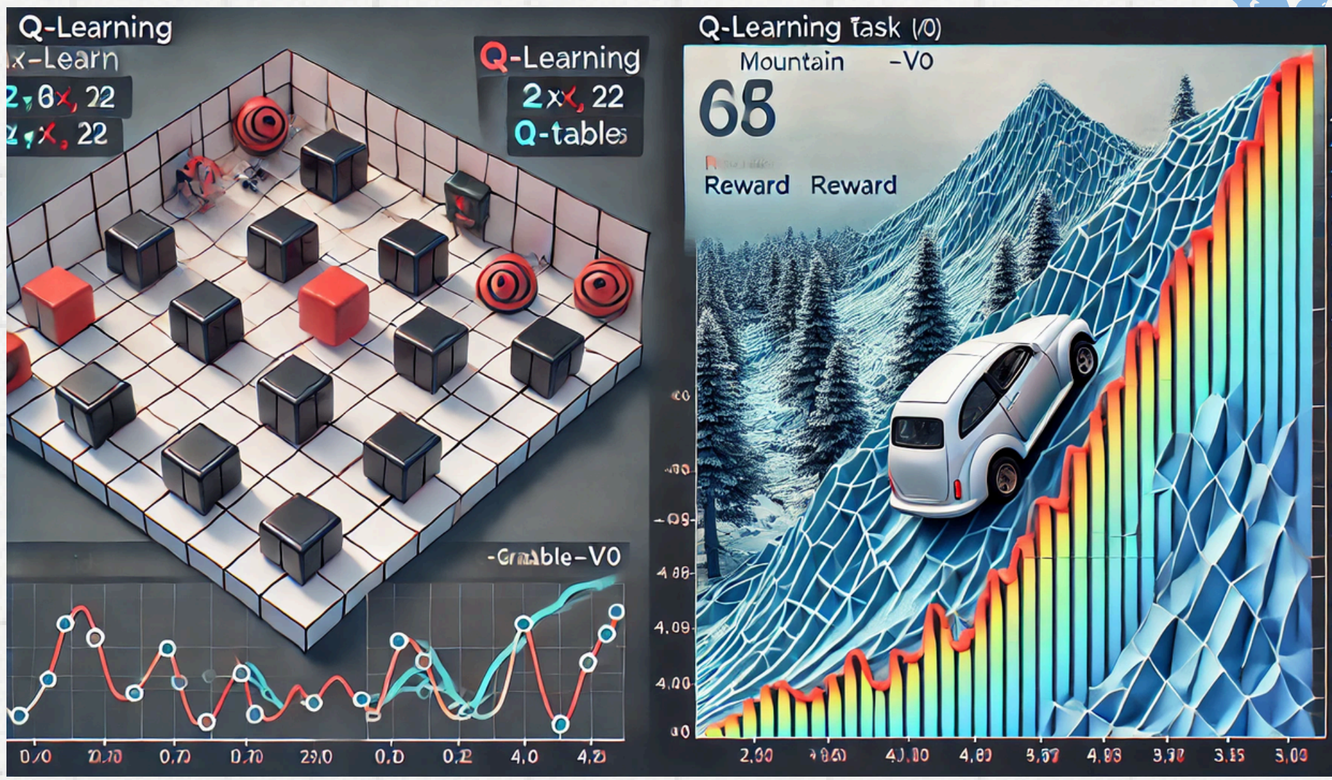
Task: Implementing Q-Learning in a Hybrid Approach

Presented by Marina Reda
221101235 - AIS

Task Overview and Approach

- Objective: Implement Q-Learning in two environments: a custom Gridworld and MountainCar-v0.
- Custom Gridworld: 5x5 grid, start at (0,0), goal at (4,4), obstacles, and reward system.
- MountainCar-v0: Gym environment requiring momentum to reach the goal.
- Approach:
  - Initialize Q-table.
  - Use epsilon-greedy policy.
  - Train over multiple episodes.
  - Visualize and evaluate policies.



