IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

# Squeezing the Last Drop of Accuracy: Hand-Eye Calibration via Deep Reinforcement Learning-Guided Pose Tuning

Seunghui Shin, Daeho Kim, *Student Member, IEEE*, and Hyoseok Hwang, *Associate Member, IEEE*

***Abstract*—Hand-eye calibration is a fundamental task in robotics, requiring high precision to ensure accurate manipulation. This is especially crucial for recent markerless methods, which depend on precise pose estimation for effective end-effector calibration. In this letter, we propose a novel approach that improves calibration performance by adjusting the end-effector's pose to reduce prediction error. Our method utilizes a reward structure derived from trained pose estimation networks, enabling a Soft Actor-Critic-Discrete agent to learn in a simulated environment how to enhance calibration performance through action selection. Our experiments show that calibration results achieved with our method outperform those from initial poses alone in both markerless and marker-based methods. Real-world experiments further validate the efficacy of our approach in actual robotic systems. These results demonstrate that our proposed method effectively enhances the performance of pose estimation-based hand-eye calibration.**

***Index Terms*—Calibration and identification, reinforcement learning, machine learning for robot control.**

## I. Introduction

RECENT advancements in automated systems have enabled manipulators to shift from executing pre-defined tasks in static environments to operating in dynamic settings [1], [2]. In such environments, robots can perform effectively when equipped with visual sensors like cameras and LiDAR, which enhance environmental perception [3], [4]. Cameras are favored for their versatility in vision tasks such as detection and pose estimation [5]. They offer high-resolution imaging at a lower cost than other sensors [6], and offer real-time feedback [7]. This makes cameras a key tool for robots to interact with and adapt to dynamic conditions.

To effectively leverage camera data, it is critical to determine the relative pose between the camera and the manipulator. In robotic tasks [8], where the robot grasps objects identified by the camera, the success of grasping depends on accurately estimating the pose between the camera and the robot. Any misalignment can lead to incorrect grasping. This highlights the importance of *hand-eye calibration*, which estimates the transformation between the camera and manipulator coordinate systems, typically formulated as solving the equation $AX = XB$, where $A$ and $B$ denote the relative motions of the robot end-effector and the camera, and $X$ is the sought transformation between them.

Hand-eye calibration is a well-established field that traditionally estimates fiducial marker poses [9], [10], [11], then computes the camera-to-robot transformation using the manipulator's known forward kinematics [12], [13], [14], [15]. While these methods are effective for acquiring 6-degrees of freedom (DoF) poses, they become impractical in environments with restricted human access or when online calibration is required [16], [17]. Moreover, their accuracy is influenced by factors such as frame count, marker size, and joint configurations [18]. To address these limitations, recent works introduced markerless approaches that directly estimate the 6-DoF pose of the robot, eliminating the need for markers [19], [20], [21], [22], [23]. Nevertheless, the effectiveness of hand-eye calibration is fundamentally dependent on pose estimation performance.

Even with state-of-the-art pose estimators, calibration does not always yield optimal results. This is because pose estimation accuracy depends on factors such as keypoint localization reliability [24], [25], ambiguities in regression frameworks [26], and depth precision [27], which cause variability even under the same estimator. Consequently, selecting good poses with lower errors is crucial for improving calibration performance. However, identifying good poses is challenging. The only viable approach is to exhaustively compare all possible poses against ground truth (GT) in simulation, which is computationally intractable, and even minor environmental changes require re-evaluating the entire pose space. More fundamentally, in practical scenarios where GT is unavailable, assessing candidate poses with respect to calibration accuracy is infeasible. To address this issue, we hypothesize the existence of better poses and believe this challenge can be solved by training a model to guide the end-effector toward configurations with lower estimation errors. This is particularly necessary because pose estimation accuracy

Received 2 July 2025; accepted 17 October 2025. Date of publication 24 October 2025; date of current version 3 November 2025. This article was recommended for publication by Associate Editor S. Zhou and Editor S.-J. Chung upon evaluation of the reviewers' comments. This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korean Government (MSIT) under Grant RS-2025-00564137, in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) grant funded by the Korean Government (MSIT) under Grant RS-2022-00155911, in part byArtificial Intelligence Convergence Innovation Human Resources Development (Kyung Hee University), and in part by Convergence Security Core Talent Training Business Support Program under Grant IITP-2023-RS-2023-00266615. *(Corresponding author: Hyoseok Hwang.)*

The authors are with the Department of Software Convergence, Kyung Hee University, Yongin-si 17104, Republic of Korea (e-mail: jumin1116@khu.ac.kr; kdh2769@khu.ac.kr; hyoseok@khu.ac.kr).

Our project page is at https://airlabkhu.github.io/Squeezing-HEC/.

This article has supplementary downloadable material available at https://doi.org/10.1109/LRA.2025.3625510, provided by the authors.

Digital Object Identifier 10.1109/LRA.2025.3625510

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

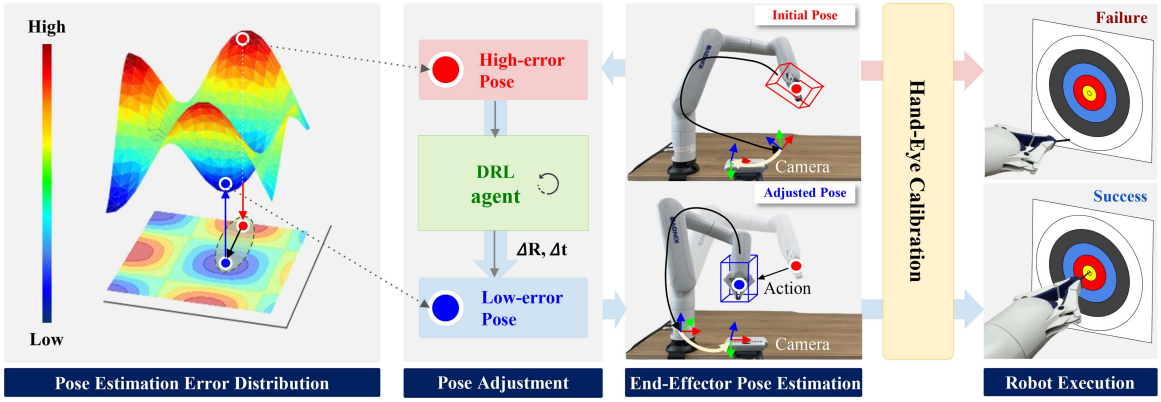


Fig. 1. Conventional pose-based hand-eye calibration relies on the initial end-effector pose, often yielding inaccurate calibration (trajectory shown by the red arrow). Our approach leverages deep reinforcement learning (DRL) to train an agent that adjusts the end-effector toward better poses to improve calibration accuracy (trajectory shown by the blue arrow).

