Cairo University

Faculty of Engineering

Computer Engineering Department

CMPS458 Reinforcement Learning - Assignment 2

Team Name/Number: Team 1
First member name: Mariam Mahrous 1210301
Second member name: Menna Salah 1210032

Third member name: Farida Ahmed 1210276

Supervisor: Ayman AboElhassan

November 13, 2025

Deliverables

Repo link: https://github.com/Mennasalah140/Reinforcement-Learning

Video record link:
https://drive.google.com/drive/folders/1xS7qhv08kY4WoeNomgsLMmIuZCMSeY8V?usp=sharing

Discussion

0.1 Question Answers

Q1 — Difference Between DQN and DDQN in Terms of Training Time and Performance

Deep Q-Network (DQN) and Double Deep Q-Network (DDQN) are both value-based methods that use neural networks to approximate the Q-function. The main distinction lies in how each method estimates the target Q-value during learning.

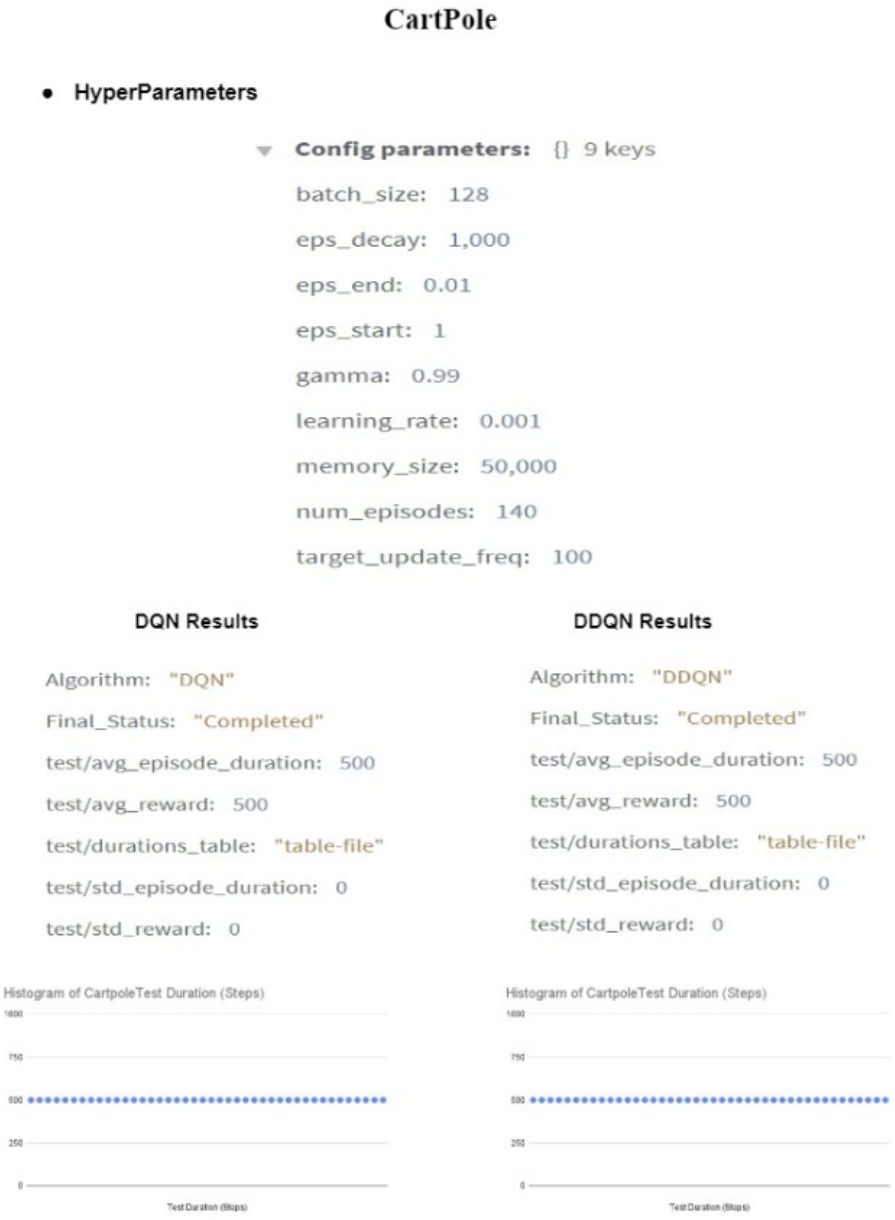
Table 1: Comparison between DQN and DDQN

Aspect	**DQN**	**DDQN**
Training Time	Slightly faster due to fewer updates	Slightly slower, but more consistent performance
Performance	Can reach high rewards quickly but unstable	Achieves higher long-term rewards and generalization

Q2 — How stable are the trained agents? Show with test episode duration figures.

