

# **Exploring the Feasibility of Sim2Real Transfer in Reinforcement Learning**

*Lucas Driessens*



Bachelor Multimedia and Creative Technologies

University of Applied Sciences

Howest Kortrijk

2024

## Abstract

The transition of reinforcement learning (RL) applications from simulated to real-world environments is fraught with challenges, particularly in the context of autonomous navigation. This thesis investigates the feasibility of sim-to-real transfer by training an RL agent to navigate a maze using a remote-controlled car, focusing on the core question: Can a trained RL agent effectively adapt from a virtual setting to a physical one? This study evaluates various virtual environments and RL algorithms to identify the most effective training platform and techniques. It also explores practical challenges and potential solutions for sim-to-real transfer. The findings aim to provide new insights that enhance the practical implementation of RL in real-world applications, with implications across robotics and beyond. This research not only bridges theoretical models and practical applications but also sets the stage for further exploration into the robust application of RL in dynamic real-world scenarios.

## Table of Contents

Glossary of Terms . . . . .	3
List of Abbreviations . . . . .	4
Introduction . . . . .	4
Background on Reinforcement Learning . . . . .	5
Research Questions . . . . .	5
Main Research Question . . . . .	6
Sub Research Questions . . . . .	6
Methodology . . . . .	6
Environment Setup (RCMazeEnv) . . . . .	6
Agent Design (DDQNAgent) . . . . .	7
Training Process . . . . .	7
Reward Function Components . . . . .	7
Scope of Real-World Testing . . . . .	9
Experimental Outcomes and Implementation Details . . . . .	10
Virtual Environment and Agent Design . . . . .	10
Implementation Highlights . . . . .	10
Outcome Analysis and Future Directions . . . . .	11
Model Architecture and Training Insights . . . . .	11
Training Parameters . . . . .	12
Training Procedure . . . . .	12
Visual Insights and Further Exploration . . . . .	13
Visual Tools and Their Impact . . . . .	14
Evaluation Metrics Overview . . . . .	15
Concluding Insights . . . . .	16
Answers to Research Questions . . . . .	16
1. Virtual Environments for RF-Car Training . . . . .	16
2. Reinforcement Learning Techniques for Virtual RF-Car Training . . . . .	17
3. Sim-to-Real Transfer Challenges and Solutions . . . . .	17
4. Contributions of Simulation in RF-Car Training . . . . .	18
5. Practical Application of Simulated Training to Real-World RF-Cars . . . . .	18
Results . . . . .	18
Reinforcement Learning Techniques Overview . . . . .	18
Conclusion . . . . .	25
Hardware Setup and Assembly . . . . .	26
Introduction to Hardware Components . . . . .	26
Components List . . . . .	28

Wiring Guide . . . . .	28
Challenges and Solutions in Implementing RL Techniques and Virtual Environments . . . . .	29
Challenge 1: Selection of an Appropriate Virtual Environment . . . . .	29
Challenge 2: Optimal Reinforcement Learning Technique . . . . .	29
Challenge 3: Sim2Real Transfer - Addressing Movement Discrepancies . . . . .	29
Challenge 4: Precise Straight-Line Movement and Alignment . . . . .	29
Challenge 5: Ensuring Consistent and Effective Training . . . . .	30
Challenge 6: Accurate Sensor Data Normalization for Sim2Real Transfer . . . . .	30
Challenge 7: Integration of Failsafe Mechanisms . . . . .	30
Challenge 8: Training Environment and Technique Efficacy . . . . .	31
Conclusion . . . . .	31
Viewing Practical Experiments . . . . .	31
Real-World Application and Limitations . . . . .	31
Introduction to Sensor and Movement Discrepancies . . . . .	31
Real-World Application . . . . .	32
Limitations . . . . .	32
Conclusion . . . . .	33
Comparative Analysis with Existing Sim-to-Real Studies . . . . .	33
Introduction . . . . .	33
Overview of Referenced Studies . . . . .	33
Methodological Comparisons . . . . .	34
Outcome and Effectiveness Comparisons . . . . .	34
Implications for Current Research . . . . .	34
Conclusion . . . . .	35
Reflection . . . . .	35
Strengths and Weaknesses . . . . .	35
Practical Applicability and Industry Relevance . . . . .	35
Encountered Alternatives and Flexibility . . . . .	36
Anticipated Implementation Barriers . . . . .	36
Ethical Considerations . . . . .	36
Societal Impact . . . . .	36
Policy and Regulation . . . . .	37
Lessons Learned and Forward Path . . . . .	37
Advice for those Embarking on Similar Research Paths . . . . .	37
General Conclusion . . . . .	38
Credits . . . . .	39
Supplementary Materials: Video Demonstrations . . . . .	39
Addressing Alignment and Orientation Challenges . . . . .	39

Enhancing Movement Precision with Encoders . . . . .	40
Real-World Application Tests . . . . .	40
Guest Speakers . . . . .	41
Innovations and Best Practices in AI Projects by Jeroen Boeye at Faktion . . . . .	41
Pioneering AI Solutions at Noest by Toon Vanhoutte . . . . .	42
Installation Steps . . . . .	43
Prerequisites . . . . .	43
Repository Setup . . . . .	43
ESP32 Setup . . . . .	43
Web Application Setup . . . . .	44
Usage Instructions . . . . .	44
Additional Information: Model Training . . . . .	44
References . . . . .	45

## Glossary of Terms

1. **Artificial Intelligence (AI)**: The simulation of human intelligence processes by machines, especially computer systems, enabling them to perform tasks that typically require human intelligence.
2. **Double Deep Q-Network (DDQN)**: An enhancement of the Deep Q-Network (DQN) algorithm that addresses the overestimation of action values, thus improving learning stability and performance.
3. **Epsilon Decay**: A technique in reinforcement learning that gradually decreases the rate of exploration over time, allowing the agent to transition from exploring the environment to exploiting known actions for better outcomes.
4. **Mean Squared Error (MSE)**: A loss function used in regression models to measure the average squared difference between the estimated values and the actual value, useful for training models by minimizing error.
5. **Motion Processing Unit (MPU6050)**: A sensor device combining a MEMS (Micro-Electro-Mechanical Systems) gyroscope and a MEMS accelerometer, providing comprehensive motion processing capabilities.
6. **Policy Network**: In reinforcement learning, a neural network model that directly maps observed environment states to actions, guiding the agent's decisions based on the current policy.
7. **Raspberry Pi (RPI)**: A small, affordable computer used for various programming projects, including robotics and educational applications.
8. **RC Car**: A remote-controlled car used as a practical application platform in reinforcement learning experiments, demonstrating how algorithms can control real-world vehicles.
9. **Reinforcement Learning (RL)**: A subset of machine learning where an agent learns to make decisions by taking actions within an environment to achieve specified goals, guided by a system of rewards and penalties.
10. **Sim2Real Transfer**: The practice of applying models and strategies developed within a simulated environment to real-world situations, crucial for bridging the gap between theoretical research and practical application.
11. **Target Network**: Utilized in the DDQN framework, a neural network that helps stabilize training by providing consistent targets for the duration of the update interval.
12. **Virtual Environment**: A simulated setting designed for training reinforcement learning agents, offering a controlled, risk-free platform for experimentation and learning.

## List of Abbreviations

1. **AI** - Artificial Intelligence
2. **DDQN** - Double Deep Q-Network
3. **DQN** - Deep Q-Network
4. **ESP32** - Espressif Systems 32-bit Microcontroller
5. **HC-SR04** - Ultrasonic Distance Sensor
6. **MSE** - Mean Squared Error
7. **MPU6050** - Motion Processing Unit (Gyroscope + Accelerometer)
8. **PPO** - Proximal Policy Optimization
9. **RC** - Remote Controlled
10. **RPI** - Raspberry Pi
11. **RL** - Reinforcement Learning
12. **RCMazeEnv** - RC Maze Environment (Custom Virtual Environment for RL Training)
13. **Sim2Real** - Simulation to Reality Transfer

## Introduction

As the realms of artificial intelligence (AI) and robotics evolve, the lines between virtual simulations and real-world applications continue to blur, creating both unparalleled opportunities and formidable challenges. This thesis investigates the potential of Reinforcement Learning (RL) to effectively bridge this gap, with a particular focus on autonomous navigation using a remote-controlled (RC) car navigating a maze. The core challenge addressed is the sim-to-real transfer—how can an RL agent trained in a virtual environment adapt effectively to a real-world setting? This question is pivotal for leveraging the full potential of RL in complex, real-world scenarios.

The aim of this study is to explore the feasibility and inherent challenges of the sim2real transition, specifically within the context of maze navigation using an RC car. The significance of this research extends beyond academic interest, proposing practical solutions to integrate theoretical RL models with real-world applications—a crucial advancement in AI and robotics.

This research draws inspiration from dynamic real-world applications of RL, including micro mouse competitions, and leverages insights from influential digital platforms. For instance, strategies from the YouTube demonstration “Self Driving and Drifting RC Car using Reinforcement Learning” [11] and the project “Reinforcement Learning with Multi-Fidelity Simulators – RC Car” [16] significantly influence the experimental approach adopted in this study. These examples highlight the practical challenges and innovative solutions within the RL domain, providing a solid foundation for this research.

Further depth is added by technical documentation and seminal academic literature that elucidate the capabilities of RL in navigating complex mazes and handling autonomous tasks. Notable contributions

include a detailed Medium article by M. A. Dharmasiri [15] and a comprehensive survey on sim-to-real transfer in deep reinforcement learning for robotics by W. Zhao, J. P. Queraltà, and T. Westerlund [17]. These works underpin the methodological framework and experimental design of this thesis.

By integrating a diverse array of sources with practical experiments, this study aims to offer a thorough examination of the sim-to-real transfer process. The blend of digital insights, theoretical underpinnings, and empirical data creates a multi-dimensional perspective on the challenges and solutions in applying RL to real-world scenarios, setting a detailed platform for exploring autonomous RC car navigation within a maze environment.

## **Background on Reinforcement Learning**

Reinforcement Learning (RL) employs a computational approach where agents learn to optimize their action sequences through trials and errors, engaging with their environment to maximize accumulated rewards over time. This learning framework is built upon the foundation of Markov Decision Processes (MDP), which includes:

- $S$ : a definitive set of environmental states,
- $A$ : a comprehensive set of possible actions for the agent,
- $P(s_{t+1}|s_t, a_t)$ : the transition probability that signifies the chance of moving from state  $s_t$  to state  $s_{t+1}$  after the agent takes action  $a_t$  at a given time  $t$ ,
- $R(s_t, a_t)$ : the reward received following the action  $a_t$  from state  $s_t$  to state  $s_{t+1}$ .

The principles of Reinforcement Learning, particularly the dynamics of Markov Decision Processes involving states  $S$ , actions  $A$ , transition probabilities  $P(s_{t+1}|s_t, a_t)$ , and rewards  $R(s_t, a_t)$ , form the foundation of how agents learn from and interact with their environment to optimize decision-making over time. This understanding is crucial in the development of autonomous vehicles, improving navigational strategies, decision-making capabilities, and adaptation to real-time environmental changes. The seminal work by R.S. Sutton and A.G. Barto significantly elucidates these principles and complexities of RL algorithms [18].

## **Research Questions**

This investigation is anchored by the question: “Can a trained RL agent be effectively transferred from a simulation to a real-world environment for maze navigation?” Addressing this question involves exploring multiple facets of RL training and implementation:

1. Selection of virtual environments for effective RL training.
2. Identification of RL techniques suited for autonomous navigation.

3. Evaluation of sim-to-real transfer in adapting to real-world dynamics.
4. Assessment of training efficacy and performance optimization through simulation.
5. Adaptation and transfer of a trained model to a real RC car, including necessary adjustments for real-world application.

A combination of qualitative and quantitative research methodologies underpins this study, encompassing simulation experiments, real-world trials, and an extensive review of existing literature. This multifaceted strategy not only seeks to corroborate the effectiveness of transferring simulations to real-world applications but also endeavors to enrich the ongoing conversation regarding the practical implementation and obstacles associated with Reinforcement Learning (RL).

