Randomized Exploration for Reinforcement Learning with General Value Function Approximation

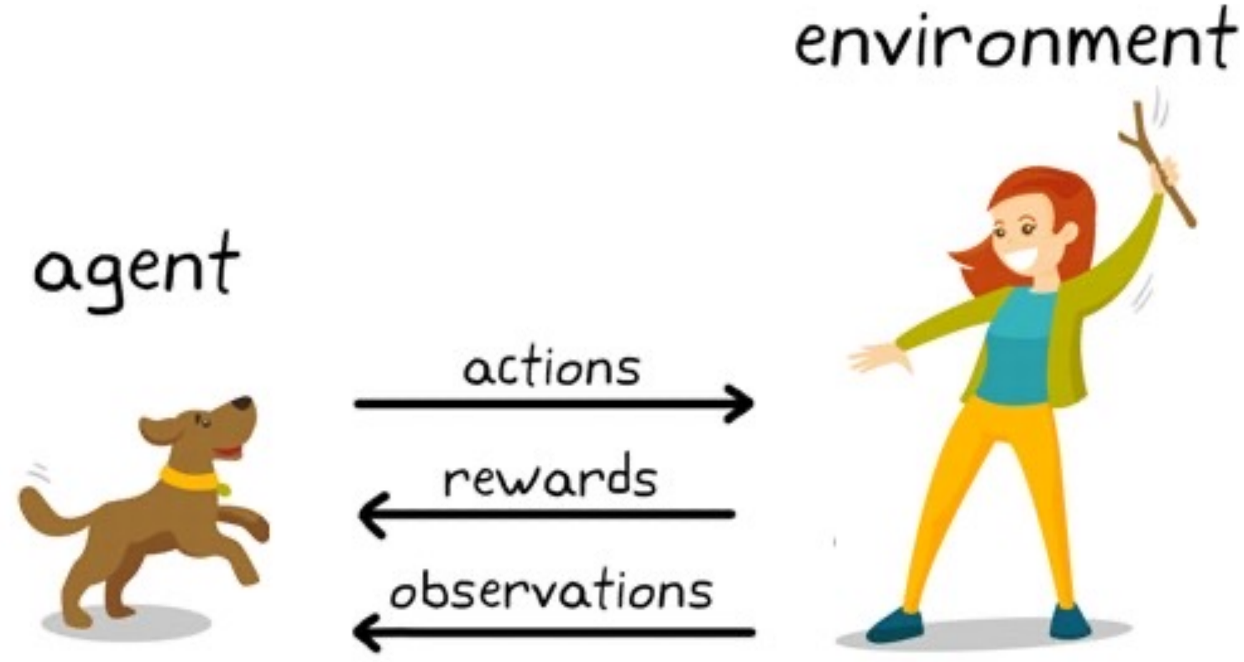
Haque Ishfaq

McGill

Mila

Reinforcement Learning

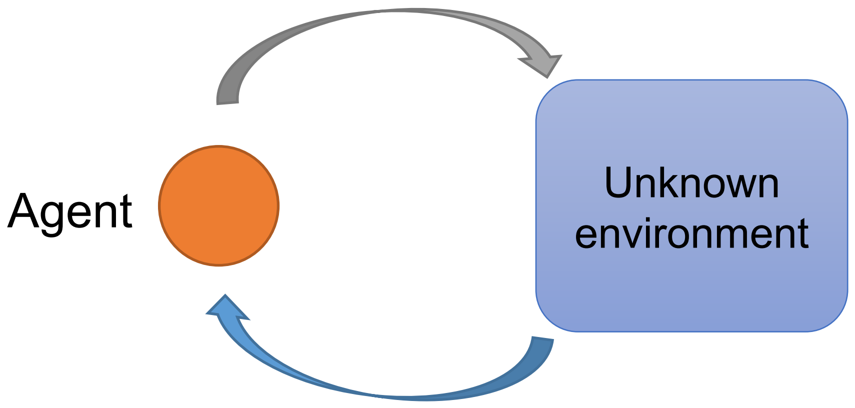
Learn to interact with an unknown environment through trial and error



Reinforcement Learning

Learn to interact with an unknown environment through trial and error

Action: $a_1, a_2, \ldots, a_H$



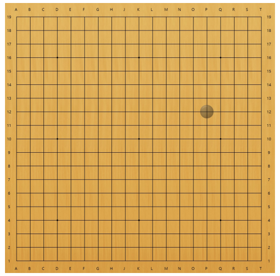
State observations: $s_1, s_2, \ldots, s_H$
Rewards: $r_1, r_2, \ldots, r_H$

Goal: maximize cumulative reward for a horizon H

Value: $\mathrm{E}[r_1 + r_2 + r_3 + \cdots r_H]$

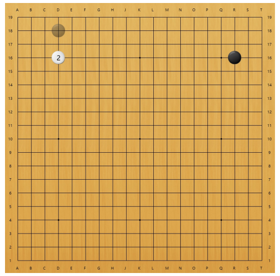
Long term effect needs to be considered.

Reinforcement Learning



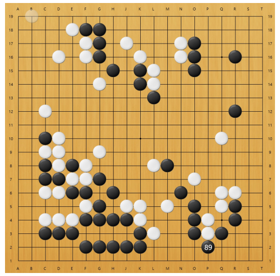
s_0

$a_0 \leftarrow$ move 1
$r_0 \leftarrow 0$



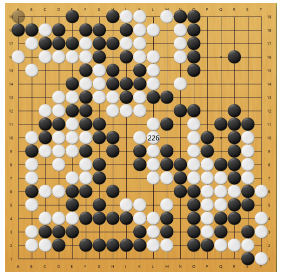
s_1

$a_1, a_2, \ldots$
$r_1, r_2 \ldots \leftarrow 0$



s_{89}

$a_{90}, a_{91}, \ldots$
$r_{90}, r_{91} \ldots \leftarrow 0$



s_{226}

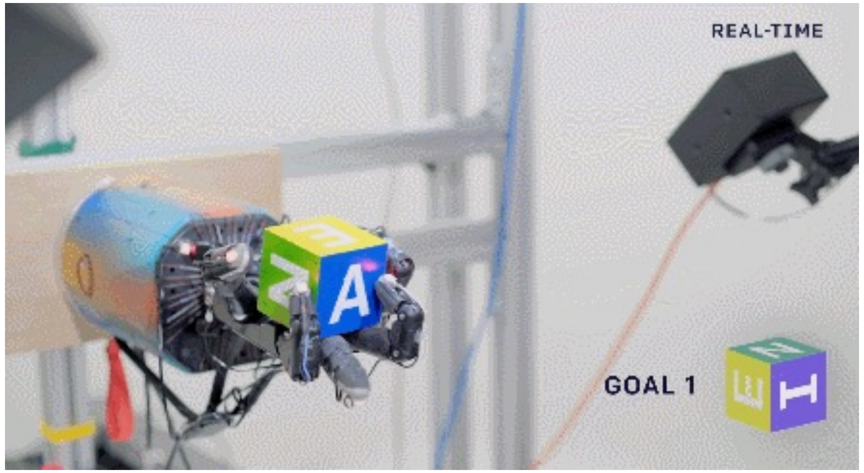
Wining: $r_{226} \leftarrow 1$

Losing: $r_{226} \leftarrow 0$

Actions?