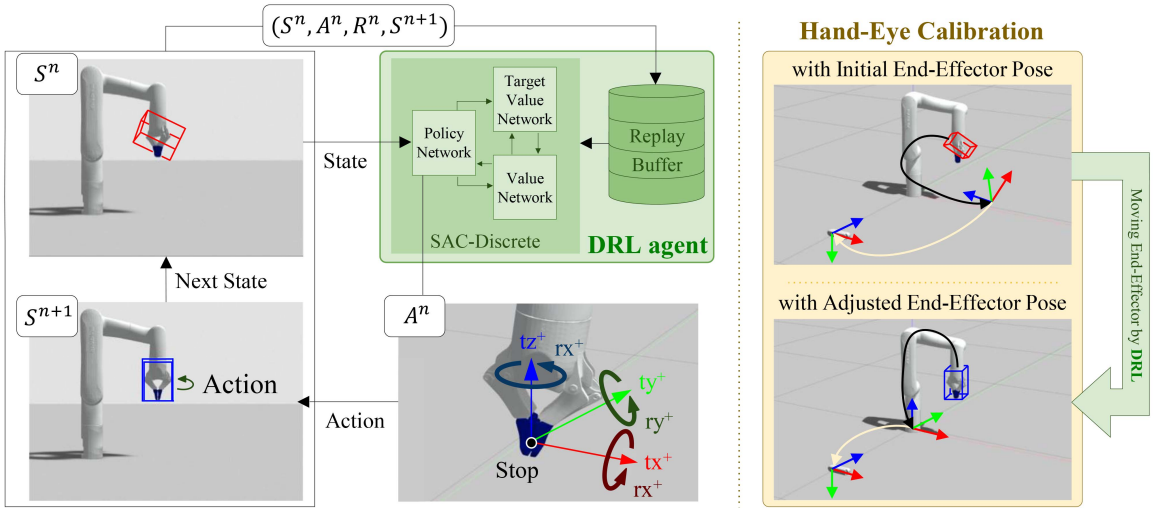


Fig. 2. Our framework first estimates the end-effector pose using a pose estimation network. Subsequently, a SAC-Discrete agent is trained to adjust the end-effector to poses that yield lower estimation errors. Hand-eye calibration is then performed using the known forward kinematics and the refined end-effector poses.

exhibits nonlinear characteristics relative to pose parameters, making learning-based methods suitable for capturing complex relationships and aiding efficient discovery of improved poses without exhaustive search.

We introduce an innovative approach to hand-eye calibration aimed at enhancing performance within the constraints of existing pose estimation methods. Our approach employs deep reinforcement learning (DRL) to enable the robot to adapt its movements, guiding the end-effector toward poses that produce more accurate calibration results, as shown in Fig. 1. By designing reward structures that promote exploration of poses with lower estimation errors, the DRL agent learns to select actions that consistently improve calibration outcomes. This DRL-based strategy is validated through both simulation and real-world experiments, showing significant improvements in hand-eye calibration and end-effector pose estimation. Furthermore, our method is versatile, applicable to both markerless and marker-based calibration scenarios.

The main contributions of this study are as follows:

- We introduce a method for improving hand-eye calibration by adjusting the end-effector to poses that reduce error, conditioned on the target object's pose. To the best of our knowledge, this is the first attempt to reduce hand-eye calibration errors by manipulating the target.
- We present a DRL framework that performs actions to guide the end-effector toward improved poses, leveraging a reward structure based on the estimated pose.
- We demonstrate the applicability of our method to pose-based calibration and validate performance gains in both markerless and marker-based settings.

## II. Related Works

### A. Hand-Eye Calibration

Hand-eye calibration includes the eye-to-hand problem, where the camera is fixed in the world, and the eye-in-hand problem, where it is fixed to the hand. The eye-to-hand problem is sometimes extended to robot-world hand-eye calibration, which additionally estimates the transformation between the robot base and the world coordinate system, improving flexibility in diverse environments [28], [29], [30]. Traditional approaches

employ fiducial markers such as checkerboard [31], ArUco [9], ARTag [10], and AprilTag [11], attached to the end-effector for eye-to-hand or placed on the ground for eye-in-hand calibration, with the camera-to-robot transformation computed using image observations and forward kinematics [12], [13], [14], [15]. Since marker-based methods depend strongly on marker visibility [16], markerless approaches have recently gained attention. In particular, our work addresses markerless eye-to-hand calibration, which has been the focus of several recent studies. For example, [17], [20], [21] estimate 2D keypoints of robot joints and apply the P*n*P algorithm [32], while EasyHEC [16] and CtRNet [23] leverage differentiable rendering of full-arm observations. Sefercik and Baris [19] instead estimate 6-DoF end-effector poses, making calibration accuracy highly dependent on the estimator. In contrast, our method improves calibration within existing pose estimators by guiding the end-effector toward poses that yield lower estimation errors.

### B. Object 6-DoF Pose Estimation

Deep neural networks have been widely adopted for estimating the 6-DoF pose of target objects. DOPE [33] and YOLO6D [34] infer 2D-3D correspondences and solve for the object pose using the P*n*P algorithm [32]. PoseCNN [35] and DenseFusion [36] directly regress poses using point matching loss. These methods often require large-scale, high-quality training datasets, which are typically generated synthetically to reduce annotation costs. DOPE, for instance, relies entirely on synthetic datasets with domain randomization [37] and photorealistic rendering [38]. More recently, FoundationPose [39] has addressed pose estimation for unseen objects without requiring training in both model-based and model-free scenarios.

### C. Deep Reinforcement Learning for Calibration

Several studies have explored DRL in calibration tasks. Ao et al. [40] formulate a compact Markov decision process (MDP) and apply soft actor-critic (SAC) [41] to improve camera-IMU calibration. While effective for visual-inertial systems, their approach does not extend to hand-eye calibration. Zheng et al. [42] utilize DRL to move the end-effector to a pre-defined pose but do not evaluate pose quality in terms of calibration accuracy. In contrast, our method uses pose estimation error as a reward, encouraging the agent to select poses that lead to improved calibration performance.

## III. Method

Our objective is to enhance hand-eye calibration by guiding the end-effector to poses with reduced estimation errors using DRL. The overall pipeline is illustrated in Fig. 2. We assume that the camera pose is fixed and that the end-effector remains within the camera's field of view (FoV) during this process. Initially, we estimate the end-effector's pose for a single configuration using pose estimation networks trained on a synthetic dataset. During DRL training, each episode starts from a different randomly sampled pose, and the SAC-Discrete [43] agent learns to select actions that move the end-effector within that pose to reduce estimation errors. When the agent selects a stop action, the current pose is assumed to yield sufficiently low estimation error. Hand-eye calibration is subsequently performed using the final estimated pose and known forward kinematics from the robot base.