CartPole-v1

Figure 1: CartPole-v1



Acrobot-v1

Figure 2: Acrobot-v1

MountainCar-v0

Figure 3: MountainCar-v0

MountainCar

- **HyperParameters**

Config parameters: {} 9 keys

batch_size: 128

eps_decay: 20,000

eps_end: 0.05

eps_start: 1

gamma: 0.9999

learning_rate: 0.001

memory_size: 2,000,000

num_episodes: 1,500

target_update_freq: 500

DQN Results

Algorithm: "DQN"

Final_Status: "Completed"

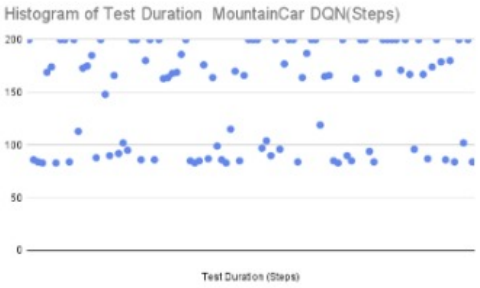
test/avg_episode_duration: 147.54

test/avg_reward: -147.54

test/durations_table: "table-file"

test/std_episode_duration: 48.28693819243461

test/std_reward: 48.28693819243461



DDQN Results

Algorithm: "DDQN"

Final_Status: "Completed"

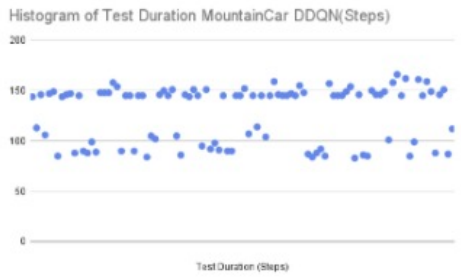
test/avg_episode_duration: 127.82

test/avg_reward: -127.82

test/durations_table: "table-file"

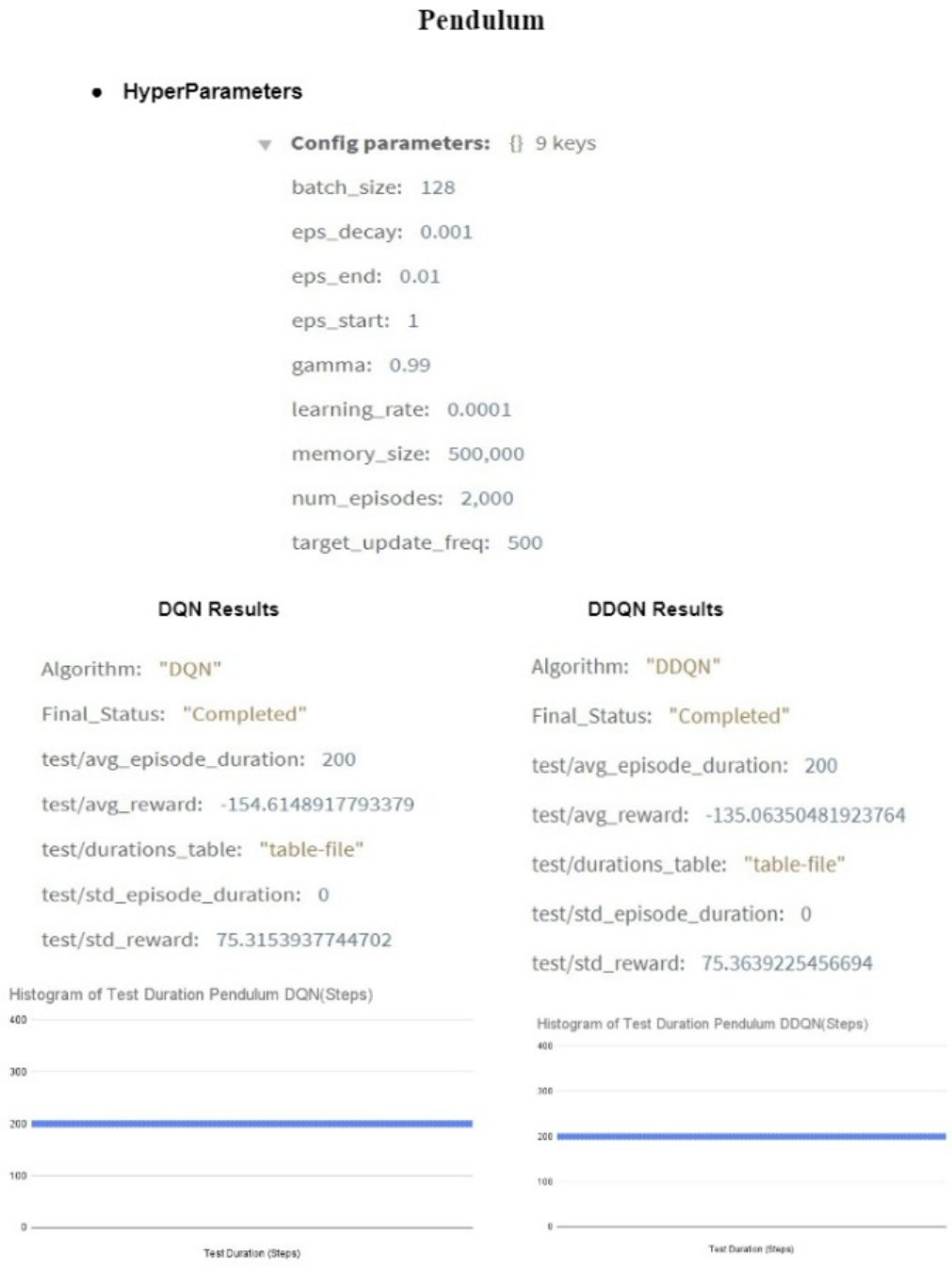
test/std_episode_duration: 27.52939519858727

