ENHANCING THE PERFORMANCE OF DRL ALGORITHMS USING CHANNEL ATTENTION.

A Project Report submitted in partial fulfilment of the requirements for the award of the degree of

Bachelor of Technology

in

Computer Science and Engineering
by

P. Shreyas Reddy (112215136)
M. Sai Dinakara Reddy (112215117)
B. Himavath Sai (112215039)
Anish Kumar S (112215022)
Samhith Kalari (112215160)

Under the Supervision of: Mrs. Anu Priya
Semester: 5th

Department of Computer Science and Engineering
Indian Institute of Information Technology, Pune

(An Institute of National Importance by an Act of Parliament)

November 2024

BONAFIDE CERTIFICATE

This is to certify that the project report entitled **"Enhancing the performance of DRL Algorithms using channel attention."** submitted by **P. Shreyas Reddy** bearing the **MIS No: 112215136**, **M.Sai Dinakara Reddy** bearing the **MIS No: 112215117**, **Anish Kumar S** bearing the **MIS No: 112215022**, **B. Himavath Sai** bearing the **MIS No: 112215039**, **Samhith Kalari** bearing the **MIS No: 112215160**, in completion of their project work under the guidance of **Ms Anupriya Mam** is accepted for the project report submission in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in the **Department of Computer Science and Engineering**, Indian Institute of Information Technology, Pune (IIIT Pune), during the academic year **2024-25**.

Supervisor's Name.
Project Supervisor : Mrs. Anu Priya

Designation of the Supervisor : Adjunct Assistant Professor

Department of the Supervisor : Department of CSE

IIIT Pune.

HOD Name :

Dr. Bhupendra Singh

Head Of Department : CSE

Project Viva-voce held on 6th November 2024

UNDERTAKING FOR PLAGIARISM

We **P. Shreyas Reddy, M. Sai Dinakara Reddy, B. Himavath Sai, Anish Kumar S , Samhith Kalari** solemnly declare that research work presented in the **report/ dissertation** titled **"Enhancing the performance of DRL Algorithms using channel attention."** is solely **our** research work with no significant contribution from any other person. Small contribution/help wherever taken has been duly acknowledged and that complete report/dissertation has been written by **us**. We understand the zero tolerance policy of **Indian Institute of Information Technology, Pune** towards plagiarism. Therefore **we** declare that no portion of my **report/dissertation** has been plagiarised and any material used as reference is properly referred/cited. I undertake that if I am found guilty of any formal plagiarism in the above titled thesis even after award of the degree, the Institute reserves the rights to withdraw/revoke my **B.Tech** degree.

Student's Name and Signature with Date

P. Shreyas Reddy _____

M. Sai Dinakara Reddy _____

B. Himavath Sai _____

Anish Kumar S _____

Samhith Kalari _____

CONFLICT OF INTEREST

Manuscript title : Enhancing the performance of DRL Algorithms using channel attention.

The authors whose names are listed immediately below certify that they have NO affiliations with or involvement in any organisation or entity with any financial interest (such as honoraria; educational grants; participation in speakers' bureaus; membership, employment, consultancies, stock ownership, or other equity interest; and expert testimony or patent-licensing arrangements), or non-financial interest (such as personal or professional relationships, affiliations, knowledge or beliefs) in the subject matter or materials discussed in this manuscript.

Student's Name and Signature with Date

P. Shreyas Reddy _____

M. Sai Dinakar Reddy _____

B. Himavath Sai _____

Anish Kumar S _____

Samhith Kalari _____

ACKNOWLEDGEMENT

This project would not have been possible without the help and cooperation of many. I would like to thank the people who helped me directly and indirectly in the completion of this project work.

First and foremost, I would like to express my gratitude to our honourable Director, **Dr. Prem Lal Patel**, for providing his kind support in various aspects. I would like to express my gratitude to my project guide **Mrs Anu Priya**, **Department of CSE**, for providing excellent guidance, encouragement, inspiration, constant and timely support throughout this **B.Tech Project**. I would like to express my gratitude to the Head of Department **Dr. Bhupendra Singh**, **Department of CSE**, for providing his kind support in various aspects. I would also like to thank all the faculty members in the **Department of CSE** and my classmates for their steadfast and strong support and engagement with this project.

ABSTRACT

This project investigates the role of channel attention and feature selection in enhancing reinforcement learning (RL) models, particularly within deep Q-learning frameworks such as DQN, SARSA, DDQN, and D3QN, along with their channel attention-enhanced variants DQN-CA, DDQN-CA and D3QN-CA. By incorporating channel attention mechanisms, we aim to improve the adaptability, generalisation, and computational efficiency of RL models across diverse tasks, with a focus on exploring both performance gains and potential limitations introduced by attention mechanisms. Effective feature selection allows RL models to focus on the most relevant state representations, which is especially important in high-dimensional environments where extraneous data can introduce noise and slow learning. By identifying and using only the most informative features, RL agents can achieve faster convergence, reduce computational load, and improve overall sample efficiency.

Our methodology involves implementing attention-based modifications to each model and evaluating their impact using key performance metrics, including cumulative reward, sample efficiency, and task-specific accuracy. Further, we address research gaps surrounding attention-based RL models, such as the comparative effectiveness of channel attention against traditional models, the effect of attention on learning stability, and its computational overhead. This project also examines the impact of channel attention on long-term task planning and generalisation capabilities across varied RL tasks, while introducing new evaluation metrics to capture the nuanced benefits of attention in RL.

By systematically assessing the impact of channel attention in these contexts, we aim to advance the field's understanding of attention-driven enhancements in RL. The findings of this work are intended to inform the design of more robust and efficient RL systems, providing a foundation for further exploration into adaptive learning mechanisms in complex environments.

Keywords:

Reinforcement Learning, DQN, SARSA, DDQN, D3QN, Channel Attention, Path Planning, Navigation, On-Policy, Off-Policy.

TABLE OF CONTENTS

Abstract i

List of Figures/Symbols/Nomenclature 7-8

1. Introduction 9

1.1 Overview of work 9

1.2 Motivation of work 9

1.3 Literature Review 10-11

1.4 Research Gap. 11

2. Problem Statement 12

2.1 Research Objectives 12

2.2 Analysis And Design 13

3. Proposed Work 14

3.1 Methodology of work. 14-17

3.2 Hardware & Software specifications. 17

3.3 Dataset Description. 18

4. Results and Discussion 19-27

5. Conclusion and Future Scope 28-29

References 30-32

LIST OF FIGURES / SYMBOLS/ NOMENCLATURE

Fig_1. Demonstration of Feature selection. Page 14
Fig_2. Demonstration of layers in channel attention. Page 17

Obstacle Display :

Fig_3. Demonstration of 2 Obstacle Environment. Page 18
Fig_4. Demonstration of 3 Obstacle Environment. Page 18
Fig_5. Demonstration of Obstacles in Complex Environment. Page 18

Accumulated Rewards :

Fig_6. Demonstration of accumulated rewards in 2 obstacle environment between SARSA vs DQN vs DQN-CA. Page 19
Fig_7. Demonstration of accumulated rewards in 2 obstacle environment between DDQN Vs DDQN-CA. Page 19
Fig_8. Demonstration of accumulated rewards in 2 obstacle environment between D3QN Vs D3QN-CA. Page 19

Steps Taken :

Fig_9. Demonstration of steps taken in 2 obstacle environment between SARSA vs DQN vs DQN-CA. Page 20
Fig_10. Demonstration of steps taken in 2 obstacle environment between DDQN Vs DDQN-CA. Page 20
Fig_11. Demonstration of steps taken in 2 obstacle environment between D3QN Vs D3QN-CA. Page 20

Paths Taken :

Fig_12. Demonstration of paths taken in 2 obstacle environment between SARSA vs DQN vs DQN-CA. Page 21
Fig_13. Demonstration of paths taken in 2 obstacle environment between DDQN Vs DDQN-CA. Page 21
Fig_14. Demonstration of paths taken in 2 obstacle environment between D3QN Vs D3QN-CA. Page 21

Accumulated Rewards :

Fig_15. Demonstration of accumulated rewards in 3 obstacle environment between SARSA vs DQN vs DQN-CA. Page 22
Fig_16. Demonstration of accumulated rewards in 3 obstacle environment between DDQN Vs DDQN-CA. Page 22
Fig_17. Demonstration of accumulated rewards in 3 obstacle environment between D3QN Vs D3QN-CA. Page 22

Steps Taken :

Fig_18.Demonstration of steps taken in 3 obstacle environment between SARSA vs DQN vs

DQN-CA. Page 23
Fig_19. Demonstration of steps taken in 3 obstacle environment between DDQN Vs DDQN-CA. Page 23
Fig_20. Demonstration of steps taken in 3 obstacle environment between D3QN Vs D3QN-CA. Page 23

Paths Taken :

Fig_21. Demonstration of paths taken in 3 obstacle environment between SARSA vs DQN vs DQN-CA. Page 24
Fig_22. Demonstration of paths taken in 3 obstacle environment between DDQN Vs DDQN-CA. Page 24
Fig_23. Demonstration of paths taken in 3 obstacle environment between D3QN Vs D3QN-CA. Page 24

Accumulated Rewards :

Fig_24. Demonstration of accumulated rewards in Complex obstacle environment between SARSA vs DQN vs DQN-CA. Page 25
Fig_25. Demonstration of accumulated rewards in Complex obstacle environment between DDQN Vs DDQN-CA. Page 25
Fig_26. Demonstration of accumulated rewards in Complex obstacle environment between D3QN Vs D3QN-CA. Page 25

Steps Taken :

Fig_27.Demonstration of steps taken in Complex obstacle environment between SARSA vs DQN vs DQN-CA. Page 26
Fig_28. Demonstration of steps taken in Complex obstacle environment between DDQN Vs DDQN-CA. Page 26
Fig_29. Demonstration of steps taken in Complex obstacle environment between D3QN Vs D3QN-CA. Page 26

Paths Taken :

Fig_30. Demonstration of paths taken in Complex obstacle environment between SARSA vs DQN vs DQN-CA. Page 27
Fig_31. Demonstration of paths taken in Complex obstacle environment between DDQN Vs DDQN-CA. Page 27
Fig_32. Demonstration of paths taken in Complex obstacle environment between D3QN Vs D3QN-CA. Page 27

Abbreviations used :

DQN : Deep Q-Network, **DDQN :** Double Deep Q-Network, **D3QN :** Dueling Double Q-Network, **SARSA :** State–action–reward–state–action, **CA :** Channel Attention

.

