CS258/EE227-Tejas Machkar Report

Intro to RL

### Assigned: October 16, 2025

### Due: October, 28, 2025

### Project 2 – Deep Q Learning

## Codebase Instructions:

* For question 2 and 3 we can set the parameter MODE to ‘plot\_and\_video’ (Line 436 for Q2, Line 441 for Q3) to evaluate the best trained models.
* The plot would be generated using the best model in figures/ folder and the video of the example run in videos/ folder.
* The best model for submission are located in q<question\_number>\_models/Best\_Model\_For\_Submission/

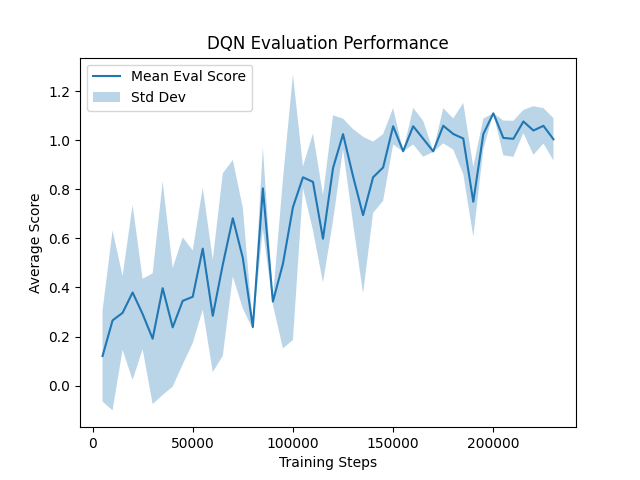
## Question 1 : DQN

ANS:

### Q-Network Architecture

* The Q-network itself is a Multi-Layer Perceptron (MLP) with the following architecture:
* Input Layer: 61 nodes (corresponding to the 60 LiDAR readings and 1 distance-to-goal value)
* Hidden Layer 1: 64 nodes with ReLU activation
* Hidden Layer 2: 64 nodes with ReLU activation
* Output Layer: 3 nodes (linear activation), representing the Q-value for each of the 3 possible actions (stand still, move up, move down).

### Evaluation performance of model during training



This plot shows the mean evaluation score (solid line) and standard deviation (shaded area) of the DQN agent, averaged over three random seeds. The x-axis represents the total number of environment interactions (training steps), while the y-axis shows the average reward obtained during a greedy 10-episode evaluation. The clear upward trend, rising from an initial score near 0.1 to a stable plateau around 1.0, demonstrates that the agent successfully learned a robust policy to solve the frogger-v0 environment by consistently reaching the goal.

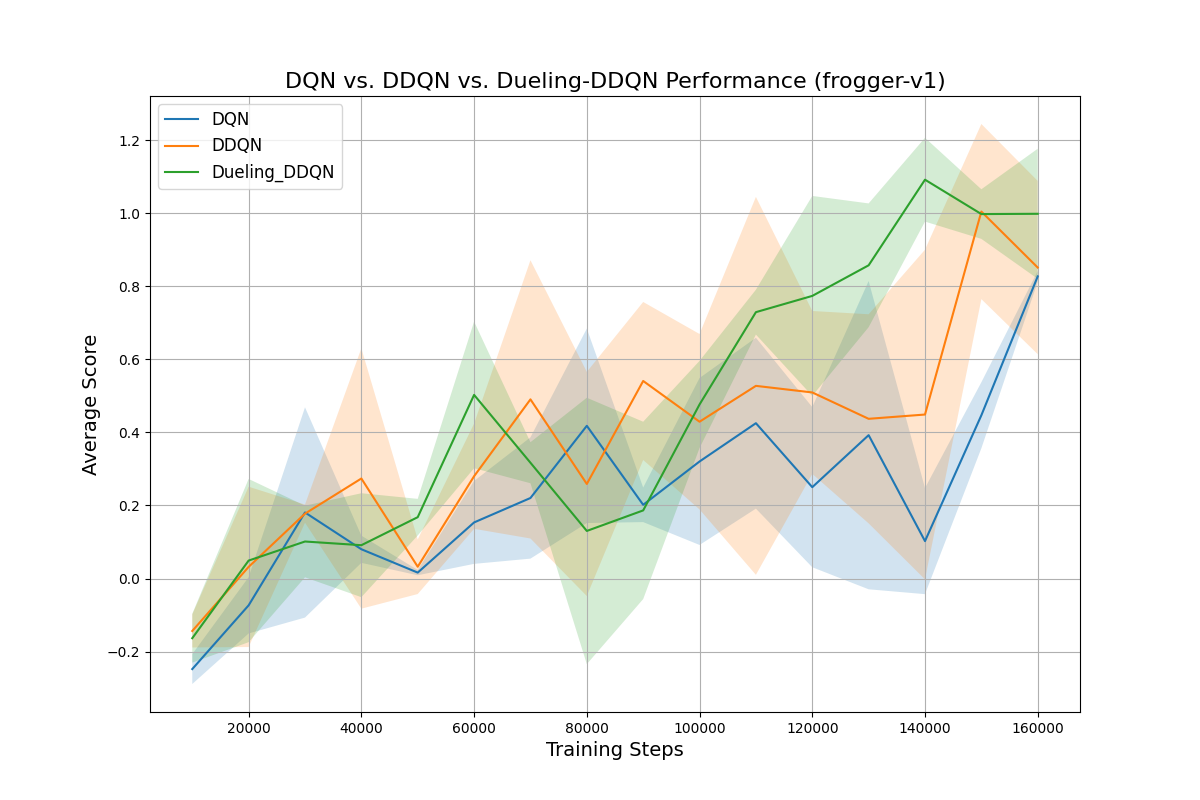
### Evaluation Report

* The best model performance was achieved for seed 0.
* Final agent achieved an average score of: 1.11

