Multi-Strategy Rainbow DQN Implementation: An Ablation Study on Discrete Control and Vision Tasks

Mikey Ferguson

December 2025

Abstract

This project implements a modified "Rainbow" Deep Q-Network (DQN) agent integrating Distributional Reinforcement Learning (C51), Magnetic Mirror Descent (MMD), and KL-Divergence regularization. A full ablation study was conducted across 2^4 (16) configurations to analyze the impact of each technique. The agent utilizes a hybrid architecture, dynamically switching between a Multi-Layer Perceptron (MLP) for vector-based environments and a Convolutional Neural Network (CNN) for visual grid-world tasks. The agent was evaluated on CartPole-v1, Leduc Hold'em, and MiniGrid environments to test control stability, imperfect information handling, and sparse-reward exploration.

1 Introduction

Reinforcement Learning (RL) stability is often compromised by the deadly triad of function approximation, bootstrapping, and off-policy learning. This project aims to mitigate these instabilities by implementing a subset of the Rainbow DQN enhancements. Specifically, we investigate whether tethering the online network to the target network (Magnetic Mirror Descent) and enforcing policy closeness (KL Penalty) can improve convergence in diverse domains ranging from simple control to visual navigation.

2 Methodology

The agent is built upon a standard DQN foundation with the following modular enhancements:

2.1 Distributional RL (C51)

Instead of estimating a single scalar Q-value, the network outputs a categorical distribution over a support set of atoms (V_{min} to V_{max}). This captures the variance and multi-modality of the value function, providing a richer learning signal.

2.2 Adversarial & Regularization Modifications

- **Magnetic Mirror Descent (MMD):** A regularization term is added to the loss function, penalizing the Euclidean distance between the online network weights and the target network weights. This prevents the online policy from diverging too rapidly during updates.
- **KL Penalty:** A penalty term based on the Kullback-Leibler divergence is applied to keep the current policy distribution close to the target distribution, smoothing the learning trajectory.

2.3 Hybrid Architecture

To handle diverse environments, the agent implements a hybrid neural network:

- **Vector Head (MLP):** For environments like CartPole and Leduc, the state is processed via standard fully connected layers.
- **Visual Head (CNN):** For MiniGrid, the agent utilizes a 3-layer Convolutional Neural Network to process $7 \times 7 \times 3$ grid images before feeding the features into the value heads.

3 Experimental Setup

The ablation study evaluated 16 configurations (permutations of C51, Delayed Update, Magnet, and KL Penalty) across three domains:

1. **CartPole-v1:** Dense reward control.
2. **Leduc-v0:** discrete, imperfect information card game.
3. **MiniGrid (Empty & FourRooms):** Visual navigation with sparse rewards.

4 Results and Analysis

4.1 Control Tasks: CartPole-v1

The agent demonstrated robust learning in the control domain. As shown in Figure 1, nearly all configurations successfully converged to the maximum reward of 500. This validates the correctness of the underlying C51 and Magnet implementations.

