Reinforce Learning

2025 March 28

Deep Q-Networks for Atari Game Playing

Sheng Zhang, Hepple Xi, Wenqing Yu, Hanyu Liu

Motivation

- Classic game: Atari Breakout is a well-known game where players use a paddle to bounce a ball and break bricks.
- Simple actions, complex strategy
- Benchmark for RL research

Problem Statement

Problem Statement

- Environment: Atari BreakoutNoFrameskip-v4
  https://ale.farama.org/environments/breakout/

- Gym(ALE Atari Environment): more than 100 Atari Game Environment
  https://ale.farama.org/environments/

Problem Statement

- Input: 84×84 grayscale images (4-frame stack)
- Actions: NOOP(0), FIRE(1), LEFT(2), RIGHT(3)
- Reward: You score points by destroying bricks in the wall.

Problem Statement

Challenges:

- Long-Term Decision Making: Balancing exploration versus exploitation
- Data Efficiency:

  Online RL: Requires extensive interactions and long training times.

  Offline RL: Relies on the quality and distribution of the pre-collected dataset.

Problem Statement

Project Objectives:

- Train an agent to achieve high scores in Breakout using RL techniques.
- Compare Online (DQN) versus Offline (CQL) training approaches.

Methodology

Online RL: DQN

DQN Setup:

- Configured via d3rlpy.algos.DQNConfig
- Learning Rate: 1e-4
- Target network updated every 10,000 steps

Exploration & Replay:

- Epsilon-Greedy
- Replay buffer

Online RL: DQN

Training:

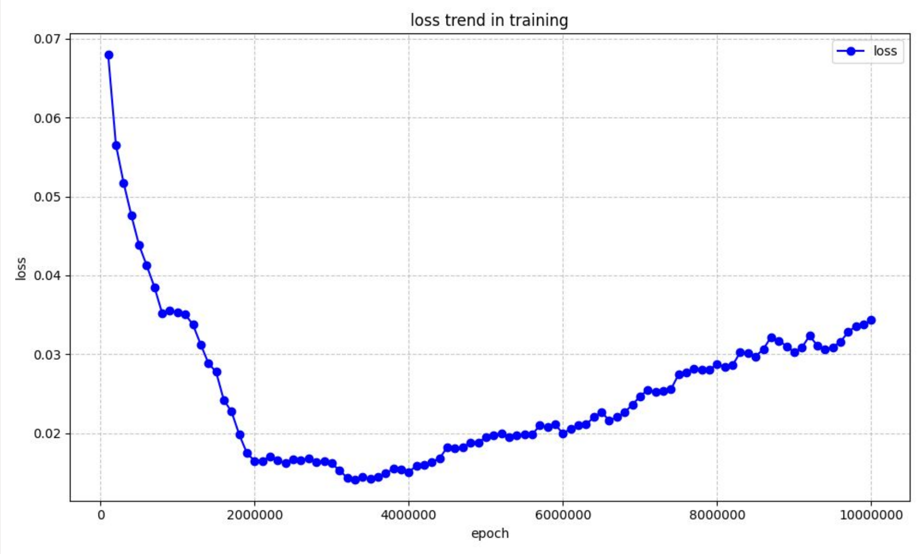
- Total of 10M steps with updates every 4 steps
- Separate environment for periodic performance evaluation

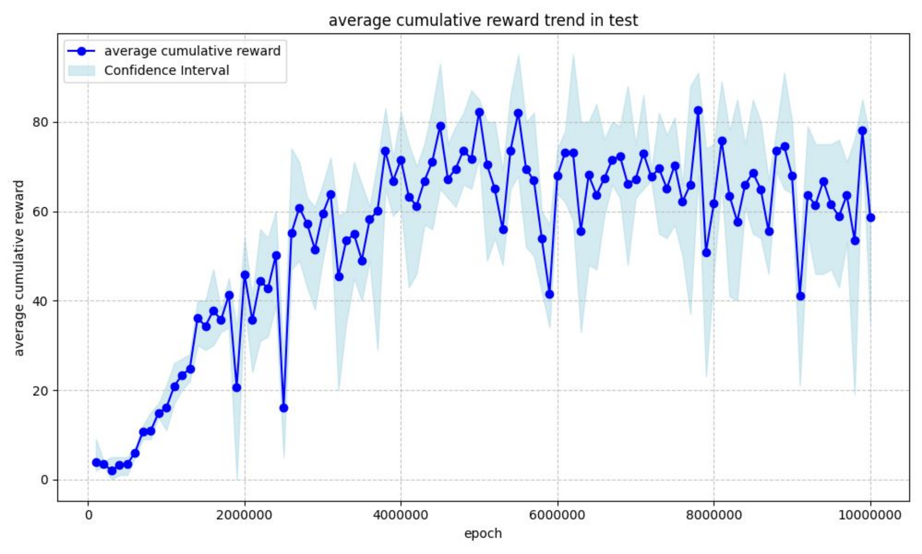
Metrics:

- Tracks average reward, training loss across episodes.