Chapter 1
INTRODUCTION

1.1 Overview of work

This paper works with various reinforcement learning models, such as SARSA, DQN, DDQN, and D3QN, alongside specialised attention mechanisms like channel attention, to enhance the agent's learning and decision-making capabilities on a custom environment with discrete movements. DQN is a deep reinforcement learning approach that applies a neural network to approximate Q-values so agents can decide in high state-action spaces. Double DQN is another improvement from DQN because it eliminates the overestimation bias within the action-value estimate, while channel attention introduces attention mechanisms in order to allow the network to focus more on the important states and actions that it should emphasise while improving the decision-making process. This project compares the algorithms to determine which model maximises rewards, stability, and convergence rate best.

1.2 Motivation of Work

The motivation behind this project emerges from the growing demand for autonomous systems across various real-world applications, including self-driving cars, robotic delivery systems. Practical need to evaluate the strengths and weaknesses of different reinforcement learning (RL) models, given their increased use in realistic scenarios. Interest in developing RL models that are adaptable and effective in complex, dynamic environments. So far, although DQN and SARSA are established algorithms, there are newer models such as DDQN and D3QN that hold out more hope for better performance and stability. Applying attention mechanisms ob the models can help build agents in real-world scenarios that learn effective policies but are stable and efficient as training progresses. For instance, applications in robotics and game playing require robust models for balancing exploration and exploitation efficiently. This study would analyse how these models relate to each other, making their findings useful in picking a suitable algorithm for use with a specific RL application. The importance of feature selection and its effect on reinforcement learning algorithms is our aim.

.

1.3 Literature Review

A detailed description of how blending on-policy (SARSA) and off-policy (Q-learning) strategies can balance exploration and exploitation has been given in the paper [3]. It talks about the difference in on and off policy algorithms and how the approaches can enhance stability. This research elaborates on how SARSA's on-policy learning approach makes it effective for cautious path planning and obstacle avoidance. The on-policy nature of SARSA enables it to make conservative decisions aligned with the policy, which is particularly valuable for real-time, dynamic navigation scenarios. Additionally [5] underscores SARSA's stability in dynamic environments, showing its utility in applications where safe navigation is critical. However SARSA can lead to suboptimal policies, especially in environments where exploration (non-greedy actions) is costly or risky, as it prioritises the safety of learned policies over potential rewards.

A better and revised algorithm, DQN and its fundamental principles are well-explained in [11] which discusses how DQNs leverage neural networks to approximate Q-values in high-dimensional action spaces. DQN's ability to handle complex state-action pairs allows for effective decision-making, particularly useful for autonomous navigation and robotic control in unknown environments, directly supporting the objectives of your project. The same source also details DDQN, which reduces the overestimation bias inherent in DQN by decoupling action selection and evaluation. This improvement yields a more stable learning process, ideal for scenarios in your project that involve continuous obstacle avoidance and path refinement. Another study, [10] shows how DDQN's accurate Q-value predictions can enhance navigation efficiency, reducing unnecessary detours and optimising paths.

In [12], D3QN's architecture introduces separate estimations for state value and action advantage, helping the model focus on critical actions even when some are irrelevant in certain states. D3QN's structure, combined with prioritised experience replay, is particularly beneficial in sparse reward environments, enabling your project to converge faster and choose optimal actions in complex navigation tasks. This paper highlights D3QN-PER as an effective approach for dynamic path planning. The paper [9] explores the use of a DRL model, a dueling double deep Q-network (D3QN), to allow robots to navigate and avoid obstacles using monocular vision. A comparison of different Q-network architectures to determine the best configuration for each algorithm has been done in the paper [13]. It concludes that performance metrics and algorithm insights from DQN vs. D3QN provides a strong foundation for selecting the optimal approach for efficient path planning.

An overview of attention mechanisms including channel attention has been presented in [2]. The importance of feature selection and different attention mechanisms has been listed in this paper making it a crucial paper for our study. Channel attention is used to selectively emphasise important features, helping the model prioritise relevant sensory inputs and improving obstacle detection accuracy. For reinforcement learning tasks in our project, integrating channel attention could enhance the model's ability to discern critical visual data,

thus improving overall navigation. It concludes that attention mechanisms are crucial in many domains such as image classification, object detection and in our case optimising path planning and SLAM systems. [17] offers strategies to integrate attention layers for more efficient decision-making and collaboration, enhancing RL model performance in complex tasks. The importance of attention mechanisms has been dated in the paper [15]. This paper provides a robust framework by showcasing how attention-weighted mechanisms can dynamically emphasise critical features. It also provides insights into applying attention mechanisms in reinforcement learning by dynamically weighting relevant features to improve decision-making. Combining attention mechanisms with HRL's task decomposition might further improve decision-making by allowing the model to focus on critical spatial features at different levels of abstraction as mentioned in [14]. The paper uses a hierarchical reinforcement learning approach, breaking tasks into subtasks managed by high-level goal-setting policies and low-level action policies, enabling effective navigation and problem-solving in complex 3D environments.
These papers provide a direction to the work we have done and the methodologies used in our project. They validate our project's objective and provide a deeper understanding into reinforcement learning and its application.

1.4 Research Gap

Under-explored Attention Mechanisms in RL : Although channel attention mechanisms are widely used in deep learning for their ability to focus on critical features, they have not been extensively applied in reinforcement learning. Channel attention allows models to prioritise important channels of information, which can potentially improve exploration efficiency and reward optimisation.

Efficiency : In real world applications where the input features are too many, the agent/robot takes too much time to make decisions and this is where feature selection comes in. Selecting important features and discarding others saves a lot of time and cost and this can be made possible using attention mechanisms.

Comparative Gap in Q-learning Models: Reinforcement learning has made great strides, yet there is still a gap in understanding how different Q-learning-based models perform relatively to each other under similar conditions. Most research work focuses on single-algorithm performance, making it rather difficult to identify the individual strengths and limitations of the model when tested in similar environments.

Challenges in Model Selection: Appropriate models should be selected in applications, which require such complex environments like robotics and autonomous driving. Otherwise, without any comparison studies, the right choice of an appropriate model, ensuring stability and efficiency, and still being able to adapt for such environments becomes impossible.

Chapter 2
PROBLEM STATEMENT

In reinforcement learning, deep Q-networks (DQNs) are widely used for solving complex decision-making tasks by approximating optimal policies. However, standard DQNs often struggle to effectively capture important features within high-dimensional state representations, particularly when task-relevant details are distributed unevenly across the feature space. This limitation can lead to suboptimal performance, especially in scenarios requiring a focus on specific features or regions of the input space.
This project explores the integration of channel attention mechanisms within DQN models to dynamically highlight critical features during the decision-making process. By assigning variable importance to channels in the feature maps, channel attention enables the network to focus more effectively on task-relevant aspects, potentially enhancing action selection and overall policy performance. We aim to assess the impact of channel attention on DQN, SARSA, DDQN, and D3QN models by comparing standard architectures with their channel-attention-augmented counterparts. The study will measure improvements in task performance, policy efficiency, and training stability, providing insights into the benefits of attention mechanisms in reinforcement learning frameworks.

2.1 Research Objectives

This paper sets the following key objectives:

Difference between On-policy (SARSA) and off-policy approaches, such as DQN, DDQN, all share trade-offs between stability, computational efficiency, and adaptability.

The evaluation of SARSA, DQN, DDQN and D3QN algorithms with respect to a comparison of their performance on three different simulated environment with the newly advanced variants that implement channel attention.

Investigate feature selection and channel attention mechanisms in reinforcement learning, with a focus on their relevance in affecting model performance.

Compare convergence rates, cumulative rewards, and training stability for each model to determine which one represents the most complex state-action space.

.

2.2 Analysis and Design

The main challenges include managing high-dimensional environments and being able to assess if channel attention improves decision-making efficiency by enabling the ability to select on important features when exploring a custom environment.

Core Challenges:

Dimensionality: The high-dimensional state-action space demands models that can efficiently generalise across states and avoid exhaustive exploration.
Exploration vs. Exploitation: Effective learning requires a strategic balance between exploring new actions and exploiting known paths to optimise rewards.
Feature Prioritisation: Channel attention in D3QN-CA requires an effective approach to focus on relevant features, enhancing learning efficiency.

