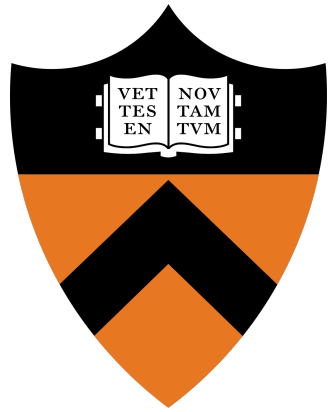
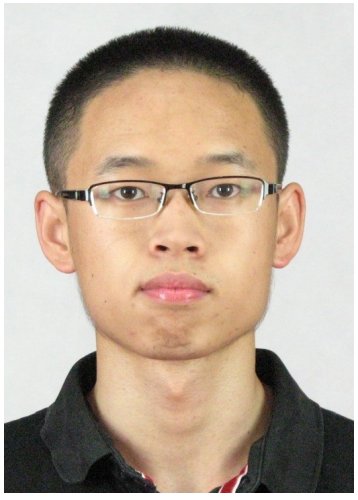
Sample Complexity of Asynchronous Q-Learning:

Sharper non-asymptotic analysis and variance reduction



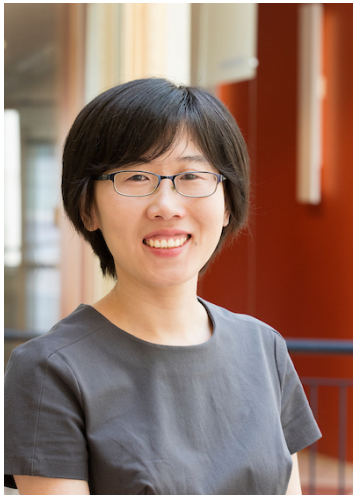
Yuxin Chen

EE, Princeton University

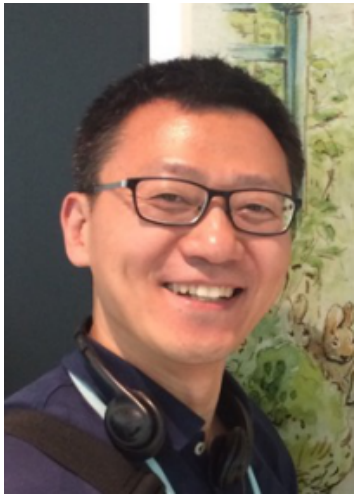


Gen Li
Tsinghua EE

Yuting Wei
CMU Statistics



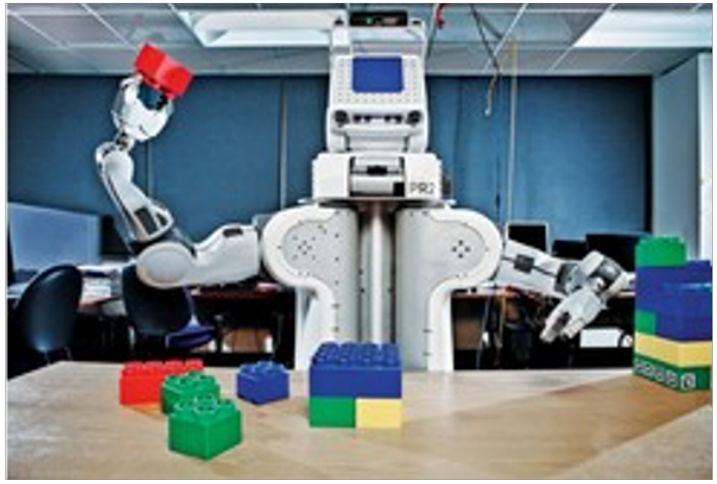
Yuejie Chi
CMU ECE



Yuantao Gu
Tsinghua EE

"Sample complexity of asynchronous Q-learning: sharper analysis and variance reduction," G. Li, Y. Wei, Y. Chi, Y. Gu, Y. Chen, NeurIPS 2020

Reinforcement learning (RL)



RL challenges

- Unknown or changing environments
- Delayed rewards
- Enormous state and action space

