestGrasp}(\mathcal{G}_t^C, \mathcal{O}_t^C)$
  Move robot gripper toward $\mathcal{O}_t^C$
  if $\text{Align}(\mathcal{G}_t^C, \mathcal{O}_t^C)$ = **True** and $\text{Dist}(\mathcal{G}_t^C, \mathcal{O}_t^C) < \epsilon$ **then**
    Grasp the object and lift
    if Grasping Failed **then**
      Invoke RL Policy and continue
    else
      Grasping Success!
      handover $\leftarrow$ **True**

is not available, we utilize BundleSDF [31] by feeding a sequence of reference RGB-D images of the target and mask of the first frame in the sequence. For the latter, ContactGraspNet [32] is applied directly to the (segmented) depth data to produce (a few) candidate grasps. Formally, given RGB-D, mask and CAD model as input $(\mathcal{I}_0, \mathcal{D}_0, \mathcal{M}_0, \mathcal{K})$, the initial pose registration phase yields object's 6D pose $\mathcal{T}_0^{\mathcal{C}}$, and grasp pose candidates $\mathcal{O}_{\text{pool}}^{\mathcal{W}}$ in the camera $\mathcal{C}$ and world $\mathcal{W}$ frames of reference respectively.

B. Real-Time Pose Tracking and Tracking Loss Recovery

1) Continuous Pose Tracking Before Tracking Loss: Starting from initial pose $\mathcal{T}_0^{\mathcal{C}}$ given by FoundationPose [11], we let the pose tracker (R2D2 [27] + BundleTrack [10] from [7]) track the object's pose $\mathcal{T}_t^{\mathcal{C}}$ in real-time. Ideally, FoundationPose can be used to also track the object, however in our EoH setup both the camera and target object move randomly in 3D space. As a result, we found that the tracker in FoundationPose often becomes unstable. We attribute this to its pose refiner model encountering object trajectories (from our setup) that were likely absent from its training data. Finetuning FoundationPose for object trajectories in our setup can be explored in the future to streamline the design.

2) Tracking Loss Detection and Recovery: During the pose tracking process, an Extended Kalman Filter (EKF) [33] is maintained to provide a secondary pose estimate ($\mathcal{T}_t^{\mathcal{W}}$, $\mathcal{W}$ denotes the world coordinate frame.). We use a relatively standard EKF implementation to track the target pose, essentially using $x_{t+1} = x_t + v_t \Delta t$ for the process update (here, $x_k = \mathcal{T}_t^{\mathcal{W}}$ and v_t are the proper linear and rotational velocities computed using known pose history $\mathcal{T}_{0:t-1}^{\mathcal{W}}$). When an object mask can be reliably detected, this EKF estimate is not used for downstream tasks. When at a time t, the target object leaves the FOV of the camera or gets occluded, we lose the mask ($\mathcal{M}_t$) of the object and this signals KARL to switch to using EKF estimated pose.

During the period where the tracked pose is provided by the EKF, because no new pose observation is available, the last known observation is used to update the EKF, which makes sense for both the case when the object moves out of view and when it is occluded by another object. This pushes the EKF's estimate to grow more and more uncertain over time, which is expected. The growing uncertainty naturally drives the control policy (to be detailed in Sec. IV-C) to behave more conservatively, which is the desired behavior. In particular, we sample from the uncertain pose estimates a pose that maximizes the expected view of the uncertain regions so that the camera can get the object back in view. We note that, as with any recovery method, the duration of the tracking loss cannot be very large. Otherwise, the RL policy can fail. BundleTrack works when the pose changes between two frames are relatively small. When the object comes back to view (at some time t'), BundleTrack generally cannot recover the pose of the object. When this happens, the pose of the object must be *re-initialized* for tracking. As the mask $\mathcal{M}_{t'}$ of the object is recovered by the 2DTracker [34], we invoke FoundationPose again to quickly re-register the pose only once. The new pose is denoted as $\mathcal{T}_{t'}^{\mathcal{C}}$. We convert this pose to the world coordinate frame as $\mathcal{T}_{t'}^{\mathcal{W}}$, and continuously recompute the object pose in camera coordinates as the robot continues to reach and grasp. Since the robot is moving and hence the camera is also moving, pose estimation needs to be both quick and accurate. We feed FoundationPose, the current mean of pose distribution from the EKF ($\mathcal{T}_{EKF(t,\mu)}^{\mathcal{C}}$), after converting to camera coordinates), and reduce the number of object 6D pose hypotheses to under 60. This estimation is both fast and accurate.

