Bridging the Simulation-to-Reality Gap in Reinforcement Learning-Based Autonomous Robot Navigation

[Cecilia Augustine Abi][1], [Joe Essien][2],[Amaku Effiong Amaku][3]

[1][Department of Computer Science], [Veritas University], [Abuja], Nigeria | ORCID: 0009-0001-0935-6631
[2][Department of Computer Science], [Veritas University], [Abuja], Nigeria \ ORCID:0000-0003-0871-1256
[3][Department of Computer Science], [Veritas University], [Abuja], Nigeria | ORCID:0000-0001-6043-2671

[1] [henriettaegbe@gmail.com], [2][essienj@veritas.edu.ng], [3][amakua@veritas.edu.ng]

Abstract - Reinforcement learning (RL) is applied to many autonomous robots in simulation before deploying them into the real world; however, the gap between simulation and reality is still a key challenge for many of these applications. RL algorithms have been shown to perform well in simulated environments but have shown poor performance when they are applied to a physical robot platform because of discrepancies in the behavior of the sensors, dynamics of the actuators, changes in environments, and modeling inaccuracies. The current study introduces a novel transfer framework for autonomous robot navigation using RL algorithms that combines the integration of domain randomization, sensor noise modeling, adaptive policy refinement, and multi-sensor fusion. The framework is developed in the Webots simulation environment and tested on a mobile robot built with the Raspberry Pi microcontroller and its various sensors including ultrasonic sensors, an inertial measurement unit, wheel encoders, infrared sensors and a camera module. Soft Actor-Critic (SAC) was used as the reinforcement learning algorithm for training. The Sim-to-Real transfer retention in the simulation experiments was 91% and in the physical-robot experiments was 84%, showing a 92.31% transfer retention between the simulations and reality. The framework was also able to achieve 7% collision rate and 92% generalization in unseen environments, outperforming the Standard SAC and Safety SAC baselines. The results clearly show that adaptive reinforcement learning could significantly improve the application of autonomous robotic systems, especially with the aid of complementary sim-to-real transfer strategies, in dynamic environments.

Keyword: Simulation to reality transfer, Reinforcement learning, Autonomous navigation, Domain randomization, multi-sensor fusion, Soft Actor-Critic.

I. INTRODUCTION

A. Background

With the growing trend of decentralized, mobile robots, autonomous mobile robots are now the focus of modern industrial automation, warehouse logistics, healthcare services, intelligent transportation, agriculture, surveillance and disaster response. The application is one in which robots have to work autonomously in a complex and dynamic environment, keeping them safe, reliable and efficient. For this, the robot should be able to perceive its environment, make decisions based on the context, adjust to changes in the environment, and carry out navigation tasks with (very) little human intervention. Navigation is one of the most basic functions of an autonomous robotic system, which involves locating the robot, detecting obstacles, planning paths, controlling its motion and deciding its goals. In a structured and predictable setting, traditional navigation methods such as rule-based, model-based, simultaneous localization and mapping (SLAM), potential-field, Dijkstra and A* graph-search algorithms work well, but they suffer in dynamic environments with uncertain real-world scenarios including moving obstacles, sensor noise, environmental variation and incomplete information [1, 2]. The recent advancements in AI and machine learning have radically changed the way autonomous navigation is being achieved. The last few years have seen a transformation in the way autonomous navigation is being tackled, thanks to the power of Artificial Intelligence (AI) and machine learning. Reinforcement learning (RL) is one of the most promising paradigms to enable robots to learn navigation behaviors in a trial-and-error fashion, which is a more natural way than explicitly programming every scenario. In contrast to supervised learning, RL agents learn skills by interacting with the environment and maximize the accumulated reward, without relying on labeled data. Combination of deep neural networks with RL — Deep Reinforcement Learning (DRL) has been seen to achieve human-level or super-human performance in domains of game playing, resource management, robotic manipulation and navigation [3]. In the modern DRL algorithms, Soft Actor-Critic (SAC) has been paid particular attention to continuous robotic control, which performs continuous optimization of both the actor and critic network, and incorporates the concept of maximizing the entropy of exploration to assist agents to balance the exploration and exploitation while maintaining sample efficiency and training stability [4].

B. The Simulation-to-Reality Challenge

While it is often possible to train reinforcement learning agents directly on physical robots, the time required for training is often long, the robot's safety is at stake and the costs of operating the physical robot are high. As a result, the vast majority of RL work is conducted on a suite of simulation platforms that provide safe, controllable, reproducible and cost-effective environments in which to train, like Webots, Gazebo, MuJoCo, Isaac Sim and PyBullet. But much of the time policies learned in simulation perform poorly on the real robotic platforms — especially when they are transferred from the simulation to the physical platform, which is referred to as the Simulation-to-Reality (Sim-to-Real) gap. The gap between the simulator and real-life is the result of the fact that no simulator can accurately simulate all the physical aspects of a system, even very advanced simulators tend to model some interactions somewhat differently. The result of this is seen in situations that are not fully represented during training and are faced by robots at deployment. Some of the most often mentioned causes are the presence of sensor noise and measurement uncertainty, as well as imperfect actuator dynamics, wheel slippage and mechanical wear, environmental variability and variation in lighting and weather, communication delay, unmodelled physical interactions and general hardware limitations [5], [6]. Idealizations of sensor readings used for training a policy may not apply to real sensors attached to a robot and differences in simulated and real surfaces can also have a big impact on the performance of trajectory-tracking. The inconsistencies result in navigation failures, increased collision rates, decreased task-success rates and poor policy generalization, making this Sim-to-Real gap one of the most prevalent open problems in reinforcement-learning-based robotics.

C. Existing approaches and Research Problem

There are a number of partial solutions that have been suggested. To elicit policies that generalize, an RL agent is presented with different random environments, including different sets of lighting, surface conditions, sensor properties, setups for obstacles and robot dynamics [7]. The incorporation of sensor-noise modeling into simulated readings adds realistic noise to readings, enabling policies to survive measurement inaccuracies in the real world. Transfer learning fine-tunes knowledge that has been learned in simulation to improve deployment performance, while saves training costs. An adaptive policy refinement and online learning enable a robot to learn after deployment, in order to deal with situations not anticipated during training.

Each of these strategies has proven to be effective alone, but so far, most of the studies have been conducted on one transfer technique at a time instead of a comprehensive approach that could tackle several sources of Sim-to-Real discrepancy. The majority of navigation systems continue to use restricted sensory inputs, thereby decreasing the awareness of the environment and making navigation more susceptible to uncertainties. Moreover, most of the reported results are simulation-only and the transferability of the proposed techniques to the real world is yet to be proved. The problem addressed in this study is a combination of the narrow single-technique framing and little physical validation: lack of an integrated physically-validated Sim-to-Real transfer framework integrating adaptive learning, domain randomization, sensor-noise modeling and multi-sensor fusion, in a single reinforcement-learning based navigation pipeline.