Sample efficiency

Collecting data samples might be expensive or time-consuming

clinical trials

online ads

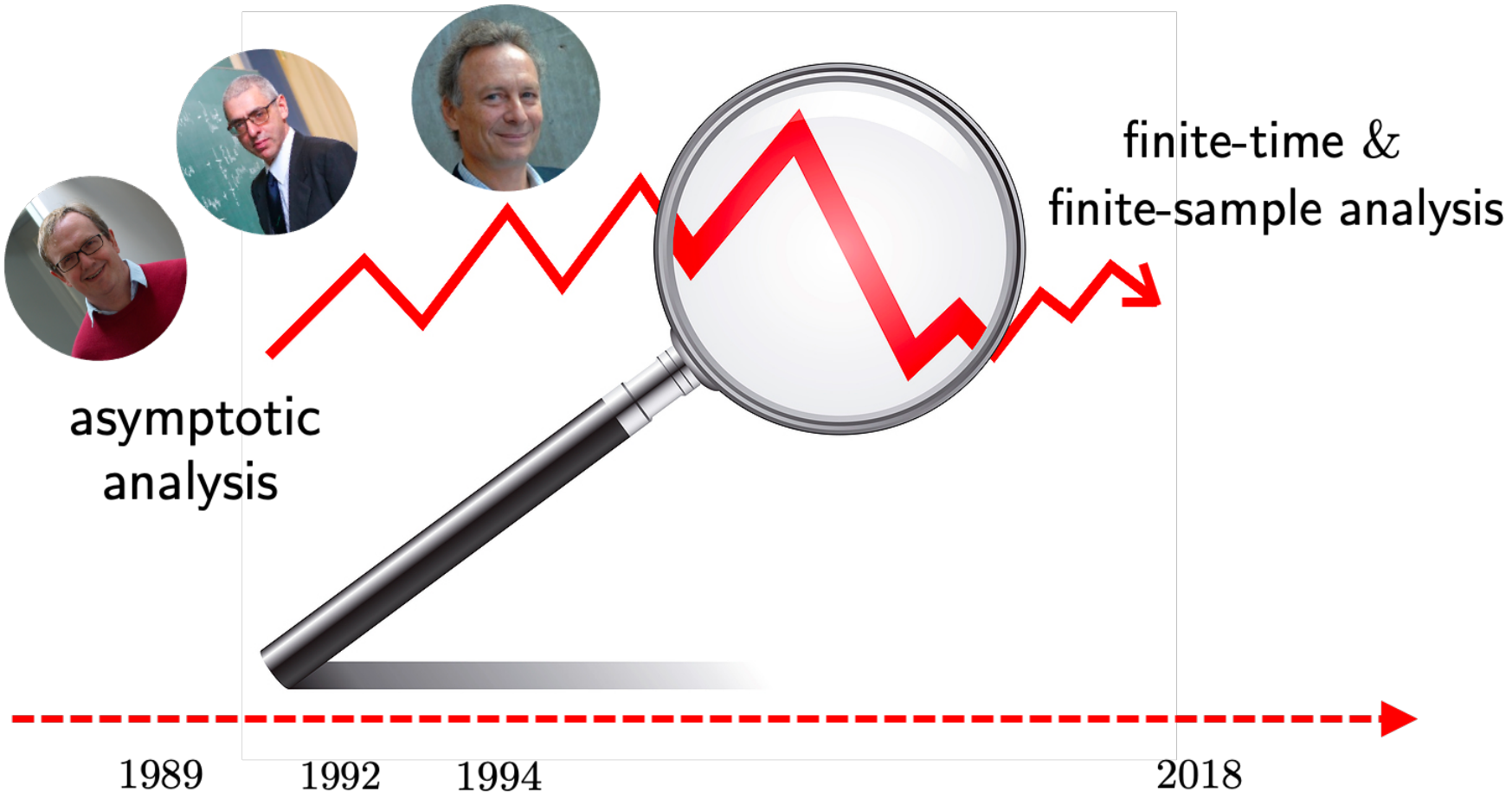
Sample efficiency

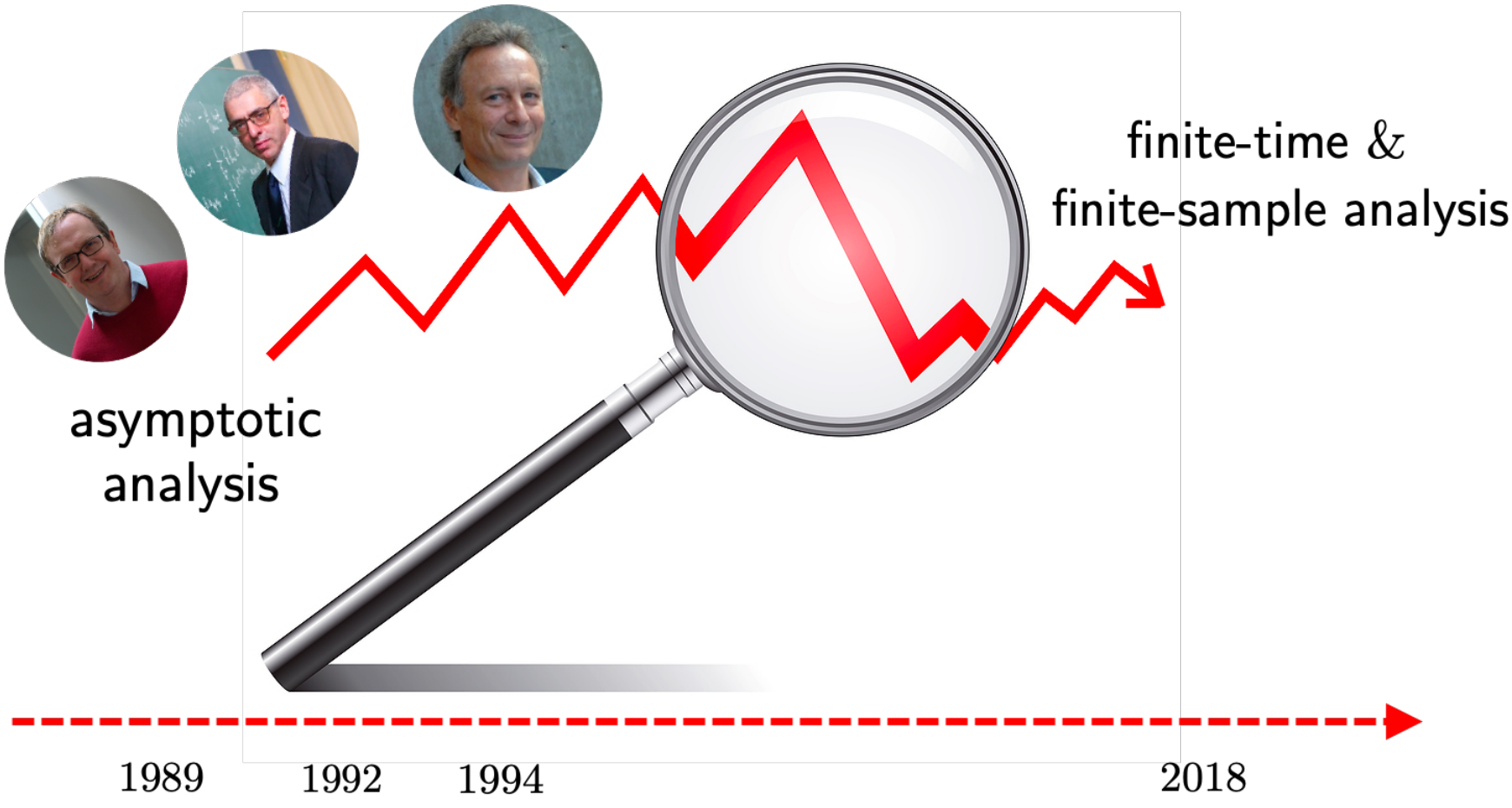
Collecting data samples might be expensive or time-consuming

clinical trials

online ads

Calls for in-depth understanding about sample efficiency of RL algorithms

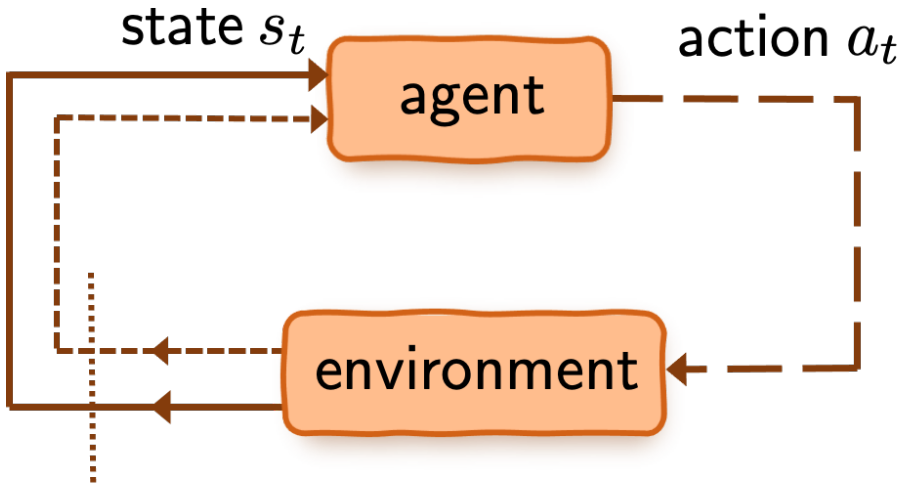

asymptotic
analysis
finite-time &
finite-sample analysis
1989
1992
1994
2018



This talk: a classical example — **Q-learning**

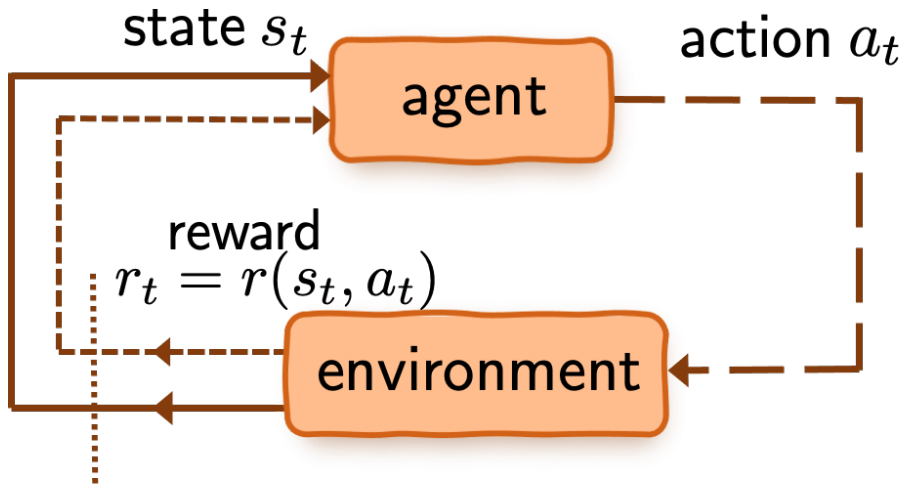
Background: Markov decision processes

Markov decision process (MDP)



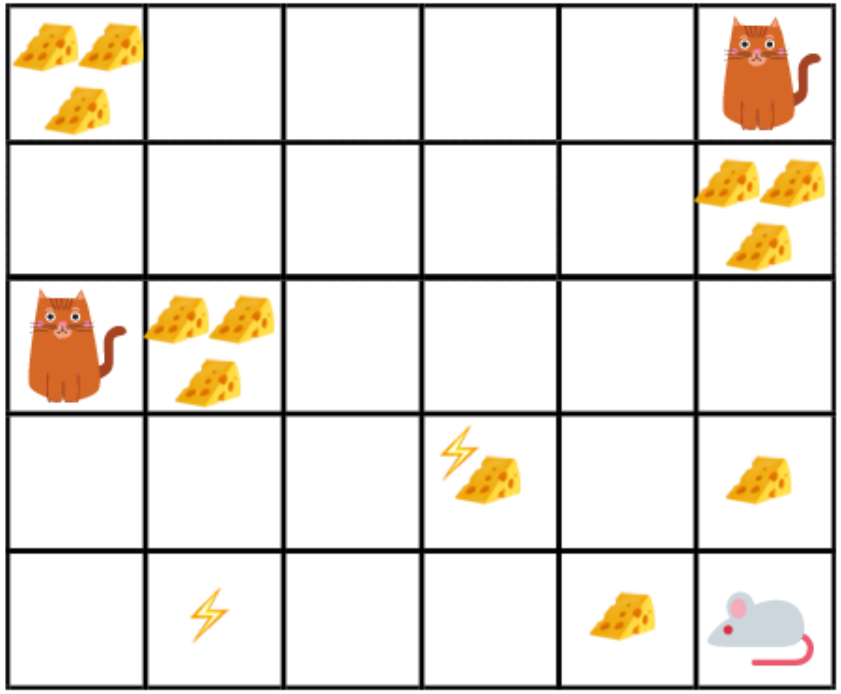
- $\mathcal{S}$: state space
- $\mathcal{A}$: action space

Markov decision process (MDP)



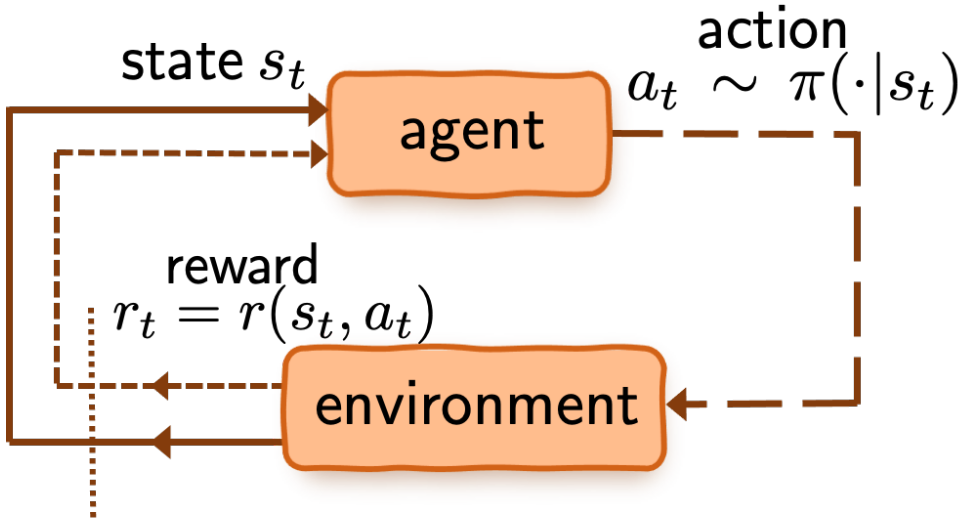
- $\mathcal{S}$: state space
- $\mathcal{A}$: action space
- $r(s, a) \in [0, 1]$: immediate reward