Findings from Custom Gridworld

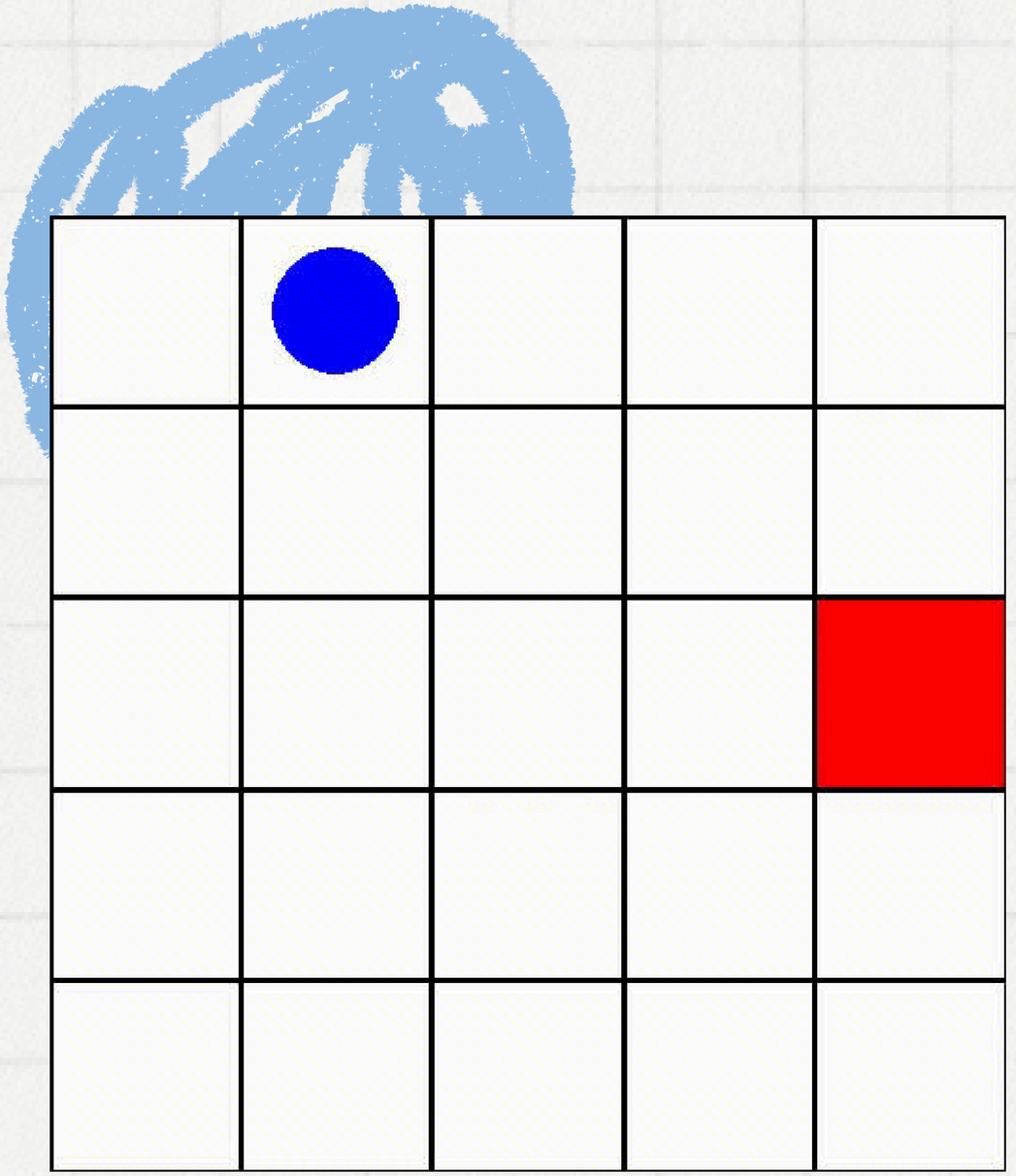
Custom Gridworld environments are widely used in reinforcement learning (RL) to test and develop algorithms due to their simplicity and flexibility. These environments consist of a grid where an agent navigates to achieve specific goals, allowing researchers to model various scenarios and challenges.

01. Final Path: [(0,0), (1,0), (2,0), (2,1), (3,1), (4,1), (4,2), (4,3), (4,4)].

02.
- Rewards:
  - Total Reward for Path: 93.
  - Consistently high performance across episodes.

03.
- Learned Policy:
  - Optimal path avoids obstacles.
  - Epsilon decay reached a stable point (0.050).

More Information:



Environment Details:

- Custom Gridworlds are two-dimensional grids where each cell represents a state. The agent can move between adjacent cells based on predefined actions. The environment can include various elements such as obstacles, goals, and rewards. For example, in MATLAB's Reinforcement Learning Toolbox, the createGridWorld function allows users to specify grid sizes and customize the environment to suit specific needs.