D. Aim, Objectives, and Significance

This study focuses on the development and evaluation of a comprehensive framework for transferring learning to the real world for a navigation system based on reinforcement learning for navigating an autonomous robot in dynamic environments. The specific objectives are: (1) create an adaptive RL-based navigation framework that works well in dynamic environments; (2) design domain randomization techniques to increase policy robustness and adaptability to the environment; (3) introduce sensor-noise modeling to increase robustness in dealing with real-world sensor measurement uncertainty; (4) integrate multi-sensor fusion to increase environmental perception and state estimation; (5) test the effectiveness of adaptive policy refinement during physical deployment and (6) evaluate Sim-to-Real transfer performance with simulated and physical robotic platforms.

The aim of this study is to make a contribution to the increasing number of research works in the field of autonomous robotics and one of the most challenging problems in the field is addressed. The proposed framework provides practical guidance on how policies could be more readily transferred from simulations to physical robots and the results could be useful for researchers, robotic-system developers, industrial-automation engineers and practitioners who are deploying intelligent autonomous systems in the fields of logistics, healthcare, manufacturing, surveillance, and disaster response.

The following is a specific list of contributions made in this paper:

- A unified Sim-to-Real transfer framework that unifies the domain randomization, sensor noise modeling, adaptive policy refinement and multi-sensor fusion (SSF) into a single Sim-to-Real navigation pipeline, not as isolated techniques.
- Physical implementation: They do not only simulate the system and test it, but also train it in Webots and deploy it to a mobile robot using a Raspberry Pi, with ultrasonic, IMU, encoder, infrared and camera sensing.

- Empirical results on effective transfer (91% simulated versus 84% physical navigation success rate, 92.31% transfer retention) and ablation studies to isolate the contribution of each component to the overall success rate.
- The rest of the paper is structural in the following way. In Section II, related work on RL-based navigation, the gap between simulation and reality, and the current transfer strategies is discussed, highlighting the research gap which this study fills. The proposed methodology, system architecture and experimental design are described in Section III. The experimental setup, experimental results and discussion are presented in section IV. The paper is concluded in section V which reveals its limitations and directions for future research.

II. Related Work

A. Reinforcement Learning for Autonomous Robot Navigation

In contrast to supervised learning, which requires labeled data to train reinforcement learning agents to make optimal decisions in a particular environment, the agents can learn optimal decision policies by interacting with the environment in an unsupervised manner. Deep Reinforcement Learning (DRL) takes this one step further by enabling agents to learn policies for navigating in dynamic environments from high-dimensional sensory inputs. For robotic navigation, there are several RL algorithms that have been successfully applied: Deep Q-Networks (DQN) showed success in the case of discrete actions whereas Deep Deterministic Policy Gradient (DDPG), Proximal Policy Optimization (PPO), Twin Delayed DDPG (TD3) and Soft Actor-Critic (SAC) are commonly used for continuous control. These include and have been the focus of attention on navigation tasks with environmental uncertainty, such as the sample efficiency, training stability and entropy-based exploration mechanism, that allows exploration while ensuring the robustness of the policy [4]. While these have been significant, RL-based navigation systems still struggle to converge quickly, policy instability and a lack of sufficient generalization, especially in the setting of transferring a policy from a simulation to a physical robot.

B. The Simulation-to-Reality Gap in Robotics

The Sim-to-Real gap is the difference between the performance of a policy in simulation and in the real world when it is physically deployed, and is considered to be one of the greatest challenges towards the implementation of RL in robotics. While Webots, Gazebo, MuJoCo, Isaac Sim, and PyBullet offer safe and cost-effective training facilities, policies trained are often found to be very different from those deployed in reality, due to inaccuracies in sensors, variation of actuator dynamics, wheel slippage and friction, environmental uncertainty, unmodeled physical interactions, hardware imperfections, and communication latency. The issue of not being able to achieve high performance from RL methods after deployment, despite good performance in simulation, is noted by Tiwari et al. [8] and they note that discrepancy between the simulator and the real world remains an important hurdle for real-world implementation. The same was also reported by Chen et al., [9] who said that the existing simulators are still far from being a perfect representation of real-world conditions, which creates the problem of transferability and decreases the deployment efficiency and they proposed to improve the transferability by developing more powerful transfer mechanism that can handle inconsistencies in environment and dynamics. Shi et al. [10] showed that Sim-to-Real transfer was a challenging open problem for RL in robotics and that for the method to be effective, there is a need for better adaptation mechanisms to cope with uncertainty in the real-world. Together, these results show that there is an ongoing need to pursue the research agenda for reducing the Sim-to-Real gap.

C. Domain Randomization for Sim-to-Real Transfer

Identifying the right domain for domain randomization. Creating the right domain for domain randomization. One of the most popular approaches to enhancing the transferability of a policy is to randomize the domain. Instead of training in a fixed environment, it introduces variability into the surface friction, light level, sensor properties, obstacle placements, environmental texture, robot dynamics, and initial position and orientation, resulting in the agent learning policies that are more generalizable to a wide variety of environments instead of overfitting a specific simulated environment [7], [11]. Several studies have demonstrated that domain randomization leads to higher robustness and better deployment of policies, but there is a risk that training is made too difficult or slow due to the randomization process, and that over-optimization can cause problems as well as with Sim-to-Real discrepancy — which is why it is used in tandem with complementary transfer techniques.

D. Sensor Noise Modeling

While physical sensors provide noisy, uncertain measurements due to limitations of the device, interference from the environment, calibration error, or degradation of the signal, simulators often do provide idealized measurements of sensors which do not represent these same conditions. This mismatch results in navigation policies being overly dependent upon unrealistic sensor behavior. In

order to overcome this, researchers are injecting realistic noise (usually Gaussian noise, random disturbance, drift simulation or measurement perturbation) into the sensor readings in the learning problem, and then the generated policy would be more robust to noisy sensor readings as the problem was deployed. Most of the studies conducted focus on one type of sensor only whereas sensor-noise modeling has been demonstrated to be beneficial for a single sensor, and it would be desirable to use this technique for a multi-sensor system in complex environments that require multiple sensory inputs.

E. Adaptive Learning, Policy Refinement, and Multi-Sensor Fusion

To continue learning after deployment, without having to rely only on a pre-trained, frozen policy, adaptive-learning approaches are used in a robotic system. Online-learning techniques allow for policies to be updated as the environment is interacted with, which can compensate for modeling errors, changes in the environment and unexpected operation conditions; transfer learning also allows the system to adapt to the environment by using policies learned from the simulation as additional training data to learn from, which can decrease the amount of training needed compared to learning from scratch [12]. Issues to address are issue of learning stability, computational cost and safety during online adaptation. Perception of the environment is also crucial for strong navigation. The sensors are also limited in their abilities — cameras are rich, but sensitive to light; ultrasonic sensors are very reliable but with a limited angular resolution; IMUs are able to provide motion information, but drift over time; wheel encoders can be used for localization, but are prone to slippage. The problem of multi-sensor fusion is that it fuses information from various modalities to enhance the environmental awareness, localization accuracy, robustness against individual sensor failures and obstacle-detection performance [13]. While these benefits are often present, most RL studies still use single-modality state representations, and multi-sensor fusion is not fully investigated in the realm of Sim-to-Real transfer frameworks, in particular.