### A. End-Effector Pose Estimation

We perform hand-eye calibration by estimating the 6-DoF pose of the end-effector, which requires training pose estimation algorithms on large datasets. To this end, we use NVIDIA deep learning dataset synthesizer (NDDS) [44] to generate synthetic end-effector images with corresponding GT annotations. To enhance generalization, we apply domain randomization with five components: (1) end-effector pose variation, (2) background randomization, (3) end-effector foreground alteration, (4) obstacle placement, and (5) variation of lighting and sensor noise. For end-effector pose variation, the camera is fixed while the end-effector translation is randomly sampled within the camera frame as $T_x \in [-0.2\tilde{}\text{m}, 0.2\tilde{}\text{m}]$, $T_y \in [-0.1\tilde{}\text{m}, 0.1\tilde{}\text{m}]$, and $T_z \in [0.2\tilde{}\text{m}, 1.2\tilde{}\text{m}]$ to remain in the FoV. Rotations are sampled as $R_x \in [-45°, 45°]$, $R_y \in [0°, 360°]$, and $R_z \in [-45°, 45°]$. Backgrounds use COCO images, grids, and solid colors, while the end-effector foreground uses random colors and patterns. Obstacles, lighting, and sensor noise are also randomized.

Using this synthetic dataset, we train DenseFusion [36] and DOPE [33] for 6-DoF pose estimation of the end-effector. Since DenseFusion and FoundationPose [39] require segmentation masks as input, we also train SegNet [45] for segmentation. For segmentation, a separate synthetic dataset is generated using domain randomization. However, to improve segmentation quality, the entire robotic arm is rendered. Since FoundationPose is capable of zero-shot 6-DoF pose estimation, no additional training is performed.

### B. Deep Reinforcement Learning

*1) Action Definition:* The action space, denoted as $A$, comprises 13 distinct actions, including various translations, rotations, and a stop action. These actions facilitate the movement of the end-effector relative to the robot base's coordinate system. Specifically, the translation actions are divided into six dimensions: $\{tx^+, tx^-, ty^+, ty^-, tz^+, tz^-\}$ representing movements along the x, y, and z axes. Similarly, the rotation actions are also categorized into six dimensions: $\{rx^+, rx^-, ry^+, ry^-, rz^+, rz^-\}$ corresponding to rotations around these axes. We define the extent of translation and rotation movements of the end-effector according to the pose estimation network. The range of end-effector movements for each pose estimation network can be found in Section IV-F2.

*2) State Definition:* The state space $S^{\tilde{}} \in {}^{\tilde{}}\mathbb{R}^{13}$ consists of the 6-DoF pose of the end-effector relative to the robot base $P_R^E$, predicted pose of the end-effector in the camera frame $P_C^E$, and a step $n$, $S^n = [(P_R^E)^n, (P_C^E)^n, n]$. If a stop action occurs in just one step, it depends on the initial pose of the end-effector, so we add a step to the state to prevent a stop action from occurring in just one step.

*3) Reward Design:* The reward is defined based on the pose accuracy of the end-effector, which is estimated by the pose estimation networks after the end-effector has moved through the action from the current state. We compute the ADD metric [35], [36] between the 3D object points reconstructed by the GT and the estimated pose of the end-effector. The ADD metric is defined as:

$$\text{ADD}\,[\text{cm}] = \frac{1}{M}\sum_i \left\| (Rx_i + t) - (\hat{R}x_i + \hat{t}) \right\|, \qquad (1)$$

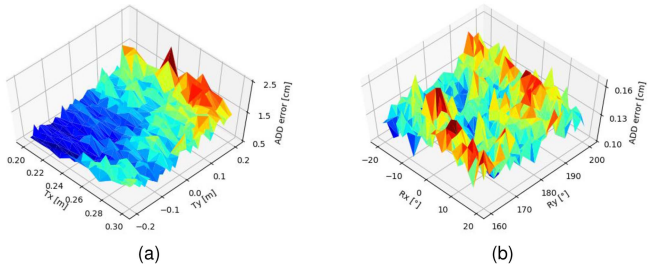


Fig. 3. Pose estimation errors across varying end-effector poses using (a) DOPE under translational variation, and (b) DenseFusion under rotational variation.

where $x_i$ denotes the $i^{th}$ point of $M$ randomly selected 3D points from the object's 3D model, $\left[\tilde{R}|\tilde{t}\right]$ are the rotation and translation of GT pose, and $\left[\hat{\tilde{R}}|\hat{\tilde{t}}\right]$ are estimated pose.

We design the discrete reward according to the ADD inspired by [42]. The reward $R_d$ is defined as:

$$R_d = \begin{cases} 1, & \text{ADD}_0 * 4 \leq \text{ADD}_n \\ 2, & \text{ADD}_0 * 2 \leq \text{ADD}_n < \text{ADD}_0 * 4 \\ 4, & \text{ADD}_0 \leq \text{ADD}_n < \text{ADD}_0 * 2 \\ 8, & \text{ADD}_n < \text{ADD}_0, \end{cases} \tag{2}$$

where $\text{ADD}_0$ is the ADD of the end-effector pose at the first step, and $\text{ADD}_n$ is the ADD at the $n$th step.

The reward continues to be kept at zero until a stop action occurs. If a stop action occurs in the first step, a reward of -1 is given. A discrete reward $R_d$ is given if a stop action occurs after the first step. So, the terminal reward $R^n$ is defined as:

$$R^n = \begin{cases} -1, & \text{if stop action occurs in first step} \\ R_d, & \text{if stop action occurs after first step} \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

*4) Policy Learning:* We adopt the SAC-Discrete [43] to enable the agent to learn an optimal policy. During online training, the agent learns policy and value networks, with network parameter updates following the maximum entropy framework. The off-policy nature of this framework improves sample efficiency, allowing the agent to identify lower-error poses and execute a stop action with reduced interaction steps. The agent's ability to perform discrete actions, particularly the crucial stop action, makes SAC-Discrete especially suitable for our objective of guiding the end-effector to improved poses and enabling it to stop at those poses, thereby enhancing calibration performance.

