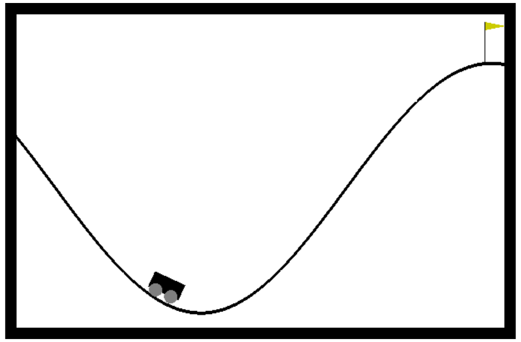
Average-Reward RL via NESM

Jacob Adamczyk[1], Argenis Arriojas[1], Stas Tiomkin[2], Rahul Kulkarni[1]

1: University of Massachusetts Boston --- Physics Dept.

2: San Jose State University --- Computer Engineering Dept.

Reinforcement Learning



MountainCar environment

Basic Idea:

- An agent interacts with the **environment**, by taking **actions**
- Positive behaviors are reinforced relative to undesirable behaviors
  - Reinforcement is implemented via a **reward** function
- The agent learns to maximize rewards received

$\gamma \in (0,1)$ ensures convergence

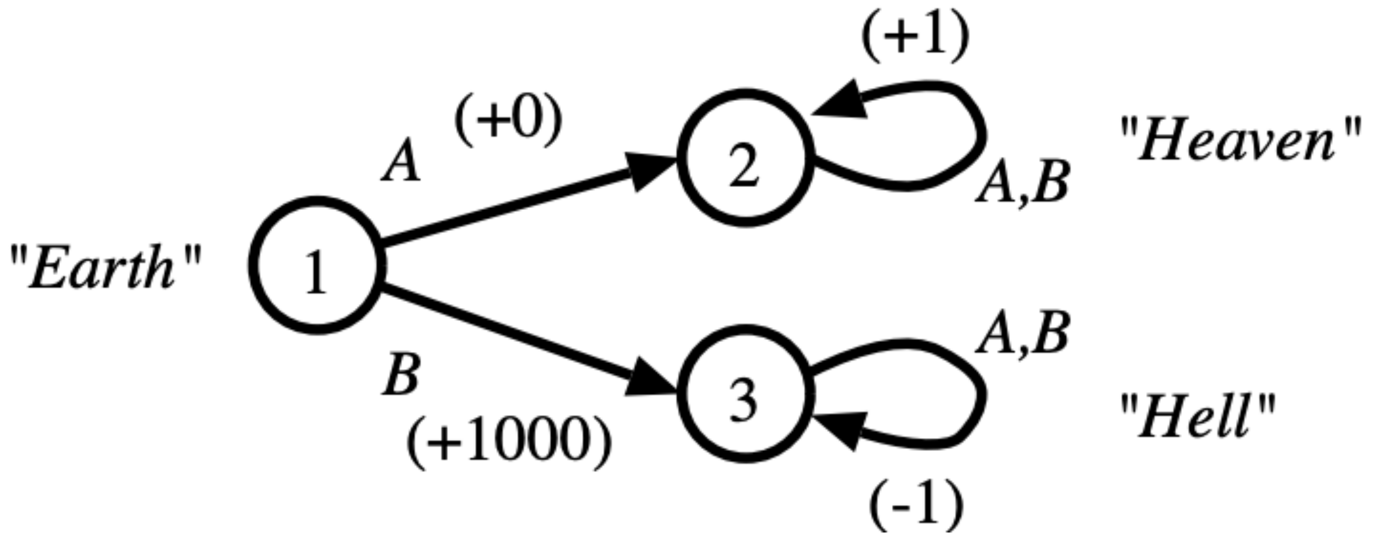
$$Q^*(s,a) = \underset{\pi}{\operatorname{argmax}}\, \mathbb{E}_{\tau\sim p,\pi}\left[\sum_{t=0}^{\infty} \gamma^t \left(r_t + \beta^{-1} \log \pi(a_t|s_t)\right) | s_0 = s, a_0 = a\right]$$