Design Decisions :

Model Selection:

SARSA : Included as an on-policy approach, beneficial for adaptive policies.
DQN : Chosen for its foundational role in RL with deep Q-value approximation.
DDQN : Used to overcome the overestimation bias in DQN model.
D3QN : Combines the benefits of both Dueling Networks and Double Q-learning to further improve stability and performance in complex environments.
Channel Attention : Feature selection to get better path.

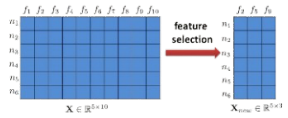
Policy Selection:

SARSA: On-policy Q-updates for consistent policy-driven learning.
DQN/DDQN: Epsilon-greedy for balanced exploration-exploitation.

.

Chapter 3
PROPOSED WORK

Implement channel attention within DQN architectures to highlight critical channels in the feature map. This will involve integrating a channel attention module after convolutional layers to dynamically assign importance to specific features. Our task starts from figuring out the difference in the working of on and off policy algorithms on the environments and ways to make the path efficient. This project proposes a comparative study of DQN, SARSA, DDQN and D3QN models along with the respective models with CA in a simulated environment to evaluate their performance in terms of reward maximisation and stability. The results will provide insights into each model's strengths and limitations within identical conditions (Environments).



Feature Selection

Fig 1 : Feature Selection

3.1 Methodology of work

1) Training Setup :

An initial application of SARSA and and understanding of its drawbacks has been noticed. SARSA is an on-policy algorithm, meaning it learns from actions it actually takes rather than from the optimal actions. This can lead to suboptimal policies, especially in environments where exploration (non-greedy actions) is costly or risky, as it prioritises the safety of learned policies over potential rewards. We then moved on to DQN and variants which provided a better result. Each of these four models: DQN, SARSA, DDQN and D3QN was trained on a simulation setup of three environments. The design of this environment is in such a manner that it allows maximisation of cumulative rewards acquired during agents navigating through the obstacles leading them to the goal states. We simulated three environments, Fig 3 shows an environment with two obstacles. Fig 4 depicts an environment that has three obstacles and Fig 5 shows a complex environment. These environments have been designed to test the different algorithms and to provide a comparison. The starting position can be seen on the environment as (10, 0) and the target as (90,100).

2) Policies and Learning Strategy :

a) SARSA : This is an on policy algorithm that learns Q-value based on the current action and which action the agent will take next. Close to the agent's behaviour and policy in time learning, it develops consistency for learned actions.

b) DQN (Deep Q-Network) : DQN, introduced by DeepMind, is a reinforcement learning algorithm that combines Q-learning with deep neural networks to handle environments with large state spaces. It approximates the Q-value function, which estimates the future rewards of taking specific actions in given states.

Key Components:

Experience Replay: DQN uses a replay buffer to store past experiences (state, action, reward, next state) and samples mini-batches for training, which breaks correlation between sequential samples and stabilises training.

Target Network: DQN maintains a second network, called the target network, to generate stable Q-value targets. This target network is updated periodically to follow the weights of the primary Q-network, reducing oscillations in Q-values.

Drawbacks: DQN often overestimates Q-values due to maximisation bias during updates, which can lead to unstable learning and suboptimal policies.

c) DDQN (Double DQN) : DDQN was developed to address the overestimation bias problem found in DQN. Overestimation occurs because, in DQN, the same network is used to both select and evaluate actions, which can lead to overly optimistic Q-values. Instead of using the maximum Q-value from the target network as in DQN, DDQN decouples action selection and action evaluation:

1.Action Selection: DDQN selects the best action in the next state using the online Q-network.

2.Action Evaluation: It then evaluates the selected action using the target Q-network to calculate the Q-value.

This separation reduces the risk of overestimation by not directly using the maximum Q-value for both selection and evaluation.

Advantages: DDQN provides more accurate Q-value estimates, leading to improved stability and performance compared to DQN.

d) D3QN (Dueling Double DQN) : D3QN (Dueling Double DQN) combines the benefits of both Dueling Networks and Double Q-learning to further improve stability and performance in complex environments. In D3QN, the Q-network is split into two separate streams:

1.Value Stream: Estimates the value V(s) of being in a particular state, representing the intrinsic value of a state regardless of the actions taken.

2.Advantage Stream: Estimates the advantage A(s, a) of each action in a given state, representing the benefit of taking a particular action over others in that state.

Advantages: The dueling architecture allows D3QN to learn more efficiently by focusing on the value of states and the advantages of actions separately, which can be particularly useful in states where the choice of action has little effect on the outcome. By combining this architecture with Double Q-learning, D3QN benefits from reduced overestimation bias and enhanced stability, making it well-suited for complex tasks.

5) Channel Attention :

Channels in a convolutional neural network (CNN) represent distinct feature maps that each capture unique aspects of the input data. When an input image is processed through convolutional layers, these channels capture different characteristics, such as edges, textures, or more complex patterns as the network deepens. In our DQN model, the conv2 layer generates a 32-channel output, meaning there are 32 separate feature maps, each representing specific features learned from the input data. Imagine a model is processing frames from a game environment. Certain channels might become particularly relevant for identifying objects or motion patterns that are key to the agent's decisions. For instance: If one channel has learned to recognise edges, and edges are critical in distinguishing objects, that channel might get a higher weight. Conversely, a channel detecting less relevant background textures may be assigned a lower weight, reducing its influence on the final decision.

By applying attention, the ChannelAttention mechanism ensures that channels with important features are emphasised, enabling the model to focus on the most informative aspects of the state, which can improve decision-making in your reinforcement learning context.

Working of channel attention :

1) Global Pooling :

• Average Pooling : This captures the "average" intensity across each channel, giving a general idea of the overall strength or presence of features in each channel.

• Max Pooling : This captures the "maximum" response within each channel, focusing on the most prominent feature locations.

These pooling operations condense the spatial information in each channel into a single representative value, thus compressing spatial details and retaining only the channel-wide significance.

2) Fully Connected (FC) Layers : After pooling, each pooled feature (one per channel) is passed through a set of fully connected layers. These layers learn to combine and weight the pooled values, capturing relationships between channels. The reduction in the number of channels by a factor of ratio (16 in this case) helps to keep the model lightweight and forces the network to focus on the most crucial relationships rather than modeling every minor detail. The fully connected layers apply transformations that learn which channels contain more useful features.

3) Sigmoid Activation : After passing through the fully connected layers, the values for each channel are scaled to [0, 1] with a sigmoid activation. This output can be thought of as the "importance" or "attention weight" for each channel. Channels with values closer to 1 are considered more important, and channels with values closer to 0 are less important for the current input data.

4) Attention Application : These attention weights are multiplied back with the original feature map. This enhances the features in channels with high weights and diminishes those in channels with low weights.

.

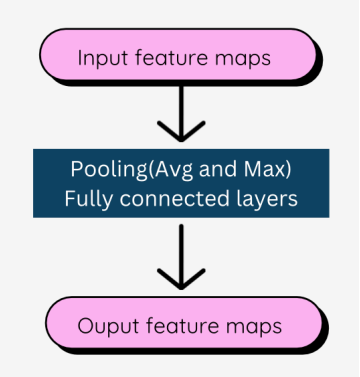


Fig 2 : Channel attention

4) Hyperparameter Tuning:

In order to optimise the performance, the learning rate, discount factor, batch size, and target network update frequency are adjusted in every model. That would ensure each model was well-adjusted to the environment and further assisted in maximising stability, convergence, and efficiency while training.

5) Evaluation Metrics:

For the performance of every model, the following evaluation is considered:

Cumulative Rewards: The total reward the agent acquires during it's course of action.

Steps to Goal: It calculates the number of steps toward goal states, which provides a sense of how efficient the learned policy is.

Reward Convergence Rate: This tracks the rate at which the model stabilises to reward convergence, which is indicative of effective learning.

Time taken to run each episode : It calculates the time taken to run an episode in milli seconds (ms) for each of the first 300 episodes that are run.

3.2 Hardware & Software specifications

The project is implemented in Python, using libraries like TensorFlow and PyTorch for neural network modeling. Keras for simpler model construction, and OpenAI Gym for environment simulation. Additional libraries such as NumPy and pandas are used for data manipulation, while Matplotlib is employed for visualising results. The training process leverages GPU acceleration, using an NVIDIA GPU to reduce computation time significantly and has been done on Kaggle.

.

3.3 Dataset Description

The environment is formed using a simulated grid world, against which the agents were trained to navigate without obstacles by reaching rewards. The states that represented rewards or penalties further challenged the agent to learn its optimal paths. The training dataset was dynamically generated and consisted of state-action-reward-next state tuples that would capture the agent's interaction with the environment. Agents will survey the grid, with a log of what they did and what rewards they collected, creating a dataset that informs subsequent decision-making. It is an excellent reinforcement learning environment since the agent has definite objectives-reach the reward states-but also challenges to test the model's ability to avoid pitfalls such as obstacles.

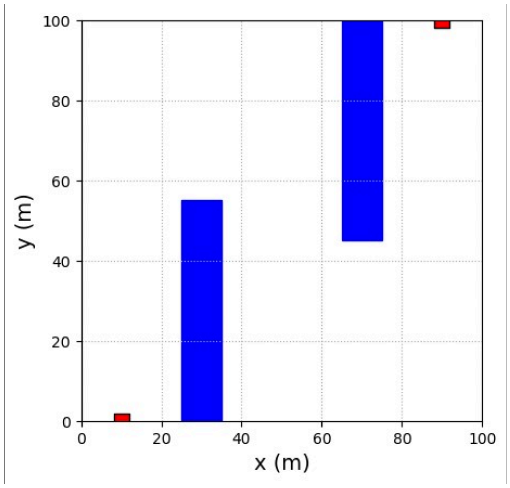


Fig 3. A Custom environment with 2 Obstacles.

