Can Modern Deep RL Overcome Sparse Rewards? A Case Study in Minesweeper

Josh Manchester
Department of Computer Science
University of Colorado Colorado Springs
Colorado Springs, CO 80918
jmanches@uccs.edu

Abstract

Sparse reward environments remain a fundamental challenge in reinforcement learning, where agents receive feedback only upon task completion rather than incrementally. This proposal investigates whether modern deep RL algorithms can achieve competent performance in Minesweeper—a domain characterized by sparse rewards, large state spaces, and partial observability. Prior work combining Deep Q-Networks with constraint propagation logic achieved an 88% win rate, while pure DQN achieved only 1%. This dramatic gap raises a central question: can advanced RL techniques such as Proximal Policy Optimization (PPO), Soft Actor-Critic (SAC), reward shaping, and curriculum learning close this performance gap, or is domain knowledge fundamentally required? This project will systematically evaluate these approaches to provide empirical guidance on the limitations of pure RL in sparse reward domains.

Introduction

Reinforcement learning has achieved remarkable success in domains ranging from Atari games to robotic control (Mnih et al. 2015; Haarnoja et al. 2018). However, sparse reward environments—where agents receive meaningful feedback only at episode termination—remain a persistent challenge (Sutton and Barto 2018). Recent work continues to address this fundamental limitation through techniques such as reward decomposition (Chen et al. 2024) and behavior-driven exploration (Zhang et al. 2025).

Minesweeper presents an ideal testbed for investigating sparse reward limitations. The game requires an agent to uncover all safe cells on a grid while avoiding hidden mines. Feedback is binary and delayed—the agent wins or loses only when the game ends. Additionally, the state space is enormous (over 5.3 million possible configurations even for a 6×6 grid with four mines), and the environment is partially observable since mine locations are hidden (Phan and Nguyen 2025).

Recent work by Phan and Nguyen (2025) achieved 93% win rates on small 6×6 grids using DQN and supervised learning, while Jiang et al. (2025) explored meta-RL approaches using Minesweeper as a test environment for language model agents. However, performance on larger boards remains challenging.

In prior work, I implemented a Deep Q-Network agent for Minesweeper that achieved only a 1% win rate on standard 16×16 difficulty. However, when combining the neural network with AC-3 constraint propagation logic—using the network only for "guessing" when logic could not determine a safe move—the hybrid system achieved an 88% win rate. This 87-point performance gap motivates the central research question:

Can modern deep RL algorithms overcome the sparse reward problem in Minesweeper, or is domain-specific knowledge fundamentally required for competent play?

Background

Minesweeper as a Markov Decision Process

We formalize Minesweeper as a Markov Decision Process (MDP) defined by the tuple $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$:

State Space $\mathcal{S}$: Each state $s \in \mathcal{S}$ is represented as an $H \times W \times C$ tensor, where H and W are the board dimensions and C is the number of feature channels. Following Phan and Nguyen (2025), we encode each cell using 12 channels: one channel for each possible neighbor count (0–8), one for unrevealed cells, one for flagged cells, and one for revealed status. For a 16×16 board, the state space contains over 12^{256} possible configurations, though the reachable subset is smaller due to game constraints.

Action Space $\mathcal{A}$: The action space consists of $H \times W$ discrete actions, where each action corresponds to revealing a specific cell. Invalid actions (revealing already-revealed cells) are masked during action selection. For a 16×16 board, $|\mathcal{A}| = 256$.

Transition Function $P(s'|s, a)$: Transitions are stochastic from the agent's perspective due to partial observability—the agent cannot observe mine locations until revealed. Revealing a safe cell with zero adjacent mines triggers recursive reveals of neighboring cells.

Reward Function $R(s, a, s')$: In the sparse reward setting, $R = +1$ for winning (all safe cells revealed), $R = -1$ for losing (mine revealed), and $R = 0$ otherwise.