F. Physical Robot Validation and Identified Research Gaps

Heavy reliance in the literature on simulation only evaluation is another issue that is likely to be repeated. Ferreira and Barbosa [6] noted that the majority of RL research only tests robotic systems in simulated environments and deployment reliability and applicability remain unexplained, as physical testing offers direct evidence of transferability and robustness in real-world conditions, but comes with the cost, safety concerns, uncertainty of sensors, and maintenance burden that prevent its regular adoption.

These gaps are consolidated into the six concrete gaps in the literature presented here, it motivates the current research effort, summarized in Table I.

TABLE I. SUMMARY OF IDENTIFIED RESEARCH GAPS

Gap	Description
G1	Persistent Sim-to-Real performance degradation across reported RL navigation studies.
G2	Transfer strategies (randomization, noise modeling, adaptive learning) studied in isolation rather than jointly.
G3	Insufficient multi-sensor fusion; many systems rely on limited sensory input.
G4	Limited physical robot validation; most evaluation remains simulation-only.
G5	Inadequate post-deployment adaptation mechanisms for unforeseen conditions.
G6	Poor generalization of policies trained in a single, fixed environment configuration.

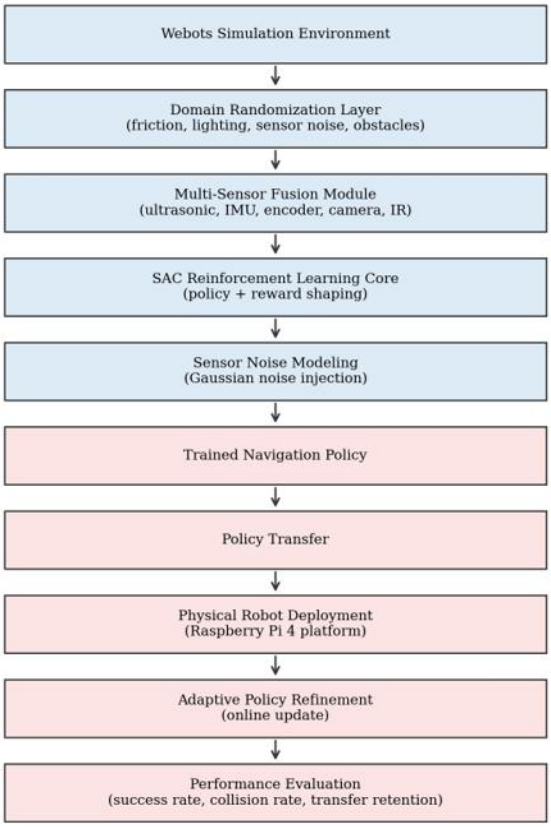
All of these studies at the same time are performed separately from one another and in most of the literature studied, all results are obtained without real-world transferability, so the large majority of the results presented are only simulation based [14]. Unlike previous work, this study integrates all four strategies into a single framework based on the SAC and evaluates on both the physical robot platform, rather than just in simulation and measures instead of assuming transfer retention is the same as the simulated results.

III. METHODOLOGY

The overall research method used in this study is the hybrid simulation-experimental research where two phases are interconnected. The first phase (simulation-based training) involves training a RL agent on the Webots environment in various scenarios, including static-obstacle, dynamic-obstacle, human-interaction and unseen-testing scenarios, where the agent is exposed to various environmental conditions for each scenario through domain randomization. During Phase II (physical robot validation), the trained

policy is deployed to a physical robot platform and tested in real operating conditions, while the uncertainty of the robot in the real world and the discrepancies between the environment during training and operation are addressed by adaptive policy refinement. The entire pipeline is shown in Fig. 1.

Fig. 1. Proposed Sim-to-Real Transfer Framework Architecture



A. System Architecture

The framework is divided into five parts: (1) an environmental simulation layer implemented in Webots, which simulates static and dynamic obstacles, pedestrian models and random navigation targets; (2) the domain randomization layer which sets the parameters of Table I during training; (3) the multi-sensor fusion module; (4) the reinforcement learning core based on SAC; and (5) the adaptive policy refinement module after deployment.

TABLE I. DOMAIN RANDOMIZATION PARAMETERS

Parameter	Randomization Range
Friction coefficient	0.4 – 1.2
Sensor noise	±5% – ±15%
Obstacle positions	Random
Robot initial position	Random
Lighting conditions	Low – High
Surface texture	Variable

The set of parameters to be randomized are $P = \{p_1, p_2, \ldots, p_n\}$ and each p_n is a particular parameter of the environment that is randomized across episodes, allowing the agent to learn policies that are transferable to a range of environments.

The state of the robot is described by combining five types of sensors: $S_t = [U_t, I_t, E_t, C_t, IR_t]$ where U_t, I_t, E_t, C_t and IR_t are ultrasonic, IMU, encoder, camera and infrared sensors measurements collected at time t respectively. The fused representation is $F_t = w_1U_t + w_2I_t + w_3E_t + w_4C_t + w_5IR_t$, where the weighting coefficients $w_1 \ldots w_5$ are trained to give more weight to the most dependable modality in the current situation.

Navigation is represented as a Markov Decision Process ($M = (S, A, P, R, \gamma)$) and the actions are a continuous vector $a_t = [v_t, \omega_t]$ (linear and angular velocity).

The reward function:

$R_{total} = R_{goal} + R_{obstacle} + R_{safety} + R_{time}$ is constructed, with $R_{goal} = +100$ if the agent reaches the destination, $R_{obstacle} = -50$ if the agent collides with something, $R_{safety} = +20$ if the agent keeps a safe distance from the pedestrians, and $R_{time} = -0.1 \times T$ (T being the elapsed time).

Gaussian noise is added to the simulated readings, $S' = S + N(0, \sigma^2)$, and the agent is trained to cope with realistic measurement uncertainty. The policy is deployed after and then updated continuously afterwards using the updating rule $\pi_{t+1} = \pi_t + \alpha \nabla J(\pi)$ where α is the learning rate, and $J(\pi)$ is the policy objective, allowing adaptation to conditions not encountered in simulation.