States?

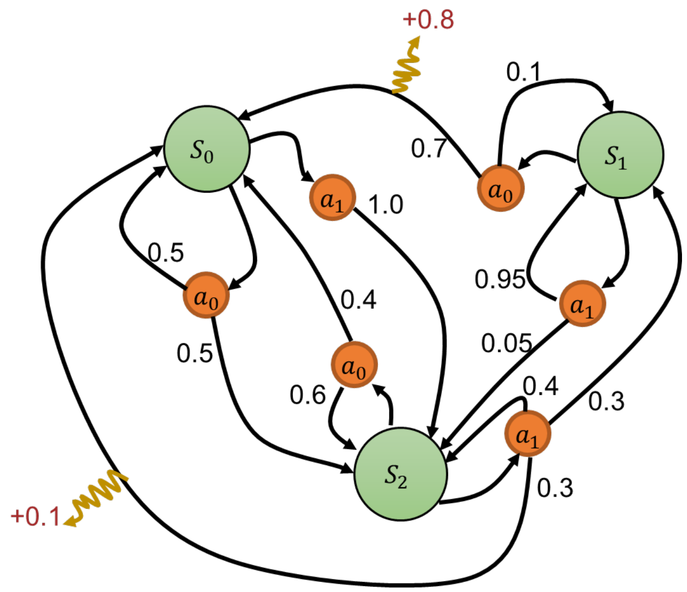
Rewards?



OpenAI Arm

Markov Decision Process (MDP)

- Environment is unknown
  - States: S; actions: A
  - Reward: $r(s, a) \in [0, 1]$
  - **Unknown** state transition: $P_h(\cdot \mid s, a)$
  - Horizon: H (a large number)
  - Goal: optimal policy $\pi^*: S \to \Delta_A$

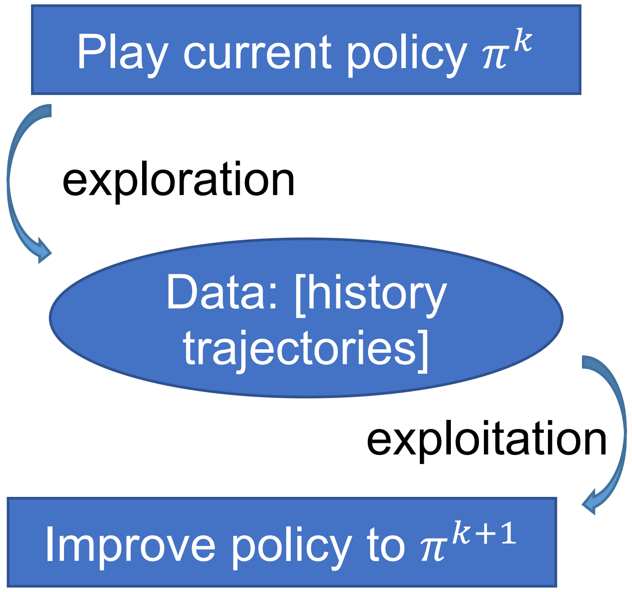


$$\max_{\pi} \mathrm{E}\left[r_1(s_1, \pi(s_1)) + r_2(s_2, \pi(s_2)) + \cdots r_H(s_H, \pi(s_2))\right] =: Q^\pi$$

$$s_i \sim P(\cdot \mid s_{i-1}, \pi(s_{i-1}))$$

Theories of RL on MDP

- Exploration + exploitation [Kearns & Singh 2002, Jaksch et al. 2010]
  - Learn from scratch
  - Exploitation: optimize policy based on existing data
  - Exploration: collect new info about the environment
    - *Regret: average error v.s. optimal policy*
- Focus has been on Tabular RL
  - Does not scale in practical problem
  - Provides sanity check for exploration algorithm
  - In deep RL, the default is ϵ-greedy exploration



Does tabular algorithm work in practice?

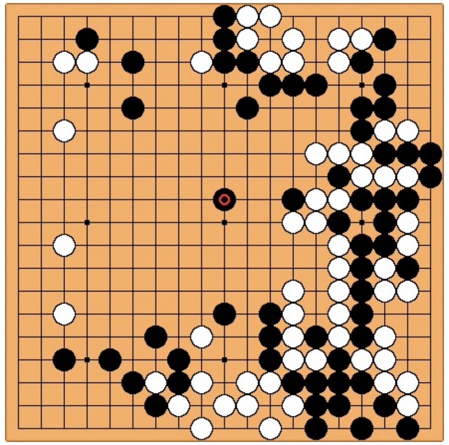
- Number of episodes required to get a good π

$$\widetilde{\Theta}[|S||A|\mathrm{poly}(H)]$$

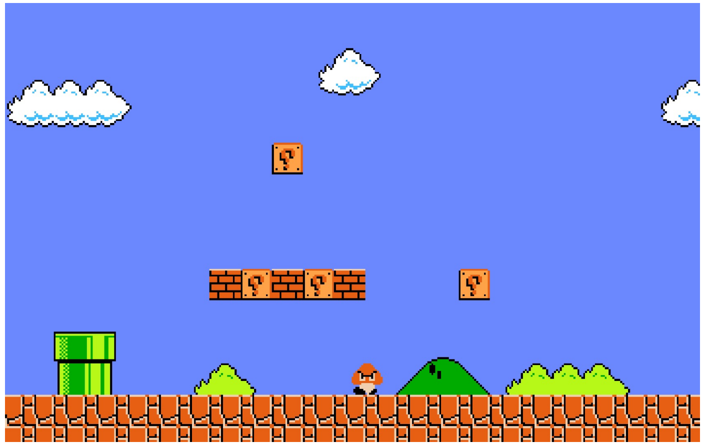
[Jin et al'2018] [Azar et al' 2017][…]