test/std_reward: 27.52939519858727



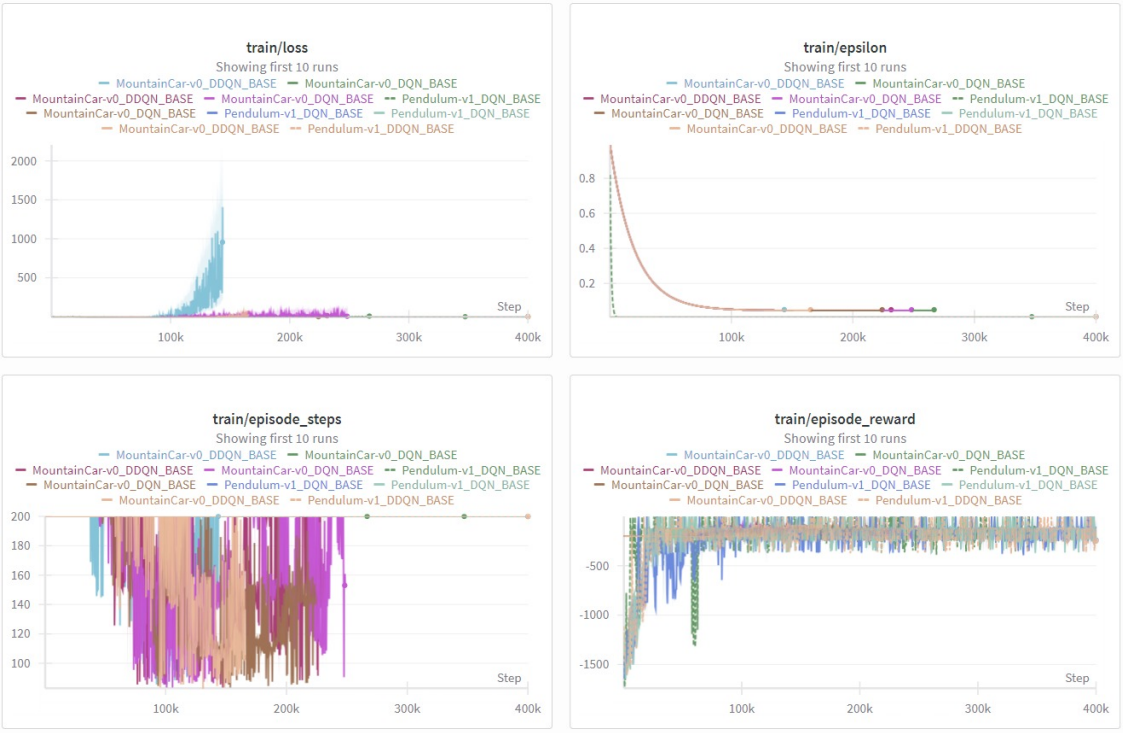
Pendulum-v1

Figure 4: Pendulum-v1



All Training Runs

Figure 5: All Training Runs



Q3 — Effect of Each Hyperparameter on RL Training and Performance

Table 2: Effect of Hyperparameters on DQN/DDQN Training

Hyperparameter	Effect on Training	Typical Observation
Discount Factor (γ)	Balances importance of future vs. immediate rewards	High γ (0.99) leads to stable, long-term policies; low γ causes short-sighted actions
Epsilon Decay Rate	Controls the speed of exploration decay	Fast decay $\rightarrow$ exploitation; slow decay $\rightarrow$ better exploration and stability
Learning Rate (α)	Determines how quickly weights are updated	High α may cause divergence; low α slows convergence
Replay Memory Size	Stores past experiences for sampling	Large memory improves diversity and stability; small memory repeats transitions
Batch Size	Number of samples per gradient update	Small batch $\rightarrow$ noisy updates; large batch $\rightarrow$ smoother but slower learning

Q4 — How Well DQN/DDQN Solve Each Classical Environment

For CartPole-v1, Acrobot-v1, and Mountain Car-v0:

- DQN/DDQN are **extremely well-suited** for these.
- **Reason:** These environments all have **discrete action spaces** (e.g., "push left" or "push right"). DQN is fundamentally designed to handle this. It works by having the neural network output one Q-value for each possible discrete action, and the agent simply picks the action with the highest Q-value.

For Pendulum-v1:

- DQN/DDQN are **fundamentally unsuited** for this environment in their standard form.
- **Reason:** Pendulum has a **continuous action space**. The action isn't "left or right," but a specific torque value (e.g., -1.56, 0.23, 1.88, etc.). There are infinitely many possible actions.
- A standard DQN network cannot output an infinite number of Q-values. Trying to use DQN here would force you to *discretize* the action space (e.g., treat it as just "hard left," "nothing," "hard right"), which is an approximation that often performs very poorly.