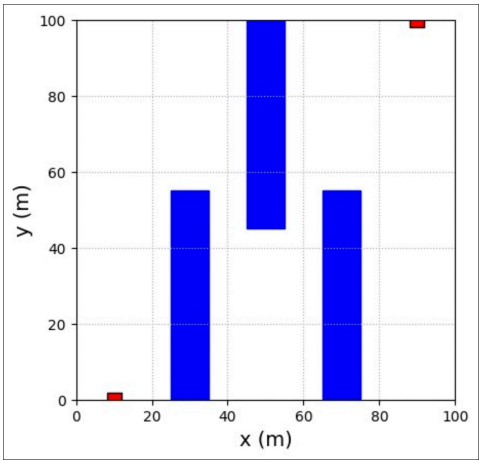


Fig 4. A custom Environment with 3 Obstacles.

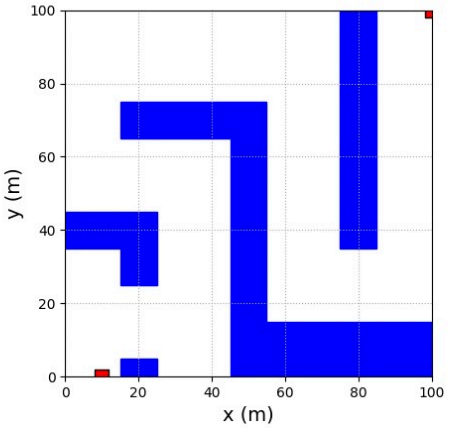


Fig 5. A custom Environment with random obstacles.

Chapter 4
RESULTS AND DISCUSSION

Comparison of accumulated rewards :

without channel attention vs with CA in a custom 2 Obstacle Environment.

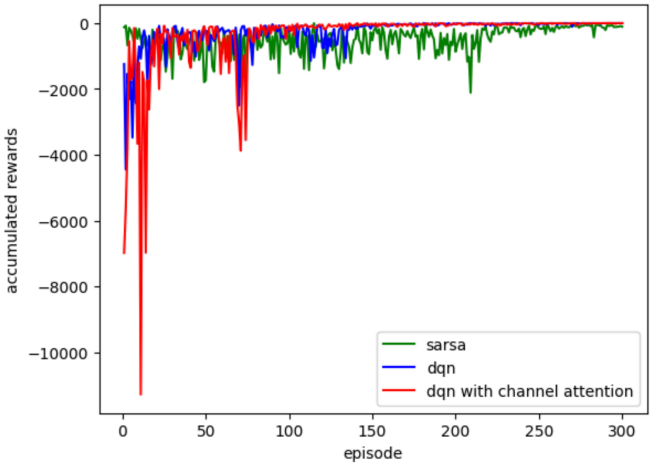


Fig 6. Sarsa Vs DQN Vs DQN-CA

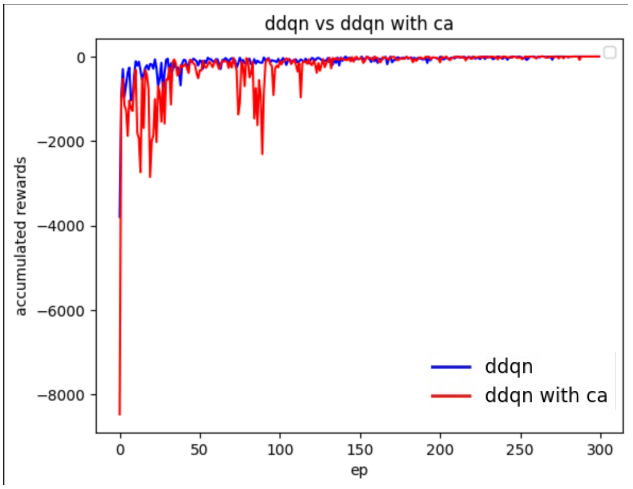


Fig 7. DDQN Vs DDQN-CA

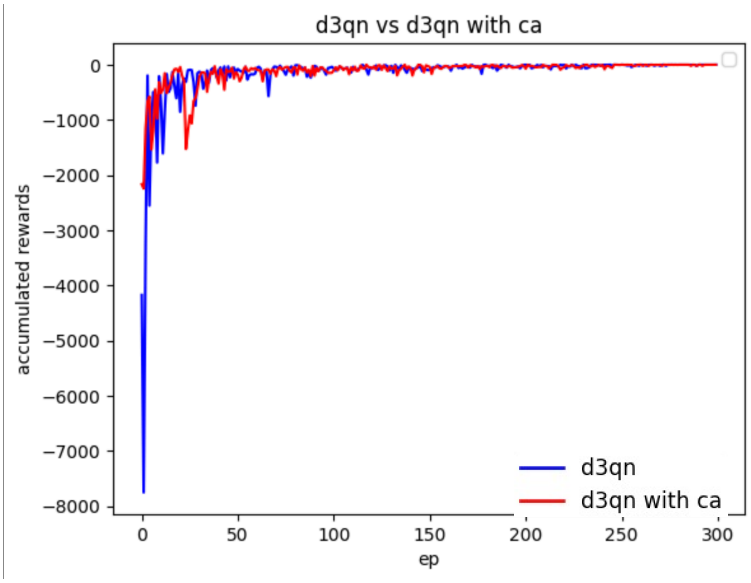


Fig 8. D3QN vs D3QN-CA

Observation :

From the above images after running these models on a 2 obstacle environment we can clearly see that D3QN is performing the best compared to other models. Also the implementation of channel attention has significantly decreased the fluctuations after 70 episodes in Fig 6 and Fig 8.

Comparison of Steps Taken :

without channel attention vs with CA in a custom 2 Obstacle Environment.

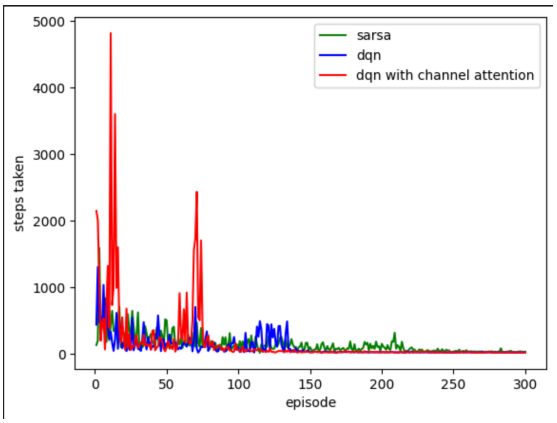


Fig 9. Sarsa Vs DQN Vs DQN-CA

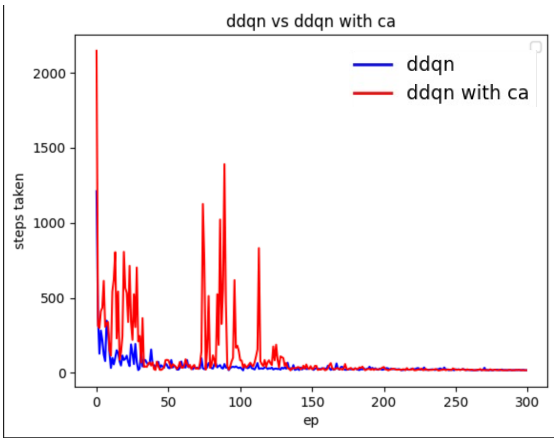


Fig 10. DDQN Vs DDQN-CA

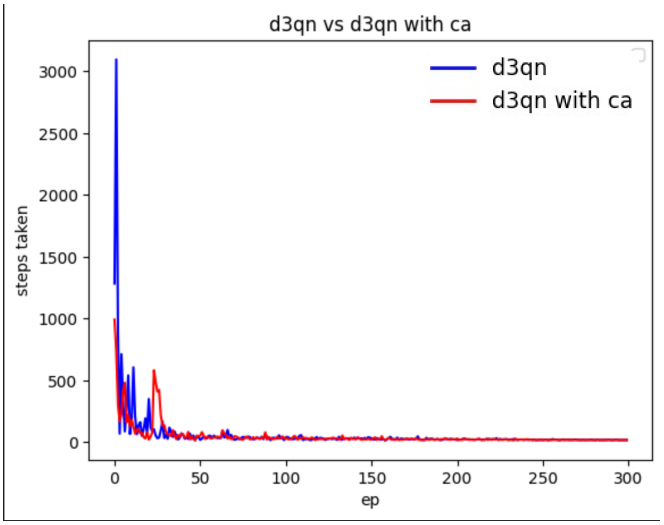


Fig 11. D3QN Vs D3QN-CA

Observation :

From the above images after running these models on a 2 obstacle environment we can clearly see that DDQN is taking the least number of steps compared to other models. But since the fluctuations after implementing channel attention is more in DDQN case it is advisable that D3QN suits better for this case. Implementing Channel attention on DQN also decreased the fluctuations after 75 episodes in all the Figures 9,10,11.

Comparison of Paths Taken :

without channel attention vs with CA in a custom 2 Obstacle Environment.