- Curse of Dimensionality

S



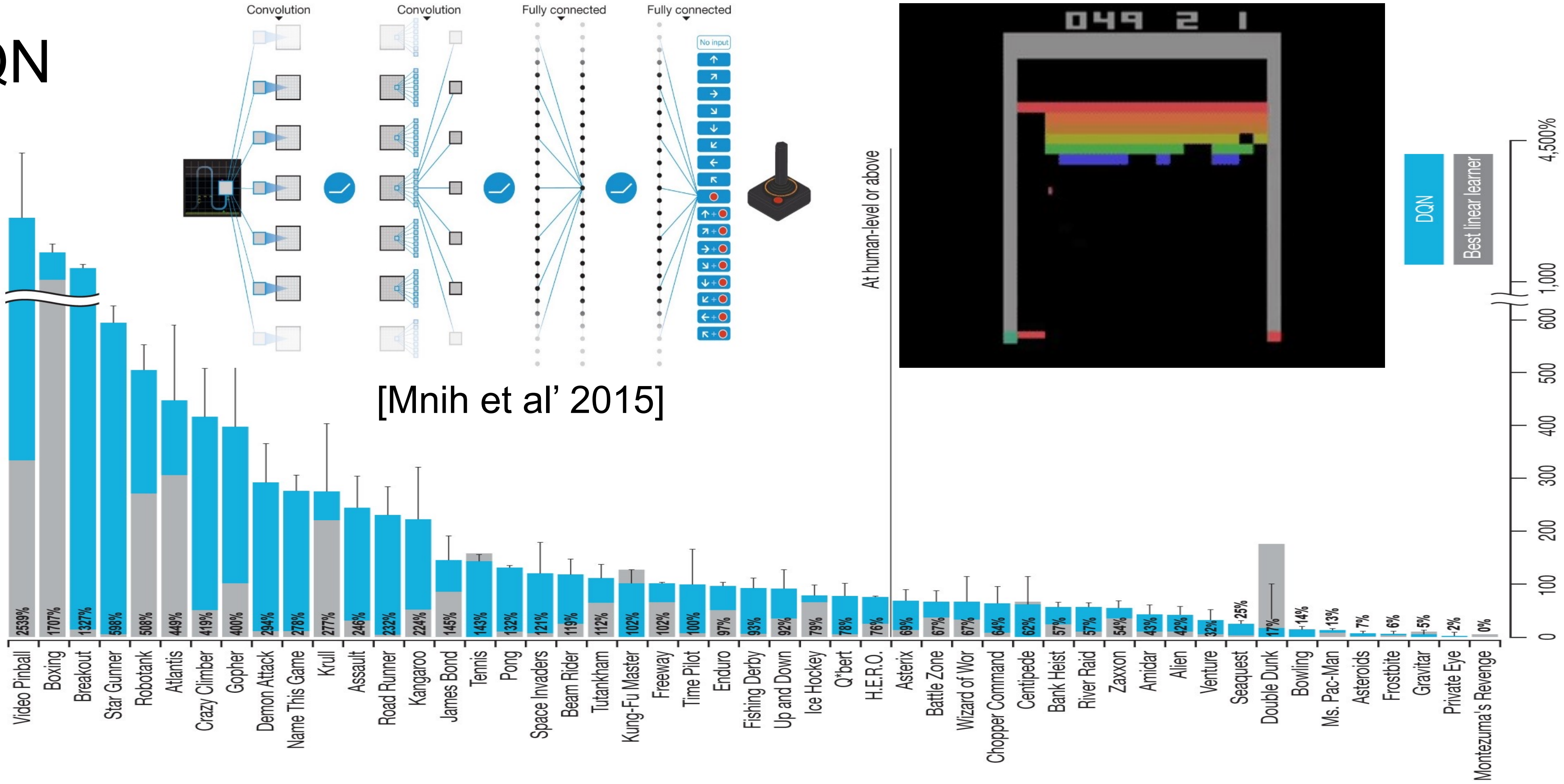
$|S| = 3^{361}$



$|S| \geq 256^{256\times 240}$

Function Approximation in Practice

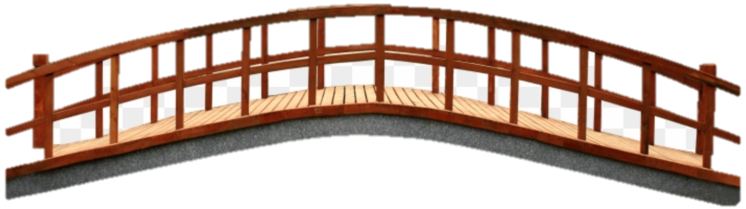
- DQN



[Mnih et al' 2015]

Limitations? **Huge** number of training samples. Hard to **understand**. No **theoretical** guarantee.

RL Theory v.s. Practice



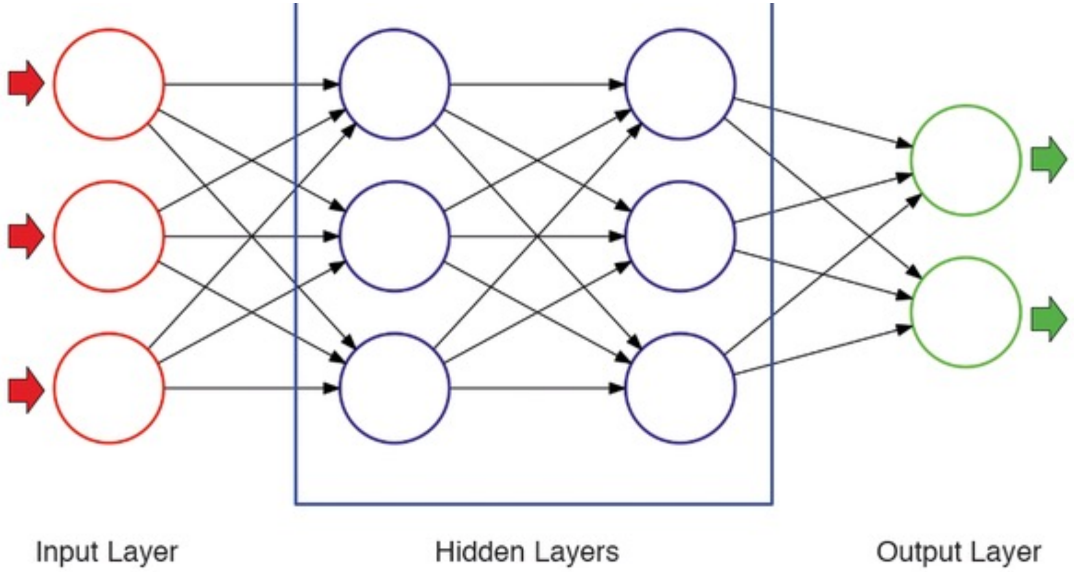
- Theory
  - Markov decision process
    - Finite state space S
    - Finite action space A
    - Finite horizon H
  - Many theoretical results
    - Mostly tabular – well understood
    - Not scalable

