1. Baseline CartPole Implementation (Previous Version)

In the initial version of this project, the CartPole environment was solved using a Deep Q-Network (DQN) with a single neural network for action-value estimation. The network consisted of three fully connected layers with dimensions $4 \rightarrow 48 \rightarrow 48 \rightarrow 2$ and used ReLU activations in the hidden layers.

The agent was trained using the Bellman equation with mean squared error (MSE) loss, and network parameters were updated using the RMSProp optimizer. An experience replay buffer was employed to improve sample efficiency, and actions were selected using an epsilon-greedy strategy with epsilon decaying over time from a high initial value to a fixed minimum.

Training results showed steady improvement in episode rewards, ultimately reaching the maximum score of 500. However, due to the use of a single Q-network without a target network, learning exhibited intermittent instability, reflected in occasional reward drops and large loss fluctuations. These effects are consistent with known limitations of vanilla DQN.

2. Modified Learning Mechanism: Conductance-Based Manhattan Weight Updates

In the updated version of this project, the overall reinforcement learning framework remains unchanged. The agent still uses DQN with experience replay, epsilon-greedy exploration, and Bellman-based MSE loss. The key modification lies in the mechanism used to update the neural network weights.

Instead of employing a conventional gradient-based optimizer such as RMSProp, the updated implementation introduces a Manhattan-style, conductance-based weight update method. This approach is motivated by hardware-aware learning scenarios, where weights are constrained to discrete values and cannot be updated using continuous gradient magnitudes.

Each network weight is represented as the difference between two conductances, denoted as G^+ and G^-. The effective weight is defined as $W = G^+ - G^-$. Both conductances are selected from a finite, discrete set of values stored in an external CSV file. Rather than directly modifying conductance values, the implementation maintains integer indices that point to entries in the conductance table. These indices represent the current physical state of each synaptic element.

Two real conductance datasets were evaluated in this work. The first dataset contains 50 discrete conductance levels, while the second contains 290 levels, providing finer resolution. In addition, a synthetic linearly spaced conductance array was tested as a reference case. The agent successfully learned the CartPole task using the linear dataset, confirming the correctness of the update mechanism in an idealized setting.

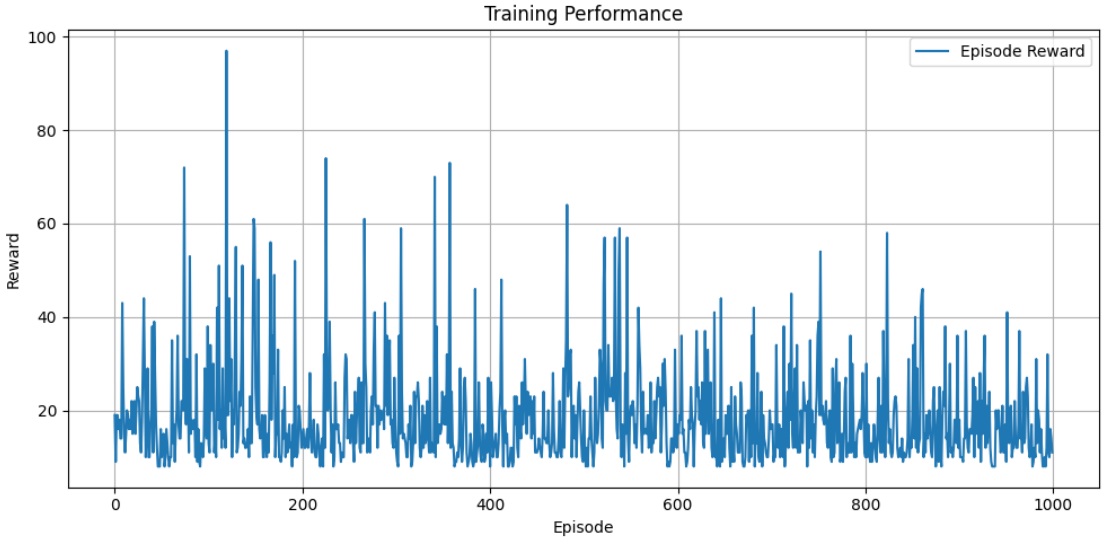
During training, gradients are computed normally through backpropagation of the MSE Bellman loss. However, weight updates depend only on the sign of the gradient. When the gradient of a weight is positive, the weight should decrease, and the algorithm preferentially increases the G^- conductance. When the gradient is negative, the weight should increase, and the algorithm preferentially increases the G^+ conductance. If the preferred conductance index reaches its maximum allowable value, a fallback mechanism reduces the opposite conductance to preserve the desired update direction. All indices are clamped to valid ranges to ensure stability.