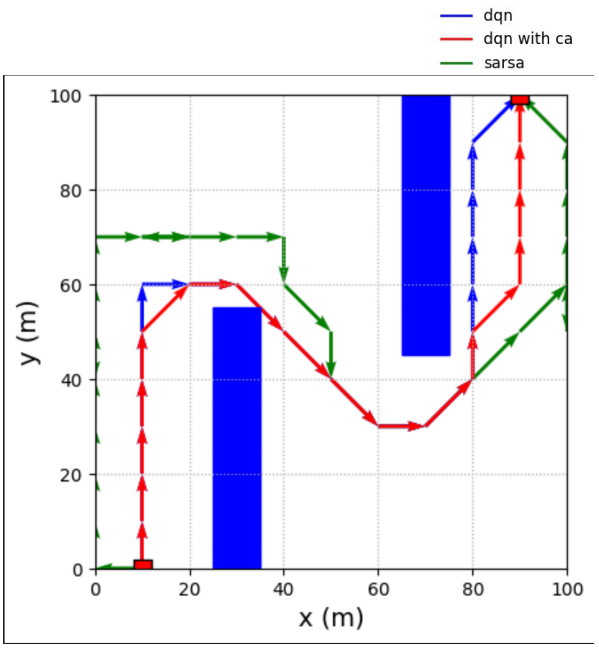


Fig 12.Sarsa Vs DQN Vs DQN-CA.

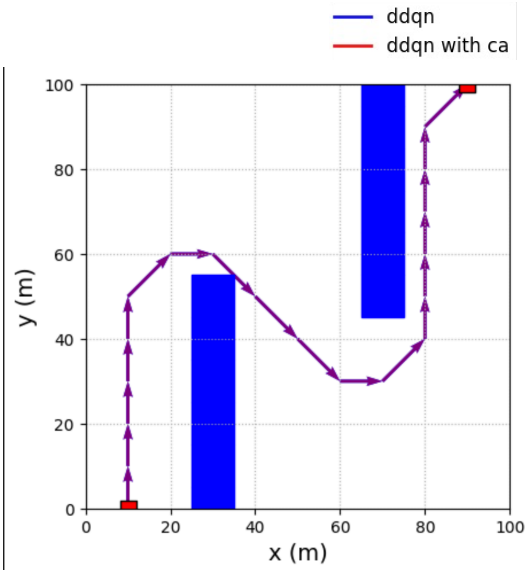


Fig13. DDQN Vs DDQN-CA

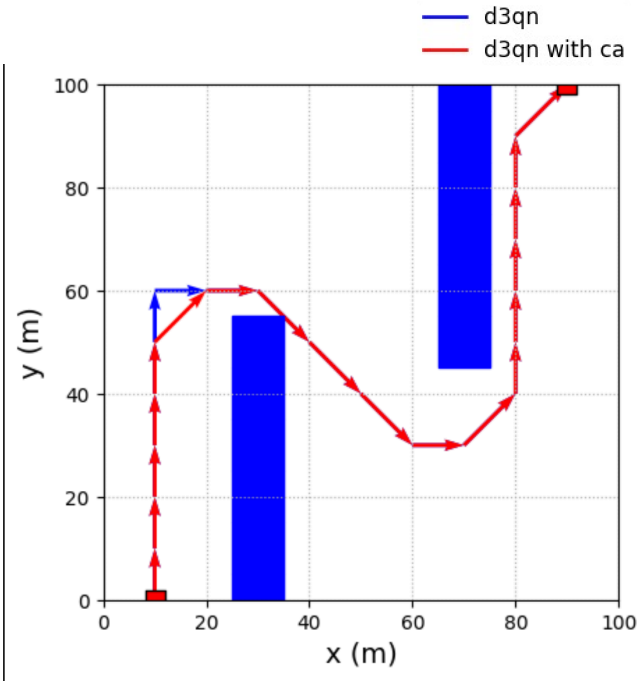


Fig 14. D3QN Vs D3QN-CA

Observation :

We can see that the final path taken by SARSA is random compared to the DQN and DQN-CA. The overall path is better after enabling channel attention for each of DQN,DDQN,D3QN. Since it is a simple environment there is no visible difference between DDQN and D3QN.

Comparison of accumulated rewards :

without channel attention vs with CA in a custom 3 Obstacle Environment.

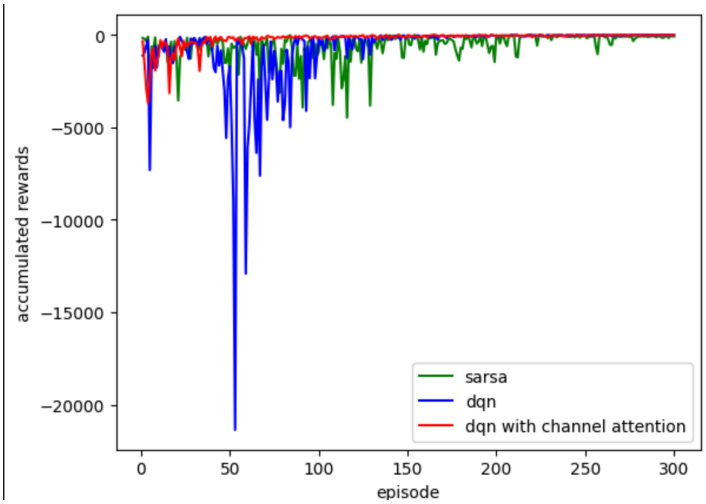


Fig 15. Sarsa Vs DQN Vs DQN-CA.

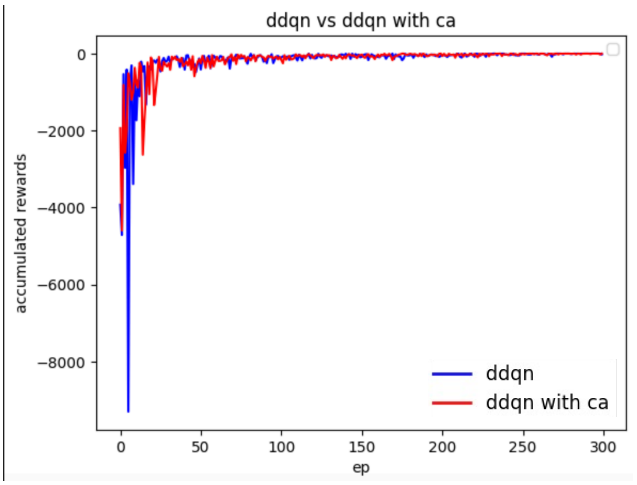


Fig 16. DDQN Vs DDQN-CA

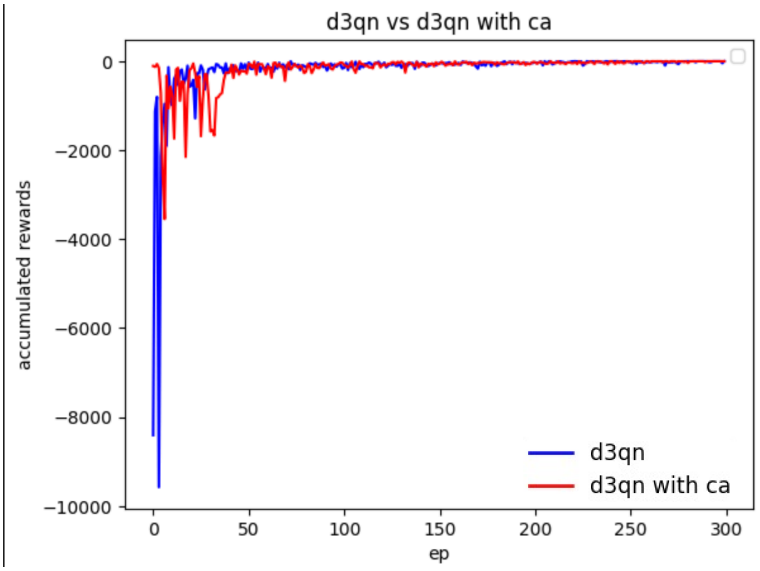


Fig 17. D3QN Vs D3QN-CA.

Observation :

From the above images after running these models on a 3 obstacle environment we can clearly see that both DDQN and D3QN are performing better than other models. Also the implementation of channel attention has significantly decreased the fluctuations after 40 episodes in Fig 16 and Fig 17 essentially allowing it to converge at a higher rate. Also, enabling channel attention decreased the minima significantly.

.

Comparison of Steps Taken :

without channel attention vs with CA in a custom 3 Obstacle Environment.

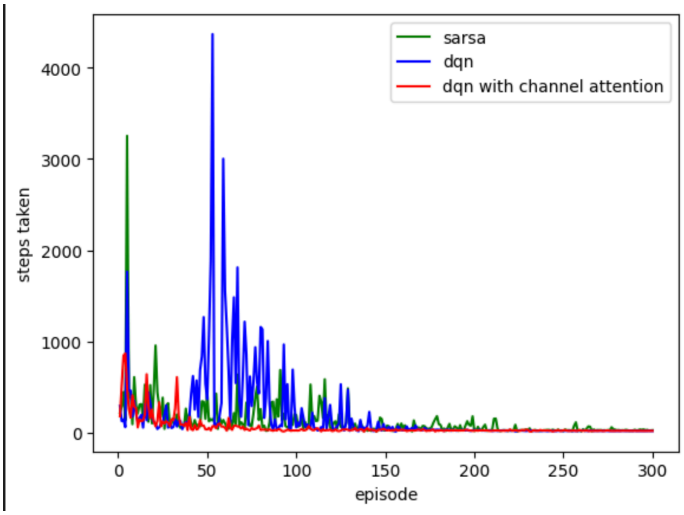


Fig 18. Sarsa Vs DQN Vs DQN-CA.