### **Main Research Question**

**Is it possible to transfer a trained RL-agent from a simulation to the real world? (case: maze)**

### **Sub Research Questions**

1. Which virtual environments exist to train a virtual RC-car?
2. Which reinforcement learning techniques can I best use in this application?
3. Can the simulation be transferred to the real world? Explore the difference between how the car moves in the simulation and in the real world.
4. Does the simulation have any useful contributions? In terms of training time or performance?
5. How can the trained model be transferred to the real RC car? (sim2real) How do you need to adjust the agent and the environment for it to translate to the real world?

### **Methodology**

This section outlines the Reinforcement Learning Maze Navigation (RCMazeEnv) methodology, utilizing a Double Deep Q-Network (DDQNAgent) architecture. It provides detailed descriptions of the maze environment setup, the agent design, and a comprehensive training algorithm, incorporating mathematical functions to clarify the system's mechanics.

### **Environment Setup (RCMazeEnv)**

The RCMazeEnv is a custom maze navigation environment derived from the OpenAI Gym framework, designed specifically for a 12x12 cell grid maze navigation task. Each cell is designated as either a wall

(1) or a path (0), with the goal located at cell position (10, 10). The agent, visualized as an RC car, starts its journey from cell (1, 1), initially facing east. The agent's possible actions include moving forward, turning left, and turning right, enabled by sensors that provide distance readings in three directions: front, left, and right. These sensors are crucial for navigation, offering real-time environmental data for decision-making.

### Agent Design (DDQNAgent)

The agent's architecture is based on a Double Deep Q-Network (DDQN), enhancing the standard DQN by reducing overestimation of Q-values—a common issue in reinforcement learning. The DDQN consists of:

- **Policy Network:** Calculates the Q-value  $Q(s, a; \theta)$  for action  $a$  in state  $s$ , using parameters  $\theta$ . This network drives action selection based on the current policy.
- **Target Network:** Uses parameters  $\theta^-$  to provide a stable target Q-value for policy updates, helping mitigate rapid policy changes that could destabilize learning.

The core equation for the DDQN update is:

$$Y_t^{DDQN} = R_{t+1} + \gamma Q\left(S_{t+1}, \operatorname{argmax}_a Q(S_{t+1}, a; \theta); \theta^-\right),$$

where  $\gamma$  represents the discount factor, emphasizing the importance of future rewards.

### Training Process

Training incorporates an experience replay mechanism to enhance learning stability and efficiency. This process involves storing state transitions  $(s, a, r, s')$  in a replay buffer and periodically updating the model by minimizing the loss function:

$$L(\theta) = \mathbb{E}_{(s, a, r, s') \sim U(D)} \left[ \left( r + \gamma \max_{a'} Q(s', a'; \theta^-) - Q(s, a; \theta) \right)^2 \right].$$

The epsilon-greedy strategy controls exploration, ensuring the agent does not settle prematurely on suboptimal policies.

### Reward Function Components

**Collision Penalty  $R_{\text{collision}}$**  When the agent attempts to move into a wall or outside the designated maze boundaries, it triggers a collision state. To discourage such actions, which are counterproductive

to the goal of reaching the destination, a significant penalty is applied. This penalty is critical for teaching the agent about the boundaries and obstacles within the environment, ensuring that it learns to navigate safely and effectively.

$$R_{\text{collision}} = -20$$

**Goal Achievement Bonus  $R_{\text{goal}}$**  Reaching the goal is the primary objective of the maze navigation task. A substantial reward is given to the agent upon achieving this objective, signifying the completion of the episode. This reward serves as a strong positive reinforcement, guiding the agent's learning towards the goal-oriented behavior. However, an additional mechanism penalizes the agent if it takes an excessively long route to reach the goal, promoting efficiency in navigation.

$$R_{\text{goal}} = \begin{cases} +500, & \text{if goal is reached} \\ -200, & \text{if steps} > 1000 \end{cases}$$

**Proximity Reward  $R_{\text{proximity}}$**  This component of the reward function incentivizes the agent to minimize its distance to the goal over time. By rewarding the agent based on its proximity to the goal, it encourages exploration and path optimization, guiding the agent to navigate the maze more effectively. The reward decreases as the distance to the goal increases, encouraging the agent to always move towards the goal.

$$R_{\text{proximity}} = \frac{50}{d_{\text{goal}} + 1}$$

**Progress Reward  $R_{\text{progress}}$**  The progress reward or penalty is designed to encourage the agent to make decisions that bring it closer to the goal and to penalize decisions that lead it away. This dynamic reward system provides immediate feedback based on the agent's movement relative to the goal, promoting smarter navigation decisions.

$$R_{\text{progress}} = \begin{cases} +50, & \text{if distance decreases} \\ -25, & \text{if distance increases} \end{cases}$$

**Exploration Penalty  $R_{\text{revisit}}$**  To discourage repetitive exploration of the same areas, which indicates inefficient pathfinding, the agent receives a penalty for re-entering previously visited cells. This penalty is crucial for encouraging the exploration of new paths and preventing the agent from getting stuck in loops or dead ends.

$$R_{\text{revisit}} = -10$$

**Efficiency Penalty  $R_{\text{efficiency}}$**  Every step the agent takes incurs a small penalty. This mechanism ensures that the agent is incentivized to find the shortest possible path to the goal, balancing the need to explore the environment with the goal of reaching the destination as efficiently as possible.

$$R_{\text{efficiency}} = -5$$

**Termination conditions** To ascertain whether the environment has reached a “done” or “ended” state, several conditions are established. These conditions include: surpassing 3000 steps, the car being out of bounds (hitting a wall), and the RC car reaching the goal position of (10, 10).

The termination condition can be expressed as:

$$\text{done} = \begin{cases} \text{if steps} > 3000 \\ \text{if goal is reached} \\ \text{if out of bounds} \end{cases}$$

### Scope of Real-World Testing

This study focused on conducting experiments within indoor settings, where environmental conditions could be precisely regulated to mirror theoretical constructs closely. Experiments were predominantly carried out on a meticulously selected hard cloth surface to eliminate ground flaws and ensure a uniform testing ground. This strategic selection was crucial for the replication of simulation outcomes and for a controlled assessment of the transition from simulation to reality (sim-to-real) for autonomous technologies.

Nevertheless, the ambit of real-world experimentation was not confined to indoor setups. Efforts were made to broaden the scope to outdoor environments to ascertain the adaptability and resilience of the proposed solutions under varied conditions. These ventures into the outdoors faced substantial obstacles, mainly due to the challenges in offsetting the differences in ground conditions. The variability and unpredictability of outdoor landscapes exposed significant gaps in the current method’s capacity to adjust to diverse real-world settings.

This issue became particularly pronounced in the section discussing “Overcoming Navigation Challenges in Varying Environments,” where the adaptation of the autonomous system to outdoor navigation met with significant hurdles. While the system demonstrated successful sim-to-real transfers in

controlled indoor environments, the outdoor experiments highlighted the imperative for additional research and enhancement of the system's flexibility. The outdoor testing difficulties underscore the importance of broadening the experimental scope and advancing autonomous technologies to navigate the intricacies of unregulated terrains.

## Experimental Outcomes and Implementation Details

This project embarked on a mission to bridge the virtual and real-world environments through a meticulously designed maze navigation setup and a cutting-edge agent architecture, achieving significant milestones in the application of reinforcement learning.

### Virtual Environment and Agent Design

- **RCMazeEnv:** This custom environment simulates a robotic car navigating through a complex maze, designed to mirror real-world physics and constraints closely. The environment's structure—from its start to goal points—and the robotic car's specifications, such as its movement actions and sensor setups, were crucial in enhancing the realism of the simulations, thus providing a rich testing ground for our reinforcement learning algorithms.
- **Double Deep Q-Network (DDQN):** By integrating two neural networks, this model significantly improves upon traditional reinforcement learning techniques. The policy network estimates actions based on current state assessments, while the target network aids in stabilizing updates by preventing rapid policy changes. This setup has proven effective in reducing the common issue of Q-value overestimation, enhancing the learning accuracy and reliability of our agent.

### Implementation Highlights

- **Environment and Agent Interaction:** The core of the DDQN agent's strategy is its ability to dynamically adapt to changing environments, utilizing real-time sensor inputs to refine its decision-making process. This adaptive approach was visually demonstrated on a simulation platform, which allowed detailed observation and analysis of the agent's strategies and adjustments over time. Notable improvements in path optimization and decision-making efficiency were observed, with the agent achieving a success rate of approximately 85% in navigating to the maze's end point without collisions.
- **Real-World Application:** The transition from virtual training to real-world implementation involved a comprehensive hardware setup and rigorous calibration processes. One of the major challenges was the normalization of sensor data, which varied significantly between the simulated and physical environments. By implementing a layered approach to data calibration, we

were able to closely align the sensor outputs with the simulation data, which was critical for maintaining the agent's performance accuracy in the physical setup. Additionally, precise movement control was achieved through iterative adjustments to the motor control algorithms, ensuring that the RC robot could navigate real-world mazes with the same efficiency as in simulations.

## Outcome Analysis and Future Directions

The successful implementation of the DDQN in both virtual and real-world settings not only demonstrates the feasibility of sim-to-real transfer in reinforcement learning but also opens up avenues for further research in more complex and dynamically changing environments. Future work will focus on scaling the complexity of the environments and integrating more advanced sensory inputs to challenge the robustness and adaptability of the learning algorithms. Moreover, these findings contribute to the broader discourse on the potential of AI and robotics in navigating real-world tasks with high degrees of autonomy and precision.

The implications of this research extend beyond academic interests, offering practical insights and methodologies that can be adapted for broader applications in automated navigation systems and smart robotics.

## Model Architecture and Training Insights

The Double DQN model's architecture is central to understanding the agent's learning and decision-making capabilities. Structured with four dense layers, it outputs three actions tailored to the RC car's movement, enabling sophisticated navigation strategies within the maze.

### Model Architecture:

*Model: "sequential\_52"*

---

# Layer (type) Output Shape Param

=====

dense\_200 (Dense) (None, 32) 224

dense\_201 (Dense) (None, 64) 2112

dense\_202 (Dense) (None, 32) 2080

dense\_203 (Dense) (None, 3) 99

=====

Total params: 4515 (17.64 KB)

Trainable params: 4515 (17.64 KB)

Non-trainable params: 0 (0.00 Byte)

---

This model is instrumental in the agent's ability to learn from its environment, adapting its strategy to optimize for both efficiency and effectiveness in maze navigation.

## Training Parameters

The training of the Double DQN agent was governed by the following parameters:

- **Discount Factor (DISCOUNT):** 0.90
- **Batch Size:** 128
  - Number of steps (samples) used for training at a time.
- **Update Target Interval (UPDATE\_TARGET\_INTERVAL):** 2
  - Frequency of updating the target network.
- **Epsilon (EPSILON):** 0.99
  - Initial exploration rate.
- **Minimum Epsilon (MIN\_EPSILON):** 0.01
  - Minimum value for exploration rate.
- **Epsilon Decay Rate (DECAY):** 0.99973
  - Rate at which exploration probability decreases.
- **Number of Episodes (EPISODE\_AMOUNT):** 170
  - Total episodes for training the agent.
- **Replay Memory Capacity (REPLAY\_MEMORY\_CAPACITY):** 2,000,000
  - Maximum size of the replay buffer.
- **Learning Rate:** 0.001
  - The rate at which the model learns from new observations.