"MaxEnt" regularization

"RL: an intro.", Sutton & Barto MIT Press

Motivating Example

$$R(Heaven) = 0 + \gamma * 1 + \gamma^2 * 1 + \ \dots = \frac{\gamma}{1-\gamma}$$

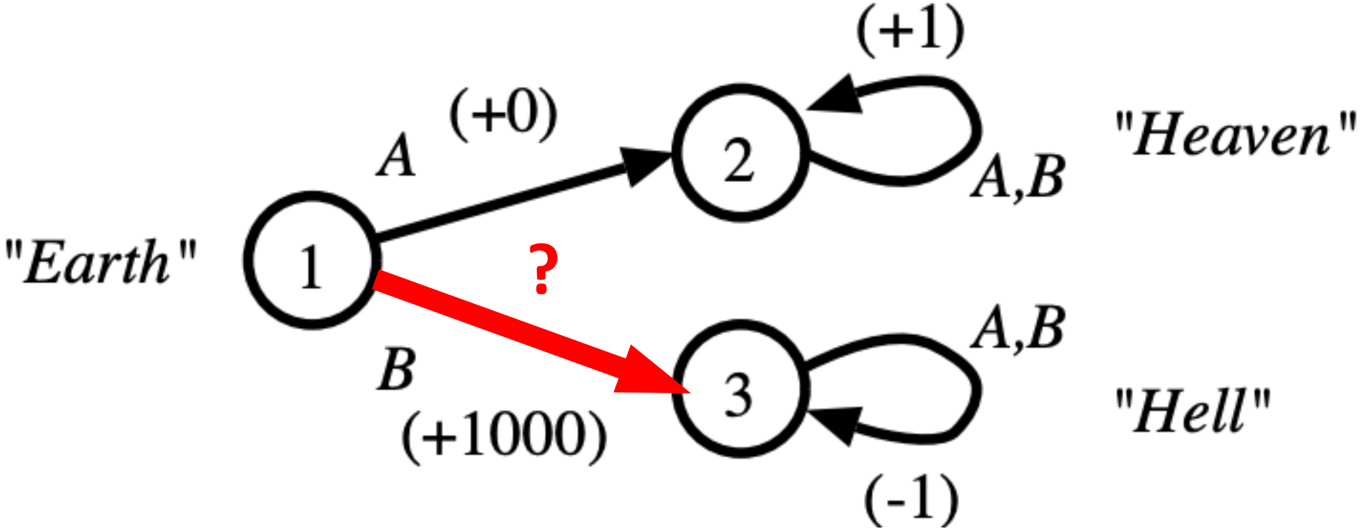


$$R(Hell) = 1000 + \gamma * (-1) + \gamma^2 * (-1) + \ \dots = 1000 - \frac{\gamma}{1-\gamma}$$

"A Reinforcement Learning Method for Maximizing Undiscounted Rewards", A. Schwartz, 1993
"Average reward reinforcement learning", S Mahadevan, 1996