Solved Criteria:

- The criteria for solving a Custom Gridworld environment depend on the specific design and objectives set by the developer. Typically, an environment is considered solved when the agent consistently achieves the desired goal or performance metric. For instance, in a simple gridworld with a goal state, the agent might be considered to have solved the environment when it reaches the goal state within a certain number of steps or with a specific success rate.

For more detailed information on creating and customizing Gridworld environments, you can refer to the following resources:

- Create Custom Grid World Environments - MathWorks
- Make your own custom environment - Gymnasium Documentation
- Building Custom Grid Environments for Reinforcement Learning in Gymnasium

RESULT

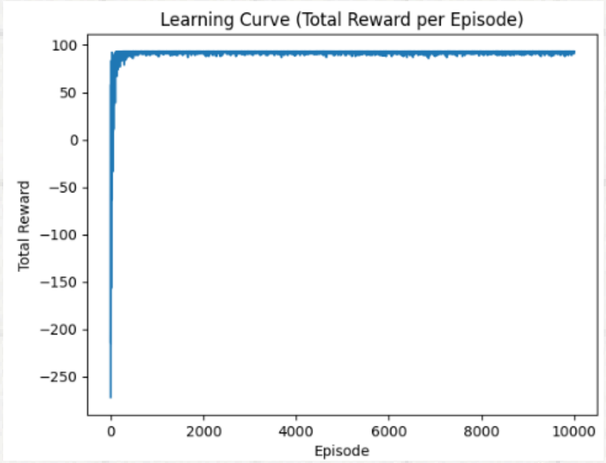
01

Episode 1000/10000, Total Reward: 93, Epsilon: 0.050
Episode 2000/10000, Total Reward: 93, Epsilon: 0.050
Episode 3000/10000, Total Reward: 93, Epsilon: 0.050
Episode 4000/10000, Total Reward: 93, Epsilon: 0.050
Episode 5000/10000, Total Reward: 93, Epsilon: 0.050
Episode 6000/10000, Total Reward: 93, Epsilon: 0.050
Episode 7000/10000, Total Reward: 92, Epsilon: 0.050
Episode 8000/10000, Total Reward: 93, Epsilon: 0.050
Episode 9000/10000, Total Reward: 93, Epsilon: 0.050
Episode 10000/10000, Total Reward: 93, Epsilon: 0.050

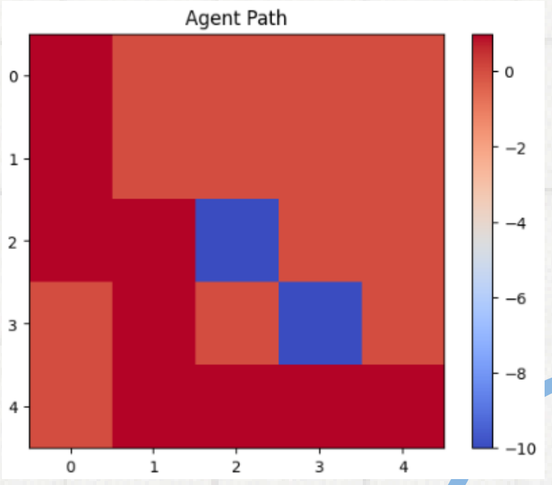
02

Final Path: [(0, 0), (1, 0), (2, 0), (2, 1), (3, 1), (4, 1), (4, 2), (4, 3), (4, 4)]
Total Reward for the Path: 93