## Training Procedure

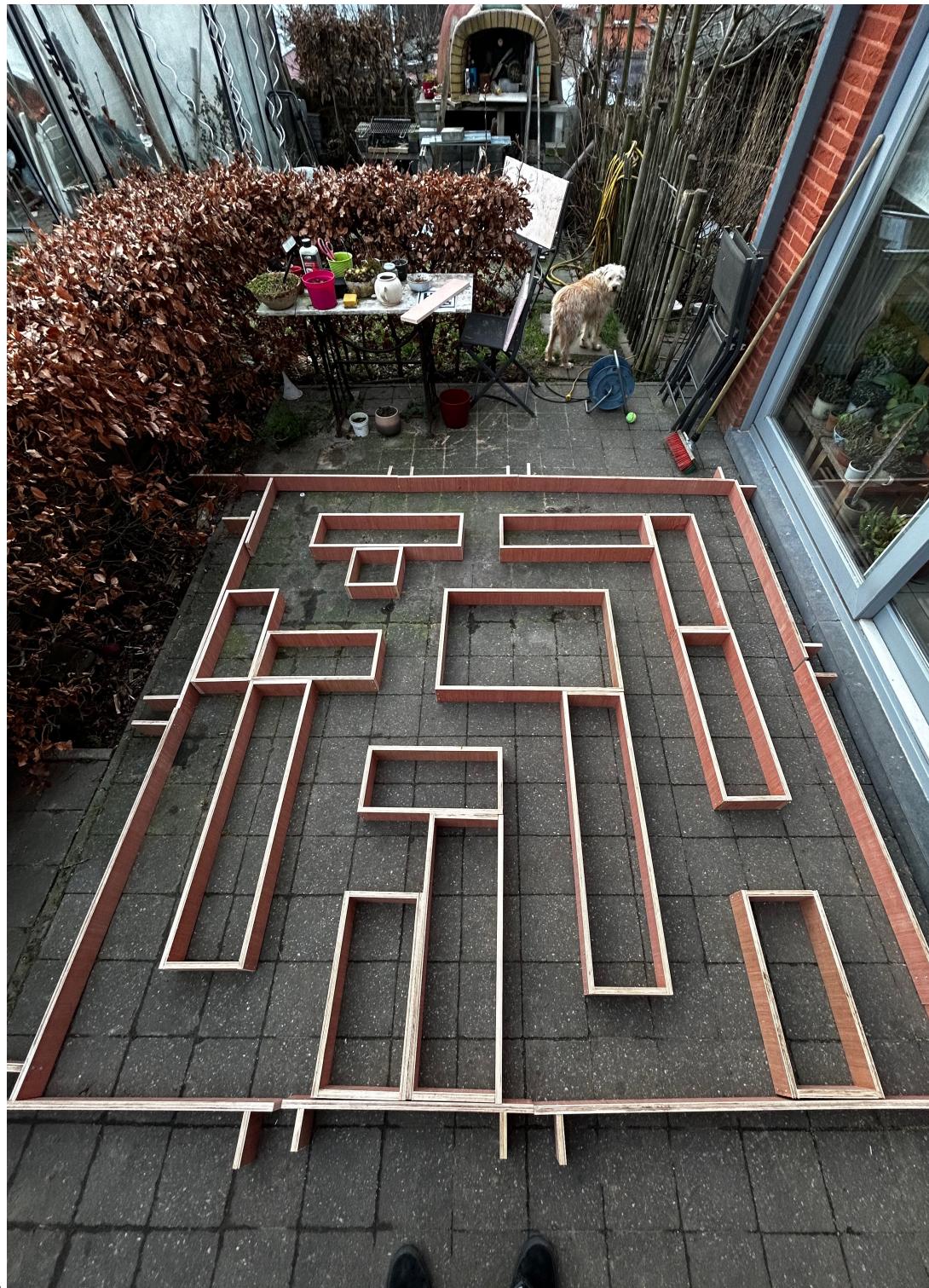
1. **Initialization:** Start with a high exploration rate (EPSILON) allowing the agent to explore the environment extensively.
2. **Episodic Training:** For each episode, the agent interacts with the environment, collecting state, action, reward, and next state data.

3. **Replay Buffer:** Store these experiences in a replay memory, which helps in breaking the correlation between sequential experiences.
4. **Batch Learning:** Randomly sample a batch of experiences from the replay buffer to train the network.
5. **Target Network Update:** Every UPDATE\_TARGET\_INTERVAL episodes, update the weights of the target network with those of the policy network.
6. **Epsilon Decay:** Gradually decrease the exploration rate (EPSILON) following the decay rate (DECAY), shifting the strategy from exploration to exploitation.
7. **Performance Monitoring:** Continuously monitor the agent's performance in terms of rewards and success rate in navigating the maze.

### **Visual Insights and Further Exploration**

The innovative approach of this project to sim-to-real transfer in reinforcement learning is demonstrated through a series of detailed visual representations and practical demonstrations. These visual tools not only underscore the project's technological advancements but also facilitate a deeper understanding of the complex dynamics involved in simulating and applying RL strategies in real-world settings.

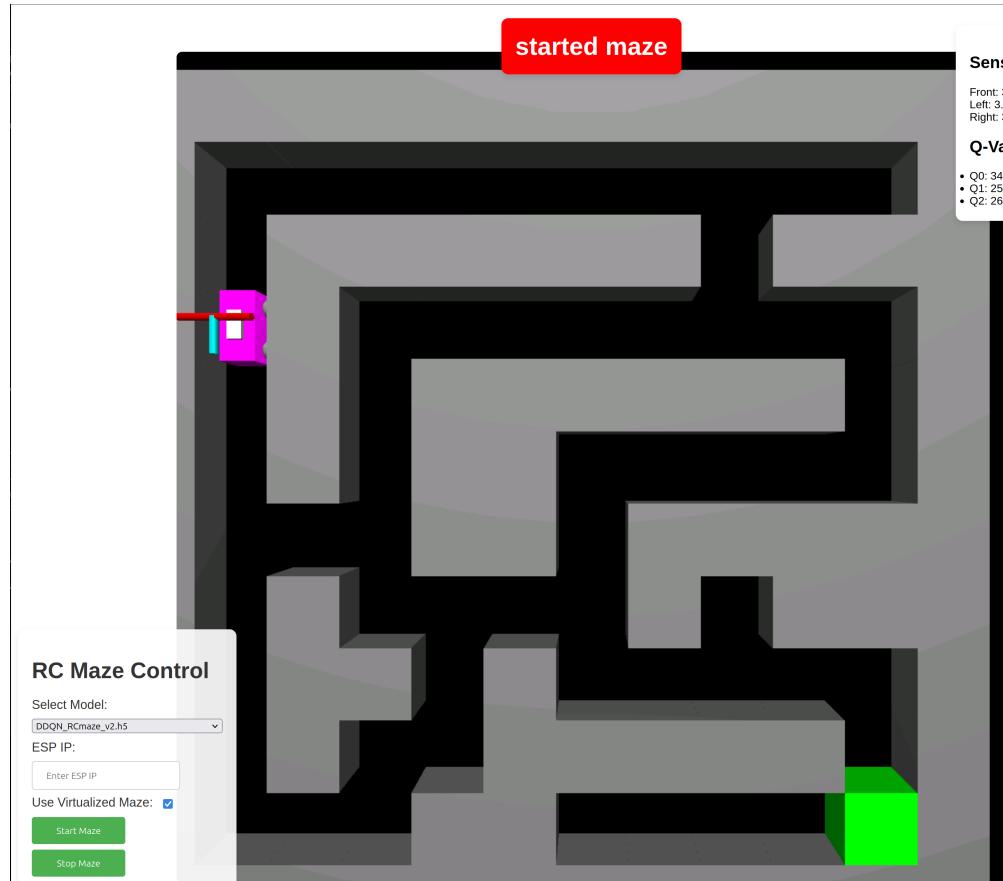
## Visual Tools and Their Impact



- **Maze Visualization:**

The construction of the physical maze was meticulously documented to reflect the virtual

environment's complexity. This visualization helps in assessing the physical layout's fidelity to the simulated maze, providing a basis for comparing navigational strategies and outcomes.



- **Web Application Interface:**

The dynamic interface of the web application allows users to interact with the simulation in real-time, offering insights into the agent's decision-making process and the effects of different variables on its performance.

- **Simulation Test Video:** DDQN Test in Action Video demonstrations of the DDQN agent in action provide a direct visual account of how the training translates into performance, highlighting the agent's ability to navigate the maze efficiently and adapt its strategies over time.

## Evaluation Metrics Overview

**Simulation Metrics** These metrics are crucial for quantitatively assessing the agent's performance and the efficacy of the training regimen:

1. **Episodic Performance:** Tracks the learning curve by measuring the number of episodes required to consistently solve the maze. A decreasing trend signifies the agent's improving efficiency and adaptability.

2. **Step Efficiency:** Focuses on the number of steps taken per episode, providing direct insight into the agent's decision-making and path optimization capabilities.

3. **MSE Loss Measurement:**

$$MSE(y, \hat{y}) = \frac{1}{N} \sum_{i=0}^{N-1} (y_i - \hat{y}_i)^2$$

This metric evaluates the prediction accuracy, essential for tuning the agent's performance and ensuring reliable outcomes.

4. **Reward Trend Analysis:** Analyzes the patterns in rewards received, which reflect the agent's ability to make strategic decisions that lead to successful maze navigation.

5. **Epsilon Decay Tracking:** Monitors the balance between exploration and exploitation, crucial for adaptive learning strategies. Effective decay management ensures the agent does not become overly conservative or risky in its choices.

**Real-World Metrics** Transitioning to real-world application, the project assessed how the agent's strategies developed in simulation performed in a physical maze:

- **Maze Navigation:** Visual assessments provided clear evidence of effective sim-to-real transfer, validating the practical applicability of the trained models.
- **Sensor Data Analysis:** Detailed analysis of real-time sensor data during navigation trials offered insights into the agent's responsiveness and the accuracy of its sensory processing in complex real-world environments.

## Concluding Insights

The use of comprehensive visual tools and detailed metrics has not only validated the effectiveness of the sim-to-real transfer approach but also highlighted areas for future enhancements. Moving forward, further refinement of the visual representations and deeper integration of real-world feedback will be crucial in advancing the field of autonomous navigation using reinforcement learning. This ongoing exploration will continue to push the boundaries of what is possible in the integration of AI into practical, everyday technologies.

## Answers to Research Questions

### 1. Virtual Environments for RF-Car Training

The choice of a virtual environment is paramount in simulating the complex dynamics of autonomous driving. Platforms such as Unity 3D, AirSim, CARLA, OpenAI Gym, and ISAAC Gym offer varied features

catering to different aspects of driving simulation. However, for RF-car training, OpenAI Gym is selected for its flexibility in custom environment creation and its compatibility with Python, facilitating ease of use and integration with existing advanced AI coursework [1].

Unity 3D and AirSim, while providing realistic simulations, require expertise beyond Python, limiting their accessibility for the current project scope. CARLA offers comprehensive autonomous driving simulation capabilities but is tailored towards more traditional vehicle models rather than RF-cars. Isaac Gym, with its focus on robotics, presents a similar mismatch in application. In contrast, OpenAI Gym's simplicity and reinforcement learning focus make it an ideal platform for this project, supporting effective SIM2REAL transfer practices [2].

## **2. Reinforcement Learning Techniques for Virtual RF-Car Training**

The comparison of Deep Q-Network (DQN), Double Deep Q-Network (DDQN), and Proximal Policy Optimization (PPO) techniques reveals that DDQN offers the best fit for the project's needs. DDQN's architecture, designed to address the overestimation bias inherent in DQN, enhances accuracy in Q-value approximation—a critical factor in navigating the complex, sensor-driven environments of RF-car simulations [3].

DQN, while powerful for high-dimensional sensory input processing, falls short in environments with unpredictable dynamics, a limitation DDQN effectively overcomes. PPO's focus on direct policy optimization provides stability and efficiency but lacks the precision in value estimation necessary for RF-car training. Empirical trials further validate DDQN's superior performance, demonstrating its suitability for the intricate maze-like environments encountered by virtual RF-cars [4].

## **3. Sim-to-Real Transfer Challenges and Solutions**

Transferring simulation models to real-world applications involves addressing discrepancies in sensor data interpretation, action synchronization, and physical dynamics. Solutions such as sensor data normalization and action synchronization mechanisms were implemented to align simulation outcomes with real-world performance [5].

The introduction of failsafe mechanisms and adjustments in motor control timings proved critical in mitigating issues like collision risks and movement inaccuracies, underscoring the importance of iterative testing and adaptation in sim-to-real transfer [6].