Discount Factor γ: Set to 0.99 to encourage efficient solutions while maintaining long-term planning.

Sparse Rewards in Reinforcement Learning

The sparse reward problem occurs when an agent receives non-zero rewards only rarely, making credit assignment difficult (Sutton and Barto 2018). Murphy (2024) provides a comprehensive overview of approaches to this challenge, including reward shaping, intrinsic motivation, and hierarchical methods. Recent advances include spatial-temporal return decomposition (Chen et al. 2024) and addressing "lazy agent" phenomena in multi-agent settings (Liu et al. 2023).

Deep Q-Network Improvements

Since the foundational DQN work (Mnih et al. 2015), numerous improvements have been proposed. Recent advances include β-DQN, which augments DQN with a behavior function for improved exploration (Zhang et al. 2025), and Elastic Step DQN, which dynamically varies step sizes to alleviate overestimation bias (Han et al. 2024). These improvements may help address the challenges of sparse reward environments.

Policy Gradient Methods

Policy gradient methods offer an alternative to value-based approaches. Proximal Policy Optimization (PPO) uses a clipped objective function for stable policy updates (Schulman et al. 2017), while Soft Actor-Critic (SAC) combines off-policy learning with maximum entropy objectives for robust exploration (Haarnoja et al. 2018). These methods may offer advantages in sparse reward settings due to their different exploration characteristics.

Curriculum Learning

Curriculum learning—training agents on progressively harder tasks—has shown promise for difficult RL problems (Parker-Holder et al. 2024). By starting with simpler versions of a task and gradually increasing difficulty, agents may develop foundational skills that transfer to harder settings.

Proposed Methods

I propose to systematically evaluate the following approaches on Minesweeper:

Baseline: Deep Q-Network

The existing DQN implementation with Double DQN and Dueling architecture will serve as the baseline, currently achieving approximately 1% win rate on 16×16 boards with 40 mines.

DQN Improvements

Building on recent advances, I will implement:

- β-DQN with behavior-driven exploration (Zhang et al. 2025)
- Elastic Step DQN for reduced overestimation (Han et al. 2024)

Policy Gradient Methods

PPO: Proximal Policy Optimization may provide more stable learning in sparse reward settings due to its conservative policy updates (Schulman et al. 2017).

SAC: Soft Actor-Critic's entropy maximization encourages exploration, which may help the agent discover winning strategies more efficiently (Haarnoja et al. 2018).

Reward Shaping

Instead of receiving reward only at game end, I will experiment with intermediate rewards:

- $+1$ for each safe cell revealed
- -0.5 for incorrect flag placement
- $+10$ for game completion (scaled by remaining cells)

Curriculum Learning

Following Parker-Holder et al. (2024), I will implement a curriculum that gradually increases difficulty:

1. 5×5 grid with 3 mines (easy)
2. 8×8 grid with 10 mines (medium)
3. 16×16 grid with 40 mines (expert)

The agent will advance to harder levels only after achieving a threshold win rate.

Evaluation

Performance will be measured by:

- **Win rate:** Percentage of games won across 10,000 test episodes
- **Sample efficiency:** Training steps required to reach threshold performance
- **Learning curves:** Win rate versus training episodes

All methods will be compared against baselines from our prior work and the literature:

Method	Board Size	Win Rate
Random baseline	16×16	<1%
Pure DQN (ours)	16×16	1%
DQN + CNN (Phan and Nguyen 2025)	6×6	93%
DQN + CNN (Phan and Nguyen 2025)	8×8	72%
Hybrid DQN + AC-3 (ours)	16×16	88%
Human expert	16×16	∼85%
Target: Modern RL	*16×16*	*>50%?*

Table 1: Baseline performance comparison. The key question is whether modern RL methods can close the gap between pure DQN (1%) and hybrid approaches (88%) on expert-level boards.

