DRL Coursework Report

Name: Elias Michael
Email: elias.michael@city.ac.uk

GitHub Link: https://github.com/eliasm99/DRL

---

1. Basic Part

1.1 Define the Environment and the Problem to Be Solved

In this project, we implement a custom reinforcement learning environment inspired by the upcoming *Pirates of the Caribbean* film, where an agent navigates a grid in search of treasure while avoiding environmental hazards. The grid is modeled as a 5×5 map with various terrain types: a starting point, a treasure goal, pits (which terminate the episode with heavy penalty), quicksand (with moderate penalties), and a small negative reward for each step to encourage efficiency.

This setup is deliberately simple yet strategically rich, allowing us to explore how a Q-learning agent balances short-term risks and long-term rewards in a penalty-heavy state space. The agent can move up, down, left, or right unless blocked by grid boundaries, and the goal is to reach the treasure while minimizing cumulative negative reward. Figure 1 below illustrates the initial layout of the environment, including hazard placements and key cells.

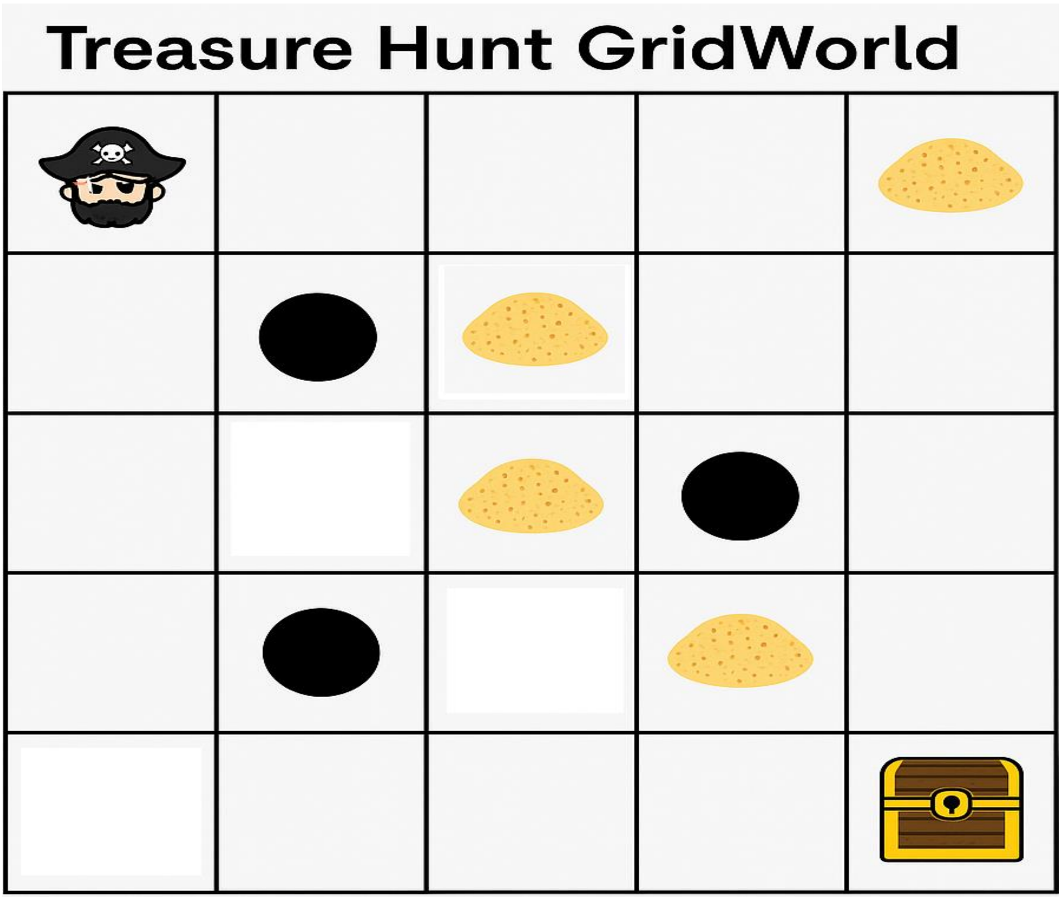


Figure 1: Showing the Grid for our Basic Part

1.2 Transition Function and Reward Function

In our environment, the state transition function is deterministic: the agent can move up, down, left, or right, and transitions to the adjacent cell in that direction unless the move would cross the grid boundary, in which case it remains in the current state. Each state is defined as a pair of coordinates (row, column) on the 5×5 grid. The reward function is designed to encourage short, safe paths to the goal while avoiding hazards. Moving into an empty cell results in a small penalty of -0.5 to discourage inefficient wandering. Stepping into a quicksand cell gives a moderate penalty of -5, while falling into a pit yields a large penalty of -15 and terminates the episode. Reaching the goal delivers a reward of +100 and also ends the episode. This reward structure, summarised in Table 1, supports the agent in learning to balance exploration, safety, and efficiency within a sparse reward landscape.

Cell Type	Symbol	Description	Reward	Terminal State
Start	S	Starting cell	0	No
Goal	G	Treasure chest	+100	Yes
Pit	P	Deadly trap	-15	Yes
Quicksand	Q	Slows movement	-5	No
Empty Cell	–	Normal path	-0.5	No

Table 1: Reward Function Table along with symbols corresponding to Figure 1

1.3 Set Up Parameters

To train the agent using Q-learning, we defined three core hyperparameters: the learning rate (α), the discount factor (γ), and the exploration rate (ε). The learning rate $\alpha \in \{0.1, 0.3, 0.5\}$ controls how much newly acquired information overrides previous knowledge, with higher values leading to faster adaptation. The discount factor $\gamma \in \{0.7, 0.9\}$ determines the importance of long-term rewards, encouraging forward planning. For exploration, we implemented an ε-greedy strategy with $\varepsilon \in \{0.1, 0.3, 0.5\}$, allowing the agent to select a random action with probability ε and exploit its best-known action otherwise. Additionally, we introduced ε-decay, gradually reducing ε after each episode by multiplying it by 0.995 until it reaches a minimum of 0.05, allowing the agent to shift from exploration to exploitation over time. These parameters form the basis for the grid search conducted in Task 5, where we investigate their effect on learning performance under both ε-greedy and softmax exploration policies.

1.4 Run the Q-learning Algorithm and Represent Its Performance

Using the defined hyperparameters, we implemented the Q-learning algorithm over 1000 training episodes. At the start of each episode, the agent was placed at the initial position and selected actions based on the ε-greedy policy. The Q-values were updated after each step using the Bellman equation [1], which adjusts the estimated value of a state-action pair based on the received reward and the expected value of the next state. Throughout training, the agent accumulated a total reward per episode, which we tracked to evaluate learning progress. To visualize performance trends, we plotted a moving average over the reward history. The first 400 episodes were characterised by high variability and noisy reward patterns, reflecting the agent's exploratory behavior and lack of learned strategy. From episode 400 onwards, the agent began to show more consistent improvements, reaching the goal more reliably and avoiding hazards more frequently. This baseline training run provides a reference point for the comparative evaluations in the next task.

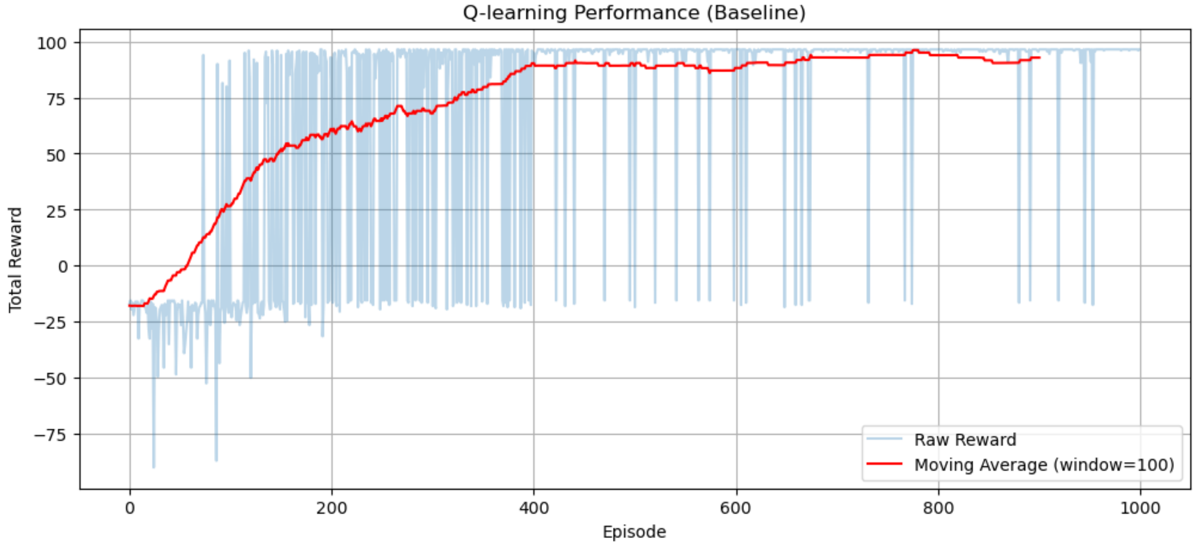


Figure 2: Total Reward with number of episodes with moving average line per 100 episodes

1.5 Repeat the Experiment with Different Parameter Values and Policies

To systematically investigate the influence of hyperparameter settings and exploration strategies on learning performance, we conducted an extensive grid search across two policy types: ε-greedy with ε-decay and softmax (Boltzmann) action selection. For the ε-greedy policy, we evaluated combinations of learning rate $\alpha \in \{0.1, 0.3, 0.5\}$, discount factor $\gamma \in \{0.7, 0.9\}$, and initial exploration rate $\varepsilon \in \{0.1, 0.3, 0.5\}$, resulting in 18 configurations. Each ε value was decayed gradually across episodes, encouraging exploitation in later training. For the softmax policy, we explored $\alpha \in \{0.1, 0.3\}$, $\gamma \in \{0.7, 0.9\}$, and temperature $\tau \in \{0.5, 1.0, 2.0\}$, which determines how closely action probabilities follow the Q-values. This produced 12 additional configurations. Every setup was trained for 1000 episodes, and we recorded both the total reward per episode and summary statistics including average reward, standard deviation, and training time. The results were visualised through smoothed learning curves and tabulated to highlight the most effective parameter combinations within each policy type.

1.6 Analyze the Results Quantitatively and Qualitatively

We evaluated agent performance using average reward, standard deviation (as a measure of stability), and training duration across all 30 tested configurations. Figures 3 and 4 present the smoothed learning curves of the top five ε-greedy and softmax configurations, respectively, while Tables 2 and 3 report their corresponding performance metrics. All top-performing softmax configurations achieved an average reward of 96.5 with a standard deviation of 0.0, indicating perfectly consistent convergence to the goal state. These policies typically converged within the first 100 episodes, demonstrating rapid and stable learning. In contrast, the top ε-greedy configuration ($\alpha = 0.1$, $\gamma = 0.9$, $\varepsilon = 0.5$) achieved an average reward of 94.875 with a standard deviation of 11.33, with convergence generally occurring around episode 400 for most configurations.

Training time across all runs remained low, though ε-greedy configurations averaged approximately **0.028 seconds** per run, while softmax configurations required around **0.26**

seconds on average — roughly **ten times longer**. Within ε-greedy runs, increasing the learning rate α from 0.1 to 0.5 often led to **nearly double** the training time. In contrast, training times for softmax were relatively insensitive to changes in α, γ, or τ. Qualitatively, softmax policies consistently demonstrated smooth reward progression and early stability. ε-greedy agents, while also effective, required more careful tuning and showed greater variance in convergence behavior, particularly under low ε or high α, where learning was occasionally unstable or slower to converge.

In conclusion, softmax offered highly stable and rapid convergence across a range of settings but at a higher computational cost. ε-greedy remained a strong alternative, particularly in terms of training efficiency and adaptability, achieving near-optimal performance when appropriately tuned. These results highlight the importance of aligning exploration strategy and parameter sensitivity with task complexity and computational constraints.

α	γ	ε	Policy	Avg Reward	Std Dev	Duration (s)
0.1	0.9	0.5	ε-greedy	94.875	11.33	0.018
0.5	0.9	0.3	ε-greedy	94.860	11.44	0.031
0.1	0.9	0.1	ε-greedy	94.745	11.45	0.028
0.1	0.9	0.3	ε-greedy	93.885	15.70	0.029
0.3	0.7	0.5	ε-greedy	93.780	15.77	0.031

Table 2: Performance metrics (average reward, standard deviation, training time) for top 5 ε-greedy configurations.

α	γ	Policy	Avg Reward	Std Dev	Duration (s)	τ
0.1	0.7	softmax	96.500	0.00	0.250	0.5
0.1	0.7	softmax	96.500	0.00	0.264	1.0
0.1	0.9	softmax	96.500	0.00	0.262	0.5
0.1	0.9	softmax	96.500	0.00	0.266	1.0
0.1	0.9	softmax	96.500	0.00	0.297	2.0

Table 3: Performance metrics (average reward, standard deviation, training time) for top 5 softmax configurations.

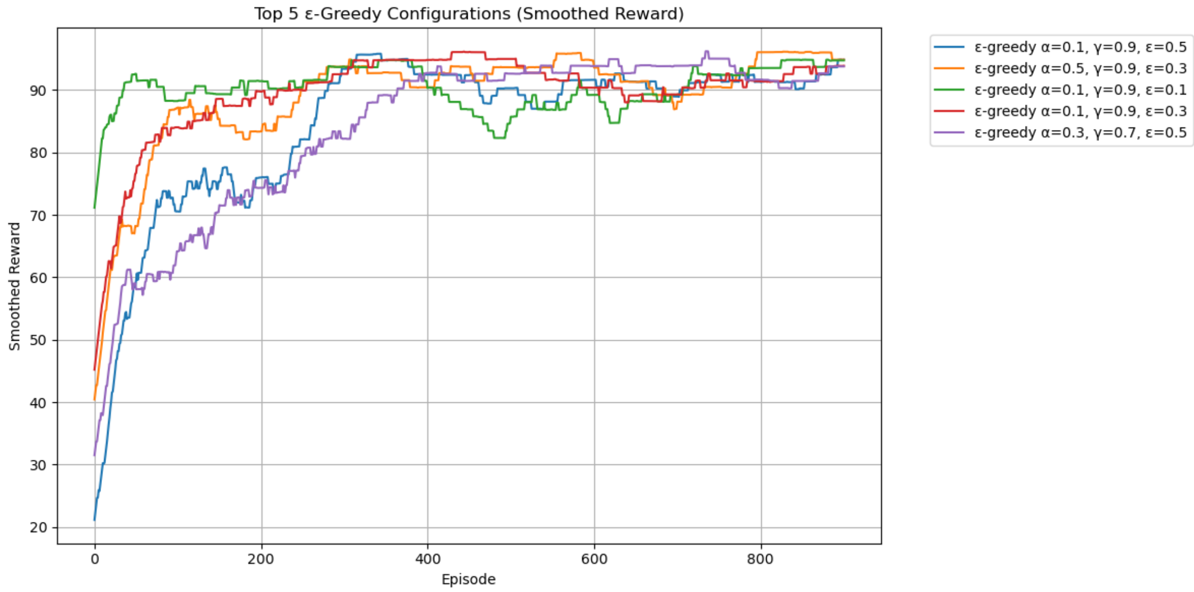


Figure 3: Smoothed total reward curves (moving average) for the top 5 ε-greedy configurations.

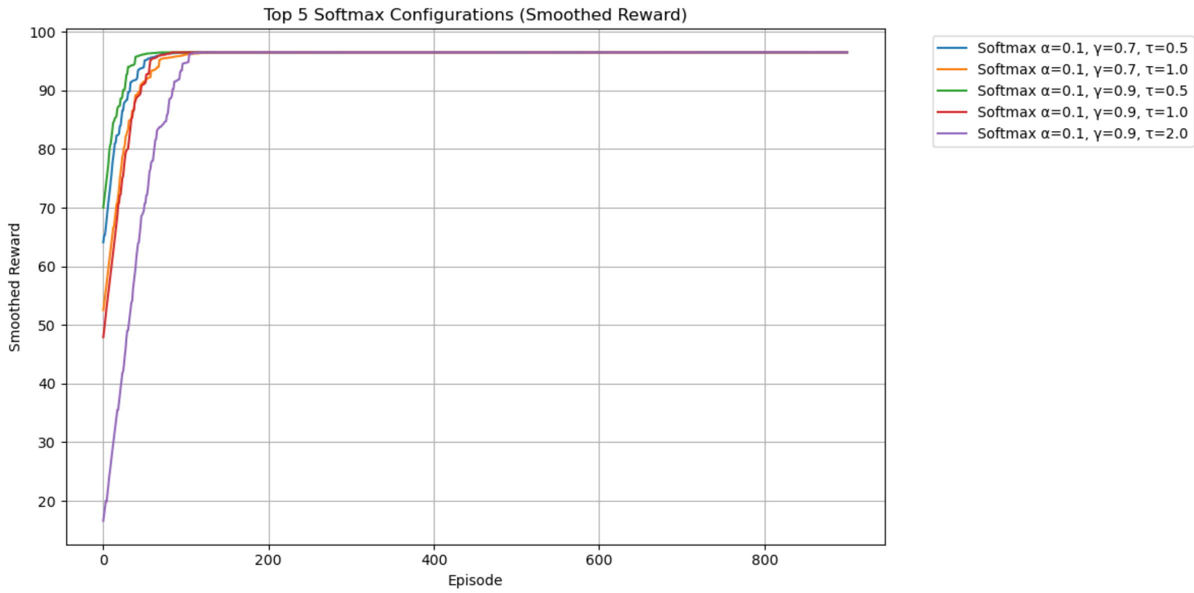


Figure 4: Smoothed total reward curves (moving average) for the top 5 softmax configurations.

---

2. Advanced Part

2.1 DQN Implementation and 2 Improvements

In this section, we apply the Deep Q-Network (DQN) algorithm to the LunarLander-v2 environment [5], a classic benchmark in reinforcement learning. Unlike tabular Q-learning methods, DQN utilizes a neural network neural network to approximate action-values [3]. The goal is to develop an agent capable of mastering complex control strategies in a continuous state space.

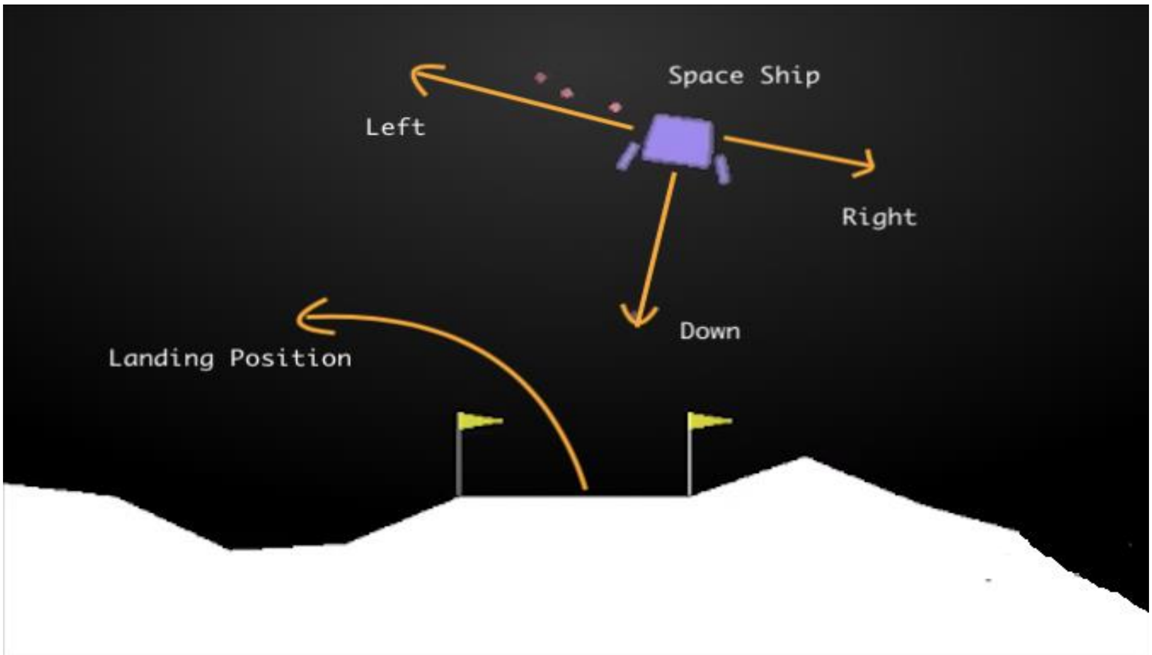


Figure 5: Showing the LunarLander-v2 Game Interface

Task and Domain

The LunarLander-v2 environment consists of an agent controlling a lander on the moon's surface. The agent observes an 8-dimensional state vector, which encodes position, velocity, angle, and contact information. It can choose from four discrete actions to fire the main or side engines. The task is to land the craft safely between flags on a designated pad. A reward is given for successful landing and penalized for crashing or using excessive fuel. Each episode ends either upon a successful landing or crash.

We define our Q-network as a multi-layer perceptron (MLP) with two hidden layers of 128 units each, followed by an output layer predicting Q-values for each of the four possible actions. We initialize all agents with fixed random seeds for reproducibility.

Key Concepts in DQN and Motivation for Improvements

Despite the powerful generalisation capabilities of deep neural networks, the practical deployment of Deep Q-Networks (DQNs) introduces a number of stability and efficiency challenges. These include the presence of temporal correlations in sequential data, non-stationary and unstable learning targets due to bootstrapping, and the well-documented issue of overestimation of action values when using a maximisation operator over noisy Q-value estimates. To address these limitations and improve both the stability and sample efficiency of the learning process, we incorporated two algorithmic enhancements into the baseline DQN framework: Prioritized Experience Replay (PER) and Double Q-learning. PER aims to accelerate convergence and improve learning efficiency by sampling transitions based on their temporal-difference (TD) error, ensuring that high-error, informative experiences are revisited more frequently. Double Q-learning, on the other hand, mitigates overestimation bias through decoupling [2] the action selection and evaluation processes, thus providing more accurate value targets. Together, these improvements offer a principled approach to stabilising and enhancing value-based reinforcement learning in complex environments such as LunarLander-v2.

Prioritized Experience Replay (PER)

In traditional Deep Q-Network (DQN) implementations, experiences stored in the replay buffer are sampled uniformly during training. While this approach is straightforward, it often results in inefficient learning, as it treats all transitions as equally important regardless of their informativeness. Prioritized Experience Replay (PER) addresses this limitation by assigning a sampling probability to each transition based on its temporal-difference (TD) error — the discrepancy between the predicted and actual Q-values. Transitions with higher TD errors are presumed to carry more useful learning signals and are thus replayed more frequently. In our implementation, we use proportional prioritization, where the probability of sampling a given transition increases with its TD error [4]. A tunable parameter (denoted alpha in literature) controls the degree of prioritization: when alpha is set to zero, the sampling reverts to uniform. Our implementation dynamically updates priorities after each learning step, enabling the agent to continually focus on the most informative transitions. This not only improves sample efficiency but also accelerates convergence by allowing the agent to revisit significant experiences more often.

Double Q-Learning

One known drawback of standard DQN is its tendency to overestimate action values due to using the same network for both action selection and evaluation when calculating targets. This overestimation can destabilise training and lead to poor policy quality. Double Q-learning tackles this issue by decoupling these two roles. In this variant, the online network is responsible for selecting the best action in the next state, while the target network evaluates the value of that action. This split reduces the positive bias in target Q-value estimates and leads to more stable learning. Instead of computing the target using the maximum Q-value directly from the target network, Double Q-learning uses the action selected by the online network but still evaluates it using the target network. This small architectural change significantly improves robustness, particularly in environments with noisy observations or high reward variance.

2.2 Training Setup and Analysis

We trained each agent variant for 500 episodes using a consistent training loop and set of hyperparameters to enable a fair comparison. The training process followed an epsilon-greedy strategy, with the exploration rate starting at 1.0 and decaying to 0.05 over time. The target network was updated every 10 episodes to stabilize training, and each episode was capped at a maximum of 1000 steps. Throughout training, transitions were stored in the replay buffer and sampled to update the Q-network. The use of Double Q-learning and Prioritized Experience Replay (PER) was toggled via configuration flags during agent initialization to isolate their individual and combined effects.

The four agent configurations evaluated were: (1) Vanilla DQN (no enhancements), (2) DQN with Double Q-learning, (3) DQN with PER, and (4) DQN with both Double Q-learning and PER. Quantitative performance was measured by computing the average reward over the final 100 episodes, along with the standard deviation and total training time. As shown in the summary table, the vanilla DQN agent achieved a low average reward of -78.57 and displayed the highest variance in performance. Introducing Double Q-learning yielded a notable improvement, reducing overestimation bias and increasing the average reward to -9.93 while also lowering variance. The PER-enhanced agent, which prioritized sampling transitions with higher TD error, achieved a modest positive reward of 9.92 but required significantly more training time. The combination of both enhancements resulted in the most robust and performant agent, achieving a final average reward of 71.45 with improved stability and only a moderate increase in training time relative to the vanilla baseline.

Agent Variant	Avg Reward (Last 100)	Std Dev	Training Time (s)
Vanilla DQN	-78.57	168.08	1130.60
DQN + Double Q-learning	-9.93	117.21	1554.09
DQN + PER	9.92	135.86	2303.06
DQN + Double + PER	71.45	138.01	1632.62

Table 4: Performance metrics (average reward, standard deviation, training time) for the 4 different Agent Variants

To ensure that the reported gains were not specific to a single seed or configuration, we performed grid search tuning on learning rate, discount factor (gamma), and epsilon decay rate, evaluating models across multiple random seeds. While the highest single-seed performance (average reward ≈140) was obtained with learning rate = 0.0005, gamma = 0.99, and epsilon decay = 0.99 (seed = 42), we selected a more generalizable configuration with learning rate = 0.0001, gamma = 0.99, and epsilon decay = 0.995 based on consistent performance across seeds. This model achieved a last-100-episode average reward of 65.5, striking a balance between stability, sample efficiency, and robustness.

Qualitative analysis further supports these findings. Agents using PER showed smoother and more stable convergence trajectories, especially during the mid-to-late training stages. The improved sampling mechanism helped reduce redundant updates and prioritize informative transitions, leading to faster policy refinement. In contrast, Double Q-learning offered improvements by mitigating overestimation, but its reward progression plateaued around episode 200 and exhibited noticeable fluctuations thereafter. When combined, the benefits of both methods became evident: early exploration was accelerated, value estimates became more accurate, and overall learning stabilized.

The performance trends are visualized in the smoothed reward curve below, highlighting the differential progress of the four agents. While the Double Q-learning agent made rapid early gains, its improvement plateaued mid-training. The PER-enhanced agent progressed more gradually but achieved steady gains. The combined approach clearly outperformed all others, benefiting from more accurate value estimation and informed sampling. In contrast, the vanilla DQN agent failed to reach positive average rewards and remained unstable throughout training.

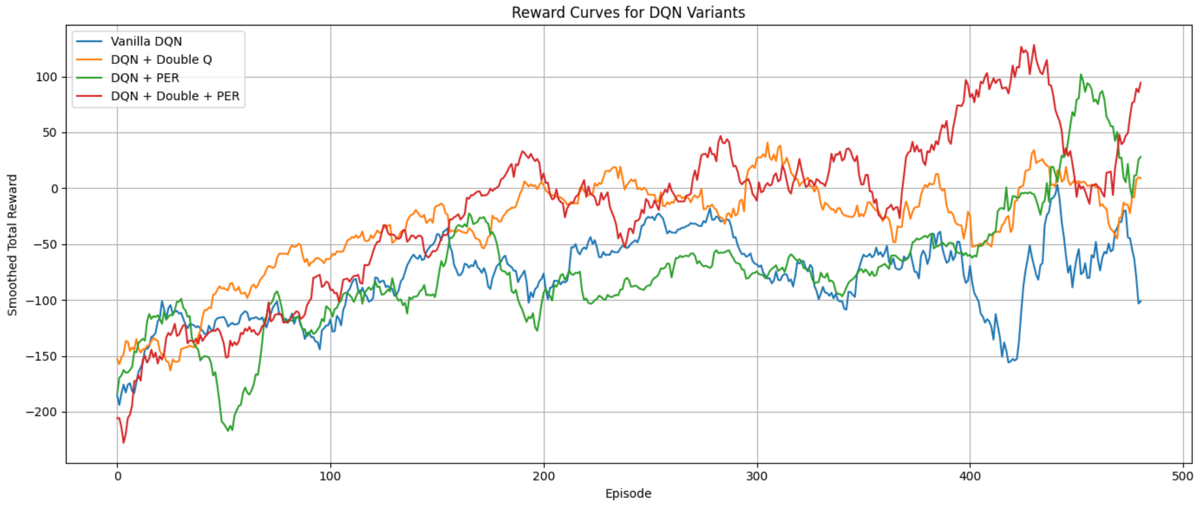


Figure 6: Total reward curves (moving average of 20) for the 4 Agent Variants.

Overall, these results underscore the importance of stabilizing targets and prioritizing experiences in deep reinforcement learning. On a complex control task such as LunarLander-v2, enhancements like PER and Double Q-learning offer tangible improvements in both efficiency and reward maximization, establishing them as effective extensions to the standard DQN framework.

2.3 RLlib Implementation of DQN on Atari Pong (RAM)

For Task 9, we implemented a Deep Q-Network (DQN) using the Ray RLlib framework on the Atari Learning Environment game Pong [6], specifically the RAM-based version ALE/Pong-ram-v5. We selected the RAM-based variant over visual counterparts to focus on the algorithmic learning dynamics without involving convolutional networks. The RAM environment encodes the full game state in just 128 bytes, enabling faster training and reduced computational complexity, making it ideal for use with a multilayer perceptron (MLP).

The agent was configured using RLlib's DQNConfig object, defining an MLP-based Q-network with two hidden layers of 256 units and ReLU activation. The environment used was ALE/Pong-ram-v5 with PyTorch as the backend. Training was carried out over 500 episodes using a single-process setup (num_rollout_workers=0) to ensure compatibility with notebook-based execution.

Exploration was handled via an epsilon-greedy strategy, with epsilon annealed linearly from 1.0 to 0.01 over 200,000 timesteps. The discount factor was set to 0.99, and the learning rate was fixed at 1e-4. A training batch size of 32 and a warm-up period of 1000 steps before learning began were used. No dueling networks or Double Q-learning enhancements were included in this baseline implementation.