After updating the indices, the corresponding conductance values are retrieved from the dataset and the weight is rewritten as the difference between G^+ and G^-. In this formulation, the learning rate and gradient magnitude no longer influence the update size. Each update corresponds to a single discrete step in conductance space, resulting in a quantized learning process.

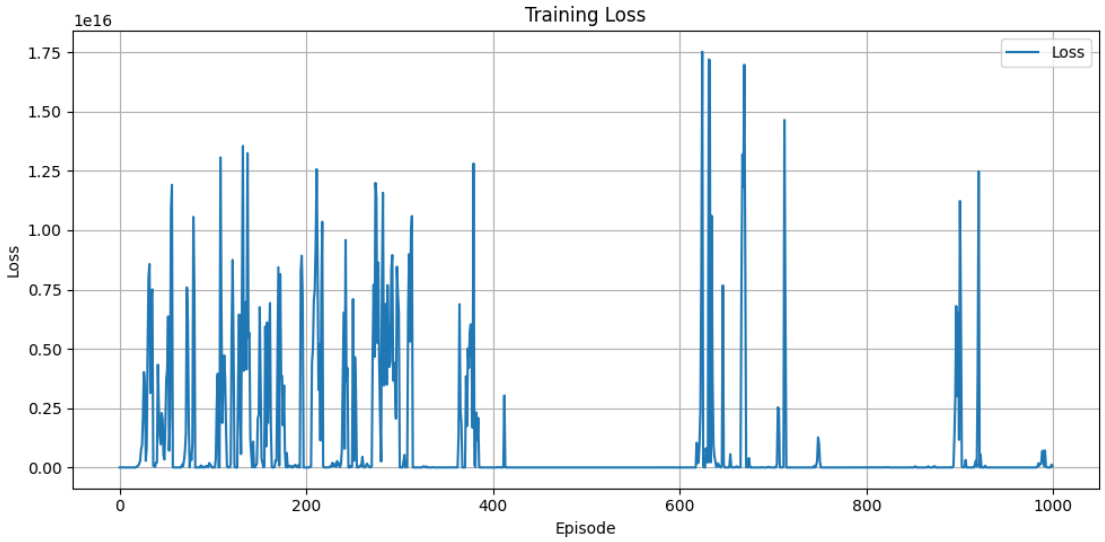
The neural network architecture remains unchanged in terms of layer dimensions (4 → 48 → 48 → 2), but the ReLU activations used in the baseline model were replaced with LeakyReLU activations. This change improves gradient flow and reduces the likelihood of inactive neurons, which is particularly important under discrete weight update dynamics.

By keeping all other components of the reinforcement learning pipeline unchanged, this design isolates the effects of the conductance-based Manhattan update rule, allowing a clear comparison with the baseline optimizer-driven approach.

3. Results with Conductance Dataset V1 (50 Discrete Levels)



Reward



Loss

When training the agent using the first conductance dataset containing 50 discrete values, learning behavior is highly unstable. The training loss exhibits frequent and very large spikes

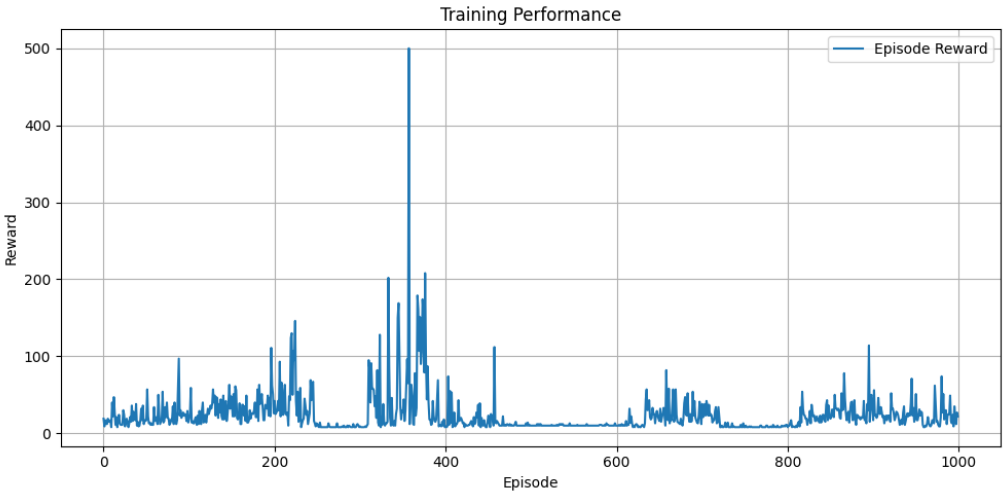
throughout the entire training process. Although the loss occasionally drops close to zero, these periods are repeatedly interrupted by abrupt increases, indicating unstable Q-value updates.