C. RL-Based Robot Controller

Similar to [7], we train the RL agent, a multi-layer perception (MLP) with PPO [29],

$$[\mathcal{Q}_v^{t+1}, b^{t+1}] = MLP(\mathcal{P}_g^t, \mathcal{E}_g^t, \Delta_g^t, \mathcal{Q}_p^t, \mathcal{Q}_v^t, b^t),$$

where Q_v is the joint velocities, b is the gripper open/close action, $\mathcal{P}_g$ are gripper keypoints, $\mathcal{E}_g = \mathcal{P}_g - \mathcal{P}_o$ is the posi-

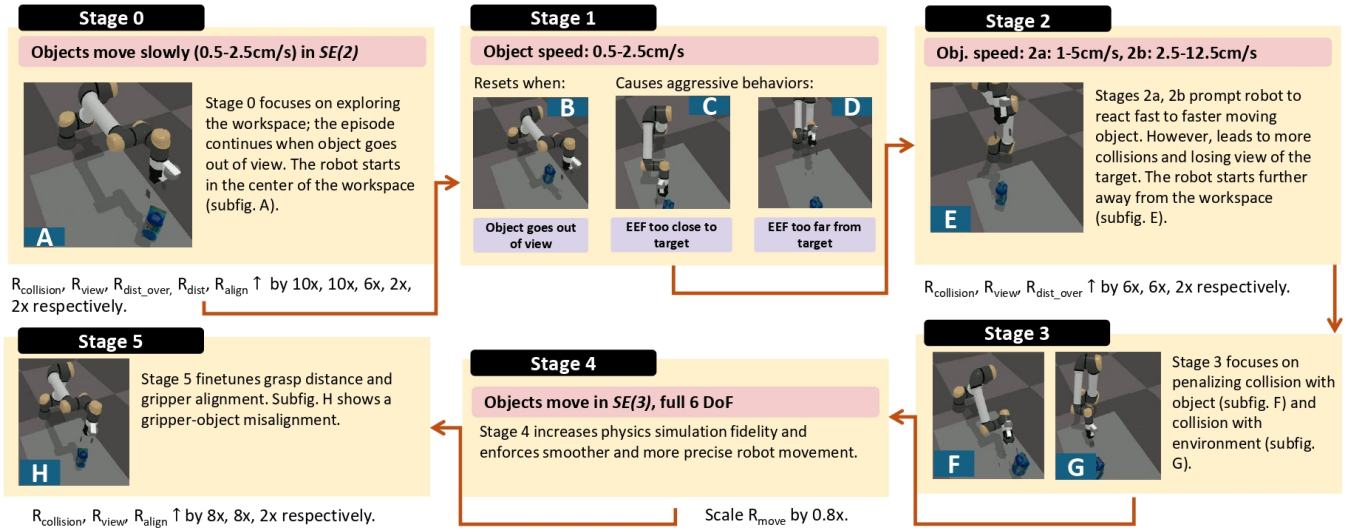

Fig. 3: The six stage curriculum used for training KARL's MLP network for RL-based robot control, using PPO [29].

tional error between the gripper and object grasp keypoints (to be minimized for a successful grasp), $\Delta_g = \mathcal{P}_o - \mathcal{P}'_o$ reflects the change in object grasp keypoints between consecutive frames, and $\mathcal{Q}_p$ denotes joint positions, normalized based on joint limits.