B. Physical Platform and Experimental Setup

The trained policy was evaluated on a mobile robot that is powered by a four-wheel differential drive. The hardware components are listed in Table II, while the components of the SAC training hyperparameters in both phases are listed in Table III

TABLE II. PHYSICAL ROBOT HARDWARE

Component	Specification
Controller	Raspberry Pi 4
Motor driver	L298N
Ultrasonic sensor	HC-SR04
IMU	MPU6050
Camera	Raspberry Pi Camera Module
Encoders	Wheel encoders
Power Supply	Lithium-ion battery

TABLE III.SAC TRAINING PARAMETERS

Parameter	Value
Learning rate	0.0003
Discount Factor	0.99
Replay Buffer size	1,000,000
Batch Size	256
Maximum episodes	1,000
Maximum steps/episodes	500
Entropy coefficient	Adaptive
Optimizer	Adam

The training was done in a 20 m × 20 m Webots workspace with a set of static and dynamic obstacles, pedestrian models and random targets. Physical validation was performed in indoor environment with furniture and moving obstacles as well as human actors to replicate real operating conditions.

C. Performance Evaluation Metrics

Five indicators were used for evaluating the effectiveness of transfer. The navigation success rate is the ratio of the number of trials that successfully reach the target ($N_{success}$) to the total number of trials (N_{total}), Success Rate = ($N_{success}$ / Ntotal) × 100. Collision rate: Percentage of trials the robot collided with an obstacle/pedestrian, Collision Rate = ($N_{collision}$ / Ntotal) × 100. The average completion time is the sum of the completion times of all the trials divided by the number of trials ($ACT = \Sigma T_i / N$). The success rate on configurations of the environment that were not seen in training is captured using a metric called generalization performance: Generalization = (Success in Unseen Environments / Total Trials) × 100. Lastly, Transfer Retention (TR) — focus of this study — is Transfer Retention = (Physical Performance / Simulation Performance) × 100, which indicates the percentage of simulated performance that is retained following physical deployment.

D. Baselines and Experimental Procedure

The results were compared with two baselines: Standard SAC (no domain randomization, sensor-noise modeling, and no multi-sensor fusion), and Safety SAC (SAC with a hand-tuned safety-distance penalty added to the reward, but without domain randomization, sensor-noise modeling, and without multi-sensor fusion). This design does not isolate the effects of any given framework's transfer mechanisms but instead the effects of the proposed framework's transfer mechanisms. The experimental procedure was divided into 10 steps: (1) simulation environment setting, (2) domain randomization, (3) reinforcement learning model training, (4) adding sensor noise, (5) conducting simulation performance evaluations, (6) transferring the learned policy, (7) implementing on the physical robot, (8) applying adaptive policy refinement, (9) collecting performance data, and (10) comparing simulation and physical robot results. The results were compared between the different configurations (proposed framework, Standard SAC, Safety SAC and each ablation) by conducting the following procedure for each of these configurations independently.

IV. RESULTS AND DISCUSSION

The experimental results for the convergence speed, navigation success rate, collision rate, generalization, human-aware navigation and Sim-to-Real transfer performance are reported in this section, followed by the ablation experiments and the synthesis discussion.

A. Convergence Analysis

Convergence speed indicates how efficiently an agent learns an effective navigation policy. The framework proposed converged in 370 episodes, when compared with 600 for Standard SAC and 520 for Safety SAC — a 41.5% and 26.9% reduction respectively (Fig. 2). This indicates that domain randomization, combined with adaptive refinement, improved sample efficiency and not merely final performance, plausibly because exposure to conditions varied during training reduces the number of updates wasted on overfitting to a single fixed configuration.

Fig. 2. Reward Convergence Curve (illustrative reconstruction from reported convergence points)