## Question 2 : DQN vs DDQN vs Duelling DDQN

ANS:

### Performance comparison of DQN vs DDQN vs Dueling DDQN



This plot compares the mean evaluation performance of the DQN, DDQN, and Dueling-DDQN agents on the more complex frogger-v1 environment. While all three models learn, the Dueling-DDQN agent (green) clearly outperforms the others, showing a more stable and consistent learning curve that converges to the highest average score. This result demonstrates the benefit of both the DDQN algorithm (which outperforms standard DQN) and the dueling architecture, which provides an additional significant boost in performance and stability.

### Evaluation Report

* From the plot it is clear that the best performing model we get is the Dueling DDQN, which we get from setting the seed to 42.
* Final best agent achieved an average score of: 1.24

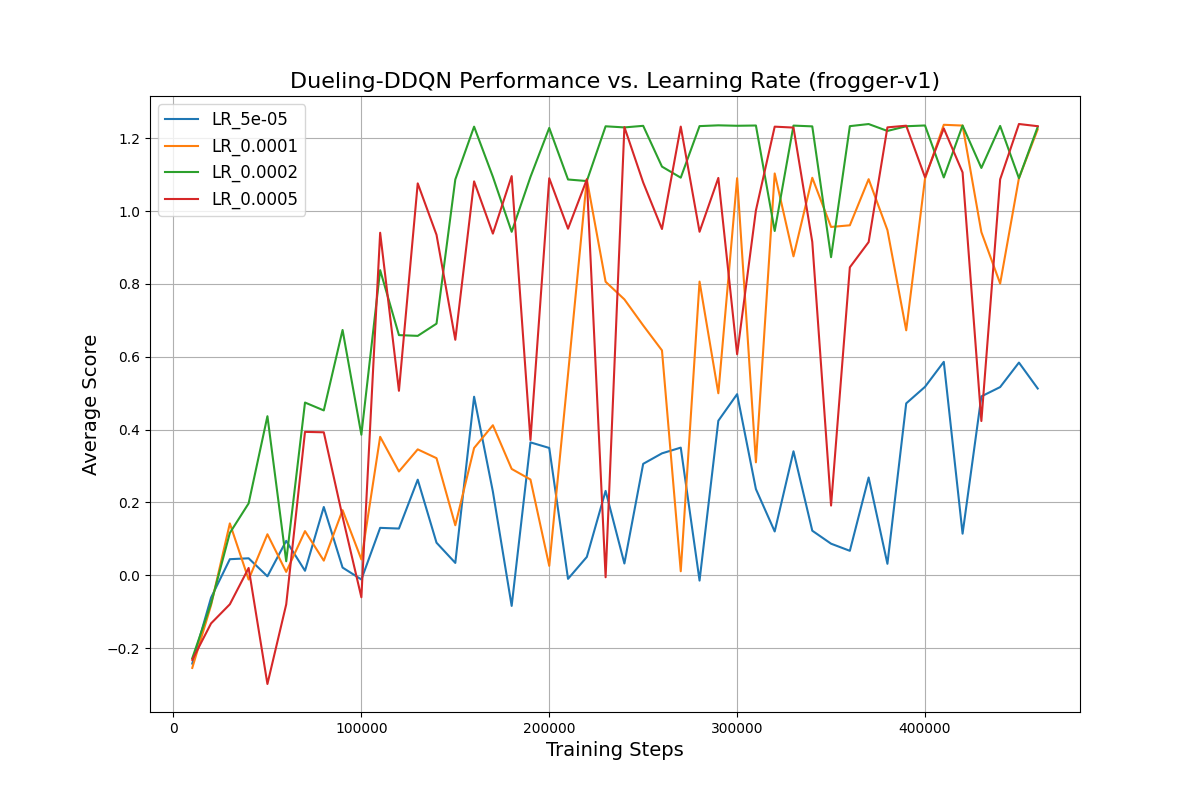
## Question 3: Hyperparameter Tuning

ANS:

### Choice of Hyperparameter:

For Question 3, we chose to tune the Learning Rate (LR) on our best model, the Dueling-DDQN. This hyperparameter was selected because it is one of the most critical factors influencing performance, directly controlling the size of the network's weight updates and managing the "non-trivial" trade-off between convergence speed and training stability. A rate that is too high can cause the model to diverge, while a rate that is too low will cause it to learn too slowly, making it the perfect candidate for tuning.

### Performance Evaluation of Model for different Learning Rates:



This plot clearly illustrates the critical impact of the learning rate on the agent's performance. The LR\_5e-05 (blue) is too low, causing the agent to learn very slowly and never reach the performance of the other models. Conversely, the LR\_0.0005 (red) is too high, leading to unstable training with erratic spikes and crashes as the agent overcorrects its policy. The LR\_0.0001 (orange) and LR\_0.0002 (green) both strike a good balance, but the LR\_0.0002 demonstrates the most ideal performance, achieving the fastest, most consistent convergence to the highest average score.

### Evaluation Report:

* We get the performance for Dueling DDQN model trained with LR 0.0002, and seed of 42.
* Final best agent achieved an average score of: 1.24.