CartPole: Deep Q-Learning vs Double DQN on CPU and GPU

Author: Antwain M. Sparks and Vojislav Stojkovic, Morgan State University, Computer Science Department, Baltimore, MD 21251

# Abstract

Reinforcement learning (RL) is a branch of machine learning in which an agent learns decision-making through interaction with an environment. Q-Learning (QL) is a foundational RL algorithm that assigns values to state–action pairs without requiring a model of the environment. Deep Q-Learning (DQL) extends QL by combining it with deep neural networks, enabling agents to handle larger and more complex state spaces.

The CartPole environment serves as a classic benchmark for evaluating RL methods, where an agent must balance an inverted pendulum by applying discrete forces to a cart. Despite its apparent simplicity, CartPole provides a meaningful platform for comparing algorithm performance.

We implement and evaluate DQL under three configurations: a CPU-based baseline, a GPU-accelerated DQN, and a GPU-accelerated Double DQN (DDQN). While GPU acceleration speeds up training, algorithmic improvements such as DDQN are necessary for stability and convergence. The combined approach (GPU + DDQN) consistently solves CartPole at the maximum reward threshold.

# Methodology

• CartPole environment setup

• DQN vs DDQN algorithms (target network decouples action selection and evaluation)

• CPU vs GPU execution differences

• Training setup: 800 episodes, epsilon decay to 0.01, replay buffer=100k, batch=64, lr=5e-4

# Results (Summary)

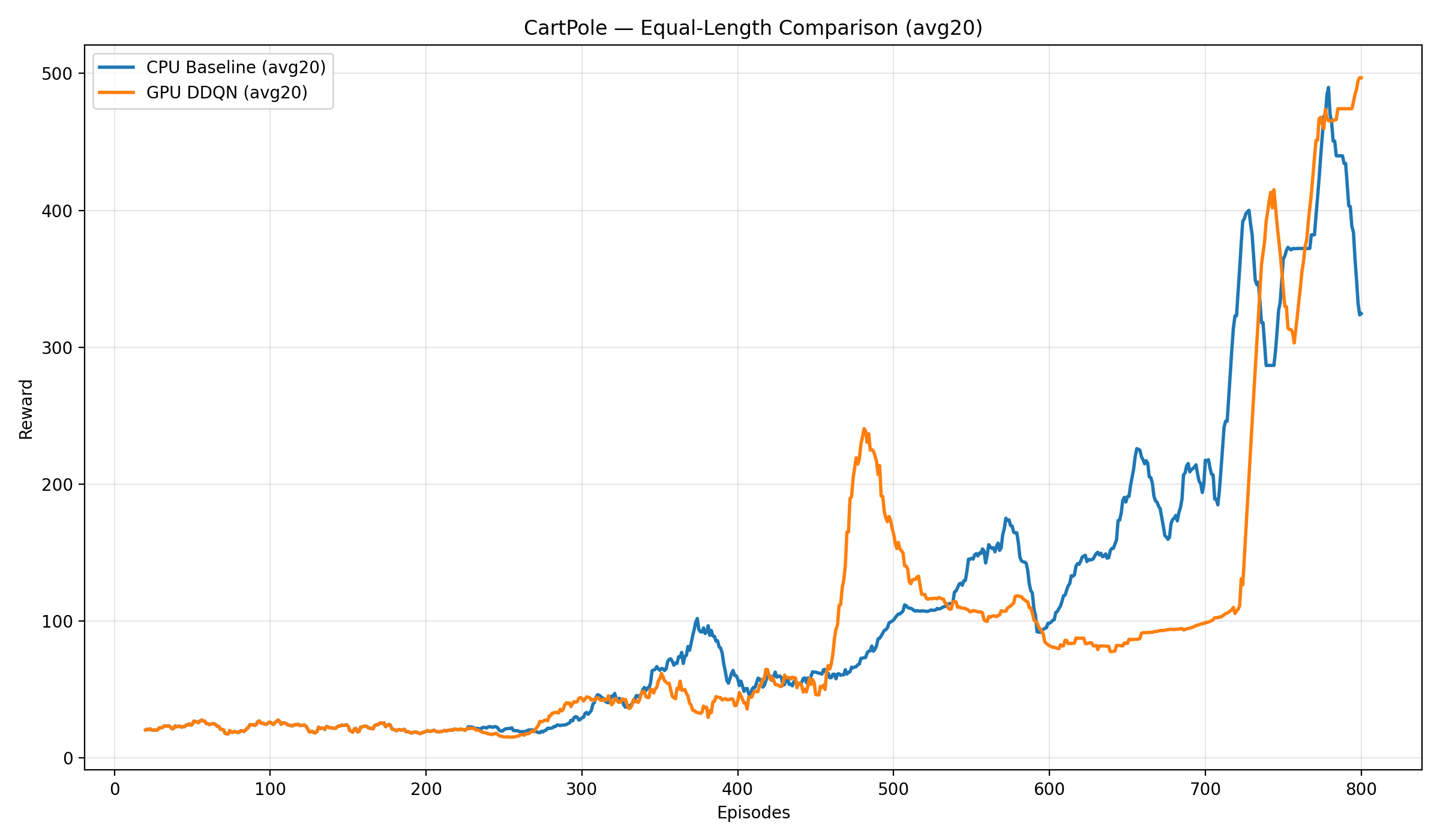
Model Device Episodes to Solve Final Avg Reward Training Time (mins)

