# Reinforcement Learning Basics:

## Sample Complexity & Beyond


Prof. Lin Yang

UCLA Samueli Electrical & Computer Engineering


RLChina 2021

# Outline

- Learning complexity for multi-armed bandits (MAB) and MDP

    1. Multi-armed bandit
    2. MDP revisit
    3. Sample complexity of MDP with a generative model
    4. Regret minimization for finite-horizon MDP
    5. Function approximation

# Multi-Arm Bandit (MAB)

- Definition

# MAB: Simple RL Problem

- Single decision
- Single state
- A set of arms (actions) to pull
  - $A = \{a_1, a_2, \ldots a_k\}$
- Once an arm is pulled, the environment returns a reward $r$
  - $r$ a random number
  - $\mathrm{E}[r|a_i] = \mu_i \in [0,1]$ (Markovian)
  - $\mathrm{Var}[r] = O(1)$
- Which arm to pull?

# Best arm

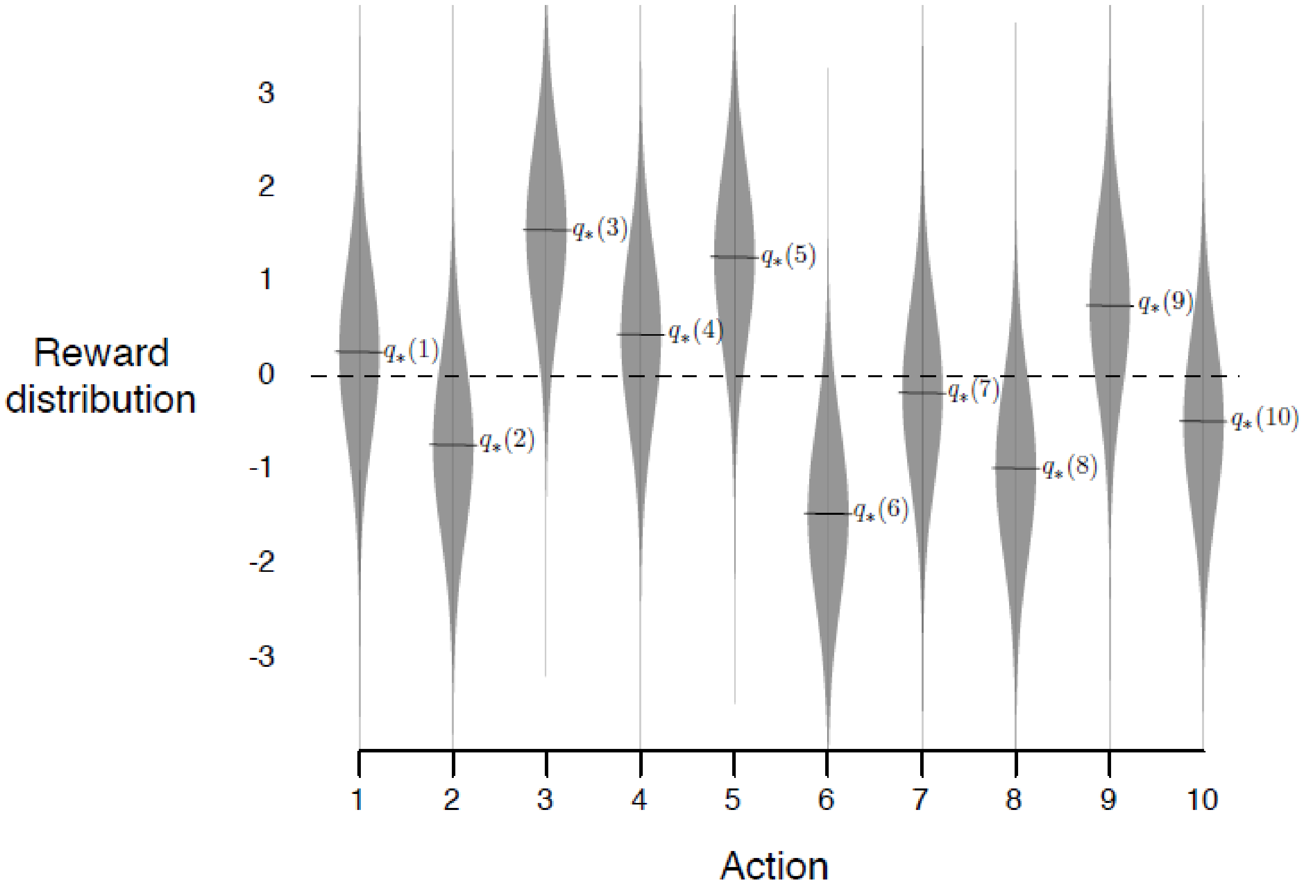
- Best arm: the arm with the highest mean $\mu_i$
  - In expectation, the best reward
  - Law of large number: the best return in a long run
- Learning:
  - $\mu_i$ is unknown!
  - How to identify the best arm?

- Optimal policy: $i^* = \mathrm{argmax}_{i \in [K]} \mu_i$ (unknown)

# Example

- Experts learning problem
  - A set of stock experts, each of which has unknown expected return per day
  - Who to be trusted to put your investment?
- How to choose the correct arm?

# MAB Example



3
2
1
0
-1
-2
-3
Reward distribution
$q_*(1)$
$q_*(2)$
$q_*(3)$
$q_*(4)$
$q_*(5)$
$q_*(6)$
$q_*(7)$
$q_*(8)$
$q_*(9)$
$q_*(10)$
1 2 3 4 5 6 7 8 9 10
Action

# Simple Algorithm: Best Arm Identification

- Algorithm:
  - Pull each arm $N$ times
    - For arm $i$, receive rewards $r_1^{(i)}, r_2^{(i)}, \dots r_N^{(i)}$
  - Estimate the mean
    - $\hat{\mu}_i = \frac{1}{N} \sum_{j=1}^{N} r_j^{(i)}$
  - Output $\hat{\imath}^* = \max_{i \in [K]} \hat{\mu}_i$

- Policy?

# Best Arm Identification: Performance

- Hoeffding Inequality
  - For each $i$, with probability at least $1-\delta$,
$$\left|\frac{1}{N}\sum_{j=1}^{N} r_j^{(i)} - \mu_i\right| \leq \sqrt{\frac{\log\frac{2}{\delta}}{2N}}$$
  - For all $i \in [K]$, $|\widehat{\mu}_i - \mu_i| \leq \sqrt{\frac{\log\frac{2K}{\delta}}{2N}} =: \epsilon_N$
  - $|\widehat{\mu}_i - \mu_i| \leq \epsilon_N \sim \frac{1}{\sqrt{N}}$

- $\mu_{\hat{\imath}^*} \geq \hat{\mu}_{\hat{\imath}^*} - \epsilon_N \geq \hat{\mu}_{i^*} - \epsilon_N \geq \mu_{i^*} - 2\epsilon_N$

# Issues?

- Need to pull each arm
  - Some arm is very sub-optimal, no need to pull that

- Better way?
  - Online:
    - ➢ Use the all the historical information to decide what to pull next
    - ➢ Improving selection all the way until the end

  - How to avoid local-optimal?
    - ➢ Random exploration
    - ➢ Upper-confidence bound

# Simple Algorithm

**A simple bandit algorithm**

Initialize, for $a = 1$ to $k$:

$Q(a) \leftarrow 0$

$N(a) \leftarrow 0$

Loop forever:

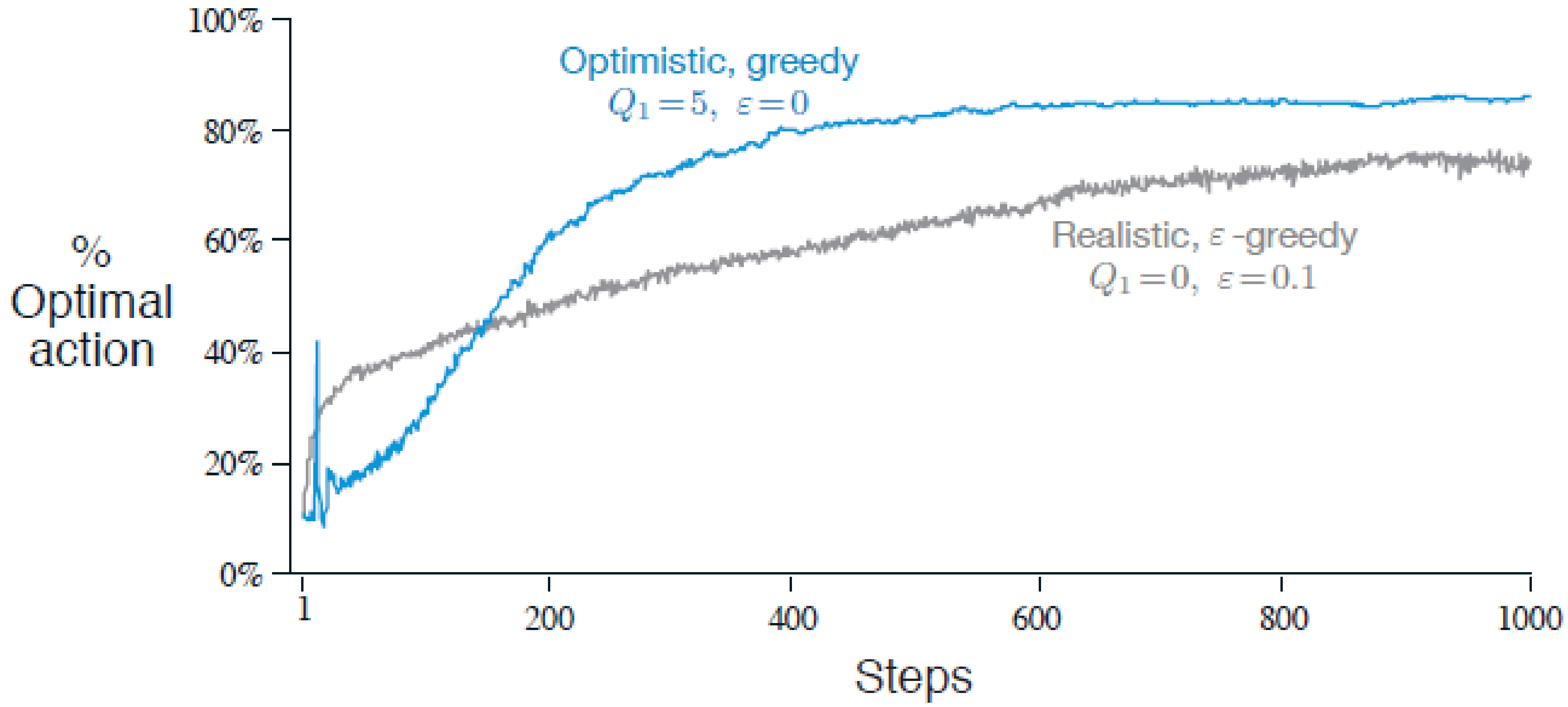
$$A \leftarrow \begin{cases} \arg\max_a Q(a) & \text{with probability } 1-\varepsilon \quad \text{(breaking ties randomly)} \\ \text{a random action} & \text{with probability } \varepsilon \end{cases}$$