1) Reward Shaping and Curriculum Design: To accommodate the EoH system's intrinsic constraints and support the desired dynamics behavior, a reward structure and curriculum of similar complexity are required. KARL settled on an overall reward of the following form:

$$\begin{aligned}\mathcal{R}^{(i)} = \mathcal{R}_{grasp} + \lambda^{(i)}_{dist}\mathcal{R}_{dist} + \lambda^{(i)}_{dist\_over}\mathcal{R}_{dist\_over} \\ +\lambda^{(i)}_{align}\mathcal{R}_{align} + \lambda^{(i)}_{collision}\mathcal{R}_{collision} + \lambda^{(i)}_{view}\mathcal{R}_{view} \\ +\lambda^{(i)}_{gripper}\mathcal{R}_{gripper\_penalty} + \lambda^{(i)}_{move}\mathcal{R}_{move},\end{aligned} \tag{1}$$

where terms denote *successful grasping reward*, *distance reduction reward*, *not staying overly distant reward*, *gripper alignment reward*, *collision avoidance reward*, *object visibility reward*, *premature gripper closure avoidance reward*, and *stable gripping reward*, in that order. For more details and rationale of the reward terms, see [7]. At stage i in the curriculum, the weight coefficients $\lambda^{(i)}$ are adjusted to guide the learning process progressively. The reward structure discourages inefficiencies while reinforcing effective grasping behavior.

Curriculum Design We settle down on a six-stage curriculum (see Fig. 3) with smoother performance-adaptive transitions between stages and improved reward shaping, with earlier stages being more conservative. In stage 0, the robot starts in the workspace center with closely sampled target object poses. The target object moves only in $SE(2)$ until stage 3. Stage 0 lets the robot freely explore without stopping it, even if the object goes out of view. In stage 1, object visibility is enforced more aggressively (losing visibility triggers environment reset), ensuring the robot learns to maintain sight of the object. Instead of gradually increasing object speeds in training, we accelerate the object movements earlier in KARL, to promote faster adaptation to higher curriculum levels with faster object movements. The system also increases rewards for firm grasps upon correct alignment. In summary, for early stages (0-1), high penalties are enforced for losing object visibility and collisions and moderate rewards for grasping. Mid-stages (2-4) focus on balancing speed and safety, with increasing importance on precise grasping (in stage 4, the robot operates with high control frequency to ensure precise robot movements). In the later stages, alignment and action penalties have been fine-tuned to discourage unnecessary movements when the robot is about to close the gripper near the target, improving efficiency and reducing grasp failures due to misalignment. The curriculum was trained in parallel over 8192 Isaac Gym [35] environments on a RTX 4090 GPU.

2) Grasp Failure Detection and Recovery: During real-world experimentation, the object may slip from the gripper due to factors such as inadequate grip force, object inertia, or external disturbances. To address this issue, we introduced a grasp failure detection and recovery mechanism within the control loop of the real robot policy. Instead of assuming a successful grasp upon closing the gripper, the system continuously monitors the gripper's position to determine whether the object has been securely held. If the gripper's position remains above a predefined threshold (indicating that the object has not been grasped), the robot immediately reopens the gripper and attempts to reposition itself for another grasp. We modified the control loop to: (1) continuously provide new pose observations by re-evaluating the target object's position and orientation in real time, (2) invoke the RL policy iteratively to adapt the robot's motion and attempt a re-grasp until success, and (3), introduce a brief stabilization delay before lifting the object to prevent premature slippage. Overall, this ensures more reliable object manipulation in dynamic environments.

V. Evaluation

To comprehensively evaluate our method, we conducted six experiments—five in simulation (**A-E**) and one on a

physical robot (**F**), with varying complexity through:

- **A**: Faster object speeds.
- **B**: Stricter grasping task time constraints.
- **C**: Expanded target workspace.
- **D**: Tracking loss scenarios.
- **E**: Combined effect of A–D in an ablation study.
- **F**: Real-world evaluation using a UR5e robotic arm.