The agent was trained using the following loop, which recorded the mean episode reward across iterations. This metric was plotted to visualize the agent’s learning progress. Over the 500 episodes, the agent showed moderate improvements in performance, with an increasing trend in total rewards despite substantial variance across episodes. The training curve reflects the non-trivial nature of the task even with simplified RAM inputs, indicating the necessity of further algorithmic improvements to achieve higher sample efficiency and reward stability.

2.4 Results and Observations

The learning curve for the DQN agent on the Pong-ram environment, shown in Figure 7, illustrates the episodic mean rewards over 500 training iterations. Initial performance fluctuated around low reward values, as expected, given the agent's random policy at the start. Over time, the agent exhibited a modest upward trend, suggesting that it was learning a partial policy capable of scoring occasionally. However, the reward curve remained highly variable and did not fully converge to a stable, high-reward regime.

This outcome aligns with established findings in the literature: baseline DQN models often suffer from sample inefficiency and unstable learning, particularly in stochastic environments like Pong. Without techniques such as experience replay prioritization, target networks, or double Q-learning, the agent struggles to make consistent progress. Moreover, the RAM input format, while efficient, can be difficult to interpret without domain-specific priors or more sophisticated learning mechanisms.

Nevertheless, this experiment serves as a foundational benchmark for subsequent enhancements. The current setup provides a clean, minimal baseline from which the impact of advanced techniques like PER, Double Q-learning, or Dueling architectures can be systematically assessed. This work demonstrates the effectiveness and limitations of a standard DQN setup on RAM-based Atari environments and lays the groundwork for future exploration into improved deep RL strategies.

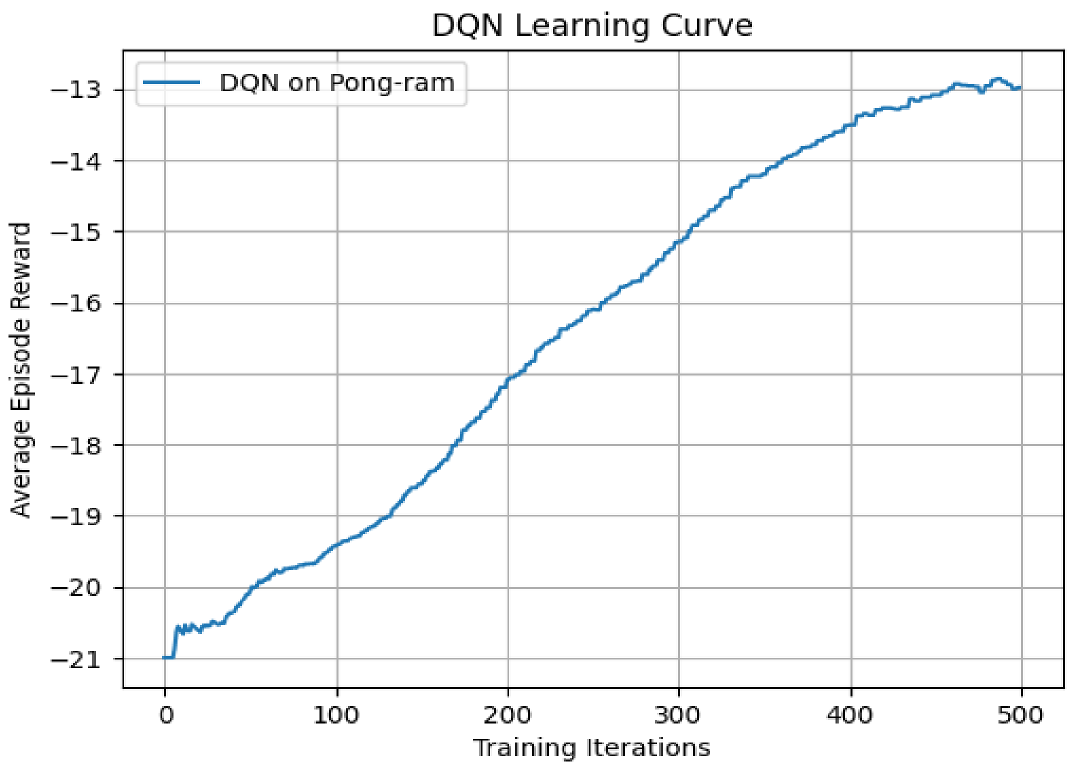


Figure 7: Rllib example without any enhancements

Extra Task: PPO Implementation

Both DQN and PPO were applied to the Pong-ram-v5 environment using a shared MLP architecture for fair comparison. DQN, a value-based method, learns Q-values for each action and uses an ε-greedy strategy for exploration. PPO, on the other hand, is a policy-gradient algorithm that directly learns a stochastic policy by optimizing a clipped surrogate objective [7].

In terms of **learning dynamics**, DQN showed slower but more stable progress, especially when combined with experience replay and target networks. PPO, while simpler in architecture, exhibited faster initial learning due to its on-policy nature but suffered from higher variance and instability during training.

Overall, DQN benefited more from enhancements like Prioritized Experience Replay and Double Q-learning, while PPO required careful tuning of learning rate and batch size. The choice between the two depends on the environment's stochasticity and the desired trade-off between stability and sample efficiency.

References:

[1] Watkins, C.J.C.H. and Dayan, P. (1992). Q-learning. *Machine Learning*, [online] 8(3-4), pp.279–292. doi:https://doi.org/10.1007/bf00992698.

[2] Van Hasselt, H. (2010). *Double Q-learning*. [online] Available at: https://proceedings.neurips.cc/paper_files/paper/2010/file/091d584fced301b442654dd8c23b3fc9-Paper.pdf.

[3] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S. and Hassabis, D. (2015). Human-level Control through Deep Reinforcement Learning. *Nature*, [online] 518(7540), pp.529–533. doi:https://doi.org/10.1038/nature14236.

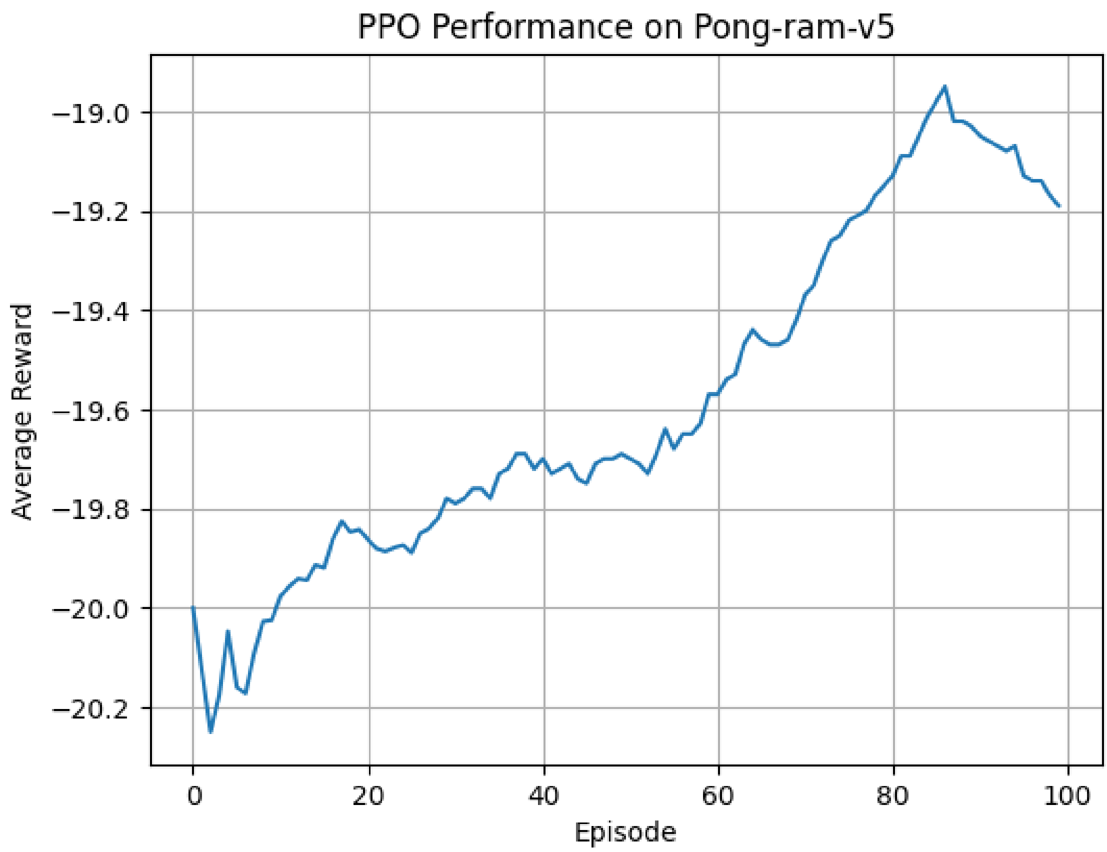
[4] Schaul, T., Quan, J., Antonoglou, I. and Silver, D. (2015). *Prioritized Experience Replay*. [online] arXiv.org. Available at: https://arxiv.org/abs/1511.05952.

[5] Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J. and Zaremba, W. (2016). *OpenAI Gym*. [online] arXiv.org. Available at: https://arxiv.org/abs/1606.01540.

[6] Liang, E., Liaw, R., Moritz, P., Nishihara, R., Fox, R., Goldberg, K., Gonzalez, J.E., Jordan, M.I. and Stoica, I. (2018). RLlib: Abstractions for Distributed Reinforcement Learning. *arXiv:1712.09381 [cs]*. [online] Available at: https://arxiv.org/abs/1712.09381.

[7] Schulman, J., Wolski, F., Dhariwal, P., Radford, A. and Klimov, O. (2017). *Proximal Policy Optimization Algorithms*. [online] arXiv.org. Available at: https://arxiv.org/abs/1707.06347.

APPENDIXES



PPO Performance Figure

Task 1: Define the Environment and Problem

We design a 5×5 **Treasure Hunt GridWorld** with a more complex layout to increase the learning challenge.

Environment Layout

- **Start** at (0, 0)
- **Goal** at (4, 4)
- **Pits**: high-penalty terminal states placed in critical path positions
- **Quicksands**: moderate-penalty areas placed to tempt shortcuts
- **Empty spaces**: minor step cost to encourage shorter paths

This layout is designed to:

- Introduce misleading paths,
- Force trade-offs between exploration and caution,
- Test how well different Q-learning configurations can adapt.

Learning Objective

Train an agent to reach the treasure while:

- Avoiding hazards,
- Learning an optimal path,
- Balancing risk and reward under uncertain exploration.

In [49]:

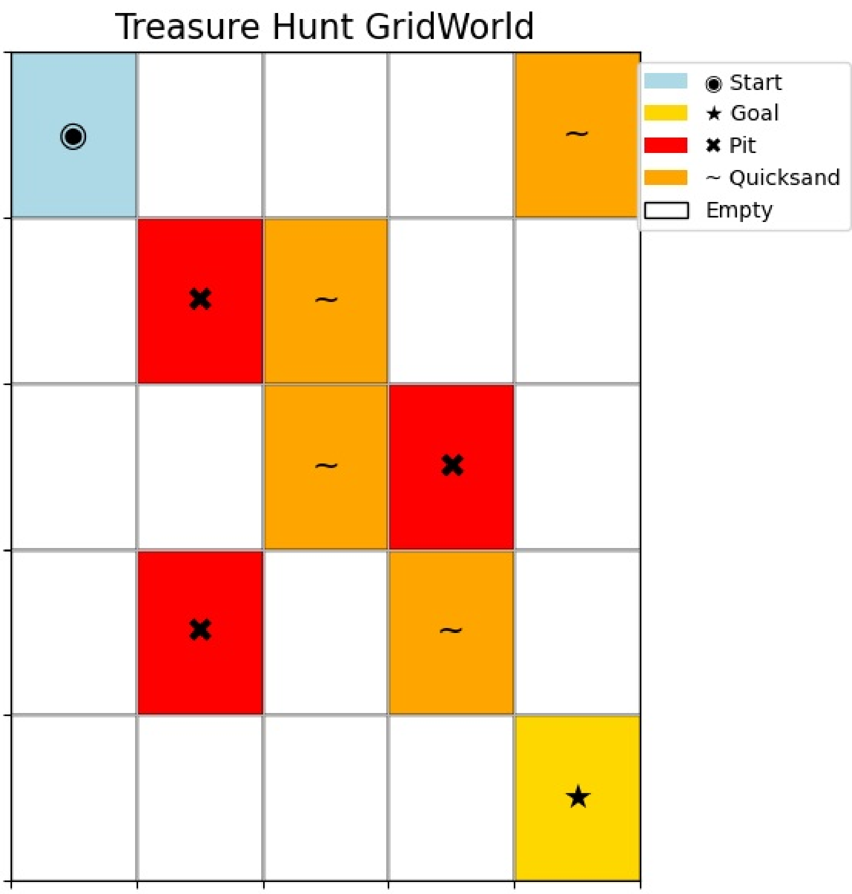
```
import numpy as np
import matplotlib.pyplot as plt
import random

# Grid dimensions
GRID_HEIGHT = 5
GRID_WIDTH = 5

# Cell types
EMPTY = 0
START = 1
GOAL = 2
PIT = 3
QUICKSAND = 4

# Action definitions
ACTIONS = ['U', 'D', 'L', 'R']
ACTION_TO_DELTA = {
    'U': (-1, 0),
    'D': (1, 0),
    'L': (0, -1),
    'R': (0, 1)
}

# Reward values
REWARD_MAP = {
    EMPTY: -0.5,
    PIT: -15,
    QUICKSAND: -5,
    GOAL: 100
}

# Environment definition
class TreasureHuntEnv:
    def __init__(self):
        self.grid = np.zeros((GRID_HEIGHT, GRID_WIDTH), dtype=int)
        self.start_pos = (0, 0)
        self.goal_pos = (4, 4)

        # Assign cell types
        self.grid[self.start_pos] = START
        self.grid[self.goal_pos] = GOAL

        # Pits placed to disrupt obvious path
        self.grid[1, 1] = PIT
        self.grid[2, 3] = PIT
        self.grid[3, 1] = PIT

        # Quicksands placed along tempting but risky routes
        self.grid[1, 2] = QUICKSAND
```

```python
        self.grid[2, 2] = QUICKSAND
        self.grid[3, 3] = QUICKSAND
        self.grid[0, 4] = QUICKSAND

        self.reset()

    def reset(self):
        self.agent_pos = self.start_pos
        return self.agent_pos

    def step(self, action):
        delta = ACTION_TO_DELTA[action]
        new_row = self.agent_pos[0] + delta[0]
        new_col = self.agent_pos[1] + delta[1]

        if 0 <= new_row < GRID_HEIGHT and 0 <= new_col < GRID_WIDTH:
            self.agent_pos = (new_row, new_col)

        cell_type = self.grid[self.agent_pos]
        reward = REWARD_MAP.get(cell_type, REWARD_MAP[EMPTY])
        done = cell_type in {GOAL, PIT}

        return self.agent_pos, reward, done

# Instantiate and print environment
env = TreasureHuntEnv()
print("□  Grid Layout:\n", env.grid)
```

```
□  Grid Layout:
 [[1 0 0 0 4]
 [0 3 4 0 0]
 [0 0 4 3 0]
 [0 3 0 4 0]
 [0 0 0 0 2]]
```

In [50]:
```python
def plot_grid_with_icons(env):
    grid = env.grid
    cmap = {
        0: 'white',       # EMPTY
        1: 'lightblue',   # START
        2: 'gold',        # GOAL
        3: 'red',         # PIT
        4: 'orange'       # QUICKSAND
    }

    icon_map = {
            0: '',      # Empty
            1: '◉',     # Start (large circle)
            2: '★',     # Goal (star)
            3: '✖',     # Pit (X)
            4: '~'      # Quicksand (wave)
}


    fig, ax = plt.subplots(figsize=(6, 6))
    for i in range(GRID_HEIGHT):
        for j in range(GRID_WIDTH):
            cell_type = grid[i, j]
            rect = plt.Rectangle((j, GRID_HEIGHT - i - 1), 1, 1,
                                 facecolor=cmap[cell_type], edgecolor='black')
            ax.add_patch(rect)
            ax.text(j + 0.5, GRID_HEIGHT - i - 0.5, icon_map[cell_type],
                    ha='center', va='center', fontsize=16)

    ax.set_xlim(0, GRID_WIDTH)
    ax.set_ylim(0, GRID_HEIGHT)
    ax.set_xticks(np.arange(0, GRID_WIDTH + 1, 1))
    ax.set_yticks(np.arange(0, GRID_HEIGHT + 1, 1))
    ax.set_xticklabels([])
    ax.set_yticklabels([])
    ax.set_title("Treasure Hunt GridWorld", fontsize=16)
    ax.grid(True)

    # Emoji legend
    from matplotlib.patches import Patch
    legend_elements = [
        Patch(facecolor='lightblue', label='◉ Start'),
        Patch(facecolor='gold', label='★ Goal'),
        Patch(facecolor='red', label='✖ Pit'),
        Patch(facecolor='orange', label='~ Quicksand'),
        Patch(facecolor='white', edgecolor='black', label='Empty')
    ]
    ax.legend(handles=legend_elements, loc='upper right', bbox_to_anchor=(1.35, 1))
    plt.tight_layout()
```

```
    plt.show()

# Run the plot with icons
plot_grid_with_icons(env)
```



Task 2: Define State Transition Function and Reward Function

State Transition Function

The agent moves on a 5×5 grid based on discrete actions:

- `'U'` = Up, `'D'` = Down, `'L'` = Left, `'R'` = Right

The environment enforces the following rules:

- Movement is **bounded by the grid** (cannot move outside).
- Each action deterministically leads to a new state (if within bounds).
- The new state is returned, along with the **corresponding reward** and a `done` flag if the episode ends.

Reward Function

Each grid cell has a type that determines the agent's reward:

Cell Type	Description	Reward	Terminal
Empty	Safe but costly	-0.5	✖
Quicksand	Non-terminal hazard	-5	✖
Pit	Deadly trap	-15	✅
Goal	Treasure (objective)	+100	✅

This setup rewards:

- **Risk-aware navigation**
- **Path length minimization**
- **Avoidance of hazardous shortcuts**

Table 1: Reward Function for Each Grid Cell Type

Cell Type	Symbol	Description	Reward	Terminal
Start	S	Starting cell	0	No

Goal	G	Treasure chest	+100	Yes
Pit	P	Deadly trap	-15	Yes
Quicksand	Q	Slows movement	-5	No
Empty Cell	–	Normal path	-0.5	No

Task 3: Set up Q-Learning Parameters and Policy

To train our pirate agent using Q-learning, we define a set of core hyperparameters and an exploration strategy that evolves over time.

Q-Learning Hyperparameters

- **Learning rate (α)**: Controls how quickly the agent updates its Q-values.
  - We set **α = 0.1** for moderately paced learning.
- **Discount factor (γ)**: Determines how much future rewards influence current decisions.
  - We use **γ = 0.9**, encouraging the agent to pursue long-term treasure over immediate gain.
- **Exploration rate (ε)**: Governs the trade-off between exploration and exploitation.
  - Instead of a fixed value, we implement **ε decay**, starting with **ε = 1.0** and reducing it gradually to **ε_min = 0.05** using a **decay rate of 0.995**.
  - This allows the agent to explore more during early episodes and exploit learned knowledge later in training.

These values will be tuned systematically in **Task 5** through grid search to observe their effects on learning performance.

ε-Greedy Policy with Decay

The agent selects actions using the ε-greedy strategy:

- With probability **ε**, it takes a **random action** (exploration),
- With probability **1−ε**, it selects the **action with the highest Q-value** (exploitation),
- Ties among best actions are broken randomly.

The value of **ε decays after each episode**, allowing the agent to shift from exploration to exploitation over time.

In [51]:
```
# Q-table initialization
Q = {
    (i, j): {a: 0.0 for a in ACTIONS}
    for i in range(GRID_HEIGHT)
    for j in range(GRID_WIDTH)
}

# Q-learning parameters
alpha = 0.1              # Learning rate
gamma = 0.9              # Discount factor
epsilon = 1.0            # Initial exploration rate
epsilon_decay = 0.995    # Decay rate per episode
epsilon_min = 0.05       # Minimum exploration

def choose_action(state, Q, epsilon):
    if np.random.rand() < epsilon:
        return random.choice(ACTIONS)
    else:
        q_vals = Q[state]
        max_q = max(q_vals.values())
        best_actions = [a for a, v in q_vals.items() if v == max_q]
        return random.choice(best_actions)
```

Task 4: Run the Q-learning Algorithm and Represent Its Performance

We now train our Q-learning agent using the baseline configuration:

- **α = 0.1** (learning rate)
- **γ = 0.9** (discount factor)
- **ε = 1.0** initially, with **decay rate = 0.995** down to a **minimum of 0.05**

Q-learning Training Loop

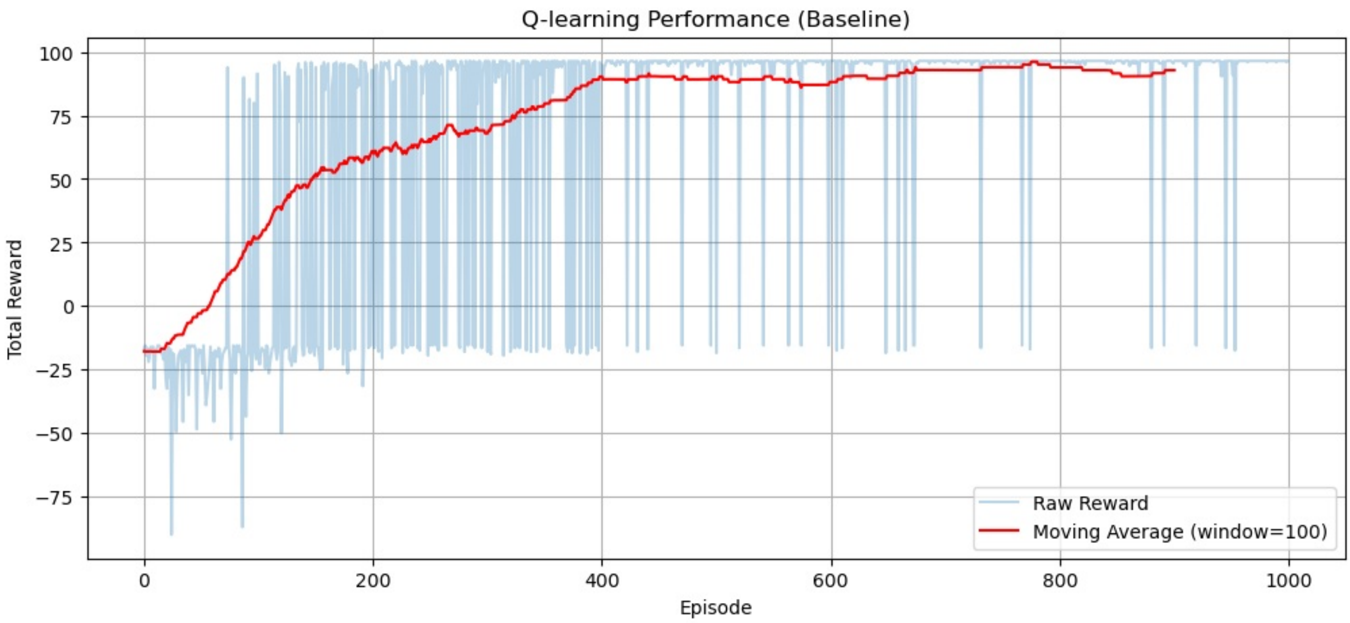
For each episode:

1. The agent starts from the initial state,
2. It selects actions using the **ε-greedy strategy** with decaying ε,
3. Q-values are updated using the Bellman equation,
4. The total reward is recorded,
5. ε is decayed at the end of each episode.

We train the agent over **1,000 episodes** and visualize learning using a **moving average** of total rewards to assess convergence and policy quality.

In [52]:
```
def run_q_learning(alpha, gamma, epsilon=1.0, epsilon_decay=0.995, epsilon_min=0.05, episodes=500, max_steps=10(
    Q_local = {
        (i, j): {a: 0.0 for a in ACTIONS}
        for i in range(GRID_HEIGHT)
        for j in range(GRID_WIDTH)
    }
    rewards = []

    for ep in range(episodes):
        state = env.reset()
        total_reward = 0

        for _ in range(max_steps):
            action = choose_action(state, Q_local, epsilon)
            next_state, reward, done = env.step(action)

            # Q-learning update rule
            old_q = Q_local[state][action]
            next_max = max(Q_local[next_state].values())
            Q_local[state][action] = old_q + alpha * (reward + gamma * next_max - old_q)

            state = next_state
            total_reward += reward
            if done:
                break

        rewards.append(total_reward)

        # ε decay
        if epsilon > epsilon_min:
            epsilon = max(epsilon * epsilon_decay, epsilon_min)

    return Q_local, rewards
```

In [53]:
```
# Run baseline experiment
Q_baseline, rewards_baseline = run_q_learning(
    alpha=0.1,
    gamma=0.9,
    epsilon=1.0,
    epsilon_decay=0.995,
    epsilon_min=0.05,
    episodes=1000
)

# Moving average
def moving_avg(data, window=100):
    return np.convolve(data, np.ones(window)/window, mode='valid')

# Plot learning curve
plt.figure(figsize=(12, 5))
plt.plot(rewards_baseline, alpha=0.3, label='Raw Reward')
plt.plot(moving_avg(rewards_baseline), label='Moving Average (window=100)', color='red')
plt.title("Q-learning Performance (Baseline)")
plt.xlabel("Episode")
plt.ylabel("Total Reward")
plt.grid(True)
plt.legend()
plt.show()
```



Task 5: Experiment with Different Parameter Values and Policies

We now conduct a structured hyperparameter search to evaluate how different learning settings and action selection policies affect the agent's performance.

ε-Greedy Policy (with ε-decay)

We perform a full grid search over the following parameters:

- **Learning rate (α)** ∈ {0.1, 0.3, 0.5}
- **Discount factor (γ)** ∈ {0.7, 0.9}
- **Initial exploration rate (ε)** ∈ {0.1, 0.3, 0.5}

This results in **18 configurations**. Instead of using a fixed ε, we apply an **ε-decay strategy**:

- ε is initialized to one of the values above,
- It decays after each episode using a decay factor of **0.995**,
- It reaches a **minimum threshold ε_min = 0.05**.

This allows the agent to explore aggressively early in training and exploit more confidently later.

Softmax (Boltzmann) Policy

We also experiment with a **temperature-based softmax policy**, which selects actions probabilistically based on their Q-values:

- **Learning rate (α)** ∈ {0.1, 0.3}
- **Discount factor (γ)** ∈ {0.7, 0.9}
- **Temperature (τ)** ∈ {0.5, 1.0, 2.0}

This results in **12 configurations**.

Softmax does not use ε. Instead, **τ controls exploration**:

- High τ → more exploration (flatter probabilities),
- Low τ → greedier decisions (sharper probabilities).

Experiment Setup

- Each configuration is trained for **1000 episodes**,
- We record the **total reward per episode**,
- We compute the **average reward and standard deviation** over the **last 100 episodes**,
- We compare the **top 5 runs** of each policy type based on final average reward,
- We visualize the learning curves to assess convergence and stability.

Results Breakdown

We divide results into two groups for comparison:

- **ε-Greedy Results**: top configurations using decaying ε
- **Softmax Results**: top configurations using τ-controlled exploration

This structure allows us to clearly interpret how each exploration strategy behaves across hyperparameter settings.