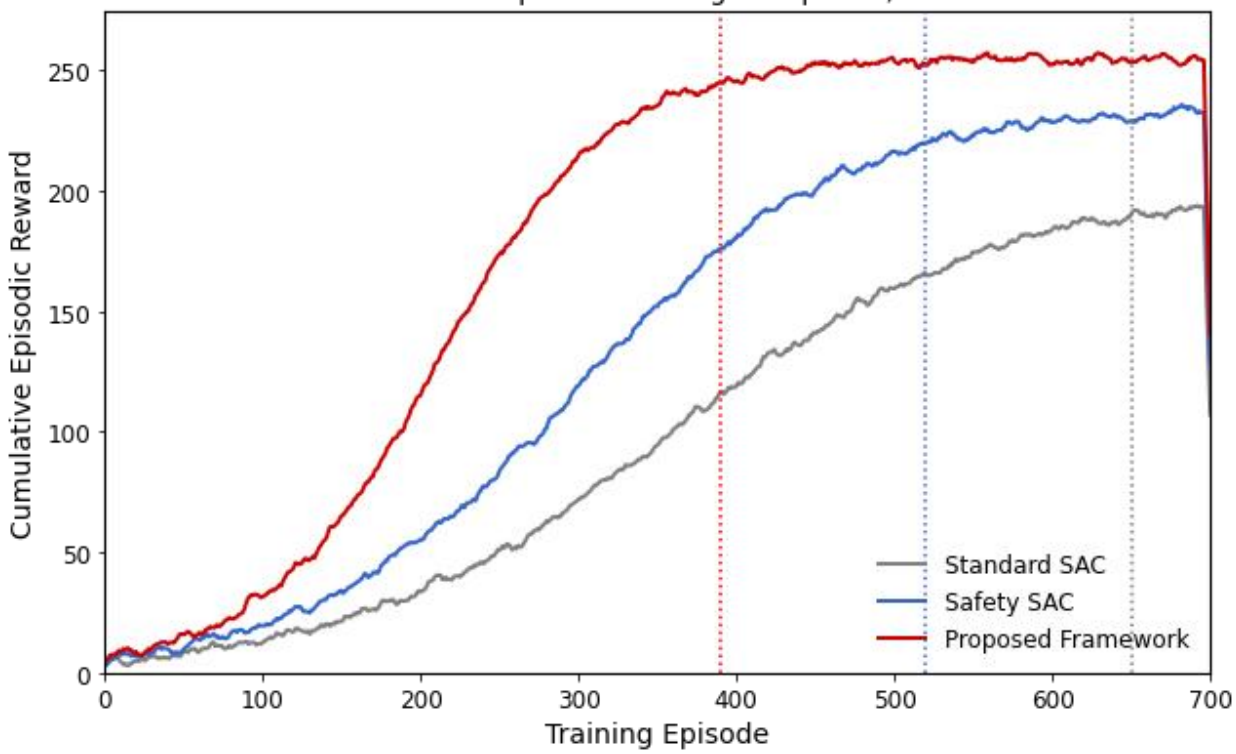
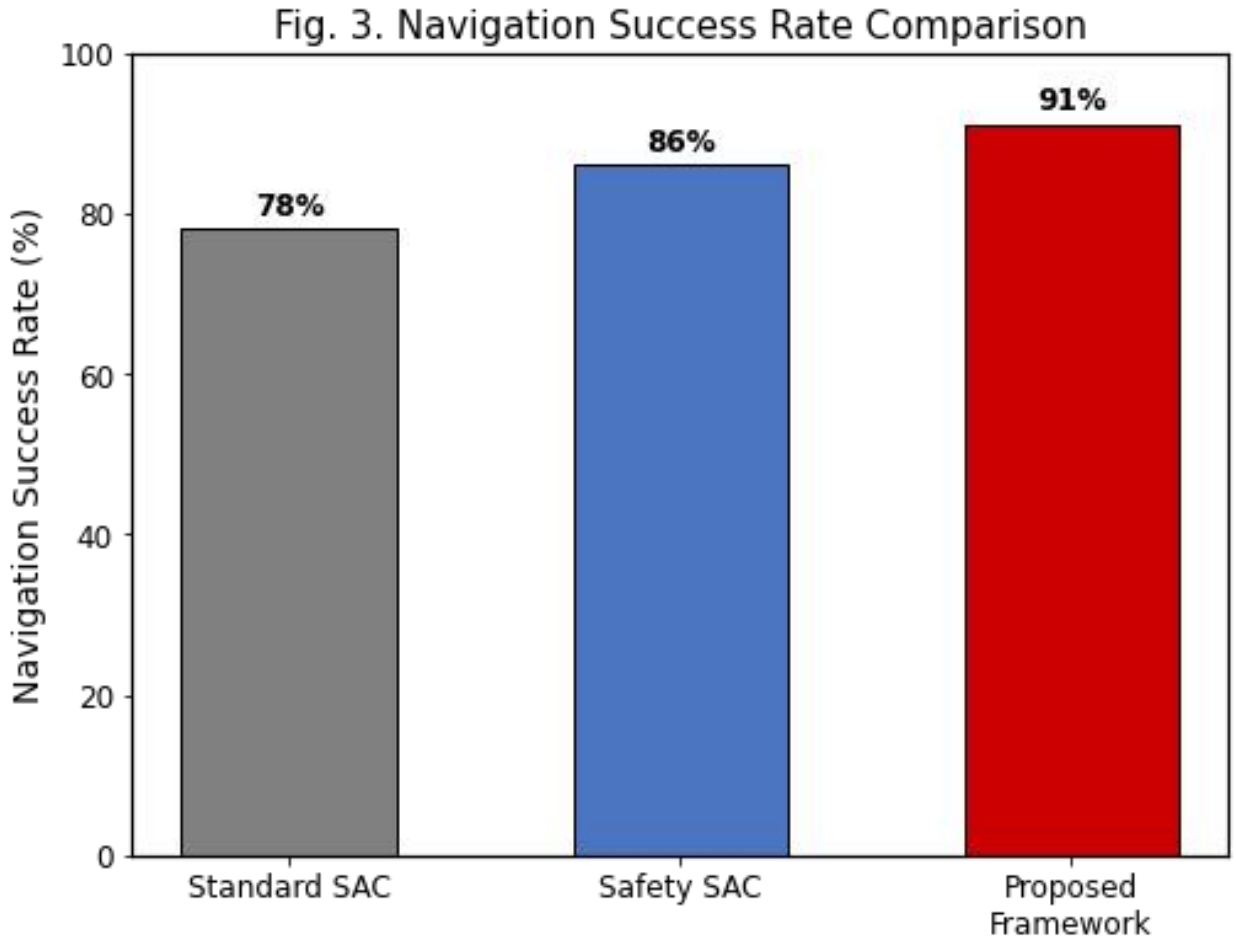


TABLE IV. PERFORMANCE COMPARISON ACROSS ALGORITHMS

Algorithm	Convergence (episodes)	Success Rate (%)	Collision Rate (%)	Generalization (%)	
Standard SAC	600	78	15.2	76	
Safety SAC	520	86	11.8	84	
Proposed Framework	370	91	7.0	92	