A. Target Object Motion

The target object, initialized at a workspace boundary, may follow the following motions:

- **Linear (Regular)**: Constant speed (0–5 cm/s), no rotation. Used in 70% of trials in Experiments B, C.
- **Linear (Fast)**: Higher speed (up to 15 cm/s), zero rotational velocity.
- **Random**: Translational velocity (0–5 cm/s) with random rotations ($0°$–$14.5°$) along X, Y, and Z axes. Used in 30% of trials in Experiments B and C.
- **Disruptive**: Sudden velocity spikes ($>$ 30 cm/s), causing temporary tracking loss.

B. Higher Object Speeds

For increasing linear motion speeds (3 cm/s to 15 cm/s), KARL maintained a near-perfect success rate (>99%) with zero collisions for speeds up to 9 cm/s. Overall, KARL outperformed EARL, achieving a 92.58% success rate. While EARL struggled with tracking failures at higher speeds, KARL's EKF module prevented these issues, ensuring robust object tracking and grasp execution. (See Table I).

TABLE I: Performance comparison of EARL and KARL under different target object speeds (1000 Isaac Gym Envs)

	Policy	Target Object Speed (cm/s)					
		3	6	9	12	15	Random
Success (↑)	EARL	94.38	95.40	95.83	80.22	64.64	89.21
	KARL	**99.21**	**99.33**	**99.10**	**82.81**	**67.34**	**92.58**
Time Limit (35 sec) (↓)	EARL	**0.45**	1.35	**1.13**	**1.01**	**0.68**	**0.79**
	KARL	0.79	**0.67**	0.90	8.65	14.08	3.82
Collision (↓)	EARL	0.11	0.22	0.11	**3.93**	**5.63**	**1.12**
	KARL	**0.00**	**0.00**	**0.00**	8.54	18.58	3.60
Tracking Failure (↓)	EARL	5.06	3.03	2.93	14.83	29.05	8.88
	KARL	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
Total Failure (↓)	EARL	5.62	4.60	4.17	19.78	35.36	10.79
	KARL	**0.79**	**0.67**	**0.90**	**17.19**	**32.66**	**7.42**

C. Decreased Task Time limits

We evaluated the grasping success of KARL under progressively tighter time constraints, ranging from 35 to 5 seconds (see TABLE II). KARL achieves 91.83% success in 15 seconds, outperforming EARL, which reaches a similar performance in 20 seconds, a 5-second advantage. Notably, 99.9% of KARL's successful grasps occurred within 25 seconds, consistently surpassing EARL. Interestingly, neither KARL nor EARL succeeded in completing the grasping task within 5 seconds. However, beyond this extreme constraint, KARL also exhibited greater robustness in time-constrained grasping scenarios.

TABLE II: Performance comparison of EARL and KARL under different time limits (1000 Isaac Gym Envs)

Metric (%)	**EARL (%)**	**KARL (Ours) (%)**
Success ($t \leq 35s$) (↑)	95.16	**96.89**
Success ($t \leq 30s$) (↑)	93.61	**96.34**
Success ($t \leq 25s$) (↑)	93.54	**96.17**
Success ($t \leq 20s$) (↑)	91.91	**94.14**
Success ($t \leq 15s$) (↑)	84.46	**91.83**
Success ($t \leq 10s$) (↑)	21.03	**26.70**
Success ($t \leq 5s$) (↑)	**0.00**	**0.00**
Timeout ($> 35s$) (↓)	2.70	**2.32**
Collision (↓)	**0.79**	**0.79**
Tracking Failure (↓)	1.35	**0.00**
Total Failure (↓)	4.84	**3.11**

TABLE III: Performance comparison of EARL and KARL on the enlarged workspace $W_{KARL} = W_{EARL} + W_A + W_B$ (1000 Isaac Gym Envs)