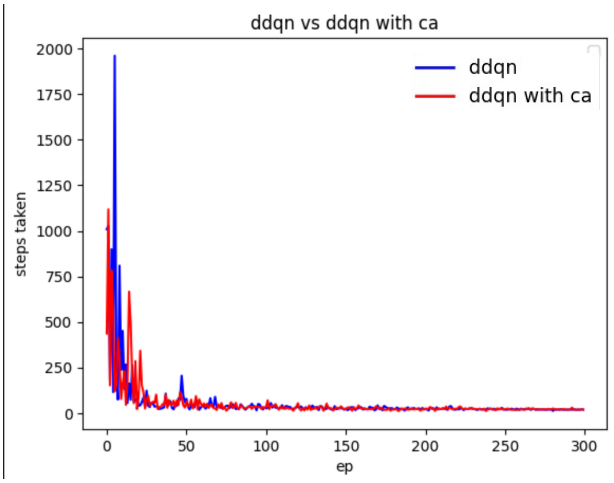


Fig 19. DDQN Vs DDQN-CA

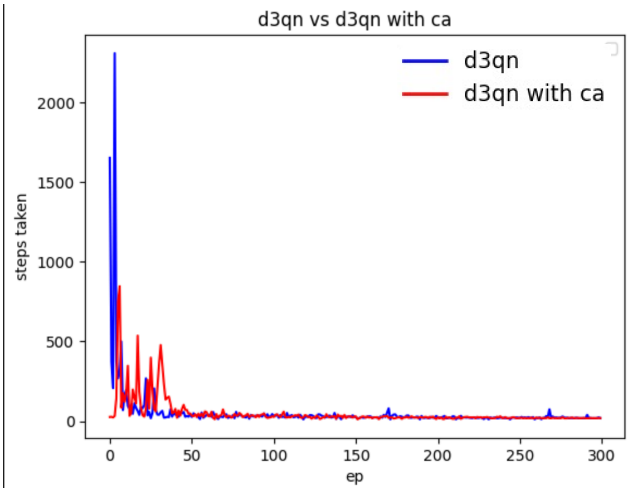


Fig 20. D3QN Vs D3QN-CA.

Observation :

From the above images after running these models on a 3 obstacle environment we can clearly see that both DDQN and D3QN are taking least number of steps compared to other models. Also the implementation of channel attention has significantly decreased the fluctuations after 40 episodes in Fig 18, Fig 19 and Fig 20. Also, enabling channel attention decreased the maximum number of steps taken significantly.

Comparison of Paths Taken :

without channel attention vs with CA in a custom 2 Obstacle Environment.

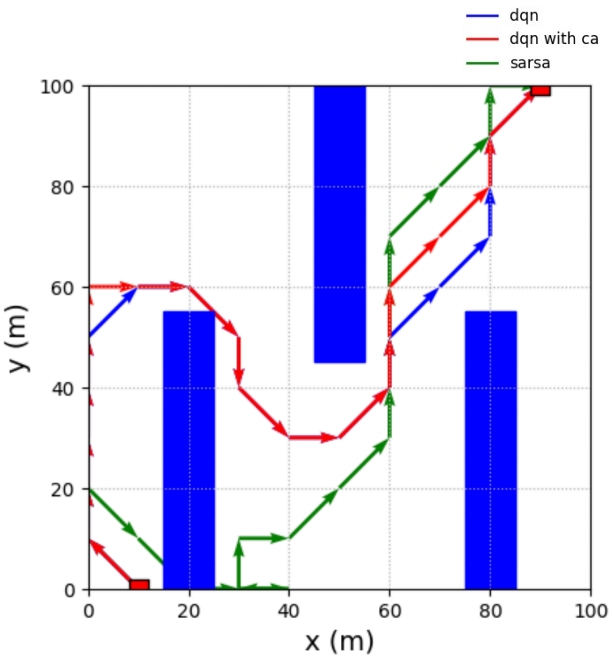


Fig 21. Sarsa Vs DQN Vs DQN-CA.

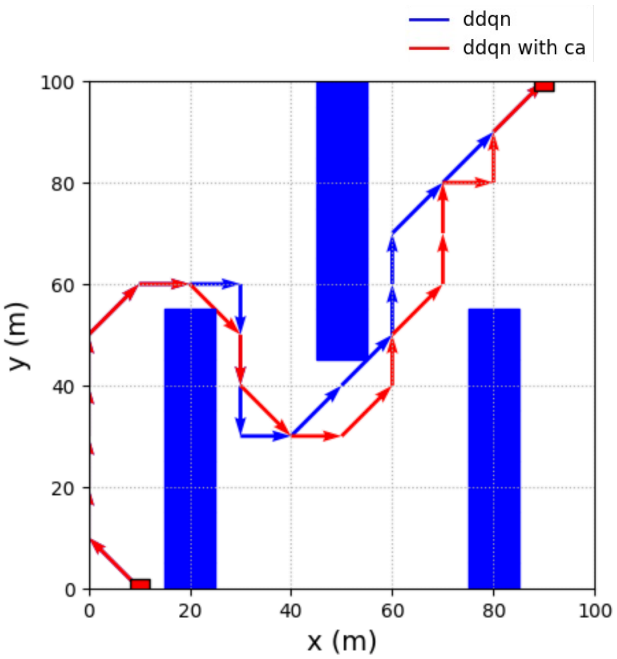


Fig 22. DDQN Vs DDQN-CA

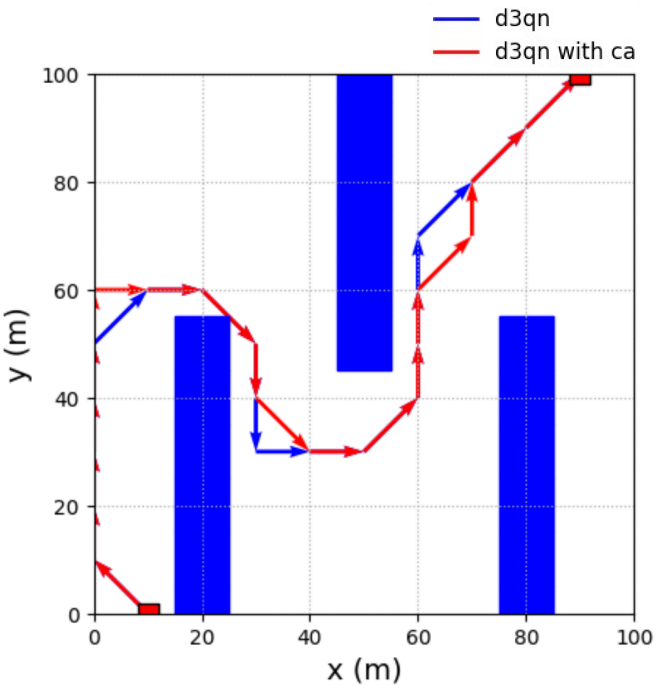


Fig 23. D3QN Vs D3QN-CA

Observation :

We can see that the final path taken by SARSA is random compared to the DQN and DQN-CA. The overall path is better after enabling channel attention for each of DQN,DDQN,D3QN. Also D3QN-CA minimised the distance travelled by the robot by selecting on the important features.

Comparison of accumulated rewards :

without channel attention vs with CA in a custom Complex Environment.

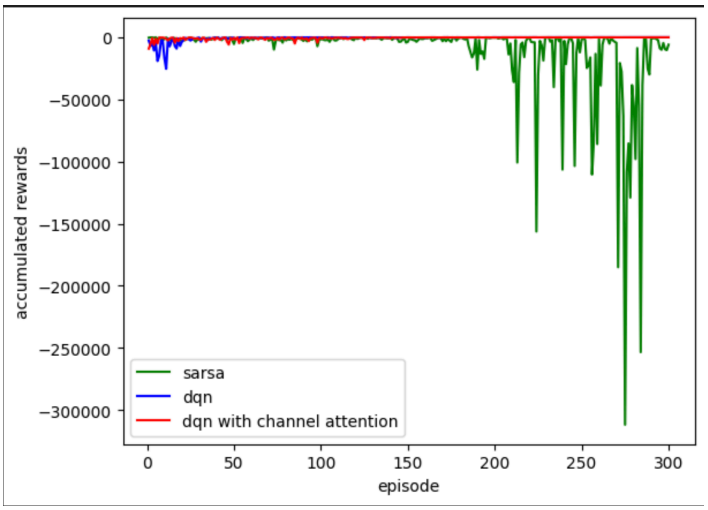


Fig 24. Sarsa Vs DQN Vs DQN-CA.