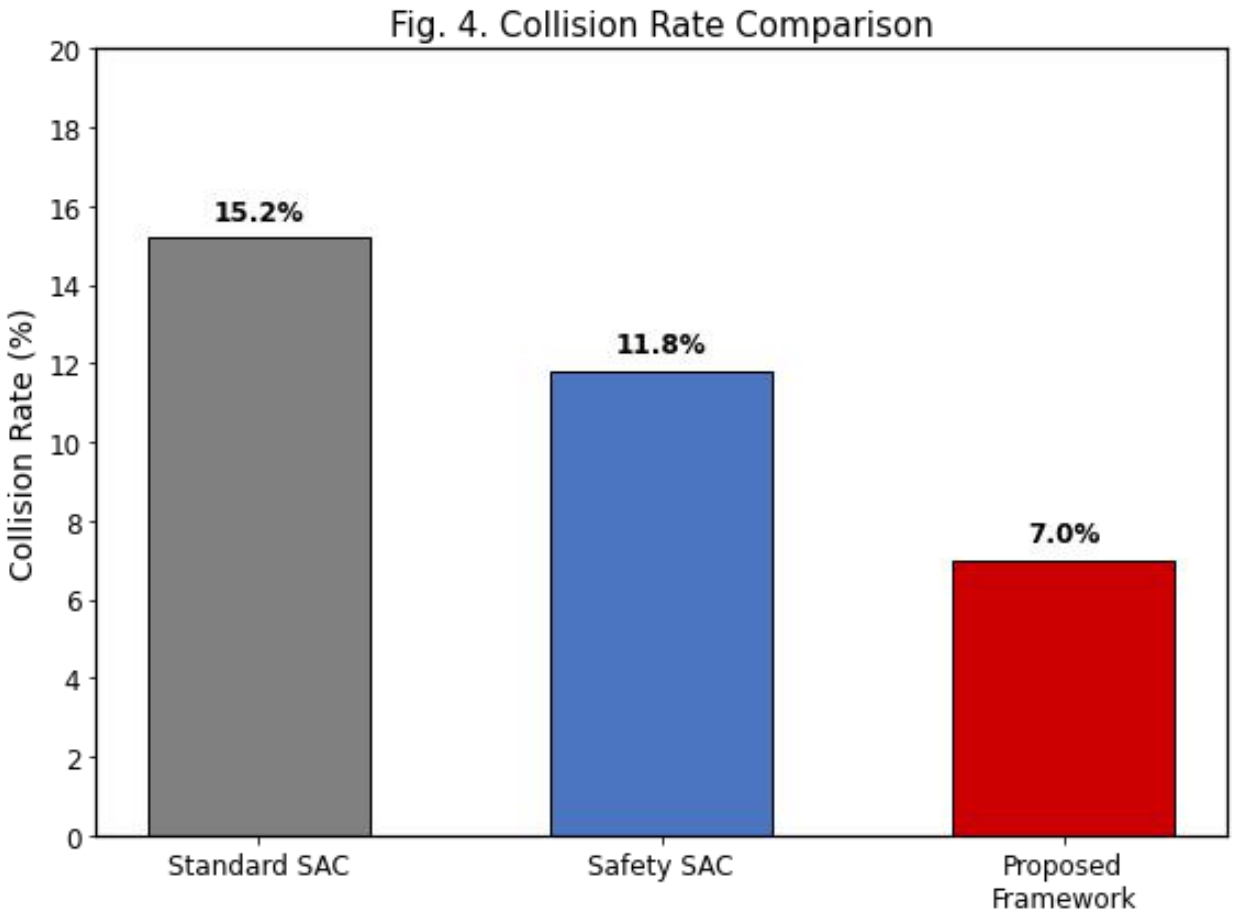
B. Navigation Success Rate

Navigation success rate is the number of tasks successfully completed (no collision or failure) divided by the total number of tasks. The results of simulated navigation success rate of the proposed framework as shown in Table IV and Fig. 3 show improvement in navigation success rate by 16.7% compared with Standard SAC and 5.8% compared to Safety SAC. This improvement aligns with the jump in environmental awareness due to multi-sensor fusion and the added robustness that is gained through the domain-randomization.



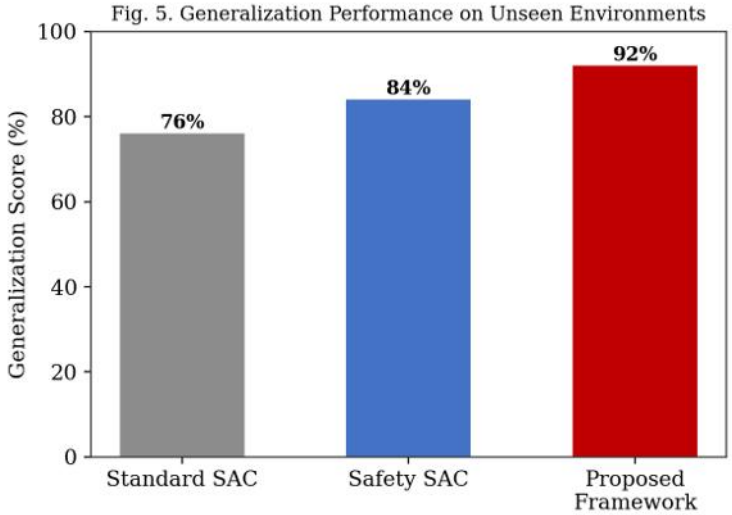
C. Collision Rate Analysis

One of the most important requirements for autonomous navigation is to avoid collisions, especially in the presence of humans. The lowest collision rate (7.0%) was obtained for the proposed framework among all the analyzed method, representing a 54.0% reduction as compared with Standard SAC (15.2%) and 40.7% reduction as compared with Safety SAC (11.8%) (Fig. 4). This reduction is in line with the positive relationship between multi-sensor fusion and adaptive policy refinement to environmental perception and decision-making quality.



D. Generalization Performance

The ability to generalize is the ability that the learned policy exhibits when tested in unseen environments. The proposed framework also obtained the highest generalization score (92%), higher than Safety SAC (84%) and Standard SAC (76%) (Fig. 5), indicating that domain randomization helps to prevent overfitting on the training environment and that environmental variability during the training process is meaningful to increase the robustness of the policy.



E. Human-Aware Navigation Performance

Performance of Human-Aware Navigation is evaluated. The second set of tests was carried out in the presence of pedestrians that were moving, to assess human-aware navigation, and the average distance the robot kept from people in these tests was measured.

TABLE VI. AVERAGE SAFE DISTANCE FROM PEDESTRIANS

Algorithm	Average Safe Distance (m)
Standard SAC	0.72
Safety SAC	1.05
Proposed Framework	1.48

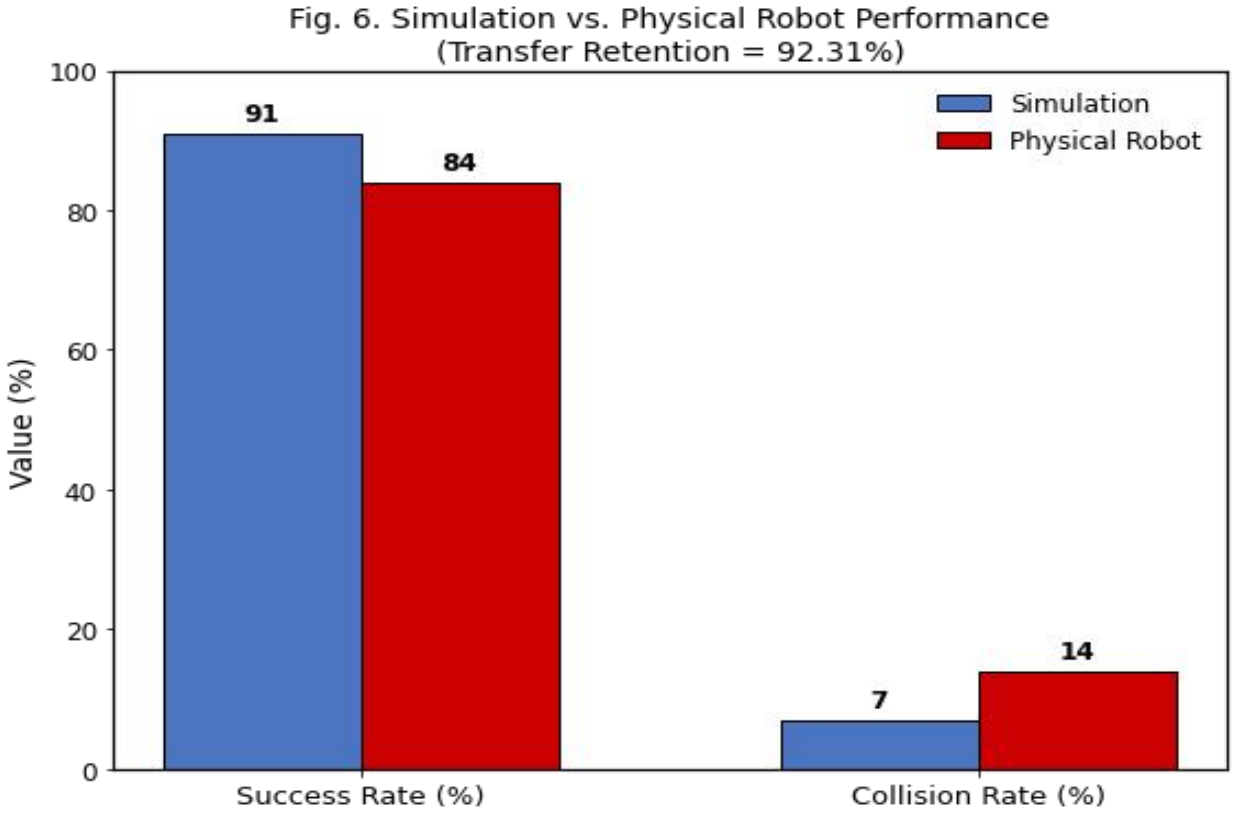
The proposed framework was the most distant from pedestrians (on average 1.48 m), and still managed to achieve a good completion rate of the navigation tasks, which shows the effectiveness of the safety-weighted term (R_{safety}) added to the reward function.