Markov decision process (MDP)



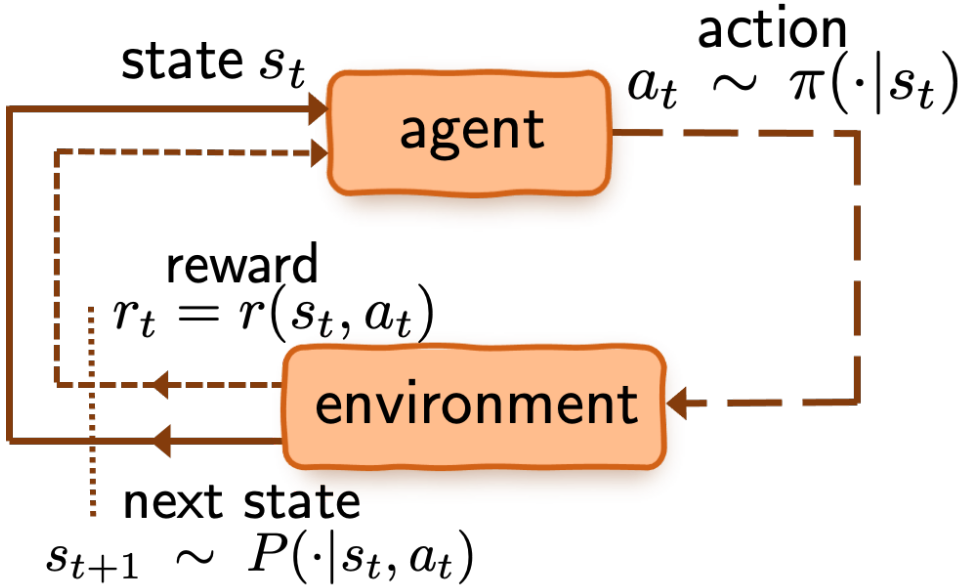
- state space $\mathcal{S}$: positions in the maze
- action space $\mathcal{A}$: up, down, left, right
- immediate reward $r(s, a)$: cheese, electricity shocks, cats

Markov decision process (MDP)



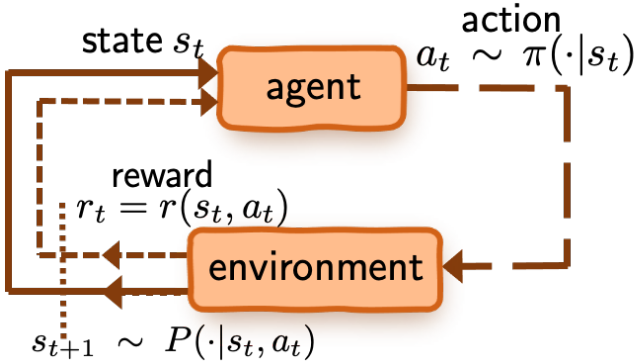
- $\mathcal{S}$: state space
- $\mathcal{A}$: action space
- $r(s, a) \in [0, 1]$: immediate reward
- $\pi(\cdot|s)$: policy (or action selection rule)

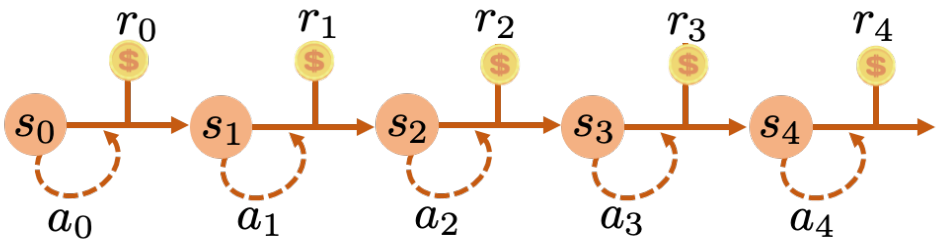
Markov decision process (MDP)



- $\mathcal{S}$: state space
- $\mathcal{A}$: action space
- $r(s, a) \in [0, 1]$: immediate reward
- $\pi(\cdot|s)$: policy (or action selection rule)
- $P(\cdot|s, a)$: unknown transition probabilities

Value function

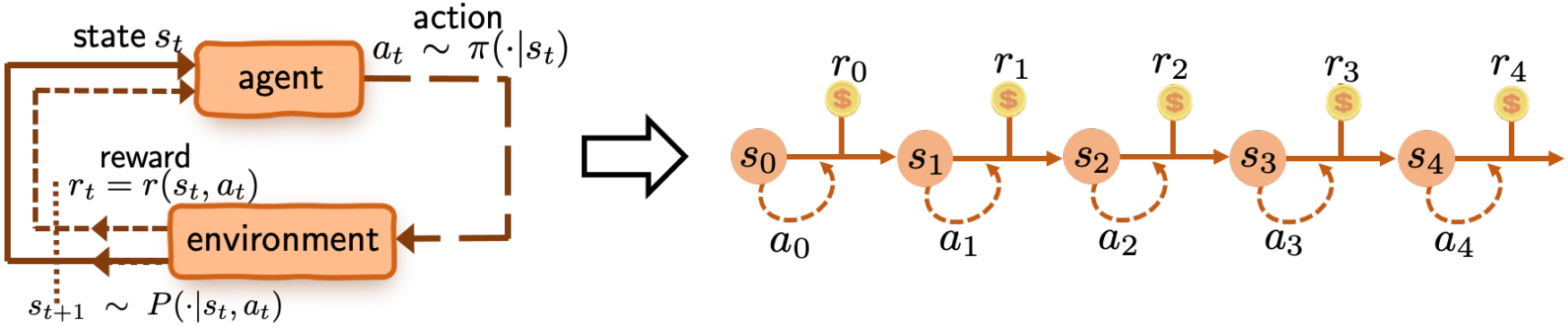




Value of policy π: long-term *discounted* reward

$$\forall s \in \mathcal{S}: \quad V^{\pi}(s) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \,\middle|\, s_0 = s\right]$$

Value function

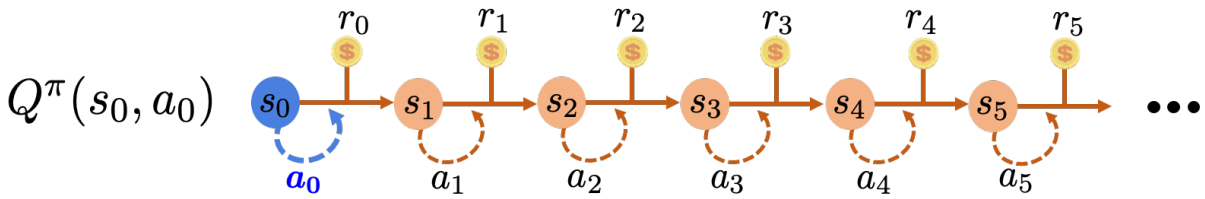


Value of policy π: long-term *discounted* reward

$$\forall s \in \mathcal{S}: \quad V^{\pi}(s) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \,\middle|\, s_0 = s\right]$$

- $\gamma \in [0, 1)$: discount factor
- $(a_0, s_1, a_1, s_2, a_2, \cdots)$: generated under policy π

Action-value function (a.k.a. Q-function)

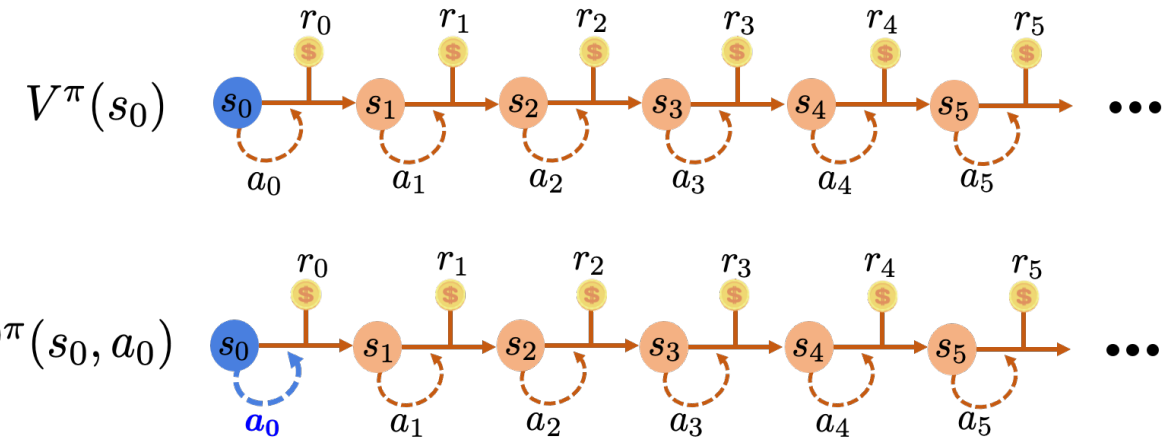


Q-function of policy π

$$\forall (s,a) \in \mathcal{S} \times \mathcal{A}: \quad Q^\pi(s,a) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \,\middle|\, s_0 = s, a_0 = a\right]$$