Metric (%)	Policy	Target Object Workspace			
		W_A	W_B	W_{EARL}	W_{KARL}
Success (↑)	EARL	78.01	84.24	95.09	81.67
	KARL	**97.16**	**97.41**	**96.89**	**90.47**
Timeout (35 sec) (↓)	EARL	11.45	3.28	2.77	12.04
	KARL	**2.03**	**1.85**	**2.32**	**7.10**
Collision (↓)	EARL	1.53	1.67	**0.79**	**2.27**
	KARL	**0.81**	**0.74**	**0.79**	2.43
Tracking Failure (↓)	EARL	9.01	10.81	1.35	4.02
	KARL	**0.00**	**0.00**	**0.00**	**0.00**
Total Failure (↓)	EARL	21.99	15.76	4.91	18.33
	KARL	**2.84**	**2.59**	**3.11**	**9.53**

D. Expanded Workspace

We expanded EARL's workspace and doubled the operational volume by adding two new partitions. We expanded EARL's target object workspace W_{EARL} by considering two adjacent workspaces W_A and W_B. KARL's RL curriculum roughly doubles the workspace to (40 x 70 x 40 cm^3). EARL struggled, while KARL sustained a $\sim 97\%$ success rate across all partitions. In the expanded workspace, where the effective range of horizontal motion is nearly doubled, KARL, as expected, experienced increased timeouts but still outperformed EARL across all metrics. This proves its adaptability in larger, more challenging workspaces (refer Table III).

E. Tracking Loss and Recovery

Beyond the workspace expansion and the tighter time constraints, we further assessed KARL's resilience to objects exiting the camera's view—one of the unavoidable circumstances in dynamic grasping. TABLE IV shows that while EARL performs well under regular motion (94.60% success), its success drops to 7.10% in tracking loss scenarios due to its inability to recover lost object poses. In contrast, KARL, leveraging EKF, accurately predicts object motion, achieving 94.30% success even under disruptions. The robot follows EKF's predicted path to the target. However, EKF's update size cannot be tuned for different object speeds, and the robot might reach the end of the episode before reaching the object, as observed in KARL's timeout rate. Despite this, KARL has

TABLE IV: Performance comparison of EARL and KARL on regular and tracking loss scenarios (1500 Isaac Gym Envs)

Metric (%)	Policy	Scenarios (#)			
		Line (700)	Random (300)	Tracking Loss (500)	Overall (1500)
Success (↑)	EARL	94.60	96.00	7.10	65.72
	KARL	**96.10**	**98.50**	**94.30**	**95.98**
Timeout (35 sec) (↓)	EARL	3.70	0.60	**0.00**	**1.85**
	KARL	**3.10**	**0.50**	5.20	3.28
Collision (↓)	EARL	1.50	2.40	1.80	1.17
	KARL	**0.80**	**1.00**	**0.50**	**0.74**
Tracking Failure (↓)	EARL	1.50	1.00	91.10	31.27
	KARL	**0.00**	**0.00**	**0.00**	**0.00**
Total Failure (↓)	EARL	5.40	4.00	92.90	34.29
	KARL	**3.90**	**1.50**	**5.70**	**4.02**

zero tracking failures and consistently outperforms EARL across all scenarios.

TABLE V: Ablation study (large workspace, high object speeds, tracking loss scenarios) (1000 Isaac Gym Envs)

Policy	Metric (%)				
	Success (↑)	Timeout(↓)	Collision(↓)	Tracking Failure(↓)	Total Failure(↓)
EARL	66.29	7.84	2.54	23.32	33.71
Old Cur.	76.70	15.21	8.08	**0.00**	23.30
No EKF	72.62	**4.00**	**2.27**	21.11	27.38
KARL	**83.46**	10.46	6.08	**0.00**	**16.54**