In [77]:
```
import time
```

Softmax Action Policy

In [78]:
```
import math

def softmax_action_selection(state, Q, temperature=1.0):
    q_vals = Q[state]
    max_q = max(q_vals.values())  # for numerical stability
    exp_q = {a: math.exp((q - max_q) / temperature) for a, q in q_vals.items()}
    sum_exp = sum(exp_q.values())
    probs = [exp_q[a] / sum_exp for a in ACTIONS]
    return np.random.choice(ACTIONS, p=probs)
```

Q-learning Runner

In [79]:
```
def run_q_learning(alpha, gamma, epsilon=None, temperature=None, use_softmax=False,
                   epsilon_decay=None, epsilon_min=0.05,
                   episodes=1000, max_steps=100):
    Q_local = {(i, j): {a: 0.0 for a in ACTIONS}
               for i in range(GRID_HEIGHT) for j in range(GRID_WIDTH)}
    rewards = []

    for ep in range(episodes):
        state = env.reset()
        total_reward = 0

        for _ in range(max_steps):
            if use_softmax:
                action = softmax_action_selection(state, Q_local, temperature)
            else:
                if np.random.rand() < epsilon:
                    action = random.choice(ACTIONS)
                else:
                    max_q = max(Q_local[state].values())
                    best_actions = [a for a, v in Q_local[state].items() if v == max_q]
                    action = random.choice(best_actions)

            next_state, reward, done = env.step(action)

            # Q update
            old_q = Q_local[state][action]
            next_max = max(Q_local[next_state].values())
            Q_local[state][action] = old_q + alpha * (reward + gamma * next_max - old_q)

            state = next_state
            total_reward += reward
            if done:
                break

        rewards.append(total_reward)

        # Apply ε decay only if using ε-greedy
        if not use_softmax and epsilon_decay is not None:
            epsilon = max(epsilon * epsilon_decay, epsilon_min)

    return Q_local, rewards
```

Run All Configurations

In [80]:
```
# ε-Greedy configurations
alphas = [0.1, 0.3, 0.5]
gammas = [0.7, 0.9]
epsilons = [0.1, 0.3, 0.5]

# Softmax configurations
temperatures = [0.5, 1.0, 2.0]
softmax_alphas = [0.1, 0.3]
softmax_gammas = [0.7, 0.9]

# Storage
```

```python
results = []
curves = {}

# ε-Greedy runs
for alpha in alphas:
    for gamma in gammas:
        for epsilon in epsilons:
            label = f"ε-greedy α={alpha}, γ={gamma}, ε={epsilon}"
            start = time.time()
            _, rewards = run_q_learning(alpha, gamma, epsilon=epsilon, epsilon_decay=0.995, epsilon_min=0.05)
            duration = time.time() - start
            avg = np.mean(rewards[-100:])
            std = np.std(rewards[-100:])
            results.append({'label': label, 'alpha': alpha, 'gamma': gamma, 'epsilon': epsilon,
                            'policy': 'ε-greedy', 'avg_reward': avg, 'std_dev': std, 'duration_sec': duration})
            curves[label] = rewards

# Softmax runs
for alpha in softmax_alphas:
    for gamma in softmax_gammas:
        for tau in temperatures:
            label = f"Softmax α={alpha}, γ={gamma}, τ={tau}"
            start = time.time()
            _, rewards = run_q_learning(alpha, gamma, use_softmax=True, temperature=tau)
            duration = time.time() - start
            avg = np.mean(rewards[-100:])
            std = np.std(rewards[-100:])
            results.append({'label': label, 'alpha': alpha, 'gamma': gamma, 'temperature': tau,
                            'policy': 'softmax', 'avg_reward': avg, 'std_dev': std, 'duration_sec': duration})
            curves[label] = rewards
```

Summary Table

In [88]:

```python
df_all = pd.DataFrame(results)
df_all_sorted = df_all.sort_values(by="avg_reward", ascending=False).reset_index(drop=True)
df_all_sorted.head(10)  # show top 10 configurations
```

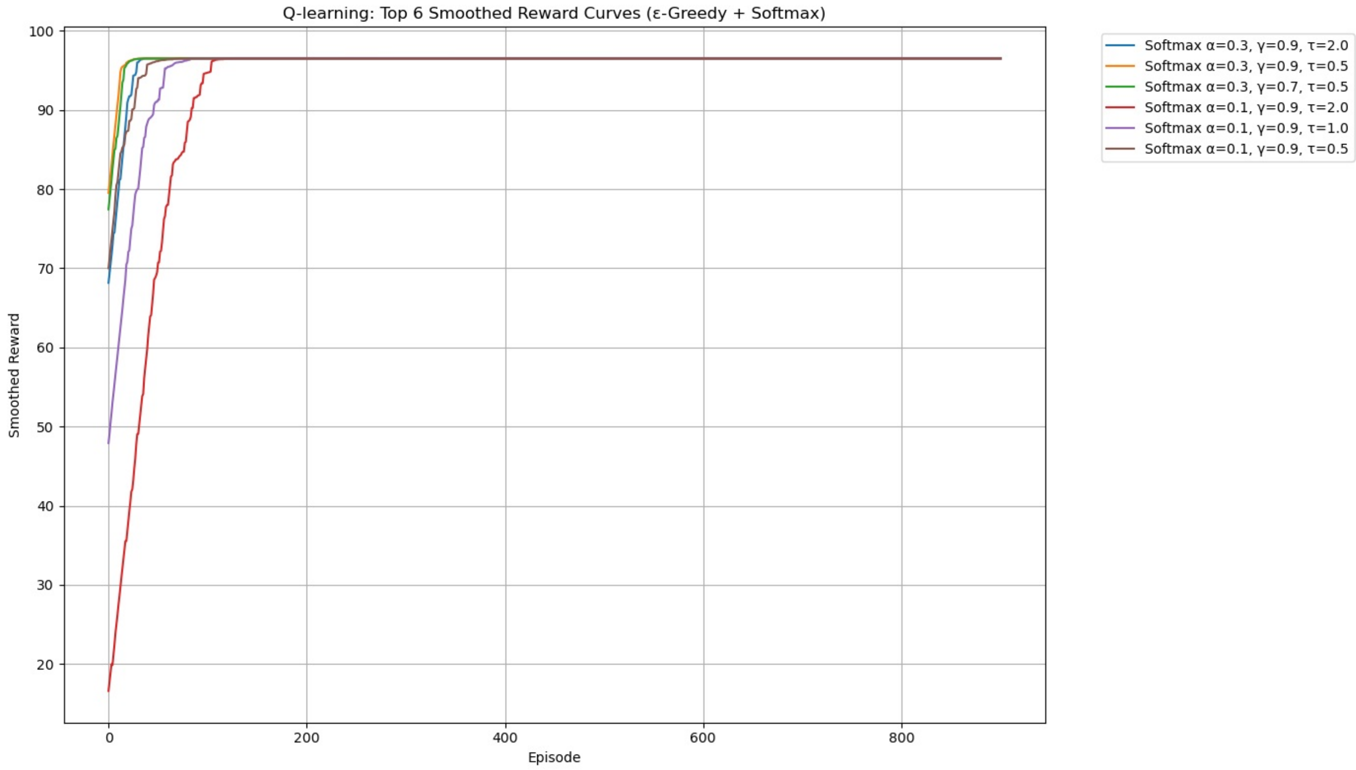
Out[88]:

	label	alpha	gamma	epsilon	policy	avg_reward	std_dev	duration_sec	temperature
0	Softmax α=0.3, γ=0.9, τ=2.0	0.3	0.9	NaN	softmax	96.500	0.000000	0.338483	2.0
1	Softmax α=0.3, γ=0.9, τ=0.5	0.3	0.9	NaN	softmax	96.500	0.000000	0.307287	0.5
2	Softmax α=0.3, γ=0.7, τ=0.5	0.3	0.7	NaN	softmax	96.500	0.000000	0.262045	0.5
3	Softmax α=0.1, γ=0.9, τ=2.0	0.1	0.9	NaN	softmax	96.500	0.000000	0.297199	2.0
4	Softmax α=0.1, γ=0.9, τ=1.0	0.1	0.9	NaN	softmax	96.500	0.000000	0.265563	1.0
5	Softmax α=0.1, γ=0.9, τ=0.5	0.1	0.9	NaN	softmax	96.500	0.000000	0.262385	0.5
6	Softmax α=0.1, γ=0.7, τ=1.0	0.1	0.7	NaN	softmax	96.500	0.000000	0.264041	1.0
7	Softmax α=0.3, γ=0.9, τ=1.0	0.3	0.9	NaN	softmax	96.500	0.000000	0.320457	1.0
8	Softmax α=0.1, γ=0.7, τ=0.5	0.1	0.7	NaN	softmax	96.500	0.000000	0.249940	0.5
9	Softmax α=0.3, γ=0.7, τ=1.0	0.3	0.7	NaN	softmax	96.465	0.127574	0.264058	1.0

Plot Best Learning Curves

In [82]:

```python
plt.figure(figsize=(14, 8))
top_labels = df_all_sorted['label'].head(6)

for label in top_labels:
    smoothed = moving_avg(curves[label])
    plt.plot(smoothed, label=label)

plt.title("Q-learning: Top 6 Smoothed Reward Curves (ε-Greedy + Softmax)")
plt.xlabel("Episode")
plt.ylabel("Smoothed Reward")
plt.grid(True)
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left')
plt.tight_layout()
plt.show()
```



Separate Comparison of ε-Greedy and Softmax Policies

To better understand the impact of different exploration strategies, we divide the experiments into two groups:

ε-Greedy Policy

- Grid search over:
  - α ∈ {0.1, 0.3, 0.5}
  - γ ∈ {0.7, 0.9}
  - ε ∈ {0.1, 0.3, 0.5}
- Total combinations: 18

Softmax Policy

- Grid search over:
  - α ∈ {0.1, 0.3}
  - γ ∈ {0.7, 0.9}
  - τ ∈ {0.5, 1.0, 2.0}
- Total combinations: 12

We now present their results **independently** to compare performance, convergence, and stability within each policy family.

In [83]:
```
# Split results
df_eps = df_all[df_all['policy'] == 'ε-greedy'].sort_values(by='avg_reward', ascending=False)
df_soft = df_all[df_all['policy'] == 'softmax'].sort_values(by='avg_reward', ascending=False)

# Show top 5 of each
print("Top ε-Greedy Configurations:")
display(df_eps.head(5))

print("Top Softmax Configurations:")
display(df_soft.head(5))
```

```
Top ε-Greedy Configurations:
```

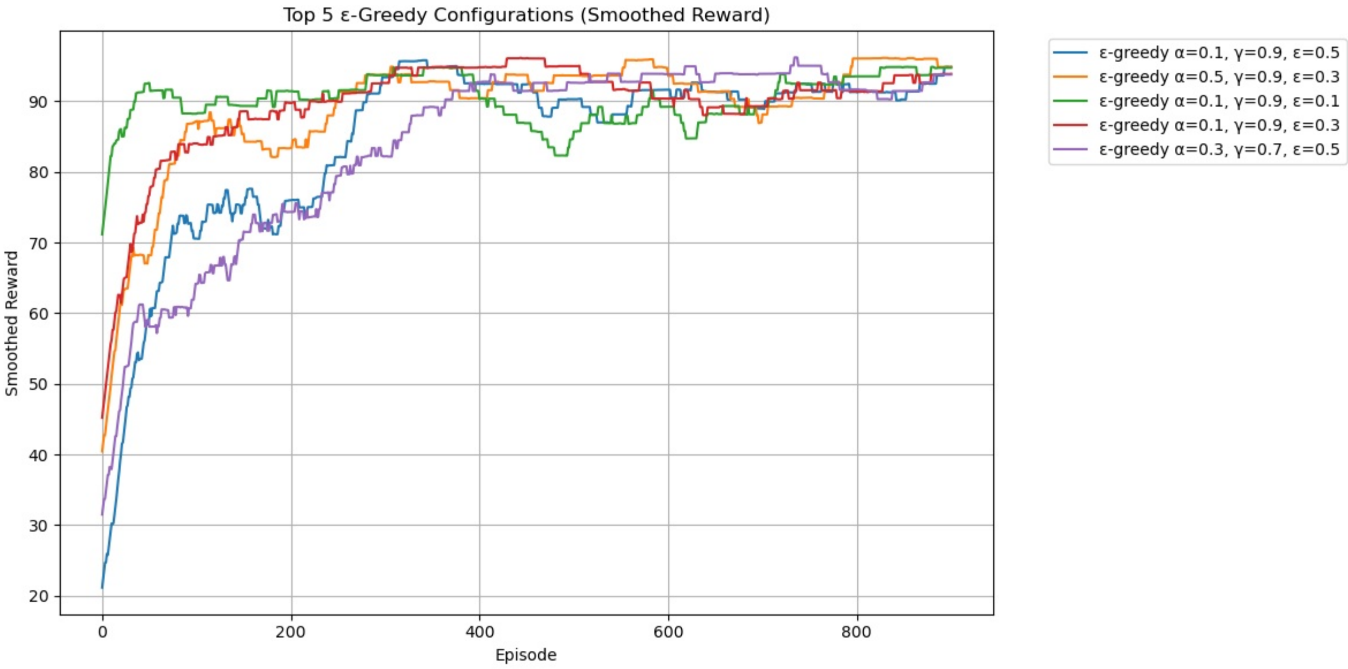
	label	alpha	gamma	epsilon	policy	avg_reward	std_dev	duration_sec	temperature
5	ε-greedy α=0.1, γ=0.9, ε=0.5	0.1	0.9	0.5	ε-greedy	94.875	11.325938	0.017785	NaN
16	ε-greedy α=0.5, γ=0.9, ε=0.3	0.5	0.9	0.3	ε-greedy	94.860	11.442264	0.031243	NaN
3	ε-greedy α=0.1, γ=0.9, ε=0.1	0.1	0.9	0.1	ε-greedy	94.745	11.453274	0.028100	NaN
4	ε-greedy α=0.1, γ=0.9, ε=0.3	0.1	0.9	0.3	ε-greedy	93.885	15.704116	0.028958	NaN
8	ε-greedy α=0.3, γ=0.7, ε=0.5	0.3	0.7	0.5	ε-greedy	93.780	15.769483	0.031250	NaN

```
Top Softmax Configurations:
```

	label	alpha	gamma	epsilon	policy	avg_reward	std_dev	duration_sec	temperature
18	Softmax α=0.1, γ=0.7, τ=0.5	0.1	0.7	NaN	softmax	96.5	0.0	0.249940	0.5
19	Softmax α=0.1, γ=0.7, τ=1.0	0.1	0.7	NaN	softmax	96.5	0.0	0.264041	1.0
21	Softmax α=0.1, γ=0.9, τ=0.5	0.1	0.9	NaN	softmax	96.5	0.0	0.262385	0.5
22	Softmax α=0.1, γ=0.9, τ=1.0	0.1	0.9	NaN	softmax	96.5	0.0	0.265563	1.0
23	Softmax α=0.1, γ=0.9, τ=2.0	0.1	0.9	NaN	softmax	96.5	0.0	0.297199	2.0

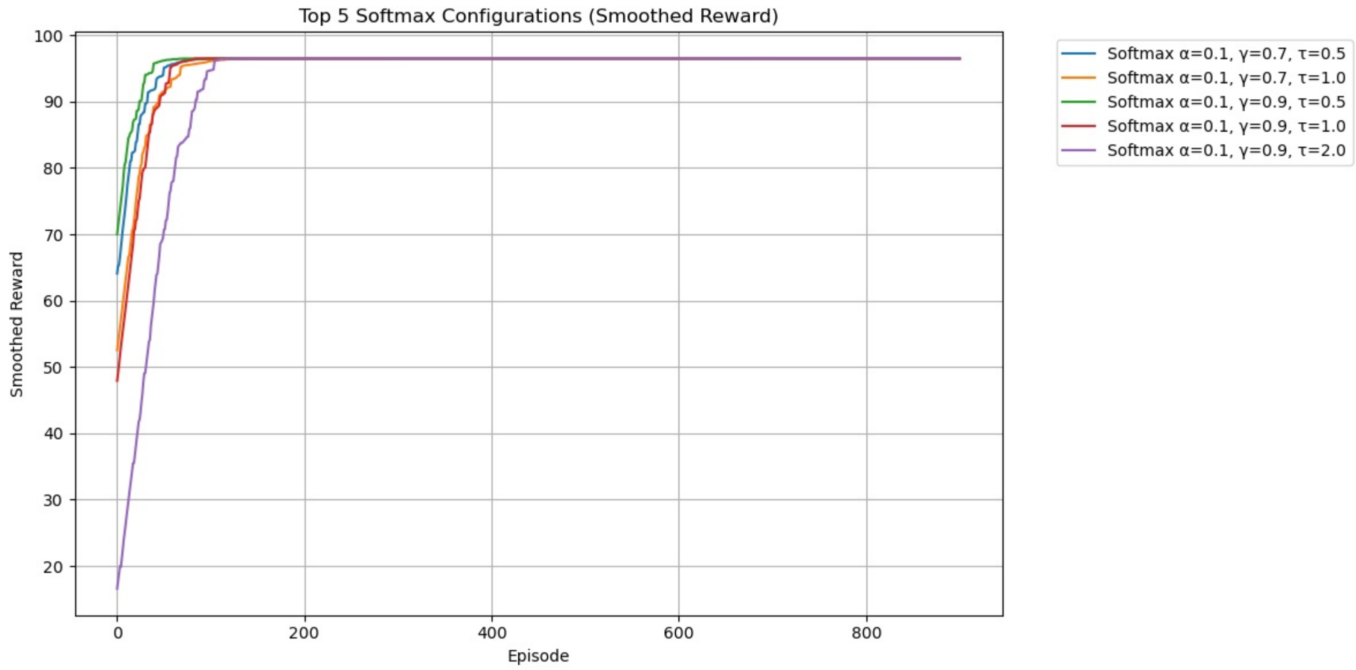
In [84]:

```
plt.figure(figsize=(12, 6))
for label in df_eps['label'].head(5):
    plt.plot(moving_avg(curves[label]), label=label)

plt.title("Top 5 ε-Greedy Configurations (Smoothed Reward)")
plt.xlabel("Episode")
plt.ylabel("Smoothed Reward")
plt.grid(True)
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left')
plt.tight_layout()
plt.show()
```



In [85]:

```
plt.figure(figsize=(12, 6))
for label in df_soft['label'].head(5):
    plt.plot(moving_avg(curves[label]), label=label)

plt.title("Top 5 Softmax Configurations (Smoothed Reward)")
plt.xlabel("Episode")
plt.ylabel("Smoothed Reward")
plt.grid(True)
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left')
plt.tight_layout()
plt.show()
```



Task 6: Analyze the Results Quantitatively and Qualitatively

We now analyze the learning outcomes from our grid search across 30 configurations: 18 using ε-greedy (with ε-decay) and 12 using the softmax (Boltzmann) policy.

In [86]:
```
from IPython.display import display

print("Top 3 ε-Greedy Configurations:")
display(df_eps.head(3)[['label', 'avg_reward', 'std_dev', 'duration_sec']])

print("Top 3 Softmax Configurations:")
display(df_soft.head(3)[['label', 'avg_reward', 'std_dev', 'duration_sec']])
```

```
Top 3 ε-Greedy Configurations:
```

	label	avg_reward	std_dev	duration_sec
5	ε-greedy α=0.1, γ=0.9, ε=0.5	94.875	11.325938	0.017785
16	ε-greedy α=0.5, γ=0.9, ε=0.3	94.860	11.442264	0.031243
3	ε-greedy α=0.1, γ=0.9, ε=0.1	94.745	11.453274	0.028100

```
Top 3 Softmax Configurations:
```

	label	avg_reward	std_dev	duration_sec
18	Softmax α=0.1, γ=0.7, τ=0.5	96.5	0.0	0.249940
19	Softmax α=0.1, γ=0.7, τ=1.0	96.5	0.0	0.264041
21	Softmax α=0.1, γ=0.9, τ=0.5	96.5	0.0	0.262385

In [ ]:

Task 7: Double DQN with Prioritized Experience Replay on LunarLander-v2

In this task, we implement a Deep Q-Network (DQN) with two improvements:

- **Double Q-learning** to reduce overestimation bias in action-value estimates.
- **Prioritized Experience Replay (PER)** to sample transitions based on TD error. We apply this to the `LunarLander-v2` environment from OpenAI Gym, which has discrete actions and shaped rewards, making it ideal for value-based methods.

In [1]:
```
import gym
import numpy as np
import random
from collections import deque, namedtuple
import torch
import torch.nn as nn
import torch.optim as optim
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings("ignore", category=DeprecationWarning)

# Set seed
SEED = 42
random.seed(SEED)
np.random.seed(SEED)
torch.manual_seed(SEED)

# Create environment with updated API
env = gym.make('LunarLander-v2')
env.reset(seed=SEED)
env.action_space.seed(SEED)

# Observation and action sizes
state_dim = env.observation_space.shape[0]
n_actions = env.action_space.n

print(f"State dimension: {state_dim}")
print(f"Number of actions: {n_actions}")
```

```
State dimension: 8
Number of actions: 4
```

Q-Network Definition (Simple MLP)

In [2]:
```
class QNetwork(nn.Module):
    def __init__(self, state_dim, action_dim):
        super(QNetwork, self).__init__()
        self.net = nn.Sequential(
            nn.Linear(state_dim, 128),
            nn.ReLU(),
            nn.Linear(128, 128),
            nn.ReLU(),
            nn.Linear(128, action_dim)
        )

    def forward(self, x):
        return self.net(x)
```

Environment and Q-Network

We define our Q-network as a simple multi-layer perceptron with two hidden layers of 128 units each. The output layer produces Q-values for all possible actions. We also set the random seed for reproducibility and inspect the shape of the environment.

Prioritized Experience Replay (PER)

In standard DQN, experiences are sampled uniformly from the replay buffer. This can be inefficient because many transitions are redundant or uninformative.

Prioritized Experience Replay addresses this by:

- Sampling transitions with probability proportional to their **temporal-difference (TD) error**,
- Storing and updating priorities to ensure **important experiences** (e.g., large errors) are replayed more often.

We implement **proportional prioritization**, where each transition is sampled with probability:

[P(i) = \frac{p_i^\alpha}{\sum_k p_k^\alpha}] where (p_i) is the priority and (\alpha) determines the level of prioritization (α = 0 is uniform sampling).

In [3]:
```
class PrioritizedReplayBuffer:
    def __init__(self, capacity, alpha=0.6):
        self.capacity = capacity
        self.buffer = []
        self.priorities = np.zeros((capacity,), dtype=np.float32)
        self.alpha = alpha
        self.position = 0
        self.Transition = namedtuple('Transition', ('state', 'action', 'reward', 'next_state', 'done'))

    def push(self, state, action, reward, next_state, done):
        max_prio = self.priorities.max() if self.buffer else 1.0
        transition = self.Transition(state, action, reward, next_state, done)

        if len(self.buffer) < self.capacity:
            self.buffer.append(transition)
        else:
            self.buffer[self.position] = transition

        self.priorities[self.position] = max_prio
        self.position = (self.position + 1) % self.capacity

    def sample(self, batch_size, beta=0.4):
        if len(self.buffer) == self.capacity:
            prios = self.priorities
        else:
            prios = self.priorities[:self.position]

        probs = prios ** self.alpha
        probs /= probs.sum()

        indices = np.random.choice(len(self.buffer), batch_size, p=probs)
        samples = [self.buffer[idx] for idx in indices]

        # Importance-sampling weights
        total = len(self.buffer)
        weights = (total * probs[indices]) ** (-beta)
        weights /= weights.max()  # normalize

        batch = self.Transition(*zip(*samples))
        return batch, indices, torch.tensor(weights, dtype=torch.float32)

    def update_priorities(self, indices, priorities):
        for idx, prio in zip(indices, priorities):
            self.priorities[idx] = prio
```

We implement a prioritized buffer where transitions with high TD error are sampled more frequently. The buffer stores a priority score for each transition and uses it to compute sampling probabilities. We also calculate **importance sampling weights** to correct for this bias during gradient updates.

DQN Agent with Double Q-learning

In standard DQN, we estimate the target Q-value using the maximum Q-value from the target network:

[y = r + \gamma \cdot \max_{a'} Q_{\text{target}}(s', a')]

This can lead to **overestimation bias**.
In **Double Q-learning**, we use the main network to select the best action and the target network to evaluate it:

[y = r + \gamma \cdot Q_{\text{target}}(s', \arg\max_{a'} Q_{\text{main}}(s', a'))]

This decouples action selection and evaluation.

In [4]:
```
class DQNAgent:
    def __init__(self, state_dim, action_dim, buffer, gamma=0.99, lr=1e-3, batch_size=64,
                 beta_start=0.4, use_per=True, use_double=True):
        self.q_net = QNetwork(state_dim, action_dim)
        self.target_net = QNetwork(state_dim, action_dim)
        self.target_net.load_state_dict(self.q_net.state_dict())
        self.target_net.eval()

        self.optimizer = optim.Adam(self.q_net.parameters(), lr=lr)
        self.replay_buffer = buffer
        self.gamma = gamma
        self.batch_size = batch_size
        self.beta = beta_start
        self.loss_fn = nn.MSELoss()
        self.action_dim = action_dim
        self.step_count = 0
```

```
        self.use_per = use_per
        self.use_double = use_double

    def act(self, state, epsilon):
        if random.random() < epsilon:
            return random.randint(0, self.action_dim - 1)
        state = torch.from_numpy(np.array(state)).float().unsqueeze(0)
        with torch.no_grad():
            q_values = self.q_net(state)
        return q_values.argmax().item()

    def update(self):
        if len(self.replay_buffer.buffer) < self.batch_size:
            return None

        self.beta = min(1.0, self.beta + 1e-4)

        if self.use_per:
            batch, indices, weights = self.replay_buffer.sample(self.batch_size, beta=self.beta)
        else:
            transitions = random.sample(self.replay_buffer.buffer, self.batch_size)
            Transition = self.replay_buffer.Transition
            batch = Transition(*zip(*transitions))
            weights = torch.ones(self.batch_size, 1, dtype=torch.float32)

        states = torch.tensor(np.array(batch.state), dtype=torch.float32)
        actions = torch.tensor(batch.action).unsqueeze(1)
        rewards = torch.tensor(batch.reward, dtype=torch.float32).unsqueeze(1)
        next_states = torch.tensor(np.array(batch.next_state), dtype=torch.float32)
        dones = torch.tensor(batch.done, dtype=torch.float32).unsqueeze(1)
        weights = weights.unsqueeze(1)

        # Q values of current state-action pairs
        q_values = self.q_net(states).gather(1, actions)

        # Compute target values
        with torch.no_grad():
            if self.use_double:
                next_actions = self.q_net(next_states).argmax(1, keepdim=True)
                next_q = self.target_net(next_states).gather(1, next_actions)
            else:
                next_q = self.target_net(next_states).max(1, keepdim=True)[0]
            targets = rewards + self.gamma * (1 - dones) * next_q

        # TD error (for PER only)
        if self.use_per:
            td_errors = (q_values.detach() - targets.detach()).abs().cpu().numpy().flatten()
            self.replay_buffer.update_priorities(indices, td_errors + 1e-6)

        # Loss with importance sampling (or uniform weights)
        loss = (weights * (q_values - targets).pow(2)).mean()

        self.optimizer.zero_grad()
        loss.backward()
        self.optimizer.step()

        return loss.item()


    def update_target_network(self):
        self.target_net.load_state_dict(self.q_net.state_dict())
```

We now have a DQN agent that:

- Uses a main and target network for Double Q-learning,
- Samples and updates transitions based on their TD error (for PER),
- Applies importance sampling weights during updates,
- Slowly increases the `beta` parameter to correct PER bias over time.

Training the Agent

We now train the Double DQN agent with Prioritized Experience Replay. At each step:

1. The agent selects an action using ε-greedy exploration.
2. The environment returns the next state and reward.
3. The transition is stored in the PER buffer.
4. The agent updates its Q-network using sampled experiences.
5. Every `target_update_freq` steps, the target network is updated.

In [5]:
```
def train(agent, env, episodes=500, epsilon_start=1.0, epsilon_end=0.05, epsilon_decay=0.995,
          target_update_freq=10, max_steps=1000):

    epsilon = epsilon_start
    episode_rewards = []
    losses = []

    for episode in range(episodes):
        result = env.reset()
        state = result[0] if isinstance(result, tuple) else result
        total_reward = 0

        for step in range(max_steps):
            action = agent.act(state, epsilon)
            step_result = env.step(action)
            if len(step_result) == 5:
                next_state, reward, terminated, truncated, _ = step_result
                done = terminated or truncated
            else:
                next_state, reward, done, _ = step_result


            agent.replay_buffer.push(state, action, reward, next_state, done)
            loss = agent.update()
            state = next_state
            total_reward += reward

            if done:
                break

        episode_rewards.append(total_reward)
        if loss is not None:
            losses.append(loss)

        epsilon = max(epsilon_end, epsilon * epsilon_decay)

        if episode % target_update_freq == 0:
            agent.update_target_network()

        if episode % 10 == 0:
            print(f"Episode {episode:4d} | Reward: {total_reward:.2f} | Epsilon: {epsilon:.3f}")

    return episode_rewards, losses
```

We now initialize the agent and replay buffer, then train the agent for 500 episodes.