### C. Hand-Eye Calibration

In our hand-eye calibration approach, we start with the assumption that the forward kinematics from the robot's base frame to its end-effector is known. Once the trained agent has maneuvered the end-effector into an improved pose and initiated a stop action, we perform hand-eye calibration. This is done by utilizing the estimated transformation of the end-effector and the pre-known forward kinematics. The process of hand-eye calibration can be described as:

$$T_C^R = T_C^E (T_R^E)^{-1}. \tag{4}$$

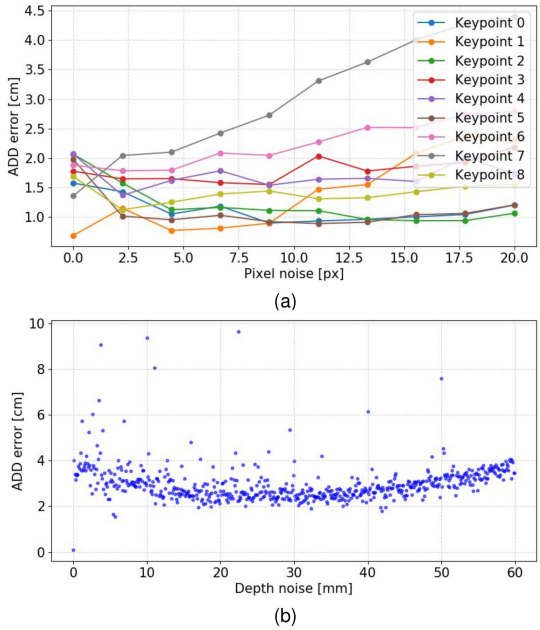


Fig. 4. Pose estimation errors with increasing (a) pixel noise, and (b) depth noise.

Here, $T_C^R$ represents the transformation of the robot base in the camera frame which is the target output of the hand-eye calibration. $T_C^E$ denotes the transformation of the end-effector in the camera frame, and $T_R^E$ signifies the transformation of the end-effector in the robot frame.

## IV. Experimental Results

### A. 6-DoF Pose Estimation Implementation Details

To train DenseFusion, we generated 2 k RGB, depth, and segmentation frames with pose annotations for training and 0.2 k frames for validation. SegNet was trained with 7 k RGB–segmentation pairs and validated on 1 k additional frames. For DOPE, we created a larger dataset of 30 k RGB images with annotated keypoints and poses for training and 3 k frames for validation. To maximize each network's performance, we constructed model-specific datasets tailored to the input requirements and learning characteristics.

### B. Experimental Setup

We used Gazebo [46] to simulate dynamics and sensors, including a Kinova Gen3-Lite and a Realsense D435. To increase realism, Gaussian noise was added to the images, with mean 0 and standard deviation (Std) 0.007 for color and mean 0 and Std 0.1 m for depth. The camera was positioned at $tx = 0.3$˜m, $ty = -1.0$˜m, $tz = 0.2$˜m, $rx = 0°$, $ry = 0°$, and $rz = 90°$ in the robot frame. For each episode, the end-effector's initial pose was randomly sampled within $tx \in$ [0.2m, 0.3 m], $ty \in$ [-0.2m, 0.2 m], $tz \in$ [0.2m, 0.3 m], $rx \in$ [-20°, 20°], $ry \in$ [160°, 200°], and $rz \in$ [-20°, 20°], and allowable poses were constrained to prevent collisions. End-effector poses were estimated using pose networks, and robot actions were executed via MoveIt [47]. Using a pre-set MDP, we trained a SAC-Discrete policy with fully connected policy and value networks of two hidden layers

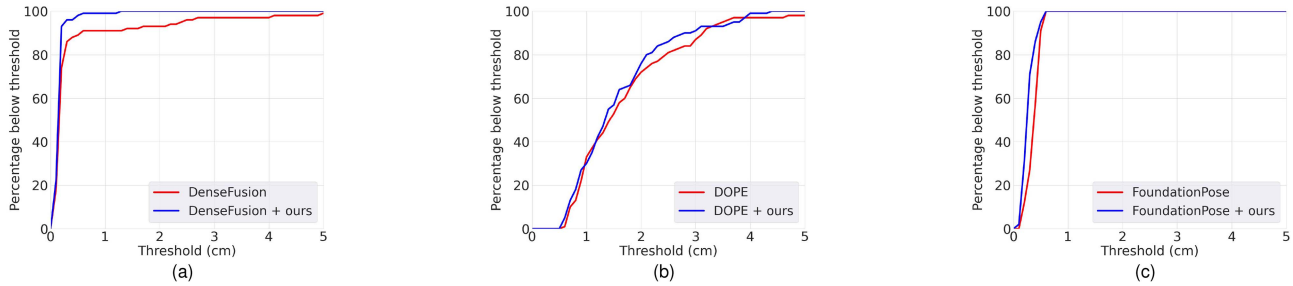


Fig. 5. ADD threshold curves obtained with (a) DenseFusion, (b) DOPE, and (c) FoundationPose.

with 256 neurons each. The policy network applied a softmax to select a single action at each step. Networks were trained using Adam with a batch size of 128 and replay buffer of 100 k. The learning rate was 0.0005 and the discount factor 0.99. Each episode was limited to 10 steps, and training ran for 20 k steps.

### C. Verification and Analysis of Pose Estimation Error

Our approach was based on the premise that an object's pose affected pose estimation accuracy. To validate this, we measured estimation errors under controlled pose variations. The end-effector's orientation was fixed at $rx = 0°$, $ry = 180°$, and $rz = 0°$ while position varied by 0.01m, and its position was fixed at $tx = 0.25$˜m, $ty = 0$˜m, and $tz = 0.25$˜m while orientation varied by 2°. Pose estimation was performed using DenseFusion [36] and DOPE [33], with errors computed using Eq. 1. Results were shown on the x-y axis in Fig. 3, confirming that errors changed with end-effector pose and that some poses yielded lower errors.

We further investigated how factors such as keypoint localization and depth noise influence pose estimation accuracy. Using DOPE, a keypoint-based method, we observed that increasing pixel errors for individual keypoints led to nonlinear changes in the end-effector's ADD error, with the effect varying across different keypoints, as shown in Fig 4(a). Similarly, with DenseFusion, an end-to-end depth-based regression model, increasing depth noise introduced a depth bias, leading to a nonlinear increase in ADD error, as shown in Fig 4(b). These results indicated that pose estimation errors responded nonlinearly to input noise, highlighting the limitations of heuristic approaches. Consequently, learning-based methods, such as reinforcement learning, were necessary to effectively identify improved poses.