$R \leftarrow bandit(A)$

$N(A) \leftarrow N(A) + 1$

$Q(A) \leftarrow Q(A) + \frac{1}{N(A)}\left[R - Q(A)\right]$

# Performance



# Regret

- How to measure the performance of an online algorithm?
  - Compare it with the best policy
  - Regret = <u>E[Rewards collected by the best policy] – E[reward by the algorithm]</u>

$$\mathrm{Regret}[T] = T\mu_{i^*} - \sum_{t=1}^{T} \mu_{\hat{\imath}_t^*}$$

  - Average regret:

$$\mathrm{Regret}[T]/T$$

  - Effective algorithm: $\frac{\mathrm{Regret}[T]}{T} \to 0$ as $T \to \infty$
  - $\epsilon$-greedy: $\frac{\mathrm{Regret}[T]}{T} \to O(\epsilon)$

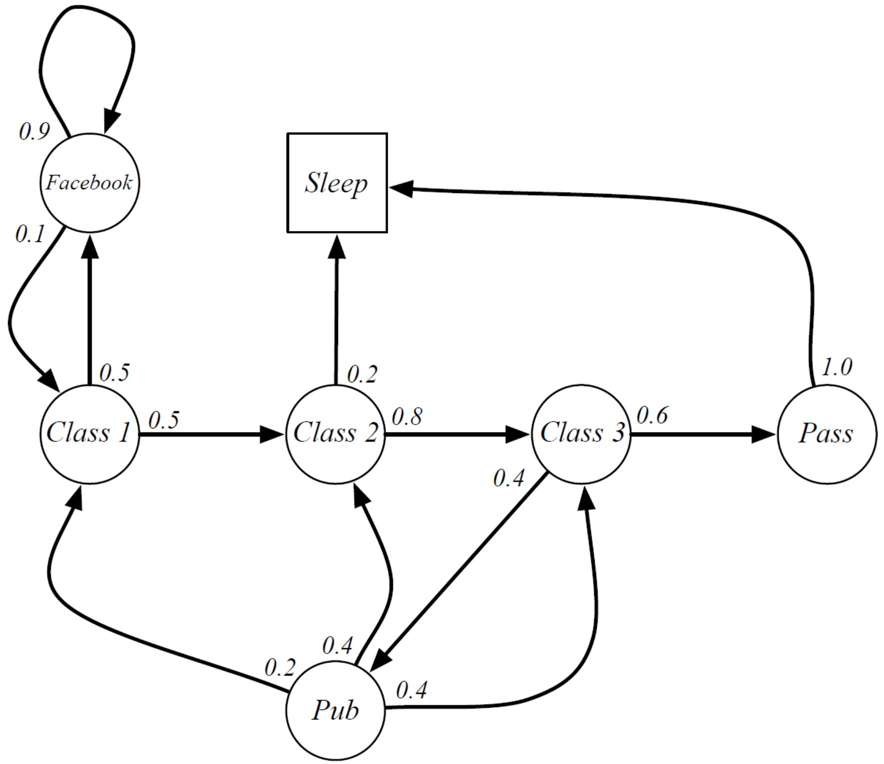
# Markov Process

- Random **process** in which the future is independent of the past

- A set of states $S$
- Probability transition: $P$

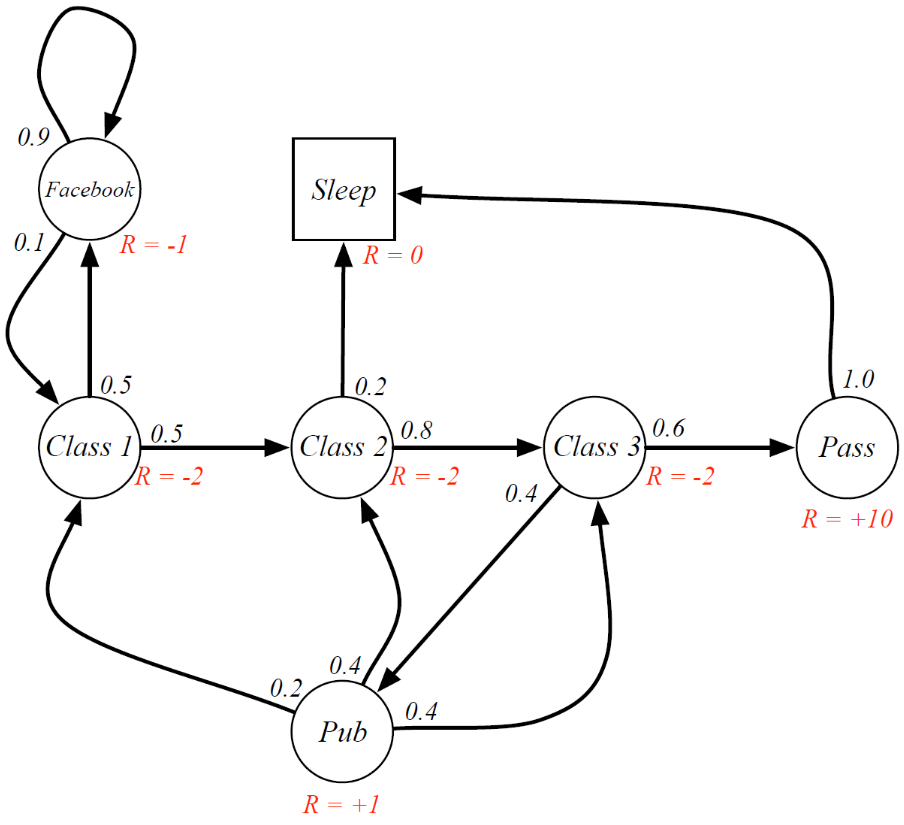
  - $P[s_{t+1}|s_t, s_{t-1}, \dots s_1] = P[s_{t+1}|s_t]$

- $(P, S)$



# Markov Reward Process (discounted)

- Markov process + Reward: MRP
  - An RL model with no need to learn
- A set of states $S$
- Probability transition: $P$
- Reward: $R: S \to \mathbb{R}$
- Discount factor: $\gamma \in (0,1]$
  - At any time $t$, future reward at time $t+i$ is discounted by $\gamma^i$
  - No discount: $\gamma = 1$
  - Avoids infinity in sum
  - Effective horizon: $\frac{1}{1-\gamma}$
- $M = (P, S, R, \gamma)$

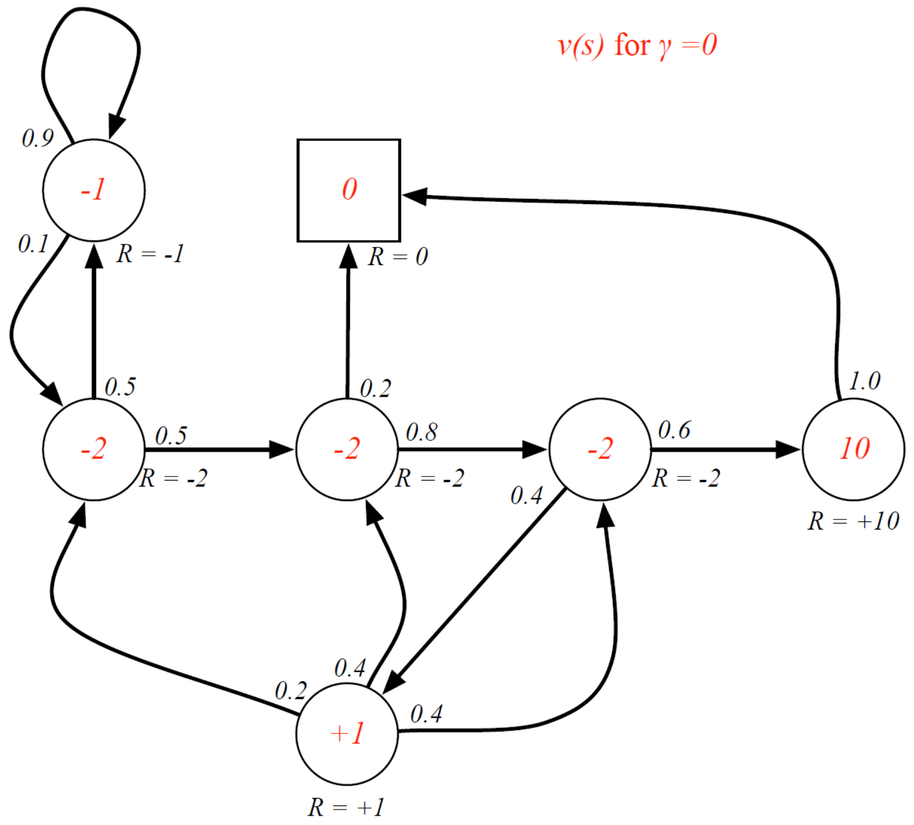


# Value Function

- A function of states $S$
  - No need to consider specific policy
  - Measures the expected long-term return starting from a given state