Online RL: DQN

Metrics:





Offline RL: CQL

Offline RL trouble: Out-of-Distribution

- State-action pairs that have never appeared in the dataset, resulting in suboptimal strategies or even loss of control

CQL Solution:

- Penalize excessive Q values to limit the range of Q

$$L_{CQL}(Q) = L_{DQN}(Q) + \alpha \cdot E_{(s,a)\sim D}[\log(\sum \exp Q(s,a')) - Q(s,a)]$$

Offline RL: CQL

Dataset & Preprocessing:

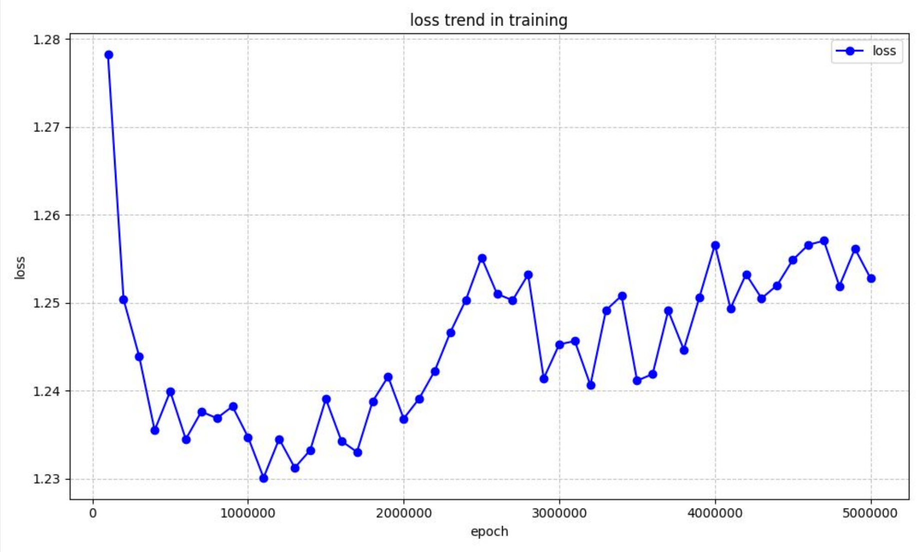
- Uses Google pre-collected Atari transitions (50% of available data)
- Maintains 4-frame stack structure

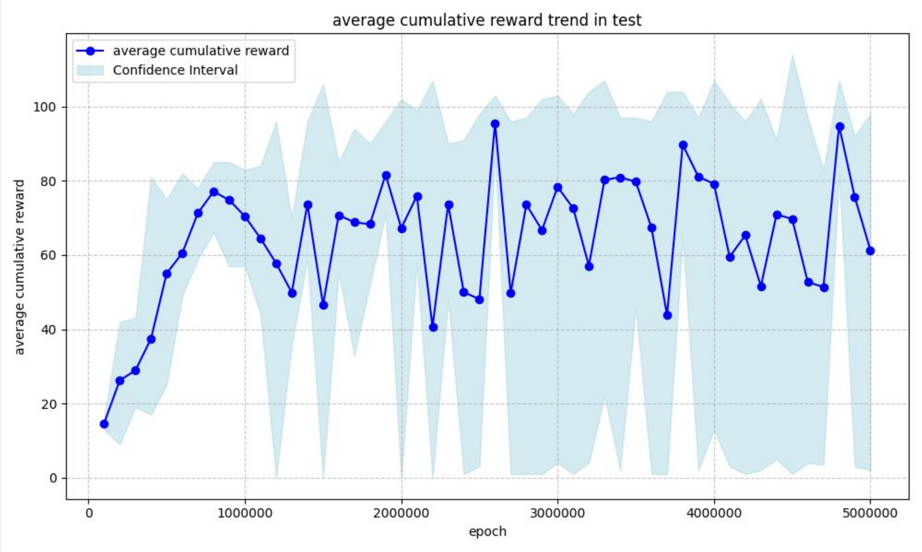
Training Configuration:

- Learn rate 6.25e-5, penalty discount 1.0
- Rewards clipping (-1.0 to 1.0)
- Target Q network update every 10K steps
- 5M steps in total and every 100K steps as a epoch

