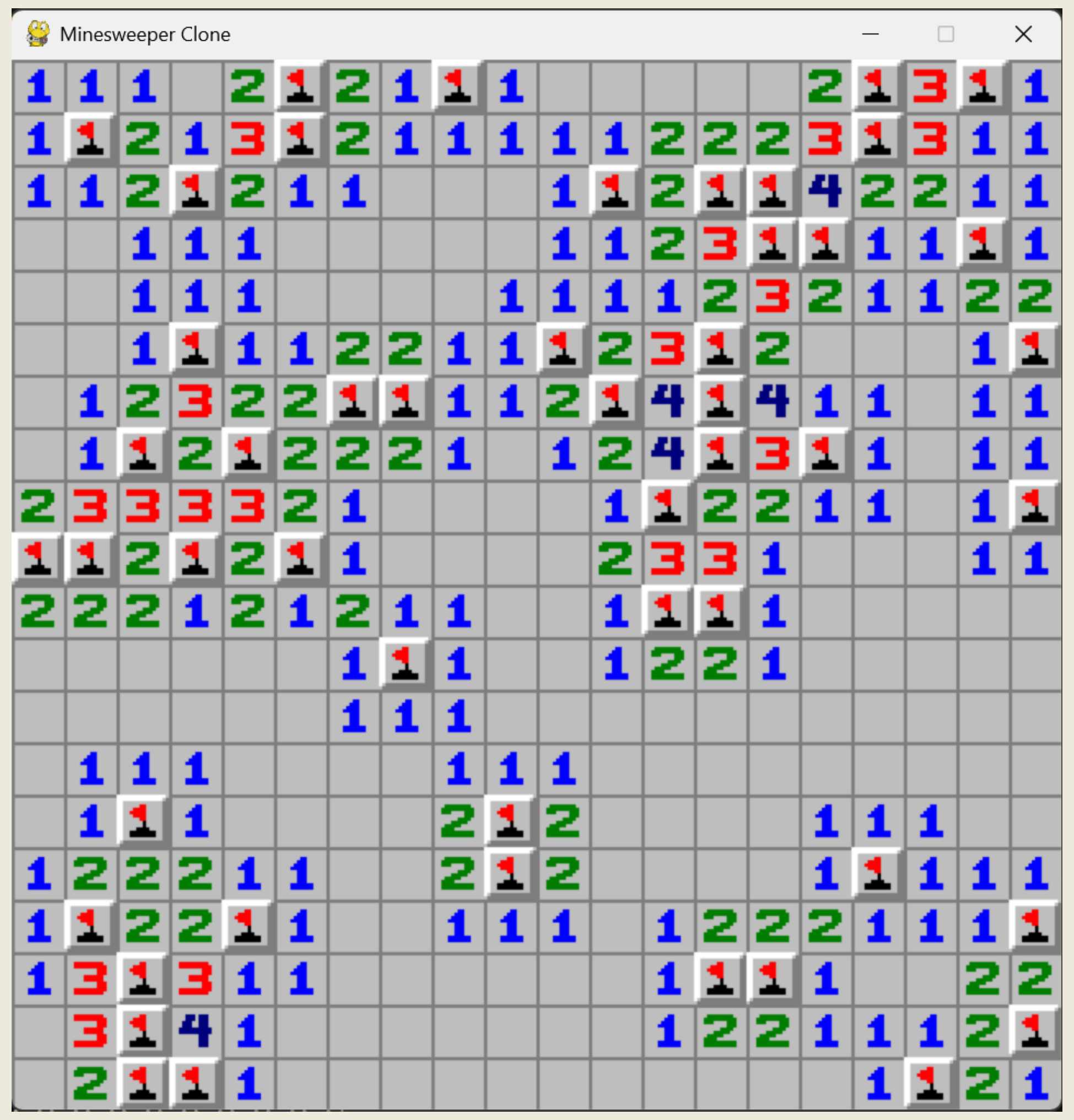
Reinforcement learning on Minesweeper

AUTHORS

Junhao Chen, Franklin Pu, Kabir Maharjan



01. Introduction

Minesweeper is a classic logic-based puzzle game in which players uncover tiles to deduce the locations of hidden mines. Although humans rely on reasoning, deduction, and probabilistic decision-making, these skills are not straightforward for reinforcement learning (RL) agents to acquire. Minesweeper presents a challenging partially observable environment with sparse rewards, irreversible mistakes, and a large action space.