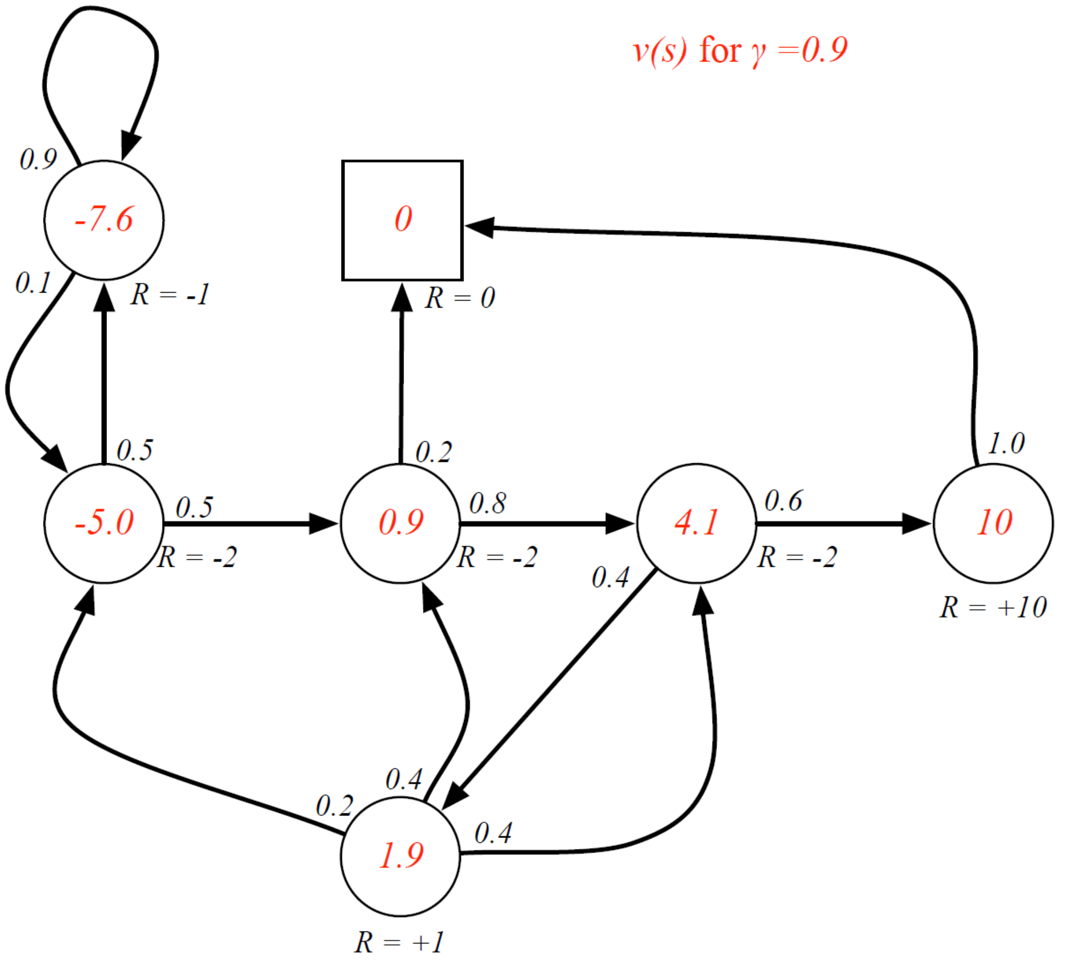
$$V(s) \coloneqq E[\sum_{t=0}^{\infty} \gamma^t R(s_t) | s_0 = s]$$

  - Has long term effect!



# Value function

- Effective horizon:
  - $\frac{1}{1-\gamma} = 10$



# Bellman Equation

- What is the value function of a Markov reward process?
  - Suppose $V$ is the value function, then

$$V(s) = R(s) + \gamma \sum_{s\prime \in S} P[s'|s]V(s')$$

  - Short form: $V = R + \gamma PV$
  - In English:
    - Current Value := Current Reward + Expected Discounted Next Step Reward

# Solution of Bellman Equation

- Given $P, R, \gamma$

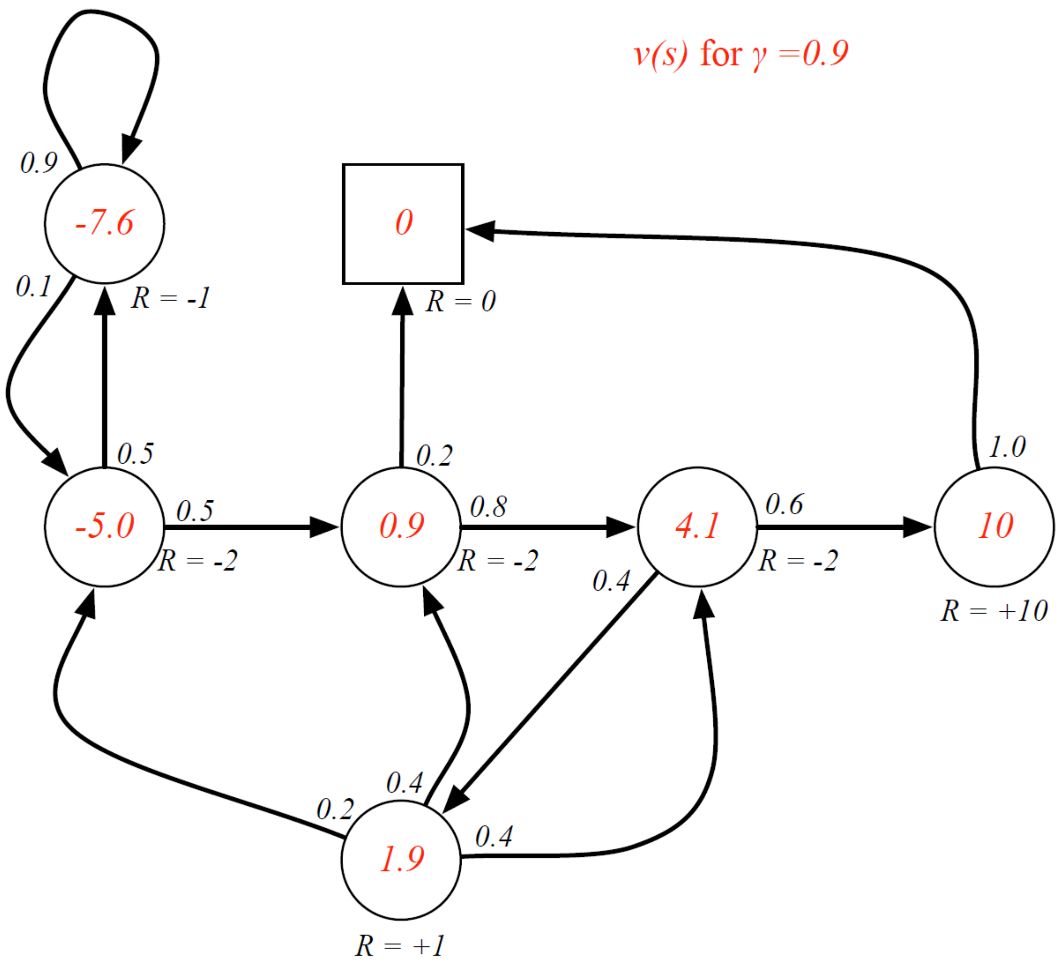
$$V = (I - \gamma P)^{-1} R$$

$\mathbb{R}^{S \times S}$ $\mathbb{R}^{S \times S}$

$$= R + \gamma P R + (\gamma P)^2 R$$

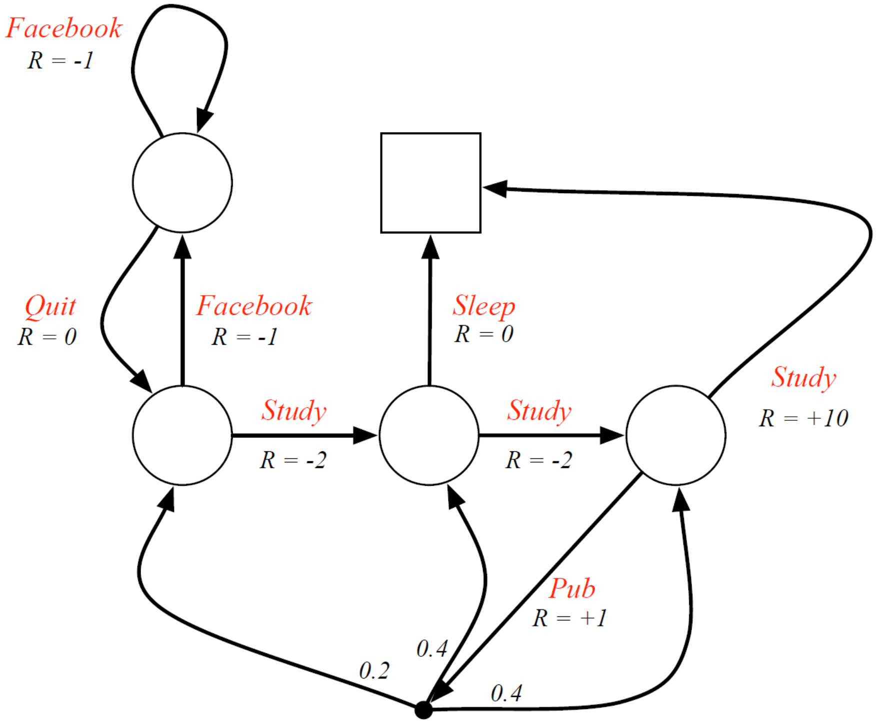
$$\cdots (\gamma P)^{\infty} R$$

$$= \sum_{t=0}^{\infty} (\gamma P)^t R$$



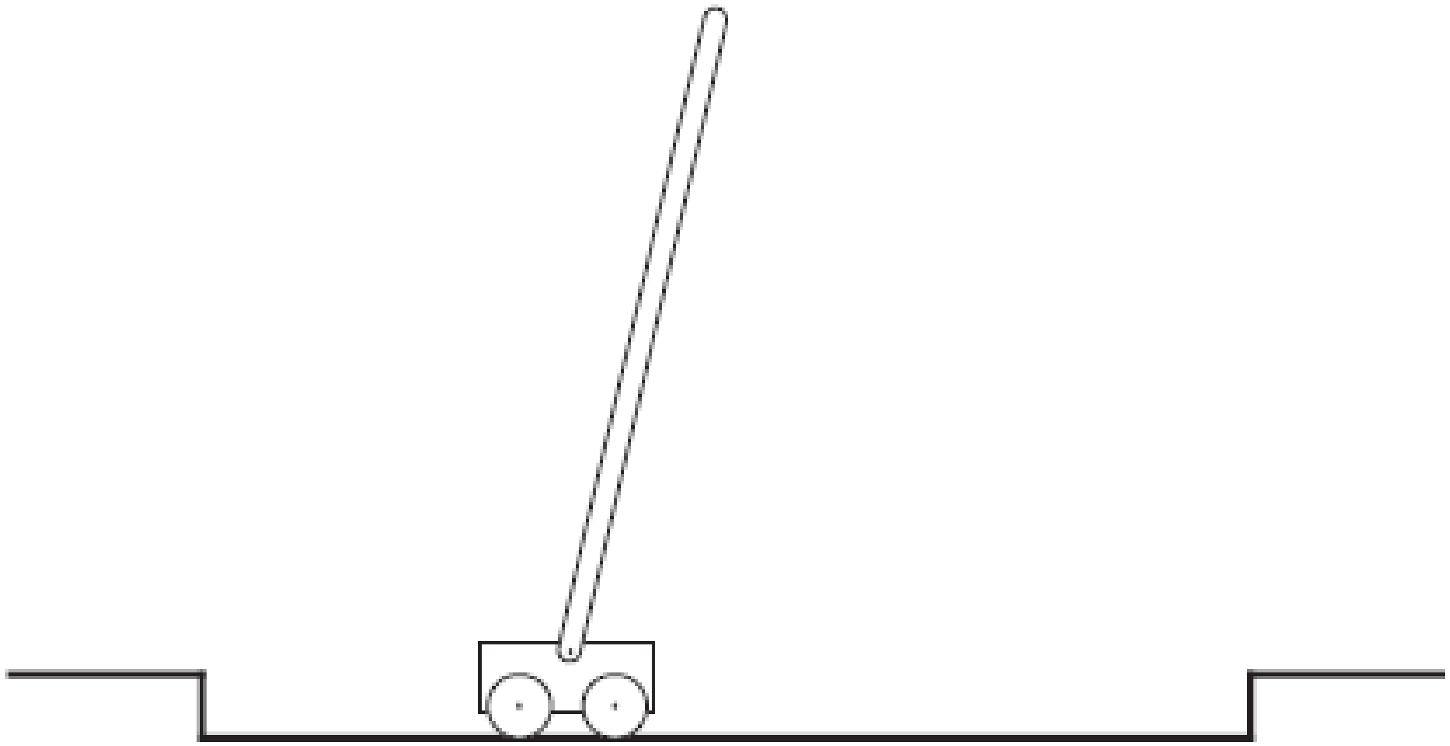
# Markov Decision Process (discounted)

- Markov Reward Process + Actions
  - States $S$
  - Actions $A$
  - Reward depends on actions:
    - $R(s, a)$
  - Probability transition depends on actions:
    - $s_{t+1} \sim P(\cdot\, | s_t, a_t)$
  - Discount $\gamma \in (0,1)$
- $M = (P, S, A, R, \gamma)$



# Pole-Balancing

- States?
- Actions?
- Rewards?
- Transitions?