- Practice
  - Infinite state space
  - Function approximation via Deep Neural Networks
  - Many empirical results
    - Little understanding
    - No guarantee

Function Approximation

- Find a function class to approximate $Q^*(s,a)$ or π^*

(s,a)



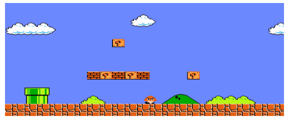
Q*(s,a)

π*(s,a)

- Generalization ability
  - Infer values/policies for unseen (s, a)

Linear Function Approximation

- Need correct features
  - Features are given: $\phi(s, a) \rightarrow R^d$

ϕ ( , (Action Left)) = (3 question marks, 1 enemies, 4 bushes, 1 chimney, …)

  - Bad features requires **exponential** time/sample to learn

    [Du-Kakade-Wang-Yang' 20] [Van Roy & Dong' 20] [Lattimore et al' 20] [Weisz et al' 20]

- Good features
  - Linear MDP [Yang & Wang' 19]:

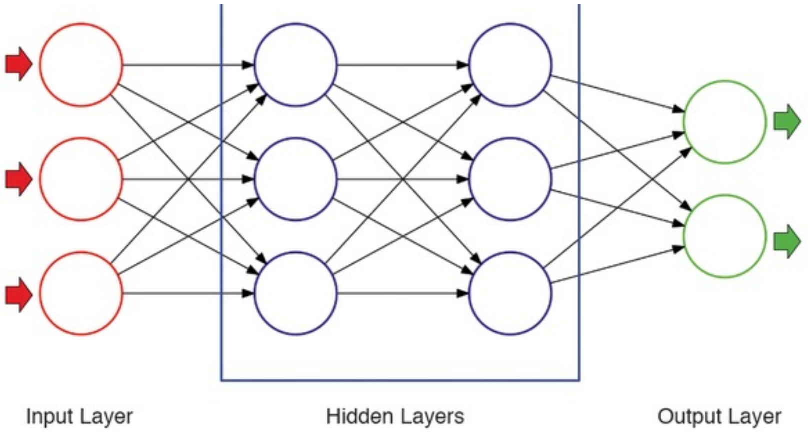
    efficient algorithm: [Jin et al' 20]
  - Low-bellman error [Zanette et al' 20]
  - Low-bellman rank [Jiang et al' 17]

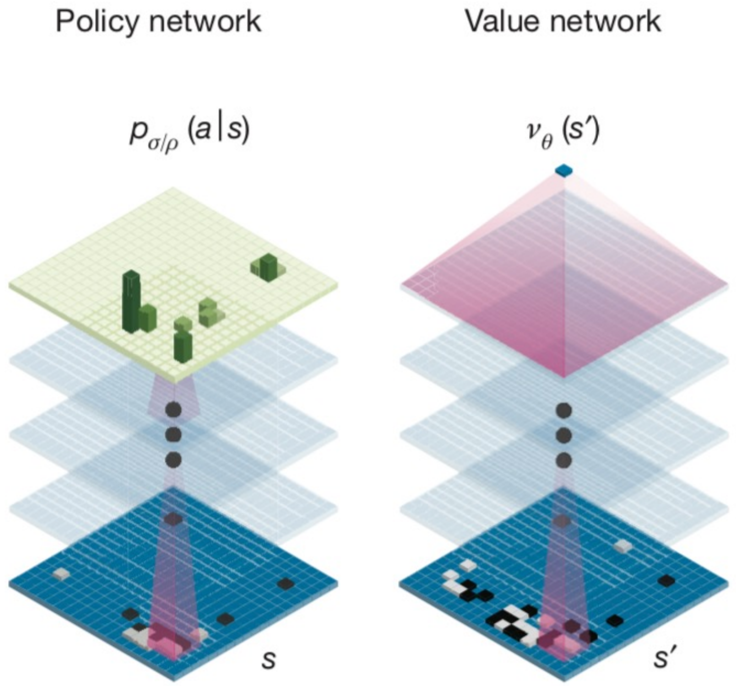
$$P(s'|s, a) = \sum_{k\in[K]} \phi_k(s, a)^\top \psi_k(s')$$

Time efficient

General function approximation

- No features are given
- Function class $\mathcal{F}$
  - Might be parametric
  - $f(s, a)$ may rep. $Q^*(s, a)$
- Used in practice





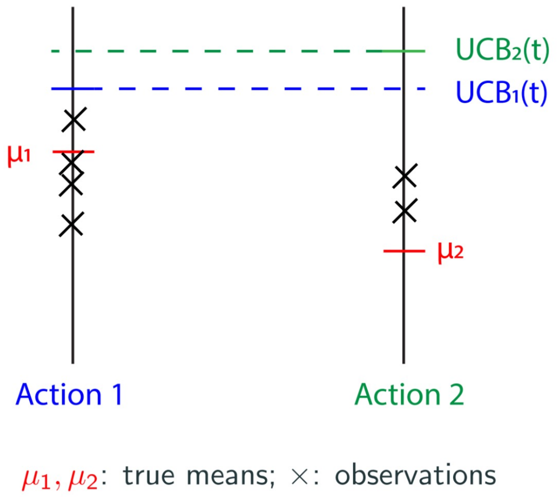
[Silver et al 2016]

Goals for RL:

- Efficient algorithms with practical potentials
- Theoretical guarantees for special cases

Strategies for Exploration

- Optimism in the face of uncertainty:
  - Upper Confidence Bound (UCB)



- Thompson Sampling
  - One of the oldest heuristics for balancing exploration exploitation trade-off. (Thompson, 1933)
  - Randomly select an action according to the probability of it being the optimal action.
  - PSRL = Thompson Sampling for MDPs. (Strens 2000)
  - Sample MDP from posterior, apply policy for an entire episode.