### D. Evaluation in Simulation

In the simulation evaluation, we assessed the trained policy over 100 test episodes with different initial end-effector poses. Evaluation focused on hand-eye calibration and pose estimation accuracy, using two key metrics: estimation errors and area under the curve (AUC). Estimation errors included translation error $(\epsilon_t)$, measured as the Euclidean distance between estimated and GT translations, and rotation error $(\epsilon_r)$, defined as the minimum rotation between estimated and GT rotations. For both errors, we reported mean $\pm$ Std, median $\pm$ median absolute deviation (MAD), and interquartile range (IQR) mean $\pm$ Std for outlier-robust analysis [48]. The AUC was derived from the ADD-threshold curve, with higher values indicating better performance. When no stop action occurred, the final pose was taken at maximum step.

*1) Markerless Hand-Eye Calibration:* We applied our approach to markerless hand-eye calibration using DenseFusion [36], DOPE [33], and FoundationPose [39]. Performance metrics for hand-eye calibration and end-effector pose estimation were summarized in Table I. Our method significantly reduced estimation errors for hand-eye calibration. Trends in end-effector pose estimation matched those in calibration, as expected, since calibration depended directly on pose accuracy. These results confirmed that our approach effectively guided the end-effector toward poses that reduced errors, enhancing both pose estimation and calibration performance. Moreover, our method was modular and could be integrated into various pose estimation networks. We compared our method with DREAM [17] and RPM [19]. Using DenseFusion, our approach outperformed both. DREAM's keypoint estimation varied with robot pose, causing calibration errors, while RPM did not leverage improved poses, causing poor performance.

Threshold plots showing the proportion of predicted end-effector poses with errors below specific ADD values were shown in Fig. 5. These were summarized via AUC up to a pre-set maximum threshold, as shown in Table II, demonstrating that our method effectively reduced ADD.

Furthermore, the improved poses were used for multi-pose hand-eye calibration, as proposed in RPM [19]. Table III showed that employing these enhanced poses also improved calibration performance compared to the initial poses.

*2) Marker-Based Hand-Eye Calibration:* We applied our method to marker-based hand-eye calibration [12], [13], [14], [15], [30], using a 20 cm $\times$ 20 cm ArUco marker across 20 viewpoints. Table IV reported the error metrics for both hand-eye calibration and marker pose estimation. The calibration performance was highly sensitive to marker pose accuracy; increased marker pose error led to larger calibration errors. By guiding the end-effector to poses with lower estimation errors, our method effectively reduced errors in both marker pose estimation and hand-eye calibration, demonstrating its utility in improving marker-based settings.

### E. Evaluation in Real-World Setting

*1) High-Precision Targeting:* To validate transfer from simulation, we performed a high-precision targeting experiment using an ArUco marker. The setup included a Kinova Gen3-Lite and a Realsense D435. A pointer on the end-effector aimed at the marker's corners, and the distance to actual corners was manually measured. We evaluated 100 test episodes with different initial poses and compared with a random moving policy. As shown in Fig. 6, our method consistently reduced calibration errors, while the random policy gave marginal improvement

TABLE I
ERRORS OF HAND-EYE CALIBRATION AND END-EFFECTOR POSE ESTIMATION FROM SIMULATION

| | | Mean±Std | | Median±MAD | | IQR Mean±Std | |
|---|---|---|---|---|---|---|---|
| | Method | $\epsilon_t$[cm] | $\epsilon_r[^\circ]$ | $\epsilon_t$[cm] | $\epsilon_r[^\circ]$ | $\epsilon_t$[cm] | $\epsilon_r[^\circ]$ |
| Hand-Eye Calibration | DenseFusion | 2.057±5.166 | 4.563±14.052 | 0.653±0.344 | 0.935±0.528 | 0.664±0.313 | 0.929±0.416 |
| | DOPE | 7.528±27.590 | 11.087±24.763 | 2.751±2.143 | 5.743±4.559 | 3.265±2.177 | 6.330±4.243 |
| | FoundationPose | 2.328±1.191 | 4.604±1.718 | 2.264±1.195 | 4.715±1.688 | 2.225±1.053 | 4.604±1.718 |
| | DREAM | 2.541±4.342 | 8.421±15.597 | 1.233±1.414 | 3.116±3.140 | 1.665±1.611 | 3.404±2.431 |
| | RPM | 4.262±10.940 | 6.521±16.696 | 1.798±0.652 | 2.816±1.008 | 1.842±0.545 | 2.695±0.866 |
| | DenseFusion + **ours** | **0.687±0.517** | **1.127±1.642** | **0.612±0.264** | **0.860±0.387** | **0.609±0.223** | **0.867±0.330** |
| | DOPE + **ours** | **3.225±2.799** | **7.137±6.492** | **2.129±1.466** | **4.729±3.302** | **2.628±1.750** | **5.146±3.527** |
| | FoundationPose + **ours** | **1.822±0.913** | **3.226±1.653** | **1.797±0.941** | **2.844±1.193** | **1.822±0.913** | **3.040±1.401** |
| End-Effector Pose Estimation | DenseFusion | 0.312±0.920 | 4.562±14.052 | 0.083±0.048 | 0.920±0.508 | 0.078±0.035 | 0.927±0.416 |
| | DOPE | 5.325±27.050 | 11.086±24.764 | 1.256±0.844 | 5.730±4.559 | 1.519±0.901 | 6.329±4.243 |
| | FoundationPose | 0.423±0.212 | 4.604±1.718 | 0.416±0.222 | 4.715±1.688 | 0.408±0.186 | 4.604±1.718 |
| | DREAM | 2.393±4.456 | 8.421±15.597 | 0.813±0.590 | 3.116±3.140 | 0.935±0.659 | 3.404±2.431 |
| | RPM | 2.154±1.649 | 6.482±16.706 | 1.715±0.220 | 2.667±0.987 | 1.758±0.242 | 2.651±0.860 |
| | DenseFusion + **ours** | **0.087±0.096** | **1.127±1.642** | **0.065±0.027** | **0.860±0.387** | **0.068±0.026** | **0.867±0.330** |
| | DOPE + **ours** | **1.479±0.879** | **7.137±6.492** | **1.158±0.650** | **4.729±3.302** | **1.376±0.732** | **5.146±3.527** |
| | FoundationPose + **ours** | **0.237±0.180** | **3.227±1.652** | **0.188±0.112** | **2.844±1.193** | **0.180±0.092** | **3.041±1.401** |

TABLE II
AUC UP TO A 5 CM THRESHOLD FOR THE ADD WHEN USING OUR METHOD

| Method | AUC | Method | AUC |
|---|---|---|---|
| DenseFusion | 91.77 | DenseFusion + **ours** | **96.96** |
| DOPE | 66.12 | DOPE + **ours** | **68.54** |
| FoundationPose | 92.74 | FoundationPose + **ours** | **94.70** |