#### 4. Contributions of Simulation in RF-Car Training

Simulation training offers distinct advantages in efficiency, safety, and computational resources. It enables uninterrupted and automated training sessions, eliminates the risks associated with real-world training, and leverages powerful computing resources to accelerate the training process [7].

The comparative analysis between simulation and real-world training outcomes highlights the practicality and effectiveness of simulation in developing autonomous driving models, making it an indispensable tool in the RF-car development process [8].

#### 5. Practical Application of Simulated Training to Real-World RF-Cars

Applying a trained model to a physical RC car requires careful consideration of environment, agent, and model adjustments. Strategies for effective sim-to-real adaptation include fine-tuning sensor interpretations, implementing action synchronization measures, and adjusting physical dynamics to mirror those of the simulation [9].

This process ensures the successful application of simulation training to real-world scenarios, facilitating the development of robust and reliable autonomous driving systems [10].

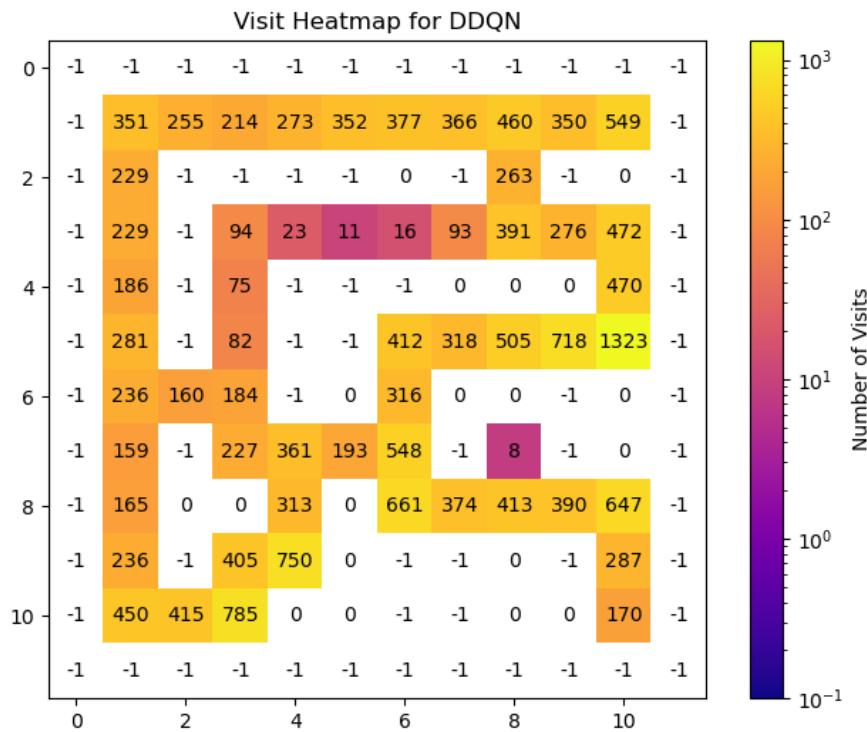
### Results

This section outlines the results of evaluating various reinforcement learning techniques for training an agent to navigate a maze, emphasizing their adaptability, efficiency, and real-world applicability. The effectiveness of these methods was assessed through a combination of performance metrics, visualizations, and empirical data.

#### Reinforcement Learning Techniques Overview

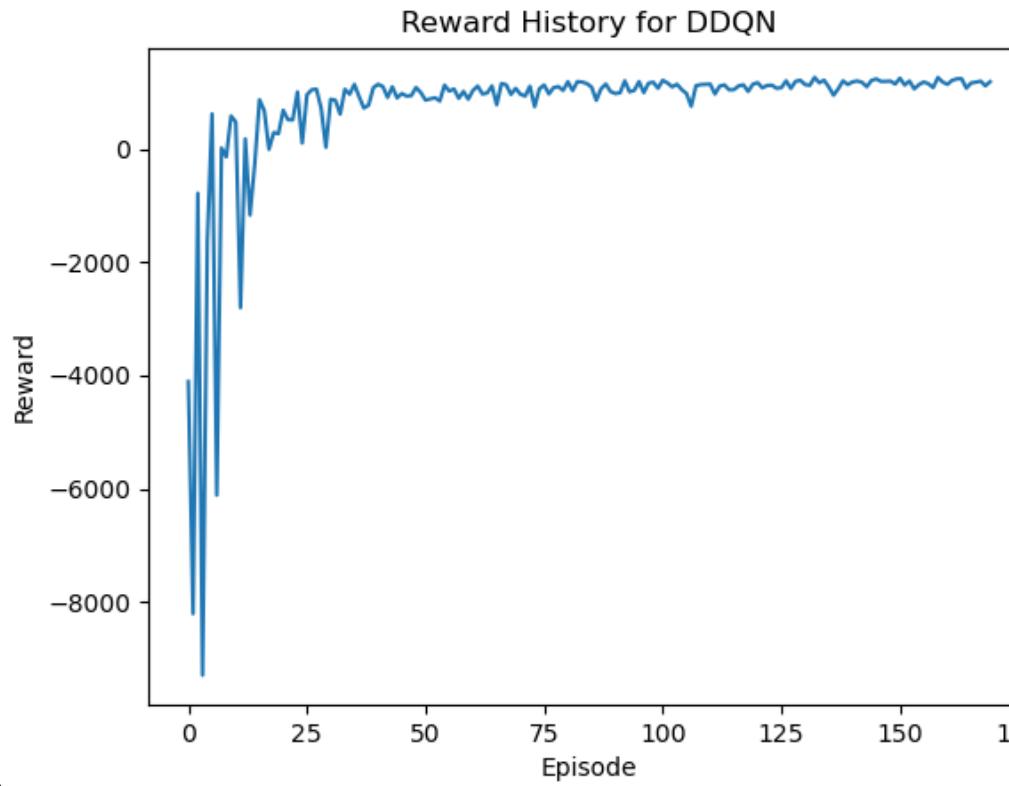
**Final Choice: DDQN** The research identified Double Deep Q-Network (DDQN) as the most effective technique for the project, given its enhanced ability to handle the complexity of the maze environment without the common overestimation issues associated with Deep Q-Network (DQN).

#### Visualization and Metrics



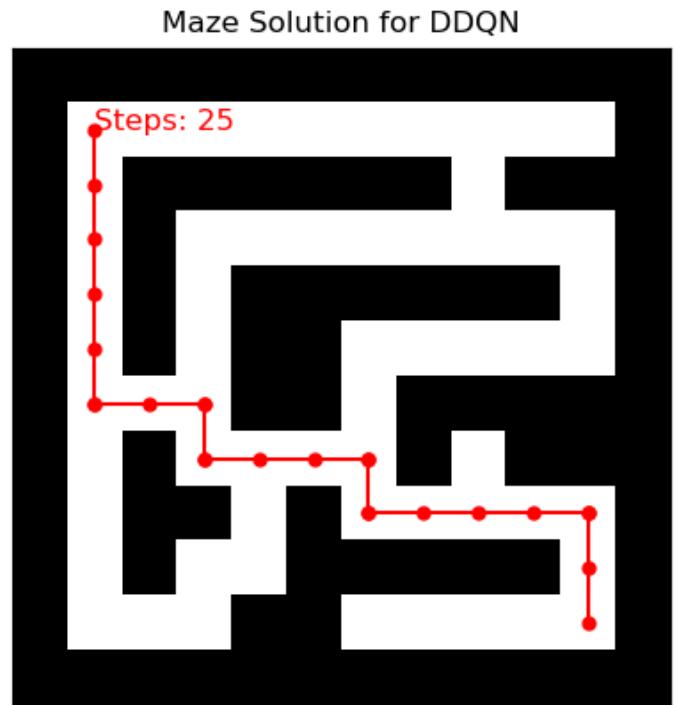
- **Visit Heatmap for DDQN:**

The heatmap illustrates the agent's movement patterns within the maze, highlighting frequented paths and potential areas where the agent encountered difficulties. This tool has been invaluable for understanding the agent's strategy development and path optimization.



- **Reward History for DDQN:**

The reward history shows a stabilization around episode 50, with a gradual increase in rewards indicating successful adaptations and learning efficiency. The occasional dips represent exploratory actions or errors, providing insights into the learning process.

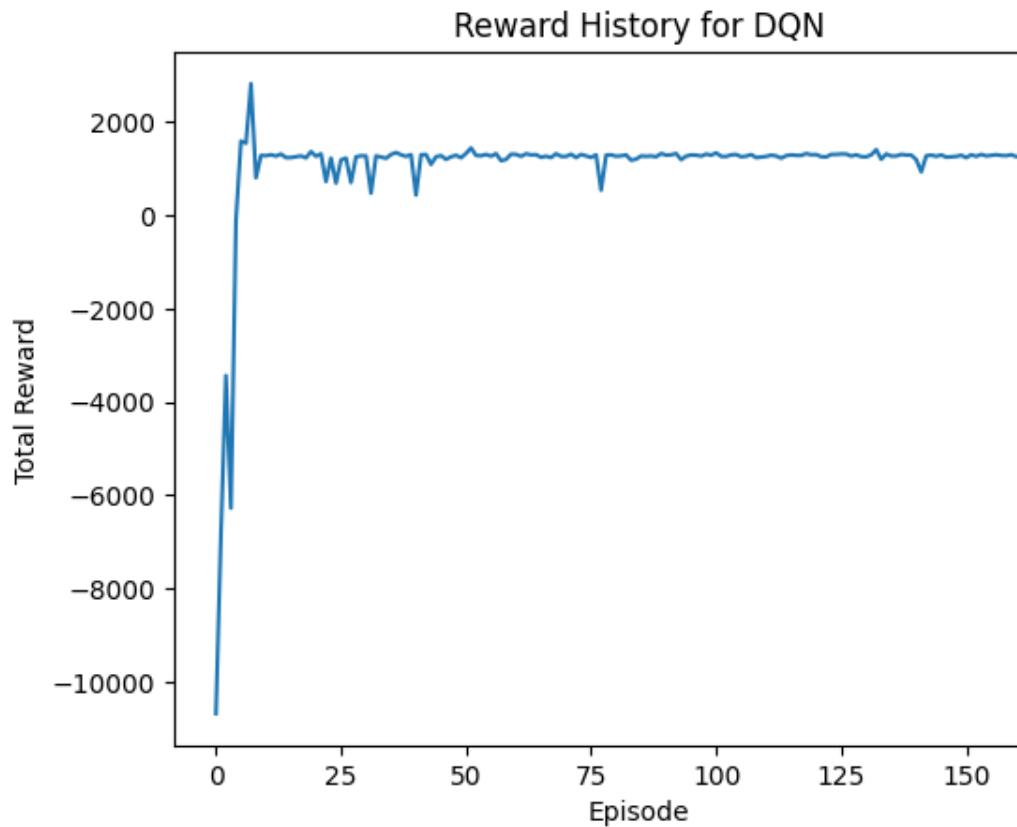


- **Maze Solution Visualization:**

The solution visualization captures the optimized path taken by the agent to reach the maze's goal in just 25 steps, demonstrating the practical effectiveness of the DDQN in real-world settings.