Randomized value functions

- Key idea: generate approximate posterior samples
  - Use standard value learning algorithms (LSVI, DQN, …)
  - Fit to randomly perturbed data

- Theory for tabular representation + LSVI:
  - Worst-case regret bound for Gaussian noise (Russo 2019)

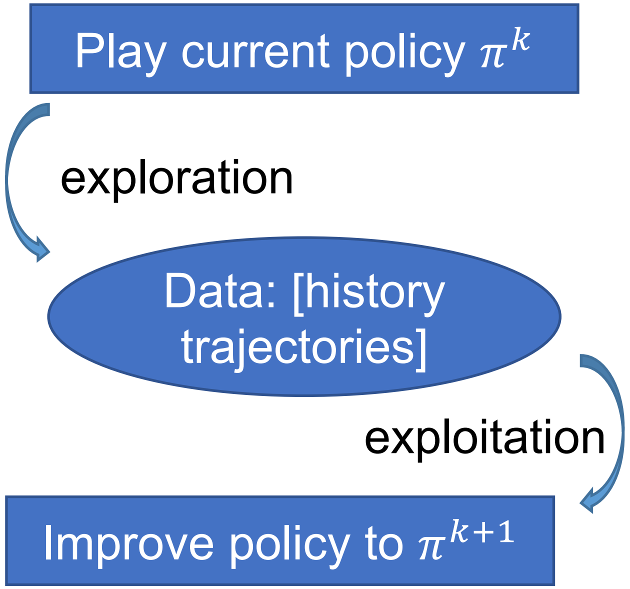
$$Regret(T) \leq \tilde{O}(H\sqrt{S^3AT})$$

- . Computational results with generalization
  - Parameterized representation for Q(s,a)
  - Scalable unlike UCB based methods or posterior sampling
  - Approximate posterior inference is good enough for efficient exploration.

Current limitations

- No theoretical result for RVF with general function approximation
  - Limited to empirical results only (Bootstrapped DQN, Ensemble sampling)
- Lack of unification between OFU and Thompson Sampling
  - Can we combine both principle for algorithm design?
- Bypassing UCB bonus in applying OFU principle
  - UCB bonus is not scalable
  - For GFA, requires complicated sensitivity sampling scheme [Wang et al, 2020]

LSVI for Online RL with **General** VFA



- Initialize an arbitrary $Q^0 \leftarrow 0$
  - For episode $k = 1, 2, \ldots K$:
    - Solve for Q_h^k using LSVI on the history

$$\theta_h^k \leftarrow \mathrm{argmin}_w \sum_t \left[f_w(s_t, a_t) - \left(r(s_t, a_t) + \max_a Q_{h+1}^k(s_{t+1}, a) \right) \right]^2$$

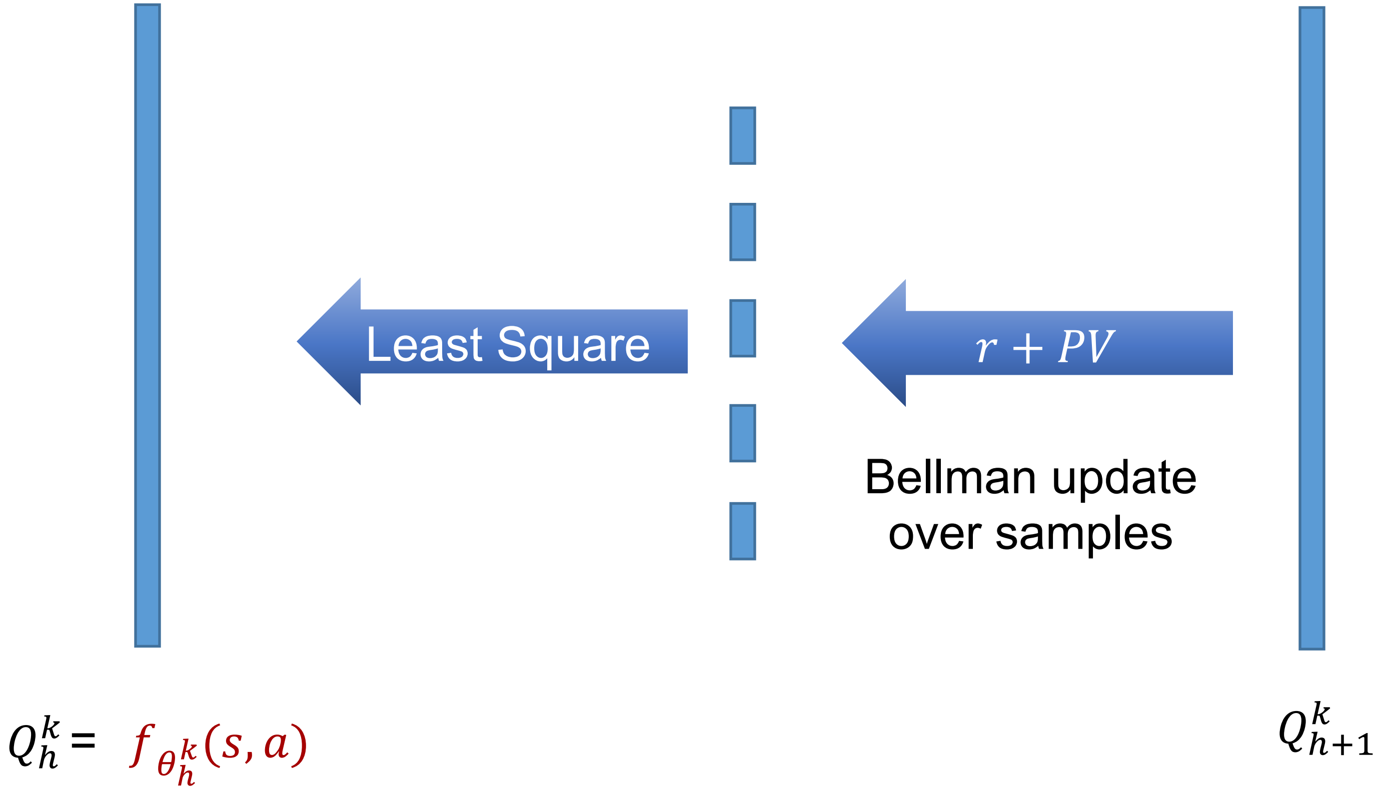
$$Q_h^k(s, a) = f_{\theta_h^k}(s, a)$$

    - Collect a trajectory of data $\boxed{\pi_h^k(s) \leftarrow \mathrm{argmax}_a\, Q_h^k(s, a)}$