- $(\cancel{a_0}, s_1, a_1, s_2, a_2, \cdots)$: generated under policy π

Action-value function (a.k.a. Q-function)

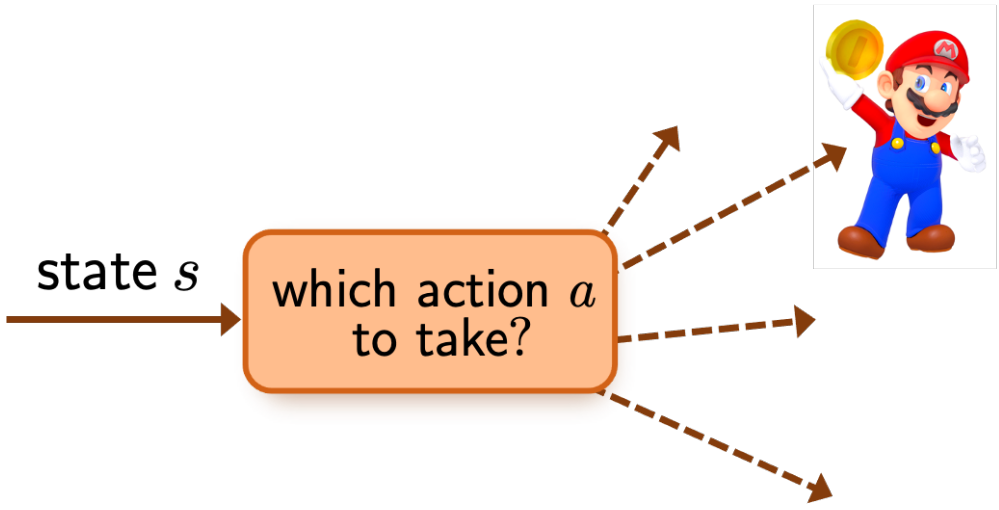


Q-function of policy π

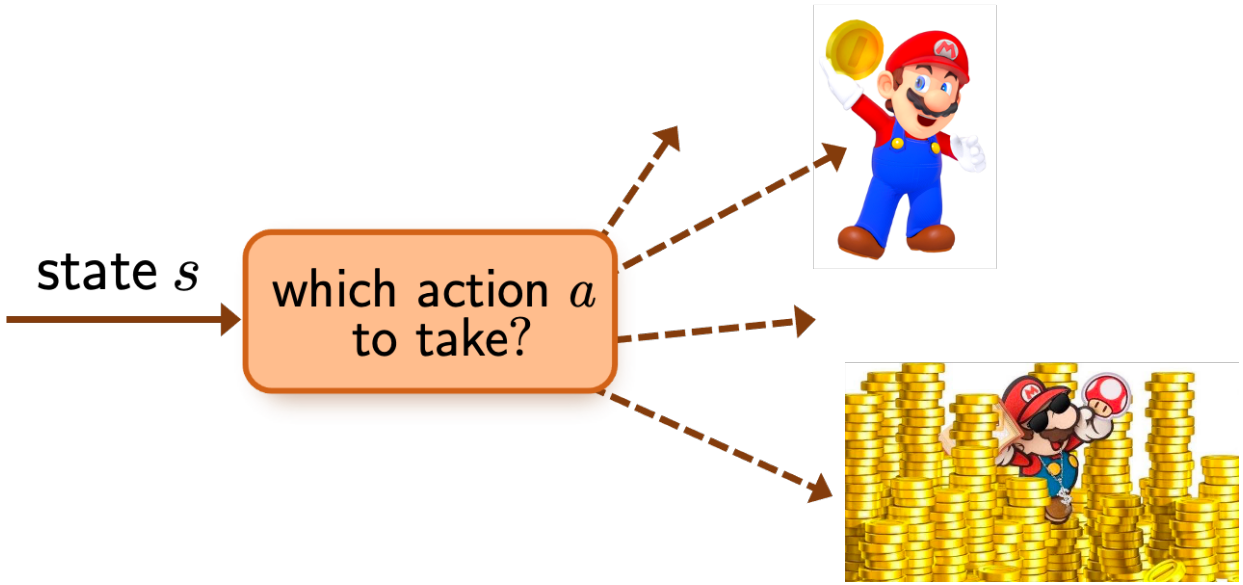
$$\forall (s,a) \in \mathcal{S} \times \mathcal{A}: \quad Q^\pi(s,a) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \,\middle|\, s_0 = s, a_0 = a\right]$$