This project investigates whether a Deep Q-Network (DQN) agent can learn patterns that outperform a random baseline and explores the limits of learning in such a combinatorial, information-sparse domain.

02. Objective

Our project aims to design, implement, and evaluate an RL agent capable of playing Minesweeper.

Specifically, we seek to:

- Build a functional Minesweeper environment with a consistent state encoding.
- Implement a DQN agent and compare its performance to a random agent.
- Analyze whether meaningful learning occurs and identify bottlenecks in reward design, network capacity, and environment complexity.

03. Methodology

We implemented a custom Minesweeper environment using Python and Pygame, then wrapped it for RL interaction. The agent receives a 15×15 (or 8×8) matrix encoding the visible board, where each tile is marked as unknown, flagged, empty, or a clue number .

Our DQN formulation includes:

- State: numerical board observation.
- Action: selecting any tile to dig (0–224 for 15×15).
- Rewards: small positive reward for safe reveals, penalty for redundant clicks, large negative reward for hitting a mine, and positive reward for winning.
- Training: ε-greedy exploration and replay buffer to stabilize learning.
- Multiple experiments were performed on different board sizes (15×15 → too hard; 8×8 → more manageable).

04. Results/Findings

On 15×15, the agent failed to improve beyond random due to extreme sparsity, large state space, and frequent early death.

On 8×8, early training showed minor improvement over random, but long-term training revealed instability: rewards decreased over time, suggesting the agent "learned badly" by overfitting to local correlations instead of global structure.

The agent often preferred conservative or repetitive actions, reflecting the difficulty of learning high-level deductive reasoning from scalar rewards alone.

05. Analysis

To evaluate whether the DQN agent learned meaningful strategies, we compared its reward curves against a random baseline across multiple Minesweeper settings.

Short-term learning (8×8 board, 500 episodes).

In early training, the DQN agent shows mild improvement over the random agent. The rewards stabilize slightly higher, suggesting that the model learns basic heuristics such as avoiding repeated clicks and uncovering local empty regions. However, this improvement is small and does not reflect deeper logical inference.

Long-term training (8×8 board, 3000 episodes — no flagging).

Using the 3000-episode curve, we observe a clear collapse in DQN performance: after initial stabilization, the reward steadily drifts downward and ultimately performs worse than the random policy. This degradation indicates catastrophic overfitting—the agent over-specializes to spurious correlations in its replay buffer rather than learning board-general strategies. Minesweeper's sparse rewards and hidden information amplify this instability, making value-based learning unreliable over long horizons.