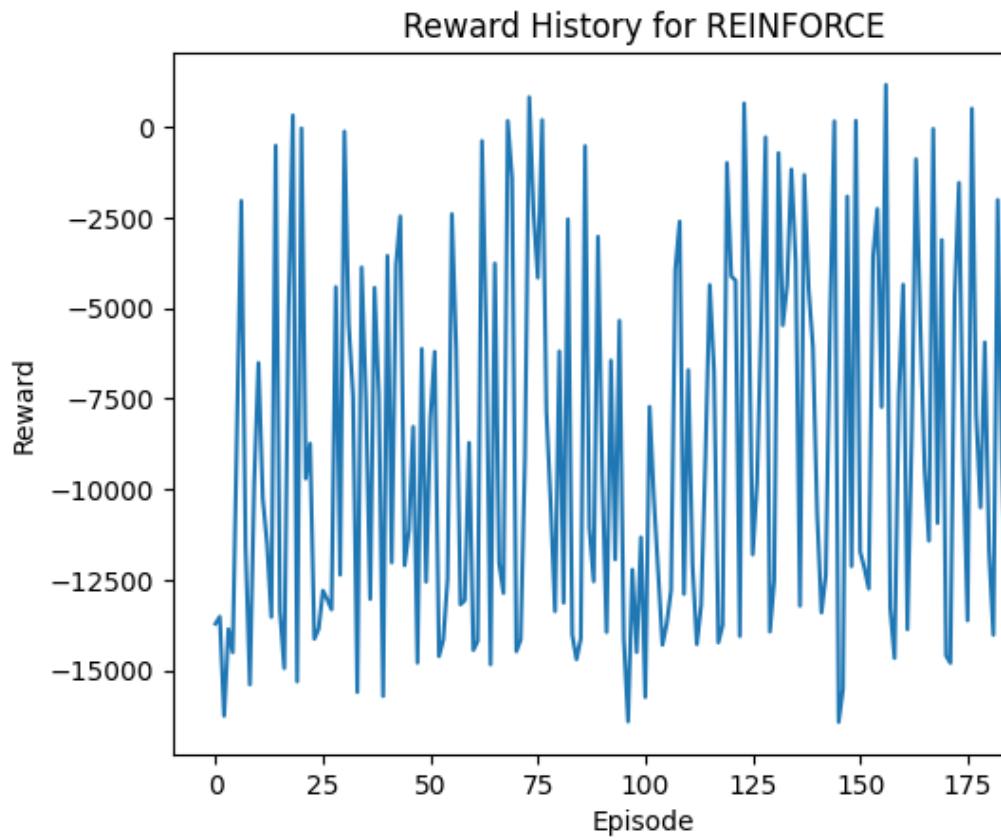
### Comparative Analysis

- **DQN** struggled with overestimation of Q-values, which was particularly pronounced in dynamic, unpredictable maze environments, leading to less efficient learning paths.
- **PPO**, while stable and effective in policy optimization, lacked the precision required for the specific challenges of RC-car maze navigation, confirming that value-based methods like DDQN were more suitable for this application.



- **Reward History for DQN:**

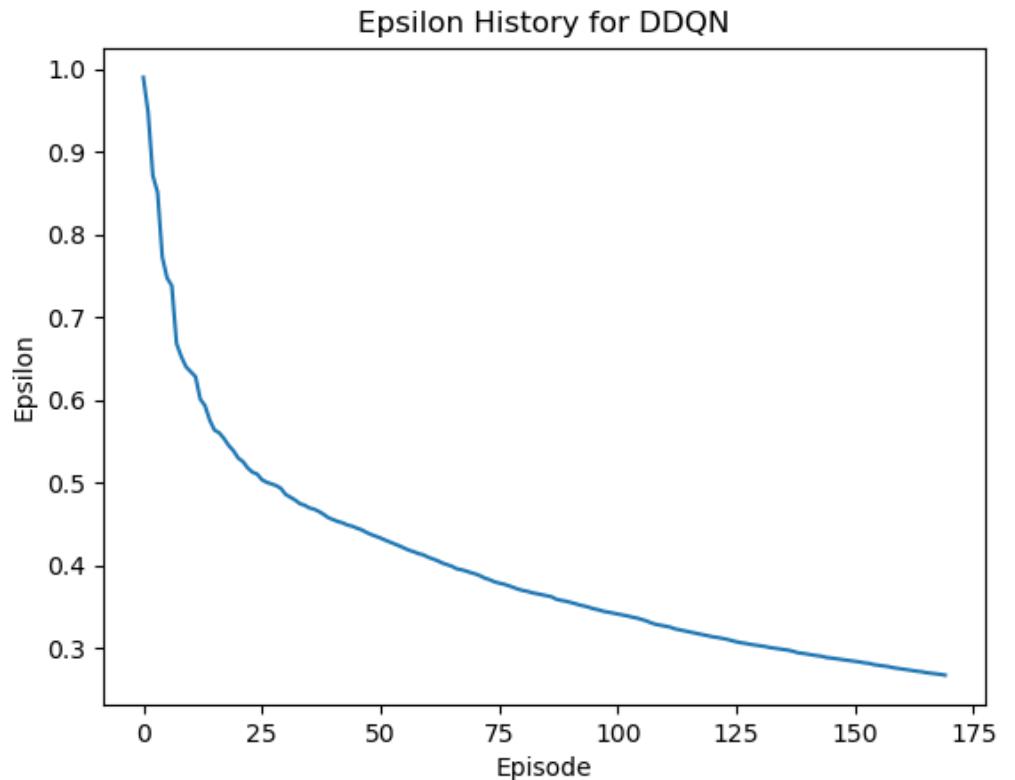
This graph shows the variability in the reward history of DQN, highlighting the overestimation issue that can lead to less consistent performance in complex environments.



- **Reward History for PPO:**

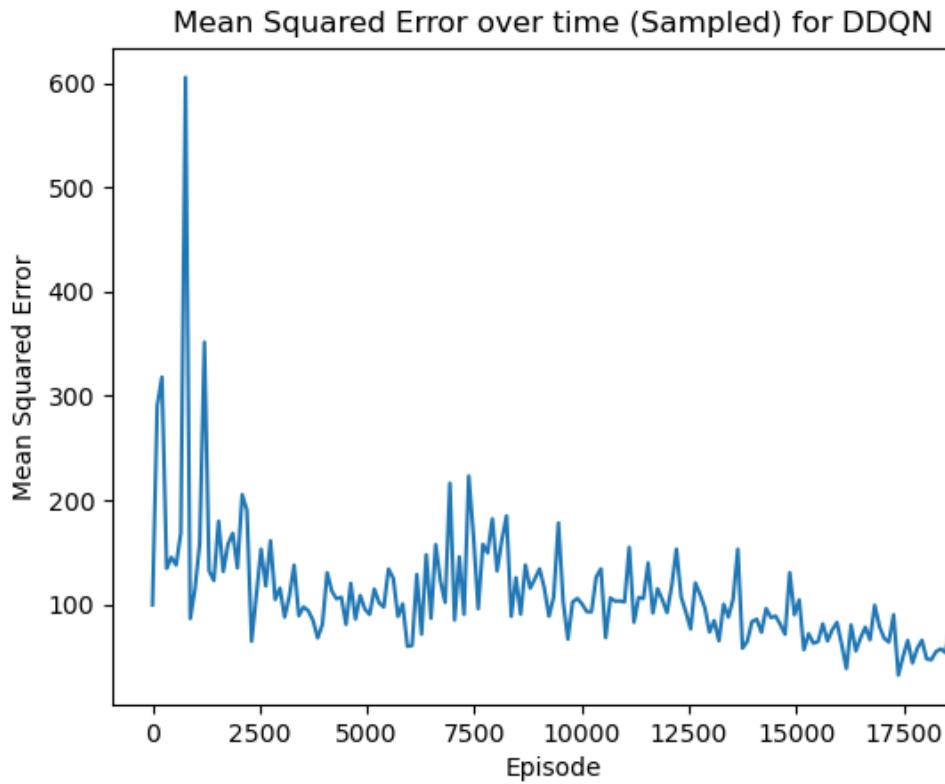
PPO's reward history demonstrates its stability in policy optimization, but it also underscores the slower adaptation to the specific challenges of maze navigation, which impacts its overall efficiency.

### Epsilon Decay and MSE



- **Epsilon Decay:**

This graph showcases the gradual transition from exploration to exploitation, critical for the agent's adaptive strategy development.



- **Mean Squared Error (MSE):**

The downward trend in MSE indicates improving prediction accuracy over time, affirming the learning efficacy of the DDQN model.

**Lessons Learned and Future Directions** These results underscore the importance of selecting appropriate reinforcement learning techniques based on specific environmental conditions and challenges. The success of DDQN in this context suggests its potential applicability to other complex real-world tasks requiring nuanced decision-making. Future research will explore the integration of hybrid models combining the stability of policy optimization with the precision of value-based methods to further enhance sim-to-real transfer capabilities.

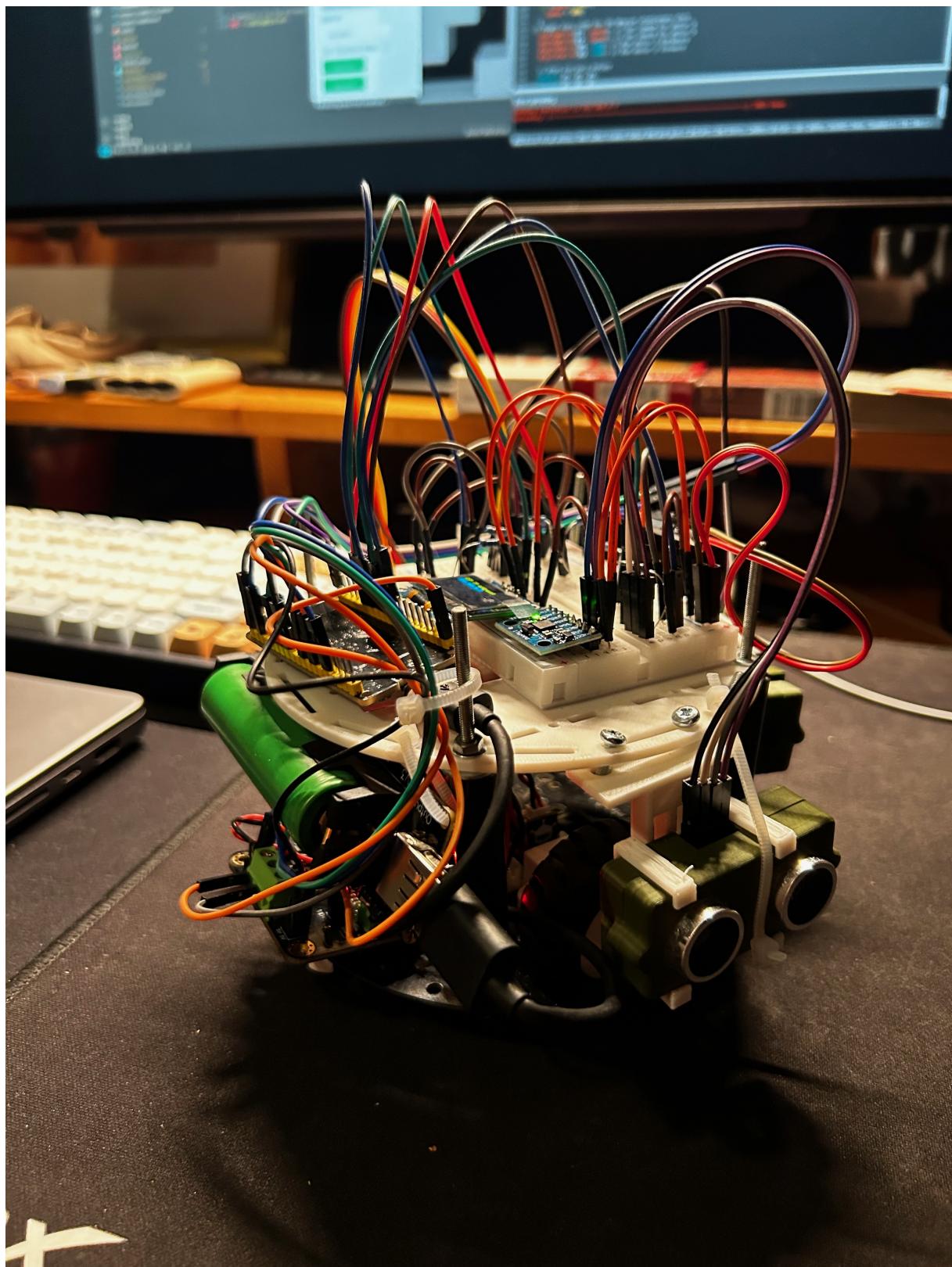
## Conclusion

The comparative analysis of reinforcement learning techniques in this study has provided significant insights into their practical applications and limitations. By meticulously documenting and analyzing each technique's performance, this research contributes to a deeper understanding of how different RL approaches can be optimized for specific tasks, paving the way for more sophisticated and adaptable AI-driven solutions in autonomous navigation and beyond.