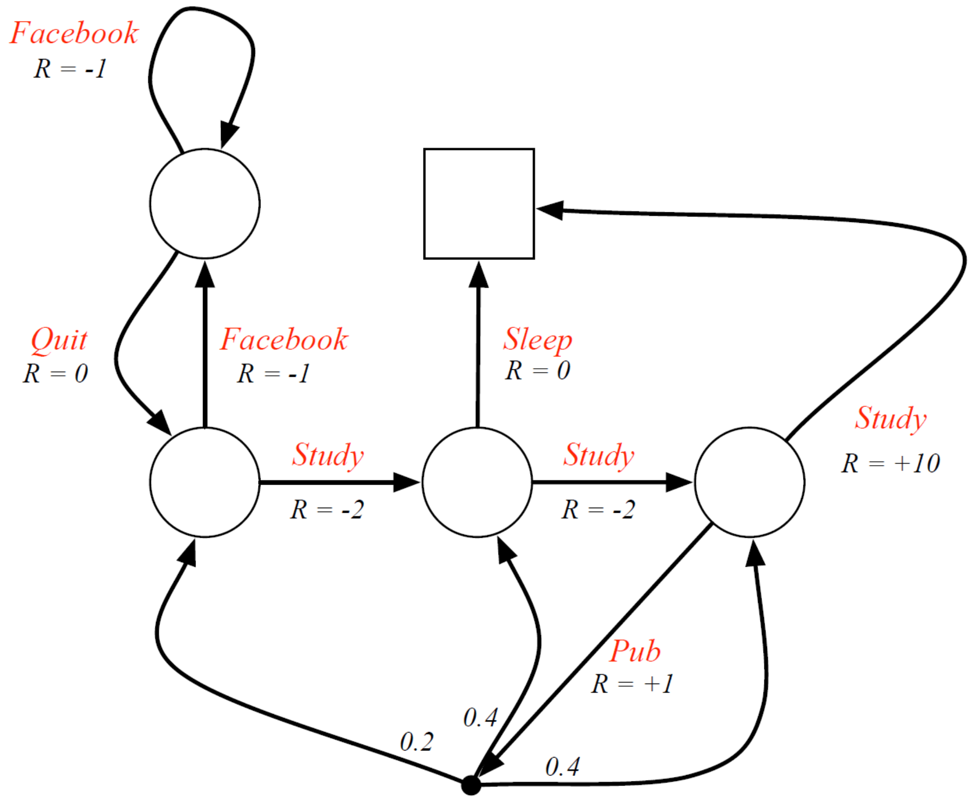


# Discounted MDP

- Stationary Policy
  - Deterministic policy
    - $\pi: S \to A$
  - Randomized policy
    - $\pi: S \to \Delta_A$ (a distribution on actions)
      - E.g. $\pi(s) = (0.1, 0.2, 0.5, 0.2)$ on $(a_1, a_2, a_3, a_4)$
- Policy + MDP = MRP (???)

# Value function of Policy

- Policy + MDP = MRP
  - $(P^\pi, S, R^\pi, \gamma)$:
  - Deterministic policy
    - $P^\pi(s'|s) = P(s'|s, \pi(s))$
    - $R^\pi(s) = R(s, \pi(s))$
  - Randomized policy
    - $P^\pi(s'|s) = \sum_{a \in A} P(s'|s, a)\, \pi(a|s)$
    - $R^\pi(s) = \sum_{a \in A} R(s, a)\, \pi(a|s)$




- Value function of policy $\pi$
  - Equal to the value function of corresponding MRP

# Bellman Equation of MDP + Policy

- What is the value function of a policy $\pi$?
  - Suppose $V^\pi$ is the value function, then

$$V^\pi(s) = R^\pi(s) + \gamma \sum_{s\prime \in S} P^\pi[s'|s] V^\pi(s)$$

  - Short form:$V^\pi = R^\pi + \gamma P^\pi V^\pi = \mathcal{T}^\pi V^\pi$

  - In English:
    - Current Value := Current Reward + Expected Discounted Next Step Reward
  - Solution:

$$V^\pi = (I - \gamma P^\pi)^{-1} R^\pi$$

# Q-function?

- Action-value

$$Q^{\pi}(s,a) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^{t} r(s_t, a_t) \;\; | \; s_0 = s, a_0 = a\right], \qquad a_t = \pi(s_t)$$

- Express using value function

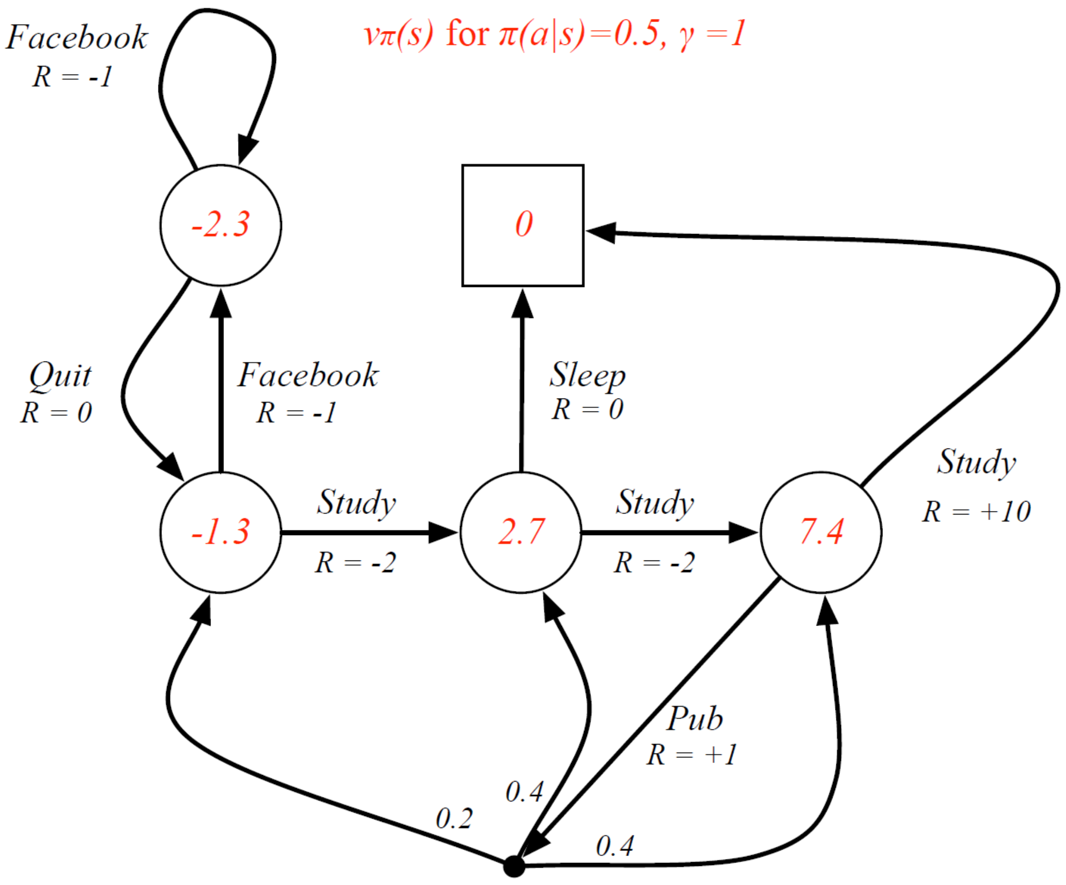
$$Q^{\pi}(s,a) = R(s,a) + \gamma P(\cdot\,|s,a)^{\top} V^{\pi}$$

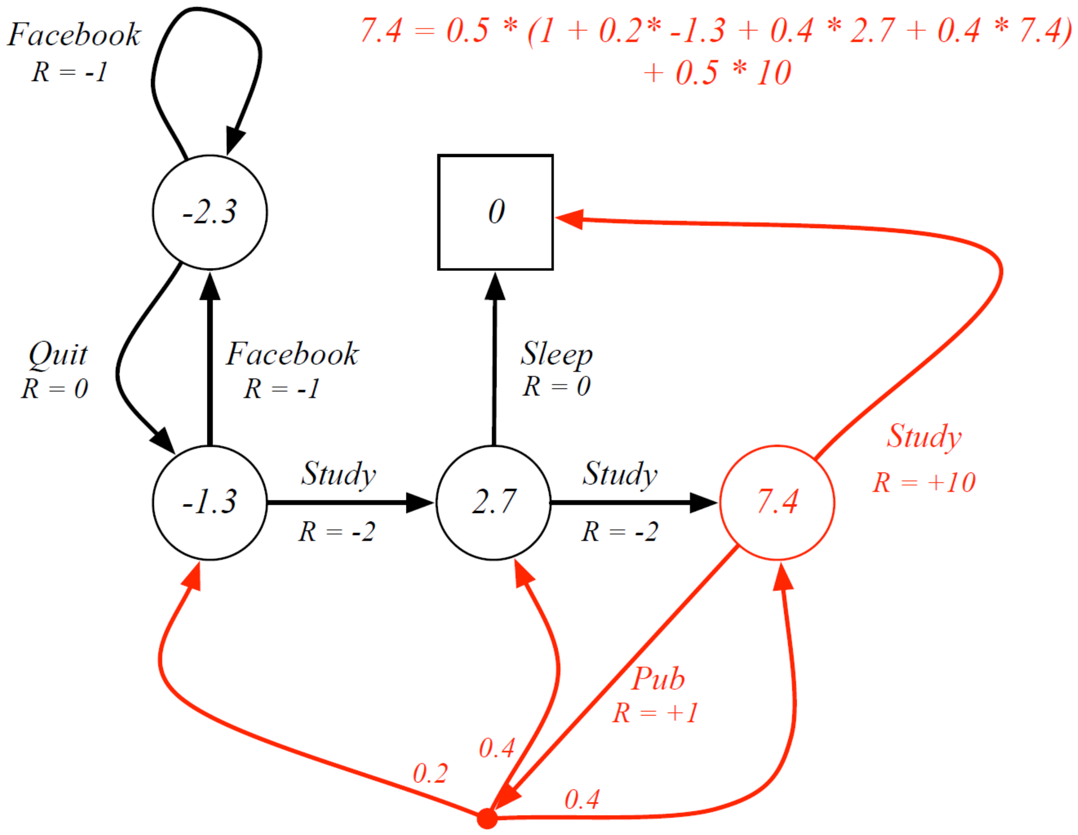
Current reward + expected discounted future rewards