F. Ablation Study on KARL

We now assess KARL on a demanding scenario where all the three scenarios in **Experiments B, C, D** are combined. 46.67% of the time, the object moves in linear motion (fast) and is forced out of view with a 20.00% probability (disruptive object motion). We did not choose to further increase the out-of-view probability in order to keep the simulation realistic. For the remaining 33.33% of trials, the object follows a random trajectory, where it may exit the camera's view inadvertently. To assess KARL's key components, we conducted an ablation study by systematically removing key components—EKF (for tracking loss recovery) and the new curriculum (for better grasp adaptation)—to isolate their contributions. As shown in Table V, KARL without the new curriculum (but with EKF) eliminates tracking failures and improves upon EARL by a 10.00% higher success rate, but timeouts remain a challenge. This is partially due to EKF update size limitations and partially due to the lack of imprecise gripper alignment avoidance in the previous curriculum with the object's grasp pose. KARL without EKF (but with the new curriculum) reduces gripper misalignment issues, decreasing timeouts and improving the success rate by 6.00% over EARL. However, without EKF, tracking failures remain high, only slightly lower than EARL. KARL achieves the highest success rate (83.46%), improving upon EARL by 17.17%, with zero tracking failures and a significantly lower overall failure rate. These results reinforce the importance of both EKF for robust object tracking recovery and the new curriculum for improved grasp alignment and success, demonstrating that their combined effect enables KARL to outperform EARL across all difficulty levels.

TABLE VI: Real-world comparison of EARL and KARL

Policy	Success Rate (%)		Premature Grasping Rate (%)	
	Regular	Complex	Regular	Complex
EARL	60.00	53.30	-	-
KARL	**93.30**	**80.00**	**0.00**	**6.60**

G. Real Robot Experiments

To validate KARL's real-world performance, we conducted grasping experiments using a UR5e robotic arm equipped with a 2F-85 Robotiq Gripper. Both KARL and EARL were tested in 30 trials, totaling 60 trials across different scenarios:

- **Regular Scenes (15 trials):** Only the target object was present, with no obstacles.
- **Complex Scenes (15 trials):** The target moved behind obstacles such as shelves or walls, requiring the robot to follow and maintain visibility, highlighting the advantage of an Eye-on-Hand (EoH) system. However, explicit collision avoidance was not in place, introducing the potential for unintended contact.

The target object's motion followed one of three patterns, with 14 trials with linear (regular) motion, 6 trials with random motion, and 10 trials where tracking was intentionally lost, forcing the robot to recover the object's location. As shown in Table VI, KARL outperforms EARL in both regular and complex scenarios. In obstacle-free scenes, KARL achieves an impressive 93.30% success rate, compared to 60.00% for EARL. In complex environments, KARL maintains a strong 80.00% success rate, whereas EARL drops to 53.30%. EARL falters mainly due to its inability to recover objects that move out of view. One observed limitation of KARL is premature grasping—in 6.60% of complex scene trials, KARL attempted to grasp an estimated pose from its EKF-based predictions before reaching the actual object. Unlike in simulation, where successful grasping is confirmed through object contact, our real-world setup lacks this feature, leading to occasional early grasp attempts.

These results reinforce the findings from the simulation studies—KARL's EKF-based tracking and adaptive grasping curriculum provide significant advantages, especially in dynamic and occlusion-heavy environments. While minor real-world limitations exist, KARL consistently outperforms EARL across all tested conditions, demonstrating superior robustness and adaptability in real-world grasping tasks.

VI. Conclusion

In this work, we introduced KARL—a Kalman-filter Assisted Reinforcement Learner that significantly enhances eye-on-hand (EoH) systems for dynamic object tracking and grasping. By integrating an Extended Kalman Filter, KARL

maintains continuous 6D pose estimates even when the target temporarily exits the camera's view or moves unpredictably. Coupled with a novel six-stage reinforcement learning curriculum, KARL doubles the operational workspace and improves grasp performance through rapid recovery and multiple retry attempts. Evaluations in simulated and real-world environments demonstrate that KARL outperforms previous methods in grasp success rate, and execution speed. KARL's combined state estimation and adaptive curriculum allow it to handle a broader range of object speeds and motion patterns, enhancing overall reliability in dynamic scenarios. Future work will focus on further refining responsiveness under extreme motions, rendering KARL an essential tool for versatile robotic manipulation.