Expected Contributions

This project will provide:

1. Empirical comparison of modern RL algorithms (including recent advances like β-DQN) on a challenging sparse reward task
2. Analysis of when pure RL succeeds versus when domain knowledge is required
3. Practical guidance for practitioners facing sparse reward problems

Timeline

Date	Milestone
Feb 17–19	Proposal presentation
Feb 19	Proposal paper due
Mar 1	PPO, SAC, and β-DQN implementations
Mar 15	Reward shaping experiments complete
Mar 31	Curriculum learning experiments complete
Apr 1	Midterm presentation and paper
Apr 15	Full experimental results
May 7	Final presentation
May 12	Final paper due

Expected Challenges

Several challenges may affect experimental outcomes:

Partial Observability: Minesweeper violates the Markov property since optimal actions depend on hidden mine locations. Standard RL algorithms assume full observability; this mismatch may fundamentally limit achievable performance regardless of algorithm sophistication.

Sample Efficiency: The large state space ($>12^{256}$ configurations) requires extensive exploration. Policy gradient methods like PPO and SAC typically require millions of environment interactions, which may be computationally prohibitive for thorough hyperparameter search.

Credit Assignment: Even with reward shaping, determining which early-game actions contributed to eventual success or failure remains difficult. A single unlucky first move can doom an otherwise perfect strategy.

Curriculum Transfer: Skills learned on small boards (5×5) may not transfer effectively to larger boards (16×16) due to qualitatively different strategic requirements. The curriculum may need careful tuning to ensure positive transfer.

Evaluation Variance: Due to the stochastic nature of mine placement, win rates exhibit high variance. Statistical significance testing will require large sample sizes (10,000+ episodes per configuration).

Conclusion

This proposal outlines a systematic investigation into whether modern deep RL algorithms can overcome the sparse reward challenge in Minesweeper. By comparing PPO, SAC, β-DQN, reward shaping, and curriculum learning against a DQN baseline, this work will provide empirical evidence on the fundamental limitations of pure RL and the necessity of domain knowledge in challenging environments.

References

Chen, S.; Zhang, Z.; Yang, Y.; and Du, Y. 2024. STAS: Spatial-temporal return decomposition for solving sparse rewards problems in multi-agent reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 17337–17345.

Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, 1861–1870. PMLR.

Han, S.; Zhang, Z.; Chen, W.; and Li, Y. 2024. Elastic step DQN: A novel multi-step algorithm to alleviate overestimation in Deep Q-Networks. *Neurocomputing*, 567: 127036.

Jiang, Y.; Zhang, H.; Chen, Q.; and Xiao, Z. 2025. Meta-RL Induces Exploration in Language Agents. *arXiv preprint arXiv:2512.16848*.

Liu, B.; Pu, Z.; Pan, Y.; Yi, J.; Liang, Y.; and Zhang, D. 2023. Lazy Agents: A New Perspective on Solving Sparse Reward Problem in Multi-agent Reinforcement Learning. In *International Conference on Machine Learning*, 21937–21950. PMLR.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529–533.

Murphy, K. P. 2024. Reinforcement Learning: An Overview. *arXiv preprint arXiv:2412.05265*.

Parker-Holder, J.; Jiang, M.; Dennis, M.; Samvelyan, M.; Foerster, J.; Grefenstette, E.; and Rocktäschel, T. 2024. Syllabus: Portable Curricula for Reinforcement Learning Agents. In *Reinforcement Learning Conference*.

Phan, A. V.; and Nguyen, T. T. H. 2025. Training a Minesweeper Agent Using a Convolutional Neural Network. *Applied Sciences*, 15(5): 2490.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Sutton, R. S.; and Barto, A. G. 2018. *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 2nd edition.

Zhang, H.; Chen, C.; Xu, Z.; Yu, L.; Yu, Y.; et al. 2025. β-DQN: Improving Deep Q-Learning By Evolving the Behavior. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems*, 2317–2325.