- Bellman Equation with Q

$$V^{\pi} = Q^{\pi}\big(s, \pi(s)\big) \text{ or } V^{\pi} = \sum_{a \in A} \pi(a|s) Q^{\pi}(s,a)$$

# Value function



# Optimal Policy and Value

- A policy $\pi^*$ is optimal, if

$$\forall \pi, \forall s \in S \colon V^{\pi^*}(s) \geq V^{\pi}(s)$$

  - ➢ The optimal policy is **better than** any other policy starting from **any state**

- The value of the optimal policy $V^{\pi^*}$
  - ➢ $\pi^*$ exists! (?) Might not be unique
  - ➢ $V^* = V^{\pi^*}$ is unique! (?)
  - ➢ $Q^* = Q^{\pi^*}$ unique too

# (Optimal) Bellman Equation

- Suppose $V^*$ is optimal, then

$$V^*(s) = \max_{a \in A} \left(R(s,a) + \gamma P(\cdot\,|s,a)^\top V^*\right) =: \mathcal{T}V^*$$

  - ➢ $V^* = \mathcal{T}V^*$
  - ➢ Optimal if I play optimal this time, and also future time

- Why $V^*$ exists and unique?
  - ➢ Contraction: $||\mathcal{T}V_1 - \mathcal{T}V_2||_\infty \leq \gamma ||V_1 - V_2||$
  - ➢ Start form any $V_0$, $\mathcal{T}^n V_0 \to V^*$ for $n \to \infty$

# Optimal Policy from Value

- $Q^* = R + \gamma P V^*$
- Greedy policy of $Q^*$: $\pi^*(s) \coloneqq \mathrm{argmax}_{a \in A} Q^*(s, a)$
  - This is **an** optimal policy (?)
    - $V^* = R^{\pi^*} + \gamma P^{\pi^*} V^*$
    - $V^* = \left(I - \gamma P^{\pi^*}\right)^{-1} R^{\pi^*} = V^{\pi^*}$

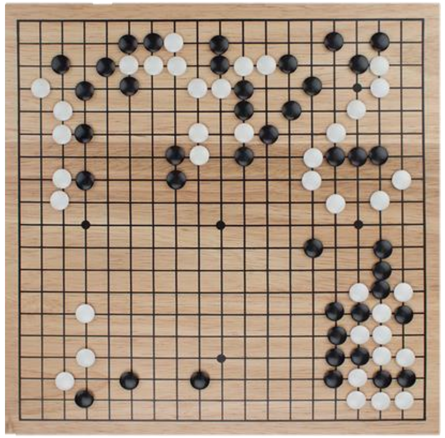
- Uniqueness?
  - If $\mathrm{arg}max_a\ Q(s, a)$ is not unique, then infinite many $\pi^*$
  - Otherwise, unique

# MDP Variants

- Finite horizon
  - $(P, S, A, R, H)$ – only consider length $H$ episodes
  - $\pi(s, h) \to A$ *[Non-stationary policy]*
  - $V_h^\pi(s) = E\left[\sum_{t=h}^{H} R(s_t, \pi(s_t, t)) | s_h = s\right]$, $Q_h^\pi(s, a) = R + PV_{h+1}^\pi$
  - Similar for the optimal policy/values
- Average Reward
  - $(P, S, A, R)$ – infinite horizon ($\gamma = 1$)
  - $V(s) \coloneqq \lim_{T \to \infty} E[\sum_{t=0}^{T} T^{-1} R(s_t) | s_0 = s]$ (Q-function does not make sense)
- POMDP – belief state
  - Observation $\neq$ states
  - Use history as information state $\Rightarrow$ MDP
  - Exponentially Hard

# Sample Complexity with a Generative Model

- What is a "generative model"? [Kearns&Singh'99][Kakade'03]
  - One can obtain samples from any $(s, a)$
  - Probability matrix is unknown
- Why generative model?
  - Clean statistic theory
  - Connection to practice: simulators
- Questions to ask
  - <u>How many samples are sufficient and necessary to obtain a good policy?</u>
    - E.g., obtain a policy $\pi$ with value $V^{\pi}(s) \geq V^{*}(s) - \epsilon$ for all $s$.
  - <u>What algorithms can achieve the optimal sample complexity?</u>



# Sample Complexity with a Generative Model

- Naïve algorithm (plug-in approach)
  - Collect $N$ samples from each state-action pair
  - Construct an empirical model $\widehat{M} = (S, A, \hat{P}, R, \gamma)$
    - $\hat{P}(s'|s,a) = \frac{\#(s,a)\to s'}{N}$
    - Example: $s, a \to [s_1, s_1, s_2, s_1]$, then $\hat{P}(s_1|s,a) = 3/4$, $\hat{P}(s_2|s,a) = 1/4$
  - Solve $\widehat{M}$ for an optimal policy $\pi^*$
    - Dynamic program
    - Linear programming

# A Coarse Analysis of the Empirical Value Iteration Approach

- Approximate dynamic programming

$$\mathcal{T}V := \max_{a\in A}\left(R(s,a) + \gamma P(\cdot\,|s,a)^\top V\right)$$

$$\hat{\mathcal{T}}V := \max_{a\in A}\left(R(s,a) + \gamma \hat{P}(\cdot\,|s,a)^\top V\right)$$

- Measure concentration: Hoeffding inequality
  - With probability at least $1-\delta$,

$$||\mathcal{T}V - \hat{\mathcal{T}}V||_\infty \le \sqrt{\frac{\log\frac{2|S||A|}{\delta}}{2N}}\cdot\frac{\gamma}{(1-\gamma)}$$

  - provided $||V||_\infty \le \frac{1}{(1-\gamma)}$

- Telescopic sum
  - $||\hat{V}^* - V^*||_\infty \le \sqrt{\frac{\log\frac{2|S||A|}{\delta}}{2N}}\cdot\frac{\gamma}{(1-\gamma)^2}$

$$||\hat{V}^* - V^*||_\infty = ||\hat{\mathcal{T}}^\infty V^* - V^*||_\infty$$

$$= ||\hat{\mathcal{T}}V^* - V^* + \hat{\mathcal{T}}^2V^* - \hat{\mathcal{T}}V^* + \hat{\mathcal{T}}^3V^* - \hat{\mathcal{T}}^2V^* \ldots||_\infty$$

$$= ||\hat{\mathcal{T}}V^* - V^* + \hat{\mathcal{T}}^2V^* - \hat{\mathcal{T}}V^* + \hat{\mathcal{T}}^3V^* - \hat{\mathcal{T}}^2V^* \ldots||_\infty$$

$$\le \sum_i \gamma^i\,||\hat{\mathcal{T}}V^* - V^*||_\infty \le \sqrt{\frac{\log\frac{2|S||A|}{\delta}}{2N}}\cdot\frac{\gamma}{(1-\gamma)^2}$$

# A Coarse Analysis of the Empirical Value Iteration Approach

- If we want $||\hat{V}^* - V^*||_\infty \leq \epsilon$, for $\epsilon \in [0, (1-\gamma)^{-2}]$, we need

$$\sqrt{\frac{\log \frac{2|S||A|}{\delta}}{2N}} \cdot \frac{\gamma}{(1-\gamma)^2} \leq \epsilon$$

- $\Rightarrow N = \Omega\left(\frac{\log\frac{|S||A|}{\delta}}{\epsilon^2} \cdot \frac{\gamma^2}{(1-\gamma)^4}\right)$
- There are $|S||A|$ state-action pairs, total samples $\propto \frac{|S||A|N}{(1-\gamma)} \cdot \log\frac{1}{\epsilon}$
- Running time? $\frac{|S||A|N}{1-\gamma} \log \epsilon^{-1}$

# A Coarse Analysis of the Empirical Value Iteration Approach

- From $\hat{V}^*$ to a policy $\hat{\pi}^*$, we have (by telescopic sum)

$$||V^{\hat{\pi}^*} - V^*||_\infty \leq \frac{\epsilon}{(1-\gamma)}$$

- Final sample complexity

$$N = \tilde{O}\left(\frac{|S||A|}{(1-\gamma)^7 \epsilon^2}\right)$$

* $\tilde{O}(\cdot)$ hides logarithmic factors

# Improved Analysis

- Variance reduced valued iteration [Sidford, Wang, Wang, Yang, Ye' 2018]
  - Bernstein inequality instead of Hoeffding, with high probability

$$\left|P(\cdot\,|s,a)^\top V - \hat{P}(\cdot\,|s,a)^\top V\right| \le \tilde{O}\left(\sqrt{\frac{\sigma(s,a)}{N}} + \frac{1}{N(1-\gamma)}\right)$$

  - Variance reduction: reuse previous samples to save samples

$$\left|(P(\cdot\,|s,a)^\top - \hat{P}(\cdot\,|s,a)^\top)(V - V^0)\right| \le \tilde{O}\left(\sqrt{\frac{1}{N}} \cdot ||V - V^0||_\infty\right)$$