DQN (CPU baseline) CPU >600 (unstable) ~183 ~8.5

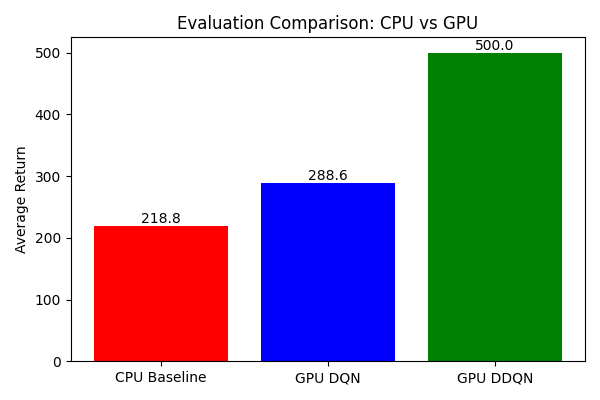
DQN (GPU) GPU 800 (not solved) 288.6 ~7

DDQN (GPU) GPU ~480 (solved) 500.0 ~8.5

# Charts

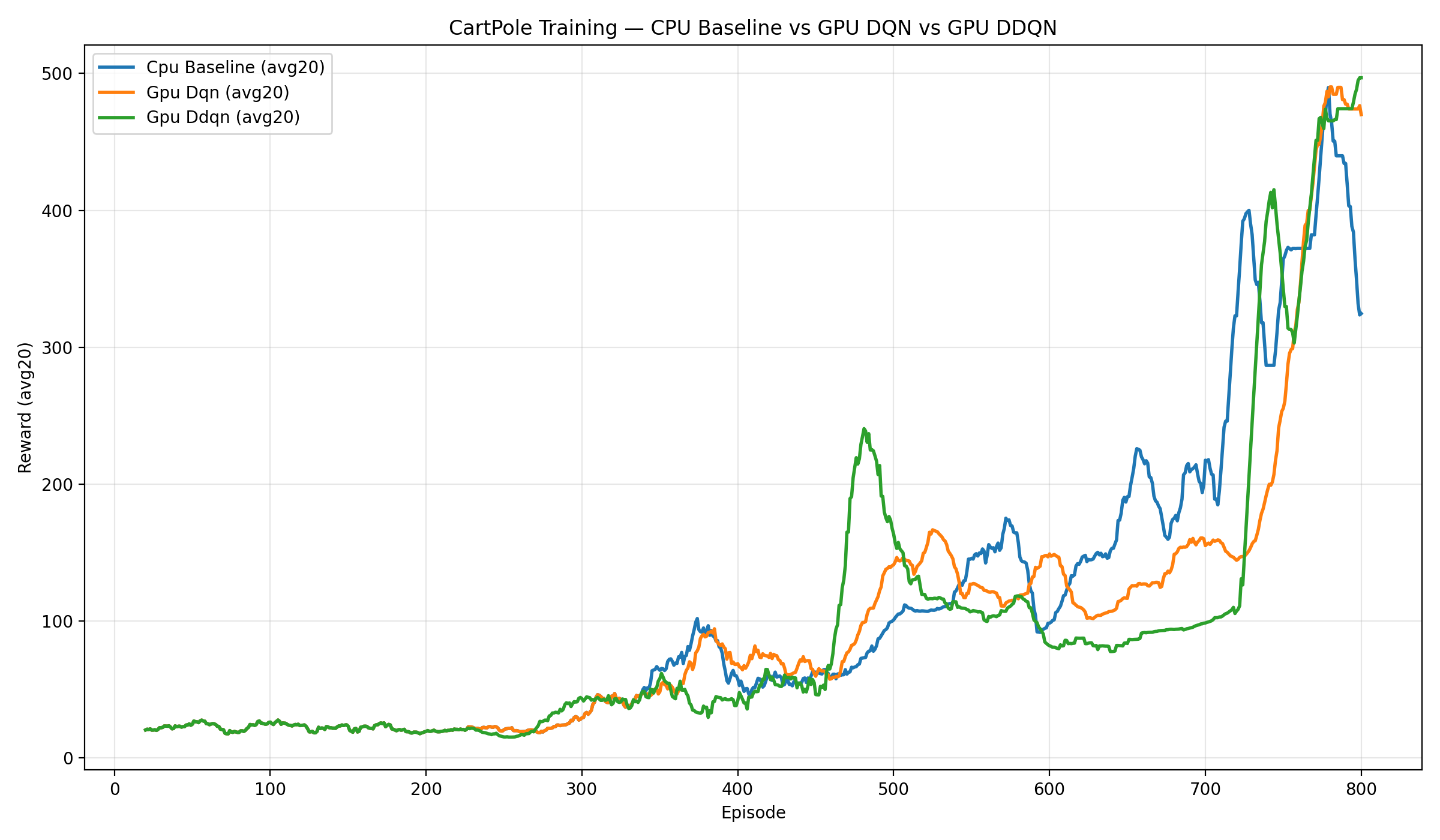