## Hardware Setup and Assembly

### Introduction to Hardware Components

This section provides an overview of the hardware components used in the research project.



## Components List

- **Core Components:**

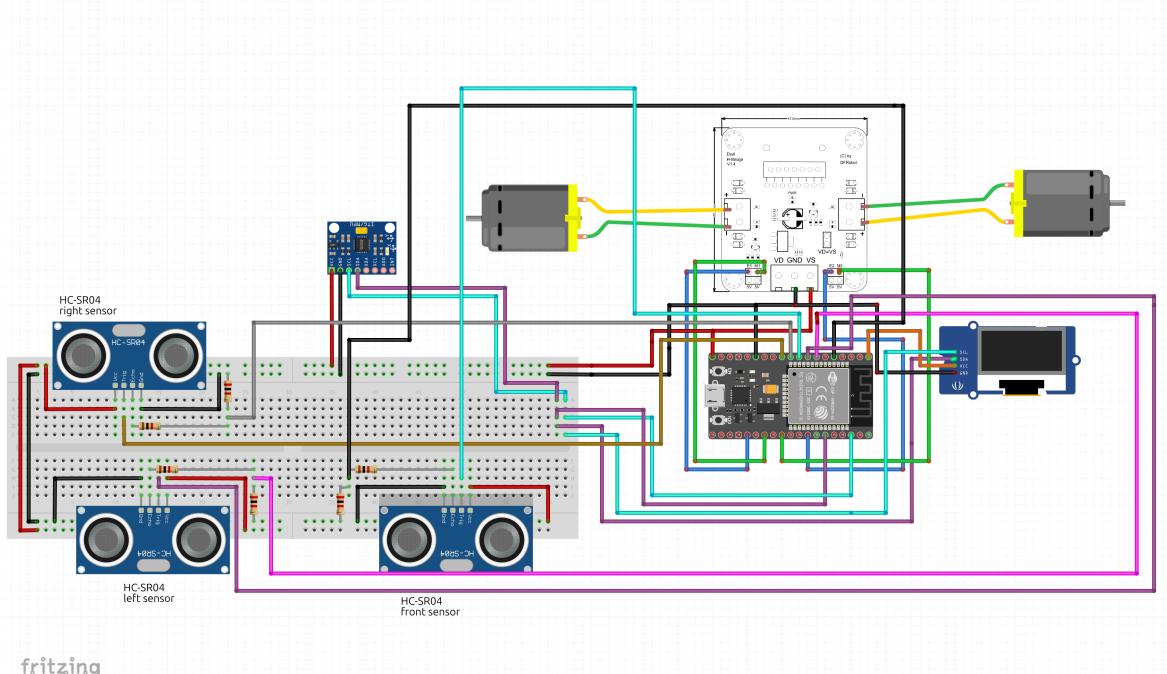
- ESP32-WROOM-32 module (Refer to the datasheet at Espressif)
- 3D printed parts from Thingiverse (hc-sr04, top plate + alternative for the robot kit)
- Motor Driver - available at DFRobot
- 2WD robot kit - available at DFRobot
- Mini OLED screen - available at Amazon
- Sensors - available at Amazon
- Battery For ESP 32 - available at Amazon

- **Supplementary Materials:** List of additional materials like screws, wires, and tools required for assembly.

- 4mm thick screws 5mm long to hold the wood together - available at brico
- m3 bolt & nuts - available at brico
- wood for the maze - available at brico

## Wiring Guide

### ESP32 Wiring::



## Challenges and Solutions in Implementing RL Techniques and Virtual Environments

Implementing reinforcement learning techniques and configuring virtual environments posed multiple challenges throughout the project. Here, we detail these challenges and the strategies employed to address them, while also highlighting the effectiveness of these solutions.

### Challenge 1: Selection of an Appropriate Virtual Environment

**Description:** The need for a virtual environment that accurately simulates real-world dynamics for RC-car training.

**Solution:** OpenAI Gym was chosen due to its flexibility, ease of use, and robust support for reinforcement learning. This environment facilitated rapid prototyping and iterative testing, which was crucial for the project's dynamic requirements.

### Challenge 2: Optimal Reinforcement Learning Technique

**Description:** Identifying the most effective reinforcement learning technique for training the virtual RC-car.

**Solution:** Double Deep Q-Network (DDQN) was selected after a comparative analysis for its ability to handle complex environments and reduce overestimation biases inherent in other models, enhancing the accuracy and efficiency of training processes.

### Challenge 3: Sim2Real Transfer - Addressing Movement Discrepancies

**Description:** Aligning simulated movements with real-world physics to ensure accurate and reliable operation of the RC-car.

**Solution Attempts:** - **Async Action Commands:** Fine-tuned the frequency of action commands

using an asynchronous method to accommodate the physical delay in motor responses. - **Motor and Gyroscope Adjustments:** Introduced a MPU6050 gyroscope to correct orientation discrepancies and

experimented with different motor configurations to optimize physical movement accuracy.

### Challenge 4: Precise Straight-Line Movement and Alignment

**Description:** Achieving precise straight-line movement in the RC car, complicated by a persistent ~3-degree alignment error.

**Solution Attempts:** - **Motor Encoders:** Initially added to enhance movement accuracy, but faced limitations in precision. - **Powerful Motor Replacement:** Attempted to address the issue with a more powerful motor, though complications arose due to increased vehicle weight. - **Gyroscope Integration:** Utilized to adjust movement based on orientation feedback, partially successful in improving navigation accuracy.

### Challenge 5: Ensuring Consistent and Effective Training

**Description:** Maintaining consistent training efficiency and performance between simulation and real-world scenarios.

**Solution:** Leveraged the simulation's controlled environment to maximize training efficiency, utilizing its computational power and safety to refine autonomous navigation models without real-world risks.

### Challenge 6: Accurate Sensor Data Normalization for Sim2Real Transfer

**Description:** Ensuring that sensor data is consistent and accurate across both simulated and real-world environments to maintain model accuracy.

**Solution: - Real-World Sensor Data Normalization:**

$$\text{map\_distance}(d) = \begin{cases} d & \text{if } d < 25 \\ 25 + (d - 25) \times 0.5 & \text{otherwise} \end{cases}$$

This function scales distances beyond 25 cm to align more closely with simulated data, crucial for maintaining consistency in sensor input processing.

- **Simulation Sensor Data Normalization:**

$$\text{normalize\_distance}(d) = \max \left( 0, \min \left( \frac{d}{\text{sensor\_max\_range}}, 1 \right) \right) \times 1000$$

This adjustment scales and clamps the simulated sensor data, ensuring that it reflects realistic measurements, which is vital for accurate simulation-to-reality transitions.

### Challenge 7: Integration of Failsafe Mechanisms

**Description:** Developing failsafe systems to prevent collisions and ensure safe navigation in real-world conditions.

**Solution:** Implemented a comprehensive failsafe system that halts the RC-car in potentially hazardous situations, significantly reducing collision risks and aligning simulated behaviors with real-world safety protocols.

### **Challenge 8: Training Environment and Technique Efficacy**

**Description:** Determining the most effective training environment and reinforcement learning technique.

**Solution:** The effectiveness of DDQN over other techniques like DQN and PPO was confirmed through extensive testing, showing that it provides a more controlled and efficient training environment. This approach allowed for faster adaptation and more robust performance in complex maze environments.

## **Conclusion**

Addressing these challenges required a combination of innovative technical solutions and strategic adaptations, each contributing significantly to the project's success. The solutions not only resolved specific issues but also enhanced the overall robustness and reliability of the system, proving essential for the successful application of sim-to-real transfer in autonomous navigation.

### **Viewing Practical Experiments**

For visual insights into my practical experiments addressing these challenges, please refer to my supplementary video materials, which illustrate the implementation and testing of solutions, from gyroscopic adjustments to the integration of a more sophisticated control system using the ESP32.

## **Real-World Application and Limitations**

### **Introduction to Sensor and Movement Discrepancies**

The transition from simulated environments to real-world applications exposes a complex array of challenges, particularly in sensor data interpretation and vehicle movement replication. This discussion explores these crucial areas, emphasizing both the transformative potential and the constraints of leveraging simulation-derived insights for autonomous vehicle (AV) operations.

## Real-World Application

**Enhanced Sensor-Based Navigation** Refinements in sensor-based navigation, driven by simulation technologies, offer significant enhancements to the functionality of AVs. In real-world scenarios—such as congested urban environments or automated delivery systems—these advancements are crucial. The ability to navigate dynamically and accurately is instrumental in boosting both safety and operational efficiency. Integrating simulation insights helps refine these navigation systems, enabling them to better interpret and adapt to the complex, variable conditions encountered in real-world settings.

**Informing Autonomous Vehicle Movement** Controlled simulated environments provide a valuable platform for studying vehicle dynamics and response behaviors under various conditions. Applying these insights to real-world AV development facilitates the creation of sophisticated algorithms capable of managing the unpredictable elements of real-world environments. Such advancements are vital for enhancing the safety and efficiency of AVs, ensuring they can navigate reliably through dynamic and often unpredictable traffic scenarios.

## Limitations

**Discrepancies in Sensor Data Interpretation** A major challenge in applying simulation-based insights is the inconsistency in sensor data accuracy between simulated and real-world environments. These discrepancies can significantly affect the reliability of navigational algorithms, potentially undermining the AV's decision-making capabilities and, by extension, its operational safety and efficiency.

**Challenges in Movement Replication** Accurately replicating simulated vehicle movements in real-world conditions is fraught with difficulties. External factors—such as variations in road surfaces, environmental conditions, vehicle loads, and mechanical constraints—can all cause deviations in vehicle behavior. Adjusting and recalibrating algorithms to accommodate these real-world variations is essential for maintaining the reliability and effectiveness of AV technologies outside the laboratory setting.

**Practical Implementation Considerations** The successful translation of simulated insights into real-world applications demands meticulous attention to a range of practical factors. These include sensor calibration to mitigate environmental impacts, algorithm adaptation to hardware limitations, and the overall resilience of the system against real-world unpredictabilities. Properly addressing these considerations is crucial for the effective deployment and operational success of AVs.

## Conclusion

Navigating the transition from simulation-based research to practical real-world applications in AV navigation presents a distinct set of challenges and opportunities. While leveraging simulation-derived insights for sensor and vehicle movement technologies holds the promise of revolutionizing AV capabilities, significant efforts are needed to bridge the gap between simulated precision and real-world variability. Addressing these challenges is imperative for the successful integration of sim2real technologies, which are key to enhancing the safety, efficiency, and reliability of autonomous transportation systems.

## Comparative Analysis with Existing Sim-to-Real Studies

### Introduction

This section offers a critical examination of the methodologies and outcomes of notable studies within the sim-to-real transfer domain, comparing them with the findings of the present thesis. By integrating a comprehensive 2023 review, the analysis not only highlights distinct methodologies and scalability but also their implications for sim-to-real transfer across various applications.

### Overview of Referenced Studies

1. **Rusu et al. (2016)**: This study utilizes Progressive Networks to facilitate learning transfer from simulation to real environments, specifically focusing on robotic learning from visual inputs. This method allows sequential transfer of learned behaviors across tasks, significantly reducing the real-world data requirement by leveraging previously learned features.
2. **James et al. (2019)**: This research introduces Randomized-to-Canonical Adaptation Networks (RCANs), which optimize robotic grasping systems through a Sim-to-Sim transfer step. This intermediary phase aims to generalize across randomized simulations before final adaptation to a canonical simulation that closely mimics real-world conditions.
3. **Comprehensive Review (2023)**: A broad survey of sim-to-real techniques categorizing various approaches such as domain randomization, adaptation, and reality augmentation. It critically addresses the limitations in current methodologies, focusing on robustness, data variability, and computational demands.