03

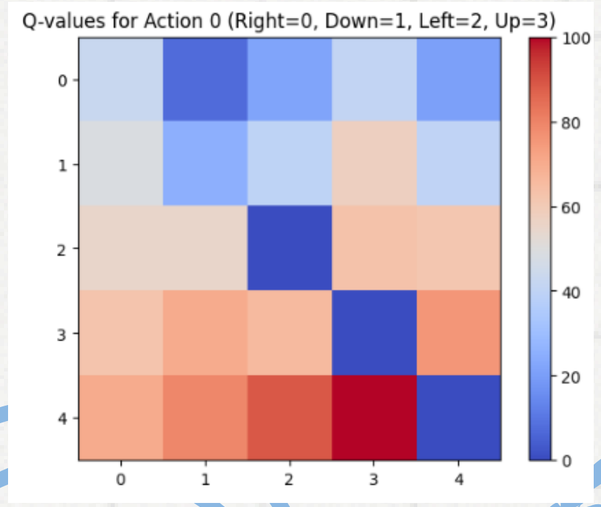


04

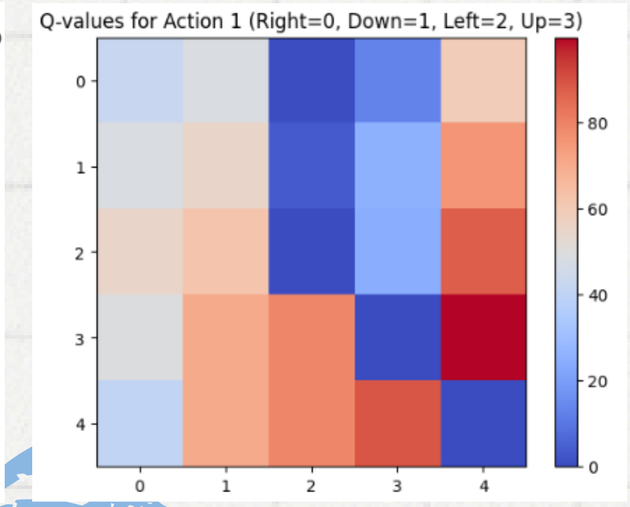


RESULT

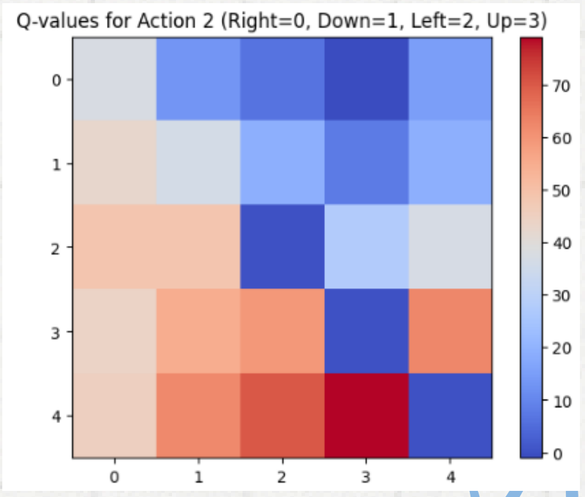
01



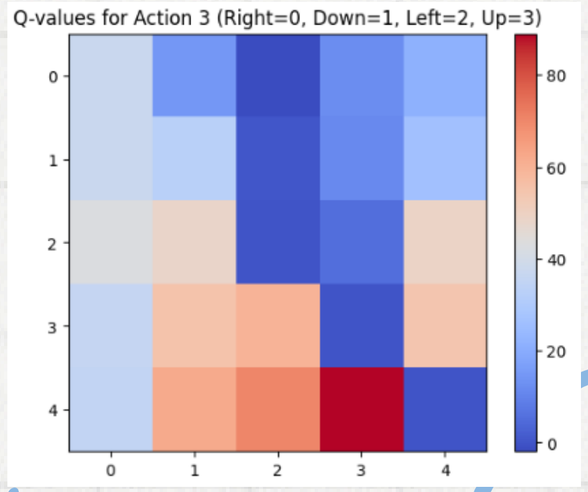
02



03



04



Findings from MountainCar-v0

MountainCar-v0 is a classic reinforcement learning environment where an underpowered car must learn to reach the top of a hill. The challenge lies in the car's inability to ascend the hill directly due to insufficient power; it must first build momentum by moving back and forth.

01.

State and Action Space:

- State Space: Continuous (position, velocity).
- Action Space: Discrete (push left, no push, push right).

02.

Training Results:

- Initial episodes struggled (-200 reward).
- Gradual improvement after 10,000 episodes (best: -151).
- Epsilon decay stabilized (0.050).

03.

