# Advancing Cooperative Goal Achievement in Multi-Agent Systems with Deep Q-Network Variants

Bharath Muppasani
Project Report Update - Oct 31, 2024

November 1, 2024

**Abstract**

This project focuses on advancing cooperative goal achievement in multi-agent systems by implementing Deep $Q$-Network (DQN) algorithms with centralized planning and coordination mechanisms. By adapting DQN variants such as Double DQN, Prioritized Replay, and Dueling Architecture, the study addresses the complexities of multi-agent interaction in environments requiring both goal-reaching behavior and conflict avoidance. This approach aims to optimize planning and decision-making across agents, providing insights into scalable coordination strategies that enhance policy convergence and effectiveness in collaborative AI settings.

## 1 Introduction and Methodology

In multi-agent systems, achieving coordinated goal-oriented behavior presents unique challenges, particularly when agents must balance goal-reaching tasks with conflict avoidance. Traditional single-agent reinforcement learning (RL) frameworks are typically designed to optimize an individual agent's policy, often neglecting the interdependencies and competitive dynamics that arise in multi-agent contexts. To address these issues, this project implements Deep

Q-Network (DQN) algorithms with enhancements for centralized planning and coordination mechanisms for multi-agent reinforcement learning.

## 1.1 Problem Definition

In our setup, we consider $N$ agents in a shared environment, each tasked with collectively covering $N$ landmarks. Agents receive rewards based on their proximity to the nearest landmarks and are penalized upon colliding with other agents, introducing a collaborative goal to maximize landmark coverage while minimizing conflict. Formally, the reward for each agent $i$ is defined as:

$$r_i = -\alpha \sum_{j=1}^{N} d_{i,j} - \beta C_i, \tag{1}$$

where $d_{i,j}$ represents the Euclidean distance between agent $i$ and landmark $j$, $\alpha$ is a scaling factor for distance-based rewards, and $C_i$ is a collision indicator for agent $i$, weighted by penalty factor $\beta$. This setup requires agents to develop policies that efficiently balance movement towards landmarks and collision avoidance.

## 1.2 Methodology

To enable effective coordination, we adopt centralized planning using Deep Q-Networks, augmented by three major modifications: Double DQN, Prioritized Experience Replay, and Dueling Network Architecture. These approaches are incorporated to mitigate overestimation bias, prioritize critical experiences, and enhance decision value separation, respectively. We describe each enhancement below.

### 1.2.1 Double DQN

The Double DQN [1] algorithm aims to reduce overestimation bias in Q-learning by decoupling the action selection and evaluation steps. The target update for Double DQN is given by:

$$y_i = r + \gamma Q_{\theta^-}(s', \arg\max_a Q_\theta(s', a)), \tag{2}$$

where $s'$ represents the next state, $\gamma$ is the discount factor, and $\theta$ and $\theta^-$ are the policy and target network parameters, respectively. This separation helps to achieve more stable policy convergence in complex multi-agent environments.

### 1.2.2 Prioritized Experience Replay

Prioritized Experience Replay [2] is employed to allocate more memory capacity to experiences with higher temporal-difference (TD) errors, allowing the model to learn effectively from more informative states. The TD error, $\delta$, measures the difference between the predicted Q-value and the target value, guiding the agent on how much it needs to update its policy based on each experience. For an experience tuple $(s, a, r, s')$, the TD error $\delta_i$ for experience $i$ can be defined as:

$$\delta_i = \left| r + \gamma \max_{a'} Q(s', a'; \theta^-) - Q(s, a; \theta) \right|, \tag{3}$$

where $r$ is the reward, $\gamma$ is the discount factor, $s'$ is the next state, and $a'$ is the action that maximizes the target Q-value with parameters $\theta^-$. The probability of sampling experience $i$ with TD error $\delta_i$ is:

$$P(i) = \frac{\delta_i^\omega}{\sum_k \delta_k^\omega}, \tag{4}$$

where $\omega$ controls the prioritization intensity, ensuring that the agent focuses on critical transitions in its experience buffer.

### 1.2.3 Dueling Network Architecture

The Dueling Network Architecture [3] decomposes Q-values into separate state-value and advantage components, improving the learning rate for policy optimization in scenarios where action relevance varies significantly. The Q-value approximation is given by:

$$Q(s, a; \theta) = V(s; \theta) + \left( A(s, a; \theta) - \frac{1}{|A|} \sum_{a'} A(s, a'; \theta) \right), \tag{5}$$

where $V(s; \theta)$ represents the state-value function and $A(s, a; \theta)$ is the advantage function. This architecture allows the agents to better differentiate between critical and non-critical actions.