This instability is caused by the combination of three factors: the use of a single Q-network without a target network, sign-based Manhattan updates, and the coarse resolution of the conductance dataset. Because each weight update corresponds to a relatively large discrete change in conductance, Q-values can easily overshoot their targets, producing large temporal-difference errors.

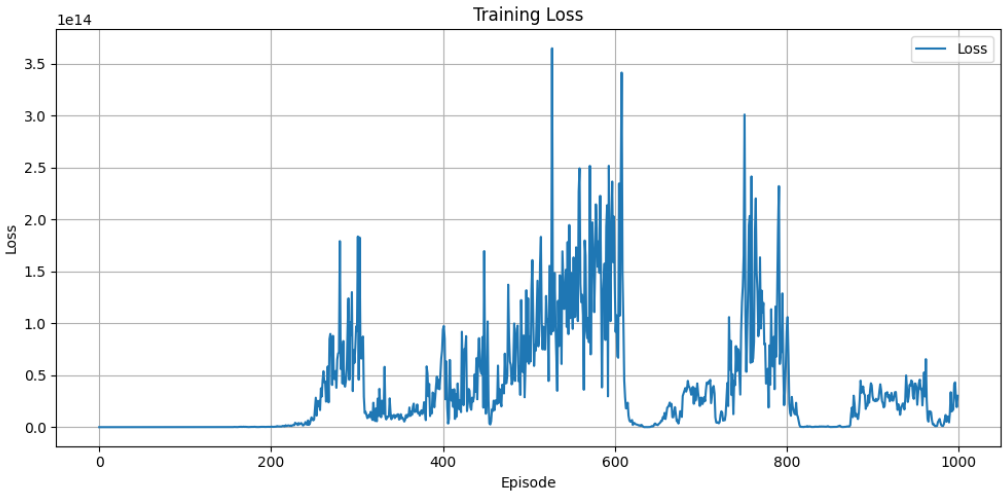
The reward curve shows no clear upward trend over training. Episode rewards fluctuate strongly, with occasional spikes to moderate values, but these improvements are not sustained. As a result, the agent fails to converge to a stable policy and does not achieve consistent task performance.

Overall, these results indicate that a conductance resolution of 50 levels is insufficient for stable learning in the CartPole task when using discrete, sign-based weight updates.

4. Results with Conductance Dataset V2 (290 Discrete Levels)



Reward



Loss

When the agent is trained using the higher-resolution conductance dataset containing 290 discrete values, learning behavior improves noticeably compared to the 50-level case. The training loss still exhibits fluctuations and occasional spikes, but these are significantly smaller in magnitude and occur less frequently. Periods of low loss are more sustained, indicating more stable Q-value updates.

The reward curve shows clear evidence of learning. Unlike the low-resolution case, the agent is able to reach high episode rewards, including a brief episode achieving the maximum score of 500. Although this performance is not consistently maintained, the presence of sustained reward improvements demonstrates that the agent can successfully learn effective control policies under finer conductance resolution.

The remaining instability can be attributed to the use of a single Q-network and discrete, sign-based updates, which still limit smooth convergence. However, the increased number of conductance levels allows smaller effective weight changes, reducing overshooting and improving learning stability.

Overall, these results demonstrate that increasing conductance resolution plays a critical role in enabling successful reinforcement learning with Manhattan-style, discrete weight updates. The 290-level dataset provides sufficient granularity for the agent to approach task mastery, in contrast to the failure observed with only 50 conductance levels.