Motivating Example

$\gamma < 0.998$



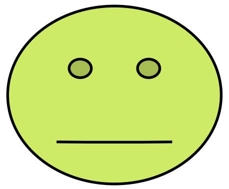
$R(Heaven) < 499$

$R(Hell) > 501$

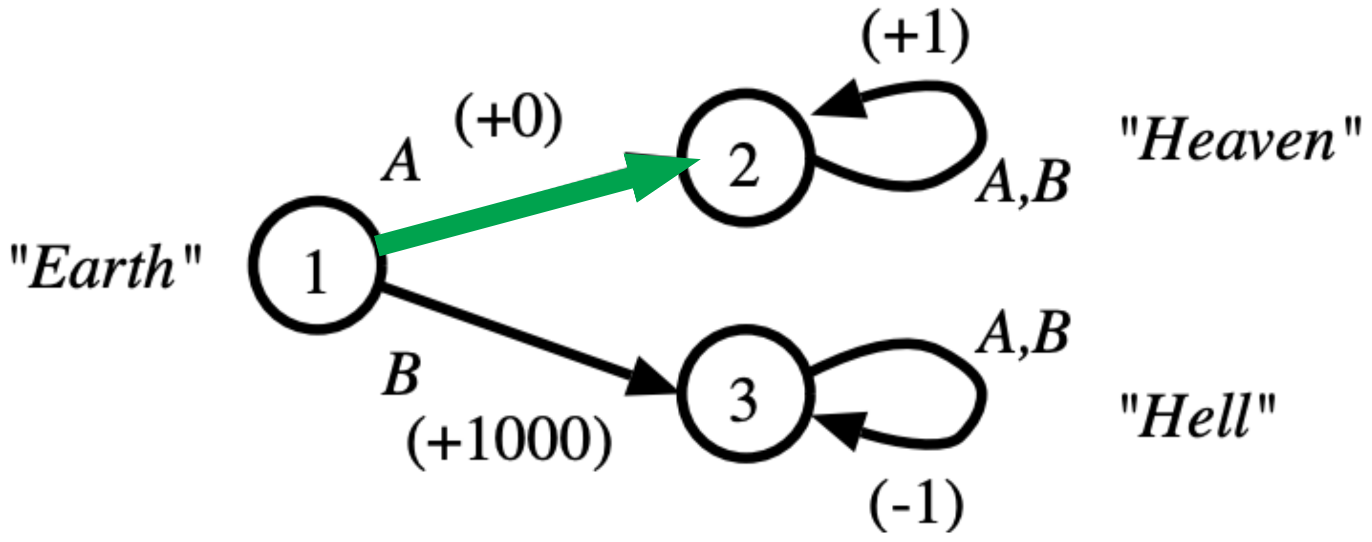
"A Reinforcement Learning Method for Maximizing Undiscounted Rewards", A. Schwartz, 1993
"Average reward reinforcement learning", S Mahadevan, 1996