In [6]:
```
# Initialize replay buffer and agent
# Replay buffer choice
buffer_per = PrioritizedReplayBuffer(100_000)
buffer_uniform = PrioritizedReplayBuffer(100_000)  # will ignore priorities

# Vanilla DQN (no PER, no Double Q)
agent_dqn = DQNAgent(state_dim, n_actions, buffer_uniform, use_per=False, use_double=False)

# DQN + Double Q-learning only
agent_double = DQNAgent(state_dim, n_actions, buffer_uniform, use_per=False, use_double=True)

# DQN + PER only
agent_per = DQNAgent(state_dim, n_actions, buffer_per, use_per=True, use_double=False)

# DQN + Double + PER
agent_full = DQNAgent(state_dim, n_actions, buffer_per, use_per=True, use_double=True)
```

Helper Function to Train & Record Metrics

In [7]:
```
import time

def evaluate_agent(agent, label, episodes=500):
    print(f"Training: {label}")
    start = time.time()
    rewards, losses = train(agent, env, episodes=episodes)
    duration = time.time() - start
    avg_reward = np.mean(rewards[-100:])
    std_reward = np.std(rewards[-100:])
    return {
        'label': label,
        'rewards': rewards,
        'losses': losses,
        'avg_reward': avg_reward,
        'std_dev': std_reward,
        'duration_sec': duration
```

```
    }
```

Run All Configurations

In [8]:

```python
results = []

# Replay buffers
buffer_uniform_1 = PrioritizedReplayBuffer(100_000)
buffer_uniform_2 = PrioritizedReplayBuffer(100_000)
buffer_per_1 = PrioritizedReplayBuffer(100_000)
buffer_per_2 = PrioritizedReplayBuffer(100_000)

# Configurations
configs = [
    {"label": "Vanilla DQN", "per": False, "double": False, "buffer": buffer_uniform_1},
    {"label": "DQN + Double Q", "per": False, "double": True, "buffer": buffer_uniform_2},
    {"label": "DQN + PER", "per": True, "double": False, "buffer": buffer_per_1},
    {"label": "DQN + Double + PER", "per": True, "double": True, "buffer": buffer_per_2},
]

# Train and evaluate each config
for cfg in configs:
    agent = DQNAgent(state_dim, n_actions, cfg['buffer'], use_per=cfg['per'], use_double=cfg['double'])
    result = evaluate_agent(agent, cfg['label'], episodes=500)
    results.append(result)
```

```
Training: Vanilla DQN
Episode    0 | Reward: -225.61 | Epsilon: 0.995
Episode   10 | Reward: -90.47 | Epsilon: 0.946
Episode   20 | Reward: -393.43 | Epsilon: 0.900
Episode   30 | Reward: -14.35 | Epsilon: 0.856
Episode   40 | Reward: -96.28 | Epsilon: 0.814
Episode   50 | Reward: -115.53 | Epsilon: 0.774
Episode   60 | Reward: -157.32 | Epsilon: 0.737
Episode   70 | Reward: -181.34 | Epsilon: 0.701
Episode   80 | Reward: -184.01 | Epsilon: 0.666
Episode   90 | Reward: -45.91 | Epsilon: 0.634
Episode  100 | Reward: -149.66 | Epsilon: 0.603
Episode  110 | Reward: -130.41 | Epsilon: 0.573
Episode  120 | Reward: -374.32 | Epsilon: 0.545
Episode  130 | Reward: -62.96 | Epsilon: 0.519
Episode  140 | Reward: -4.84 | Epsilon: 0.493
Episode  150 | Reward: -44.99 | Epsilon: 0.469
Episode  160 | Reward: -30.43 | Epsilon: 0.446
Episode  170 | Reward: -56.66 | Epsilon: 0.424
Episode  180 | Reward: -28.40 | Epsilon: 0.404
Episode  190 | Reward: -118.39 | Epsilon: 0.384
Episode  200 | Reward: 46.64 | Epsilon: 0.365
Episode  210 | Reward: -51.19 | Epsilon: 0.347
Episode  220 | Reward: -239.74 | Epsilon: 0.330
Episode  230 | Reward: -21.40 | Epsilon: 0.314
Episode  240 | Reward: -106.57 | Epsilon: 0.299
Episode  250 | Reward: -27.75 | Epsilon: 0.284
Episode  260 | Reward: -49.97 | Epsilon: 0.270
Episode  270 | Reward: -5.73 | Epsilon: 0.257
Episode  280 | Reward: -54.33 | Epsilon: 0.245
Episode  290 | Reward: 0.57 | Epsilon: 0.233
Episode  300 | Reward: 3.19 | Epsilon: 0.221
Episode  310 | Reward: -144.56 | Epsilon: 0.210
Episode  320 | Reward: -41.29 | Epsilon: 0.200
Episode  330 | Reward: -152.47 | Epsilon: 0.190
Episode  340 | Reward: 6.44 | Epsilon: 0.181
Episode  350 | Reward: -29.32 | Epsilon: 0.172
Episode  360 | Reward: -129.11 | Epsilon: 0.164
Episode  370 | Reward: -60.85 | Epsilon: 0.156
Episode  380 | Reward: -6.72 | Epsilon: 0.148
Episode  390 | Reward: -29.48 | Epsilon: 0.141
Episode  400 | Reward: 111.64 | Epsilon: 0.134
Episode  410 | Reward: 155.87 | Epsilon: 0.127
Episode  420 | Reward: -188.49 | Epsilon: 0.121
Episode  430 | Reward: -70.55 | Epsilon: 0.115
Episode  440 | Reward: -215.19 | Epsilon: 0.110
Episode  450 | Reward: -228.72 | Epsilon: 0.104
Episode  460 | Reward: -45.54 | Epsilon: 0.099
Episode  470 | Reward: -76.08 | Epsilon: 0.094
Episode  480 | Reward: -239.68 | Epsilon: 0.090
Episode  490 | Reward: -71.57 | Epsilon: 0.085
Training: DQN + Double Q
Episode    0 | Reward: -55.55 | Epsilon: 0.995
Episode   10 | Reward: -69.46 | Epsilon: 0.946
Episode   20 | Reward: -144.25 | Epsilon: 0.900
Episode   30 | Reward: -143.42 | Epsilon: 0.856
```

```
Episode   40 | Reward: -206.86 | Epsilon: 0.814
Episode   50 | Reward: -128.94 | Epsilon: 0.774
Episode   60 | Reward: -83.85 | Epsilon: 0.737
Episode   70 | Reward: -156.48 | Epsilon: 0.701
Episode   80 | Reward: -182.61 | Epsilon: 0.666
Episode   90 | Reward: -96.72 | Epsilon: 0.634
Episode  100 | Reward: -85.21 | Epsilon: 0.603
Episode  110 | Reward: -30.76 | Epsilon: 0.573
Episode  120 | Reward: -2.55 | Epsilon: 0.545
Episode  130 | Reward: -42.28 | Epsilon: 0.519
Episode  140 | Reward: 59.15 | Epsilon: 0.493
Episode  150 | Reward: 14.62 | Epsilon: 0.469
Episode  160 | Reward: 7.82 | Epsilon: 0.446
Episode  170 | Reward: 21.27 | Epsilon: 0.424
Episode  180 | Reward: -18.75 | Epsilon: 0.404
Episode  190 | Reward: -101.98 | Epsilon: 0.384
Episode  200 | Reward: 74.63 | Epsilon: 0.365
Episode  210 | Reward: 40.47 | Epsilon: 0.347
Episode  220 | Reward: -14.90 | Epsilon: 0.330
Episode  230 | Reward: -22.19 | Epsilon: 0.314
Episode  240 | Reward: -68.60 | Epsilon: 0.299
Episode  250 | Reward: -31.64 | Epsilon: 0.284
Episode  260 | Reward: -11.34 | Epsilon: 0.270
Episode  270 | Reward: -33.71 | Epsilon: 0.257
Episode  280 | Reward: -26.86 | Epsilon: 0.245
Episode  290 | Reward: 6.13 | Epsilon: 0.233
Episode  300 | Reward: -53.88 | Epsilon: 0.221
Episode  310 | Reward: 42.74 | Epsilon: 0.210
Episode  320 | Reward: -40.67 | Epsilon: 0.200
Episode  330 | Reward: 83.85 | Epsilon: 0.190
Episode  340 | Reward: 21.72 | Epsilon: 0.181
Episode  350 | Reward: -24.35 | Epsilon: 0.172
Episode  360 | Reward: -14.31 | Epsilon: 0.164
Episode  370 | Reward: 63.85 | Epsilon: 0.156
Episode  380 | Reward: -30.13 | Epsilon: 0.148
Episode  390 | Reward: -58.29 | Epsilon: 0.141
Episode  400 | Reward: 41.41 | Epsilon: 0.134
Episode  410 | Reward: -42.21 | Epsilon: 0.127
Episode  420 | Reward: -511.38 | Epsilon: 0.121
Episode  430 | Reward: 201.39 | Epsilon: 0.115
Episode  440 | Reward: -37.48 | Epsilon: 0.110
Episode  450 | Reward: -32.45 | Epsilon: 0.104
Episode  460 | Reward: 259.41 | Epsilon: 0.099
Episode  470 | Reward: 10.92 | Epsilon: 0.094
Episode  480 | Reward: -66.85 | Epsilon: 0.090
Episode  490 | Reward: 23.06 | Epsilon: 0.085
Training: DQN + PER
Episode    0 | Reward: -242.87 | Epsilon: 0.995
Episode   10 | Reward: -108.00 | Epsilon: 0.946
Episode   20 | Reward: 42.81 | Epsilon: 0.900
Episode   30 | Reward: -68.98 | Epsilon: 0.856
Episode   40 | Reward: -44.23 | Epsilon: 0.814
Episode   50 | Reward: -232.44 | Epsilon: 0.774
Episode   60 | Reward: -20.46 | Epsilon: 0.737
Episode   70 | Reward: -294.37 | Epsilon: 0.701
Episode   80 | Reward: -111.61 | Epsilon: 0.666
Episode   90 | Reward: -71.66 | Epsilon: 0.634
Episode  100 | Reward: -60.85 | Epsilon: 0.603
Episode  110 | Reward: -0.98 | Epsilon: 0.573
Episode  120 | Reward: -178.81 | Epsilon: 0.545
Episode  130 | Reward: -79.91 | Epsilon: 0.519
Episode  140 | Reward: -68.29 | Epsilon: 0.493
Episode  150 | Reward: -250.59 | Epsilon: 0.469
Episode  160 | Reward: -46.90 | Epsilon: 0.446
Episode  170 | Reward: 2.26 | Epsilon: 0.424
Episode  180 | Reward: -59.65 | Epsilon: 0.404
Episode  190 | Reward: -5.64 | Epsilon: 0.384
Episode  200 | Reward: -87.50 | Epsilon: 0.365
Episode  210 | Reward: -126.37 | Epsilon: 0.347
Episode  220 | Reward: -23.17 | Epsilon: 0.330
Episode  230 | Reward: -61.63 | Epsilon: 0.314
Episode  240 | Reward: -101.19 | Epsilon: 0.299
Episode  250 | Reward: -94.69 | Epsilon: 0.284
Episode  260 | Reward: -219.62 | Epsilon: 0.270
Episode  270 | Reward: -3.76 | Epsilon: 0.257
Episode  280 | Reward: -38.06 | Epsilon: 0.245
Episode  290 | Reward: -49.59 | Epsilon: 0.233
Episode  300 | Reward: 6.42 | Epsilon: 0.221
Episode  310 | Reward: -64.41 | Epsilon: 0.210
Episode  320 | Reward: -41.55 | Epsilon: 0.200
Episode  330 | Reward: -38.30 | Epsilon: 0.190
Episode  340 | Reward: -101.31 | Epsilon: 0.181
Episode  350 | Reward: -114.31 | Epsilon: 0.172
```

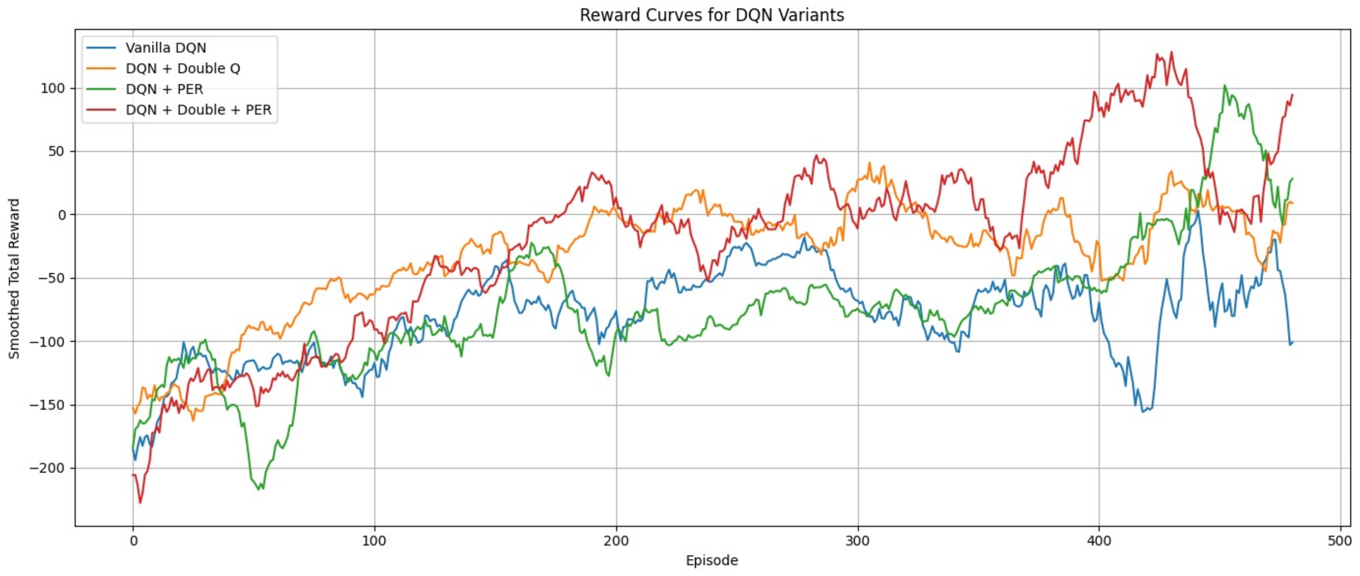
```
Episode  360 | Reward: -38.02 | Epsilon: 0.164
Episode  370 | Reward: -59.10 | Epsilon: 0.156
Episode  380 | Reward: -81.33 | Epsilon: 0.148
Episode  390 | Reward: -22.24 | Epsilon: 0.141
Episode  400 | Reward: -29.34 | Epsilon: 0.134
Episode  410 | Reward: -20.69 | Epsilon: 0.127
Episode  420 | Reward: -80.20 | Epsilon: 0.121
Episode  430 | Reward: 24.23 | Epsilon: 0.115
Episode  440 | Reward: -83.96 | Epsilon: 0.110
Episode  450 | Reward: -72.89 | Epsilon: 0.104
Episode  460 | Reward: -232.67 | Epsilon: 0.099
Episode  470 | Reward: -46.68 | Epsilon: 0.094
Episode  480 | Reward: -39.21 | Epsilon: 0.090
Episode  490 | Reward: -38.61 | Epsilon: 0.085
Training: DQN + Double + PER
Episode    0 | Reward: -107.96 | Epsilon: 0.995
Episode   10 | Reward: -80.97 | Epsilon: 0.946
Episode   20 | Reward: -105.92 | Epsilon: 0.900
Episode   30 | Reward: -173.77 | Epsilon: 0.856
Episode   40 | Reward: -170.73 | Epsilon: 0.814
Episode   50 | Reward: -98.72 | Epsilon: 0.774
Episode   60 | Reward: -107.54 | Epsilon: 0.737
Episode   70 | Reward: -350.15 | Epsilon: 0.701
Episode   80 | Reward: -148.97 | Epsilon: 0.666
Episode   90 | Reward: -67.37 | Epsilon: 0.634
Episode  100 | Reward: -99.56 | Epsilon: 0.603
Episode  110 | Reward: -12.50 | Epsilon: 0.573
Episode  120 | Reward: -93.28 | Epsilon: 0.545
Episode  130 | Reward: 32.71 | Epsilon: 0.519
Episode  140 | Reward: 12.64 | Epsilon: 0.493
Episode  150 | Reward: -24.71 | Epsilon: 0.469
Episode  160 | Reward: 60.08 | Epsilon: 0.446
Episode  170 | Reward: 72.93 | Epsilon: 0.424
Episode  180 | Reward: -35.48 | Epsilon: 0.404
Episode  190 | Reward: -7.71 | Epsilon: 0.384
Episode  200 | Reward: -17.07 | Epsilon: 0.365
Episode  210 | Reward: -34.83 | Epsilon: 0.347
Episode  220 | Reward: -189.06 | Epsilon: 0.330
Episode  230 | Reward: 121.90 | Epsilon: 0.314
Episode  240 | Reward: 0.39 | Epsilon: 0.299
Episode  250 | Reward: -127.72 | Epsilon: 0.284
Episode  260 | Reward: 2.77 | Epsilon: 0.270
Episode  270 | Reward: -116.97 | Epsilon: 0.257
Episode  280 | Reward: -57.42 | Epsilon: 0.245
Episode  290 | Reward: 30.24 | Epsilon: 0.233
Episode  300 | Reward: -187.06 | Epsilon: 0.221
Episode  310 | Reward: -70.27 | Epsilon: 0.210
Episode  320 | Reward: 142.02 | Epsilon: 0.200
Episode  330 | Reward: -36.11 | Epsilon: 0.190
Episode  340 | Reward: -55.29 | Epsilon: 0.181
Episode  350 | Reward: -38.52 | Epsilon: 0.172
Episode  360 | Reward: -20.78 | Epsilon: 0.164
Episode  370 | Reward: 127.08 | Epsilon: 0.156
Episode  380 | Reward: 152.50 | Epsilon: 0.148
Episode  390 | Reward: 201.27 | Epsilon: 0.141
Episode  400 | Reward: 130.25 | Epsilon: 0.134
Episode  410 | Reward: 132.98 | Epsilon: 0.127
Episode  420 | Reward: 189.60 | Epsilon: 0.121
Episode  430 | Reward: 233.00 | Epsilon: 0.115
Episode  440 | Reward: -18.33 | Epsilon: 0.110
Episode  450 | Reward: -22.36 | Epsilon: 0.104
Episode  460 | Reward: -129.38 | Epsilon: 0.099
Episode  470 | Reward: 214.60 | Epsilon: 0.094
Episode  480 | Reward: -163.49 | Epsilon: 0.090
Episode  490 | Reward: 39.16 | Epsilon: 0.085
```

We now visualize the learning progress over time using a moving average of total rewards.

Plot Smoothed Learning Curves

In [9]:
```python
def moving_average(values, window=20):
    return np.convolve(values, np.ones(window)/window, mode='valid')

plt.figure(figsize=(14, 6))
for res in results:
    smoothed = moving_average(res['rewards'], window=20)
    plt.plot(smoothed, label=f"{res['label']}")
plt.title("Reward Curves for DQN Variants")
plt.xlabel("Episode")
plt.ylabel("Smoothed Total Reward")
plt.grid(True)
plt.legend()
```

```
plt.tight_layout()
plt.show()
```



Tabulate Final Metrics

In [10]:

```
import pandas as pd

summary = pd.DataFrame([{
    'Label': r['label'],
    'Avg Reward (Last 100)': round(r['avg_reward'], 2),
    'Std Dev': round(r['std_dev'], 2),
    'Training Time (s)': round(r['duration_sec'], 2)
} for r in results])

summary
```

Out[10]:

	Label	Avg Reward (Last 100)	Std Dev	Training Time (s)
0	Vanilla DQN	-78.57	168.08	1130.60
1	DQN + Double Q	-9.93	117.21	1554.09
2	DQN + PER	9.92	135.86	2303.06
3	DQN + Double + PER	71.45	138.01	1632.62

Grid Search on DQN + Double Q + PER

To evaluate the effect of key hyperparameters, we perform a grid search on our best-performing DQN model with both **Double Q-learning** and **Prioritized Experience Replay**.

Hyperparameters investigated:

- **Learning Rate (lr)**: Controls how fast the Q-network updates — {1e-3, 5e-4, 1e-4}
- **Discount Factor (gamma)**: Importance of future rewards — {0.99, 0.95}
- **Epsilon Decay Rate (eps_decay)**: Controls exploration → exploitation — {0.99, 0.995}
- **Random Seed**: To test robustness across initializations — {42, 123}

Each configuration is trained for 500 episodes and evaluated on:

- Final average reward (last 100 episodes),
- Standard deviation,
- Training time (seconds).