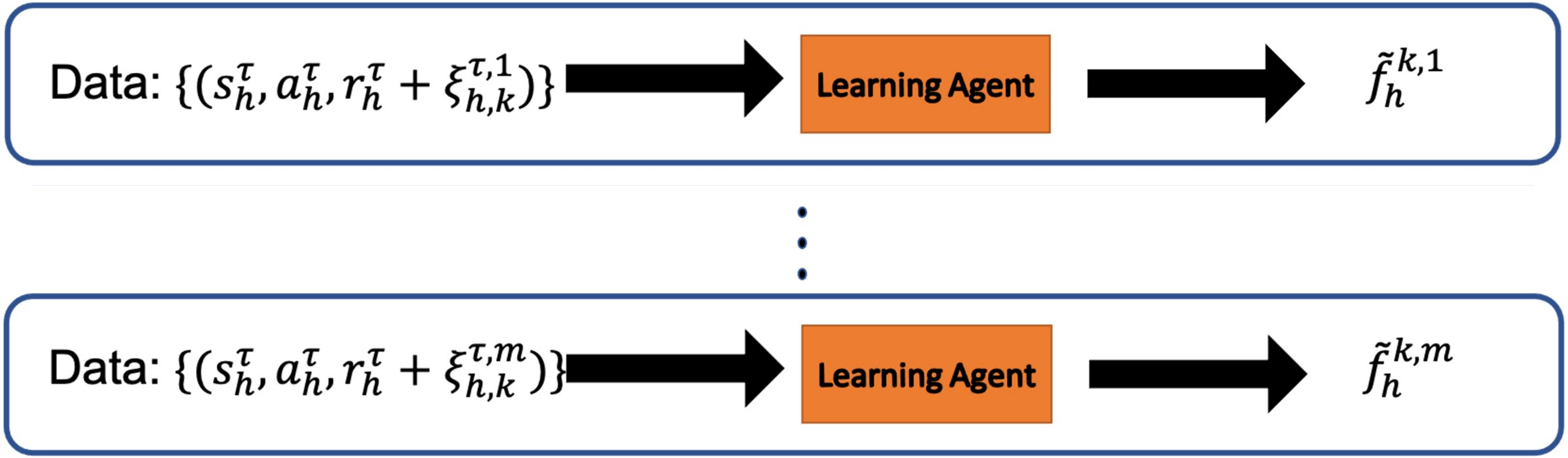
$$(s_1^k, a_1^k, r_1^k) \rightarrow (s_2^k, a_2^k, r_2^k) \rightarrow (s_3^k, a_3^k, r_3^k) \rightarrow \cdots (s_H^k, a_H^k, r_H^k)$$

LSVI as Approximate Dynamic Programming (ADP)

- Each iteration solves



Optimistic Sampling



$$Q_h^{k,m}(\cdot,\cdot) = \tilde{f}_h^{k,m}(\cdot,\cdot)$$

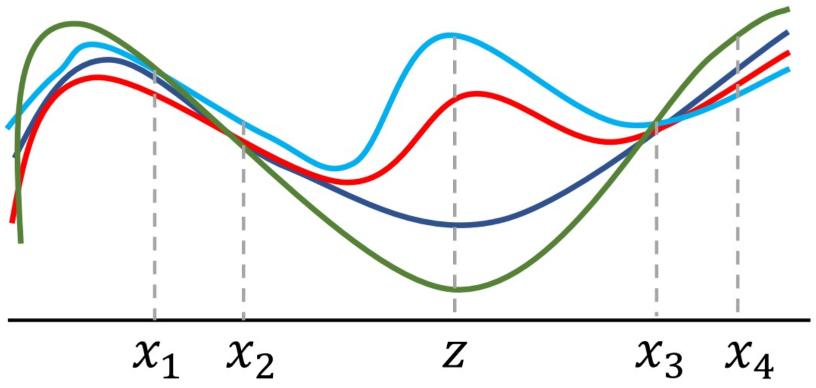
$$Q_h^{k}(\cdot,\cdot) = \min\{\max_{m \in [M]} Q_h^{k,m}(\cdot,\cdot), \mathrm{H} - \mathrm{h} + 1\}$$

Theory for General functions

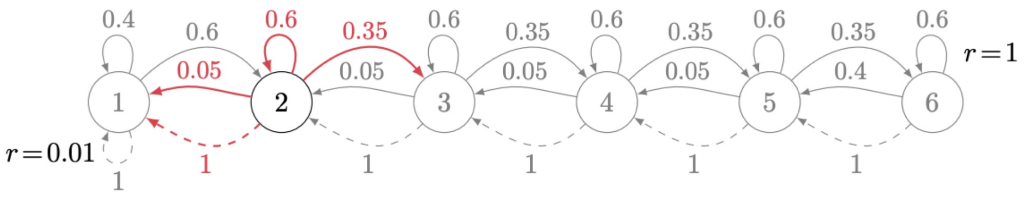
- Assumption: $r + PV \in \mathcal{F}, \forall V$
  - Realizability: The function set is the "image" of Bellman projection
  - Corresponding to linear MDP for linear setting
- Eluder dimension [Russo&Van Roy' 2013]
  - d_E: the longest determination sequence of the function set
  - d-dim linear / generalized linear: $\approx d$

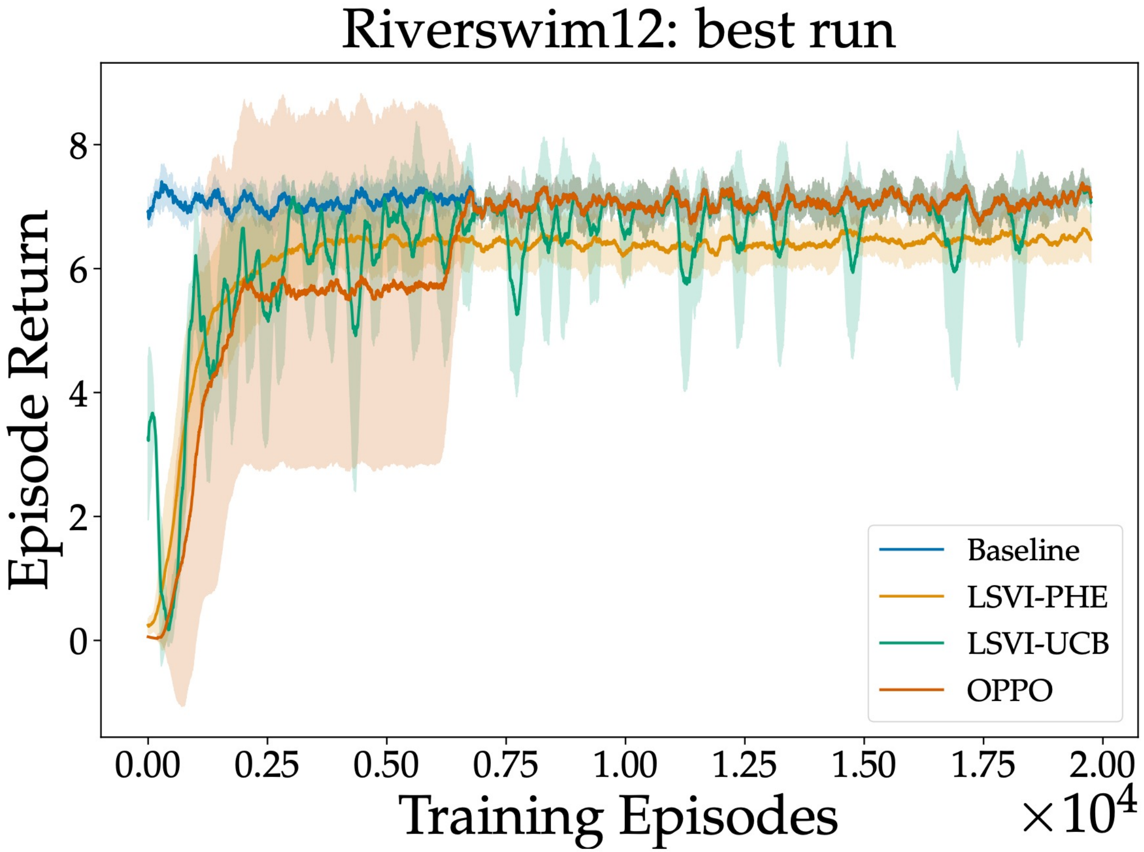
Theorem:

LSVI-PHE with **optimistic sampling** satisfies regret bound of $\boldsymbol{O}\left(\mathrm{poly}(\boldsymbol{d_E H})\sqrt{\boldsymbol{T}}\right)$ with high probability

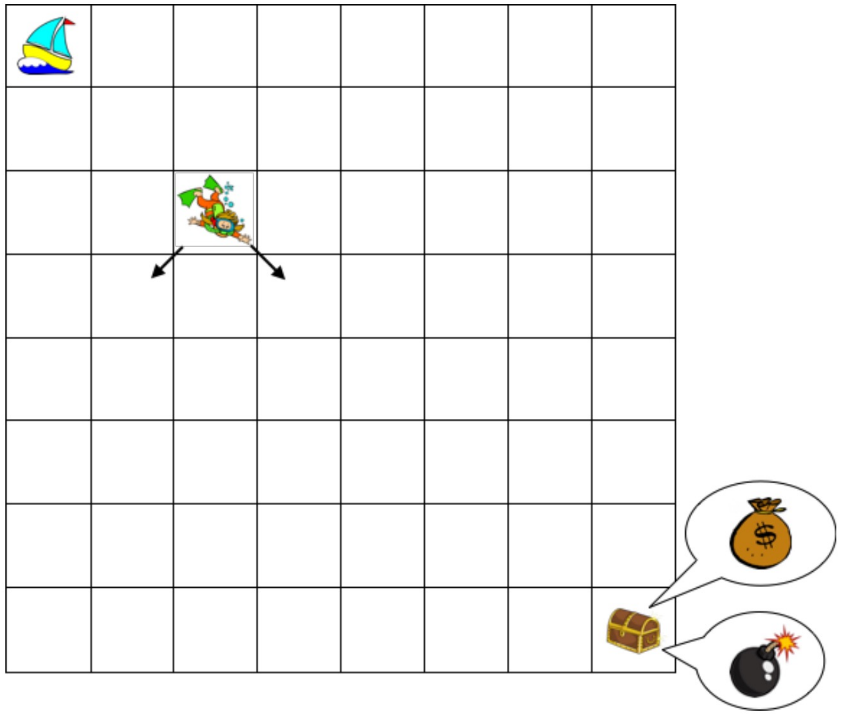


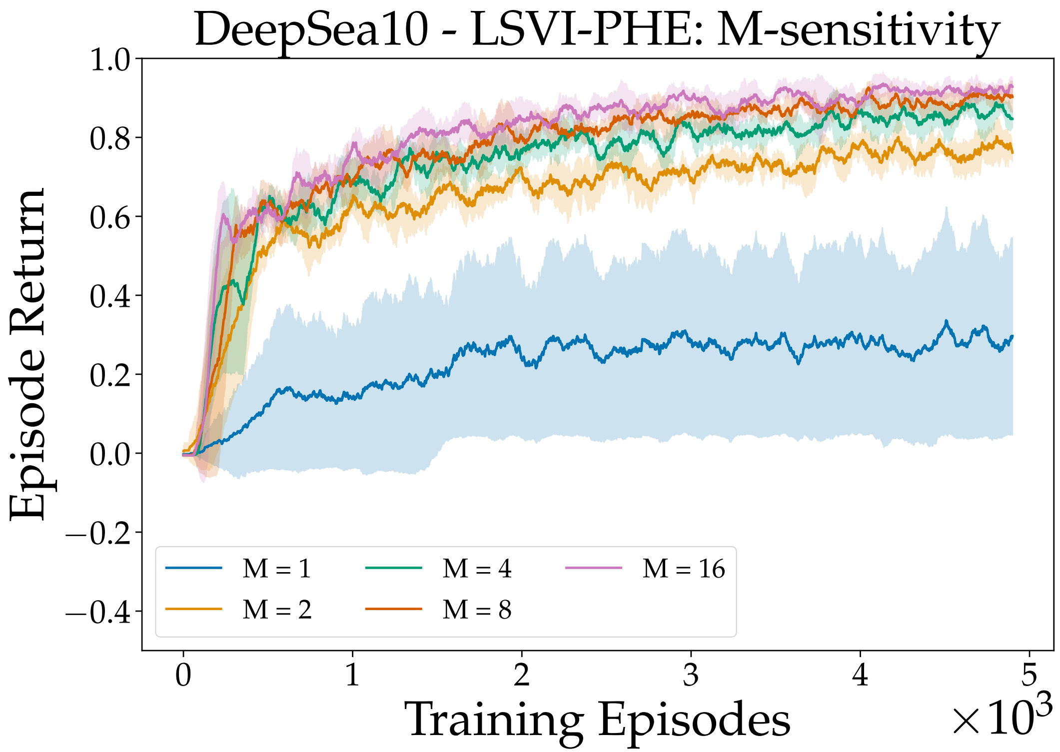
Riverswim:


0.4
0.6
0.6
0.35
0.6
0.35
0.6
0.35
0.6
0.35
0.6
0.05
0.05
0.05
0.05
0.4
$r = 1$
1
2
3
4
5
6
$r = 0.01$
1
1
1
1
1
1