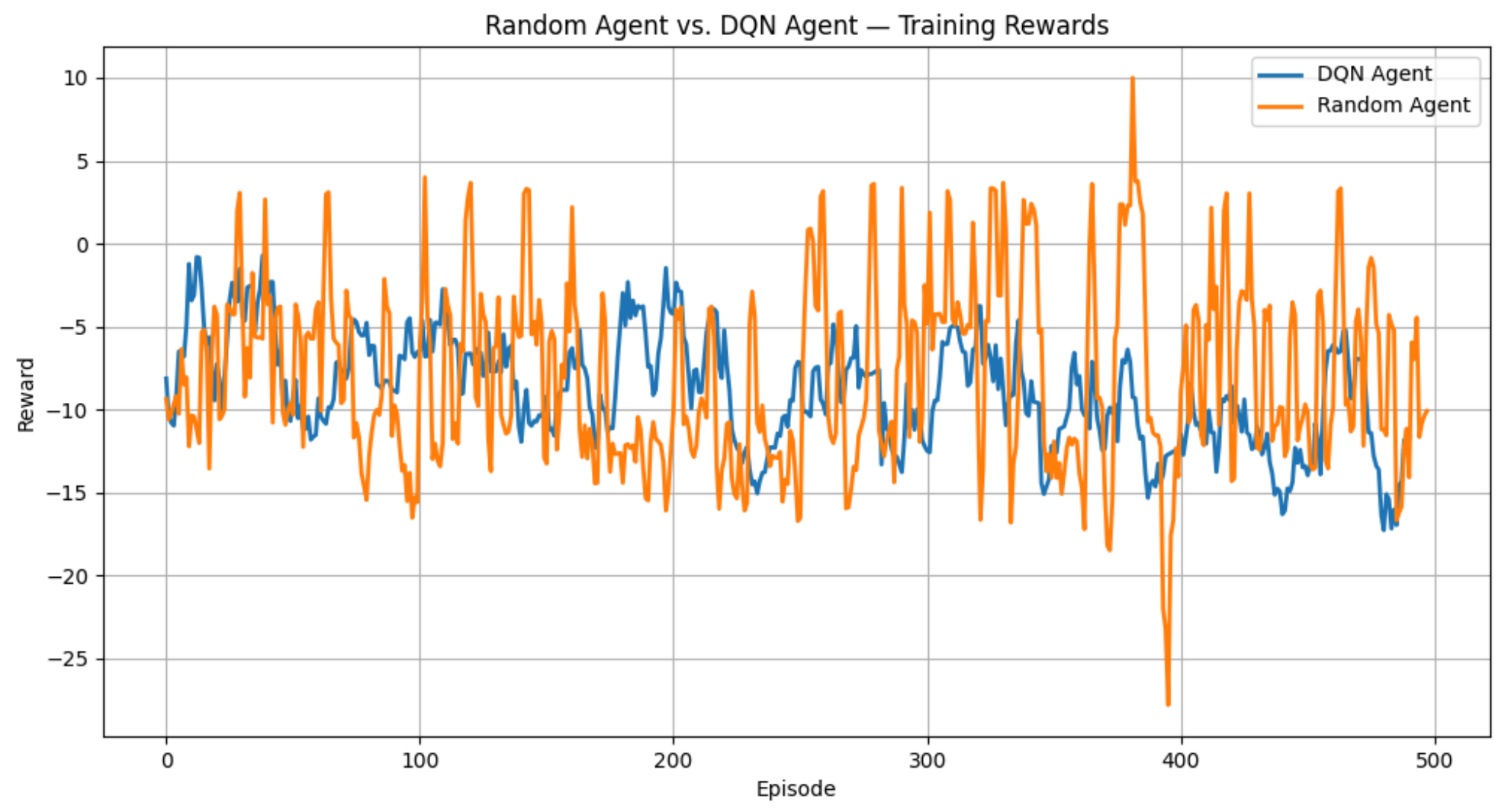
Impact of expanding the action space (flagging enabled).

When flagging is added, the DQN's performance deteriorates even more dramatically. Flagging requires deductive reasoning about hidden tiles, but the agent treats it as guesswork, accumulating large penalties. The expanded action set increases credit-assignment difficulty and reveals the fundamental mismatch between Minesweeper's logical structure and DQN's pattern-based learning.