- $(\cancel{a_0}, s_1, a_1, s_2, a_2, \cdots)$: generated under policy π

Optimal policy and optimal value

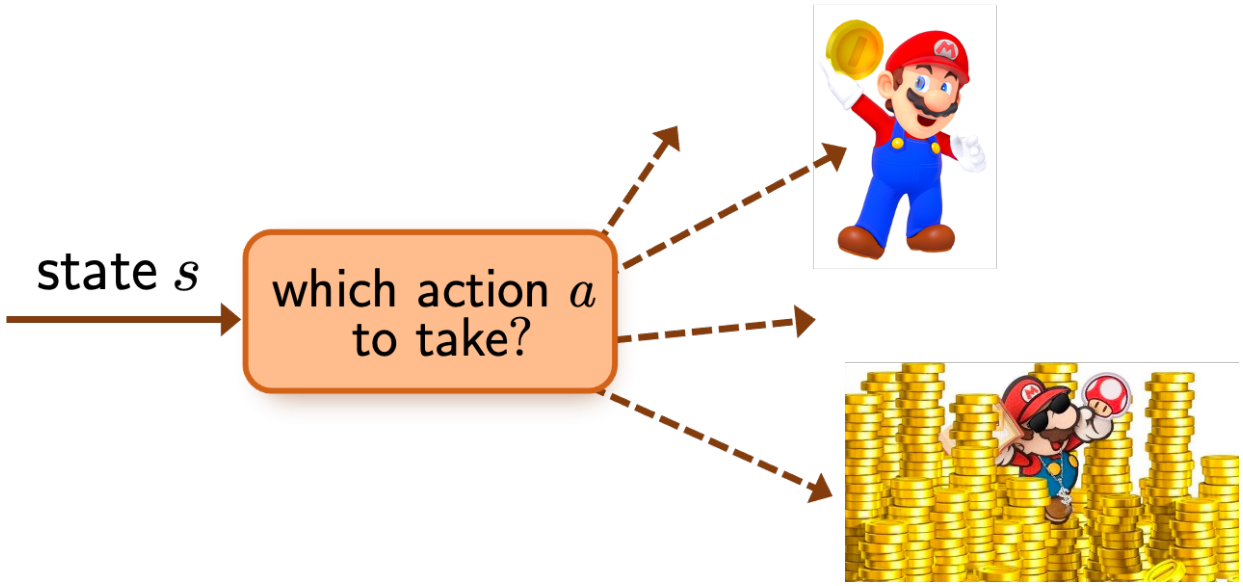


Optimal policy and optimal value

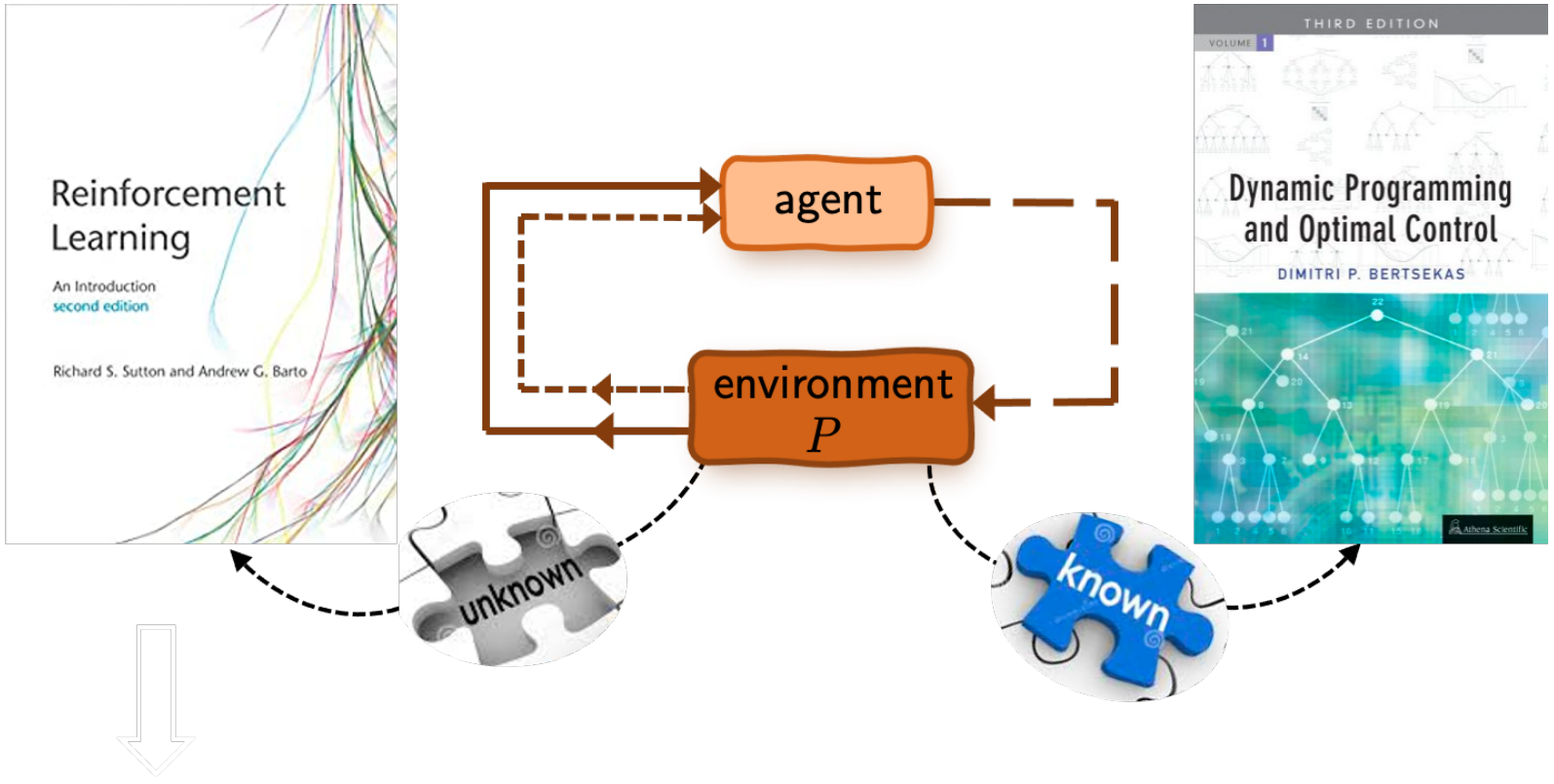


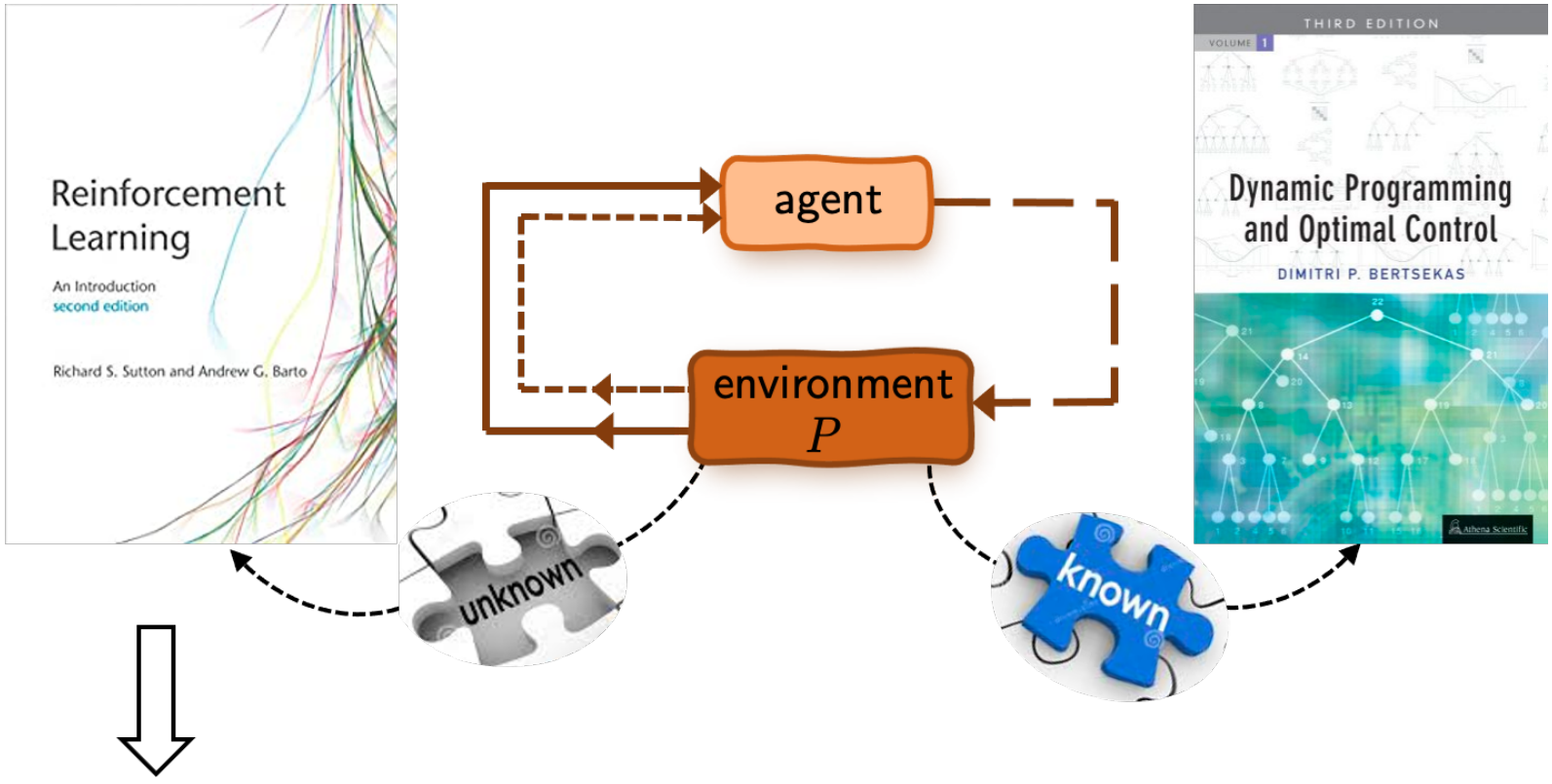
- **optimal policy** $\pi^\star$: maximizing value

Optimal policy and optimal value



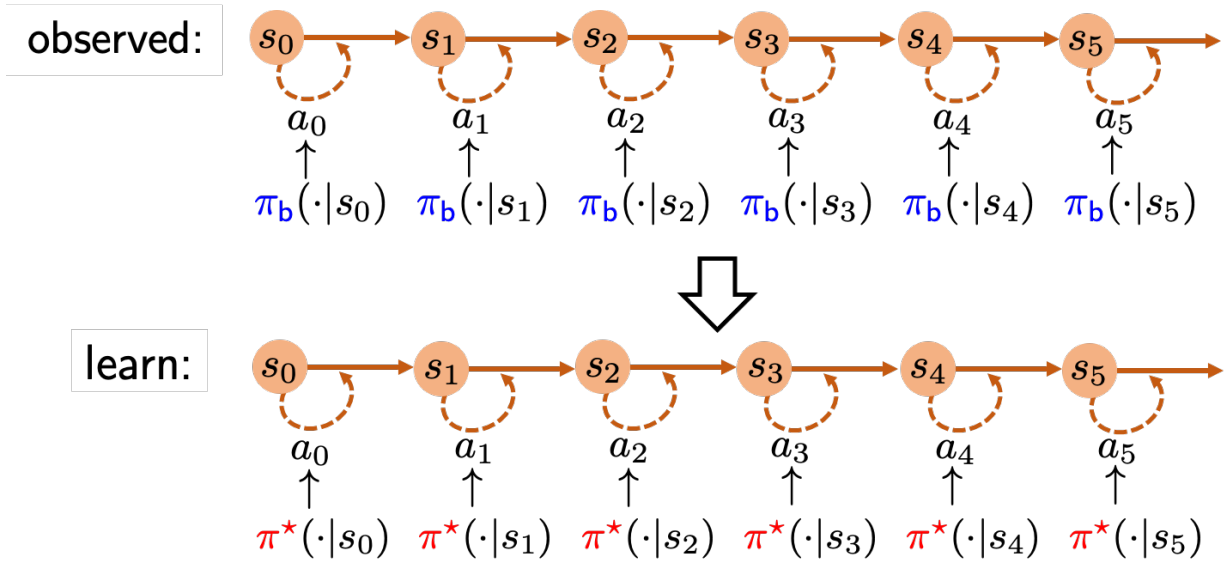
- **optimal policy** $\pi^\star$: maximizing value
- optimal value / Q function: $V^\star := V^{\pi^\star}$, $Q^\star := Q^{\pi^\star}$


Reinforcement Learning
An Introduction
second edition
Richard S. Sutton and Andrew G. Barto
agent
environment
P
unknown
known
THIRD EDITION
VOLUME 1
Dynamic Programming and Optimal Control
DIMITRI P. BERTSEKAS
Athena Scientific