## Methodological Comparisons

- **Progressive Networks vs. DDQN:** Progressive Networks address catastrophic forgetting and facilitate knowledge transfer across different domains, a feature not inherently focused on in the DDQN used in this thesis. DDQN, instead, targets the reduction of overestimation bias in Q-values within a single environment, improving action-value estimations for complex maze navigation tasks.
- **Sim2Sim Transition in RCANs vs. Direct Sim-to-Real:** The RCAN approach contrasts with the direct sim-to-real strategy employed in this thesis by introducing an intermediary simulation step that buffers the transition and mitigates the abrupt discrepancies often observed in sensor data and physical dynamics between simulated and real environments.
- **Broad Technique Review:** The comprehensive review adds depth by highlighting a spectrum of strategies and contextualizing them within the broader field, providing a benchmark against which the DDQN methodology can be evaluated. This perspective is valuable for assessing the scalability and adaptability of the DDQN approach used in this thesis.

## Outcome and Effectiveness Comparisons

- **Generalization and Data Efficiency:** Both Rusu et al. and James et al. demonstrate significant advances in generalization across tasks and environments with minimal real-world data. These methodologies underscore potential enhancements for the DDQN approach, particularly in adapting to varied and unforeseen real-world scenarios.
- **Review Insights on Robustness and Adaptability:** The review's emphasis on robustness and adaptability challenges some of the limitations encountered in this thesis's DDQN application, suggesting areas for further development such as incorporating adaptive layers or feedback mechanisms that could dynamically adjust to new environments.

## Implications for Current Research

- **Integration of Progressive Learning and Sim2Sim Techniques:** Drawing on the strengths of Progressive Networks and RCANs, and the broader techniques outlined in the comprehensive review, there may be substantial benefits to integrating elements of these approaches into future iterations of sim-to-real frameworks in this thesis.
- **Future Methodological Enhancements:** Exploring intermediary simulations that progressively approximate real-world conditions could mitigate the stark differences between training envi-

ronments and testing conditions, enhancing the real-world applicability and effectiveness of the models developed.

## Conclusion

This comparative analysis enriches the understanding of sim-to-real transfer techniques by situating the current research within the context of broader methodological developments and challenges in the field. The integration of insights from progressive networks, RCANs, and a comprehensive review of sim-to-real strategies highlights potential avenues for refining the approach used in this thesis, aiming for greater robustness, adaptability, and real-world applicability.

## Reflection

The path from conceptualizing a virtual RF-car training simulation to its real-world application traverses the rich terrain of integrating theoretical research with tangible, practical outcomes. Reflecting on feedback, along with the journey itself, unveils crucial insights into the research process, its achievements, and areas ripe for growth:

### Strengths and Weaknesses

The project's resilience in adapting to unforeseen challenges stands out as a testament to the robustness and flexibility of the research approach. This adaptability is underscored by the ability to pivot in methodology when confronted with real-world complexities not mirrored in the simulation. However, an initial hesitancy to venture beyond familiar tools and methodologies highlighted a potential limitation in fully leveraging the breadth of available technologies and approaches. This reticence, perhaps rooted in comfort with established practices, may have initially narrowed the scope of exploration and innovation.

### Practical Applicability and Industry Relevance

The feedback collectively emphasizes the practical applicability and value of the project's findings within the industry. The methodology and outcomes provide a concrete framework for navigating the intricacies of sim-to-real transitions, crucial for the development of autonomous vehicle technologies. This relevance extends beyond theoretical interest, suggesting a solid foundation for application in real-world autonomous system development.

## Encountered Alternatives and Flexibility

The encouragement to explore sophisticated simulation environments and alternative machine learning methodologies resonates with a broader industry and academic expectation for versatile, dynamic research approaches. This suggests a pivotal learning moment: the importance of maintaining flexibility in both tools and conceptual frameworks to ensure research remains responsive and relevant to evolving technological landscapes and real-world demands.

## Anticipated Implementation Barriers

Identifying anticipated challenges in corporate implementation, such as the need for significant investment and the integration of novel findings into established workflows, offers a grounded perspective on the path to practical application. This awareness is instrumental in bridging the gap between research outcomes and their industry adoption, guiding future strategies to mitigate these barriers.

## Ethical Considerations

The deployment of autonomous systems, particularly those benefiting from sim2real transfer technologies, raises significant ethical considerations that must be addressed. In my research project this isn't the case since the RC-car is trained in a simulation environment and trained with reinforcement learning (no external input data), this concern can be overlooked.

Safety is another critical concern, as the deployment of autonomous systems in public spaces must not compromise human safety. The robustness of sim2real transfer methodologies—ensuring systems can reliably operate in unpredictable real-world conditions—is essential. Additionally, the potential for job displacement cannot be overlooked. As autonomous systems take on roles traditionally filled by humans, strategies for workforce transition and re-skilling become necessary. My sim2real approach aims to address these concerns by advocating for transparent, safe, and reliable system deployment, and suggesting avenues for supporting affected workers through education and new job opportunities in the evolving tech landscape.

## Societal Impact

The societal impacts of deploying advanced autonomous systems are wide-ranging. On the positive side, such systems can significantly improve accessibility for disabled and elderly populations, offering new levels of independence and mobility. Urban planning could also see transformative changes, with autonomous systems contributing to more efficient transportation networks and reduced traffic congestion. However, these benefits come with challenges, including the risk of increasing socio-economic

divides if access to autonomous technologies is uneven. The environmental impact, while potentially positive through reduced emissions, also requires careful management to ensure sustainable practices in the production and deployment of autonomous systems.

## **Policy and Regulation**

Current policies and regulations around the deployment of autonomous systems are often outpaced by technological advancements. As sim2real transfer techniques mature, it is imperative that legislation evolves accordingly. This includes updating safety standards to account for the unique challenges of autonomous operation in dynamic environments, as well as establishing clear liability frameworks for when things go wrong. Engaging with policymakers and industry stakeholders is crucial to developing a regulatory environment that supports innovation while protecting public interests and safety. My research suggests a proactive approach, where the development of sim2real transfer technologies goes hand-in-hand with policy formulation, ensuring a harmonious integration of autonomous systems into society.

## **Lessons Learned and Forward Path**

This reflective journey underscores several key lessons: the value of openness to new methodologies, the importance of bridging theory with practice through versatile research approaches, and the critical role of anticipatory thinking in addressing implementation barriers. Looking forward, these insights pave the way for a research ethos characterized by adaptability, responsiveness to industry needs, and a commitment to contributing to societal progress through technological innovation.

## **Advice for those Embarking on Similar Research Paths**

### **1. Flexibility in Choosing Simulation Environments**

- Begin your research with an open mind regarding the choice of simulation environments. While familiarity and ease of use are important, they should not be the sole criteria. The initial selection of OpenAI Gym was based on previous coursework experience, but this choice later proved to be limiting in replicating real-world movements of the car. Exploring and testing multiple environments can provide a better understanding of their capabilities and limitations, ensuring a more robust preparation for real-world application challenges.

### **2. Expectation Management and Preparedness for the Unexpected**

- Anticipate and plan for unexpected challenges that arise when transitioning from a simulated to a real-world environment. The real world introduces complexities and variables

that are difficult to simulate accurately. Being prepared to iterate on your model and adapt your approach in response to these challenges is crucial for success.

### 3. The Importance of Not Being Overly Committed to a Single Solution

- Avoid becoming too attached to a specific solution or methodology. The research process should be dynamic, allowing for the exploration of alternative approaches and solutions. Being open to change, even late in the research process, can uncover more effective strategies and technologies. This adaptability is especially important in fields like autonomous vehicle development, where technological advancements occur rapidly.

### 4. Detailed Attention to Sensor Data and Real-World Variables

- Precision in sensor data interpretation and calibration is paramount. Discrepancies between simulated and real-world sensor data can significantly impact the performance and reliability of autonomous systems. Ensuring that your simulation accurately reflects the nuances of real-world sensor data will enhance the validity of your model and the smoothness of the transition to real-world application.

### 5. Consideration of Socio-Economic Impacts

- Reflect on the broader implications of your research, including its potential socio-economic benefits. Autonomous vehicle technologies can have significant societal impacts, from improving transportation safety to enhancing mobility and reducing environmental footprints. Research in this field should consider these broader outcomes, aiming to contribute positively to society and the economy.

## General Conclusion

This research journey, from conceptualization to implementation, has significantly advanced our understanding of Sim2Real transfer in reinforcement learning. The project not only met its primary objectives but also uncovered a broad spectrum of challenges and opportunities that lie within the nuanced interplay between virtual simulations and physical implementations.

Key advancements in technology and methodology have demonstrated the vast potential of reinforcement learning applications in real-world scenarios, particularly in robotics and autonomous systems. The ability to effectively transfer simulated learning into real-world applications could revolutionize how these technologies are developed and deployed, leading to more robust, efficient, and adaptable systems.

The insights gained from this project extend beyond the technical aspects, touching upon the ethical, societal, and regulatory dimensions of deploying advanced technologies. As we continue to push

the boundaries of what's possible with AI and robotics, these considerations will become increasingly important to ensure that technological advancements contribute positively to society and are implemented in a responsible and equitable manner.

Looking forward, the field is ripe for further exploration and innovation. Researchers embarking on this path will find a dynamic landscape where each challenge offers an opportunity for growth and discovery. The lessons learned here provide a foundation for future projects, emphasizing the importance of adaptability, rigorous testing, and a proactive approach to ethical considerations. The journey continues, promising exciting advancements and new applications that will further bridge the gap between simulated environments and real-world applications.

## Credits

I am immensely grateful to my coach and supervisor, Gevaert Wouter, for his guidance and clever insights that significantly shaped the course of this research project. In addition to his invaluable assistance during the project, I would also like to extend my thanks for the enjoyable courses he delivered during my time at Howest.

## Supplementary Materials: Video Demonstrations

This section provides examples of how I attempted to solve some of the challenges encountered in this research project.

### Addressing Alignment and Orientation Challenges

One of the key challenges I faced was ensuring precise orientation and alignment of the RC-car during movement. To tackle this, I utilized the MPU6050 gyroscope, aiming to correct alignment issues and achieve accurate 90-degree turns.

- **Utilizing the MPU6050 Gyroscope for Precise Orientation:** My first set of experiments focused on leveraging the gyroscope to correct the car's orientation for accurate navigation. This approach was pivotal in my attempts to ensure the RC-car could navigate mazes with high precision.
  - To address alignment issues when attempting precise 90-degree turns, I explored the potential of the MPU6050 gyroscope to adjust the car's movement based on its orientation. This experiment aimed to refine my control over the vehicle's navigation through the maze ([View Test 1](#), [View Test 2](#)).

- Further testing focused on using the gyroscope for realigning the car's forward movement, aiming to rectify the persistent ~3-degree offset. Despite my efforts, completely eliminating this offset proved challenging, showcasing the complexities of simulating real-world physics (View Test 1, View Test 2, View Test 3).

### Enhancing Movement Precision with Encoders

The pursuit of enhancing the RC-car's movement precision led us to experiment with rotary encoders. These devices were integrated to measure wheel rotations accurately, aiming to improve straight-line movements and correct the noted ~3-degree offset.

- **Experimenting with Rotary Encoders:** I introduced rotary encoders to my setup, hoping to gain more precise control over the car's movements by accurately measuring wheel rotations. This experiment represented a significant effort to refine the vehicle's navigation capabilities by ensuring more accurate movement and orientation.
  - Initial tests with a new RC-car model, equipped with an encoder and a more powerful motor, showed promise in addressing the forward movement precision. However, the addition of extra components increased the vehicle's weight, impacting its movement and reintroducing the alignment challenge (View Test 1, View Test 2).
  - Despite an encouraging start, a malfunction with one of the encoders halted further tests using this specific setup, highlighting the practical challenges of hardware reliability in real-world applications (View Test).

### Real-World Application Tests

Moving beyond controlled environments, I conducted tests in both outdoor and indoor settings to evaluate the RC-car's performance in real-world conditions. These tests were crucial for assessing the practical application of my research findings.

- **Outdoor and Indoor Maze Tests:** Real-world testing scenarios presented unique challenges, such as varying surface textures and unpredictable environmental conditions, which significantly impacted the RC-car's navigation capabilities.
  - The outdoor test attempted to navigate the RC-car on uneven surfaces, where surface texture variations greatly affected its performance. This test underscored the importance of environmental factors in autonomous navigation (View Test 1, View Test 2).
  - Indoor testing provided a more controlled environment, allowing us to closely monitor and adjust the RC-car's navigation strategies. Despite the controlled conditions, these

tests highlighted the challenge of accurately translating simulation models to real-world applications, reflecting on the complexities of sim-to-real transfer (View Test 1, View Test 2, View Test 3, View Test 4).