TABLE III
ERRORS OF HAND-EYE CALIBRATION USING MULTIPLE POSES FROM SIMULATION

| | Hand-Eye Calibration | |
|---|---|---|
| Method | $\epsilon_t$[cm] | $\epsilon_r[^\circ]$ |
| RPM | 1.445 | 1.699 |
| Densefusion | 0.587 | 0.789 |
| Densefusion + **ours** | **0.528** | **0.710** |
| DOPE | 1.115 | 3.067 |
| DOPE + **ours** | **0.892** | **2.027** |
| FoundationPose | 1.958 | 3.815 |
| FoundationPose + **ours** | **1.228** | **1.625** |

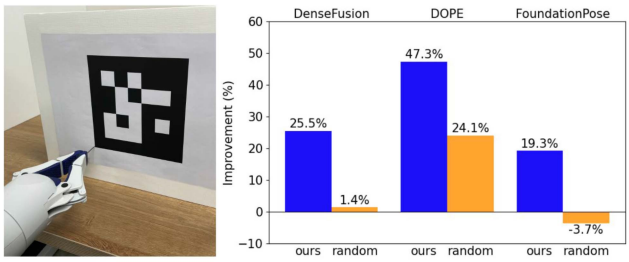


Fig. 6. Setup and evaluation results in the real-world high-precision targeting experiment.

or increased error with FoundationPose. These results confirmed the effectiveness of our approach in improving real-world hand-eye calibration.

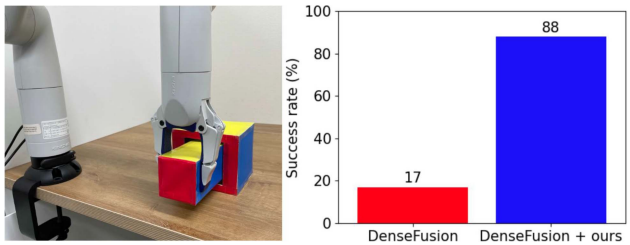


Fig. 7. Setup and evaluation results in the real-world peg-insertion experiment.

*2) Peg-Insertion:* We conducted a peg-insertion task to assess the impact of improved calibration on manipulation performance. 100 test episodes with varying initial poses were evaluated using DenseFusion for pose estimation. The end-effector maintained a fixed insertion depth for safety. As shown in Fig. 7, integrating our method increased task success 71%, showing even small improvements in hand-eye calibration substantially enhance real-world task execution.

### *F. Ablation Studies*

*1) Optimizing State Space:* The state space $S$ was defined as $S^n = [(P_R^E)^n, (P_C^E)^n, n]$, including the end-effector pose relative to the robot base, its pose relative to the camera estimated by the pose networks, and the current step index $n$, which discourages premature stop actions. To evaluate its effect, we performed ablation studies comparing policies trained with and without $n$. As shown in Fig. 8, omitting $n$ resulted in slower reward convergence in both markerless and marker-based settings, confirming that incorporating the step index stabilizes policy learning.

*2) Optimizing Action Scale:* To reduce the action space and aid policy convergence, end-effector motions were limited to fixed translation and rotation magnitudes. We assessed the impact of these scales through ablation: translation was fixed at 0.2 cm while rotation varied between 1°, 2°, and 4°; conversely, rotation was fixed at 2° while translation varied between 0.1, 0.2, 0.5, and 1.0 cm. Fig. 9 shows that performance remained largely consistent across scales. Based on these results, we

TABLE IV
ERRORS OF HAND-EYE CALIBRATION AND MARKER POSE ESTIMATION FROM SIMULATION

| | | Mean±Std | | Median±MAD | | IQR Mean±Std | |
|---|---|---|---|---|---|---|---|
| | Method | $\epsilon_t$[cm] | $\epsilon_r$[°] | $\epsilon_t$[cm] | $\epsilon_r$[°] | $\epsilon_t$[cm] | $\epsilon_r$[°] |
| Hand-Eye Calibration | TSAI [12] | 2.100±0.521 | 0.962±0.521 | 1.915±0.783 | 0.946±0.833 | 1.558±0.401 | 0.962±0.521 |
| | PARK [13] | 2.039±1.053 | 1.071±0.533 | 1.610±0.579 | 0.993±0.887 | 1.535±0.338 | 1.071±0.533 |
| | HORAUD [14] | 2.040±1.052 | 1.074±0.534 | 1.608±0.583 | 0.996±0.889 | 1.536±0.336 | 1.074±0.534 |
| | DANIILIDIS [15] | 1.729±0.771 | 0.922±0.385 | 1.468±0.186 | 0.868±0.548 | 1.437±0.068 | 0.922±0.385 |
| | ALI [30] | 2.635±1.195 | 1.115±0.576 | 2.258±1.250 | 1.010±0.982 | 2.635±1.195 | 1.115±0.576 |
| | TSAI + **ours** | **1.162±0.361** | **0.497±0.188** | **0.981±0.233** | **0.517±0.275** | **1.162±0.361** | **0.497±0.188** |
| | PARK + **ours** | **1.136±0.400** | **0.535±0.257** | **0.875±0.195** | **0.605±0.397** | **1.136±0.400** | **0.535±0.257** |
| | HORAUD + **ours** | **1.135±0.402** | **0.537±0.257** | **0.867±0.181** | **0.607±0.394** | **1.135±0.402** | **0.537±0.257** |
| | DANIILIDIS + **ours** | **1.053±0.269** | **0.466±0.168** | **1.055±0.351** | **0.487±0.214** | **1.053±0.269** | **0.466±0.168** |
| | ALI + **ours** | **1.522±0.672** | **0.864±0.500** | **1.589±1.038** | **0.927±0.749** | **1.522±0.672** | **0.864±0.500** |
| Marker Pose Estimation | Marker-Based | 0.702±0.205 | 0.998±0.569 | 0.689±0.172 | 0.892±0.607 | 0.686±0.184 | 0.998±0.569 |
| | Marker-Based + **ours** | **0.664±0.197** | **0.956±0.586** | **0.665±0.176** | **0.827±0.554** | **0.662±0.171** | **0.920±0.535** |

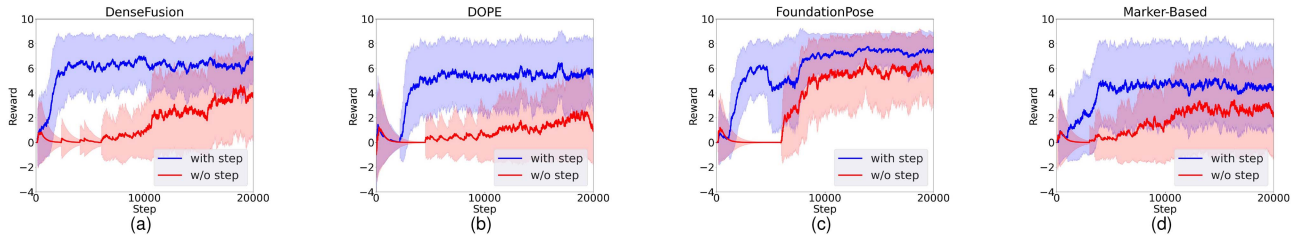


Fig. 8. Exponential moving average of rewards across different state definitions using (a) DenseFusion, (b) DOPE, (c) FoundationPose, and (d) Marker-Based.