Motivating Example

“Correct” behavior depends on choice of hyperparameter



$\gamma > 0.998$



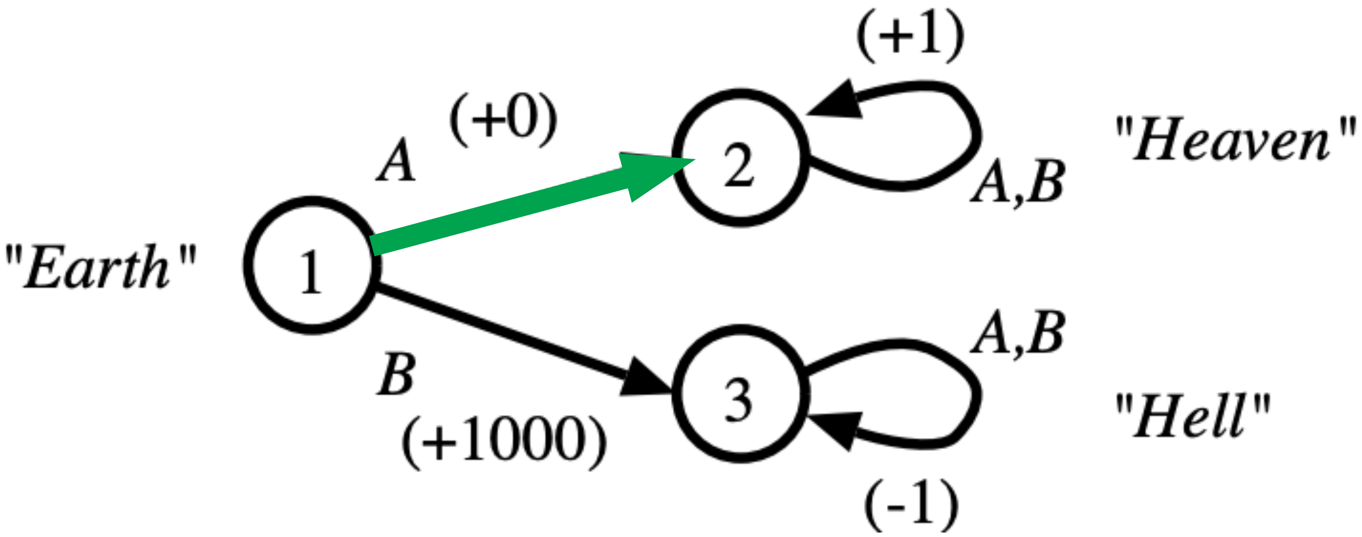
$R(Heaven) > 501$

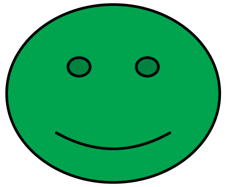
$R(Hell) < 499$

“A Reinforcement Learning Method for Maximizing Undiscounted Rewards”, A. Schwartz, 1993
“Average reward reinforcement learning”, S Mahadevan, 1996

Motivating Example

Average Reward





$R(Heaven) = +1$

$R(Hell) = -1$

"A Reinforcement Learning Method for Maximizing Undiscounted Rewards", A. Schwartz, 1993
"Average reward reinforcement learning", S Mahadevan, 1996

Average-Reward Formulation

Instead of introducing a hyperparameter γ, we can optimize the average reward (time homogeneous):

$$\theta = \max_{\pi} \lim_{T\to\infty} \frac{1}{T}\ \mathbb{E}_{\tau\sim p,\pi}\left[\sum_{t=0}^{\infty} r_t + \beta^{-1}\log\pi(a_t|s_t)\right]$$

$$Q^*(s,a) = \max_{\pi} \lim_{T\to\infty}\ \mathbb{E}_{\tau\sim p,\pi}\left[\sum_{t=0}^{\infty} r_t + \beta^{-1}\log\pi(a_t|s_t) - \theta \mid s, a\right]$$

Solution Method

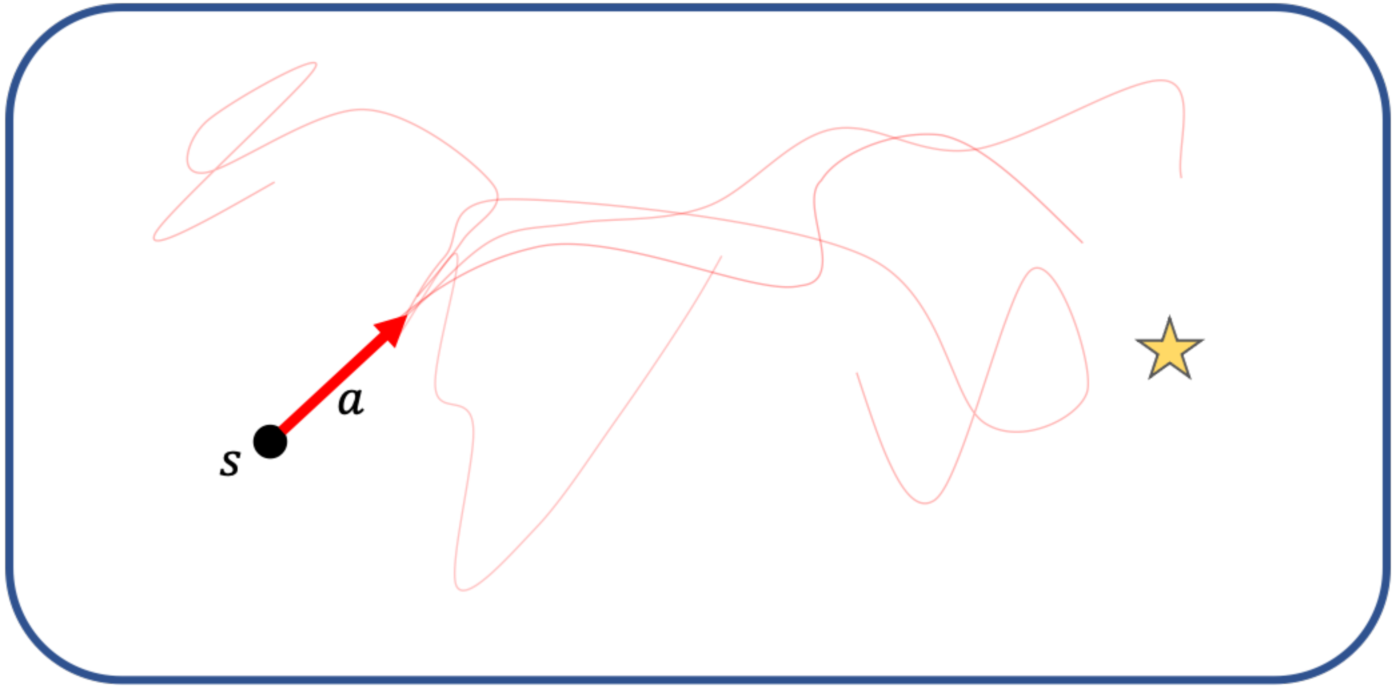
- At a high level, we want to bias the agent toward trajectories with **high** reward
  - Despite prior policy / dynamics typically leading to **low** reward
- To study the dynamics of such rare events we use large deviations theory
- LDT tells us (similar to eq. stat-mech) to include a Boltzmann factor