Training rewards (moving avg) — CPU vs GPU DQN vs GPU DDQN

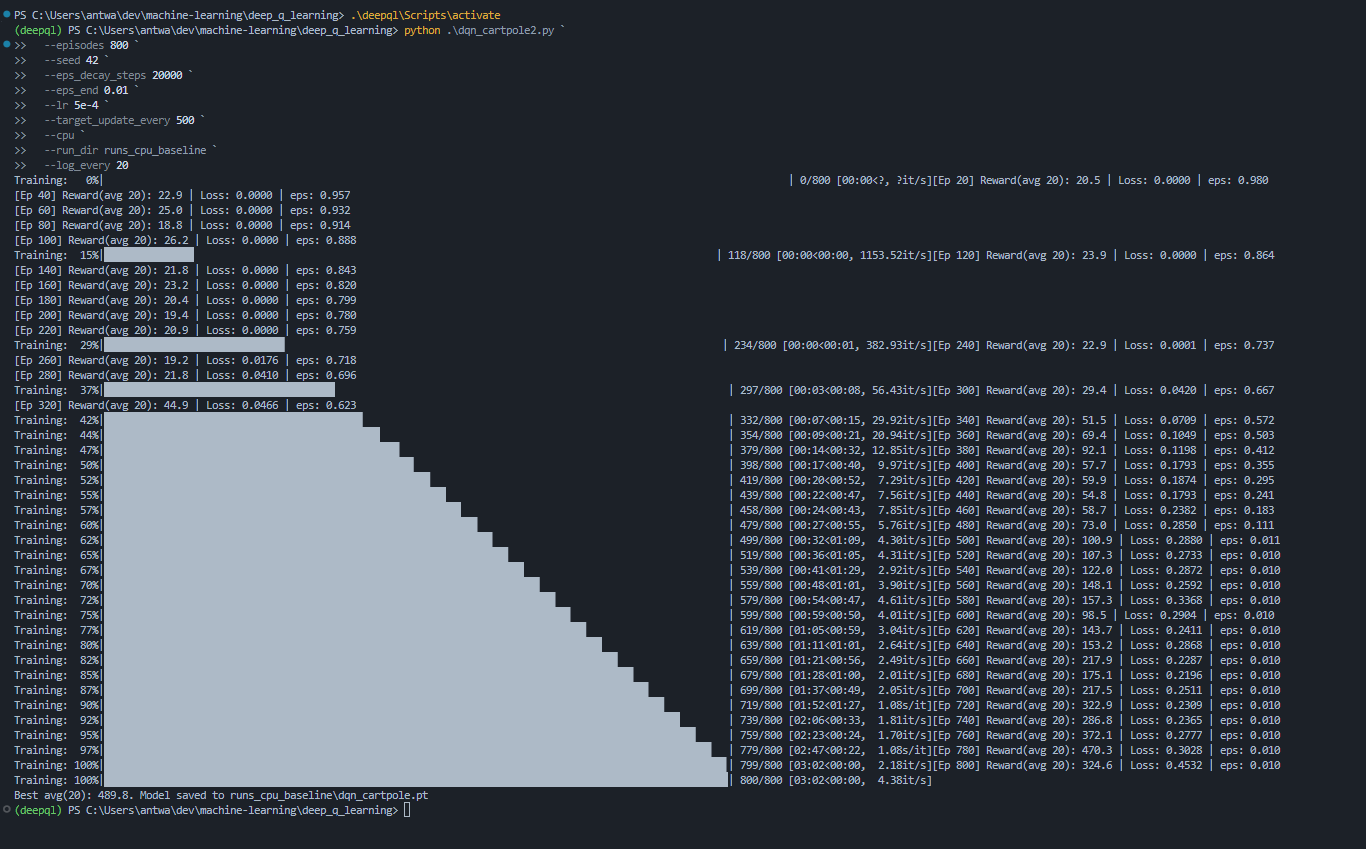


Evaluation comparison: average return (CPU DQN vs GPU DQN vs GPU DDQN)