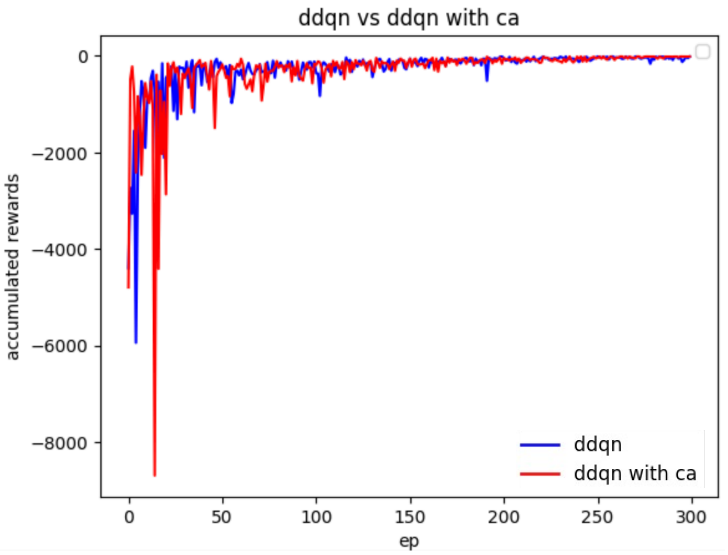


Fig 25. DDQN Vs DDQN-CA

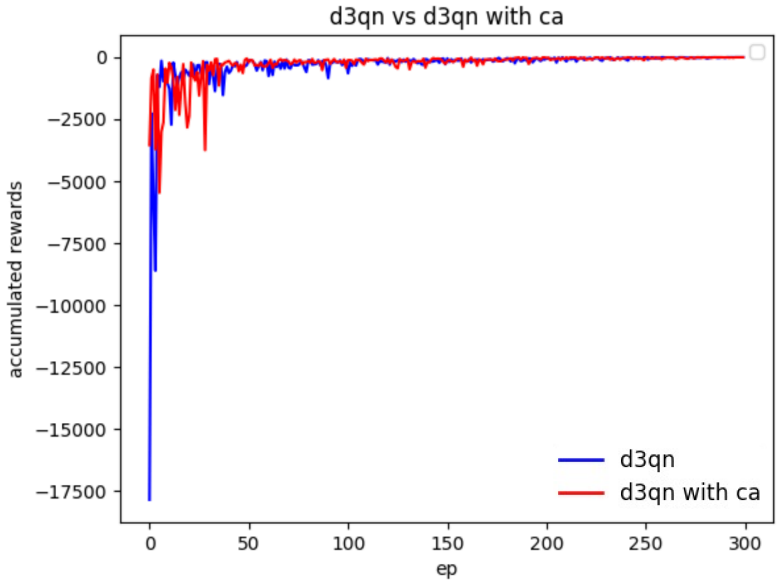


Fig 26. D3QN Vs D3QN-CA

Observation :

From the above images after running these models on a Complex environment we can clearly see that DDQN(smallest minima) is performing the best compared to other models. DDQN also has more fluctuations compared to D3QN. The deduction is that D3QN is better in this particular environment. SARSA also shows abnormal rewards and is not suitable for complicated environments.

.

Comparison of Steps Taken :

without channel attention vs with CA in a custom Complex Environment.

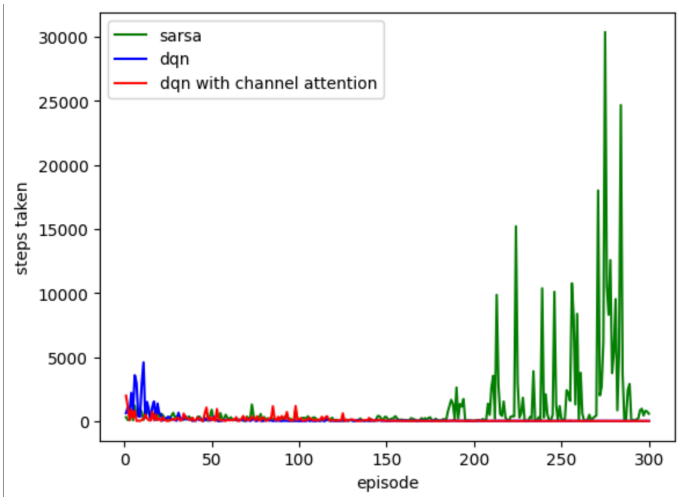


Fig 27. Sarsa Vs DQN Vs DQN-CA.

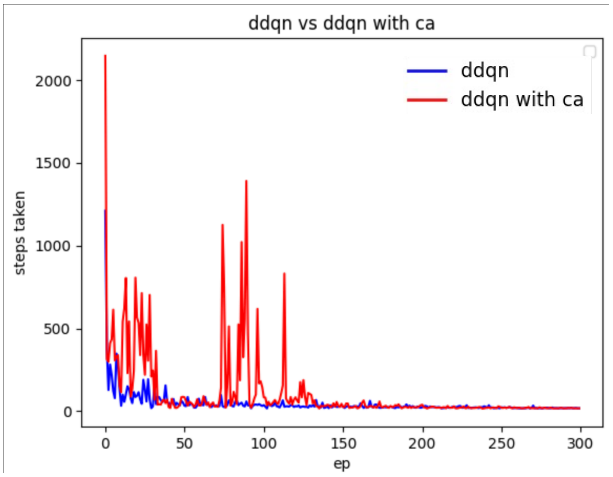


Fig 28. DDQN Vs DDQN-CA

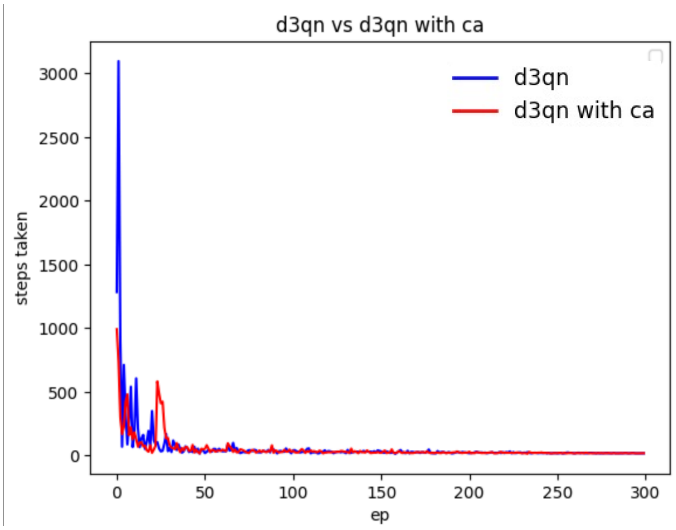


Fig 29. D3QN Vs D3QN-CA.

Observation :

From the above images after running these models on a complex environment we can clearly see that both DDQN and D3QN are taking least number of steps compared to other models. Also the implementation of channel attention has significantly decreased the fluctuations after 25 episodes in Fig 29. Also, enabling channel attention decreased the maximum number of steps taken significantly in the case of D3QN.

Comparison of Paths Taken :

without channel attention vs with CA in a custom Complex Environment.

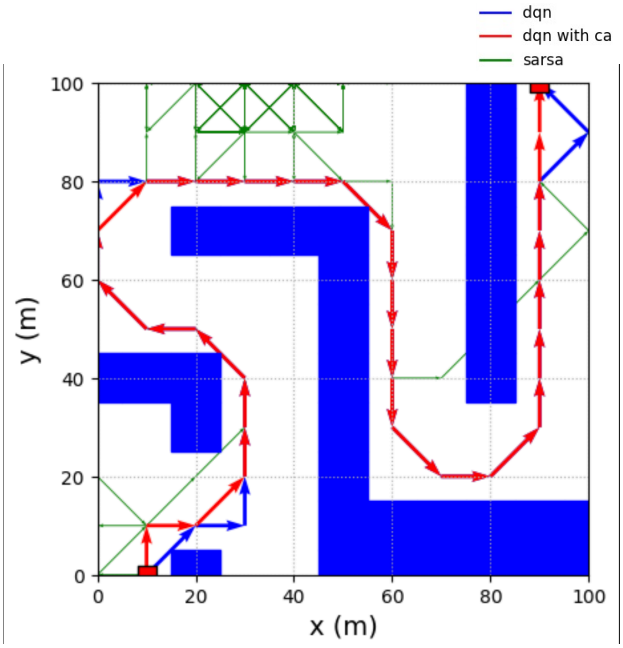


Fig 30. Sarsa Vs DQN Vs DQN-CA.

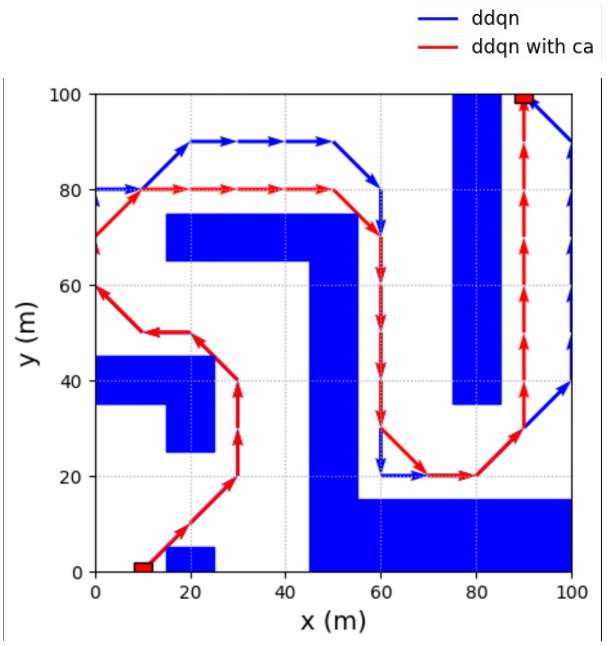


Fig 31. DDQN Vs DDQN-CA

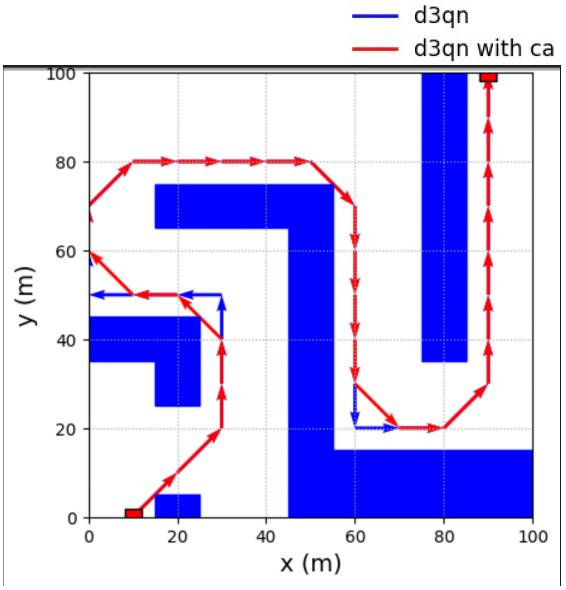


Fig 32. D3QN Vs D3QN-CA.