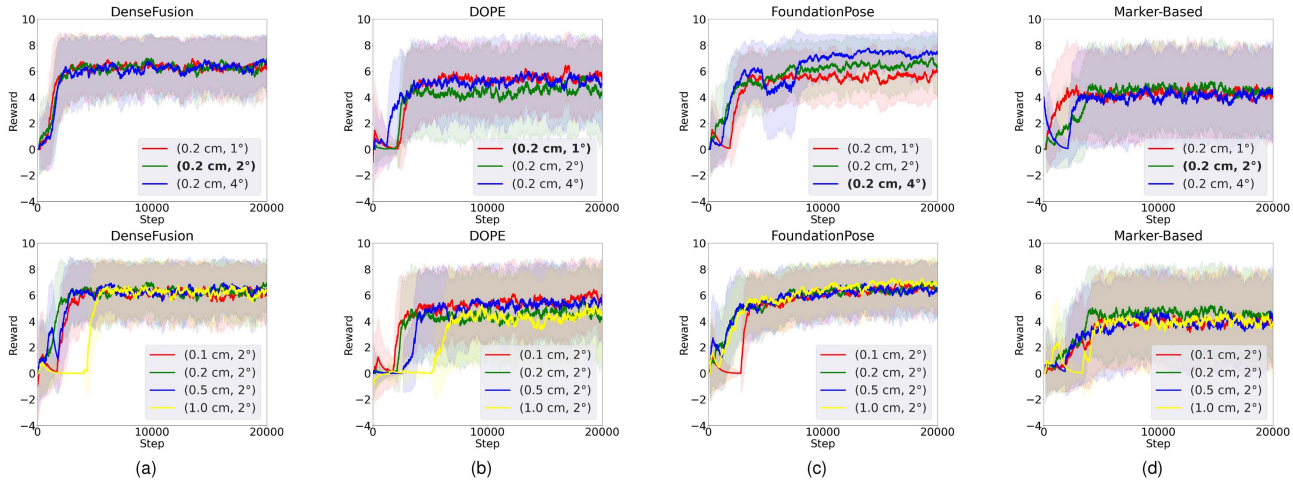


Fig. 9. Exponential moving average of rewards across different action scales when using (a) DenseFusion, (b) DOPE, (c) FoundationPose, and (d) Marker-Based. The first row reports rewards under rotational variation, while the second row presents rewards under translational variation. **Bold** indicates optimal action scales.

selected the scales that yielded optimal performance for each method: 0.2 cm translation and 2° rotation for DenseFusion and marker-based settings, 0.2 cm and 1° for DOPE, and 0.2 cm and 4° for FoundationPose.

## V. CONCLUSION

We propose a DRL-based method to move the end-effector for better hand-eye calibration. We train a SAC-Discrete agent in simulation with a reward structure based on estimated pose, guiding the agent toward lower estimation errors. Experimental results show that our method improves performance in both markerless and marker-based hand-eye calibration and is successfully applied in real-world settings. These results demonstrate that our proposed method effectively enhances the performance of pose estimation-based hand-eye calibration. Nevertheless, our method requires additional DRL training to identify better poses and still relies on fixed actions that vary across pose estimation networks. These constraints highlight the need for future work on more efficient training schemes

and optimized action strategies to further improve calibration performance.

## REFERENCES


[1] B. Huang, J. Yu, and S. Jain, "Earl: Eye-on-hand reinforcement learner for dynamic grasping with active pose estimation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2023, pp. 2963–2970.

[2] P. Xie et al., "GAP-RL: Grasps as points for RL towards dynamic object grasping," *IEEE Robot. Automat. Lett.*, vol. 10, no. 1, pp. 40–47, Jan. 2025.

[3] H. Mnyusiwalla et al., "A bin-picking benchmark for systematic evaluation of robotic pick-and-place systems," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 1389–1396, Apr. 2020.

[4] K. Matsubara, K. Nagatani, and Y. Hirata, "Improvement in measurement area of 3D LiDAR for a mobile robot using a mirror mounted on a manipulator," *IEEE Robot. Automat. Lett.*, vol. 5, no. 4, pp. 6350–6356, Oct. 2020.

[5] J. P. N. Cruz, M. L. Dimaala, L. G. L. Francisco, E. J. S. Franco, A. A. Bandala, and E. P. Dadios, "Object recognition and detection by shape and color pattern recognition utilizing artificial neural networks," in *Proc. 2013 Int. Conf. Inf. Commun. Technol.*, 2013, pp. 140–144.

[6] K. D. Mankoff and T. A. Russo, "The Kinect: A low-cost, high-resolution, short-range 3D camera," *Earth Surf. Processes Landforms*, vol. 38, no. 9, pp. 926–936, 2013.

[7] C. Caraffi, T. Vojíř, J. Trefn, J. Šochman, and J. Matas, "A system for real-time detection and tracking of vehicles from a single car-mounted camera," in *Proc. 15th Int. IEEE Conf. Intell. Transp. Syst.*, 2012, pp. 975–982.

[8] A. Cordeiro, L. F. Rocha, C. Costa, P. Costa, and M. F. Silva, "Bin picking approaches based on deep learning techniques: A state-of-the-art survey," in *Proc. 2022 IEEE Int. Conf. Auton. Robot Syst. Competitions*, 2022, pp. 110–117.

[9] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognit.*, vol. 47, no. 6, pp. 2280–2292, 2014.

[10] M. Fiala, "Artag, a fiducial marker system using digital techniques," in *Proc. 2005 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2005, vol. 2, pp. 590–596.

[11] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in *Proc. 2011 IEEE Int. Conf. Robot. Automat.*, 2011, pp. 3400–3407.

[12] R. Y. Tsai and R. K. Lenz, "A new technique for fully autonomous and efficient 3D robotics hand/eye calibration," *IEEE Trans. Robot. Automat.*, vol. 5, no. 3, pp. 345–358, Jun. 1989.