Need to learn optimal value / policy from data samples

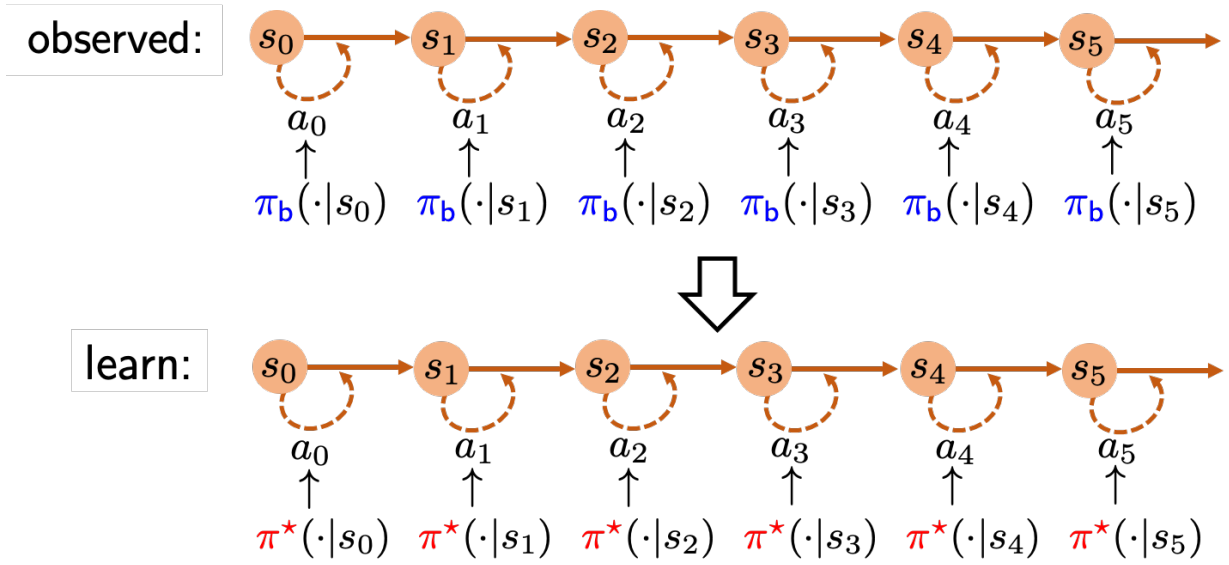
Markovian samples and behavior policy



Observed: $\underbrace{\{s_t, a_t, r_t\}_{t\geq 0}}_{\text{Markovian trajectory}}$ generated by behavior policy π_b

Goal: learn optimal value $V^\star$ and $Q^\star$ based on sample trajectory

Markovian samples and behavior policy



Key quantities of sample trajectory

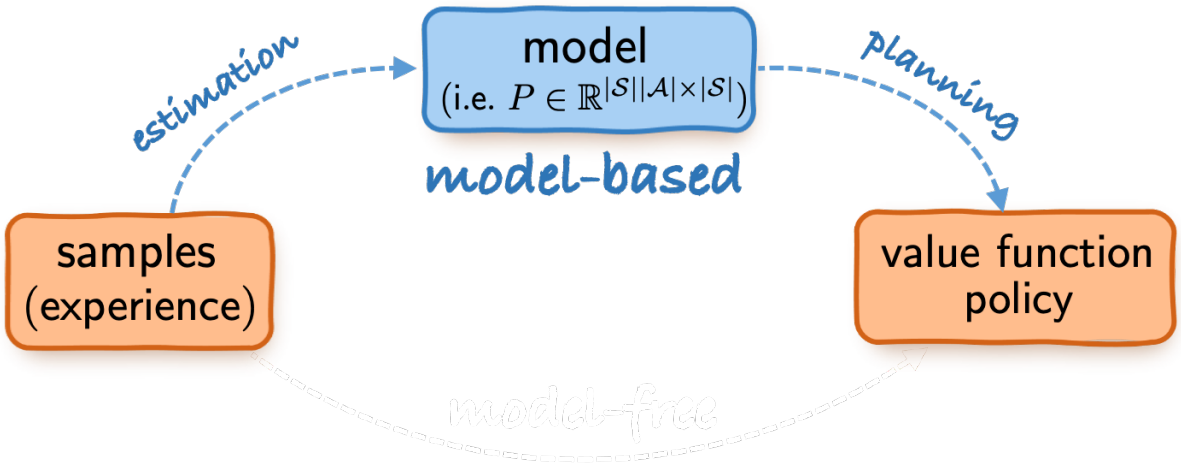
- minimum state-action occupancy probability

$$\mu_{\mathsf{min}} := \min \underbrace{\mu_{\pi_b}(s,a)}_{\text{stationary distribution}}$$

- mixing time: t_{mix}

Asynchronous Q-learning (on Markovian samples)

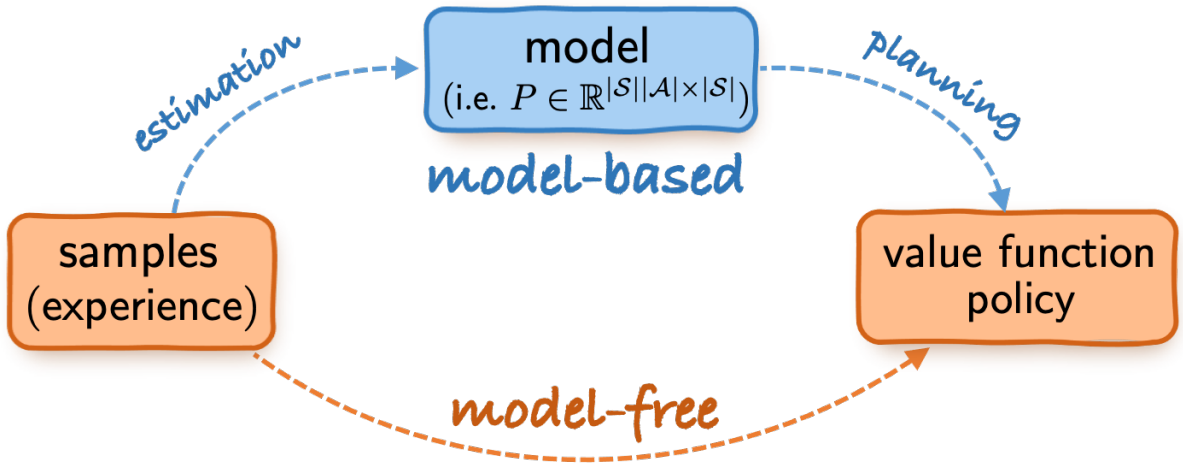
Model-based vs. model-free RL



Model-based approach ("plug-in")

1. build empirical estimate $\widehat{P}$ for P
2. planning based on empirical $\widehat{P}$

Model-based vs. model-free RL



Model-based approach ("plug-in")

1. build empirical estimate $\widehat{P}$ for P
2. planning based on empirical $\widehat{P}$

Model-free approach

— learning w/o modeling & estimating environment explicitly

Q-learning: a classical model-free algorithm

Chris Watkins

Peter Dayan

Stochastic approximation for solving **Bellman equation** $Q = \mathcal{T}(Q)$

Robbins & Monro '51

Aside: Bellman optimality principle

Bellman operator

$$\mathcal{T}(Q)(s,a) := \underbrace{r(s,a)}_{\text{immediate reward}} + \gamma \mathop{\mathbb{E}}_{s' \sim P(\cdot|s,a)} \Big[\underbrace{\max_{a' \in \mathcal{A}} Q(s',a')}_{\text{next state's value}} \Big]$$

- one-step look-ahead

Aside: Bellman optimality principle

Bellman operator

$$\mathcal{T}(Q)(s,a) := \underbrace{r(s,a)}_{\text{immediate reward}} + \gamma \mathop{\mathbb{E}}_{s' \sim P(\cdot|s,a)} \Big[\underbrace{\max_{a' \in \mathcal{A}} Q(s',a')}_{\text{next state's value}} \Big]$$

- one-step look-ahead

Bellman equation: $Q^\star$ is *unique* solution to

$$\mathcal{T}(Q^\star) = Q^\star$$

Richard Bellman

Q-learning: a classical model-free algorithm

Chris Watkins

Peter Dayan

Stochastic approximation for solving Bellman equation $Q = \mathcal{T}(Q)$

$$\underbrace{Q_{t+1}(s_t, a_t) = (1 - \eta_t) Q_t(s_t, a_t) + \eta_t \mathcal{T}_t(Q_t)(s_t, a_t)}_{\textit{only} \text{ update } (s_t, a_t)\text{-th entry}}, \quad t \geq 0$$