Observation :

We can see that the final path taken by SARSA is random and going through obstacles compared to the DQN and DQN-CA. The overall path is better after enabling channel attention for each of DQN,DDQN,D3QN. Also D3QN-CA minimised the distance travelled by the robot by selecting on the important features whereas DDQN took some extra steps to reach the goal.

Chapter 5
CONCLUSION AND FUTURE SCOPE

The results from evaluating DQN, DQN with channel attention, SARSA, DDQN, D3QN, D3QN with channel attention, and DDQN with channel attention (D3QN-CA) reveal significant insights into the effects of both Double Q-learning and channel attention mechanisms on performance. The addition of channel attention consistently improves model efficiency and reward optimisation, especially when combined with advanced Q-learning architectures like DDQN and D3QN. Key findings include:

Path Efficiency: Models enhanced with channel attention, particularly D3QN-CA and DDQN with CA, demonstrate improved path efficiency. By focusing on relevant features, channel attention allows these models to take more direct routes to the goal, reducing the unnecessary exploration.

Higher Reward Maximisation: Channel attention also contributes to improved cumulative rewards. Models like D3QN-CA and DDQN with CA prioritise high-reward actions more effectively than their counterparts without attention, leading to consistently higher total rewards .

Enhanced Stability and Convergence: The use of DDQN in DDQN with CA and D3QN-CA mitigates overestimation bias, resulting in more stable convergence. The channel attention mechanism further accelerates this process by highlighting essential features early in training, allowing these models to converge faster and stabilise reward levels.

Efficient Exploration: The models enabled with channel attention reduce the exploration burden by filtering out less relevant information, which allows DQN with CA, D3QN with CA, and DDQN with CA .

Improved Feature Prioritisation: Channel attention enhances feature prioritisation in all models where it is applied. By concentrating on critical state-action information, models like D3QN with CA and DDQN with CA navigate complex environments more effectively and avoid distractions from irrelevant features.

In summary, this study demonstrates that adding channel attention to models like DDQN and D3QN substantially enhances their performance, making them particularly effective for tasks requiring high reward accumulation and efficient path planning. The results validate channel attention as a powerful addition to Q-learning-based RL models, especially in challenging environments.

.

Future Scope

Expanding this project into a 3D environment opens opportunities to test DQN, DQN with channel attention, SARSA, DDQN, D3QN, D3QN with channel attention, and DDQN with channel attention in more realistic, dynamic settings. A 3D environment introduces greater state-action complexity, spatial awareness, and depth perception requirements, allowing for:

Advanced Model Training: Evaluating Each model's ability to adapt to a 3D environment navigating through obstacles, highlighting strengths in reward maximisation and stability. SARSA's on-policy nature and models like D3QN-CA's attention mechanisms can be tested for effectiveness in handling complex 3D tasks.

Channel Attention Optimisation: Channel attention in DQN, DDQN, and D3QN can be refined to prioritise spatially relevant features, enhancing decision-making efficiency in 3D environments by using continuous camera feed. The refining of attention feed is expected to further improve path efficiency and maximise reward for models like D3QN-CA and DDQN with CA.

Additional Attention Mechanisms: Exploring spatial and goal-based attention, such as using visual cues from a camera feed, could enhance situational awareness. Models like DDQN and D3QN could benefit from these mechanisms, allowing more focused and goal-oriented decision-making.

Real-World Applications: Moving to 3D environments brings these models closer to real-world applications in robotics, virtual reality, and automated navigation, where effective decision-making in multi-dimensional spaces is critical.

Testing on open environments : Testing the developed model on complicated environments where actions of the robot are restricted and require more caution to avoid collision with obstacles. Modifying the tolerance of the robot to avoid collision with wall.

This transition will help assess each model's scalability and prepare them for practical use in high-dimensional environments.

.

References

1) Yu Wu, Niansheng Chen, Guangyu Fan, Dingyu Yang, Lei Rao, Songlin Cheng, Xiaoyong Song, Yiping Ma (2024). NAVS: A Neural Attention-Based Visual SLAM for Autonomous Navigation in Unknown 3D Environments. Journal of Neural Processing letters. https://link.springer.com/article/10.1007/s11063-024-11502-6

2) Guo M-H, Xu T-X, Liu J-J, Liu Z-N, Jiang P-T, Mu T-J, Zhang S-H, Martin RR, Cheng M-M, Hu S-M (2022). Attention mechanisms in computer vision: a survey. Journal of Computational Visual Media. https://link.springer.com/article/10.1007/s41095-022-0271-y

3) Yin-Hao Wang, Tzuu-Hseng S. Li, Chih-Jui Lin, Backward Q-learning: The combination of Sarsa algorithm and Q-learning,Engineering Applications of Artificial Intelligence,Volume 26, Issue 9,2013,Pages 2184-2193,ISSN 0952-1976, https://doi.org/10.1016/j.engappai.2013.06.016

4) Karaman, S., & Frazzoli, E. (2011). Sampling-based algorithms for optimal motion planning. The International Journal of Robotics Research, 30(7), 846-894. https://doi.org/10.1177/0278364911406761

5) Singh, A., Sridhar, Y., Kalaichelvi, V., & Karthikeyan, R. (2024). Performance Evaluation of Vision-Based Path Planning for Dynamic Real-Time Scenarios of Mobile Robot. Springer.https://link.springer.com/article/10.1007/s11042-024-19267-9

6) Pfeiffer, M., Schaeuble, M., Nieto, J., Siegwart, R., & Cadena, C. (2017). From perception to decision: A data-driven approach to end-to-end motion planning for autonomous ground robots. 2017 IEEE International Conference on Robotics and Automation (ICRA), 1527-1533. Available at: https://doi.org/10.1109/ICRA.2017.7989182

7) Prasuna, R. G., & Potturu, S. R. (2024). Deep Reinforcement Learning in Mobile Robotics – A Concise Review. Springer. https://link.springer.com/article/10.1007/s11042-024-18152-9

8) Tai, L., Paolo, G., & Liu, M. (2017). Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation. 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 31-36. Available at: https://doi.org/10.1109/IROS.2017.8202134

.

9) Xie, L., Wang, S., Markham, A., & Trigoni, N. (2018). Towards monocular vision based obstacle avoidance through deep reinforcement learning. arXiv preprint arXiv:1706.09829. Available at: https://arxiv.org/abs/1706.09829

10) Martinez, J., Gao, Y., & Smith, R. (2022). Distance and Heading Integration for Improved Navigation Efficiency. Processes, 10(12), 2748. Available at: https://www.mdpi.com/2227-9717/10/12/2748

11) Reference List : Sewak. M. (2019).
Deep Q Network (DQN), Double DQN, and Dueling DQN. In: Deep Reinforcement Learning. Springer, Singapore.
https://doi.org/10.1007/978-981-13-8285-7_8
In-text : (Sewak.M, 2019)

12) Mehmet Gök, Dynamic path planning via Dueling Double Deep Q-Network (D3QN) with prioritised experience replay, Journal of Applied Soft Computing, Volume 158, 2024, 111503, ISSN 1568-4946, https://doi.org/10.1016/j.asoc.2024.111503
https://www.sciencedirect.com/science/article/pii/S1568494624002771

13) Haosen Qin, Tao Meng, Kan Chen, Zhengwei Li, A comparative study of DQN and D3QN for HVAC system optimisation control, Journal of Energy, Volume 307, 2024, 132740, ISSN 0360-5442, https://doi.org/10.1016/j.energy.2024.132740
https://www.sciencedirect.com/science/article/pii/S0360544224025143

14) Bernardo Avila Pires, Feryal Behbahani, Hubert Soyer, Kyriacos Nikiforou, Thomas Keck, Satinder Singh (2023). Hierarchical Reinforcement Learning in Complex 3D Environments. https://doi.org/10.48550/arXiv.2302.14451

15) Lennart Bramlage, Aurelio Cortese, Generalised attention-weighted reinforcement learning, Journal of Neural Networks, Volume 145, 2022, Pages 10-21, ISSN 0893-6080,
https://doi.org/10.1016/j.neunet.2021.09.023
https://www.sciencedirect.com/science/article/pii/S0893608021003853

.

16) Zhou, Z., Zhu, P., Zeng, Z. et al. Robot navigation in a crowd by integrating deep reinforcement learning and online planning. Appl Intell 52, 15600–15616 (2022). https://doi.org/10.1007/s10489-022-03191-2

17) Mao, H., Zhang, Z., Xiao, Z. et al. Learning multi-agent communication with double attentional deep reinforcement learning. Auton Agent Multi-Agent Syst 34, 32 (2020). https://doi.org/10.1007/s10458-020-09455-w

.