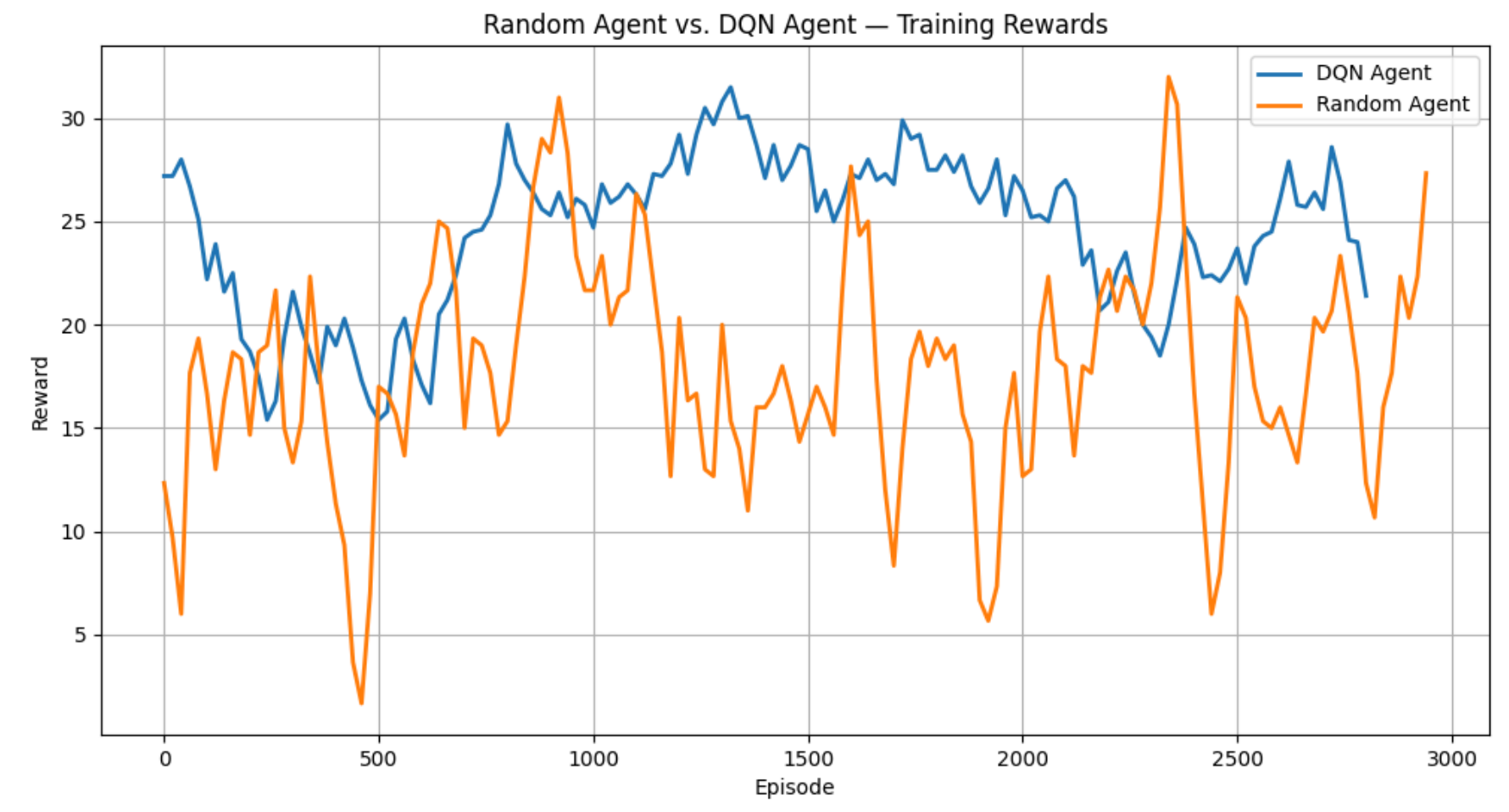
Overall insight.

Across experiments, DQN learns limited short-term heuristics but fails to generalize and becomes increasingly unstable with longer training or more complex action choices. These results suggest that pure DQN is insufficient for solving Minesweeper, and that successful agents may require structured reasoning components, model-based approaches, or hybrid symbolic-neural methods.