## Guest Speakers

### **Innovations and Best Practices in AI Projects by Jeroen Boeye at Faktion**

Jeroen Boeye's comprehensive lecture, representing Faktion, offered profound insights into the symbiotic relationship between software engineering and artificial intelligence in the realm of AI solutions development. He emphasized the critical importance of not merely focusing on AI technology but also on the software engineering principles that underpin the development of robust, scalable, and maintainable AI systems. This approach ensures that AI solutions are not only technically proficient but also practical and sustainable in long-term applications.

The discussion delved into various aspects of AI applications, notably highlighting Chatlayer's contributions to the field of conversational AI. Jeroen detailed how Chatlayer enhances chatbot functionalities through dynamic conversational flows, significantly improving the accuracy and contextuality of user interactions. Another spotlight was on Metamaze, praised for its innovative approach to automating document processing. By generating concise summaries from documents and emails, Metamaze exemplifies the potential of supervised machine learning to streamline and improve administrative tasks.

Jeroen provided a clear roadmap for the successful implementation of AI projects, emphasizing the importance of validating business cases and adopting a problem-first approach. He highlighted the necessity of quality data as the foundation for any AI initiative and discussed strategies for overcoming data limitations creatively. The lecture also touched on the crucial mindset of embracing failure as a stepping stone to innovation, stressing the importance of open communication with stakeholders about challenges and setbacks.

The lecture further explored several practical use cases, demonstrating the versatility and potential of AI across various industries. From the detection of solar panels and unauthorized pools to the damage inspection of air freight containers and early warning systems for wind turbine gearboxes, Jeroen showcased how AI can address complex challenges through innovative data sourcing, synthetic data generation, and anomaly detection techniques. He also presented case studies on energy analysis in brick ovens and egg incubation processes, highlighting the critical role of data preprocessing and the application of machine learning models to enhance efficiency and outcomes.

Key takeaways from Jeroen's lecture underscored the importance of mastering data preprocessing and treating data as a dynamic asset to tailor AI models more precisely to specific needs. He offered

practical advice on operational efficiency, including the use of host mounts for code integration and Streamlit for dashboard creation, to streamline development processes.

In conclusion, Jeroen Boeye's lecture provided a rich and detailed perspective on the integration of AI technologies in real-world scenarios. His insights into the critical importance of software engineering principles, combined with a deep understanding of AI's capabilities and limitations, offered valuable guidance for developing effective, sustainable AI solutions. This lecture not only highlighted the current state and future directions of AI but also imparted practical wisdom on navigating the complexities of AI project implementation.

### **Pioneering AI Solutions at Noest by Toon Vanhoutte**

Toon Vanhoutte's enlightening lecture, representing Noest, a notable entity within the Cronos Group, shared profound insights into the harmonious blend of artificial intelligence and software engineering in crafting state-of-the-art business solutions. With a strong team of 56 local experts, Noest prides itself on its pragmatic approach to projects, aiming for a global impact while emphasizing craftsmanship, partnership, and pleasure as its foundational pillars. This philosophy extends across their diverse service offerings, including application development, cloud computing, data analytics, AI innovations, low-code platforms, ERP solutions, and comprehensive system integrations, all underpinned by a strong partnership with Microsoft.

A particularly captivating case study presented was a project for a packaging company, aimed at revolutionizing image search capabilities based on product labels. The project encountered various challenges, from dealing with inconsistent PDF formats to managing large file sizes and overcoming processing limitations. These hurdles were adeptly navigated using a combination of Azure Blob Storage for data management and event-driven processing strategies for efficient and cost-effective solutions, showcasing Noest's adeptness in leveraging cloud technologies to solve complex problems.

Enhancing searchability of images, a task that encompassed recognizing text and objects within images, was another significant challenge tackled by employing Azure AI Search, complemented by the power of Large Language Models (LLMs) and vector search techniques. This innovative approach enabled nuanced search functionalities beyond traditional text queries, demonstrating the advanced capabilities of AI in understanding and interpreting complex data.

Toon's lecture further delved into the advancements in semantic search, revealing how keyword, vector, and hybrid searches, alongside semantic ranking, could dramatically enhance the accuracy and contextuality of search results. Through practical demonstrations, including comparisons between OCR and GPT-4 vision, attendees were shown the potential of AI to transcend basic search functionalities and offer deeper, more meaningful insights based on semantic understanding.

A key takeaway from the lecture was the importance of setting realistic expectations with clients regarding AI's capabilities and potential inaccuracies, emphasizing the experimental nature of these technologies. The journey through AI's evolving landscape highlighted the necessity of prompt engineering, the challenges of navigating an immature yet rapidly developing field, and the crucial role of client education in managing expectations around the capabilities of AI technologies like GPT.

In conclusion, Toon Vanhoutte's presentation not only showcased Noest's cutting-edge work in AI and software engineering but also imparted valuable lessons on innovation, the importance of adaptable problem-solving strategies, and the need for continuous learning in the ever-evolving AI domain. It was a testament to Noest's commitment to pushing the boundaries of technology to create impactful, pragmatic solutions that leverage the full spectrum of AI's potential.

## Installation Steps

This section outlines the required steps to install and set up the project environment. Adherence to these instructions will ensure the successful deployment of the autonomous navigation system.

### Prerequisites

Before initiating the setup process, ensure the following prerequisites are met:

- **Git:** Necessary for cloning the project repository.
- **Docker:** Utilized for containerizing the web application and ensuring a consistent runtime environment.
- Optionally, **Python 3.11** and **pip** may be installed along with the dependencies listed in `requirements.txt` for running the project without Docker.

### Repository Setup

To clone the repository and navigate to the project directory, execute the following commands:

```
git clone https://github.com/driessenslucas/researchproject.git  
cd researchproject
```

### ESP32 Setup

#### Hardware Installation

1. **Hardware Connections:** Follow the instructions provided in the Hardware Installation Guide for connecting the ESP32 modules.

## Software Configuration

2. **Library Installation:** Install the ESP32\_SSD1306 library to support the OLED display functionality.
3. **Code Upload:** Transfer the scripts located in the esp32 folder to the ESP32 device. Modify the WiFi settings in the script to match your local network configuration for connectivity.

## Web Application Setup

**Note:** To ensure a seamless setup of the virtual display, it is recommended to execute `docker-compose down` following each session.

### Steps:

1. The web application's source code is stored within the web app directory. Access this directory:  
`cd ./web_app/`
2. To launch the Docker containers, use the following commands:  
`docker-compose up -d`

## Usage Instructions

1. Access the web application by navigating to `http://localhost:8500` or `http://localhost:5000` on your web browser.
2. Enter the ESP32's IP address within the web app and select the desired model for deployment.
3. The system provides an option for a virtual demonstration, allowing for operation without engaging the physical vehicle.
4. Initiate the maze navigation by clicking the Start Maze button.

A demonstration of the project is available [here](#).

## Additional Information: Model Training

- Opt between utilizing a pre-trained model or conducting new training sessions using the script available in `train`.
- This training script is optimized for resource efficiency and can be executed directly on the Raspberry Pi.
- Upon completion, you will be prompted to save the new model. If saved, it will be stored within the `models` directory of the `web_app` folder.

## References

- [1] G. Brockman et al., “OpenAI Gym,” arXiv preprint arXiv:1606.01540, 2016.
- [2] A. Dosovitskiy et al., “CARLA: An Open Urban Driving Simulator,” in Proceedings of the 1st Annual Conference on Robot Learning, 2017.
- [3] H. Van Hasselt, A. Guez, and D. Silver, “Deep Reinforcement Learning with Double Q-learning,” in Proceedings of the AAAI Conference on Artificial Intelligence, 2016.
- [4] J. Schulman et al., “Proximal Policy Optimization Algorithms,” arXiv preprint arXiv:1707.06347, 2017.
- [5] J. Tobin et al., “Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World,” in 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2017.
- [6] K. Bousmalis et al., “Using Simulation and Domain Adaptation to Improve Efficiency of Deep Robotic Grasping,” in IEEE International Conference on Robotics and Automation (ICRA), 2018.
- [7] Y. Pan and Q. Yang, “A Survey on Transfer Learning,” IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 10, pp. 1345-1359, Oct. 2010.
- [8] A. A. Rusu et al., “Sim-to-Real Robot Learning from Pixels with Progressive Nets,” in Proceedings of the Conference on Robot Learning, 2016.
- [9] S. James et al., “Sim-to-Real via Sim-to-Sim: Data-efficient Robotic Grasping via Randomized-to-Canonical Adaptation Networks,” in Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), 2019.
- [10] F. Sadeghi and S. Levine, “(CAD)<sup>2</sup>RL: Real Single-Image Flight without a Single Real Image,” in Proceedings of Robotics: Science and Systems, 2016.
- [11] “Self Driving and Drifting RC Car using Reinforcement Learning,” YouTube, Aug. 19, 2019. [Online Video]. Available: <https://www.youtube.com/watch?v=U0-Jswwf0hw>. [Accessed: Jan. 29, 2024].
- [12] Q. Song et al., “Autonomous Driving Decision Control Based on Improved Proximal Policy Optimization Algorithm,” Applied Sciences, vol. 13, no. 11, Art. no. 11, Jan. 2023. [Online]. Available: <https://www.mdpi.com/2076-3417/13/11/6400>. [Accessed: Jan. 29, 2024].
- [13] DailyL, “Sim2Real\_autonomous\_vehicle,” GitHub repository, Nov. 14, 2023. [Online]. Available: [https://github.com/DailyL/Sim2Real\\_autonomous\\_vehicle](https://github.com/DailyL/Sim2Real_autonomous_vehicle). [Accessed: Jan. 29, 2024].
- [14] “OpenGL inside Docker containers, this is how I did it,” Reddit, r/docker. [Online]. Available: [https://www.reddit.com/r/docker/comments/8d3qox/opengl\\_inside\\_docker\\_containers\\_this\\_is\\_how\\_i\\_did/](https://www.reddit.com/r/docker/comments/8d3qox/opengl_inside_docker_containers_this_is_how_i_did/). [Accessed: Jan. 29, 2024].

- [15] M. A. Dharmasiri, “Micromouse from scratch | Algorithm- Maze traversal | Shortest path | Floodfill,” Medium, [Online]. Available: <https://medium.com/@minikiraniamayadharmasiri/micromouse-from-scratch-algorithm-maze-traversal-shortest-path-floodfill-741242e8510>. [Accessed: Jan. 29, 2024].
- [16] “Reinforcement Learning with Multi-Fidelity Simulators – RC Car,” YouTube, Dec. 30, 2014. [Online Video]. Available: [https://www.youtube.com/watch?v=c\\_d0ls3bxXA](https://www.youtube.com/watch?v=c_d0ls3bxXA). [Accessed: Jan. 29, 2024].
- [17] W. Zhao, J. P. Queralta, and T. Westerlund, “Sim-to-Real Transfer in Deep Reinforcement Learning for Robotics: A Survey,” in 2020 IEEE Symposium Series on Computational Intelligence (SSCI), Dec. 2020, pp. 737–744. [Online]. Available: <https://arxiv.org/pdf/2009.13303.pdf>.
- [18] R. S. Sutton and A.G. Barto, Reinforcement Learning: An Introduction, 2nd ed. Cambridge, MA: The MIT Press, 2018.
- [19] H. van Hasselt, A. Guez, D. Silver, et al., “Deep Reinforcement Learning with Double Q-learning,” arXiv preprint arXiv:1509.06461, 2015.
- [20] Papers With Code, “Double DQN Explained,” [Online]. Available: <https://paperswithcode.com/method/double-dqn>.
- [21] D. Jayakody, “Double Deep Q-Networks (DDQN) - A Quick Intro (with Code),” 2020. [Online]. Available: <https://dilithjay.com/blog/2020/04/18/double-deep-q-networks-ddqn-a-quick-intro-with-code/>.