Q-learning: a classical model-free algorithm

Chris Watkins

Peter Dayan

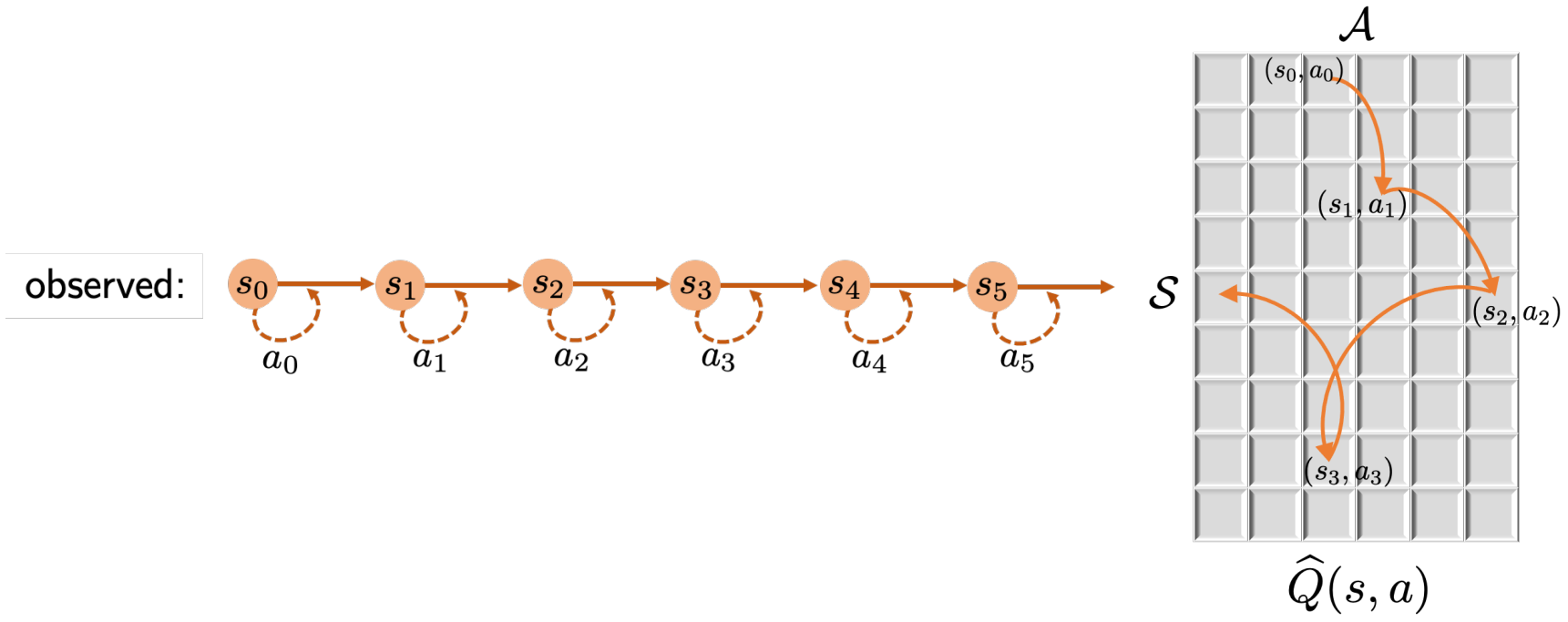
Stochastic approximation for solving Bellman equation $Q = \mathcal{T}(Q)$

$$\underbrace{Q_{t+1}(s_t, a_t) = (1 - \eta_t) Q_t(s_t, a_t) + \eta_t \mathcal{T}_t(Q_t)(s_t, a_t)}_{\text{only update } (s_t, a_t)\text{-th entry}}, \quad t \geq 0$$

$$\mathcal{T}_t(Q)(s_t, a_t) = r(s_t, a_t) + \gamma \max_{a'} Q(s_{t+1}, a')$$

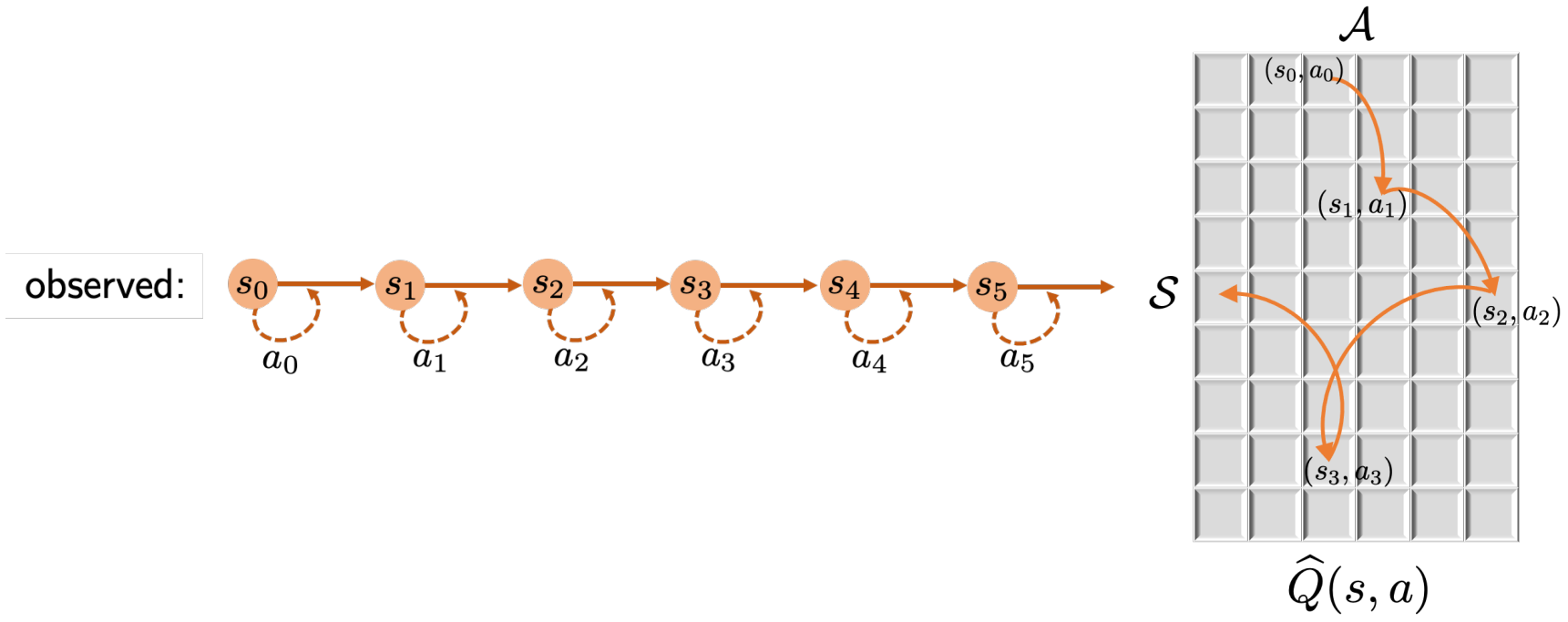
$$\mathcal{T}(Q)(s, a) = r(s, a) + \gamma \mathop{\mathbb{E}}_{s' \sim P(\cdot|s,a)} \Big[\max_{a'} Q(s', a') \Big]$$

Q-learning on Markovian samples



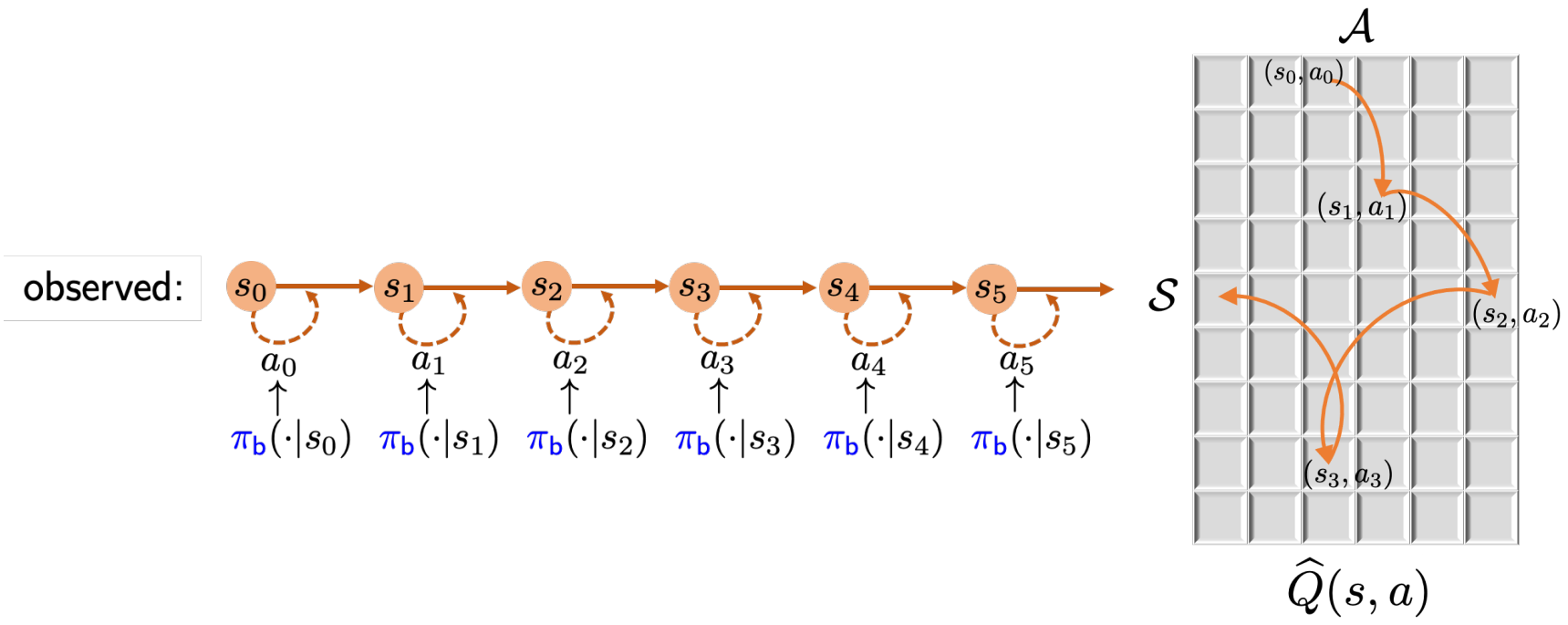
- **asynchronous:** only a single entry is updated each iteration