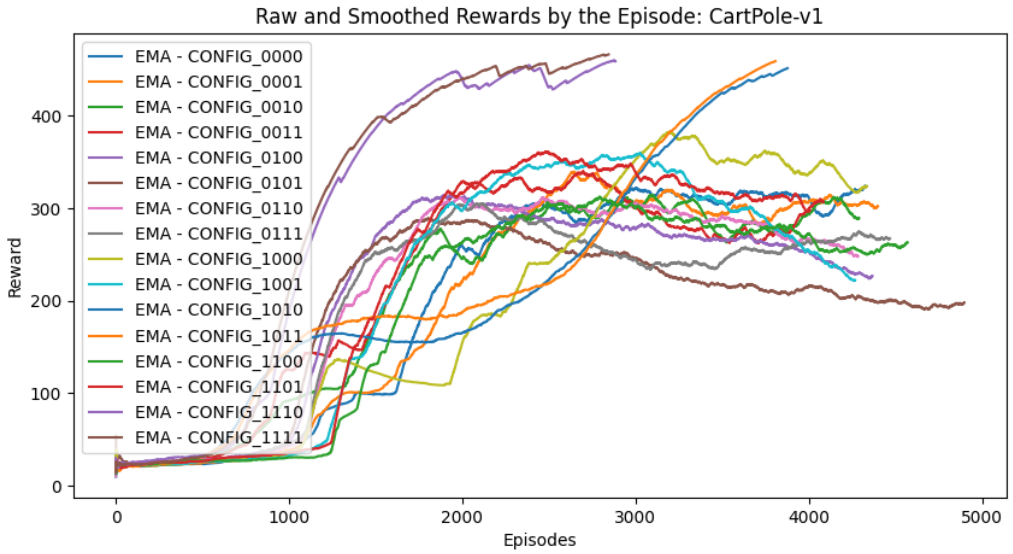

Figure 1: Training curves for CartPole-v1. The rapid ascent to 500 reward demonstrates the stability of the Rainbow implementation.

4.2 Visual Navigation: MiniGrid-Empty-8x8

The DQN architecture proved highly effective in the fully observable empty room. Figure 2 shows near-instant convergence to optimal pathing (Reward ≈ 1.0). A visual CNN hybrid architecture was not even necessary to extract the spatial features (walls, goal) from the pixel input.

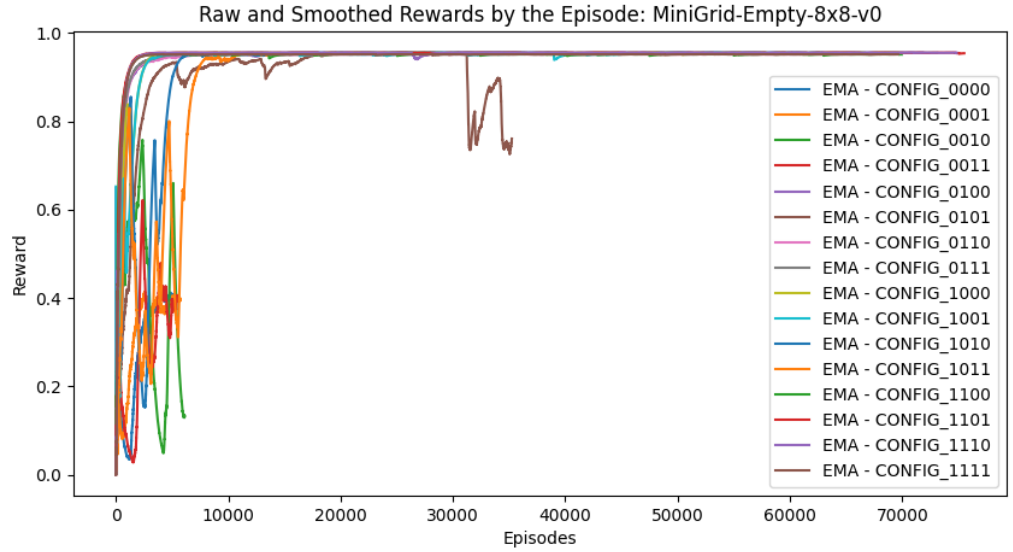

Figure 2: Training curves for MiniGrid-Empty-8x8. The agent quickly learned to interpret visual inputs and navigate to the goal.

4.3 Imperfect Information: Leduc Hold'em

In the stochastic domain of Poker, the agent demonstrated the ability to not only survive but dominate a random opponent. As shown in Figure 3, while some baselines hovered near a break-even point (Reward ≈ 0), the most effective configurations (incorporating C51 and Magnet) achieved a sustained positive mean reward (≈ 1.5). This indicates that the agent successfully moved beyond a simple Nash Equilibrium strategy and learned to aggressively exploit the sub-optimal, random betting patterns of its adversary.