REFERENCES

[1] M. T. Mason, "Toward robotic manipulation," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, no. 1, pp. 1–28, 2018.

[2] 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.

[3] Z. Pan, A. Zeng, Y. Li, J. Yu, and K. Hauser, "Algorithms and systems for manipulating multiple objects," *IEEE Transactions on Robotics*, vol. 39, no. 1, pp. 2–20, 2022.

[4] M. Yan, I. Frosio, S. Tyree, and J. Kautz, "Sim-to-real transfer of accurate grasping with eye-in-hand observations and continuous control," *arXiv preprint arXiv:1712.03303*, 2017.

[5] L.-W. Cheng, S.-W. Liu, and J.-Y. Chang, "Design of an eye-in-hand smart gripper for visual and mechanical adaptation in grasping," *Applied Sciences*, vol. 12, no. 10, p. 5024, 2022.

[6] F. Vasile, E. Maiettini, G. Pasquale, A. Florio, N. Boccardo, and L. Natale, "Grasp pre-shape selection by synthetic training: Eye-in-hand shared control on the hannes prosthesis," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 13 112–13 119.

[7] B. Huang, J. Yu, and S. Jain, "Earl: Eye-on-hand reinforcement learner for dynamic grasping with active pose estimation," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 2963–2970.

[8] Z. Fu, T. Z. Zhao, and C. Finn, "Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation," *arXiv preprint arXiv:2401.02117*, 2024.

[9] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter *et al.*, "π_0: A vision-language-action flow model for general robot control," *arXiv preprint arXiv:2410.24164*, 2024.

[10] B. Wen and K. Bekris, "Bundletrack: 6d pose tracking for novel objects without instance or category-level 3d models," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 8067–8074.

[11] B. Wen, W. Yang, J. Kautz, and S. Birchfield, "Foundationpose: Unified 6d pose estimation and tracking of novel objects," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 17 868–17 879.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] Y. Zhang, L. Ke, A. Deshpande, A. Gupta, and S. Srinivasa, "Cherry-picking with reinforcement learning," *arXiv preprint arXiv:2303.05508*, vol. 15, 2023.

[18] M. Tuscher, 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.

[19] 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.

[20] 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.

[21] N. Marturi, M. Kopicki, A. Rastegarpanah, V. Rajasekaran, M. Adjigble, 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.

[22] 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.

[23] J. Wu, Z. Jin, A. Liu, L. Yu, and F. Yang, "A survey of learning-based control of robotic visual servoing systems," *Journal of the Franklin Institute*, vol. 359, no. 1, pp. 556–577, 2022.

[24] B. Burgess-Limerick, C. Lehnert, J. Leitner, and P. Corke, "An architecture for reactive mobile manipulation on-the-move," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 1623–1629.

[25] 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*, vol. 5(3), pp. 4978–4985, 2020.

[26] 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.

[27] J. Revaud, C. De Souza, M. Humenberger, and P. Weinzaepfel, "R2d2: Reliable and repeatable detector and descriptor," *Advances in neural information processing systems*, vol. 32, 2019.

[28] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox, "Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 13 438–13 444.

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

[30] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su *et al.*, "Grounding dino: Marrying dino with grounded pre-training for open-set object detection," in *European Conference on Computer Vision*. Springer, 2024, pp. 38–55.

[31] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 606–617.

[32] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox, "Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes," 2021.

[33] K. Fujii, "Extended kalman filter," *Refernce Manual*, vol. 14, p. 41, 2013.

[34] C. Mayer, M. Danelljan, G. Bhat, M. Paul, D. P. Paudel, F. Yu, and L. Van Gool, "Transforming model prediction for tracking," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 8731–8740.

[35] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa *et al.*, "Isaac gym: High performance gpu-based physics simulation for robot learning," *arXiv preprint arXiv:2108.10470*, 2021.