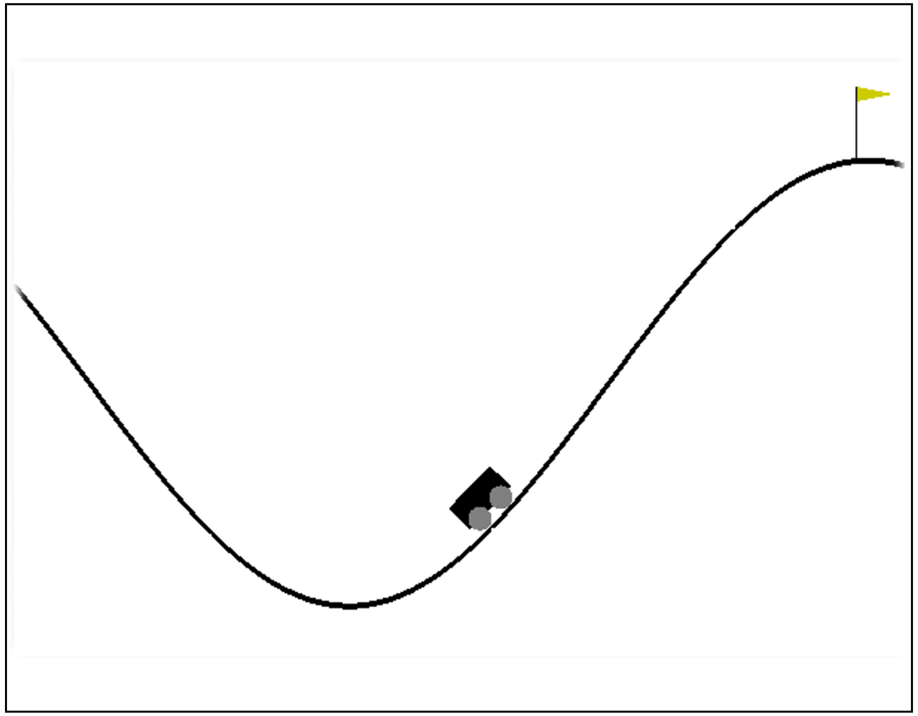

Riverswim12: best run
Episode Return
Training Episodes
$\times 10^4$
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
0
2
4
6
8
Baseline
LSVI-PHE
LSVI-UCB
OPPO

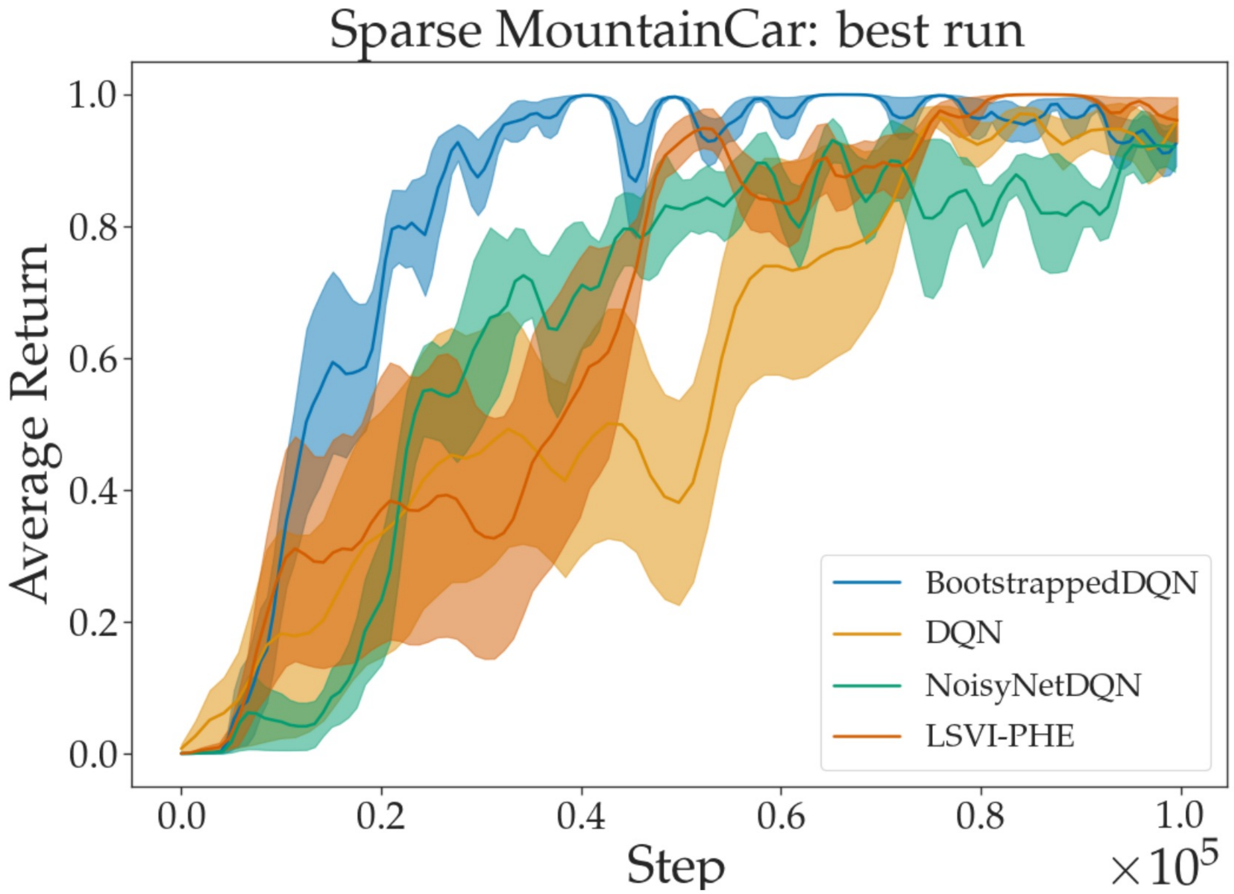
Deep Sea: M sensitivity





Mountain Car:





Summary

- Provably efficient RVF method for RL with general function approximation
  - Sublinear regret
  - Computationally efficient

- *Optimistic sampling allows us the unify OFU and Thompson Sampling*

Collaborators

Qiwen Cui
University of Washington

Viet Nguyen
McGill, Mila

Alex Ayoub
University of Alberta, Amii

Zhuoran Yang
Princeton

Zhaoran Wang
Northwestern University

Doina Precup
McGill, Mila

Lin Yang
UCLA

Paper Link:

[Ishfaq, Cui, Nguyen, Ayoub, Yang, Wang, Precup, Yang' ICML 2021] *Randomized Exploration for Reinforcement Learning with General Value Function Approximation*
https://arxiv.org/abs/2106.07841