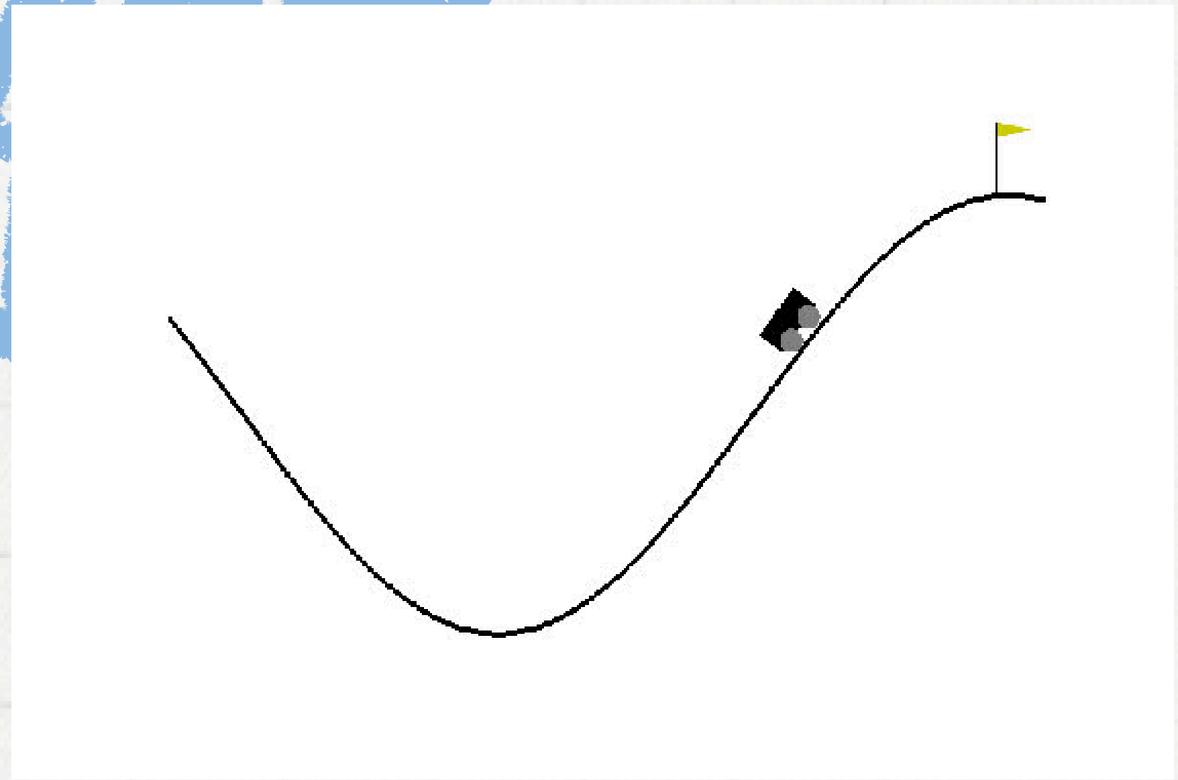
Challenges:

- Sparse reward structure required extensive exploration.
- High sensitivity to hyperparameters.

More Information:

Environment Details:

- State Space: Continuous, defined by two variables: position (ranging from -1.2 to 0.6) and velocity (ranging from -0.07 to 0.07).
- Action Space: Discrete, with three possible actions: push left, no push, and push right.
- Reward Structure: The agent receives a reward of -1 for each time step until it reaches the goal position of 0.5.
- Episode Termination: An episode concludes when the car reaches the position of 0.5 or after 200 time steps, whichever comes first.



Solved Criteria:

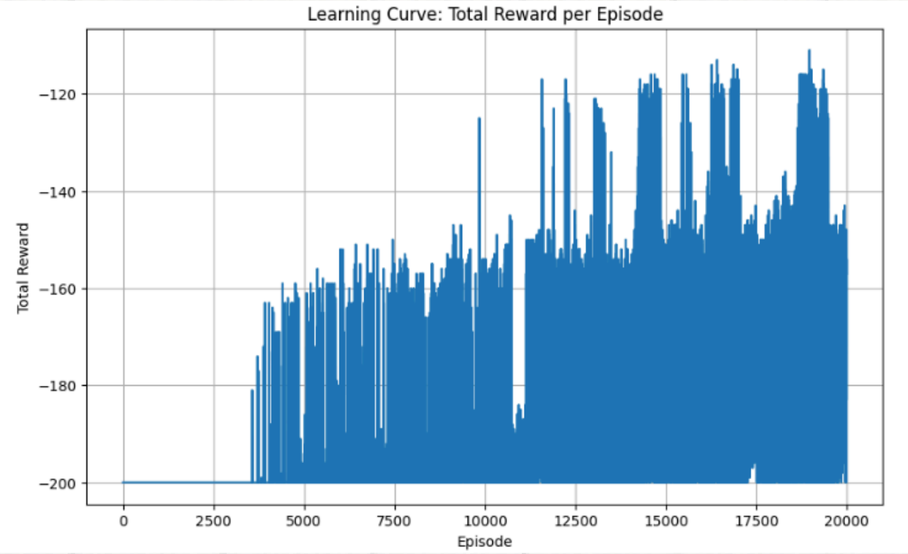
- The environment is considered "solved" when an agent achieves an average reward of -110.0 over 100 consecutive episodes.