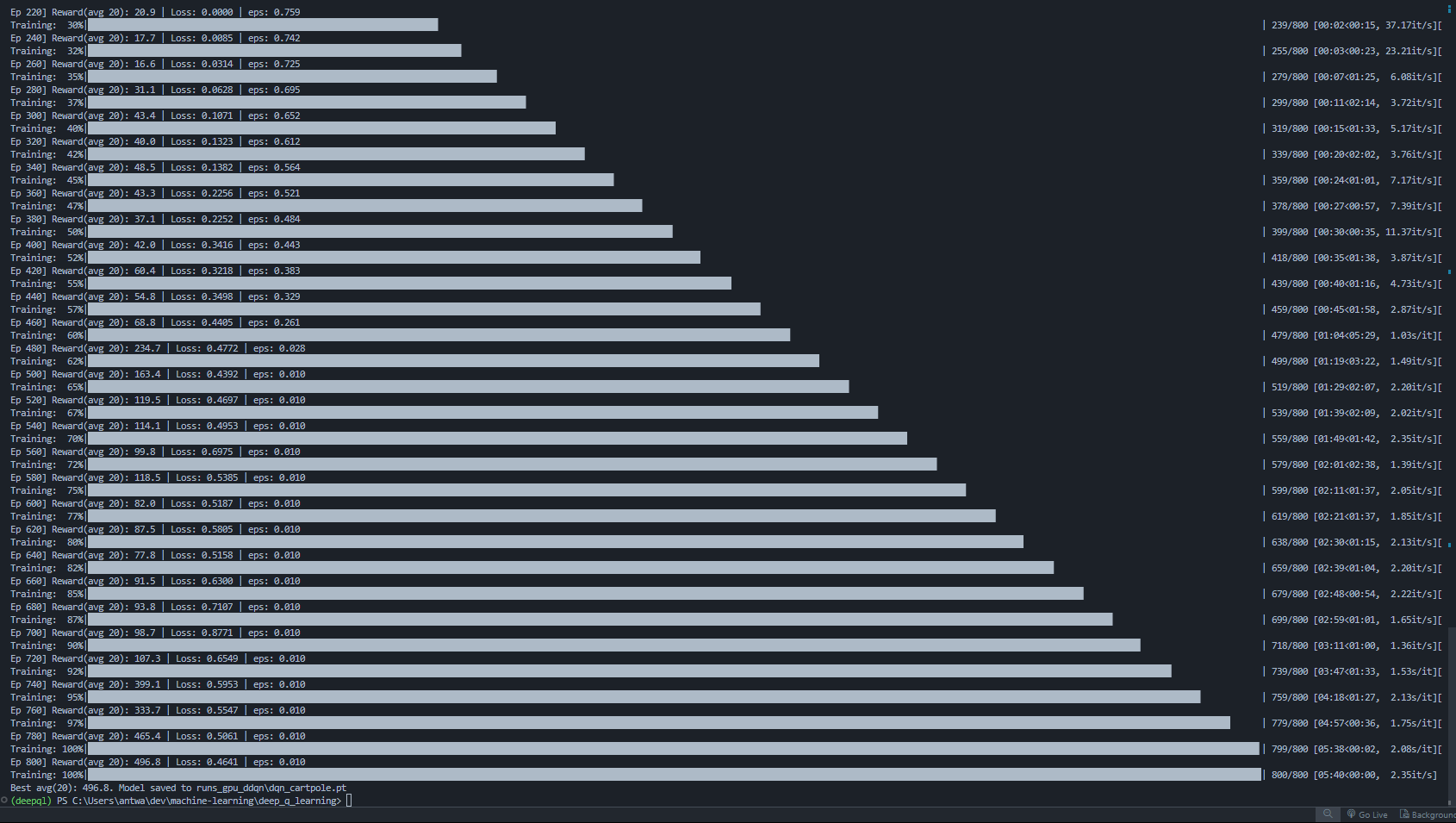
# Figures (Screenshots & Extras)



auto\_track\_comparison



cpu\_tuned



gpu\_ddqn



gpu\_dqn

# Technical Specs

• CPU: Intel i7-12700H (20 cores)

• RAM: 16 GB

• GPU: NVIDIA GeForce RTX 3050 Ti Laptop GPU (4 GB VRAM)

• CUDA: 12.4

• Python: 3.11.9, PyTorch 2.6.0+cu124

# Key Takeaways

• CPU DQN: slow, unstable, fails to solve.

• GPU DQN: faster but underperforms vs DDQN.

• GPU DDQN: consistently solves CartPole (avg return 500).

• Lesson: Algorithmic improvements (DDQN) + GPU acceleration deliver the best results.

# Conclusion

GPU acceleration helps, but algorithmic refinement (DDQN vs DQN) is essential to reliably solve CartPole. The combination of DDQN with GPU provided the strongest performance.