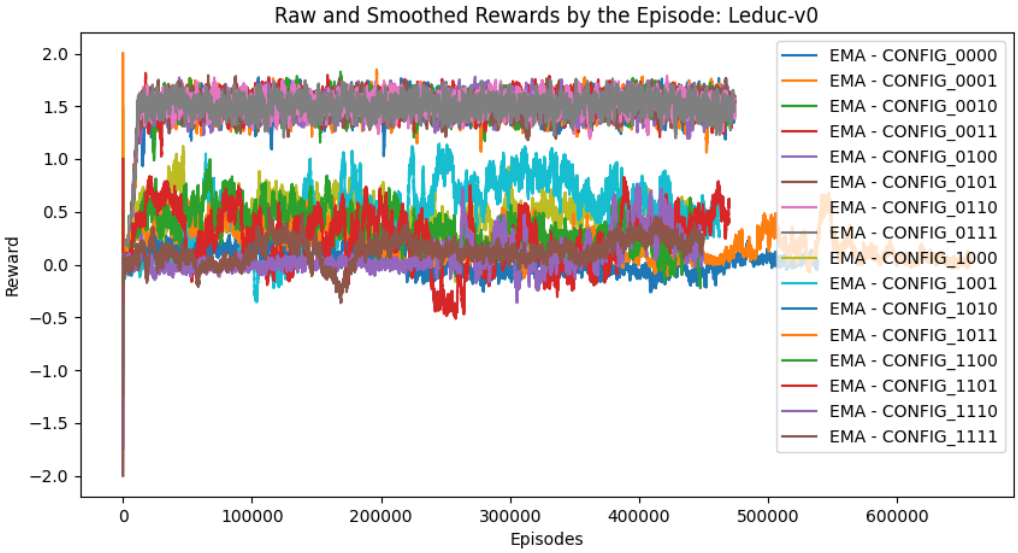

Figure 3: Training curves for Leduc-v0. Top-performing configurations consistently extracted positive value from the random opponent.

4.4 The Challenge of Sparse Rewards: FourRooms

The FourRooms environment presented a significant challenge. A hybrid CNN architecture was used, but as seen in Figure 4, the agent struggled to maintain a high reward policy.

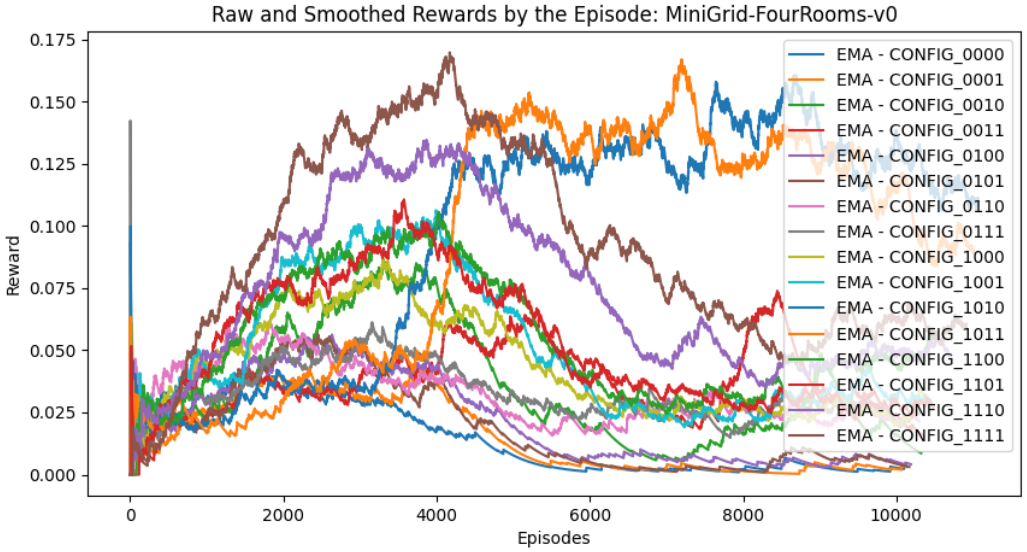

Figure 4: Training curves for MiniGrid-FourRooms. Performance decayed as exploration (epsilon) decreased.

5 Discussion

The failure in FourRooms highlights the *Exploration-Exploitation Dilemma*. The agent occasionally found the goal (peaking at 0.17 reward), but the epsilon decay schedule (50,000 frames) forced the agent to exploit its policy too early. In a sparse-reward maze, if the agent has not firmly memorized the path to the goal before exploration ends, the policy collapses to a local optimum (staying still or hitting walls) to avoid negative step penalties.

6 Conclusion

The implemented Multi-Strategy Rainbow Agent successfully generalized across vector and visual domains, solving CartPole and MiniGrid-Empty efficiently. While standard Rainbow techniques improve stability, they do not inherently solve the sparse reward problem found in complex mazes. Future work would require intrinsic motivation modules (such as RND or ICM) to encourage exploration beyond the epsilon-greedy horizon.