In [12]:

```
from itertools import product
import time

# Hyperparameter grid
learning_rates = [1e-3, 5e-4, 1e-4]
gammas = [0.99, 0.95]
eps_decays = [0.99, 0.995]
seeds = [42, 123]

# Storage
grid_results = []
```

```python
# Run grid search
for lr, gamma, eps_decay, seed in product(learning_rates, gammas, eps_decays, seeds):
    print(f"\n Running config: lr={lr}, gamma={gamma}, eps_decay={eps_decay}, seed={seed}")

    # Set random seeds
    random.seed(seed)
    np.random.seed(seed)
    torch.manual_seed(seed)

    # Create fresh env and buffer
    env = gym.make('LunarLander-v2')
    env.reset(seed=seed)
    env.action_space.seed(seed)
    buffer = PrioritizedReplayBuffer(100_000)

    # Create agent with DQN + Double + PER
    agent = DQNAgent(state_dim, n_actions, buffer, lr=lr, gamma=gamma, use_per=True, use_double=True)

    # Train and time it
    start_time = time.time()
    rewards, _ = train(agent, env, episodes=500, epsilon_decay=eps_decay)
    duration = time.time() - start_time

    # Evaluate
    avg_reward = np.mean(rewards[-100:])
    std_reward = np.std(rewards[-100:])

    grid_results.append({
        'lr': lr,
        'gamma': gamma,
        'eps_decay': eps_decay,
        'seed': seed,
        'avg_reward': avg_reward,
        'std_dev': std_reward,
        'duration_sec': duration
    })
```

```
 Running config: lr=0.001, gamma=0.99, eps_decay=0.99, seed=42
Episode    0 | Reward: -206.09 | Epsilon: 0.990
Episode   10 | Reward: -79.64 | Epsilon: 0.895
Episode   20 | Reward: -327.10 | Epsilon: 0.810
Episode   30 | Reward: -268.75 | Epsilon: 0.732
Episode   40 | Reward: -54.60 | Epsilon: 0.662
Episode   50 | Reward: -167.18 | Epsilon: 0.599
Episode   60 | Reward: -197.47 | Epsilon: 0.542
Episode   70 | Reward: -38.22 | Epsilon: 0.490
Episode   80 | Reward: -27.50 | Epsilon: 0.443
Episode   90 | Reward: 21.02 | Epsilon: 0.401
Episode  100 | Reward: -10.03 | Epsilon: 0.362
Episode  110 | Reward: -0.81 | Epsilon: 0.328
Episode  120 | Reward: -11.33 | Epsilon: 0.296
Episode  130 | Reward: -44.74 | Epsilon: 0.268
Episode  140 | Reward: -19.86 | Epsilon: 0.242
Episode  150 | Reward: -41.91 | Epsilon: 0.219
Episode  160 | Reward: -70.94 | Epsilon: 0.198
Episode  170 | Reward: -43.66 | Epsilon: 0.179
Episode  180 | Reward: -98.84 | Epsilon: 0.162
Episode  190 | Reward: -47.74 | Epsilon: 0.147
Episode  200 | Reward: -62.22 | Epsilon: 0.133
Episode  210 | Reward: -9.10 | Epsilon: 0.120
Episode  220 | Reward: -32.00 | Epsilon: 0.108
Episode  230 | Reward: -92.77 | Epsilon: 0.098
Episode  240 | Reward: -34.93 | Epsilon: 0.089
Episode  250 | Reward: -32.48 | Epsilon: 0.080
Episode  260 | Reward: -2.99 | Epsilon: 0.073
Episode  270 | Reward: -40.12 | Epsilon: 0.066
Episode  280 | Reward: -91.18 | Epsilon: 0.059
Episode  290 | Reward: -42.12 | Epsilon: 0.054
Episode  300 | Reward: -105.93 | Epsilon: 0.050
Episode  310 | Reward: 102.13 | Epsilon: 0.050
Episode  320 | Reward: -116.24 | Epsilon: 0.050
Episode  330 | Reward: -13.64 | Epsilon: 0.050
Episode  340 | Reward: -26.84 | Epsilon: 0.050
Episode  350 | Reward: -178.15 | Epsilon: 0.050
Episode  360 | Reward: 76.54 | Epsilon: 0.050
Episode  370 | Reward: -41.72 | Epsilon: 0.050
Episode  380 | Reward: -11.77 | Epsilon: 0.050
Episode  390 | Reward: -127.54 | Epsilon: 0.050
Episode  400 | Reward: 50.64 | Epsilon: 0.050
Episode  410 | Reward: 78.90 | Epsilon: 0.050
Episode  420 | Reward: -121.31 | Epsilon: 0.050
Episode  430 | Reward: 246.16 | Epsilon: 0.050
Episode  440 | Reward: 164.60 | Epsilon: 0.050
```

```
Episode  450 | Reward: 248.62 | Epsilon: 0.050
Episode  460 | Reward: 99.63 | Epsilon: 0.050
Episode  470 | Reward: -59.73 | Epsilon: 0.050
Episode  480 | Reward: -32.73 | Epsilon: 0.050
Episode  490 | Reward: -106.05 | Epsilon: 0.050

 Running config: lr=0.001, gamma=0.99, eps_decay=0.99, seed=123
Episode    0 | Reward: -418.96 | Epsilon: 0.990
Episode   10 | Reward: -122.24 | Epsilon: 0.895
Episode   20 | Reward: -14.87 | Epsilon: 0.810
Episode   30 | Reward: -240.15 | Epsilon: 0.732
Episode   40 | Reward: -126.00 | Epsilon: 0.662
Episode   50 | Reward: -73.80 | Epsilon: 0.599
Episode   60 | Reward: -62.68 | Epsilon: 0.542
Episode   70 | Reward: -129.99 | Epsilon: 0.490
Episode   80 | Reward: -84.35 | Epsilon: 0.443
Episode   90 | Reward: -119.83 | Epsilon: 0.401
Episode  100 | Reward: -140.79 | Epsilon: 0.362
Episode  110 | Reward: -355.63 | Epsilon: 0.328
Episode  120 | Reward: -221.77 | Epsilon: 0.296
Episode  130 | Reward: 21.33 | Epsilon: 0.268
Episode  140 | Reward: -22.04 | Epsilon: 0.242
Episode  150 | Reward: -56.40 | Epsilon: 0.219
Episode  160 | Reward: -68.76 | Epsilon: 0.198
Episode  170 | Reward: 32.22 | Epsilon: 0.179
Episode  180 | Reward: -47.93 | Epsilon: 0.162
Episode  190 | Reward: -108.15 | Epsilon: 0.147
Episode  200 | Reward: -94.00 | Epsilon: 0.133
Episode  210 | Reward: -32.67 | Epsilon: 0.120
Episode  220 | Reward: -104.50 | Epsilon: 0.108
Episode  230 | Reward: 5.23 | Epsilon: 0.098
Episode  240 | Reward: -30.11 | Epsilon: 0.089
Episode  250 | Reward: -13.65 | Epsilon: 0.080
Episode  260 | Reward: -17.81 | Epsilon: 0.073
Episode  270 | Reward: -115.80 | Epsilon: 0.066
Episode  280 | Reward: -64.59 | Epsilon: 0.059
Episode  290 | Reward: 158.19 | Epsilon: 0.054
Episode  300 | Reward: -113.98 | Epsilon: 0.050
Episode  310 | Reward: 184.02 | Epsilon: 0.050
Episode  320 | Reward: 196.22 | Epsilon: 0.050
Episode  330 | Reward: -169.83 | Epsilon: 0.050
Episode  340 | Reward: 166.62 | Epsilon: 0.050
Episode  350 | Reward: -5.58 | Epsilon: 0.050
Episode  360 | Reward: -71.35 | Epsilon: 0.050
Episode  370 | Reward: -107.36 | Epsilon: 0.050
Episode  380 | Reward: -55.68 | Epsilon: 0.050
Episode  390 | Reward: 188.47 | Epsilon: 0.050
Episode  400 | Reward: 215.38 | Epsilon: 0.050
Episode  410 | Reward: -56.06 | Epsilon: 0.050
Episode  420 | Reward: -33.51 | Epsilon: 0.050
Episode  430 | Reward: 186.74 | Epsilon: 0.050
Episode  440 | Reward: 17.98 | Epsilon: 0.050
Episode  450 | Reward: -113.59 | Epsilon: 0.050
Episode  460 | Reward: -27.74 | Epsilon: 0.050
Episode  470 | Reward: 235.48 | Epsilon: 0.050
Episode  480 | Reward: 248.96 | Epsilon: 0.050
Episode  490 | Reward: -35.08 | Epsilon: 0.050

 Running config: lr=0.001, gamma=0.99, eps_decay=0.995, seed=42
Episode    0 | Reward: -206.09 | Epsilon: 0.995
Episode   10 | Reward: -120.50 | Epsilon: 0.946
Episode   20 | Reward: -87.29 | Epsilon: 0.900
Episode   30 | Reward: -189.06 | Epsilon: 0.856
Episode   40 | Reward: -243.73 | Epsilon: 0.814
Episode   50 | Reward: -84.27 | Epsilon: 0.774
Episode   60 | Reward: -110.99 | Epsilon: 0.737
Episode   70 | Reward: -280.57 | Epsilon: 0.701
Episode   80 | Reward: -358.69 | Epsilon: 0.666
Episode   90 | Reward: -91.18 | Epsilon: 0.634
Episode  100 | Reward: -96.32 | Epsilon: 0.603
Episode  110 | Reward: -56.71 | Epsilon: 0.573
Episode  120 | Reward: 5.11 | Epsilon: 0.545
Episode  130 | Reward: -36.75 | Epsilon: 0.519
Episode  140 | Reward: -25.71 | Epsilon: 0.493
Episode  150 | Reward: -135.50 | Epsilon: 0.469
Episode  160 | Reward: -30.54 | Epsilon: 0.446
Episode  170 | Reward: -30.42 | Epsilon: 0.424
Episode  180 | Reward: -150.90 | Epsilon: 0.404
Episode  190 | Reward: -17.13 | Epsilon: 0.384
Episode  200 | Reward: -28.50 | Epsilon: 0.365
Episode  210 | Reward: -165.72 | Epsilon: 0.347
Episode  220 | Reward: -0.86 | Epsilon: 0.330
Episode  230 | Reward: 37.87 | Epsilon: 0.314
```

```
Episode  240 | Reward: -85.25 | Epsilon: 0.299
Episode  250 | Reward: 19.49 | Epsilon: 0.284
Episode  260 | Reward: -97.22 | Epsilon: 0.270
Episode  270 | Reward: -34.73 | Epsilon: 0.257
Episode  280 | Reward: -46.18 | Epsilon: 0.245
Episode  290 | Reward: 10.66 | Epsilon: 0.233
Episode  300 | Reward: 10.72 | Epsilon: 0.221
Episode  310 | Reward: -51.93 | Epsilon: 0.210
Episode  320 | Reward: -53.11 | Epsilon: 0.200
Episode  330 | Reward: -32.78 | Epsilon: 0.190
Episode  340 | Reward: -53.95 | Epsilon: 0.181
Episode  350 | Reward: -102.77 | Epsilon: 0.172
Episode  360 | Reward: -48.72 | Epsilon: 0.164
Episode  370 | Reward: 35.70 | Epsilon: 0.156
Episode  380 | Reward: -66.66 | Epsilon: 0.148
Episode  390 | Reward: -32.96 | Epsilon: 0.141
Episode  400 | Reward: 13.24 | Epsilon: 0.134
Episode  410 | Reward: -46.17 | Epsilon: 0.127
Episode  420 | Reward: -46.59 | Epsilon: 0.121
Episode  430 | Reward: 235.07 | Epsilon: 0.115
Episode  440 | Reward: -30.35 | Epsilon: 0.110
Episode  450 | Reward: 265.29 | Epsilon: 0.104
Episode  460 | Reward: -14.06 | Epsilon: 0.099
Episode  470 | Reward: -16.46 | Epsilon: 0.094
Episode  480 | Reward: 23.51 | Epsilon: 0.090
Episode  490 | Reward: -21.96 | Epsilon: 0.085

 Running config: lr=0.001, gamma=0.99, eps_decay=0.995, seed=123
Episode    0 | Reward: -418.96 | Epsilon: 0.995
Episode   10 | Reward: -78.34 | Epsilon: 0.946
Episode   20 | Reward: -123.32 | Epsilon: 0.900
Episode   30 | Reward: -320.29 | Epsilon: 0.856
Episode   40 | Reward: -127.74 | Epsilon: 0.814
Episode   50 | Reward: -116.72 | Epsilon: 0.774
Episode   60 | Reward: -120.33 | Epsilon: 0.737
Episode   70 | Reward: -255.31 | Epsilon: 0.701
Episode   80 | Reward: -39.49 | Epsilon: 0.666
Episode   90 | Reward: -112.86 | Epsilon: 0.634
Episode  100 | Reward: -91.19 | Epsilon: 0.603
Episode  110 | Reward: -51.45 | Epsilon: 0.573
Episode  120 | Reward: -20.84 | Epsilon: 0.545
Episode  130 | Reward: -71.63 | Epsilon: 0.519
Episode  140 | Reward: -63.04 | Epsilon: 0.493
Episode  150 | Reward: -52.37 | Epsilon: 0.469
Episode  160 | Reward: -6.66 | Epsilon: 0.446
Episode  170 | Reward: -17.45 | Epsilon: 0.424
Episode  180 | Reward: 23.66 | Epsilon: 0.404
Episode  190 | Reward: -382.15 | Epsilon: 0.384
Episode  200 | Reward: -6.69 | Epsilon: 0.365
Episode  210 | Reward: -48.22 | Epsilon: 0.347
Episode  220 | Reward: -32.35 | Epsilon: 0.330
Episode  230 | Reward: -8.42 | Epsilon: 0.314
Episode  240 | Reward: -60.34 | Epsilon: 0.299
Episode  250 | Reward: -20.88 | Epsilon: 0.284
Episode  260 | Reward: 39.78 | Epsilon: 0.270
Episode  270 | Reward: -67.29 | Epsilon: 0.257
Episode  280 | Reward: -65.05 | Epsilon: 0.245
Episode  290 | Reward: -15.69 | Epsilon: 0.233
Episode  300 | Reward: -12.36 | Epsilon: 0.221
Episode  310 | Reward: 36.70 | Epsilon: 0.210
Episode  320 | Reward: -56.69 | Epsilon: 0.200
Episode  330 | Reward: -77.31 | Epsilon: 0.190
Episode  340 | Reward: 66.28 | Epsilon: 0.181
Episode  350 | Reward: 84.07 | Epsilon: 0.172
Episode  360 | Reward: -47.42 | Epsilon: 0.164
Episode  370 | Reward: 8.12 | Epsilon: 0.156
Episode  380 | Reward: -75.65 | Epsilon: 0.148
Episode  390 | Reward: 78.93 | Epsilon: 0.141
Episode  400 | Reward: 140.59 | Epsilon: 0.134
Episode  410 | Reward: 246.34 | Epsilon: 0.127
Episode  420 | Reward: -111.58 | Epsilon: 0.121
Episode  430 | Reward: 1.87 | Epsilon: 0.115
Episode  440 | Reward: -99.04 | Epsilon: 0.110
Episode  450 | Reward: 252.05 | Epsilon: 0.104
Episode  460 | Reward: 2.04 | Epsilon: 0.099
Episode  470 | Reward: -1.71 | Epsilon: 0.094
Episode  480 | Reward: -213.71 | Epsilon: 0.090
Episode  490 | Reward: -101.25 | Epsilon: 0.085

 Running config: lr=0.001, gamma=0.95, eps_decay=0.99, seed=42
Episode    0 | Reward: -206.09 | Epsilon: 0.990
Episode   10 | Reward: -69.37 | Epsilon: 0.895
Episode   20 | Reward: -23.41 | Epsilon: 0.810
```

```
Episode   30 | Reward: -191.73 | Epsilon: 0.732
Episode   40 | Reward: -76.47 | Epsilon: 0.662
Episode   50 | Reward: -90.89 | Epsilon: 0.599
Episode   60 | Reward: -49.53 | Epsilon: 0.542
Episode   70 | Reward: -107.20 | Epsilon: 0.490
Episode   80 | Reward: -33.14 | Epsilon: 0.443
Episode   90 | Reward: -9.83 | Epsilon: 0.401
Episode  100 | Reward: -33.43 | Epsilon: 0.362
Episode  110 | Reward: -35.68 | Epsilon: 0.328
Episode  120 | Reward: -9.13 | Epsilon: 0.296
Episode  130 | Reward: -78.28 | Epsilon: 0.268
Episode  140 | Reward: -28.32 | Epsilon: 0.242
Episode  150 | Reward: 24.66 | Epsilon: 0.219
Episode  160 | Reward: -67.89 | Epsilon: 0.198
Episode  170 | Reward: -48.87 | Epsilon: 0.179
Episode  180 | Reward: -81.91 | Epsilon: 0.162
Episode  190 | Reward: -92.75 | Epsilon: 0.147
Episode  200 | Reward: -60.06 | Epsilon: 0.133
Episode  210 | Reward: -106.55 | Epsilon: 0.120
Episode  220 | Reward: -140.60 | Epsilon: 0.108
Episode  230 | Reward: -50.98 | Epsilon: 0.098
Episode  240 | Reward: -18.12 | Epsilon: 0.089
Episode  250 | Reward: -191.43 | Epsilon: 0.080
Episode  260 | Reward: -97.28 | Epsilon: 0.073
Episode  270 | Reward: 82.20 | Epsilon: 0.066
Episode  280 | Reward: -155.31 | Epsilon: 0.059
Episode  290 | Reward: -63.31 | Epsilon: 0.054
Episode  300 | Reward: -241.29 | Epsilon: 0.050
Episode  310 | Reward: -136.09 | Epsilon: 0.050
Episode  320 | Reward: -62.38 | Epsilon: 0.050
Episode  330 | Reward: 35.78 | Epsilon: 0.050
Episode  340 | Reward: -194.51 | Epsilon: 0.050
Episode  350 | Reward: -102.24 | Epsilon: 0.050
Episode  360 | Reward: -199.56 | Epsilon: 0.050
Episode  370 | Reward: 206.52 | Epsilon: 0.050
Episode  380 | Reward: -108.92 | Epsilon: 0.050
Episode  390 | Reward: -24.16 | Epsilon: 0.050
Episode  400 | Reward: -36.41 | Epsilon: 0.050
Episode  410 | Reward: -79.42 | Epsilon: 0.050
Episode  420 | Reward: -54.74 | Epsilon: 0.050
Episode  430 | Reward: -83.07 | Epsilon: 0.050
Episode  440 | Reward: -5.16 | Epsilon: 0.050
Episode  450 | Reward: -65.94 | Epsilon: 0.050
Episode  460 | Reward: -35.30 | Epsilon: 0.050
Episode  470 | Reward: -64.10 | Epsilon: 0.050
Episode  480 | Reward: -38.56 | Epsilon: 0.050
Episode  490 | Reward: -67.65 | Epsilon: 0.050

 Running config: lr=0.001, gamma=0.95, eps_decay=0.99, seed=123
Episode    0 | Reward: -418.96 | Epsilon: 0.990
Episode   10 | Reward: -241.66 | Epsilon: 0.895
Episode   20 | Reward: -146.29 | Epsilon: 0.810
Episode   30 | Reward: -142.17 | Epsilon: 0.732
Episode   40 | Reward: -154.78 | Epsilon: 0.662
Episode   50 | Reward: -15.19 | Epsilon: 0.599
Episode   60 | Reward: -119.66 | Epsilon: 0.542
Episode   70 | Reward: -64.12 | Epsilon: 0.490
Episode   80 | Reward: 44.00 | Epsilon: 0.443
Episode   90 | Reward: 12.71 | Epsilon: 0.401
Episode  100 | Reward: -138.66 | Epsilon: 0.362
Episode  110 | Reward: -225.33 | Epsilon: 0.328
Episode  120 | Reward: -63.94 | Epsilon: 0.296
Episode  130 | Reward: 24.81 | Epsilon: 0.268
Episode  140 | Reward: 16.14 | Epsilon: 0.242
Episode  150 | Reward: 78.25 | Epsilon: 0.219
Episode  160 | Reward: -115.28 | Epsilon: 0.198
Episode  170 | Reward: -71.65 | Epsilon: 0.179
Episode  180 | Reward: -98.70 | Epsilon: 0.162
Episode  190 | Reward: -64.00 | Epsilon: 0.147
Episode  200 | Reward: -9.95 | Epsilon: 0.133
Episode  210 | Reward: -68.03 | Epsilon: 0.120
Episode  220 | Reward: -67.51 | Epsilon: 0.108
Episode  230 | Reward: -80.23 | Epsilon: 0.098
Episode  240 | Reward: -74.45 | Epsilon: 0.089
Episode  250 | Reward: -102.87 | Epsilon: 0.080
Episode  260 | Reward: 45.21 | Epsilon: 0.073
Episode  270 | Reward: -97.64 | Epsilon: 0.066
Episode  280 | Reward: -103.62 | Epsilon: 0.059
Episode  290 | Reward: -77.99 | Epsilon: 0.054
Episode  300 | Reward: -104.63 | Epsilon: 0.050
Episode  310 | Reward: -90.20 | Epsilon: 0.050
Episode  320 | Reward: -91.02 | Epsilon: 0.050
Episode  330 | Reward: -57.03 | Epsilon: 0.050
```

```
Episode  340 | Reward: -80.72 | Epsilon: 0.050
Episode  350 | Reward: -88.71 | Epsilon: 0.050
Episode  360 | Reward: -63.14 | Epsilon: 0.050
Episode  370 | Reward: -81.81 | Epsilon: 0.050
Episode  380 | Reward: -125.40 | Epsilon: 0.050
Episode  390 | Reward: -104.30 | Epsilon: 0.050
Episode  400 | Reward: -48.88 | Epsilon: 0.050
Episode  410 | Reward: -58.74 | Epsilon: 0.050
Episode  420 | Reward: -83.09 | Epsilon: 0.050
Episode  430 | Reward: -87.35 | Epsilon: 0.050
Episode  440 | Reward: -95.58 | Epsilon: 0.050
Episode  450 | Reward: -49.04 | Epsilon: 0.050
Episode  460 | Reward: -105.55 | Epsilon: 0.050
Episode  470 | Reward: -69.05 | Epsilon: 0.050
Episode  480 | Reward: -72.41 | Epsilon: 0.050
Episode  490 | Reward: -27.80 | Epsilon: 0.050

 Running config: lr=0.001, gamma=0.95, eps_decay=0.995, seed=42
Episode    0 | Reward: -206.09 | Epsilon: 0.995
Episode   10 | Reward: -80.82 | Epsilon: 0.946
Episode   20 | Reward: -123.72 | Epsilon: 0.900
Episode   30 | Reward: -148.30 | Epsilon: 0.856
Episode   40 | Reward: -218.76 | Epsilon: 0.814
Episode   50 | Reward: 27.60 | Epsilon: 0.774
Episode   60 | Reward: -141.58 | Epsilon: 0.737
Episode   70 | Reward: -62.31 | Epsilon: 0.701
Episode   80 | Reward: -211.78 | Epsilon: 0.666
Episode   90 | Reward: -39.10 | Epsilon: 0.634
Episode  100 | Reward: -82.00 | Epsilon: 0.603
Episode  110 | Reward: -134.13 | Epsilon: 0.573
Episode  120 | Reward: -85.01 | Epsilon: 0.545
Episode  130 | Reward: -32.19 | Epsilon: 0.519
Episode  140 | Reward: -105.94 | Epsilon: 0.493
Episode  150 | Reward: -173.25 | Epsilon: 0.469
Episode  160 | Reward: -59.83 | Epsilon: 0.446
Episode  170 | Reward: 112.53 | Epsilon: 0.424
Episode  180 | Reward: 19.85 | Epsilon: 0.404
Episode  190 | Reward: -6.31 | Epsilon: 0.384
Episode  200 | Reward: 53.13 | Epsilon: 0.365
Episode  210 | Reward: 134.67 | Epsilon: 0.347
Episode  220 | Reward: -76.87 | Epsilon: 0.330
Episode  230 | Reward: 109.58 | Epsilon: 0.314
Episode  240 | Reward: -33.90 | Epsilon: 0.299
Episode  250 | Reward: -106.62 | Epsilon: 0.284
Episode  260 | Reward: -82.79 | Epsilon: 0.270
Episode  270 | Reward: 114.50 | Epsilon: 0.257
Episode  280 | Reward: -62.62 | Epsilon: 0.245
Episode  290 | Reward: 92.52 | Epsilon: 0.233
Episode  300 | Reward: -95.03 | Epsilon: 0.221
Episode  310 | Reward: -49.64 | Epsilon: 0.210
Episode  320 | Reward: -131.49 | Epsilon: 0.200
Episode  330 | Reward: -82.25 | Epsilon: 0.190
Episode  340 | Reward: -38.36 | Epsilon: 0.181
Episode  350 | Reward: -92.43 | Epsilon: 0.172
Episode  360 | Reward: -124.36 | Epsilon: 0.164
Episode  370 | Reward: -54.46 | Epsilon: 0.156
Episode  380 | Reward: -87.32 | Epsilon: 0.148
Episode  390 | Reward: -43.59 | Epsilon: 0.141
Episode  400 | Reward: 47.82 | Epsilon: 0.134
Episode  410 | Reward: -46.63 | Epsilon: 0.127
Episode  420 | Reward: 263.61 | Epsilon: 0.121
Episode  430 | Reward: -21.83 | Epsilon: 0.115
Episode  440 | Reward: -123.52 | Epsilon: 0.110
Episode  450 | Reward: 2.49 | Epsilon: 0.104
Episode  460 | Reward: -9.44 | Epsilon: 0.099
Episode  470 | Reward: -61.18 | Epsilon: 0.094
Episode  480 | Reward: -111.68 | Epsilon: 0.090
Episode  490 | Reward: -95.90 | Epsilon: 0.085

 Running config: lr=0.001, gamma=0.95, eps_decay=0.995, seed=123
Episode    0 | Reward: -418.96 | Epsilon: 0.995
Episode   10 | Reward: -114.87 | Epsilon: 0.946
Episode   20 | Reward: -269.67 | Epsilon: 0.900
Episode   30 | Reward: -127.43 | Epsilon: 0.856
Episode   40 | Reward: -86.68 | Epsilon: 0.814
Episode   50 | Reward: -170.18 | Epsilon: 0.774
Episode   60 | Reward: -198.49 | Epsilon: 0.737
Episode   70 | Reward: -78.62 | Epsilon: 0.701
Episode   80 | Reward: -74.72 | Epsilon: 0.666
Episode   90 | Reward: -25.72 | Epsilon: 0.634
Episode  100 | Reward: -97.22 | Epsilon: 0.603
Episode  110 | Reward: -7.75 | Epsilon: 0.573
Episode  120 | Reward: -114.79 | Epsilon: 0.545
```

```
Episode  130 | Reward: -49.37 | Epsilon: 0.519
Episode  140 | Reward: -20.03 | Epsilon: 0.493
Episode  150 | Reward: -58.55 | Epsilon: 0.469
Episode  160 | Reward: -33.48 | Epsilon: 0.446
Episode  170 | Reward: 1.05 | Epsilon: 0.424
Episode  180 | Reward: -55.29 | Epsilon: 0.404
Episode  190 | Reward: -40.55 | Epsilon: 0.384
Episode  200 | Reward: -27.55 | Epsilon: 0.365
Episode  210 | Reward: -130.41 | Epsilon: 0.347
Episode  220 | Reward: 0.54 | Epsilon: 0.330
Episode  230 | Reward: -141.59 | Epsilon: 0.314
Episode  240 | Reward: -80.50 | Epsilon: 0.299
Episode  250 | Reward: -69.60 | Epsilon: 0.284
Episode  260 | Reward: -400.85 | Epsilon: 0.270
Episode  270 | Reward: -144.67 | Epsilon: 0.257
Episode  280 | Reward: -92.88 | Epsilon: 0.245
Episode  290 | Reward: -25.01 | Epsilon: 0.233
Episode  300 | Reward: -194.51 | Epsilon: 0.221
Episode  310 | Reward: -212.49 | Epsilon: 0.210
Episode  320 | Reward: -39.54 | Epsilon: 0.200
Episode  330 | Reward: -77.91 | Epsilon: 0.190
Episode  340 | Reward: -178.43 | Epsilon: 0.181
Episode  350 | Reward: -67.62 | Epsilon: 0.172
Episode  360 | Reward: -82.04 | Epsilon: 0.164
Episode  370 | Reward: -43.78 | Epsilon: 0.156
Episode  380 | Reward: -68.51 | Epsilon: 0.148
Episode  390 | Reward: -106.78 | Epsilon: 0.141
Episode  400 | Reward: -21.38 | Epsilon: 0.134
Episode  410 | Reward: -77.65 | Epsilon: 0.127
Episode  420 | Reward: -48.87 | Epsilon: 0.121
Episode  430 | Reward: -113.44 | Epsilon: 0.115
Episode  440 | Reward: -34.35 | Epsilon: 0.110
Episode  450 | Reward: -89.58 | Epsilon: 0.104
Episode  460 | Reward: -58.27 | Epsilon: 0.099
Episode  470 | Reward: -30.83 | Epsilon: 0.094
Episode  480 | Reward: -79.31 | Epsilon: 0.090
Episode  490 | Reward: -107.39 | Epsilon: 0.085

 Running config: lr=0.0005, gamma=0.99, eps_decay=0.99, seed=42
Episode    0 | Reward: -206.09 | Epsilon: 0.990
Episode   10 | Reward: -344.42 | Epsilon: 0.895
Episode   20 | Reward: -413.60 | Epsilon: 0.810
Episode   30 | Reward: -199.65 | Epsilon: 0.732
Episode   40 | Reward: -117.51 | Epsilon: 0.662
Episode   50 | Reward: -176.43 | Epsilon: 0.599
Episode   60 | Reward: -103.66 | Epsilon: 0.542
Episode   70 | Reward: -168.58 | Epsilon: 0.490
Episode   80 | Reward: 24.69 | Epsilon: 0.443
Episode   90 | Reward: 33.29 | Epsilon: 0.401
Episode  100 | Reward: -228.43 | Epsilon: 0.362
Episode  110 | Reward: -339.22 | Epsilon: 0.328
Episode  120 | Reward: -54.04 | Epsilon: 0.296
Episode  130 | Reward: -94.51 | Epsilon: 0.268
Episode  140 | Reward: -160.07 | Epsilon: 0.242
Episode  150 | Reward: -111.32 | Epsilon: 0.219
Episode  160 | Reward: -121.22 | Epsilon: 0.198
Episode  170 | Reward: -66.77 | Epsilon: 0.179
Episode  180 | Reward: -95.96 | Epsilon: 0.162
Episode  190 | Reward: -104.50 | Epsilon: 0.147
Episode  200 | Reward: 4.39 | Epsilon: 0.133
Episode  210 | Reward: -6.62 | Epsilon: 0.120
Episode  220 | Reward: -68.97 | Epsilon: 0.108
Episode  230 | Reward: 9.19 | Epsilon: 0.098
Episode  240 | Reward: -21.40 | Epsilon: 0.089
Episode  250 | Reward: -44.36 | Epsilon: 0.080
Episode  260 | Reward: -15.99 | Epsilon: 0.073
Episode  270 | Reward: -21.54 | Epsilon: 0.066
Episode  280 | Reward: -16.59 | Epsilon: 0.059
Episode  290 | Reward: -94.29 | Epsilon: 0.054
Episode  300 | Reward: -15.42 | Epsilon: 0.050
Episode  310 | Reward: -6.71 | Epsilon: 0.050
Episode  320 | Reward: 219.61 | Epsilon: 0.050
Episode  330 | Reward: -40.37 | Epsilon: 0.050
Episode  340 | Reward: -31.11 | Epsilon: 0.050
Episode  350 | Reward: -22.28 | Epsilon: 0.050
Episode  360 | Reward: 237.25 | Epsilon: 0.050
Episode  370 | Reward: -57.79 | Epsilon: 0.050
Episode  380 | Reward: -47.52 | Epsilon: 0.050
Episode  390 | Reward: 313.89 | Epsilon: 0.050
Episode  400 | Reward: 188.21 | Epsilon: 0.050
Episode  410 | Reward: -149.86 | Epsilon: 0.050
Episode  420 | Reward: 137.55 | Epsilon: 0.050
Episode  430 | Reward: -77.53 | Epsilon: 0.050
```

```
Episode  440 | Reward: -93.88 | Epsilon: 0.050
Episode  450 | Reward: 199.22 | Epsilon: 0.050
Episode  460 | Reward: 187.31 | Epsilon: 0.050
Episode  470 | Reward: 247.68 | Epsilon: 0.050
Episode  480 | Reward: 204.56 | Epsilon: 0.050
Episode  490 | Reward: 233.93 | Epsilon: 0.050

 Running config: lr=0.0005, gamma=0.99, eps_decay=0.99, seed=123
Episode    0 | Reward: -418.96 | Epsilon: 0.990
Episode   10 | Reward: -227.68 | Epsilon: 0.895
Episode   20 | Reward: -88.76 | Epsilon: 0.810
Episode   30 | Reward: -154.81 | Epsilon: 0.732
Episode   40 | Reward: -307.04 | Epsilon: 0.662
Episode   50 | Reward: -19.41 | Epsilon: 0.599
Episode   60 | Reward: -353.92 | Epsilon: 0.542
Episode   70 | Reward: -343.83 | Epsilon: 0.490
Episode   80 | Reward: -1.34 | Epsilon: 0.443
Episode   90 | Reward: -19.29 | Epsilon: 0.401
Episode  100 | Reward: -311.71 | Epsilon: 0.362
Episode  110 | Reward: -172.71 | Epsilon: 0.328
Episode  120 | Reward: -118.23 | Epsilon: 0.296
Episode  130 | Reward: -82.43 | Epsilon: 0.268
Episode  140 | Reward: -66.11 | Epsilon: 0.242
Episode  150 | Reward: -64.26 | Epsilon: 0.219
Episode  160 | Reward: -32.90 | Epsilon: 0.198
Episode  170 | Reward: -26.82 | Epsilon: 0.179
Episode  180 | Reward: -97.00 | Epsilon: 0.162
Episode  190 | Reward: -72.12 | Epsilon: 0.147
Episode  200 | Reward: -48.58 | Epsilon: 0.133
Episode  210 | Reward: 0.02 | Epsilon: 0.120
Episode  220 | Reward: -3.79 | Epsilon: 0.108
Episode  230 | Reward: 14.56 | Epsilon: 0.098
Episode  240 | Reward: -8.11 | Epsilon: 0.089
Episode  250 | Reward: -51.77 | Epsilon: 0.080
Episode  260 | Reward: -75.41 | Epsilon: 0.073
Episode  270 | Reward: -140.78 | Epsilon: 0.066
Episode  280 | Reward: -19.78 | Epsilon: 0.059
Episode  290 | Reward: -48.56 | Epsilon: 0.054
Episode  300 | Reward: -37.98 | Epsilon: 0.050
Episode  310 | Reward: -37.30 | Epsilon: 0.050
Episode  320 | Reward: -55.94 | Epsilon: 0.050
Episode  330 | Reward: -26.13 | Epsilon: 0.050
Episode  340 | Reward: -26.97 | Epsilon: 0.050
Episode  350 | Reward: -73.62 | Epsilon: 0.050
Episode  360 | Reward: 217.57 | Epsilon: 0.050
Episode  370 | Reward: -45.47 | Epsilon: 0.050
Episode  380 | Reward: 186.85 | Epsilon: 0.050
Episode  390 | Reward: -24.72 | Epsilon: 0.050
Episode  400 | Reward: -35.27 | Epsilon: 0.050
Episode  410 | Reward: -25.47 | Epsilon: 0.050
Episode  420 | Reward: -5.51 | Epsilon: 0.050
Episode  430 | Reward: 10.67 | Epsilon: 0.050
Episode  440 | Reward: -9.24 | Epsilon: 0.050
Episode  450 | Reward: -23.96 | Epsilon: 0.050
Episode  460 | Reward: -49.49 | Epsilon: 0.050
Episode  470 | Reward: -52.64 | Epsilon: 0.050
Episode  480 | Reward: 112.74 | Epsilon: 0.050
Episode  490 | Reward: -101.82 | Epsilon: 0.050

 Running config: lr=0.0005, gamma=0.99, eps_decay=0.995, seed=42
Episode    0 | Reward: -206.09 | Epsilon: 0.995
Episode   10 | Reward: -151.16 | Epsilon: 0.946
Episode   20 | Reward: -375.37 | Epsilon: 0.900
Episode   30 | Reward: -160.75 | Epsilon: 0.856
Episode   40 | Reward: -126.62 | Epsilon: 0.814
Episode   50 | Reward: -134.20 | Epsilon: 0.774
Episode   60 | Reward: -183.71 | Epsilon: 0.737
Episode   70 | Reward: -321.44 | Epsilon: 0.701
Episode   80 | Reward: -36.79 | Epsilon: 0.666
Episode   90 | Reward: -71.52 | Epsilon: 0.634
Episode  100 | Reward: -37.39 | Epsilon: 0.603
Episode  110 | Reward: -32.76 | Epsilon: 0.573
Episode  120 | Reward: -82.75 | Epsilon: 0.545
Episode  130 | Reward: -67.32 | Epsilon: 0.519
Episode  140 | Reward: -35.43 | Epsilon: 0.493
Episode  150 | Reward: -15.44 | Epsilon: 0.469
Episode  160 | Reward: -77.17 | Epsilon: 0.446
Episode  170 | Reward: -155.98 | Epsilon: 0.424
Episode  180 | Reward: -199.41 | Epsilon: 0.404
Episode  190 | Reward: -47.78 | Epsilon: 0.384
Episode  200 | Reward: -115.90 | Epsilon: 0.365
Episode  210 | Reward: -265.23 | Epsilon: 0.347
Episode  220 | Reward: -16.45 | Epsilon: 0.330
```

```
Episode  230 | Reward: -169.27 | Epsilon: 0.314
Episode  240 | Reward: -92.17 | Epsilon: 0.299
Episode  250 | Reward: -4.84 | Epsilon: 0.284
Episode  260 | Reward: -36.21 | Epsilon: 0.270
Episode  270 | Reward: -130.30 | Epsilon: 0.257
Episode  280 | Reward: -8.20 | Epsilon: 0.245
Episode  290 | Reward: -27.78 | Epsilon: 0.233
Episode  300 | Reward: -21.50 | Epsilon: 0.221
Episode  310 | Reward: -17.91 | Epsilon: 0.210
Episode  320 | Reward: -61.64 | Epsilon: 0.200
Episode  330 | Reward: -48.73 | Epsilon: 0.190
Episode  340 | Reward: 174.41 | Epsilon: 0.181
Episode  350 | Reward: 17.65 | Epsilon: 0.172
Episode  360 | Reward: -71.43 | Epsilon: 0.164
Episode  370 | Reward: -17.58 | Epsilon: 0.156
Episode  380 | Reward: -15.56 | Epsilon: 0.148
Episode  390 | Reward: -73.26 | Epsilon: 0.141
Episode  400 | Reward: -4.47 | Epsilon: 0.134
Episode  410 | Reward: -14.06 | Epsilon: 0.127
Episode  420 | Reward: 4.13 | Epsilon: 0.121
Episode  430 | Reward: -43.27 | Epsilon: 0.115
Episode  440 | Reward: 38.07 | Epsilon: 0.110
Episode  450 | Reward: 2.16 | Epsilon: 0.104
Episode  460 | Reward: 45.17 | Epsilon: 0.099
Episode  470 | Reward: 280.76 | Epsilon: 0.094
Episode  480 | Reward: 77.13 | Epsilon: 0.090
Episode  490 | Reward: 207.09 | Epsilon: 0.085

 Running config: lr=0.0005, gamma=0.99, eps_decay=0.995, seed=123
Episode    0 | Reward: -418.96 | Epsilon: 0.995
Episode   10 | Reward: -76.65 | Epsilon: 0.946
Episode   20 | Reward: -183.51 | Epsilon: 0.900
Episode   30 | Reward: -173.93 | Epsilon: 0.856
Episode   40 | Reward: -72.05 | Epsilon: 0.814
Episode   50 | Reward: -119.66 | Epsilon: 0.774
Episode   60 | Reward: -123.62 | Epsilon: 0.737
Episode   70 | Reward: -46.84 | Epsilon: 0.701
Episode   80 | Reward: -214.62 | Epsilon: 0.666
Episode   90 | Reward: -56.75 | Epsilon: 0.634
Episode  100 | Reward: -41.35 | Epsilon: 0.603
Episode  110 | Reward: -12.91 | Epsilon: 0.573
Episode  120 | Reward: -62.76 | Epsilon: 0.545
Episode  130 | Reward: -58.58 | Epsilon: 0.519
Episode  140 | Reward: 11.08 | Epsilon: 0.493
Episode  150 | Reward: -313.56 | Epsilon: 0.469
Episode  160 | Reward: -2.61 | Epsilon: 0.446
Episode  170 | Reward: -99.56 | Epsilon: 0.424
Episode  180 | Reward: -16.33 | Epsilon: 0.404
Episode  190 | Reward: -242.35 | Epsilon: 0.384
Episode  200 | Reward: -145.21 | Epsilon: 0.365
Episode  210 | Reward: -38.82 | Epsilon: 0.347
Episode  220 | Reward: -176.64 | Epsilon: 0.330
Episode  230 | Reward: -116.03 | Epsilon: 0.314
Episode  240 | Reward: -52.17 | Epsilon: 0.299
Episode  250 | Reward: -86.45 | Epsilon: 0.284
Episode  260 | Reward: -124.04 | Epsilon: 0.270
Episode  270 | Reward: -18.04 | Epsilon: 0.257
Episode  280 | Reward: -63.81 | Epsilon: 0.245
Episode  290 | Reward: -55.89 | Epsilon: 0.233
Episode  300 | Reward: -89.66 | Epsilon: 0.221
Episode  310 | Reward: -109.22 | Epsilon: 0.210
Episode  320 | Reward: -66.48 | Epsilon: 0.200
Episode  330 | Reward: -91.64 | Epsilon: 0.190
Episode  340 | Reward: -29.96 | Epsilon: 0.181
Episode  350 | Reward: -63.78 | Epsilon: 0.172