$$\sum \mathbf{1}[\epsilon_i = E] \to \sum e^{-\beta \epsilon_i}$$

Introduce conjugate var. to control $\langle E \rangle$

Large Deviation Theory

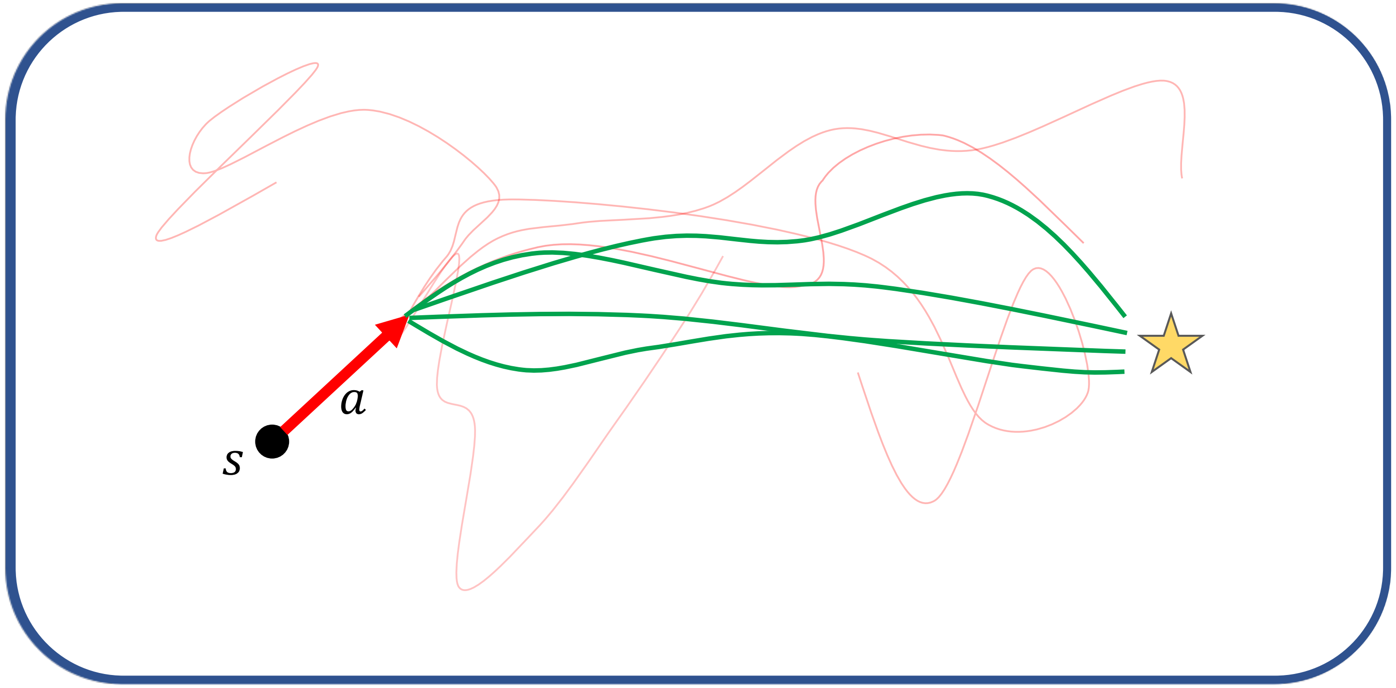
Normally, the (prior) dynamics will evolve the agent to low-reward states.