[13] F. C. Park and B. J. Martin, "Robot sensor calibration: Solving AX= XB on the euclidean group," *IEEE Trans. Robot. Autom.*, vol. 10, no. 5, pp. 717–721, Oct. 1994.

[14] R. Horaud and F. Dornaika, "Hand-eye calibration," *Int. J. Robot. Res.*, vol. 14, no. 3, pp. 195–210, 1995.

[15] K. Daniilidis, "Hand-eye calibration using dual quaternions," *Int. J. Robot. Res.*, vol. 18, no. 3, pp. 286–298, 1999.

[16] L. Chen, Y. Qin, X. Zhou, and H. Su, "EasyHeC: Accurate and automatic hand-eye calibration via differentiable rendering and space exploration," *IEEE Robot. Automat. Lett.*, vol. 8, no. 11, pp. 7234–7241, Nov. 2023.

[17] T. E. Lee et al., "Camera-to-robot pose estimation from a single image," in *Proc. 2020 IEEE Int. Conf. Robot. Automat.*, 2020, pp. 9426–9432.

[18] K. Pentenrieder et al., "Analysis of tracking accuracy for single-camera square-marker-based tracking," in *Proc. Proc. Dritter Workshop Virtuelle und Erweiterte Realitt der GIFachgruppe VR/AR*, Koblenz, Germany, Citeseer, 2006.

[19] B. C. Sefercik and B. Akgun, "Learning markerless robot-depth camera calibration and end-effector pose estimation," in *Proc. Conf. Robot Learn.*, 2023, pp. 1586–1595.

[20] J. Lambrecht and L. Kästner, "Towards the usage of synthetic data for marker-less pose estimation of articulated robots in RGB images," in *Proc. 19th Int. Conf. Adv. Robot.*, 2019, pp. 240–247.

[21] J. Lambrecht, "Robust few-shot pose estimation of articulated robots using monocular cameras and deep-learning-based keypoint detection," in *Proc. 7th Int. Conf. Robot Intell. Technol. Appl.*, 2019, pp. 136–141.

[22] Y. Labbé, J. Carpentier, M. Aubry, and J. Sivic, "Single-view robot pose and joint angle estimation via render & compare," in *Proc. Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 1654–1663.

[23] J. Lu, F. Richter, and M. C. Yip, "Markerless camera-to-robot pose estimation via self-supervised sim-to-real transfer," in *Proc. Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 21296–21306.

[24] S. Peng, Y. Liu, Q. Huang, X. Zhou, and H. Bao, "PVNet: Pixel-wise voting network for 6dof pose estimation," in *Proc. Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4561–4570.

[25] Y. Wu and M. Greenspan, "Learning better keypoints for multi-object 6dof pose estimation," in *Proc. Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2024, pp. 564–574.

[26] Y. Di, F. Manhardt, G. Wang, X. Ji, N. Navab, and F. Tombari, "SO-Pose: Exploiting self-occlusion for direct 6D pose estimation," in *Proc. Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 12396–12405.

[27] Z. Huang, K. Yao, Z. Zhao, C. Pan, and A. Yang, "Robust 6DoF pose estimation against depth noise and a comprehensive evaluation on a mobile dataset," in *Proc. Proc. Comput. Vis. Pattern Recognit. Conf.*, 2025, pp. 1848–1857.

[28] J. Heller, D. Henrion, and T. Pajdla, "Hand-eye and robot-world calibration by global polynomial optimization," in *Proc. 2014 IEEE Int. Conf. Robot. Automat.*, 2014, pp. 3157–3164.

[29] A. Tabb and K. M. Ahmad Yousef, "Solving the robot-world hand-eye (S) calibration problem with iterative methods," *Mach. Vis. Appl.*, vol. 28, no. 5, pp. 569–590, 2017.

[30] I. Ali, O. Suominen, A. Gotchev, and E. R. Morales, "Methods for simultaneous robot-world-hand–eye calibration: A comparative study," *Sensors*, vol. 19, no. 12, 2019, Art. no. 2837.

[31] J. Ilonen and V. Kyrki, "Robust robot-camera calibration," in *Proc. 15th Int. Conf. Adv. Robot.*, 2011, pp. 67–74.

[32] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EP N P: An accurate O (N) solution to the P N P problem," *Int. J. Comput. Vis.*, vol. 81, pp. 155–166, 2009.

[33] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield, "Deep object pose estimation for semantic robotic grasping of household objects," in *Proc. Conf. Robot Learn.*, 2018, pp. 306–316.

[34] B. Tekin, S. N. Sinha, and P. Fua, "Real-time seamless single shot 6D object pose prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 292–301.

[35] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes," in *Proc. Robot. Sci. Syst.*, 2018.

[36] C. Wang et al., "Densefusion: 6D object pose estimation by iterative dense fusion," in *Proc. Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3343–3352.

[37] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *Proc. 2017 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 23–30.

[38] M. Roberts et al., "Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding," in *Proc. Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 10912–10922.

[39] B. Wen, W. Yang, J. Kautz, and S. Birchfield, "Foundationpose: Unified 6D pose estimation and tracking of novel objects," in *Proc. Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 17868–17879.

[40] Y. Ao, L. Chen, F. Tschopp, M. Breyer, R. Siegwart, and A. Cramariuc, "Unified data collection for visual-inertial calibration via deep reinforcement learning," in *Proc. 2022 Int. Conf. Robot. Automat.*, 2022, pp. 1646–1652.

[41] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1861–1870.

[42] Z. Zheng, M. Yu, P. Guo, and D. Zeng, "NeuroDynamics adaptive reward and action for hand-to-eye calibration with deep reinforcement learning," *IEEE Access*, vol. 11, pp. 60292–60304, 2023.

[43] P. Christodoulou, "Soft actor-critic for discrete action settings," 2019, *arXiv:1910.07207*.

[44] T. To et al., "NDDS: NVIDIA deep learning dataset synthesizer," 2018. [Online]. Available: https://github.com/NVIDIA/Dataset_Synthesizer

[45] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[46] N. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," in *Proc. 2004 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2004, vol. 3, pp. 2149–2154.

[47] D. Coleman, I. A. Şucan, S. Chitta, and N. Correll, "Reducing the barrier to entry of complex robotic software: A moveit! Case study," *J. Softw. Eng. Robot.*, vol. 5, no. 1, pp. 3–16, 2014.

[48] R. Agarwal, M. Schwarzer, P. S. Castro, A. C. Courville, and M. Bellemare, "Deep reinforcement learning at the edge of the statistical precipice," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, vol. 34, pp. 29304–29320 .