Episode  360 | Reward: -30.58 | Epsilon: 0.164
Episode  370 | Reward: -4.32 | Epsilon: 0.156
Episode  380 | Reward: -21.09 | Epsilon: 0.148
Episode  390 | Reward: -26.11 | Epsilon: 0.141
Episode  400 | Reward: -55.00 | Epsilon: 0.134
Episode  410 | Reward: 6.36 | Epsilon: 0.127
Episode  420 | Reward: 19.62 | Epsilon: 0.121
Episode  430 | Reward: -290.61 | Epsilon: 0.115
Episode  440 | Reward: -8.40 | Epsilon: 0.110
Episode  450 | Reward: -33.80 | Epsilon: 0.104
Episode  460 | Reward: 2.13 | Epsilon: 0.099
Episode  470 | Reward: -58.48 | Epsilon: 0.094
Episode  480 | Reward: -68.66 | Epsilon: 0.090
Episode  490 | Reward: -43.58 | Epsilon: 0.085

 Running config: lr=0.0005, gamma=0.95, eps_decay=0.99, seed=42
Episode    0 | Reward: -206.09 | Epsilon: 0.990
Episode   10 | Reward: -110.15 | Epsilon: 0.895
```

```
Episode   20 | Reward: -120.82 | Epsilon: 0.810
Episode   30 | Reward: -48.88 | Epsilon: 0.732
Episode   40 | Reward: -156.22 | Epsilon: 0.662
Episode   50 | Reward: -128.93 | Epsilon: 0.599
Episode   60 | Reward: -156.09 | Epsilon: 0.542
Episode   70 | Reward: -11.67 | Epsilon: 0.490
Episode   80 | Reward: -83.24 | Epsilon: 0.443
Episode   90 | Reward: -137.49 | Epsilon: 0.401
Episode  100 | Reward: -179.73 | Epsilon: 0.362
Episode  110 | Reward: -111.95 | Epsilon: 0.328
Episode  120 | Reward: -38.82 | Epsilon: 0.296
Episode  130 | Reward: -92.73 | Epsilon: 0.268
Episode  140 | Reward: -12.61 | Epsilon: 0.242
Episode  150 | Reward: -82.62 | Epsilon: 0.219
Episode  160 | Reward: -34.49 | Epsilon: 0.198
Episode  170 | Reward: -50.11 | Epsilon: 0.179
Episode  180 | Reward: -99.91 | Epsilon: 0.162
Episode  190 | Reward: -127.75 | Epsilon: 0.147
Episode  200 | Reward: -76.01 | Epsilon: 0.133
Episode  210 | Reward: -58.95 | Epsilon: 0.120
Episode  220 | Reward: -36.17 | Epsilon: 0.108
Episode  230 | Reward: -118.67 | Epsilon: 0.098
Episode  240 | Reward: -127.18 | Epsilon: 0.089
Episode  250 | Reward: -103.47 | Epsilon: 0.080
Episode  260 | Reward: -89.30 | Epsilon: 0.073
Episode  270 | Reward: -120.23 | Epsilon: 0.066
Episode  280 | Reward: -123.02 | Epsilon: 0.059
Episode  290 | Reward: -94.22 | Epsilon: 0.054
Episode  300 | Reward: -74.39 | Epsilon: 0.050
Episode  310 | Reward: -44.68 | Epsilon: 0.050
Episode  320 | Reward: -59.65 | Epsilon: 0.050
Episode  330 | Reward: -43.66 | Epsilon: 0.050
Episode  340 | Reward: -93.67 | Epsilon: 0.050
Episode  350 | Reward: -76.97 | Epsilon: 0.050
Episode  360 | Reward: -15.74 | Epsilon: 0.050
Episode  370 | Reward: -65.02 | Epsilon: 0.050
Episode  380 | Reward: -24.75 | Epsilon: 0.050
Episode  390 | Reward: -111.42 | Epsilon: 0.050
Episode  400 | Reward: -55.77 | Epsilon: 0.050
Episode  410 | Reward: -54.32 | Epsilon: 0.050
Episode  420 | Reward: -82.46 | Epsilon: 0.050
Episode  430 | Reward: -26.67 | Epsilon: 0.050
Episode  440 | Reward: -79.21 | Epsilon: 0.050
Episode  450 | Reward: -95.49 | Epsilon: 0.050
Episode  460 | Reward: -72.05 | Epsilon: 0.050
Episode  470 | Reward: -83.25 | Epsilon: 0.050
Episode  480 | Reward: -63.17 | Epsilon: 0.050
Episode  490 | Reward: -65.89 | Epsilon: 0.050

 Running config: lr=0.0005, gamma=0.95, eps_decay=0.99, seed=123
Episode    0 | Reward: -418.96 | Epsilon: 0.990
Episode   10 | Reward: -102.79 | Epsilon: 0.895
Episode   20 | Reward: -55.73 | Epsilon: 0.810
Episode   30 | Reward: -285.62 | Epsilon: 0.732
Episode   40 | Reward: -50.08 | Epsilon: 0.662
Episode   50 | Reward: -93.25 | Epsilon: 0.599
Episode   60 | Reward: -70.43 | Epsilon: 0.542
Episode   70 | Reward: -18.23 | Epsilon: 0.490
Episode   80 | Reward: -59.12 | Epsilon: 0.443
Episode   90 | Reward: 8.02 | Epsilon: 0.401
Episode  100 | Reward: -221.91 | Epsilon: 0.362
Episode  110 | Reward: -32.27 | Epsilon: 0.328
Episode  120 | Reward: -245.58 | Epsilon: 0.296
Episode  130 | Reward: 57.75 | Epsilon: 0.268
Episode  140 | Reward: 58.63 | Epsilon: 0.242
Episode  150 | Reward: -88.31 | Epsilon: 0.219
Episode  160 | Reward: 251.21 | Epsilon: 0.198
Episode  170 | Reward: 13.61 | Epsilon: 0.179
Episode  180 | Reward: -161.25 | Epsilon: 0.162
Episode  190 | Reward: -80.56 | Epsilon: 0.147
Episode  200 | Reward: -115.94 | Epsilon: 0.133
Episode  210 | Reward: -59.59 | Epsilon: 0.120
Episode  220 | Reward: -126.16 | Epsilon: 0.108
Episode  230 | Reward: -86.85 | Epsilon: 0.098
Episode  240 | Reward: -84.03 | Epsilon: 0.089
Episode  250 | Reward: -73.86 | Epsilon: 0.080
Episode  260 | Reward: -120.78 | Epsilon: 0.073
Episode  270 | Reward: -86.31 | Epsilon: 0.066
Episode  280 | Reward: -97.42 | Epsilon: 0.059
Episode  290 | Reward: -108.94 | Epsilon: 0.054
Episode  300 | Reward: -133.76 | Epsilon: 0.050
Episode  310 | Reward: -88.86 | Epsilon: 0.050
Episode  320 | Reward: -130.99 | Epsilon: 0.050
```

```
Episode  330 | Reward: -88.15 | Epsilon: 0.050
Episode  340 | Reward: -84.34 | Epsilon: 0.050
Episode  350 | Reward: -31.18 | Epsilon: 0.050
Episode  360 | Reward: -35.72 | Epsilon: 0.050
Episode  370 | Reward: -50.65 | Epsilon: 0.050
Episode  380 | Reward: -50.05 | Epsilon: 0.050
Episode  390 | Reward: -199.76 | Epsilon: 0.050
Episode  400 | Reward: -102.08 | Epsilon: 0.050
Episode  410 | Reward: -87.81 | Epsilon: 0.050
Episode  420 | Reward: -50.69 | Epsilon: 0.050
Episode  430 | Reward: 46.87 | Epsilon: 0.050
Episode  440 | Reward: 69.69 | Epsilon: 0.050
Episode  450 | Reward: 77.96 | Epsilon: 0.050
Episode  460 | Reward: -237.05 | Epsilon: 0.050
Episode  470 | Reward: -50.05 | Epsilon: 0.050
Episode  480 | Reward: -38.64 | Epsilon: 0.050
Episode  490 | Reward: -38.98 | Epsilon: 0.050

 Running config: lr=0.0005, gamma=0.95, eps_decay=0.995, seed=42
Episode    0 | Reward: -206.09 | Epsilon: 0.995
Episode   10 | Reward: -135.72 | Epsilon: 0.946
Episode   20 | Reward: -205.52 | Epsilon: 0.900
Episode   30 | Reward: -75.99 | Epsilon: 0.856
Episode   40 | Reward: -110.99 | Epsilon: 0.814
Episode   50 | Reward: -2.84 | Epsilon: 0.774
Episode   60 | Reward: -85.87 | Epsilon: 0.737
Episode   70 | Reward: -176.74 | Epsilon: 0.701
Episode   80 | Reward: -103.17 | Epsilon: 0.666
Episode   90 | Reward: -97.18 | Epsilon: 0.634
Episode  100 | Reward: -20.66 | Epsilon: 0.603
Episode  110 | Reward: -63.47 | Epsilon: 0.573
Episode  120 | Reward: -142.51 | Epsilon: 0.545
Episode  130 | Reward: -17.94 | Epsilon: 0.519
Episode  140 | Reward: 37.02 | Epsilon: 0.493
Episode  150 | Reward: 32.57 | Epsilon: 0.469
Episode  160 | Reward: -279.97 | Epsilon: 0.446
Episode  170 | Reward: -178.95 | Epsilon: 0.424
Episode  180 | Reward: -37.67 | Epsilon: 0.404
Episode  190 | Reward: -46.31 | Epsilon: 0.384
Episode  200 | Reward: -216.54 | Epsilon: 0.365
Episode  210 | Reward: 10.51 | Epsilon: 0.347
Episode  220 | Reward: -159.80 | Epsilon: 0.330
Episode  230 | Reward: 18.50 | Epsilon: 0.314
Episode  240 | Reward: 82.85 | Epsilon: 0.299
Episode  250 | Reward: 4.32 | Epsilon: 0.284
Episode  260 | Reward: -50.14 | Epsilon: 0.270
Episode  270 | Reward: -132.40 | Epsilon: 0.257
Episode  280 | Reward: -30.55 | Epsilon: 0.245
Episode  290 | Reward: -30.97 | Epsilon: 0.233
Episode  300 | Reward: -73.80 | Epsilon: 0.221
Episode  310 | Reward: -86.92 | Epsilon: 0.210
Episode  320 | Reward: -16.05 | Epsilon: 0.200
Episode  330 | Reward: -55.63 | Epsilon: 0.190
Episode  340 | Reward: -65.27 | Epsilon: 0.181
Episode  350 | Reward: -62.79 | Epsilon: 0.172
Episode  360 | Reward: -150.21 | Epsilon: 0.164
Episode  370 | Reward: -68.10 | Epsilon: 0.156
Episode  380 | Reward: -38.12 | Epsilon: 0.148
Episode  390 | Reward: 140.37 | Epsilon: 0.141
Episode  400 | Reward: -38.08 | Epsilon: 0.134
Episode  410 | Reward: 71.95 | Epsilon: 0.127
Episode  420 | Reward: -17.68 | Epsilon: 0.121
Episode  430 | Reward: -25.99 | Epsilon: 0.115
Episode  440 | Reward: -21.21 | Epsilon: 0.110
Episode  450 | Reward: -72.67 | Epsilon: 0.104
Episode  460 | Reward: -28.86 | Epsilon: 0.099
Episode  470 | Reward: -131.17 | Epsilon: 0.094
Episode  480 | Reward: -84.50 | Epsilon: 0.090
Episode  490 | Reward: -74.76 | Epsilon: 0.085

 Running config: lr=0.0005, gamma=0.95, eps_decay=0.995, seed=123
Episode    0 | Reward: -418.96 | Epsilon: 0.995
Episode   10 | Reward: -73.72 | Epsilon: 0.946
Episode   20 | Reward: -135.36 | Epsilon: 0.900
Episode   30 | Reward: -249.30 | Epsilon: 0.856
Episode   40 | Reward: -110.28 | Epsilon: 0.814
Episode   50 | Reward: -92.51 | Epsilon: 0.774
Episode   60 | Reward: -80.16 | Epsilon: 0.737
Episode   70 | Reward: 10.67 | Epsilon: 0.701
Episode   80 | Reward: -58.68 | Epsilon: 0.666
Episode   90 | Reward: -349.84 | Epsilon: 0.634
Episode  100 | Reward: -39.38 | Epsilon: 0.603
Episode  110 | Reward: -139.75 | Epsilon: 0.573
```

```
Episode  120 | Reward: -29.99 | Epsilon: 0.545
Episode  130 | Reward: -46.85 | Epsilon: 0.519
Episode  140 | Reward: -70.43 | Epsilon: 0.493
Episode  150 | Reward: -286.29 | Epsilon: 0.469
Episode  160 | Reward: -196.71 | Epsilon: 0.446
Episode  170 | Reward: -95.44 | Epsilon: 0.424
Episode  180 | Reward: -2.66 | Epsilon: 0.404
Episode  190 | Reward: 69.77 | Epsilon: 0.384
Episode  200 | Reward: -153.36 | Epsilon: 0.365
Episode  210 | Reward: -66.36 | Epsilon: 0.347
Episode  220 | Reward: -7.57 | Epsilon: 0.330
Episode  230 | Reward: -63.87 | Epsilon: 0.314
Episode  240 | Reward: -152.46 | Epsilon: 0.299
Episode  250 | Reward: -67.35 | Epsilon: 0.284
Episode  260 | Reward: 46.56 | Epsilon: 0.270
Episode  270 | Reward: -58.43 | Epsilon: 0.257
Episode  280 | Reward: 19.95 | Epsilon: 0.245
Episode  290 | Reward: 63.71 | Epsilon: 0.233
Episode  300 | Reward: -95.72 | Epsilon: 0.221
Episode  310 | Reward: -101.77 | Epsilon: 0.210
Episode  320 | Reward: -70.90 | Epsilon: 0.200
Episode  330 | Reward: -61.91 | Epsilon: 0.190
Episode  340 | Reward: -97.57 | Epsilon: 0.181
Episode  350 | Reward: -102.36 | Epsilon: 0.172
Episode  360 | Reward: -84.15 | Epsilon: 0.164
Episode  370 | Reward: -50.93 | Epsilon: 0.156
Episode  380 | Reward: -76.61 | Epsilon: 0.148
Episode  390 | Reward: -115.95 | Epsilon: 0.141
Episode  400 | Reward: -164.49 | Epsilon: 0.134
Episode  410 | Reward: -62.21 | Epsilon: 0.127
Episode  420 | Reward: -27.62 | Epsilon: 0.121
Episode  430 | Reward: -110.05 | Epsilon: 0.115
Episode  440 | Reward: -122.93 | Epsilon: 0.110
Episode  450 | Reward: -92.50 | Epsilon: 0.104
Episode  460 | Reward: -51.67 | Epsilon: 0.099
Episode  470 | Reward: -91.01 | Epsilon: 0.094
Episode  480 | Reward: -150.68 | Epsilon: 0.090
Episode  490 | Reward: -130.69 | Epsilon: 0.085

 Running config: lr=0.0001, gamma=0.99, eps_decay=0.99, seed=42
Episode    0 | Reward: -206.09 | Epsilon: 0.990
Episode   10 | Reward: -84.93 | Epsilon: 0.895
Episode   20 | Reward: -185.06 | Epsilon: 0.810
Episode   30 | Reward: -120.70 | Epsilon: 0.732
Episode   40 | Reward: -48.26 | Epsilon: 0.662
Episode   50 | Reward: -397.35 | Epsilon: 0.599
Episode   60 | Reward: -135.08 | Epsilon: 0.542
Episode   70 | Reward: -20.46 | Epsilon: 0.490
Episode   80 | Reward: -18.28 | Epsilon: 0.443
Episode   90 | Reward: -51.46 | Epsilon: 0.401
Episode  100 | Reward: -235.45 | Epsilon: 0.362
Episode  110 | Reward: -220.69 | Epsilon: 0.328
Episode  120 | Reward: -157.17 | Epsilon: 0.296
Episode  130 | Reward: -80.94 | Epsilon: 0.268
Episode  140 | Reward: -3.56 | Epsilon: 0.242
Episode  150 | Reward: -85.35 | Epsilon: 0.219
Episode  160 | Reward: -81.92 | Epsilon: 0.198
Episode  170 | Reward: -78.04 | Epsilon: 0.179
Episode  180 | Reward: -92.41 | Epsilon: 0.162
Episode  190 | Reward: -112.12 | Epsilon: 0.147
Episode  200 | Reward: -121.86 | Epsilon: 0.133
Episode  210 | Reward: -106.15 | Epsilon: 0.120
Episode  220 | Reward: -126.99 | Epsilon: 0.108
Episode  230 | Reward: -108.85 | Epsilon: 0.098
Episode  240 | Reward: -81.97 | Epsilon: 0.089
Episode  250 | Reward: -39.18 | Epsilon: 0.080
Episode  260 | Reward: -67.75 | Epsilon: 0.073
Episode  270 | Reward: -53.99 | Epsilon: 0.066
Episode  280 | Reward: -61.92 | Epsilon: 0.059
Episode  290 | Reward: -69.66 | Epsilon: 0.054
Episode  300 | Reward: 0.52 | Epsilon: 0.050
Episode  310 | Reward: -87.08 | Epsilon: 0.050
Episode  320 | Reward: -55.31 | Epsilon: 0.050
Episode  330 | Reward: -74.59 | Epsilon: 0.050
Episode  340 | Reward: -53.69 | Epsilon: 0.050
Episode  350 | Reward: -56.00 | Epsilon: 0.050
Episode  360 | Reward: 15.80 | Epsilon: 0.050
Episode  370 | Reward: 12.97 | Epsilon: 0.050
Episode  380 | Reward: -62.50 | Epsilon: 0.050
Episode  390 | Reward: -68.46 | Epsilon: 0.050
Episode  400 | Reward: -51.14 | Epsilon: 0.050
Episode  410 | Reward: -69.77 | Epsilon: 0.050
Episode  420 | Reward: -18.92 | Epsilon: 0.050
```

```
Episode  430 | Reward: -41.98 | Epsilon: 0.050
Episode  440 | Reward: -5.68 | Epsilon: 0.050
Episode  450 | Reward: 12.87 | Epsilon: 0.050
Episode  460 | Reward: -2.57 | Epsilon: 0.050
Episode  470 | Reward: -20.73 | Epsilon: 0.050
Episode  480 | Reward: -49.85 | Epsilon: 0.050
Episode  490 | Reward: -31.40 | Epsilon: 0.050

 Running config: lr=0.0001, gamma=0.99, eps_decay=0.99, seed=123
Episode    0 | Reward: -418.96 | Epsilon: 0.990
Episode   10 | Reward: -88.93 | Epsilon: 0.895
Episode   20 | Reward: -102.73 | Epsilon: 0.810
Episode   30 | Reward: -214.83 | Epsilon: 0.732
Episode   40 | Reward: -152.39 | Epsilon: 0.662
Episode   50 | Reward: -243.30 | Epsilon: 0.599
Episode   60 | Reward: -226.13 | Epsilon: 0.542
Episode   70 | Reward: -131.54 | Epsilon: 0.490
Episode   80 | Reward: -179.54 | Epsilon: 0.443
Episode   90 | Reward: -27.61 | Epsilon: 0.401
Episode  100 | Reward: -110.78 | Epsilon: 0.362
Episode  110 | Reward: -172.75 | Epsilon: 0.328
Episode  120 | Reward: 12.35 | Epsilon: 0.296
Episode  130 | Reward: -71.90 | Epsilon: 0.268
Episode  140 | Reward: -36.78 | Epsilon: 0.242
Episode  150 | Reward: -27.77 | Epsilon: 0.219
Episode  160 | Reward: 89.08 | Epsilon: 0.198
Episode  170 | Reward: -81.64 | Epsilon: 0.179
Episode  180 | Reward: -143.76 | Epsilon: 0.162
Episode  190 | Reward: -39.89 | Epsilon: 0.147
Episode  200 | Reward: -36.71 | Epsilon: 0.133
Episode  210 | Reward: -148.27 | Epsilon: 0.120
Episode  220 | Reward: -114.69 | Epsilon: 0.108
Episode  230 | Reward: -49.32 | Epsilon: 0.098
Episode  240 | Reward: -66.94 | Epsilon: 0.089
Episode  250 | Reward: -53.67 | Epsilon: 0.080
Episode  260 | Reward: -57.78 | Epsilon: 0.073
Episode  270 | Reward: -11.88 | Epsilon: 0.066
Episode  280 | Reward: -82.26 | Epsilon: 0.059
Episode  290 | Reward: 8.93 | Epsilon: 0.054
Episode  300 | Reward: -23.88 | Epsilon: 0.050
Episode  310 | Reward: -15.33 | Epsilon: 0.050
Episode  320 | Reward: -51.72 | Epsilon: 0.050
Episode  330 | Reward: 27.15 | Epsilon: 0.050
Episode  340 | Reward: -23.22 | Epsilon: 0.050
Episode  350 | Reward: -108.22 | Epsilon: 0.050
Episode  360 | Reward: -24.89 | Epsilon: 0.050
Episode  370 | Reward: -6.62 | Epsilon: 0.050
Episode  380 | Reward: -101.25 | Epsilon: 0.050
Episode  390 | Reward: -90.35 | Epsilon: 0.050
Episode  400 | Reward: -48.52 | Epsilon: 0.050
Episode  410 | Reward: -83.22 | Epsilon: 0.050
Episode  420 | Reward: -299.74 | Epsilon: 0.050
Episode  430 | Reward: -25.40 | Epsilon: 0.050
Episode  440 | Reward: -24.23 | Epsilon: 0.050
Episode  450 | Reward: -11.88 | Epsilon: 0.050
Episode  460 | Reward: -17.04 | Epsilon: 0.050
Episode  470 | Reward: -1.17 | Epsilon: 0.050
Episode  480 | Reward: -11.11 | Epsilon: 0.050
Episode  490 | Reward: -6.13 | Epsilon: 0.050

 Running config: lr=0.0001, gamma=0.99, eps_decay=0.995, seed=42
Episode    0 | Reward: -206.09 | Epsilon: 0.995
Episode   10 | Reward: -151.28 | Epsilon: 0.946
Episode   20 | Reward: -58.78 | Epsilon: 0.900
Episode   30 | Reward: -178.72 | Epsilon: 0.856
Episode   40 | Reward: -132.15 | Epsilon: 0.814
Episode   50 | Reward: -284.80 | Epsilon: 0.774
Episode   60 | Reward: -99.13 | Epsilon: 0.737
Episode   70 | Reward: -40.35 | Epsilon: 0.701
Episode   80 | Reward: -15.02 | Epsilon: 0.666
Episode   90 | Reward: -126.65 | Epsilon: 0.634
Episode  100 | Reward: -151.98 | Epsilon: 0.603
Episode  110 | Reward: -45.67 | Epsilon: 0.573
Episode  120 | Reward: -10.49 | Epsilon: 0.545
Episode  130 | Reward: -28.41 | Epsilon: 0.519
Episode  140 | Reward: -84.53 | Epsilon: 0.493
Episode  150 | Reward: -85.78 | Epsilon: 0.469
Episode  160 | Reward: 15.39 | Epsilon: 0.446
Episode  170 | Reward: -3.42 | Epsilon: 0.424
Episode  180 | Reward: -27.24 | Epsilon: 0.404
Episode  190 | Reward: -219.87 | Epsilon: 0.384
Episode  200 | Reward: 37.33 | Epsilon: 0.365
Episode  210 | Reward: 16.02 | Epsilon: 0.347
```

```
Episode  220 | Reward: 23.51 | Epsilon: 0.330
Episode  230 | Reward: -39.53 | Epsilon: 0.314
Episode  240 | Reward: -124.76 | Epsilon: 0.299
Episode  250 | Reward: -130.43 | Epsilon: 0.284
Episode  260 | Reward: -52.67 | Epsilon: 0.270
Episode  270 | Reward: -96.21 | Epsilon: 0.257
Episode  280 | Reward: -96.08 | Epsilon: 0.245
Episode  290 | Reward: -22.85 | Epsilon: 0.233
Episode  300 | Reward: -0.75 | Epsilon: 0.221
Episode  310 | Reward: -19.36 | Epsilon: 0.210
Episode  320 | Reward: -33.48 | Epsilon: 0.200
Episode  330 | Reward: 17.16 | Epsilon: 0.190
Episode  340 | Reward: -124.18 | Epsilon: 0.181
Episode  350 | Reward: -84.71 | Epsilon: 0.172
Episode  360 | Reward: -89.68 | Epsilon: 0.164
Episode  370 | Reward: -83.49 | Epsilon: 0.156
Episode  380 | Reward: -72.28 | Epsilon: 0.148
Episode  390 | Reward: -98.03 | Epsilon: 0.141
Episode  400 | Reward: -41.63 | Epsilon: 0.134
Episode  410 | Reward: 14.89 | Epsilon: 0.127
Episode  420 | Reward: -64.25 | Epsilon: 0.121
Episode  430 | Reward: 154.29 | Epsilon: 0.115
Episode  440 | Reward: -51.69 | Epsilon: 0.110
Episode  450 | Reward: 4.79 | Epsilon: 0.104
Episode  460 | Reward: -46.01 | Epsilon: 0.099
Episode  470 | Reward: 238.45 | Epsilon: 0.094
Episode  480 | Reward: 244.80 | Epsilon: 0.090
Episode  490 | Reward: -83.12 | Epsilon: 0.085

 Running config: lr=0.0001, gamma=0.99, eps_decay=0.995, seed=123
Episode    0 | Reward: -418.96 | Epsilon: 0.995
Episode   10 | Reward: -114.55 | Epsilon: 0.946
Episode   20 | Reward: -156.56 | Epsilon: 0.900
Episode   30 | Reward: -165.83 | Epsilon: 0.856
Episode   40 | Reward: -240.93 | Epsilon: 0.814
Episode   50 | Reward: -120.83 | Epsilon: 0.774
Episode   60 | Reward: -292.09 | Epsilon: 0.737
Episode   70 | Reward: -123.78 | Epsilon: 0.701
Episode   80 | Reward: -149.20 | Epsilon: 0.666
Episode   90 | Reward: -16.72 | Epsilon: 0.634
Episode  100 | Reward: -37.16 | Epsilon: 0.603
Episode  110 | Reward: -4.34 | Epsilon: 0.573
Episode  120 | Reward: -64.05 | Epsilon: 0.545
Episode  130 | Reward: -88.60 | Epsilon: 0.519
Episode  140 | Reward: 7.83 | Epsilon: 0.493
Episode  150 | Reward: 1.99 | Epsilon: 0.469
Episode  160 | Reward: -43.50 | Epsilon: 0.446
Episode  170 | Reward: -105.31 | Epsilon: 0.424
Episode  180 | Reward: -6.39 | Epsilon: 0.404
Episode  190 | Reward: -200.82 | Epsilon: 0.384
Episode  200 | Reward: -9.17 | Epsilon: 0.365
Episode  210 | Reward: -42.13 | Epsilon: 0.347
Episode  220 | Reward: -42.55 | Epsilon: 0.330
Episode  230 | Reward: -14.99 | Epsilon: 0.314
Episode  240 | Reward: -10.80 | Epsilon: 0.299
Episode  250 | Reward: -63.28 | Epsilon: 0.284
Episode  260 | Reward: -89.88 | Epsilon: 0.270
Episode  270 | Reward: -42.92 | Epsilon: 0.257
Episode  280 | Reward: -19.32 | Epsilon: 0.245
Episode  290 | Reward: -19.40 | Epsilon: 0.233
Episode  300 | Reward: -47.70 | Epsilon: 0.221
Episode  310 | Reward: 13.37 | Epsilon: 0.210
Episode  320 | Reward: -19.95 | Epsilon: 0.200
Episode  330 | Reward: -91.13 | Epsilon: 0.190
Episode  340 | Reward: -37.86 | Epsilon: 0.181
Episode  350 | Reward: -26.94 | Epsilon: 0.172
Episode  360 | Reward: -37.25 | Epsilon: 0.164
Episode  370 | Reward: 4.12 | Epsilon: 0.156
Episode  380 | Reward: -61.37 | Epsilon: 0.148
Episode  390 | Reward: -43.20 | Epsilon: 0.141
Episode  400 | Reward: -50.80 | Epsilon: 0.134
Episode  410 | Reward: -52.40 | Epsilon: 0.127
Episode  420 | Reward: -80.45 | Epsilon: 0.121
Episode  430 | Reward: 1.07 | Epsilon: 0.115
Episode  440 | Reward: -46.67 | Epsilon: 0.110
Episode  450 | Reward: 262.57 | Epsilon: 0.104
Episode  460 | Reward: 33.12 | Epsilon: 0.099
Episode  470 | Reward: 189.88 | Epsilon: 0.094
Episode  480 | Reward: 229.29 | Epsilon: 0.090
Episode  490 | Reward: 198.32 | Epsilon: 0.085

 Running config: lr=0.0001, gamma=0.95, eps_decay=0.99, seed=42
Episode    0 | Reward: -206.09 | Epsilon: 0.990
```

```
Episode   10 | Reward: -110.27 | Epsilon: 0.895
Episode   20 | Reward: -40.97 | Epsilon: 0.810
Episode   30 | Reward: -139.55 | Epsilon: 0.732
Episode   40 | Reward: -337.03 | Epsilon: 0.662
Episode   50 | Reward: -134.74 | Epsilon: 0.599
Episode   60 | Reward: -45.64 | Epsilon: 0.542
Episode   70 | Reward: -95.13 | Epsilon: 0.490
Episode   80 | Reward: -205.21 | Epsilon: 0.443
Episode   90 | Reward: -8.74 | Epsilon: 0.401
Episode  100 | Reward: 8.81 | Epsilon: 0.362
Episode  110 | Reward: -7.10 | Epsilon: 0.328
Episode  120 | Reward: -87.95 | Epsilon: 0.296
Episode  130 | Reward: 0.86 | Epsilon: 0.268
Episode  140 | Reward: -206.51 | Epsilon: 0.242
Episode  150 | Reward: 49.49 | Epsilon: 0.219
Episode  160 | Reward: -2.85 | Epsilon: 0.198
Episode  170 | Reward: 178.49 | Epsilon: 0.179
Episode  180 | Reward: -141.88 | Epsilon: 0.162
Episode  190 | Reward: -26.65 | Epsilon: 0.147
Episode  200 | Reward: -137.56 | Epsilon: 0.133
Episode  210 | Reward: -72.66 | Epsilon: 0.120
Episode  220 | Reward: -93.99 | Epsilon: 0.108
Episode  230 | Reward: -104.78 | Epsilon: 0.098
Episode  240 | Reward: -100.37 | Epsilon: 0.089
Episode  250 | Reward: -170.95 | Epsilon: 0.080
Episode  260 | Reward: -71.48 | Epsilon: 0.073
Episode  270 | Reward: -88.90 | Epsilon: 0.066
Episode  280 | Reward: -64.23 | Epsilon: 0.059
Episode  290 | Reward: -99.05 | Epsilon: 0.054
Episode  300 | Reward: -87.46 | Epsilon: 0.050
Episode  310 | Reward: -97.22 | Epsilon: 0.050
Episode  320 | Reward: -126.80 | Epsilon: 0.050
Episode  330 | Reward: -81.81 | Epsilon: 0.050
Episode  340 | Reward: -80.88 | Epsilon: 0.050
Episode  350 | Reward: -93.17 | Epsilon: 0.050
Episode  360 | Reward: -39.30 | Epsilon: 0.050
Episode  370 | Reward: -123.32 | Epsilon: 0.050
Episode  380 | Reward: -63.84 | Epsilon: 0.050
Episode  390 | Reward: -129.65 | Epsilon: 0.050
Episode  400 | Reward: -20.59 | Epsilon: 0.050
Episode  410 | Reward: -45.52 | Epsilon: 0.050
Episode  420 | Reward: -43.68 | Epsilon: 0.050
Episode  430 | Reward: -58.06 | Epsilon: 0.050
Episode  440 | Reward: -98.25 | Epsilon: 0.050
Episode  450 | Reward: -79.39 | Epsilon: 0.050
Episode  460 | Reward: -27.04 | Epsilon: 0.050
Episode  470 | Reward: -27.11 | Epsilon: 0.050
Episode  480 | Reward: -100.23 | Epsilon: 0.050
Episode  490 | Reward: -59.41 | Epsilon: 0.050

 Running config: lr=0.0001, gamma=0.95, eps_decay=0.99, seed=123
Episode    0 | Reward: -418.96 | Epsilon: 0.990
Episode   10 | Reward: -88.93 | Epsilon: 0.895
Episode   20 | Reward: -197.23 | Epsilon: 0.810
Episode   30 | Reward: -82.11 | Epsilon: 0.732
Episode   40 | Reward: -5.18 | Epsilon: 0.662
Episode   50 | Reward: -114.87 | Epsilon: 0.599
Episode   60 | Reward: -117.32 | Epsilon: 0.542
Episode   70 | Reward: -55.86 | Epsilon: 0.490
Episode   80 | Reward: -103.76 | Epsilon: 0.443
Episode   90 | Reward: -15.74 | Epsilon: 0.401
Episode  100 | Reward: -67.57 | Epsilon: 0.362
Episode  110 | Reward: 39.33 | Epsilon: 0.328
Episode  120 | Reward: 23.47 | Epsilon: 0.296
Episode  130 | Reward: -14.52 | Epsilon: 0.268
Episode  140 | Reward: -69.19 | Epsilon: 0.242
Episode  150 | Reward: -233.71 | Epsilon: 0.219
Episode  160 | Reward: -108.30 | Epsilon: 0.198
Episode  170 | Reward: -94.49 | Epsilon: 0.179
Episode  180 | Reward: -144.06 | Epsilon: 0.162
Episode  190 | Reward: -21.25 | Epsilon: 0.147
Episode  200 | Reward: -71.64 | Epsilon: 0.133
Episode  210 | Reward: -126.44 | Epsilon: 0.120
Episode  220 | Reward: -83.85 | Epsilon: 0.108
Episode  230 | Reward: -119.29 | Epsilon: 0.098
Episode  240 | Reward: -44.99 | Epsilon: 0.089
Episode  250 | Reward: -28.48 | Epsilon: 0.080
Episode  260 | Reward: -43.91 | Epsilon: 0.073
Episode  270 | Reward: -56.52 | Epsilon: 0.066
Episode  280 | Reward: -113.24 | Epsilon: 0.059
Episode  290 | Reward: -96.32 | Epsilon: 0.054
Episode  300 | Reward: -74.23 | Epsilon: 0.050
Episode  310 | Reward: -78.13 | Epsilon: 0.050
```

```
Episode  320 | Reward: -21.89 | Epsilon: 0.050
Episode  330 | Reward: -52.48 | Epsilon: 0.050
Episode  340 | Reward: -111.83 | Epsilon: 0.050
Episode  350 | Reward: -67.32 | Epsilon: 0.050
Episode  360 | Reward: -11.79 | Epsilon: 0.050
Episode  370 | Reward: -58.13 | Epsilon: 0.050
Episode  380 | Reward: -27.64 | Epsilon: 0.050
Episode  390 | Reward: -119.44 | Epsilon: 0.050
Episode  400 | Reward: 18.80 | Epsilon: 0.050
Episode  410 | Reward: -43.37 | Epsilon: 0.050
Episode  420 | Reward: -67.09 | Epsilon: 0.050
Episode  430 | Reward: 69.78 | Epsilon: 0.050
Episode  440 | Reward: -1.75 | Epsilon: 0.050
Episode  450 | Reward: -80.99 | Epsilon: 0.050
Episode  460 | Reward: 162.22 | Epsilon: 0.050
Episode  470 | Reward: -89.71 | Epsilon: 0.050
Episode  480 | Reward: 161.97 | Epsilon: 0.050
Episode  490 | Reward: -52.70 | Epsilon: 0.050

 Running config: lr=0.0001, gamma=0.95, eps_decay=0.995, seed=42
Episode    0 | Reward: -206.09 | Epsilon: 0.995
Episode   10 | Reward: -120.68 | Epsilon: 0.946
Episode   20 | Reward: -88.70 | Epsilon: 0.900
Episode   30 | Reward: -258.83 | Epsilon: 0.856
Episode   40 | Reward: -209.60 | Epsilon: 0.814
Episode   50 | Reward: -43.93 | Epsilon: 0.774
Episode   60 | Reward: -107.15 | Epsilon: 0.737
Episode   70 | Reward: -21.88 | Epsilon: 0.701
Episode   80 | Reward: -132.84 | Epsilon: 0.666
Episode   90 | Reward: -16.57 | Epsilon: 0.634
Episode  100 | Reward: -135.65 | Epsilon: 0.603
Episode  110 | Reward: 5.72 | Epsilon: 0.573
Episode  120 | Reward: -107.42 | Epsilon: 0.545
Episode  130 | Reward: -42.70 | Epsilon: 0.519
Episode  140 | Reward: -206.41 | Epsilon: 0.493
Episode  150 | Reward: -83.33 | Epsilon: 0.469
Episode  160 | Reward: -245.80 | Epsilon: 0.446
Episode  170 | Reward: -46.25 | Epsilon: 0.424
Episode  180 | Reward: 43.22 | Epsilon: 0.404
Episode  190 | Reward: 40.99 | Epsilon: 0.384
Episode  200 | Reward: -121.92 | Epsilon: 0.365
Episode  210 | Reward: -97.10 | Epsilon: 0.347
Episode  220 | Reward: -194.70 | Epsilon: 0.330
Episode  230 | Reward: 23.65 | Epsilon: 0.314
Episode  240 | Reward: -20.32 | Epsilon: 0.299
Episode  250 | Reward: -42.75 | Epsilon: 0.284
Episode  260 | Reward: -105.28 | Epsilon: 0.270
Episode  270 | Reward: -40.40 | Epsilon: 0.257
Episode  280 | Reward: 5.21 | Epsilon: 0.245
Episode  290 | Reward: -45.12 | Epsilon: 0.233
Episode  300 | Reward: -84.95 | Epsilon: 0.221
Episode  310 | Reward: -99.56 | Epsilon: 0.210
Episode  320 | Reward: 16.66 | Epsilon: 0.200
Episode  330 | Reward: -62.69 | Epsilon: 0.190
Episode  340 | Reward: -126.94 | Epsilon: 0.181
Episode  350 | Reward: -93.11 | Epsilon: 0.172
Episode  360 | Reward: -47.49 | Epsilon: 0.164
Episode  370 | Reward: 238.93 | Epsilon: 0.156
Episode  380 | Reward: 54.54 | Epsilon: 0.148
Episode  390 | Reward: -96.54 | Epsilon: 0.141
Episode  400 | Reward: -58.03 | Epsilon: 0.134
Episode  410 | Reward: 187.06 | Epsilon: 0.127
Episode  420 | Reward: 170.33 | Epsilon: 0.121
Episode  430 | Reward: -63.32 | Epsilon: 0.115
Episode  440 | Reward: -114.49 | Epsilon: 0.110
Episode  450 | Reward: -78.86 | Epsilon: 0.104
Episode  460 | Reward: -81.97 | Epsilon: 0.099
Episode  470 | Reward: -81.63 | Epsilon: 0.094
Episode  480 | Reward: -28.84 | Epsilon: 0.090
Episode  490 | Reward: 180.11 | Epsilon: 0.085

 Running config: lr=0.0001, gamma=0.95, eps_decay=0.995, seed=123
Episode    0 | Reward: -418.96 | Epsilon: 0.995
Episode   10 | Reward: -114.55 | Epsilon: 0.946
Episode   20 | Reward: -105.20 | Epsilon: 0.900
Episode   30 | Reward: -158.35 | Epsilon: 0.856
Episode   40 | Reward: -220.29 | Epsilon: 0.814
Episode   50 | Reward: -91.52 | Epsilon: 0.774
Episode   60 | Reward: -104.33 | Epsilon: 0.737
Episode   70 | Reward: -171.77 | Epsilon: 0.701
Episode   80 | Reward: -173.71 | Epsilon: 0.666
Episode   90 | Reward: -103.09 | Epsilon: 0.634
Episode  100 | Reward: -125.32 | Epsilon: 0.603
```

```
Episode  110 | Reward: -64.32 | Epsilon: 0.573
Episode  120 | Reward: -128.22 | Epsilon: 0.545
Episode  130 | Reward: -65.26 | Epsilon: 0.519
Episode  140 | Reward: 14.99 | Epsilon: 0.493
Episode  150 | Reward: -242.76 | Epsilon: 0.469
Episode  160 | Reward: -91.08 | Epsilon: 0.446
Episode  170 | Reward: -14.80 | Epsilon: 0.424
Episode  180 | Reward: -55.51 | Epsilon: 0.404
Episode  190 | Reward: -31.42 | Epsilon: 0.384
Episode  200 | Reward: -65.75 | Epsilon: 0.365
Episode  210 | Reward: -88.01 | Epsilon: 0.347
Episode  220 | Reward: -6.58 | Epsilon: 0.330
Episode  230 | Reward: 36.43 | Epsilon: 0.314
Episode  240 | Reward: -10.49 | Epsilon: 0.299
Episode  250 | Reward: 74.18 | Epsilon: 0.284
Episode  260 | Reward: -76.76 | Epsilon: 0.270
Episode  270 | Reward: -69.46 | Epsilon: 0.257
Episode  280 | Reward: -86.06 | Epsilon: 0.245
Episode  290 | Reward: 37.03 | Epsilon: 0.233
Episode  300 | Reward: -72.88 | Epsilon: 0.221
Episode  310 | Reward: -57.51 | Epsilon: 0.210
Episode  320 | Reward: -77.76 | Epsilon: 0.200
Episode  330 | Reward: -76.80 | Epsilon: 0.190
Episode  340 | Reward: -60.18 | Epsilon: 0.181
Episode  350 | Reward: -113.83 | Epsilon: 0.172
Episode  360 | Reward: -35.35 | Epsilon: 0.164
Episode  370 | Reward: -35.44 | Epsilon: 0.156
Episode  380 | Reward: 41.81 | Epsilon: 0.148
Episode  390 | Reward: -64.10 | Epsilon: 0.141
Episode  400 | Reward: -28.81 | Epsilon: 0.134
Episode  410 | Reward: -99.82 | Epsilon: 0.127
Episode  420 | Reward: -9.91 | Epsilon: 0.121
Episode  430 | Reward: -8.61 | Epsilon: 0.115
Episode  440 | Reward: -14.92 | Epsilon: 0.110
Episode  450 | Reward: -78.04 | Epsilon: 0.104
Episode  460 | Reward: -152.92 | Epsilon: 0.099
Episode  470 | Reward: -82.54 | Epsilon: 0.094
Episode  480 | Reward: -90.32 | Epsilon: 0.090
Episode  490 | Reward: -43.53 | Epsilon: 0.085
```