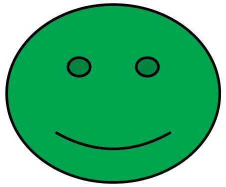


“Entropy regularized RL using LDT”, A. Arriojas, **JA**, S. Tiomkin, and R. V. Kulkarni, PRR 2023

Large Deviation Theory

To counteract this, we can steer the agent by ***tilting*** the dynamics:





The generator of this dynamics is given by: $\widetilde{P}_{(s',a'),(s,a)} = p(s'|s,a)\pi_0(a'|s')e^{\beta r(s,a)}$

“Entropy regularized RL using LDT”, A. Arriojas, **JA**, S. Tiomkin, and R. V. Kulkarni, PRR 2023

Solution Technique

- In the long-time limit, the dynamics of $\tilde{P}$ is generated[1] by a control policy $\pi^*(a|s) \propto u(s,a)$, the left eigenvector of $\tilde{P}$.

- The value function is given by: $Q(s,a) = \beta^{-1} \log u(s,a)$

- For general MDPs, the eigenvector equation is intractable, so we resort to *learning* the left eigenvector, u
  - *Without* the need to fully know/learn the dynamics $\tilde{P}$ ("model-free")

[1]For deterministic dynamics

Solution Technique

- As in DQN, we parameterize the left eigenvector of $\tilde{P}$ with a neural network
- We rewrite the e.v. equation in temporal-difference form:

$$\hat{u}_{\bar{\psi}}(s,a) = e^{\beta(r(s,a)-\theta)} \mathop{\mathbb{E}}_{s'\sim p, a'\sim \pi_0} u_{\bar{\psi}}(s',a')$$

$$e^{\beta\theta} = \frac{1}{|\mathcal{B}|} \sum_{\{s,a,r,s'\}\in\mathcal{B}} \frac{e^{\beta r}\, \mathbb{E}_{a'\sim\pi_0}\, u_{\psi}(s',a')}{u_{\psi}(s,a)}$$

$$\mathcal{J}(\psi) = \frac{1}{2} \mathop{\mathbb{E}}_{s,a\sim\mathcal{D}} \left(u_{\psi}(s,a) - \hat{u}_{\bar{\psi}}(s,a)\right)^2$$