  - Law of total variance: total variance = sum of per step variance

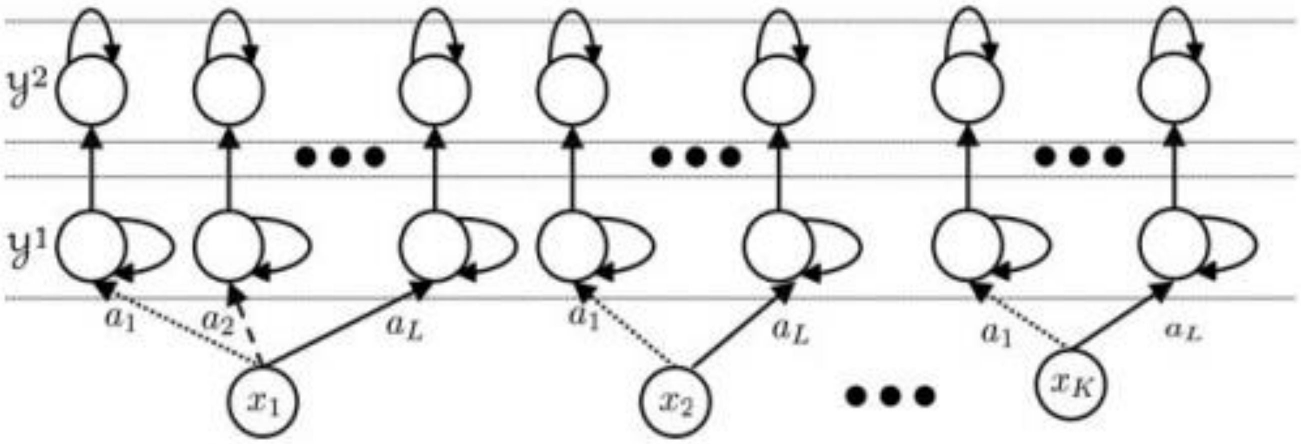
$$\mathrm{Var}[r_1 + \gamma r_2 + \gamma^2 r_3 \ldots] = 1 + \gamma^2 P^\pi \sigma^\pi + \gamma^4 (P^\pi)^2 \sigma^\pi \ldots$$

  - Final sample complexity: for $\epsilon \in (0,1)$, by take $N = \tilde{O}\left(\frac{1}{(1-\gamma)^3} \cdot \frac{1}{\epsilon^2}\right)$

$$V^* - V^{\hat{\pi}^*} \le \epsilon$$

# Lower bound

- [Azar, Munos, Kappen' 2013]: any algorithm requires $\Omega\left(\frac{|S||A|}{(1-\gamma)^3} \cdot \frac{1}{\epsilon^2}\right)$ samples to output an $\epsilon$-optimal policy



# Recap

- Recent advances
  - [Agarwal, Kakade, Yang' 2020] Plug-in approach is minimax optimal for $\epsilon \in \left(0, \frac{1}{\sqrt{1-\gamma}}\right]$
  - [Li et al' 2020] for $\epsilon \in \left(0, \frac{1}{(1-\gamma)}\right]$
  - [Zhang, Kakade, Baser, Yang' 2020] for two-player game
- Take-home Messages
  - True model has $|S|^2|A|$ entries
  - Approximate model has $\propto \frac{|S||A|}{(1-\gamma)^3}$ entries, but preserves good policy
  - Saves planning time

# Online Algorithm

- Online decision making (usually with fixed budget)
    - Exploitation: make the best decision given the current information
        - Current information may not be sufficient to decide the best
    - Exploration: gather more information
        - Exploration may need bad short-term decisions
- A dilemma?
    - The best long-term strategy may involve short-term sacrifices
    - Gather enough information to make the best overall decision

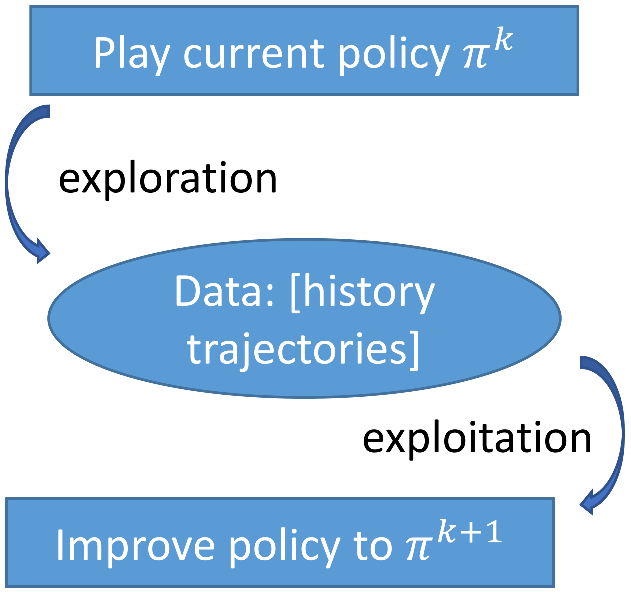
# Sample Efficiency of Online RL

- Episodic finite-horizon setting
  - Unknown $H$-horizon MDP: $M = (S, A, P, R, H)$
  - Episodic: agent interact with the MDP episodically
    - Each episode is of length $H$
    - $(s_1, a_1, r_1) \to (s_2, a_2, r_2) \to \cdots (s_H, a_H, r_H)$
    - Restart from $s_1$ (can be from a distribution)
  - Optimal policy maybe non-stationary (but can be deterministic)
    - $\pi^*: S \times [H] \to A$
    - Value function: $V_1^*, V_2^* \ldots V_H^*$
- Regret
  - An algorithm plays $K$ episodes
  - The policy at time $k \in [K]$ is $\pi^k$

$$\text{Regret}(K) = \sum_{k=1}^{K} V_1^*(s_1) - V_1^{\pi^k}(s_1)$$

# Theories of RL on MDP

- Exploration + exploitation
  - Learn from scratch
  - Exploitation: improve policy based on existing data
  - Exploration: collect more info about the environment
    - *Regret: average error v.s. optimal*




| Algorithm | Ave. Regret | Time | Space |
|---|---|---|---|
| UCRL2 [Jakcsh et al. 2010] | $\geq \tilde{O}(\sqrt{H^4S^2A/T})$ | $\Omega(TS^2A)$ | $O(S^2AH)$ |
| [Agrawal and Jia 2017] | $\geq \tilde{O}(\sqrt{H^3S^2A/T})$ | | |
| UCBVI [Azar et al. 2017] | $\tilde{O}(\sqrt{H^2SA/T})$ | $\tilde{O}(TS^2A)$ | |
| UCB-H [Jin et al. 2018] | $\tilde{O}(\sqrt{H^4SA/T})$ | $O(T)$ | $O(SAH)$ |
| UCB-B [Jin et al. 2018] | $\tilde{O}(\sqrt{H^3SA/T})$ | | |
| Lower bound | $\Omega(\sqrt{H^2SA/T})$ | - | - |

*Recent results include [Zanette&Brunskil' 2018, Zhang et al' 2020 ...]

# References


- Kearns, Michael, and Satinder Singh. "Finite-sample convergence rates for Q-learning and indirect algorithms." *Advances in neural information processing systems* (1999): 996-1002.
- Kakade, S. M. (2003). *On the sample complexity of reinforcement learning*. University of London, University College London (United Kingdom).
- Azar, M. G., Munos, R., & Kappen, H. J. (2013). Minimax PAC bounds on the sample complexity of reinforcement learning with a generative model. *Machine learning*, *91*(3), 325-349.
- Sidford, A., Wang, M., Wu, X., Yang, L. F., & Ye, Y. (2018, December). Near-optimal time and sample complexities for solving Markov decision processes with a generative model. *In Proceedings of the 32nd International Conference on Neural Information Processing Systems (pp. 5192-5202).*
- Agarwal, A., Kakade, S., & Yang, L. F. (2020, July). Model-based reinforcement learning with a generative model is minimax optimal. *In Conference on Learning Theory (pp. 67-83). PMLR.*
- Wainwright, M. J. (2019). Variance-reduced \$ Q \$-learning is minimax optimal. *arXiv preprint arXiv:1906.04697.*

# Does tabular algorithm in practice?

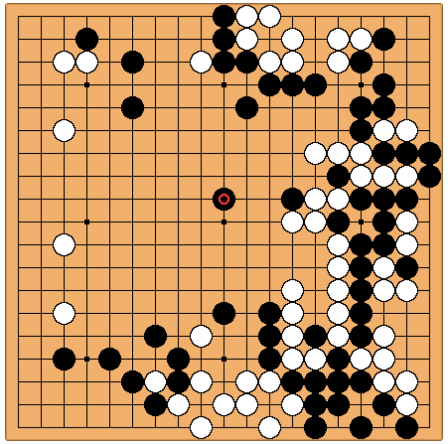
- Number of episodes required to get a good $\pi$

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

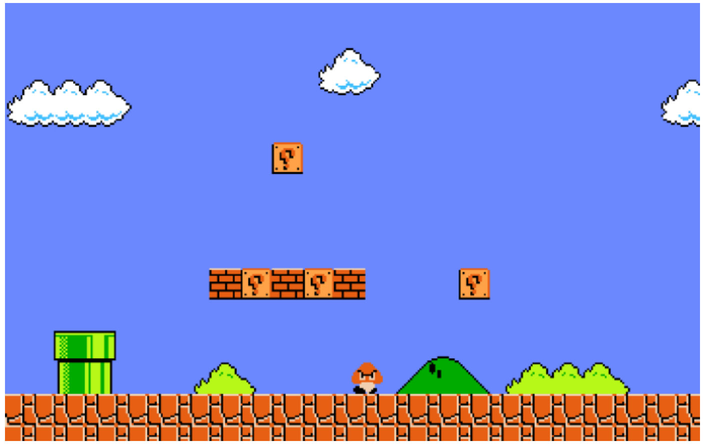
[Jin et al'2018] [Azar et al' 2017][…]