Results Summary

We sort all tested configurations by average final reward (last 100 episodes) and compare them to identify the most effective combinations.

In [13]:
```python
import pandas as pd
df_grid = pd.DataFrame(grid_results)
df_sorted = df_grid.sort_values(by="avg_reward", ascending=False).reset_index(drop=True)
df_sorted.head(10)
```

Out[13]:

	lr	gamma	eps_decay	seed	avg_reward	std_dev	duration_sec
0	0.0005	0.99	0.990	42	140.703416	112.061120	3634.382368
1	0.0001	0.99	0.995	123	80.863015	107.208706	3513.435068
2	0.0010	0.99	0.990	123	71.288781	137.173763	3466.417691
3	0.0005	0.99	0.995	42	63.842453	109.502094	2999.978555
4	0.0001	0.99	0.995	42	50.166372	104.169694	3395.397919
5	0.0010	0.99	0.990	42	47.222137	140.701891	3288.428832
6	0.0010	0.99	0.995	123	17.472470	154.431482	2990.609757
7	0.0010	0.99	0.995	42	2.729524	99.865054	3013.518514
8	0.0001	0.95	0.995	42	-15.707685	100.380314	3302.664332
9	0.0001	0.99	0.990	42	-15.955816	38.007294	4292.460358

Group and Average Across Seeds

In [14]:
```python
import pandas as pd

# Convert list of dictionaries to DataFrame
df_grid = pd.DataFrame(grid_results)

# Group by hyperparameter combination
grouped = df_grid.groupby(['lr', 'gamma', 'eps_decay']).agg({
    'avg_reward': 'mean',
    'std_dev': 'mean',
    'duration_sec': 'mean'
```

```python
}).reset_index()

# Sort by best average reward
grouped = grouped.sort_values(by='avg_reward', ascending=False).reset_index(drop=True)

# Display top results
grouped.head(10)
```

Out[14]:

	lr	gamma	eps_decay	avg_reward	std_dev	duration_sec
0	0.0001	0.99	0.995	65.514693	105.689200	3454.416494
1	0.0010	0.99	0.990	59.255459	138.937827	3377.423262
2	0.0005	0.99	0.990	45.626943	100.523673	3811.938594
3	0.0005	0.99	0.995	16.644817	91.807266	3280.342795
4	0.0010	0.99	0.995	10.100997	127.148268	3002.064135
5	0.0001	0.99	0.990	-23.264571	40.931198	4313.485603
6	0.0001	0.95	0.995	-35.133470	79.215033	3300.800516
7	0.0010	0.95	0.995	-52.771130	66.708893	3082.342064
8	0.0001	0.95	0.990	-52.866462	63.318945	4307.575562
9	0.0005	0.95	0.990	-54.036470	67.344576	4188.730985

Aggregated Results Across Seeds

To improve the robustness of our conclusions, we aggregate results across both tested random seeds. We compute the mean of:

- Final average reward (last 100 episodes),
- Standard deviation (stability),
- Training time (efficiency),

for each unique combination of:

- **Learning Rate**
- **Discount Factor (γ)**
- **Epsilon Decay Rate**

This allows us to identify strong general-performing hyperparameter settings, independent of randomness.

Task 9: Apply RLlib Algorithm to Atari Environment

We applied the **Deep Q-Network (DQN)** algorithm from **Ray RLlib** to the **ALE/Pong-ram-v5** environment, using Gymnasium's Atari integration.

Why DQN?

- DQN is well-suited for environments with **discrete action spaces** like Atari games.
- It uses deep neural networks to approximate Q-values, enabling it to learn effective policies directly from raw RAM inputs.
- RLlib's implementation supports scalable and modular training with easy integration of exploration strategies.

□ Why Pong-ram?

- Unlike image-based Pong variants, `Pong-ram-v5` uses a compact 128-byte RAM state representation, allowing **faster training** and **lower computational overhead**.
- The environment retains the **same game mechanics and reward structure** as visual Pong, making it an effective benchmark.
- It enables testing of learning dynamics without convolutional networks, making it ideal for MLP-based DQN agents.

⚙ Setup Summary

- **Algorithm**: DQN
- **Environment**: ALE/Pong-ram-v5
- **Training Episodes**: 500
- **Model Architecture**: MLP with two hidden layers (256 units, ReLU activation)
- **Exploration**: Epsilon-Greedy (from 1.0 → 0.01 over 200k steps)
- **Learning Rate**: 1e-4
- **Discount Factor**: 0.99
- **Batch Size**: 32
- **Framework**: PyTorch (`framework="torch"`)
- **Rollout Workers**: 0 (single-process training for compatibility with notebooks)

The agent was trained using a basic DQN setup with no dueling or double Q enhancements to evaluate the baseline performance on RAM-based Atari inputs.

In [1]:
```python
import gymnasium as gym
import ray
import matplotlib.pyplot as plt
from ray.rllib.algorithms.dqn import DQNConfig
```

```
2025-05-11 03:22:44,879 WARNING deprecation.py:50 -- DeprecationWarning: `DirectStepOptimizer` has been deprecat
ed. This will raise an error in the future!
2025-05-11 03:22:46,259 WARNING deprecation.py:50 -- DeprecationWarning: `build_tf_policy` has been deprecated.
This will raise an error in the future!
2025-05-11 03:22:46,267 WARNING deprecation.py:50 -- DeprecationWarning: `build_policy_class` has been deprecate
d. This will raise an error in the future!
```

In [2]:
```python
config = (
    DQNConfig()
    .environment(env="ALE/Pong-ram-v5")  # Gymnasium Atari env format
    .framework("torch")
    .resources(num_gpus=0)  # Set to 1 if CUDA GPU available and torch w/ CUDA installed
    .rollouts(num_rollout_workers=0)  # Use 1 or more if not on Jupyter
    .training(
        gamma=0.99,
        lr=1e-4,
        train_batch_size=32,
        model={"fcnet_hiddens": [256], "fcnet_activation": "relu"},
        num_steps_sampled_before_learning_starts=1000
    )
    .exploration(
        exploration_config={
            "type": "EpsilonGreedy",
            "initial_epsilon": 1.0,
            "final_epsilon": 0.01,
            "epsilon_timesteps": 200_000
        }
    )
    .debugging(log_level="WARN")
)
```

In [3]:
```python
from ray.rllib.algorithms.dqn import DQN
```

```python
algo = config.build()
avg_rewards = []

for episode in range(500):
    result = algo.train()
    reward = result["episode_reward_mean"]
    print(f"Episode {episode} - Avg Reward: {reward:.2f}")
    avg_rewards.append(reward)
```

```
2025-05-11 03:22:46,347 WARNING deprecation.py:50 -- DeprecationWarning: `rllib/algorithms/simple_q/` has been deprecated. Use `rllib_contrib/simple_q/` instead. This will raise an error in the future!
C:\Users\elias\anaconda3\envs\rllib-atari\lib\site-packages\ray\rllib\algorithms\algorithm.py:484: RayDeprecationWarning: This API is deprecated and may be removed in future Ray releases. You could suppress this warning by setting env variable PYTHONWARNINGS="ignore::DeprecationWarning"
`UnifiedLogger` will be removed in Ray 2.7.
  return UnifiedLogger(config, logdir, loggers=None)
C:\Users\elias\anaconda3\envs\rllib-atari\lib\site-packages\ray\tune\logger\unified.py:53: RayDeprecationWarning: This API is deprecated and may be removed in future Ray releases. You could suppress this warning by setting env variable PYTHONWARNINGS="ignore::DeprecationWarning"
The `JsonLogger interface is deprecated in favor of the `ray.tune.json.JsonLoggerCallback` interface and will be removed in Ray 2.7.
  self._loggers.append(cls(self.config, self.logdir, self.trial))
C:\Users\elias\anaconda3\envs\rllib-atari\lib\site-packages\ray\tune\logger\unified.py:53: RayDeprecationWarning: This API is deprecated and may be removed in future Ray releases. You could suppress this warning by setting env variable PYTHONWARNINGS="ignore::DeprecationWarning"
The `CSVLogger interface is deprecated in favor of the `ray.tune.csv.CSVLoggerCallback` interface and will be removed in Ray 2.7.
  self._loggers.append(cls(self.config, self.logdir, self.trial))
C:\Users\elias\anaconda3\envs\rllib-atari\lib\site-packages\ray\tune\logger\unified.py:53: RayDeprecationWarning: This API is deprecated and may be removed in future Ray releases. You could suppress this warning by setting env variable PYTHONWARNINGS="ignore::DeprecationWarning"
The `TBXLogger interface is deprecated in favor of the `ray.tune.tensorboardx.TBXLoggerCallback` interface and will be removed in Ray 2.7.
  self._loggers.append(cls(self.config, self.logdir, self.trial))
C:\Users\elias\anaconda3\envs\rllib-atari\lib\site-packages\gym\utils\passive_env_checker.py:233: DeprecationWarning: `np.bool8` is a deprecated alias for `np.bool_`.  (Deprecated NumPy 1.24)
  if not isinstance(terminated, (bool, np.bool8)):
2025-05-11 03:22:46,645 WARNING deprecation.py:50 -- DeprecationWarning: `LearningRateSchedule` has been deprecated. This will raise an error in the future!
2025-05-11 03:22:46,649 WARNING deprecation.py:50 -- DeprecationWarning: `ray.rllib.models.torch.fcnet.FullyConnectedNetwork` has been deprecated. This will raise an error in the future!
2025-05-11 03:22:46,649 WARNING deprecation.py:50 -- DeprecationWarning: `ray.rllib.models.torch.torch_modelv2.TorchModelV2` has been deprecated. Use `ray.rllib.core.rl_module.rl_module.RLModule` instead. This will raise an error in the future!
2025-05-11 03:22:46,669 WARNING deprecation.py:50 -- DeprecationWarning: `ray.rllib.models.torch.torch_action_dist.get_torch_categorical_class_with_temperature` has been deprecated. Use `ray.rllib.models.torch.torch_distributions.TorchCategorical` instead. This will raise an error in the future!
2025-05-11 03:22:46,673 WARNING deprecation.py:50 -- DeprecationWarning: `TorchPolicy` has been deprecated. This will raise an error in the future!
2025-05-11 03:22:46,673 WARNING deprecation.py:50 -- DeprecationWarning: `EpsilonGreedy` has been deprecated. This will raise an error in the future!
2025-05-11 03:22:46,673 WARNING deprecation.py:50 -- DeprecationWarning: `Exploration` has been deprecated. This will raise an error in the future!
2025-05-11 03:22:50,157 WARNING deprecation.py:50 -- DeprecationWarning: `TargetNetworkMixin` has been deprecated. This will raise an error in the future!
2025-05-11 03:22:50,168 WARNING deprecation.py:50 -- DeprecationWarning: `ray.rllib.models.torch.torch_action_dist.TorchDistributionWrapper` has been deprecated. Use `ray.rllib.models.torch.torch_distributions.TorchCategorical` instead. This will raise an error in the future!
2025-05-11 03:22:50,407 WARNING util.py:68 -- Install gputil for GPU system monitoring.
2025-05-11 03:23:00,765 WARNING deprecation.py:50 -- DeprecationWarning: `ray.rllib.execution.train_ops.multi_gpu_train_one_step` has been deprecated. This will raise an error in the future!
Episode 0 - Avg Reward: -21.00
Episode 1 - Avg Reward: -21.00
Episode 2 - Avg Reward: -21.00
Episode 3 - Avg Reward: -21.00
Episode 4 - Avg Reward: -21.00
Episode 5 - Avg Reward: -21.00
Episode 6 - Avg Reward: -20.86
Episode 7 - Avg Reward: -20.62
Episode 8 - Avg Reward: -20.56
Episode 9 - Avg Reward: -20.60
Episode 10 - Avg Reward: -20.64
Episode 11 - Avg Reward: -20.67
Episode 12 - Avg Reward: -20.54
Episode 13 - Avg Reward: -20.60
Episode 14 - Avg Reward: -20.62
Episode 15 - Avg Reward: -20.62
Episode 16 - Avg Reward: -20.53
Episode 17 - Avg Reward: -20.56
Episode 18 - Avg Reward: -20.58
Episode 19 - Avg Reward: -20.60
Episode 20 - Avg Reward: -20.62
Episode 21 - Avg Reward: -20.64
```

```
Episode 22 - Avg Reward: -20.57
Episode 23 - Avg Reward: -20.54
Episode 24 - Avg Reward: -20.56
Episode 25 - Avg Reward: -20.54
Episode 26 - Avg Reward: -20.56
Episode 27 - Avg Reward: -20.54
Episode 28 - Avg Reward: -20.48
Episode 29 - Avg Reward: -20.50
Episode 30 - Avg Reward: -20.52
Episode 31 - Avg Reward: -20.53
Episode 32 - Avg Reward: -20.53
Episode 33 - Avg Reward: -20.52
Episode 34 - Avg Reward: -20.50
Episode 35 - Avg Reward: -20.51
Episode 36 - Avg Reward: -20.42
Episode 37 - Avg Reward: -20.41
Episode 38 - Avg Reward: -20.37
Episode 39 - Avg Reward: -20.37
Episode 40 - Avg Reward: -20.36
Episode 41 - Avg Reward: -20.35
Episode 42 - Avg Reward: -20.29
Episode 43 - Avg Reward: -20.26
Episode 44 - Avg Reward: -20.26
Episode 45 - Avg Reward: -20.21
Episode 46 - Avg Reward: -20.18
Episode 47 - Avg Reward: -20.16
Episode 48 - Avg Reward: -20.11
Episode 49 - Avg Reward: -20.11
Episode 50 - Avg Reward: -20.06
Episode 51 - Avg Reward: -20.00
Episode 52 - Avg Reward: -20.02
Episode 53 - Avg Reward: -20.00
Episode 54 - Avg Reward: -20.00
Episode 55 - Avg Reward: -19.92
Episode 56 - Avg Reward: -19.92
Episode 57 - Avg Reward: -19.94
Episode 58 - Avg Reward: -19.91
Episode 59 - Avg Reward: -19.91
Episode 60 - Avg Reward: -19.87
Episode 61 - Avg Reward: -19.89
Episode 62 - Avg Reward: -19.82
Episode 63 - Avg Reward: -19.83
Episode 64 - Avg Reward: -19.83
Episode 65 - Avg Reward: -19.76
Episode 66 - Avg Reward: -19.78
Episode 67 - Avg Reward: -19.80
Episode 68 - Avg Reward: -19.80
Episode 69 - Avg Reward: -19.79
Episode 70 - Avg Reward: -19.75
Episode 71 - Avg Reward: -19.75
Episode 72 - Avg Reward: -19.75
Episode 73 - Avg Reward: -19.74
Episode 74 - Avg Reward: -19.74
Episode 75 - Avg Reward: -19.73
Episode 76 - Avg Reward: -19.73
Episode 77 - Avg Reward: -19.74
Episode 78 - Avg Reward: -19.71
Episode 79 - Avg Reward: -19.70
Episode 80 - Avg Reward: -19.70
Episode 81 - Avg Reward: -19.70
Episode 82 - Avg Reward: -19.68
Episode 83 - Avg Reward: -19.68
Episode 84 - Avg Reward: -19.68
Episode 85 - Avg Reward: -19.68
Episode 86 - Avg Reward: -19.68
Episode 87 - Avg Reward: -19.68
Episode 88 - Avg Reward: -19.66
Episode 89 - Avg Reward: -19.66
Episode 90 - Avg Reward: -19.62
Episode 91 - Avg Reward: -19.59
Episode 92 - Avg Reward: -19.59
Episode 93 - Avg Reward: -19.53
Episode 94 - Avg Reward: -19.53
Episode 95 - Avg Reward: -19.51
Episode 96 - Avg Reward: -19.48
Episode 97 - Avg Reward: -19.48
Episode 98 - Avg Reward: -19.44
Episode 99 - Avg Reward: -19.43
Episode 100 - Avg Reward: -19.43
Episode 101 - Avg Reward: -19.40
Episode 102 - Avg Reward: -19.40
Episode 103 - Avg Reward: -19.39
Episode 104 - Avg Reward: -19.36
```

```
Episode 105 - Avg Reward: -19.36
Episode 106 - Avg Reward: -19.36
Episode 107 - Avg Reward: -19.36
Episode 108 - Avg Reward: -19.33
Episode 109 - Avg Reward: -19.31
Episode 110 - Avg Reward: -19.31
Episode 111 - Avg Reward: -19.30
Episode 112 - Avg Reward: -19.30
Episode 113 - Avg Reward: -19.29
Episode 114 - Avg Reward: -19.29
Episode 115 - Avg Reward: -19.24
Episode 116 - Avg Reward: -19.24
Episode 117 - Avg Reward: -19.22
Episode 118 - Avg Reward: -19.22
Episode 119 - Avg Reward: -19.18
Episode 120 - Avg Reward: -19.17
Episode 121 - Avg Reward: -19.17
Episode 122 - Avg Reward: -19.15
Episode 123 - Avg Reward: -19.15
Episode 124 - Avg Reward: -19.10
Episode 125 - Avg Reward: -19.10
Episode 126 - Avg Reward: -19.05
Episode 127 - Avg Reward: -19.05
Episode 128 - Avg Reward: -19.04
Episode 129 - Avg Reward: -19.04
Episode 130 - Avg Reward: -19.01
Episode 131 - Avg Reward: -19.01
Episode 132 - Avg Reward: -19.01
Episode 133 - Avg Reward: -18.93
Episode 134 - Avg Reward: -18.89
Episode 135 - Avg Reward: -18.89
Episode 136 - Avg Reward: -18.84
Episode 137 - Avg Reward: -18.84
Episode 138 - Avg Reward: -18.79
Episode 139 - Avg Reward: -18.79
Episode 140 - Avg Reward: -18.71
Episode 141 - Avg Reward: -18.71
Episode 142 - Avg Reward: -18.66
Episode 143 - Avg Reward: -18.66
Episode 144 - Avg Reward: -18.61
Episode 145 - Avg Reward: -18.61
Episode 146 - Avg Reward: -18.61
Episode 147 - Avg Reward: -18.61
Episode 148 - Avg Reward: -18.56
Episode 149 - Avg Reward: -18.56
Episode 150 - Avg Reward: -18.56
Episode 151 - Avg Reward: -18.51
Episode 152 - Avg Reward: -18.51
Episode 153 - Avg Reward: -18.44
Episode 154 - Avg Reward: -18.44
Episode 155 - Avg Reward: -18.38
Episode 156 - Avg Reward: -18.38
Episode 157 - Avg Reward: -18.37
Episode 158 - Avg Reward: -18.37
Episode 159 - Avg Reward: -18.33
Episode 160 - Avg Reward: -18.33
Episode 161 - Avg Reward: -18.27
Episode 162 - Avg Reward: -18.27
Episode 163 - Avg Reward: -18.22
Episode 164 - Avg Reward: -18.22
Episode 165 - Avg Reward: -18.22
Episode 166 - Avg Reward: -18.13
Episode 167 - Avg Reward: -18.13
Episode 168 - Avg Reward: -18.02
Episode 169 - Avg Reward: -18.02
Episode 170 - Avg Reward: -18.02
Episode 171 - Avg Reward: -17.94
Episode 172 - Avg Reward: -17.94
Episode 173 - Avg Reward: -17.94
Episode 174 - Avg Reward: -17.80
Episode 175 - Avg Reward: -17.80
Episode 176 - Avg Reward: -17.80
Episode 177 - Avg Reward: -17.74
Episode 178 - Avg Reward: -17.74
Episode 179 - Avg Reward: -17.69
Episode 180 - Avg Reward: -17.69
Episode 181 - Avg Reward: -17.69
Episode 182 - Avg Reward: -17.63
Episode 183 - Avg Reward: -17.63
Episode 184 - Avg Reward: -17.63
Episode 185 - Avg Reward: -17.54
Episode 186 - Avg Reward: -17.54
Episode 187 - Avg Reward: -17.54
```

```
Episode 188 - Avg Reward: -17.48
Episode 189 - Avg Reward: -17.48
Episode 190 - Avg Reward: -17.48
Episode 191 - Avg Reward: -17.38
Episode 192 - Avg Reward: -17.38
Episode 193 - Avg Reward: -17.38
Episode 194 - Avg Reward: -17.29
Episode 195 - Avg Reward: -17.29
Episode 196 - Avg Reward: -17.20
Episode 197 - Avg Reward: -17.20
Episode 198 - Avg Reward: -17.20
Episode 199 - Avg Reward: -17.20
Episode 200 - Avg Reward: -17.09
Episode 201 - Avg Reward: -17.09
Episode 202 - Avg Reward: -17.06
Episode 203 - Avg Reward: -17.06
Episode 204 - Avg Reward: -17.06
Episode 205 - Avg Reward: -17.02
Episode 206 - Avg Reward: -17.02
Episode 207 - Avg Reward: -17.02
Episode 208 - Avg Reward: -16.97
Episode 209 - Avg Reward: -16.97
Episode 210 - Avg Reward: -16.97
Episode 211 - Avg Reward: -16.88
Episode 212 - Avg Reward: -16.88
Episode 213 - Avg Reward: -16.88
Episode 214 - Avg Reward: -16.83
Episode 215 - Avg Reward: -16.83
Episode 216 - Avg Reward: -16.83
Episode 217 - Avg Reward: -16.68
Episode 218 - Avg Reward: -16.68
Episode 219 - Avg Reward: -16.68
Episode 220 - Avg Reward: -16.63
Episode 221 - Avg Reward: -16.63
Episode 222 - Avg Reward: -16.58
Episode 223 - Avg Reward: -16.58
Episode 224 - Avg Reward: -16.58
Episode 225 - Avg Reward: -16.54
Episode 226 - Avg Reward: -16.54
Episode 227 - Avg Reward: -16.54
Episode 228 - Avg Reward: -16.50
Episode 229 - Avg Reward: -16.50
Episode 230 - Avg Reward: -16.50
Episode 231 - Avg Reward: -16.38
Episode 232 - Avg Reward: -16.38
Episode 233 - Avg Reward: -16.38
Episode 234 - Avg Reward: -16.38
Episode 235 - Avg Reward: -16.38
Episode 236 - Avg Reward: -16.33
Episode 237 - Avg Reward: -16.33
Episode 238 - Avg Reward: -16.33
Episode 239 - Avg Reward: -16.30
Episode 240 - Avg Reward: -16.30
Episode 241 - Avg Reward: -16.26
Episode 242 - Avg Reward: -16.26
Episode 243 - Avg Reward: -16.26
Episode 244 - Avg Reward: -16.19
Episode 245 - Avg Reward: -16.19
Episode 246 - Avg Reward: -16.19
Episode 247 - Avg Reward: -16.12
Episode 248 - Avg Reward: -16.12
Episode 249 - Avg Reward: -16.12
Episode 250 - Avg Reward: -16.10
Episode 251 - Avg Reward: -16.10
Episode 252 - Avg Reward: -16.11
Episode 253 - Avg Reward: -16.11
Episode 254 - Avg Reward: -16.11
Episode 255 - Avg Reward: -16.01
Episode 256 - Avg Reward: -16.01
Episode 257 - Avg Reward: -16.01
Episode 258 - Avg Reward: -15.95
Episode 259 - Avg Reward: -15.95
Episode 260 - Avg Reward: -15.95
Episode 261 - Avg Reward: -15.90
Episode 262 - Avg Reward: -15.90
Episode 263 - Avg Reward: -15.90
Episode 264 - Avg Reward: -15.86
Episode 265 - Avg Reward: -15.86
Episode 266 - Avg Reward: -15.79
Episode 267 - Avg Reward: -15.79
Episode 268 - Avg Reward: -15.79
Episode 269 - Avg Reward: -15.77
Episode 270 - Avg Reward: -15.77
```

```
Episode 271 - Avg Reward: -15.75
Episode 272 - Avg Reward: -15.75
Episode 273 - Avg Reward: -15.71
Episode 274 - Avg Reward: -15.71
Episode 275 - Avg Reward: -15.71
Episode 276 - Avg Reward: -15.70
Episode 277 - Avg Reward: -15.70
Episode 278 - Avg Reward: -15.70
Episode 279 - Avg Reward: -15.61
Episode 280 - Avg Reward: -15.61
Episode 281 - Avg Reward: -15.61
Episode 282 - Avg Reward: -15.61
Episode 283 - Avg Reward: -15.54
Episode 284 - Avg Reward: -15.54
Episode 285 - Avg Reward: -15.49
Episode 286 - Avg Reward: -15.49
Episode 287 - Avg Reward: -15.49
Episode 288 - Avg Reward: -15.41
Episode 289 - Avg Reward: -15.41
Episode 290 - Avg Reward: -15.41
Episode 291 - Avg Reward: -15.41
Episode 292 - Avg Reward: -15.31
Episode 293 - Avg Reward: -15.31
Episode 294 - Avg Reward: -15.31
Episode 295 - Avg Reward: -15.25
Episode 296 - Avg Reward: -15.25
Episode 297 - Avg Reward: -15.25
Episode 298 - Avg Reward: -15.16
Episode 299 - Avg Reward: -15.16
Episode 300 - Avg Reward: -15.15
Episode 301 - Avg Reward: -15.15
Episode 302 - Avg Reward: -15.15
Episode 303 - Avg Reward: -15.10
Episode 304 - Avg Reward: -15.10
Episode 305 - Avg Reward: -15.10
Episode 306 - Avg Reward: -15.00
Episode 307 - Avg Reward: -15.00
Episode 308 - Avg Reward: -14.92
Episode 309 - Avg Reward: -14.92
Episode 310 - Avg Reward: -14.92
Episode 311 - Avg Reward: -14.92
Episode 312 - Avg Reward: -14.84
Episode 313 - Avg Reward: -14.84
Episode 314 - Avg Reward: -14.84
Episode 315 - Avg Reward: -14.79
Episode 316 - Avg Reward: -14.79
Episode 317 - Avg Reward: -14.79
Episode 318 - Avg Reward: -14.71
Episode 319 - Avg Reward: -14.71
Episode 320 - Avg Reward: -14.71
Episode 321 - Avg Reward: -14.71
Episode 322 - Avg Reward: -14.65
Episode 323 - Avg Reward: -14.65
Episode 324 - Avg Reward: -14.65
Episode 325 - Avg Reward: -14.56
Episode 326 - Avg Reward: -14.56
Episode 327 - Avg Reward: -14.53
Episode 328 - Avg Reward: -14.53
Episode 329 - Avg Reward: -14.53
Episode 330 - Avg Reward: -14.53
Episode 331 - Avg Reward: -14.40
Episode 332 - Avg Reward: -14.40
Episode 333 - Avg Reward: -14.38
Episode 334 - Avg Reward: -14.38
Episode 335 - Avg Reward: -14.38
Episode 336 - Avg Reward: -14.38
Episode 337 - Avg Reward: -14.29
Episode 338 - Avg Reward: -14.29
Episode 339 - Avg Reward: -14.29
Episode 340 - Avg Reward: -14.29
Episode 341 - Avg Reward: -14.23
Episode 342 - Avg Reward: -14.23
Episode 343 - Avg Reward: -14.23
Episode 344 - Avg Reward: -14.23
Episode 345 - Avg Reward: -14.23
Episode 346 - Avg Reward: -14.23
Episode 347 - Avg Reward: -14.23
Episode 348 - Avg Reward: -14.23
Episode 349 - Avg Reward: -14.20
Episode 350 - Avg Reward: -14.20
Episode 351 - Avg Reward: -14.20
Episode 352 - Avg Reward: -14.12
Episode 353 - Avg Reward: -14.12
```