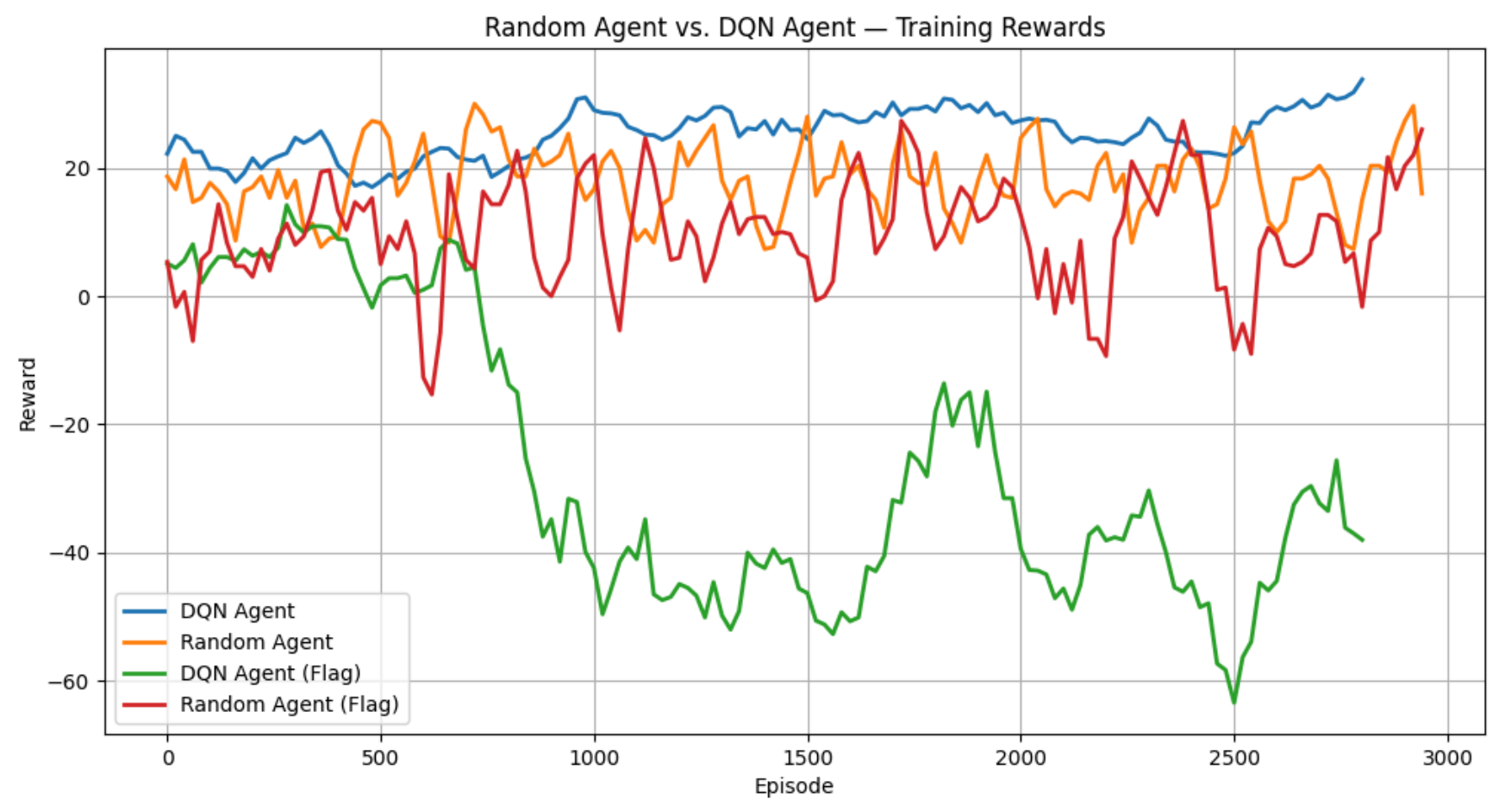
8×8 board — 500 episodes



8×8 board — 3000 episodes (no flagging)



8×8 board — 3000 episodes with flagging



06. Conclusion

Our experiments show that while a DQN agent can learn small short-term heuristics in Minesweeper, it fails to develop stable or generalizable strategies. Even in simplified 8×8 environments, extended training leads to performance collapse, and adding realistic actions such as flagging makes learning significantly worse. These patterns highlight a fundamental limitation: Minesweeper requires deductive, probabilistic reasoning about hidden information, while DQN relies on pattern association and dense feedback.

Overall, the results suggest that value-based reinforcement learning alone is not well suited for Minesweeper. Future approaches may require richer state representations, curriculum learning, or hybrid methods that combine RL with logical inference or supervised pattern recognition.