- Curse of Dimensionality

S



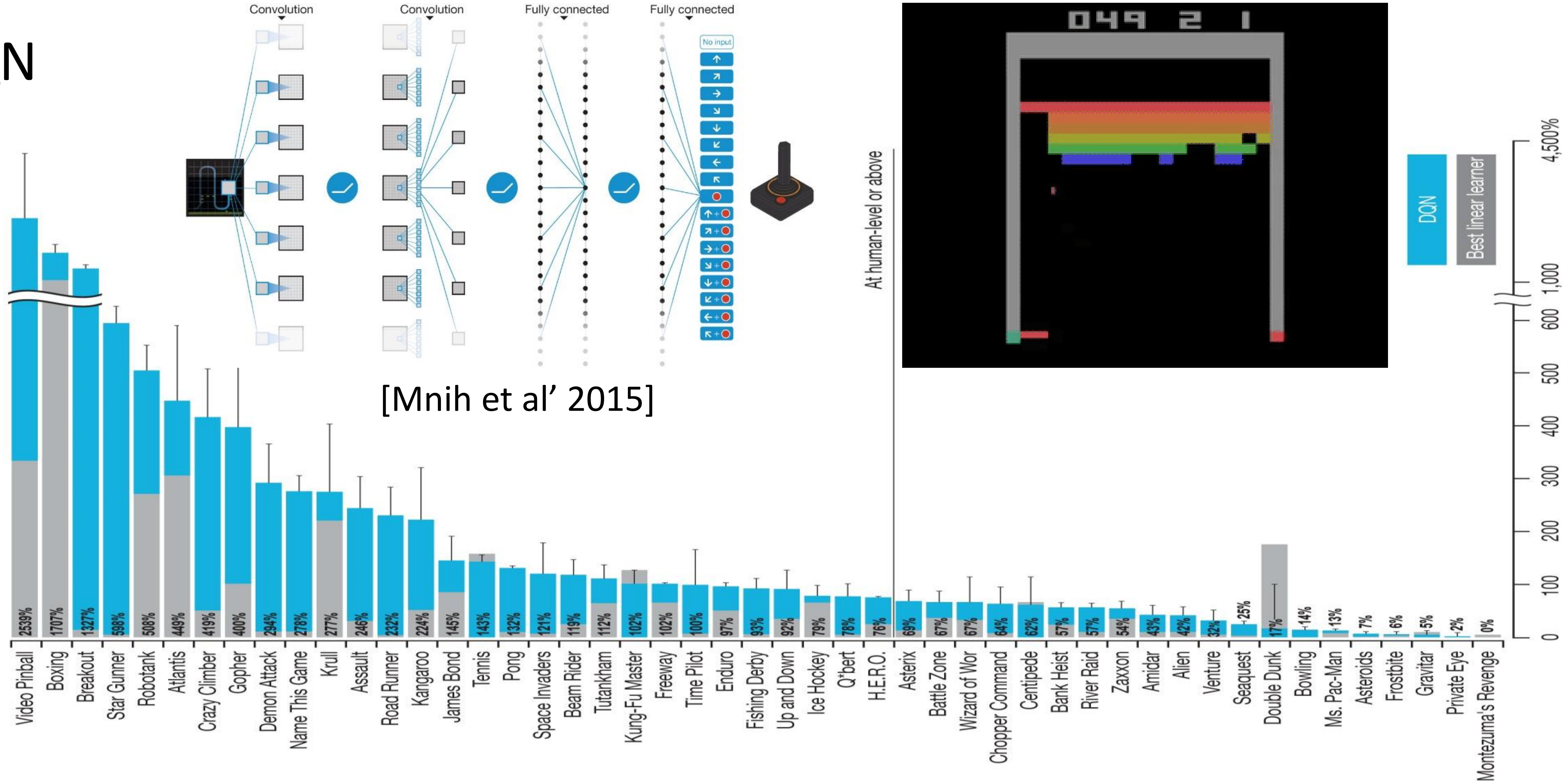
$|S| = 3^{361}$



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

# 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

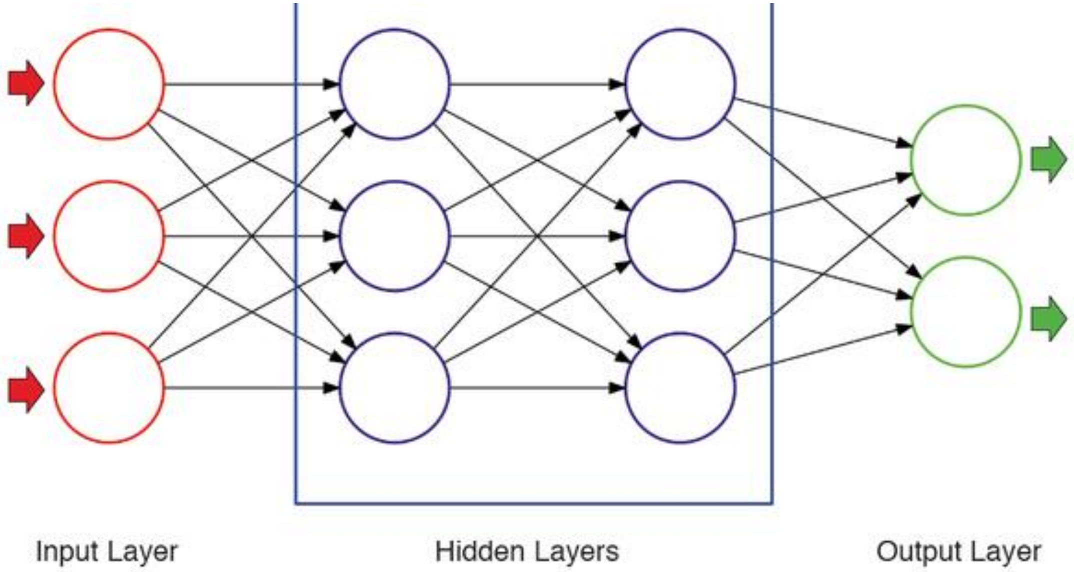
- Theory
  - Markov decision process
    - Finite state space $S$
    - Finite action space $A$
    - Finite horizon $H$
  - Many theoretical results
    - Mostly tabular – well understood
    - Not scalable

- 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 $Q^*(s,a)$

$(s,a)$



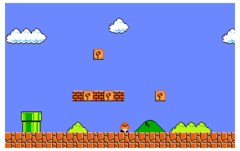

$Q^*(s,a)$

$\pi^*(s,a)$

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

# Linear Function Approximation

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

$\phi$ (  , (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

# LSVI with Generative Model

- Linear MDP [Yang & Wang' 19]

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

- Approximate dynamic programming by sampling

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

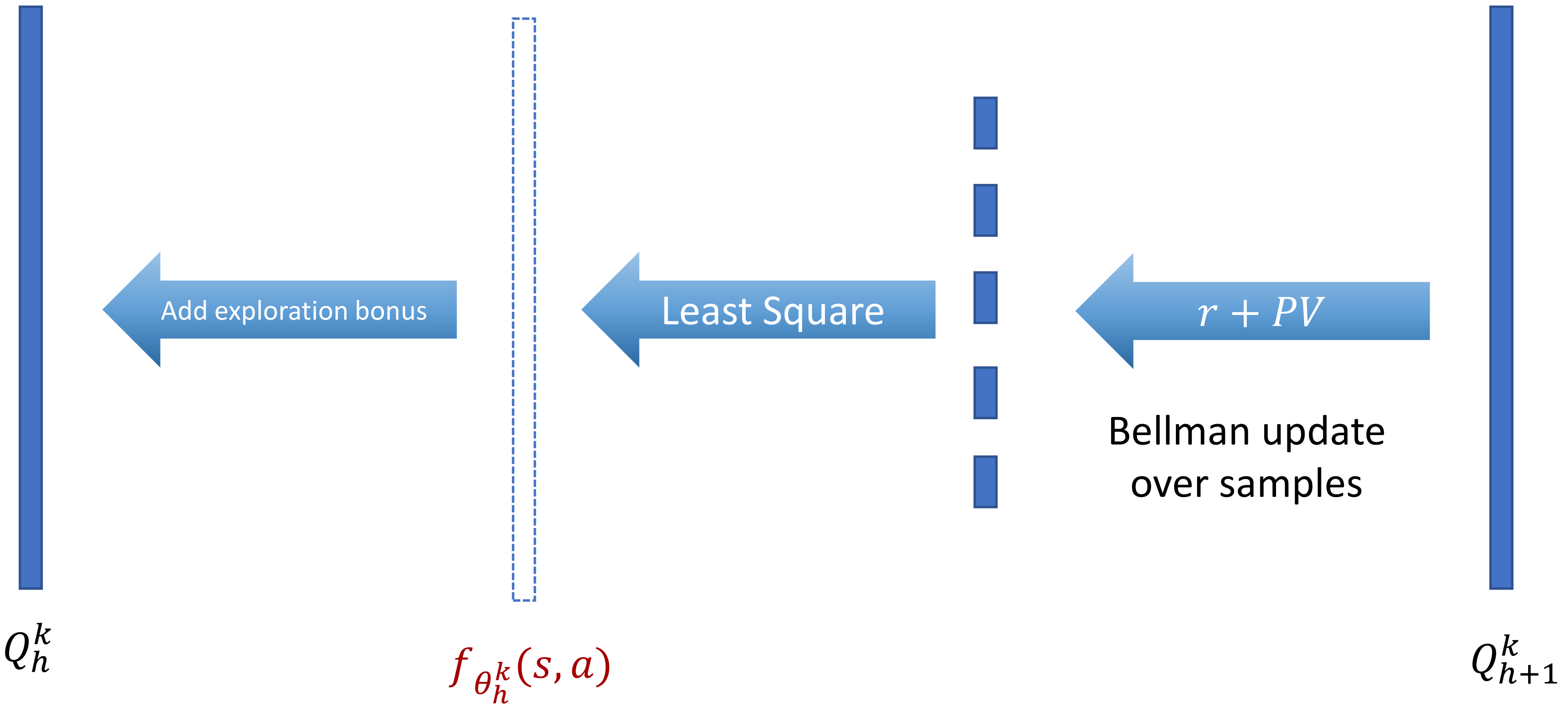
- Number samples needed

$$\frac{\mathrm{poly}(dH)}{\epsilon^2}$$

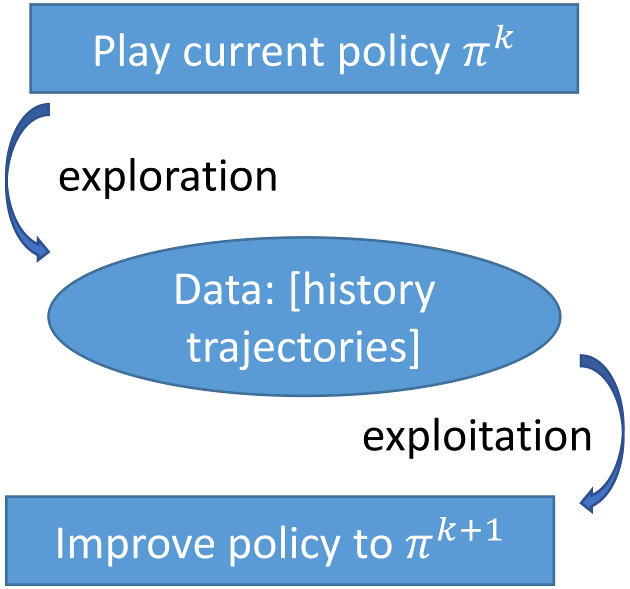
- Stronger anchor condition: $\tilde{O}\left( \frac{d}{\epsilon^2} \cdot \frac{1}{(1-\gamma)^3} \right)$

# LSVI as Approximate Dynamic Programming (ADP)

- Each iteration solves



# 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

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

$$Q_h^k(s, a) = f_{\theta_h^k}(s, a) + \text{exploration bonus}$$

    - Collect a trajectory of data

$$\boxed{\pi_h^k(s) \leftarrow \mathrm{argmax}_a\, Q_h^k(s, a)}$$

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

[R.Wang, Salakhutdinov, **Yang**' 2020]: Functional-LSVI

Extending [Jin et al, 2020]