RESULT

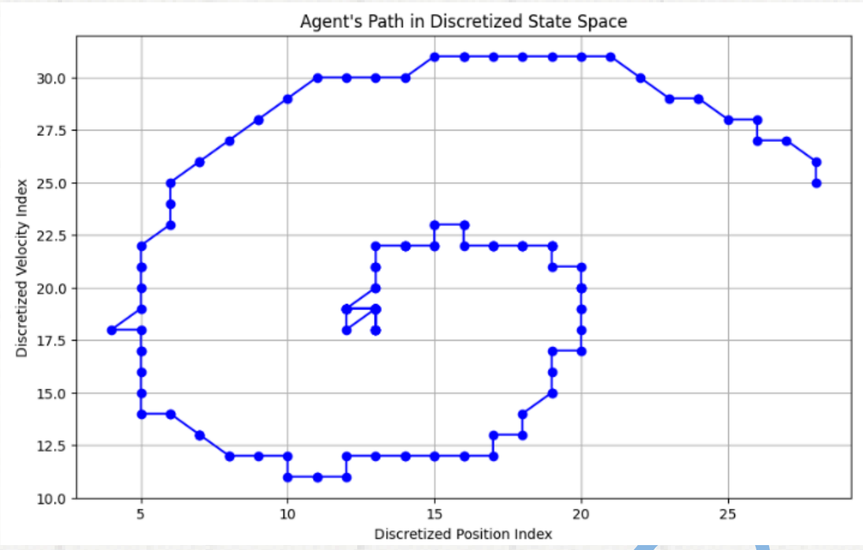
01

State Space: Box([-1.2 -0.07], [0.6 0.07], (2,), float32) Action Space: Discrete(3) Episode 1000/20000, Total Reward: -200.0, Epsilon: 0.050 Episode 2000/20000, Total Reward: -200.0, Epsilon: 0.050 Episode 3000/20000, Total Reward: -200.0, Epsilon: 0.050 Episode 4000/20000, Total Reward: -200.0, Epsilon: 0.050 Episode 5000/20000, Total Reward: -200.0, Epsilon: 0.050 Episode 6000/20000, Total Reward: -200.0, Epsilon: 0.050 Episode 7000/20000, Total Reward: -200.0, Epsilon: 0.050 Episode 8000/20000, Total Reward: -200.0, Epsilon: 0.050 Episode 9000/20000, Total Reward: -200.0, Epsilon: 0.050 Episode 10000/20000, Total Reward: -200.0, Epsilon: 0.050 Episode 11000/20000, Total Reward: -192.0, Epsilon: 0.050 Episode 12000/20000, Total Reward: -200.0, Epsilon: 0.050 Episode 13000/20000, Total Reward: -167.0, Epsilon: 0.050 Episode 14000/20000, Total Reward: -162.0, Epsilon: 0.050 Episode 15000/20000, Total Reward: -200.0, Epsilon: 0.050 Episode 16000/20000, Total Reward: -159.0, Epsilon: 0.050 Episode 17000/20000, Total Reward: -151.0, Epsilon: 0.050 Episode 18000/20000, Total Reward: -197.0, Epsilon: 0.050 Episode 19000/20000, Total Reward: -200.0, Epsilon: 0.050 Episode 20000/20000, Total Reward: -157.0, Epsilon: 0.050

02



03



Comparative Analysis

- Custom Gridworld:
  - Clear convergence to an optimal policy.
  - Rewards steadily improved.
- MountainCar-v0:
  - More challenging due to continuous state space.
  - Slow improvement; best performance at -151 total reward.
- Key Observations:
  - Custom environment is easier to train due to discrete states and well-defined rewards.
  - Gym environments require more sophisticated exploration and tuning.

Challenges and Insights

- Challenges:
  - Gridworld: Balancing exploration and exploitation.
  - MountainCar-v0: Handling continuous states and sparse rewards.
- Insights:
  - Importance of reward design for faster convergence.
  - Epsilon-greedy effectively balances exploration and exploitation.
  - Gym environments demand more advanced strategies for optimal performance.

Thank you very much!

marina.mekhael@gu.edu.eg