### 1.3 Coordination Mechanism

We implement a centralized planner that aggregates information from all agents and optimizes the joint policy across agents. By aligning each agent's action with the global objective, the centralized planner ensures that agents achieve their goals cooperatively, effectively navigating the balance between landmark coverage and collision avoidance. Experimental results on this multi-agent framework will validate the impact of each enhancement in achieving robust multi-agent coordination and goal-oriented planning.

## 2 Initial Implementation

In the initial phase of this project, DQN algorithms, including DQN, Double DQN, and Prioritized Experience Replay, were implemented using the CartPole environment from OpenAI Gym. This environment serves as a standard benchmark for reinforcement learning due to its simplicity and well-defined goal, providing a controlled setup to validate the core functionalities of each algorithm variant.

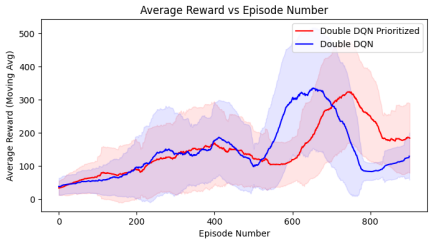
### 2.1 Implementation Details

The DQN algorithm was implemented as a baseline, followed by enhancements with Double DQN and Prioritized Replay.

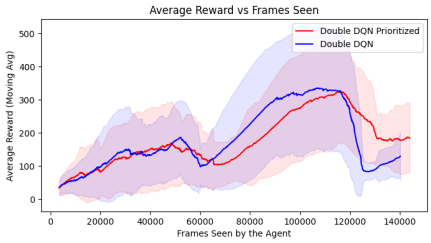
### 2.2 Results on CartPole

Figure 1a and Figure 1b illustrate the performance comparison between Double DQN and Double DQN with Prioritized Replay in terms of average reward per episode and frames seen by the agent, respectively. The results suggest that prioritized replay accelerates learning, with Double DQN Prioritized showing a more rapid increase in average reward over episodes and frames seen.

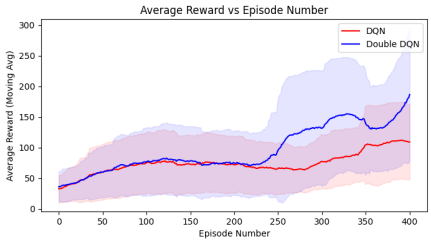
Similarly, Figure 1c and Figure 1d show the comparison between DQN and Double DQN. Double DQN consistently outperforms standard DQN, achieving higher average rewards over time, indicating that it effectively addresses overestimation issues present in traditional DQN.



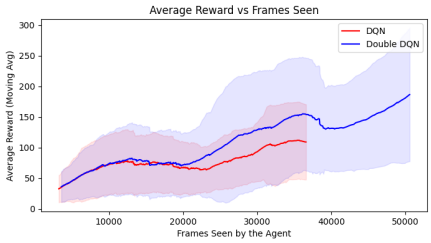

(a) Average Reward vs Episode Number for Double DQN and Double DQN with Prioritized Replay.




(b) Average Reward vs Frames Seen for Double DQN and Double DQN with Prioritized Replay.




(c) Average Reward vs Episode Number for DQN and Double DQN.




(d) Average Reward vs Frames Seen for DQN and Double DQN.

Figure 1: Comparison of Average Rewards for DQN, Double DQN, and Double DQN with Prioritized Replay across Episodes and Frames Seen.

### 2.3 Next Steps

The next phase of this project will involve adapting these algorithms to a multi-agent environment. This transition will require modifications to accommodate multi-agent dynamics, including a centralized planning mechanism and coordination strategies to handle inter-agent interactions and shared goal structures. These changes will allow for a more complex testing environment to evaluate the efficacy of each algorithm variant in a multi-agent setup.

## References

[1] H. van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[2] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," in *International Conference on Learning Representations (ICLR)*, 2016.

[3] Z. Wang, T. Schaul, M. Hessel, H. van Hasselt, M. Lanctot, and N. de Freitas, "Dueling Network Architectures for Deep Reinforcement Learning," in *Proceedings of the 33rd International Conference on Machine Learning*, 2016.