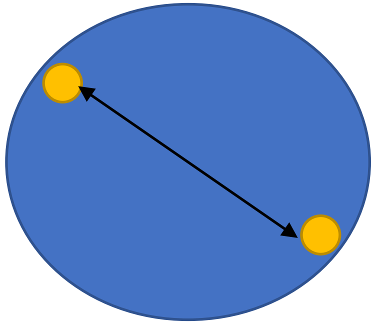
# Exploration bonus

- "prediction uncertainty"
  - Optimism in face of uncertainty (OFU)
  - Natural choice:

$$w^k(s,a) = \text{argmax}_{f_1, f_2 \text{ fits data well}} |f_1(s,a) - f_2(s,a)|$$

defined using the whole experience buffer

width function



  - Linear counterpart:

$$w^k(s,a) = \sqrt{\phi(s,a)^\top \Sigma_t^{-1} \phi(s,a)}$$

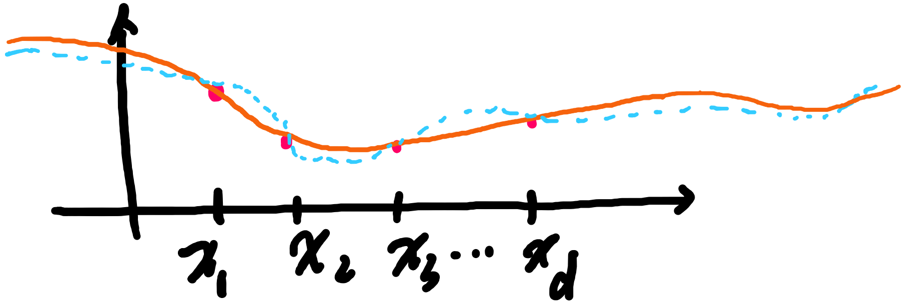
# Theory for General functions

- Realizability Assumption: $r + PV \in \mathcal{F}, \forall V$
  - The function set is the "image" of Bellman projection
  - Corresponding to linear MDP for linear setting

    *other assumptions work, but not time-efficient
- Eluder dimension [Russo&Van Roy' 2013]
  - $d_E$: the longest determination sequence of the function set
  - d-dim linear / generalized linear: $\approx d$
    - * ignoring other factors, see paper for detail

Theorem: [Wang, Salakhutdinov, Yang' 2020]

F-LSVI with **Stable** exploration bonus function takes $\boldsymbol{O}(\mathrm{poly}(\boldsymbol{d_E H}))$ episodes to obtain a good $Q^*$ approximation with high probability

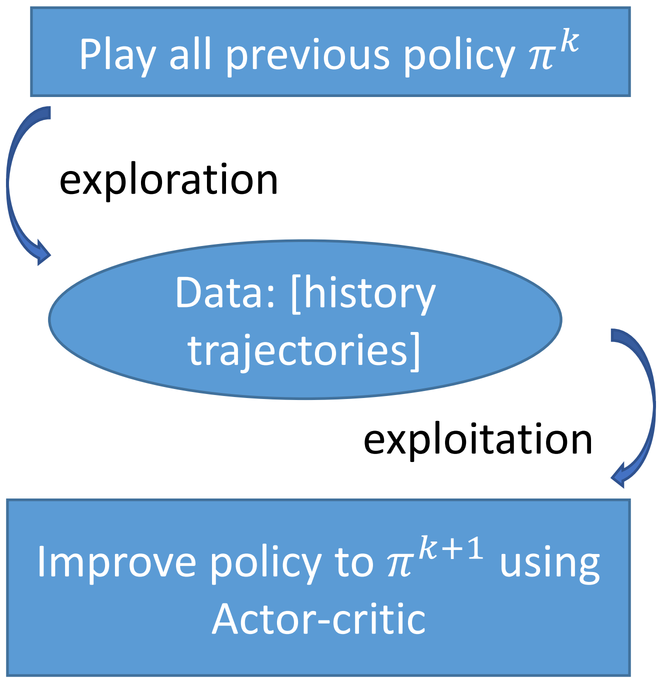



Subsampled buffer size: $\mathrm{M} = \mathrm{poly}(d_E)$

# Policy Gradient Method

- Policy parametrization $\pi_f(a|s) \propto \exp(f(s,a))$

Exploratory Non-linear Incremental Actor Critic (ENIAC):




*Natural policy gradient update:*

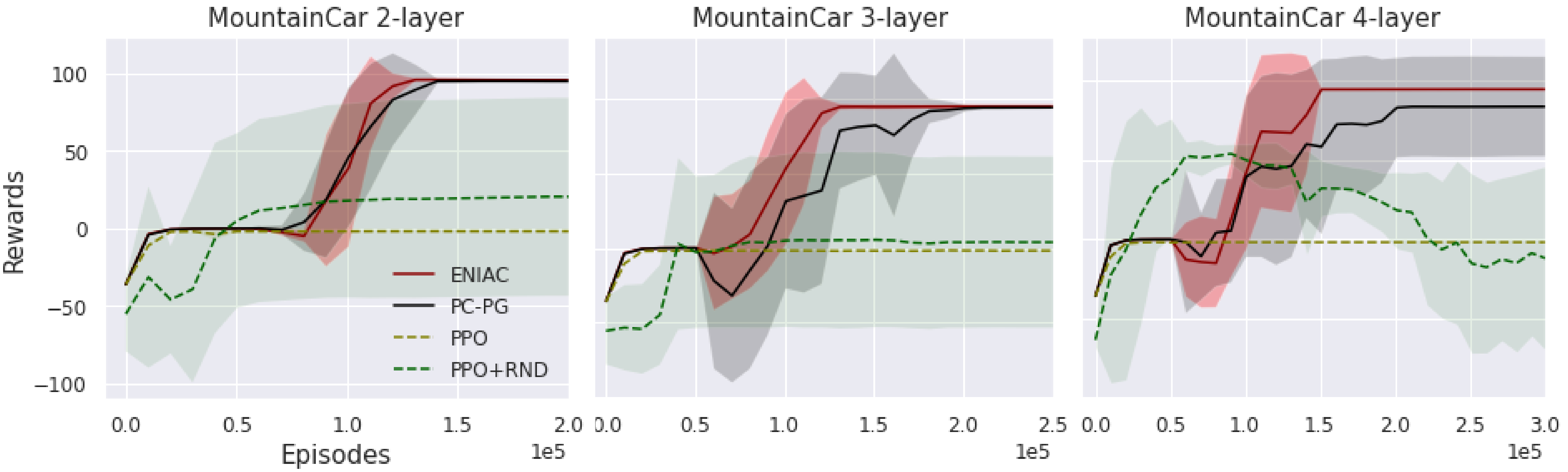
$$u_t \leftarrow \arg\min_u \sum_{i=1}^{M} \left(\hat{A}^{\pi_t}(s_i, a_i, r+b) - \bar{b}_t(s_i, a_i) - u^\top \nabla_{\theta_t} \log \pi_{f_{\theta_t}}(s,a)\right)^2.$$

$$\theta_{t+1} = \theta_t + \eta u_t,$$
$$\pi_{t+1}(a|s) \propto \exp\left((f_{\theta_{t+1}}(s,a) + b(s,a))\mathbf{1}\{s \in \mathcal{K}\}\right);$$

Theorem: [Feng, Yin, Agarwal, Yang' 2021]

ENAIC with exploration bonus function takes $\tilde{O}(\mathrm{poly}(\boldsymbol{d_E H}))$ episodes to obtains a good policy, w.h.p.

# Experiments



# Recap

- Breaking curse of dimensionality:
  - Function approximation
  - Not every function approximation works …
    - Linear function can have exponential lower bound
- Good function approximation ⇒ efficient algorithms, no dependence on state-action pairs
  - Linear-MDP: least-square Q-value iteration
    - Works for both online and offline
  - General function approximation
    - Sufficient condition: Bounded Eluder-dimension

# Reference


- Yang, L., & Wang, M. (2019, May). Sample-optimal parametric Q-learning using linearly additive features. In *International Conference on Machine Learning* (pp. 6995-7004). PMLR.
- Jin, C., Yang, Z., Wang, Z., & Jordan, M. I. (2020, July). Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory* (pp. 2137-2143). PMLR.
- Cai, Q., Yang, Z., Jin, C., & Wang, Z. (2020, November). Provably efficient exploration in policy optimization. In *International Conference on Machine Learning* (pp. 1283-1294). PMLR.
- Du, S. S., Kakade, S. M., Wang, R., & Yang, L. F. (2019). Is a good representation sufficient for sample efficient reinforcement learning?. *arXiv preprint arXiv:1910.03016*.
- Lattimore, T., Szepesvari, C., & Weisz, G. (2020, November). Learning with good feature representations in bandits and in rl with a generative model. In *International Conference on Machine Learning* (pp. 5662-5670). PMLR.
- Duan, Y., Jia, Z., & Wang, M. (2020, November). Minimax-optimal off-policy evaluation with linear function approximation. In *International Conference on Machine Learning* (pp. 2701-2709). PMLR.
- Zhou, D., He, J., & Gu, Q. (2021, July). Provably efficient reinforcement learning for discounted mdps with feature mapping. In *International Conference on Machine Learning* (pp. 12793-12802). PMLR.
- Agarwal, A., Kakade, S., Krishnamurthy, A., & Sun, W. (2020). Flambe: Structural complexity and representation learning of low rank mdps. *arXiv preprint arXiv:2006.10814*.
- Modi, A., Jiang, N., Tewari, A., & Singh, S. (2020, June). Sample complexity of reinforcement learning using linearly combined model ensembles. In *International Conference on Artificial Intelligence and Statistics* (pp. 2010-2020). PMLR.
- Zhang, Z., Ji, X., & Du, S. S. (2021). Is reinforcement learning more difficult than bandits? a near-optimal algorithm escaping the curse of horizon. *Proceedings of Machine Learning Research vol*, *134*, 1-28.
- Agarwal, A., Kakade, S. M., Lee, J. D., & Mahajan, G. (2021). On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *Journal of Machine Learning Research*, *22*(98), 1-76.
- Wang, R., Salakhutdinov, R., & Yang, L. F. (2020). Reinforcement learning with general value function approximation: Provably efficient approach via bounded eluder dimension. *arXiv preprint arXiv:2005.10804*.
- Wang, R., Du, S. S., Yang, L. F., & Salakhutdinov, R. (2020). On reward-free reinforcement learning with linear function approximation. arXiv preprint arXiv:2006.11274.
- Kong, D., Salakhutdinov, R., Wang, R., & Yang, L. F. (2021). Online Sub-Sampling for Reinforcement Learning with General Function Approximation. *arXiv preprint arXiv:2106.07203*.
- Feng, Fei, et al. "Provably correct optimization and exploration with non-linear policies." *arXiv preprint arXiv:2103.11559* (2021).