Offline RL: CQL

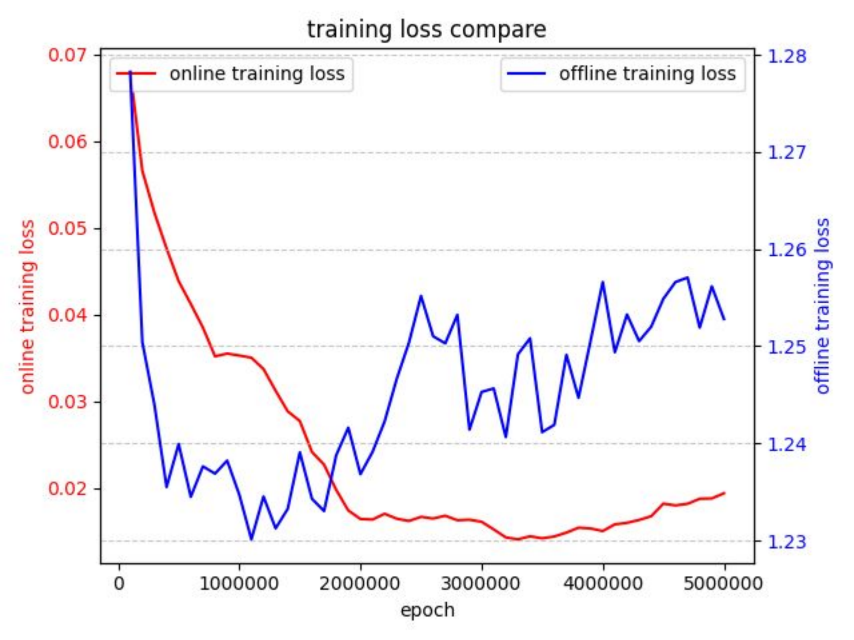
Metrics:

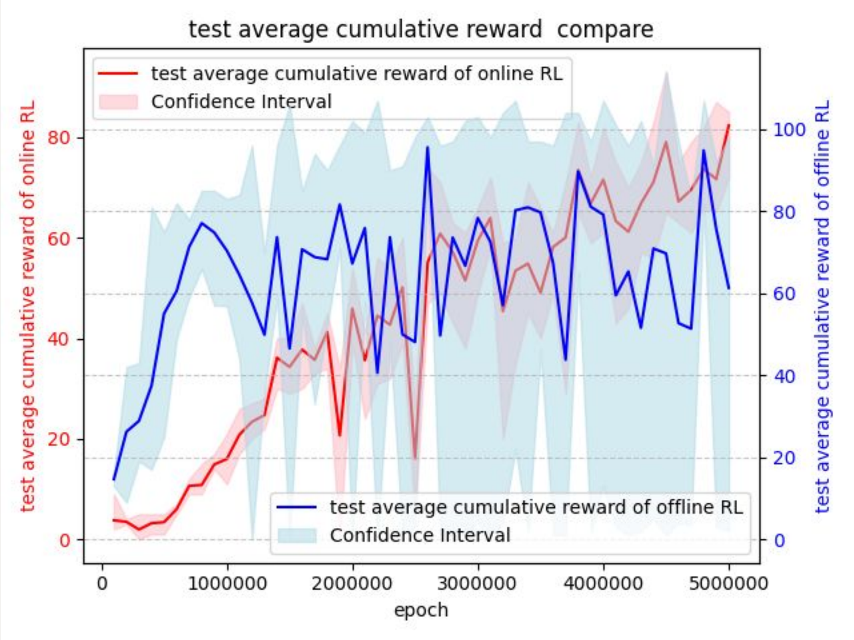




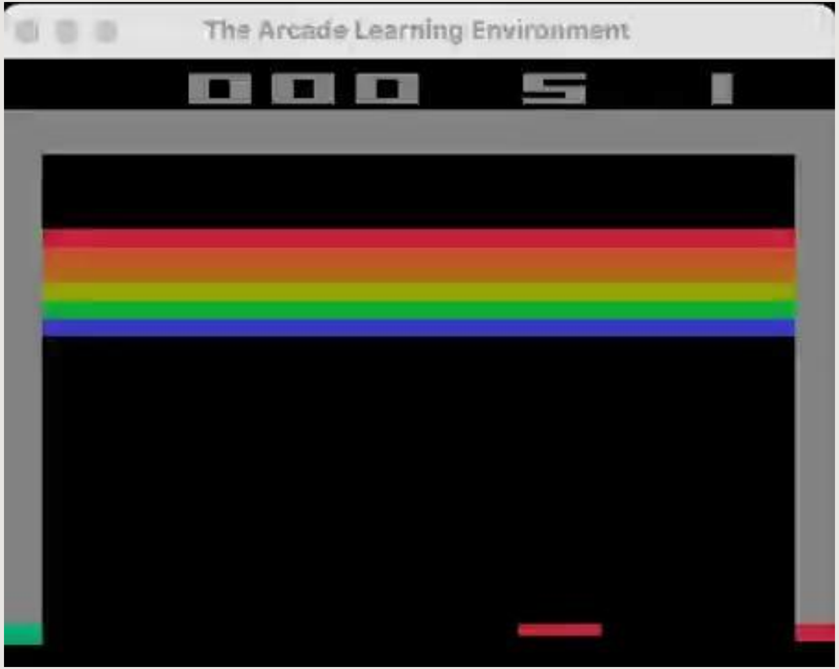
Online v.s. Offline

Comparison of DQN & CQL





Online RL



Offline RL

Next Steps

- Try to speed up DQN convergence (e.g. PER)
- Try adjusting the CQL penalty factor to mitigate reward fluctuation
- Combine online fine-tune to improve offline RL (e.g. AWAS)

THANK YOU FOR YOUR TIME

Sheng Zhang, Hepple Xi, Wenqing Yu, Hanyu Liu