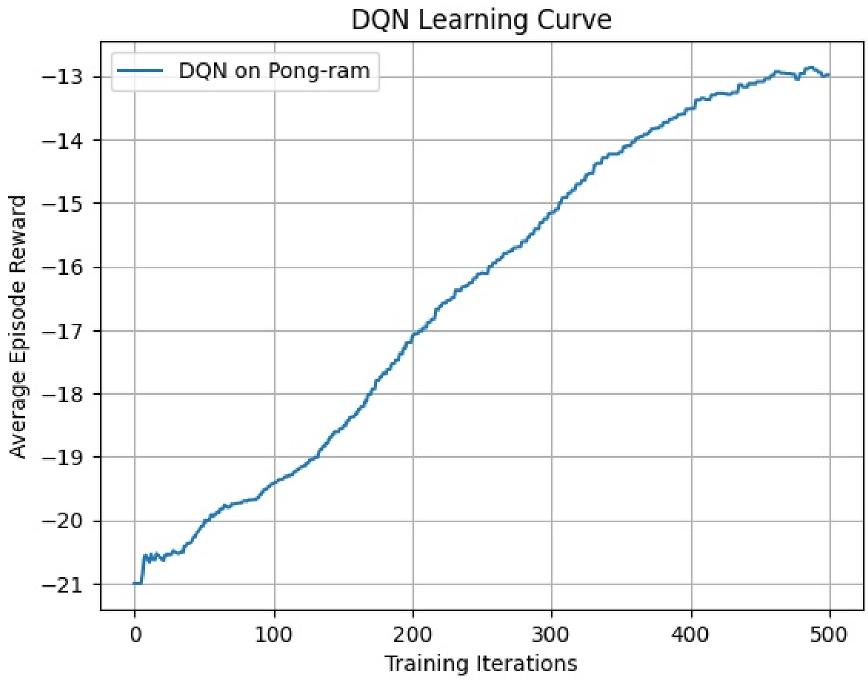
```
Episode 354 - Avg Reward: -14.10
Episode 355 - Avg Reward: -14.10
Episode 356 - Avg Reward: -14.10
Episode 357 - Avg Reward: -14.10
Episode 358 - Avg Reward: -14.04
Episode 359 - Avg Reward: -14.04
Episode 360 - Avg Reward: -14.04
Episode 361 - Avg Reward: -13.98
Episode 362 - Avg Reward: -13.98
Episode 363 - Avg Reward: -13.98
Episode 364 - Avg Reward: -13.95
Episode 365 - Avg Reward: -13.95
Episode 366 - Avg Reward: -13.95
Episode 367 - Avg Reward: -13.92
Episode 368 - Avg Reward: -13.92
Episode 369 - Avg Reward: -13.92
Episode 370 - Avg Reward: -13.88
Episode 371 - Avg Reward: -13.88
Episode 372 - Avg Reward: -13.83
Episode 373 - Avg Reward: -13.83
Episode 374 - Avg Reward: -13.83
Episode 375 - Avg Reward: -13.82
Episode 376 - Avg Reward: -13.82
Episode 377 - Avg Reward: -13.82
Episode 378 - Avg Reward: -13.79
Episode 379 - Avg Reward: -13.79
Episode 380 - Avg Reward: -13.79
Episode 381 - Avg Reward: -13.73
Episode 382 - Avg Reward: -13.73
Episode 383 - Avg Reward: -13.73
Episode 384 - Avg Reward: -13.73
Episode 385 - Avg Reward: -13.68
Episode 386 - Avg Reward: -13.68
Episode 387 - Avg Reward: -13.68
Episode 388 - Avg Reward: -13.66
Episode 389 - Avg Reward: -13.66
Episode 390 - Avg Reward: -13.66
Episode 391 - Avg Reward: -13.61
Episode 392 - Avg Reward: -13.61
Episode 393 - Avg Reward: -13.61
Episode 394 - Avg Reward: -13.60
Episode 395 - Avg Reward: -13.60
Episode 396 - Avg Reward: -13.60
Episode 397 - Avg Reward: -13.52
Episode 398 - Avg Reward: -13.52
Episode 399 - Avg Reward: -13.52
Episode 400 - Avg Reward: -13.51
Episode 401 - Avg Reward: -13.51
Episode 402 - Avg Reward: -13.51
Episode 403 - Avg Reward: -13.51
Episode 404 - Avg Reward: -13.38
Episode 405 - Avg Reward: -13.38
Episode 406 - Avg Reward: -13.38
Episode 407 - Avg Reward: -13.38
Episode 408 - Avg Reward: -13.35
Episode 409 - Avg Reward: -13.35
Episode 410 - Avg Reward: -13.35
Episode 411 - Avg Reward: -13.37
Episode 412 - Avg Reward: -13.37
Episode 413 - Avg Reward: -13.37
Episode 414 - Avg Reward: -13.37
Episode 415 - Avg Reward: -13.30
Episode 416 - Avg Reward: -13.30
Episode 417 - Avg Reward: -13.30
Episode 418 - Avg Reward: -13.30
Episode 419 - Avg Reward: -13.27
Episode 420 - Avg Reward: -13.27
Episode 421 - Avg Reward: -13.27
Episode 422 - Avg Reward: -13.27
Episode 423 - Avg Reward: -13.27
Episode 424 - Avg Reward: -13.27
Episode 425 - Avg Reward: -13.28
Episode 426 - Avg Reward: -13.28
Episode 427 - Avg Reward: -13.29
Episode 428 - Avg Reward: -13.29
Episode 429 - Avg Reward: -13.29
Episode 430 - Avg Reward: -13.26
Episode 431 - Avg Reward: -13.26
Episode 432 - Avg Reward: -13.26
Episode 433 - Avg Reward: -13.26
Episode 434 - Avg Reward: -13.26
Episode 435 - Avg Reward: -13.14
Episode 436 - Avg Reward: -13.14
```

```
Episode 437 - Avg Reward: -13.14
Episode 438 - Avg Reward: -13.17
Episode 439 - Avg Reward: -13.17
Episode 440 - Avg Reward: -13.17
Episode 441 - Avg Reward: -13.17
Episode 442 - Avg Reward: -13.12
Episode 443 - Avg Reward: -13.12
Episode 444 - Avg Reward: -13.12
Episode 445 - Avg Reward: -13.12
Episode 446 - Avg Reward: -13.12
Episode 447 - Avg Reward: -13.12
Episode 448 - Avg Reward: -13.09
Episode 449 - Avg Reward: -13.09
Episode 450 - Avg Reward: -13.09
Episode 451 - Avg Reward: -13.09
Episode 452 - Avg Reward: -13.09
Episode 453 - Avg Reward: -13.09
Episode 454 - Avg Reward: -13.04
Episode 455 - Avg Reward: -13.04
Episode 456 - Avg Reward: -13.04
Episode 457 - Avg Reward: -13.04
Episode 458 - Avg Reward: -12.99
Episode 459 - Avg Reward: -12.99
Episode 460 - Avg Reward: -12.99
Episode 461 - Avg Reward: -12.93
Episode 462 - Avg Reward: -12.93
Episode 463 - Avg Reward: -12.93
Episode 464 - Avg Reward: -12.93
Episode 465 - Avg Reward: -12.95
Episode 466 - Avg Reward: -12.95
Episode 467 - Avg Reward: -12.95
Episode 468 - Avg Reward: -12.95
Episode 469 - Avg Reward: -12.96
Episode 470 - Avg Reward: -12.96
Episode 471 - Avg Reward: -12.96
Episode 472 - Avg Reward: -12.96
Episode 473 - Avg Reward: -12.97
Episode 474 - Avg Reward: -12.97
Episode 475 - Avg Reward: -12.97
Episode 476 - Avg Reward: -13.05
Episode 477 - Avg Reward: -13.05
Episode 478 - Avg Reward: -13.05
Episode 479 - Avg Reward: -12.96
Episode 480 - Avg Reward: -12.96
Episode 481 - Avg Reward: -12.96
Episode 482 - Avg Reward: -12.96
Episode 483 - Avg Reward: -12.88
Episode 484 - Avg Reward: -12.88
Episode 485 - Avg Reward: -12.88
Episode 486 - Avg Reward: -12.86
Episode 487 - Avg Reward: -12.86
Episode 488 - Avg Reward: -12.86
Episode 489 - Avg Reward: -12.90
Episode 490 - Avg Reward: -12.90
Episode 491 - Avg Reward: -12.90
Episode 492 - Avg Reward: -12.94
Episode 493 - Avg Reward: -12.94
Episode 494 - Avg Reward: -12.94
Episode 495 - Avg Reward: -13.00
Episode 496 - Avg Reward: -13.00
Episode 497 - Avg Reward: -13.00
Episode 498 - Avg Reward: -12.98
Episode 499 - Avg Reward: -12.98
```

Plot the Learning Curve

In [4]:

```python
plt.plot(avg_rewards, label="DQN on Pong-ram")
plt.xlabel("Training Iterations")
plt.ylabel("Average Episode Reward")
plt.title("DQN Learning Curve")
plt.legend()
plt.grid(True)
plt.show()
```


DQN Learning Curve
DQN on Pong-ram
Average Episode Reward
Training Iterations
−13
−14
−15
−16
−17
−18
−19
−20
−21
0
100
200
300
400
500

In [ ]:

In [1]:

```python
import gymnasium as gym
import ray
import matplotlib.pyplot as plt
from ray.rllib.algorithms.ppo import PPOConfig
```

```
2025-05-11 13:06:18,065 WARNING deprecation.py:50 -- DeprecationWarning: `DirectStepOptimizer` has been deprecat
ed. This will raise an error in the future!
```

In [2]:

```python
# Initialize Ray
ray.init(ignore_reinit_error=True)
```

```
2025-05-11 13:06:26,263 INFO worker.py:1621 -- Started a local Ray instance.
```

Out[2]:

RAY

Python version:	3.10.16
Ray version:	2.6.3

In [3]:

```python
# PPO Configuration
config = (
    PPOConfig()
    .environment(env="ALE/Pong-ram-v5")  # Atari RAM environment
    .framework("torch")
    .resources(num_gpus=0)  # Set to 1 if using CUDA
    .rollouts(num_rollout_workers=0)  # Use >0 if not on notebook
    .training(
        gamma=0.99,
        lr=5e-5,  # Often smaller for PPO
        train_batch_size=4000,
        sgd_minibatch_size=128,
        num_sgd_iter=10,
        model={"fcnet_hiddens": [256], "fcnet_activation": "relu"},
    )
    .debugging(log_level="WARN")
)
```

```
2025-05-11 13:08:20,898 WARNING algorithm_config.py:2558 -- Setting `exploration_config={}` because you set `_en
able_rl_module_api=True`. When RLModule API are enabled, exploration_config can not be set. If you want to imple
ment custom exploration behaviour, please modify the `forward_exploration` method of the RLModule at hand. On co
nfigs that have a default exploration config, this must be done with `config.exploration_config={}`.
```

In [4]:

```python
from ray.rllib.algorithms.ppo import PPO

# Build PPO trainer
algo = config.build()
avg_rewards = []

# Training Loop
for episode in range(100):
    result = algo.train()
    reward = result["episode_reward_mean"]
    print(f"PPO Episode {episode} - Avg Reward: {reward:.2f}")
    avg_rewards.append(reward)
```

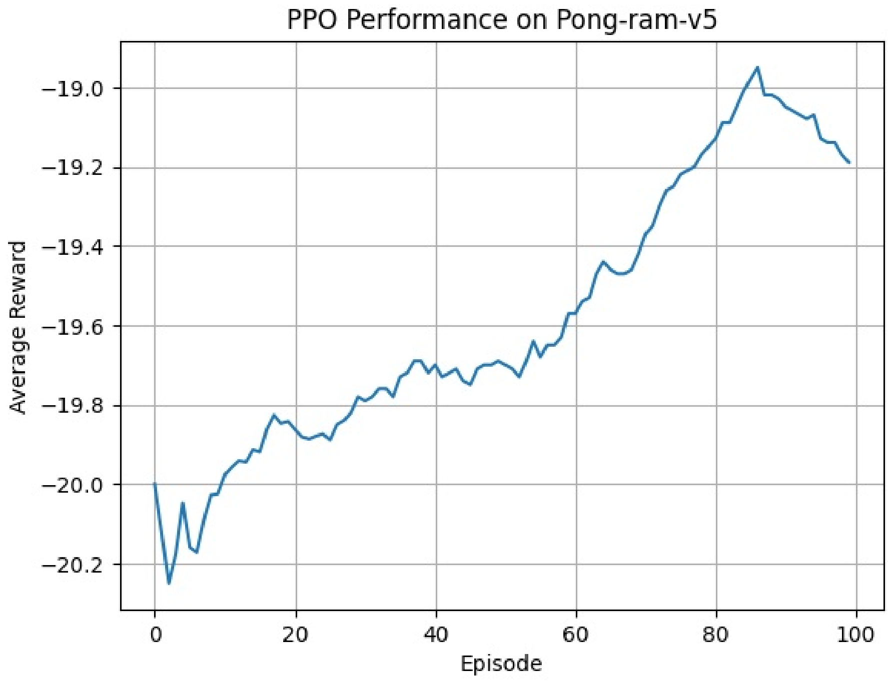
```
2025-05-11 13:08:22,631 WARNING algorithm_config.py:2558 -- Setting `exploration_config={}` because you set `_en
able_rl_module_api=True`. When RLModule API are enabled, exploration_config can not be set. If you want to imple
ment custom exploration behaviour, please modify the `forward_exploration` method of the RLModule at hand. On co
nfigs that have a default exploration config, this must be done with `config.exploration_config={}`.
C:\Users\elias\anaconda3\envs\rllib-atari\lib\site-packages\ray\rllib\algorithms\algorithm.py:484: RayDeprecatio
nWarning: This API is deprecated and may be removed in future Ray releases. You could suppress this warning by s
etting env variable PYTHONWARNINGS="ignore::DeprecationWarning"
`UnifiedLogger` will be removed in Ray 2.7.
  return UnifiedLogger(config, logdir, loggers=None)
C:\Users\elias\anaconda3\envs\rllib-atari\lib\site-packages\ray\tune\logger\unified.py:53: RayDeprecationWarning
: This API is deprecated and may be removed in future Ray releases. You could suppress this warning by setting e
nv variable PYTHONWARNINGS="ignore::DeprecationWarning"
The `JsonLogger interface is deprecated in favor of the `ray.tune.json.JsonLoggerCallback` interface and will be
removed in Ray 2.7.
  self._loggers.append(cls(self.config, self.logdir, self.trial))
C:\Users\elias\anaconda3\envs\rllib-atari\lib\site-packages\ray\tune\logger\unified.py:53: RayDeprecationWarning
: This API is deprecated and may be removed in future Ray releases. You could suppress this warning by setting e
nv variable PYTHONWARNINGS="ignore::DeprecationWarning"
The `CSVLogger interface is deprecated in favor of the `ray.tune.csv.CSVLoggerCallback` interface and will be re
moved in Ray 2.7.
  self._loggers.append(cls(self.config, self.logdir, self.trial))
C:\Users\elias\anaconda3\envs\rllib-atari\lib\site-packages\ray\tune\logger\unified.py:53: RayDeprecationWarning
: This API is deprecated and may be removed in future Ray releases. You could suppress this warning by setting e
nv variable PYTHONWARNINGS="ignore::DeprecationWarning"
The `TBXLogger interface is deprecated in favor of the `ray.tune.tensorboardx.TBXLoggerCallback` interface and w
ill be removed in Ray 2.7.
  self._loggers.append(cls(self.config, self.logdir, self.trial))
C:\Users\elias\anaconda3\envs\rllib-atari\lib\site-packages\gym\utils\passive_env_checker.py:233: DeprecationWar
ning: `np.bool8` is a deprecated alias for `np.bool_`.  (Deprecated NumPy 1.24)
  if not isinstance(terminated, (bool, np.bool8)):
2025-05-11 13:08:23,030 WARNING algorithm_config.py:2558 -- Setting `exploration_config={}` because you set `_en
able_rl_module_api=True`. When RLModule API are enabled, exploration_config can not be set. If you want to imple
ment custom exploration behaviour, please modify the `forward_exploration` method of the RLModule at hand. On co
nfigs that have a default exploration config, this must be done with `config.exploration_config={}`.
2025-05-11 13:08:23,057 WARNING deprecation.py:50 -- DeprecationWarning: `ValueNetworkMixin` has been deprecated
. This will raise an error in the future!
2025-05-11 13:08:23,059 WARNING deprecation.py:50 -- DeprecationWarning: `LearningRateSchedule` has been depreca
ted. This will raise an error in the future!
2025-05-11 13:08:23,060 WARNING deprecation.py:50 -- DeprecationWarning: `EntropyCoeffSchedule` has been depreca
ted. This will raise an error in the future!
2025-05-11 13:08:23,061 WARNING deprecation.py:50 -- DeprecationWarning: `KLCoeffMixin` has been deprecated. Thi
s will raise an error in the future!
2025-05-11 13:08:27,551 WARNING util.py:68 -- Install gputil for GPU system monitoring.
PPO Episode 0 - Avg Reward: -20.00
PPO Episode 1 - Avg Reward: -20.12
PPO Episode 2 - Avg Reward: -20.25
PPO Episode 3 - Avg Reward: -20.18
PPO Episode 4 - Avg Reward: -20.05
PPO Episode 5 - Avg Reward: -20.16
PPO Episode 6 - Avg Reward: -20.17
PPO Episode 7 - Avg Reward: -20.09
PPO Episode 8 - Avg Reward: -20.03
PPO Episode 9 - Avg Reward: -20.02
PPO Episode 10 - Avg Reward: -19.98
PPO Episode 11 - Avg Reward: -19.96
PPO Episode 12 - Avg Reward: -19.94
PPO Episode 13 - Avg Reward: -19.94
PPO Episode 14 - Avg Reward: -19.91
PPO Episode 15 - Avg Reward: -19.92
PPO Episode 16 - Avg Reward: -19.86
PPO Episode 17 - Avg Reward: -19.83
PPO Episode 18 - Avg Reward: -19.85
PPO Episode 19 - Avg Reward: -19.84
PPO Episode 20 - Avg Reward: -19.86
PPO Episode 21 - Avg Reward: -19.88
PPO Episode 22 - Avg Reward: -19.89
PPO Episode 23 - Avg Reward: -19.88
PPO Episode 24 - Avg Reward: -19.87
PPO Episode 25 - Avg Reward: -19.89
PPO Episode 26 - Avg Reward: -19.85
PPO Episode 27 - Avg Reward: -19.84
PPO Episode 28 - Avg Reward: -19.82
PPO Episode 29 - Avg Reward: -19.78
PPO Episode 30 - Avg Reward: -19.79
PPO Episode 31 - Avg Reward: -19.78
PPO Episode 32 - Avg Reward: -19.76
PPO Episode 33 - Avg Reward: -19.76
PPO Episode 34 - Avg Reward: -19.78
PPO Episode 35 - Avg Reward: -19.73
PPO Episode 36 - Avg Reward: -19.72
PPO Episode 37 - Avg Reward: -19.69
PPO Episode 38 - Avg Reward: -19.69
PPO Episode 39 - Avg Reward: -19.72
```

```
PPO Episode 40 - Avg Reward: -19.70
PPO Episode 41 - Avg Reward: -19.73
PPO Episode 42 - Avg Reward: -19.72
PPO Episode 43 - Avg Reward: -19.71
PPO Episode 44 - Avg Reward: -19.74
PPO Episode 45 - Avg Reward: -19.75
PPO Episode 46 - Avg Reward: -19.71
PPO Episode 47 - Avg Reward: -19.70
PPO Episode 48 - Avg Reward: -19.70
PPO Episode 49 - Avg Reward: -19.69
PPO Episode 50 - Avg Reward: -19.70
PPO Episode 51 - Avg Reward: -19.71
PPO Episode 52 - Avg Reward: -19.73
PPO Episode 53 - Avg Reward: -19.69
PPO Episode 54 - Avg Reward: -19.64
PPO Episode 55 - Avg Reward: -19.68
PPO Episode 56 - Avg Reward: -19.65
PPO Episode 57 - Avg Reward: -19.65
PPO Episode 58 - Avg Reward: -19.63
PPO Episode 59 - Avg Reward: -19.57
PPO Episode 60 - Avg Reward: -19.57
PPO Episode 61 - Avg Reward: -19.54
PPO Episode 62 - Avg Reward: -19.53
PPO Episode 63 - Avg Reward: -19.47
PPO Episode 64 - Avg Reward: -19.44
PPO Episode 65 - Avg Reward: -19.46
PPO Episode 66 - Avg Reward: -19.47
PPO Episode 67 - Avg Reward: -19.47
PPO Episode 68 - Avg Reward: -19.46
PPO Episode 69 - Avg Reward: -19.42
PPO Episode 70 - Avg Reward: -19.37
PPO Episode 71 - Avg Reward: -19.35
PPO Episode 72 - Avg Reward: -19.30
PPO Episode 73 - Avg Reward: -19.26
PPO Episode 74 - Avg Reward: -19.25
PPO Episode 75 - Avg Reward: -19.22
PPO Episode 76 - Avg Reward: -19.21
PPO Episode 77 - Avg Reward: -19.20
PPO Episode 78 - Avg Reward: -19.17
PPO Episode 79 - Avg Reward: -19.15
PPO Episode 80 - Avg Reward: -19.13
PPO Episode 81 - Avg Reward: -19.09
PPO Episode 82 - Avg Reward: -19.09
PPO Episode 83 - Avg Reward: -19.05
PPO Episode 84 - Avg Reward: -19.01
PPO Episode 85 - Avg Reward: -18.98
PPO Episode 86 - Avg Reward: -18.95
PPO Episode 87 - Avg Reward: -19.02
PPO Episode 88 - Avg Reward: -19.02
PPO Episode 89 - Avg Reward: -19.03
PPO Episode 90 - Avg Reward: -19.05
PPO Episode 91 - Avg Reward: -19.06
PPO Episode 92 - Avg Reward: -19.07
PPO Episode 93 - Avg Reward: -19.08
PPO Episode 94 - Avg Reward: -19.07
PPO Episode 95 - Avg Reward: -19.13
PPO Episode 96 - Avg Reward: -19.14
PPO Episode 97 - Avg Reward: -19.14
PPO Episode 98 - Avg Reward: -19.17
PPO Episode 99 - Avg Reward: -19.19
```

Plot the Learning Curve

In [5]:

```
# Plotting
plt.plot(avg_rewards)
plt.xlabel("Episode")
plt.ylabel("Average Reward")
plt.title("PPO Performance on Pong-ram-v5")
plt.grid(True)
plt.show()
```


PPO Performance on Pong-ram-v5
Average Reward
Episode
−19.0
−19.2
−19.4
−19.6
−19.8
−20.0
−20.2
0
20
40
60
80
100

In [ ]: