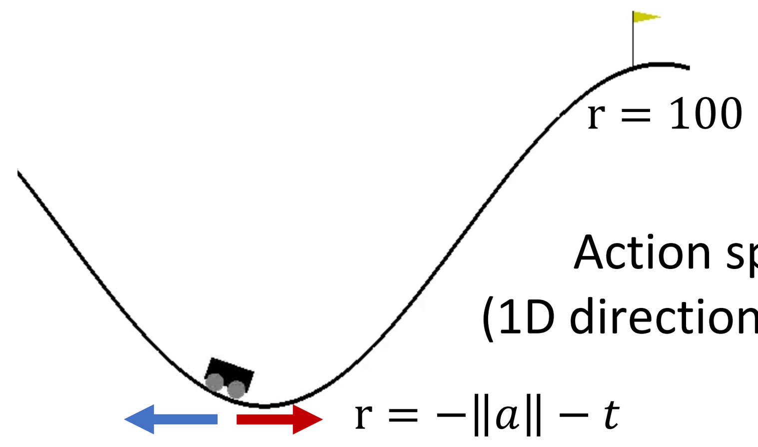
EE-568 Reinforcement Learning Project

Saibo Geng, Shuqi Wang, Yitao Xu, Liangze Jiang

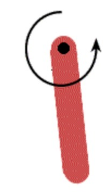
MountainCarContinuous



Action space
(1D directional force)

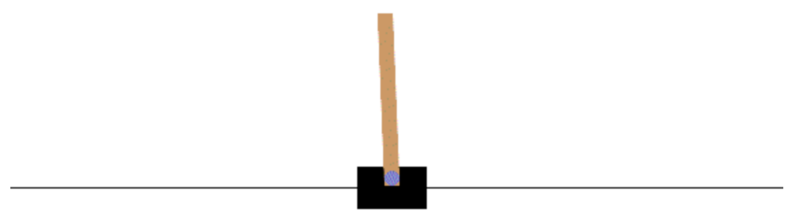
Pendulum

Action space
Torque applied to free end



CartPole

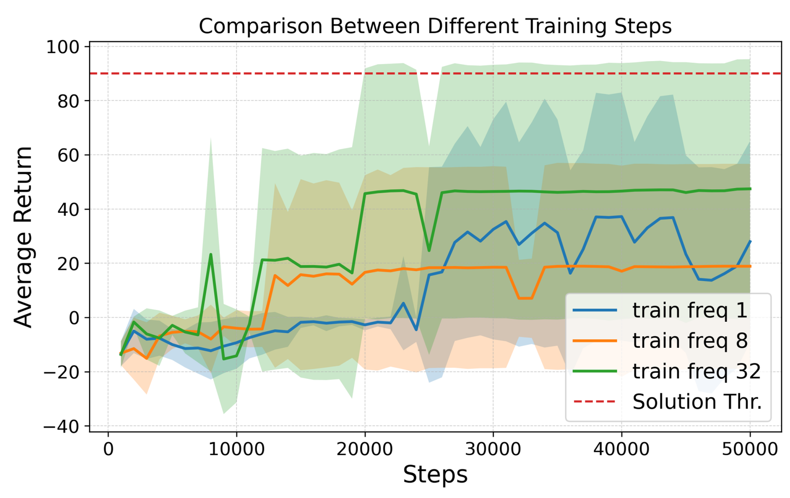
Action space
Left or right (discrete)

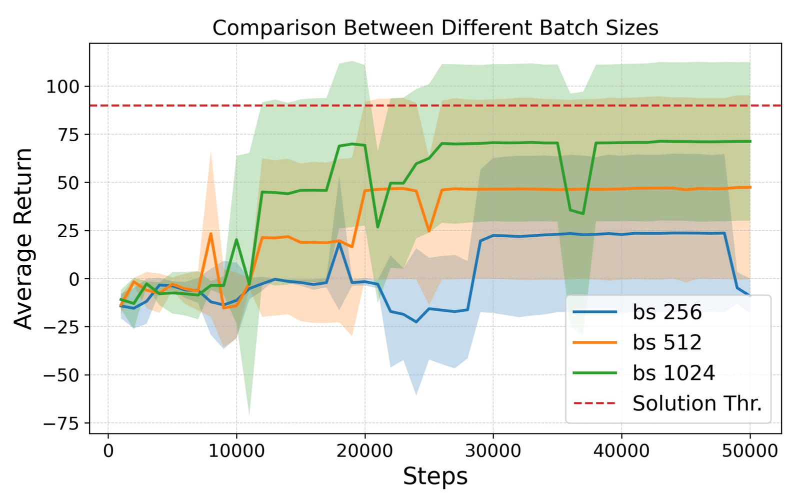


SAC

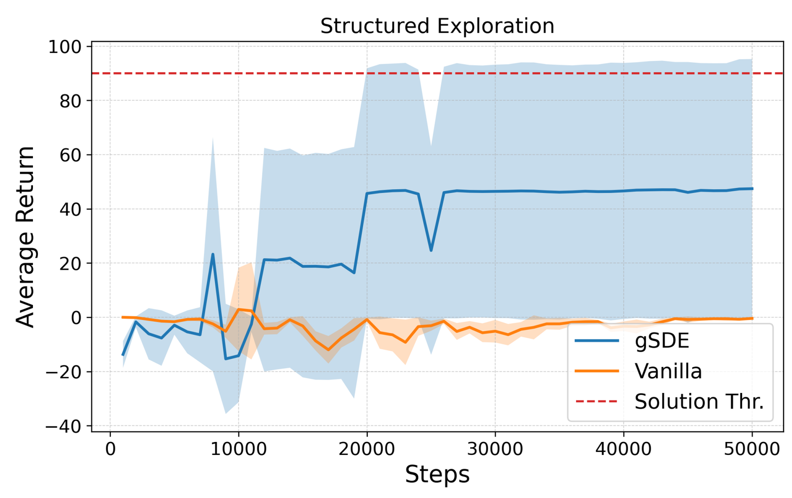
MountainCarContinuous

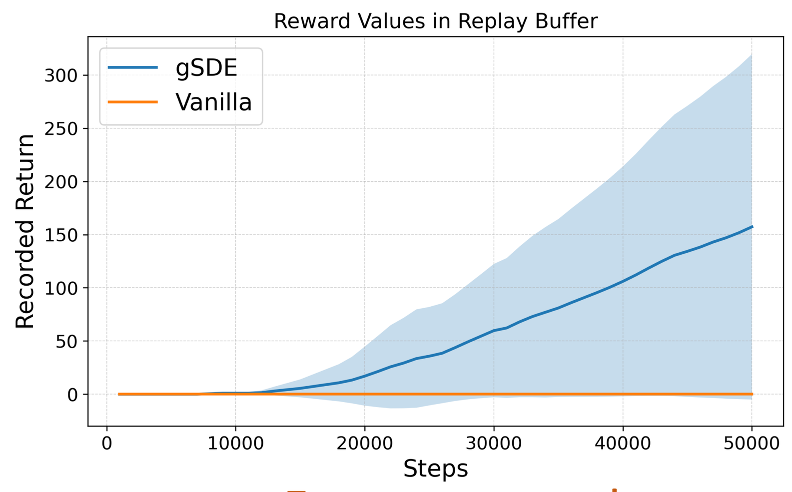
- Caveat: Extremely sparse reward. Can choose to stay/jitter instead of moving
- Key strategies:
  1. Structured exploration within an episode
  2. Learning happens sparsely, while each learning lasts longer



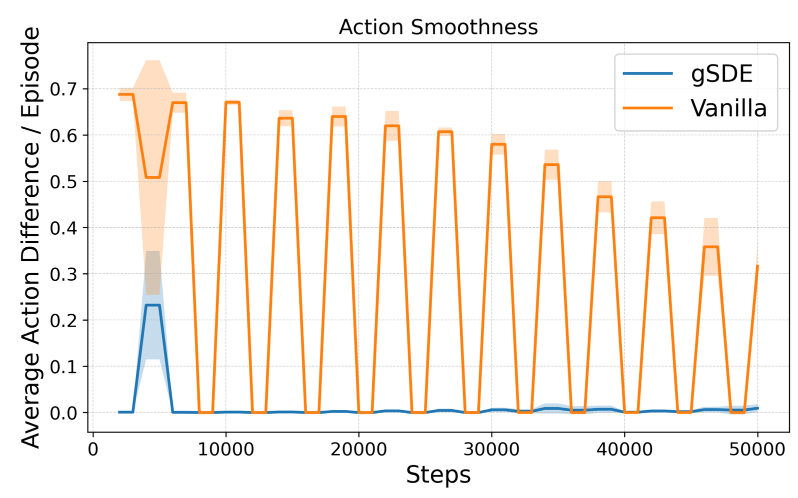


1. Structured exploration within an episode





Ensures reward



Keep same actions

Take away: Reaching the summit demands resolve!

TD3

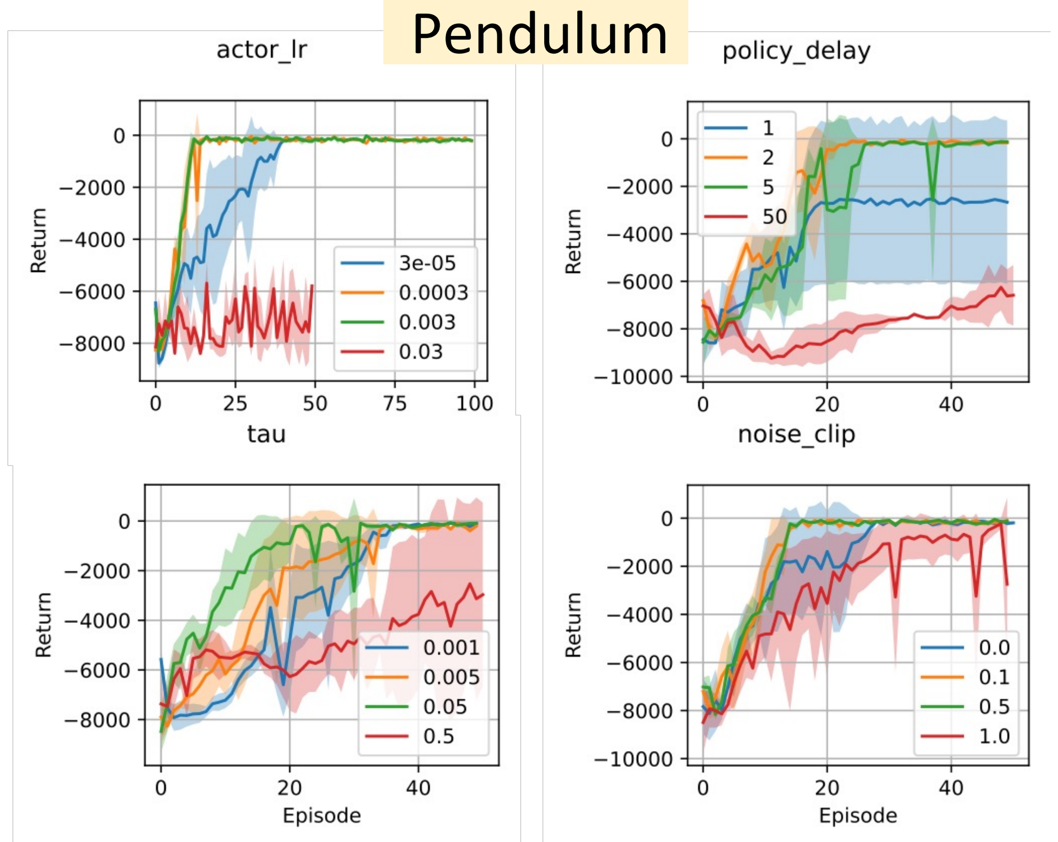
Algorithm 1 TD3

Initialize critic networks Q_{θ_1}, Q_{θ_2}, and actor network π_ϕ with random parameters θ_1, θ_2, ϕ
Initialize target networks $\theta'_1 \leftarrow \theta_1$, $\theta'_2 \leftarrow \theta_2$, $\phi' \leftarrow \phi$
Initialize replay buffer $\mathcal{B}$
for $t = 1$ **to** T **do**
 Select action with exploration noise $a \sim \pi_\phi(s) + \epsilon$, $\epsilon \sim \mathcal{N}(0, \sigma)$ and observe reward r and new state s'
 Store transition tuple (s, a, r, s') in $\mathcal{B}$

 Sample mini-batch of N transitions (s, a, r, s') from $\mathcal{B}$
 $\tilde{a} \leftarrow \pi_{\phi'}(s') + \epsilon$, $\epsilon \sim \text{clip}(\mathcal{N}(0, \tilde{\sigma}), -c, c)$
 $y \leftarrow r + \gamma \min_{i=1,2} Q_{\theta'_i}(s', \tilde{a})$
 Update critics $\theta_i \leftarrow \text{argmin}_{\theta_i} N^{-1} \sum (y - Q_{\theta_i}(s, a))^2$
 if t mod d **then**
  Update ϕ by the deterministic policy gradient:
  $\nabla_\phi J(\phi) = N^{-1} \sum \nabla_a Q_{\theta_1}(s, a)|_{a=\pi_\phi(s)} \nabla_\phi \pi_\phi(s)$
  Update target networks:
  $\theta'_i \leftarrow \tau \theta_i + (1 - \tau)\theta'_i$
  $\phi' \leftarrow \tau \phi + (1 - \tau)\phi'$
 end if
end for

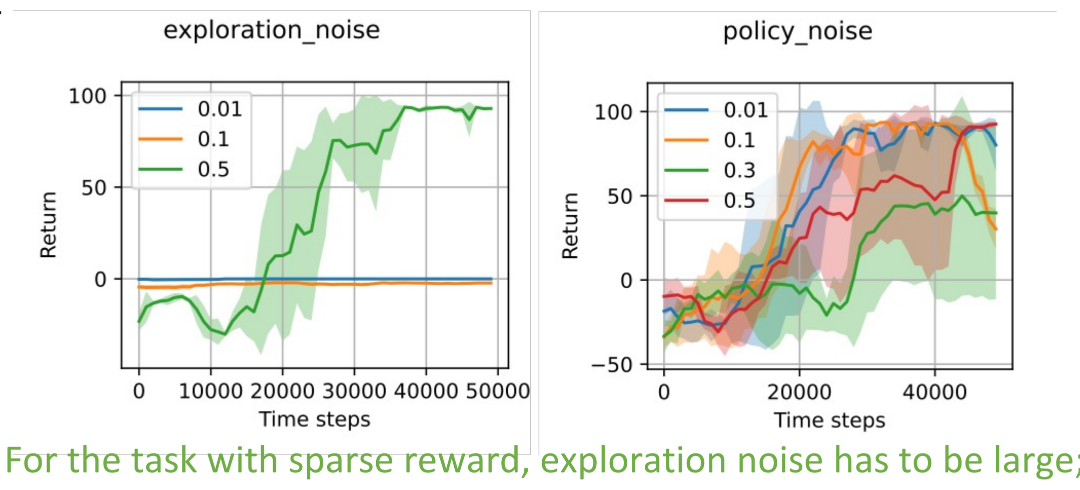
Fujimoto, Scott, Herke Hoof, and David Meger. "Addressing function approximation error in actor-critic methods." *International conference on machine learning*. PMLR, 2018.

Pendulum

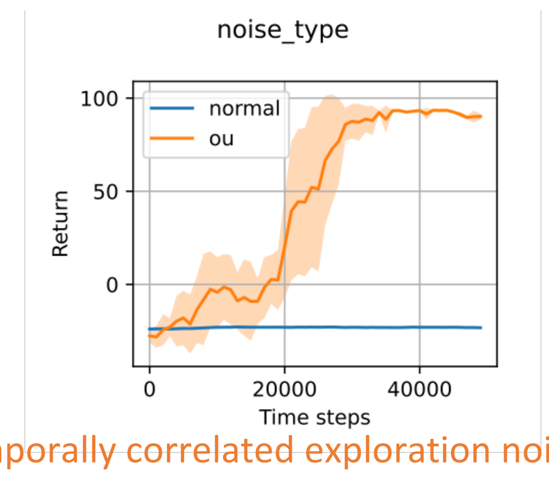


Learning rates, policy delay, tau of targets updating, policy noise, … all have an optimal range of values.

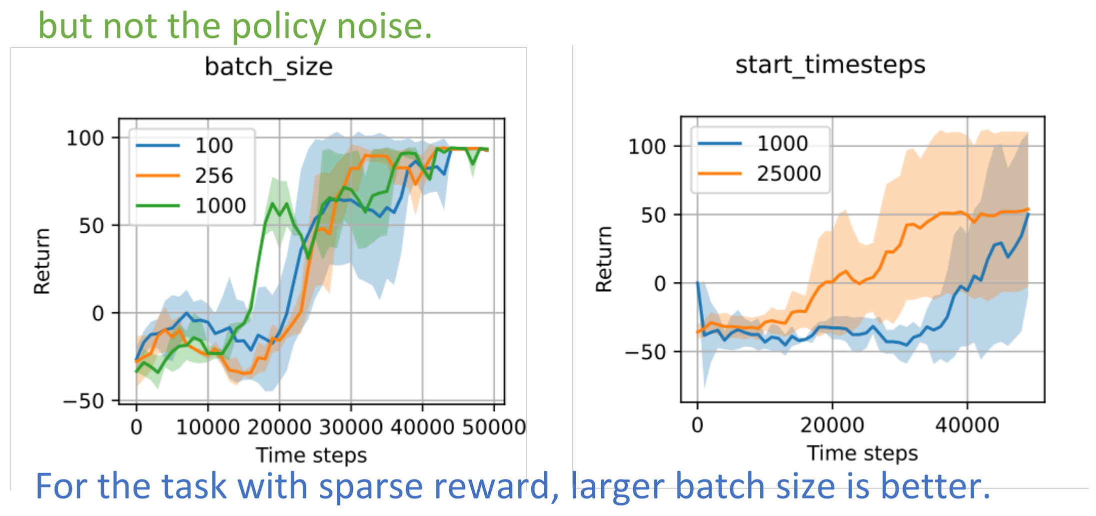
MountainCarContinuous (Sparse reward)



For the task with sparse reward, exploration noise has to be large; but not the policy noise.



Temporally correlated exploration noise (e.g., OU noise) is crucial to generate smooth dynamics.



For the task with sparse reward, larger batch size is better.
The time when training starts can be late to encourage exploration.

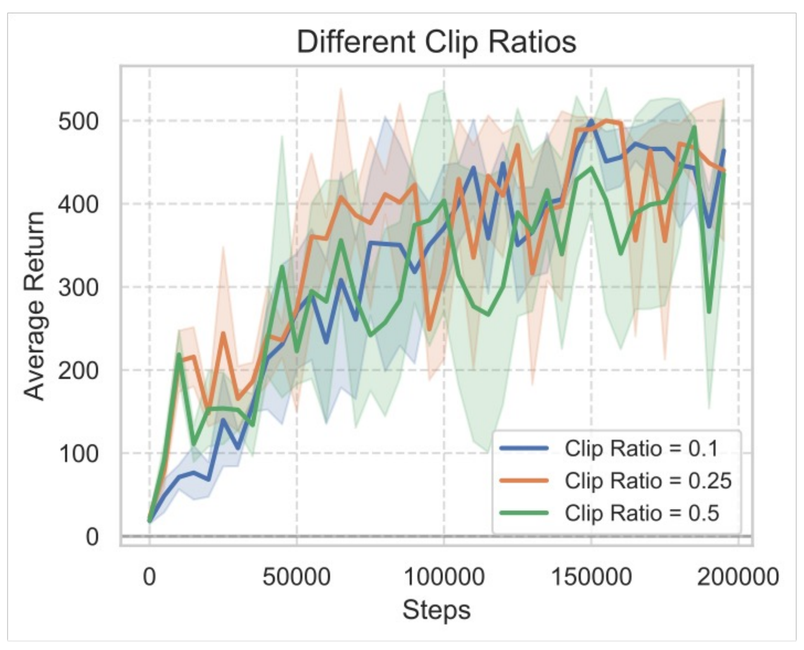
PPO

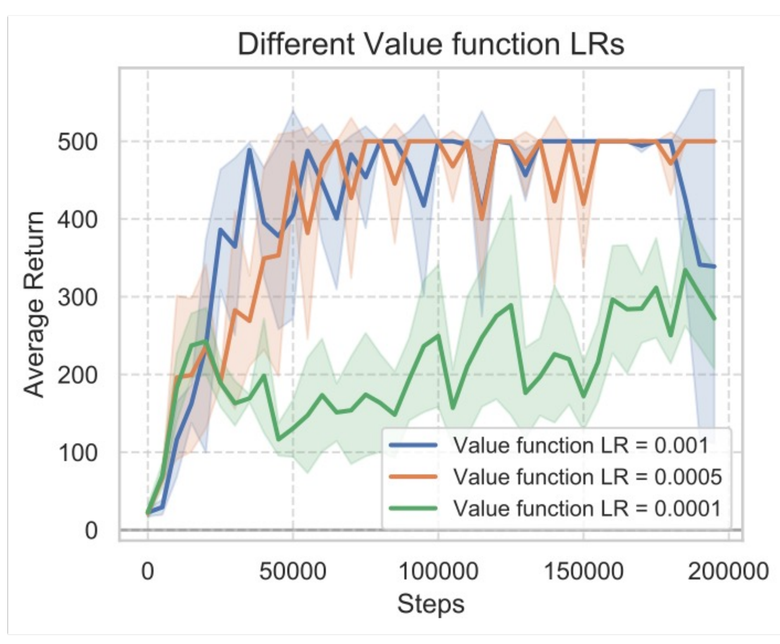
$$L(s, a, \theta_k, \theta) = \min\left(\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)} A^{\pi_{\theta_k}}(s, a),\ \ g(\epsilon, A^{\pi_{\theta_k}}(s, a))\right)$$

CartPole

Keep the pole balanced upright!

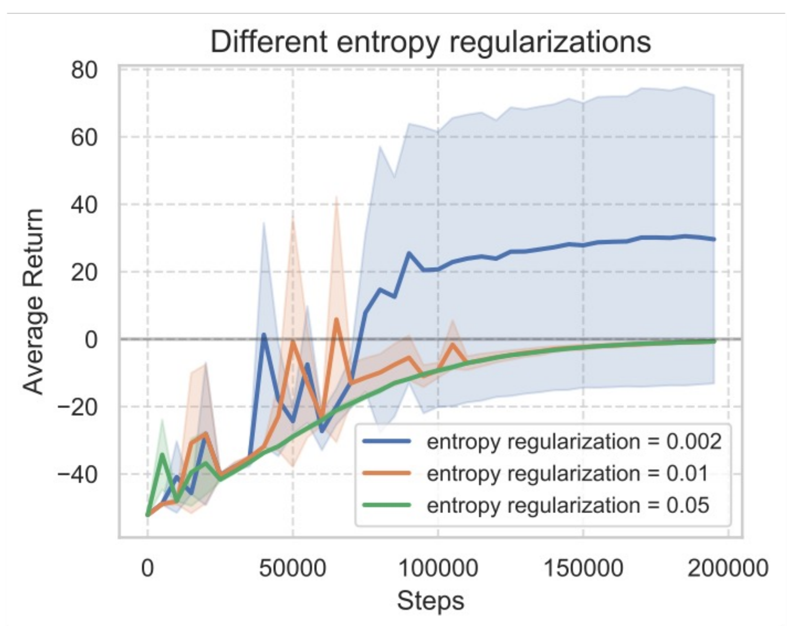
Take away: PPO works seamlessly with proper tuning.

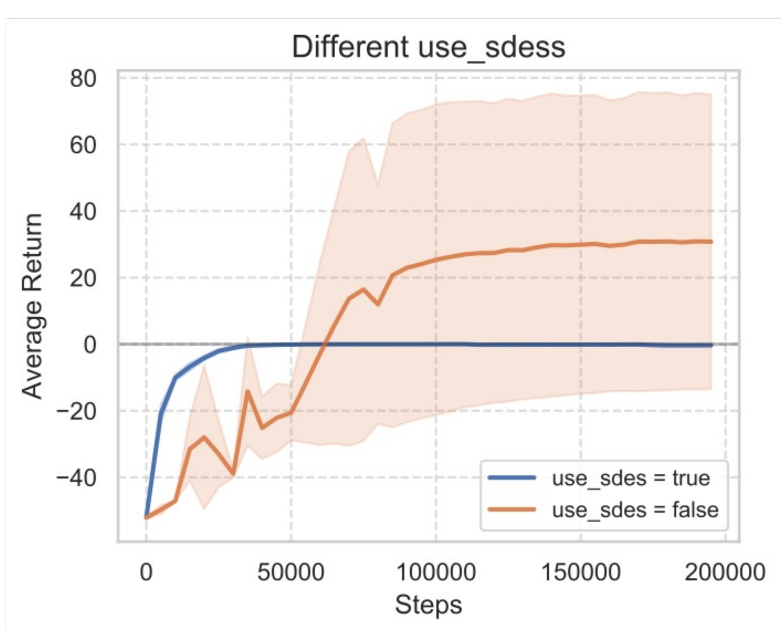




MountainCarContinuous

Drive the car to reach the flag!



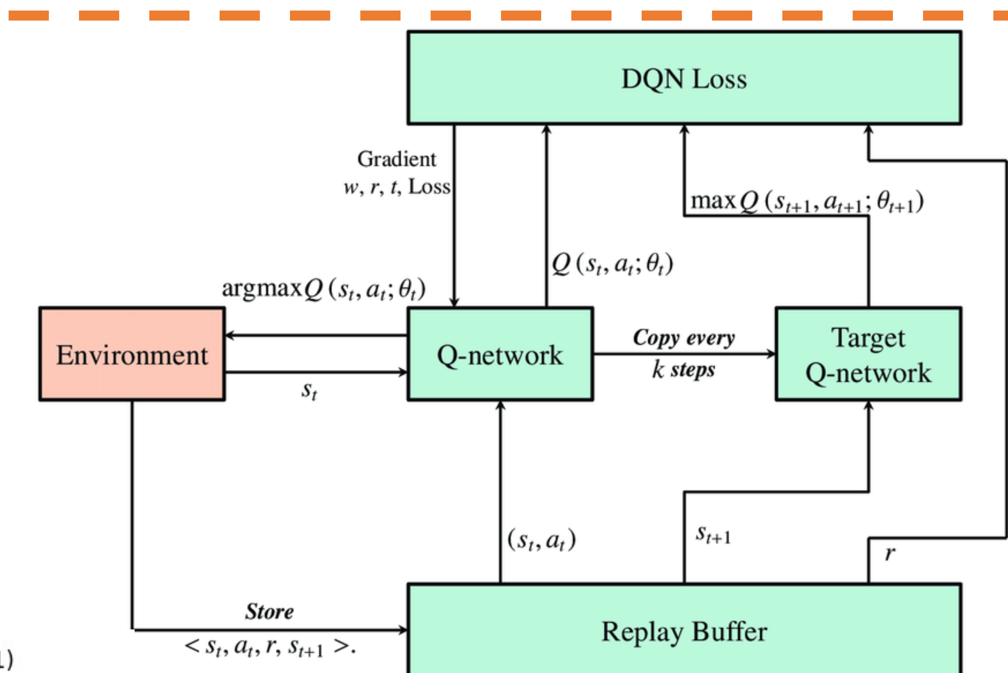


Other tried tricks: Reward shaping/State norm./Reward norm./Orthogonal init, etc.

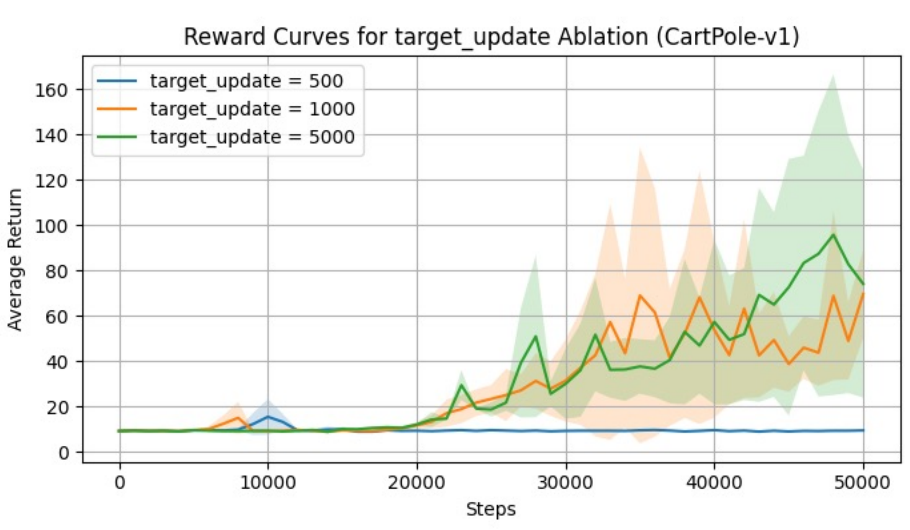
Take away: As an on-policy algorithm, PPO struggles in sparse reward environments, even with common tricks!

DQN

Keywords:
- Off-policy
- Online Learning
- Model-Free
- Value-Based

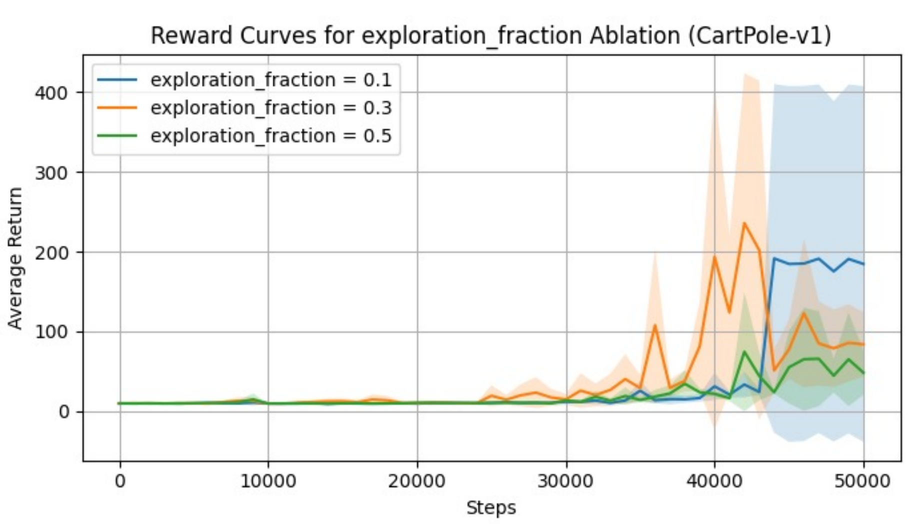


CartPole



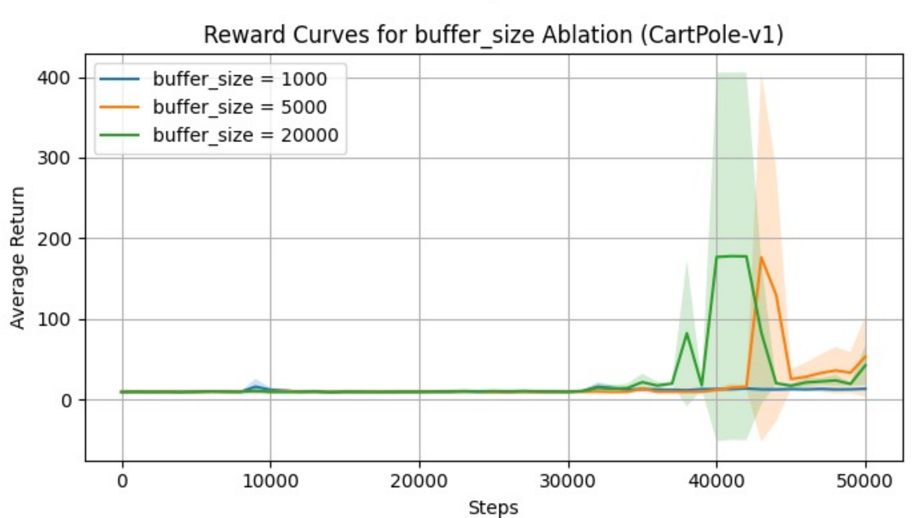
Q1. How does the frequency of syncing the target network affect the stability and performance of DQN training?

Findings: Too frequent updates may destabilize learning



Q1. How does the exploration fraction affect DQN's ability to learn ?

Findings: A proper balance between exploration and exploitation is critical. Too large exploration



Q1. What is the effect of replay buffer capacity on DQN's learning performance and sample efficiency?

Findings: Small buffers may lack diversity; large buffers may contain outdated experiences — both can affect stability and convergence.