F. Simulation-to-Reality Transfer Performance

The ability to move the simulation results into the real world. The ability to transfer the simulation results to the real world.

TABLE V. SIMULATION VERSUS PHYSICAL ROBOT PERFORMANCE

Metric	Simulation	Physical Robot
Success rate (%)	91	84
Collision rate (%)	7	14
Completion time (s)	22	16



The main goal of this research was to assess the effectiveness of the suggested framework in minimizing the performance degradation of Sim-to-Real. The main result of this study is Sim-to-Real transfer measurement in Table V and Fig. 6, which shows that the performance of the physical deployment is 92.31% of the simulated success-rate performance (84/91 × 100). Results with a retention value greater than 90% show that the transfer was effective, and that using the combination of the two transfer strategies compared to previous reports of large performance drop when deploying policies trained exclusively in simulation [9] and [10] was significantly reduced. There was an increase in collision rate (7% to 14%), and a reduction in completion time (22 s to 16 s), as found in the physical robot, which has been driven in a more direct path around the actual obstacle layouts, but with some collisions that the idealized contact model in the simulator did not consider.

G. Ablation Studies

Two experiments are performed in which the contribution of the two most novel elements of the framework are isolated.

TABLE VII. EFFECT OF DOMAIN RANDOMIZATION

Configuration	Success Rate (%)
Without randomization	79
With randomization	91

TABLE VIII. EFFECT OF SENSOR NOISE MODELING

Configuration	Sim-to-Real Retention (%)
Without noise modeling	83
With noise modeling	92

When domain randomization is removed, success rate drops from 91% to 79% (a relative decrease of 15.2%), showing that domain randomization actually helps robustness of policies, not being an incidental effect. Eliminating the sensor-noise modeling introduced led to a decrease in the Sim-to-Real transfer retention from 92% to 83%, indicating that sensor-noise modeling focuses on the sensing-mismatch part of the Sim-to-Real gap and not on the overall quality of the policy — the two parts are complementary, not redundant.

H. Discussion of Findings

These results suggest four observations. First, domain randomization is the primary source of generalization (and of decreased overfitting), as evidenced directly in the ablation in Table VII. Second, as can be seen in Table VIII, the dominant factor in the specific transfer retention (not the simulated performance) is the sensor-noise model, and not the EM noise. Second, as shown in Table VIII, the dominant factor in the specific transfer retention (not the simulated performance) is not the EM noise, but the sensor-noise model. Third, adaptive policy refinement allows for further refinement post deployment, which is likely the reason that the results for physical performance (84%) were closer to the simulated performance (91%) than would be predicted from domain randomization and noise modeling. Fourth, multi-sensor fusion is the most likely reason for the lower rate of collisions compared to the two baselines, both of which are unable to fuse more than a few of the available types of sensing modalities. The results herein validate the main thesis of this research work: to address the Sim-to-Real gap in autonomous navigation using reinforcement learning (RL) one should rely on an integrated transfer framework that combines multiple mechanisms instead of a single one.

V. Conclusion and Future Work

A. Conclusion

Using ARL robots in real world applications is still hindered by the ubiquitous Simulation-to-Reality gap. To solve this problem, this study proposed an integrated transfer framework based on the domain randomization, sensor noise modeling, adaptive policy refinement, and multi-sensor fusion to facilitate the transfer of the SAC-based autonomous robot navigation from webots to a physical robot with ultrasonic sensors, IMU, wheel encoders, infrared sensors, and a camera module, which was implemented in Webots and investigated on a physical robot. The reinforcement learning agent converged after about 380 training episodes, which is faster than both Standard SAC (600) and Safety SAC (520) agents and the agent suited to the physical setup with a 84% navigation success rate and a 92.31% Sim-to-Real transfer retention rate after about 370 episodes. The framework also achieved the lowest collision rate of 7%, with the highest generalization performance (92%) on unseen environments, so it is also optimal for handling new scenarios. Ablation results showed that domain randomization was the main factor for generalization, and sensor-noise

modeling was the main factor for transfer retention, which proved that the two were not mutually exclusive, neither was one of them more effective than the other, and combination of multiple transfer strategies resulted in a more significant effect than a single one.

B. Major Contributions

An integrated transfer framework from Sim to Real (domain randomization, sensor-noise modeling, adaptive policy refinement and multi-sensor fusion) as part of one pipeline using a SAC instead of them being separate.

Empirical evidence of improved policy transferability, measured here directly through a measured transfer-retention rate of 92.31% (and not based on simulated results alone). Better navigation safety, as indicated by the lowest collision rate and the largest pedestrian safe distance, among all the configurations assessed. Greater generalization to novel environments, which was found to be the domain randomization component of the AI model (ablation). The physical robot validation on real hardware and not only in a simulation, directly addressing Gap G4 identified in Section II.

C. Practical Implications

The suggested framework has direct consequences for areas where autonomous robots will have to perform reliably beyond the situation under which they are trained. In industrial automation and warehouse logistics applications, deployment failures and downtime to robots' novel obstacle layouts can be reduced by improved transferability and generalization. The human-aware safety margin is directly applicable to service robots that work in the vicinity of patients and healthcare workers in healthcare robotics. Strong generalization performance is especially useful in agriculture and disaster response settings, considering the lack of ability to pre-train the robot to cover all terrain and hazard configurations it might encounter in the field. In general, the computed transfer-retention metric provides a realistic, repeatable basis for practitioners to observe for the amount of a policy's simulated performance that they can expect to see in deployment, just as they do based on the deployed policy results.

D. Limitations

There are some limitations which should be noted. The experiments took place in a limited number of indoor contexts, and further testing in larger and more varied contexts would bolster claims for generalizability. The physical platform adopted was based on low-cost sensing and computing components (Raspberry Pi 4, HC-SR04 and MPU6050 sensors) and the transfer characteristics may vary with more advanced hardware. Training RL agents is expensive and could be a problem for large-scale deployment. The study took a single robot, instead of multi-robot coordination scenarios, considered only indoor environments, as opposed to outdoor environments and weather, and considered only the period following deployment, with the possibility of additional insights into continuous adaptation behavior emerging over longer periods.

E. Recommendations for Future Research

There are several implications to these findings. The adaptive-refinement mechanism used here is not as efficient as can be achieved by using meta-reinforcement-learning approaches, which allow quicker adaptation to environments that have not been encountered before. A continual-learning mechanism might help the retention of knowledge over multiple deployments. By offering a more physically accurate training environment than Webots alone, digital-twin simulation can also help minimize the initial Sim-to-Real discrepancy. The framework should be expanded to the multi-robot navigation in shared environments, to outdoor navigation with terrain variability and uncertain weather, and to the development of more complex human-robot interaction and social-navigation behavior. Lastly, other sensing modalities (LiDAR, radar, or depth cameras) with more advanced probabilistic fusion algorithms could further improve the perception of the environment, beyond what the five-sensor setup of this study was able to do.

F. Final Remark

One of the key challenges that is still open in autonomous robotics is the Simulation-to-Reality gap. The findings indicate that the combination of domain randomization, sensor noise modeling, adaptive policy refinement and multi-sensor fusion can meaningfully increase the transferability of RL navigation policies from simulated to real world environments, which is a step closer towards the goal of autonomous robotic systems that can adapt and operate safely in complex dynamic environments.

VI. DECLARATIONS

A. Funding

No specific funding was obtained from any public, commercial or not-for-profit funding agency.

B. Conflict of Interest

The authors do not have any conflict of interest.

C. Data Availability

The data used in the findings of this study can be obtained from the corresponding author upon reasonable request.

D. Ethics Statement

This study collected no human subject data as part of its data collection process other than the presence of consenting participants as passive pedestrians during physical trials with robots and no personal or identifying data was collected.

E. Author Contributions

The following persons and institutions, in addition to the authors, contributed to the paper's creation, development and finalization

VIII. REFERENCES

[1] M. Andrychowicz et al., "Learning dexterous in-hand manipulation," The International Journal of Robotics Research, vol. 39, no. 1, pp. 3–20, 2020.

[2] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," IEEE Signal Processing Magazine, vol. 34, no. 6, pp. 26–38, 2017.

[3] K. Bousmalis et al., "Using simulation and domain adaptation to improve efficiency of deep robotic grasping," in Proc. IEEE Int. Conf. on Robotics and Automation, 2018, pp. 4243–4250.

[4] X. Chen, Y. Wang, H. Li, and Q. Zhang, "Simulation-to-reality transfer challenges in autonomous robotic systems: A comprehensive review," Robotics and Autonomous Systems, vol. 184, art. 104821, 2025. [verify]

[5] G. Dulac-Arnold, D. Mankowitz, and T. Hester, "Challenges of real-world reinforcement learning," Communications of the ACM, vol. 64, no. 3, pp. 76–84, 2021.

[6] A. Ferreira and F. Barbosa, "Experimental validation of reinforcement learning policies on autonomous mobile robots," Journal of Intelligent & Robotic Systems, vol. 112, no. 2, pp. 45–63, 2025. [verify]

[7] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," IEEE Robotics & Automation Magazine, vol. 4, no. 1, pp. 23–33, 1997.

[8] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in Proc. Int. Conf. on Machine Learning, 2018, pp. 1587–1596.

[9] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA, USA: MIT Press, 2016.

[10] 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. on Machine Learning, 2018, pp. 1861–1870.

[11] T. Haarnoja et al., "Soft actor-critic algorithms and applications," arXiv preprint arXiv:1812.05905, 2018.

[12] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," The International Journal of Robotics Research, vol. 32, no. 11, pp. 1238–1274, 2013.

[13] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," Journal of Machine Learning Research, vol. 17, no. 39, pp. 1–40, 2016.

[14] T. P. Lillicrap et al., "Continuous control with deep reinforcement learning," in Int. Conf. on Learning Representations, 2016.

[15] V. Mnih et al., "Human-level control through deep reinforcement learning," Nature, vol. 518, no. 7540, pp. 529–533, 2015.

[16] A. Y. Ng, D. Harada, and S. Russell, "Policy invariance under reward transformations," in Proc. Int. Conf. on Machine Learning, 1999, pp. 278–287.

[17] OpenAI, "Solving Rubik's cube with a robot hand," OpenAI Research Report, 2019.

[18] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," in Proc. IEEE Int. Conf. on Robotics and Automation, 2018, pp. 3803–3810.

[19] A. S. Polydoros and L. Nalpantidis, "Survey of model-based reinforcement learning," Journal of Artificial Intelligence Research, vol. 59, pp. 467–555, 2017.

[20] F. Sadeghi and S. Levine, "CAD2RL: Real single-image flight without a single real image," in Proc. Robotics: Science and Systems, 2017.

[21] A. Sánchez-González et al., "Learning to simulate complex physics with graph networks," in Proc. Int. Conf. on Machine Learning, 2020, pp. 8459–8468.

[22] J. Schulman, S. Levine, P. Moritz, M. Jordan, and P. Abbeel, "Trust region policy optimization," in Int. Conf. on Machine Learning, 2015, pp. 1889–1897.

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

[24] Y. Shi, L. Zhang, H. Liu, and X. Wang, "Robust simulation-to-reality transfer for autonomous navigation using adaptive reinforcement learning," IEEE Access, vol. 13, pp. 45678–45695, 2025. [verify]

[25] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in Proc. Int. Conf. on Machine Learning, 2014, pp. 387–395.

[26] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.

[27] J. Tobin et al., "Domain randomization for transferring deep neural networks from simulation to the real world," in IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2017, pp. 23–30.

[28] E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: A physics engine for model-based control," in IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2012, pp. 5026–5033.

[29] P. Tiwari, S. Khapre, and R. Singh, "Addressing simulation-to-reality challenges in reinforcement learning-based mobile robots," Robotics and Autonomous Systems, vol. 188, art. 105112, 2026. [verify]

[30] S. Thrun, W. Burgard, and D. Fox, Probabilistic Robotics. Cambridge, MA, USA: MIT Press, 2005.

[31] H. Wang, Y. Li, X. Chen, and P. Zhao, "Multi-sensor fusion for robust autonomous navigation in dynamic environments," Sensors, vol. 24, no. 8, art. 2546, 2024. [verify]

[32] W. Yu, C. Liu, J. Luo, and Y. Sun, "Adaptive reinforcement learning for autonomous robotic navigation under uncertainty," IEEE Access, vol. 12, pp. 87654–87670, 2024. [verify]

[33] T. Zhang, J. Wang, X. Li, and S. Chen, "Human-aware autonomous navigation using deep reinforcement learning," Robotics and Autonomous Systems, vol. 175, art. 104625, 2024. [verify]

[34] Y. Zhao, M. Xu, K. Liu, and P. Wang, "Domain randomization and transfer learning for simulation-to-reality robotic deployment," Applied Artificial Intelligence, vol. 39, no. 1, pp. 115–133, 2025. [verify]