PPI – Unregularized / Standard RL

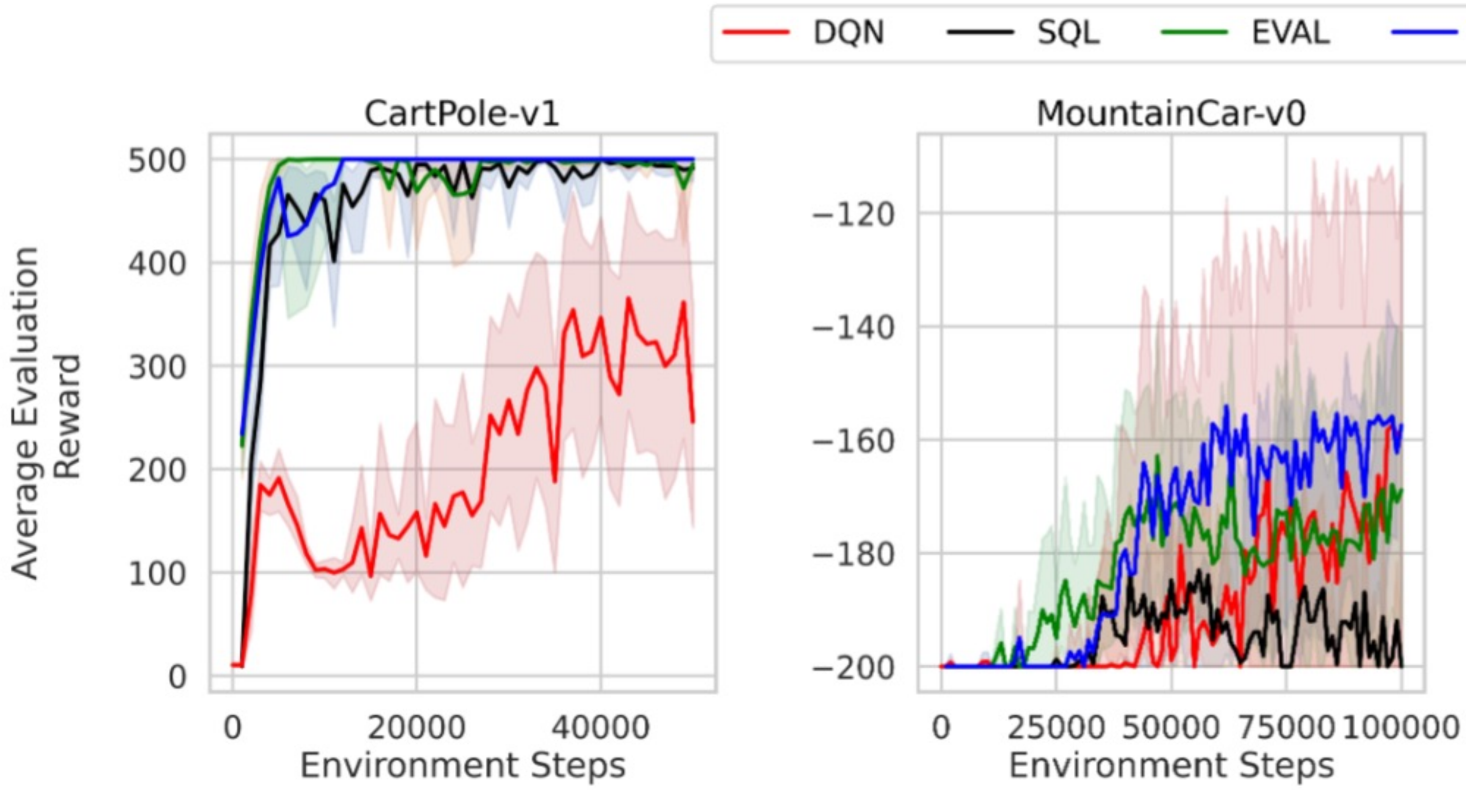
To solve the RL problem *without* entropy regularization, we use a method of Rawlik et. al.[2]:

Algorithm 2 Posterior Policy Iteration (PPI)

Initialize: Prior policy π_0, $\beta > 0$, solve budget.
while $N <$ solve budget **do**
  $\pi_0 \leftarrow \text{Solve}(\pi_0, \beta)$
end while
Output: Deterministic optimal policy $\pi^*_{\beta=\infty} = \pi_0$

[2]: "On stochastic optimal control and reinforcement learning by approximate inference", IJCAI 2013 Rawlik, Toussaint, Vijayakumar

Results



“Off-Policy Average-Reward RL with Entropy Regularization” **JA**, V. Makarenko, S. Tiomkin, R. V. Kulkarni (under review)

Conclusions

- No discounting needed (can solve physically-relevant problems)
- PPI implemented (can solve with/wo entropy regularization)
- Comparable or outperforms SOTA in sample complexity

Future Work:

- Continuous action spaces
- Exploit eigen-structure
- Continue to explore the new avenues of deep RL research enabled by this work

Thank you!

Argenis Arriojas

Rahul Kulkarni

Stas Tiomkin

Volodymyr
Makarenko

UMass Boston

SJSU SAN JOSÉ STATE UNIVERSITY

NSF

ORACLE