Q-learning on Markovian samples



- **asynchronous:** only a single entry is updated each iteration
  - resembles Markov-chain *coordinate descent*

Q-learning on Markovian samples



- **asynchronous:** only a single entry is updated each iteration
  - resembles Markov-chain *coordinate descent*
- **off-policy:** target policy $\pi^\star \neq$ behavior policy π_b

A highly incomplete list of prior work

- Watkins, Dayan '92
- Tsitsiklis '94
- Jaakkola, Jordan, Singh '94
- Szepesvári '98
- Kearns, Singh '99
- Borkar, Meyn '00
- Even-Dar, Mansour '03
- Beck, Srikant '12
- Chi, Zhu, Bubeck, Jordan '18
- Shah, Xie '18
- Lee, He '18
- Wainwright '19
- Chen, Zhang, Doan, Maguluri, Clarke '19
- Yang, Wang '19
- Du, Lee, Mahajan, Wang '20
- Chen, Maguluri, Shakkottai, Shanmugam '20
- Qu, Wierman '20
- Devraj, Meyn '20
- Weng, Gupta, He, Ying, Srikant '20
- ...

What is sample complexity of (async) Q-learning?

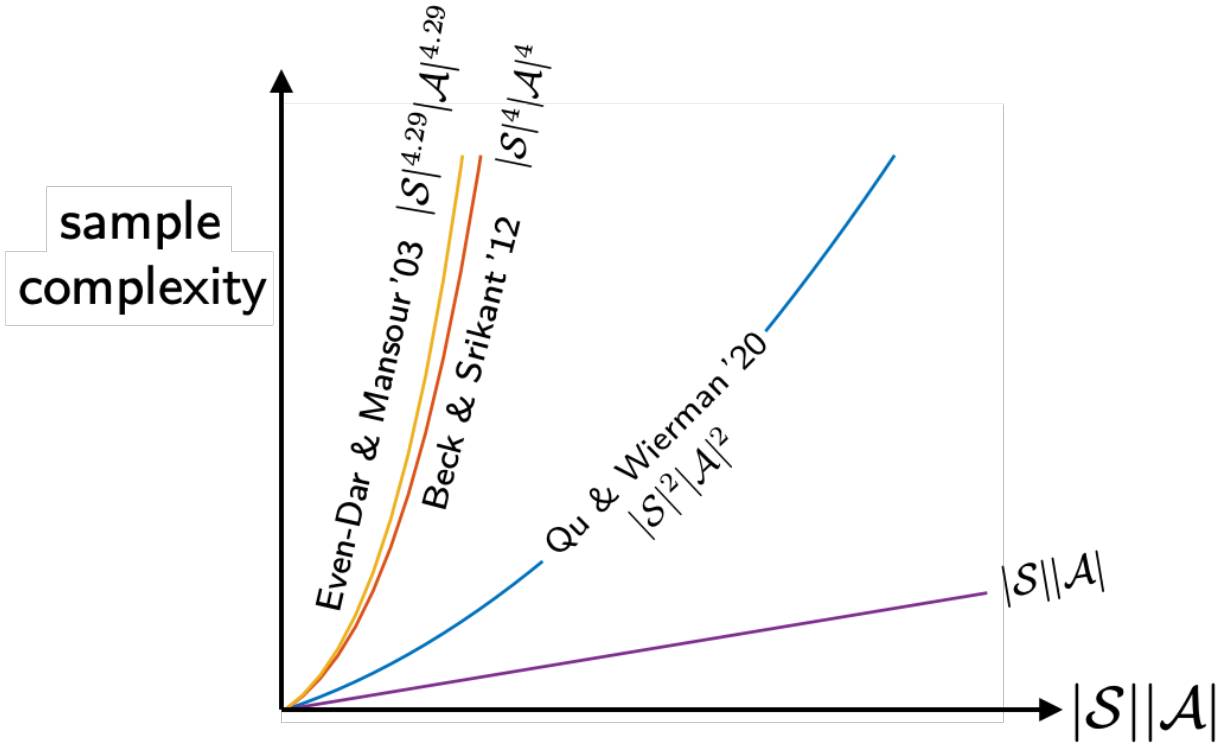
Prior art: async Q-learning

Question: how many samples are needed to ensure $\|\widehat{Q} - Q^\star\|_\infty \leq \varepsilon$?

paper	sample complexity	learning rate
Even-Dar & Mansour '03	$\frac{(t_{\mathsf{cover}})^{\frac{1}{1-\gamma}}}{(1-\gamma)^4\varepsilon^2}$	linear: $\frac{1}{t}$
Even-Dar & Mansour '03	$\left(\frac{t_{\mathsf{cover}}^{1+3\omega}}{(1-\gamma)^4\varepsilon^2}\right)^{\frac{1}{\omega}} + \left(\frac{t_{\mathsf{cover}}}{1-\gamma}\right)^{\frac{1}{1-\omega}}$	poly: $\frac{1}{t^\omega}$, $\omega \in (\frac{1}{2}, 1)$
Beck & Srikant '12	$\frac{t_{\mathsf{cover}}^3 \lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert}{(1-\gamma)^5\varepsilon^2}$	constant
Qu & Wierman '20	$\frac{t_{\mathsf{mix}}}{\mu_{\mathsf{min}}^2(1-\gamma)^5\varepsilon^2}$	rescaled linear

Prior art: async Q-learning

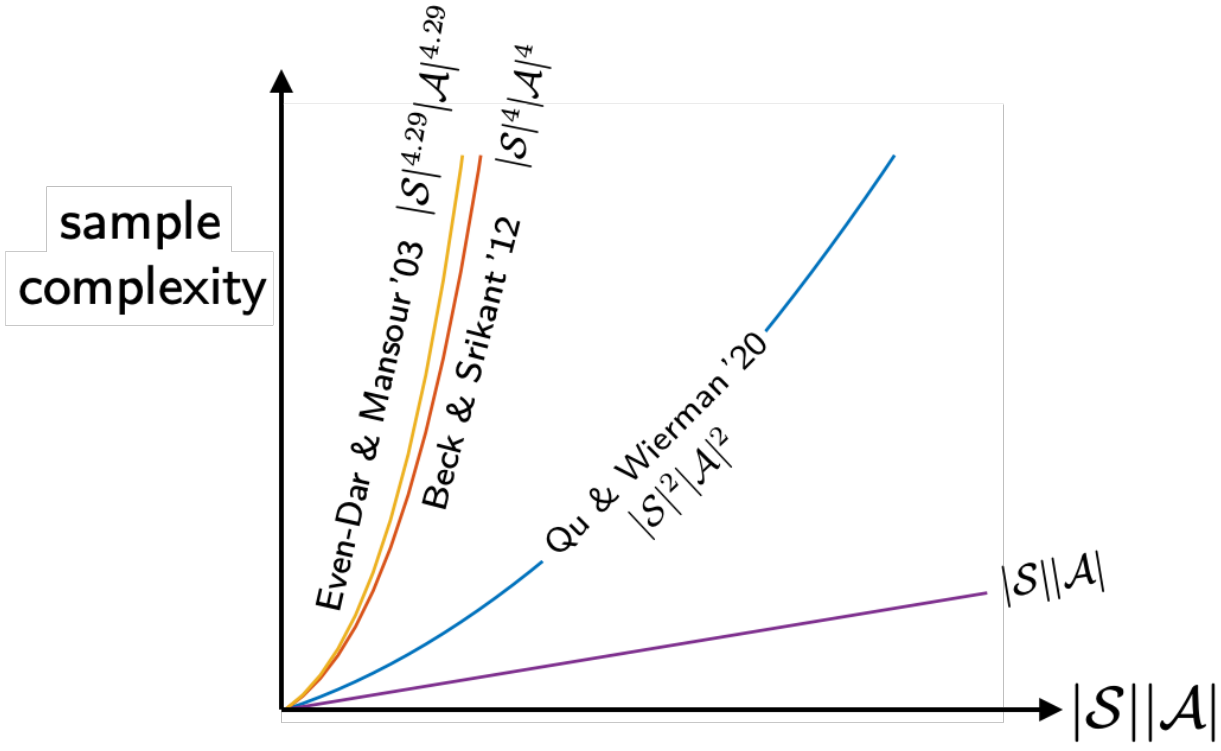
Question: how many samples are needed to ensure $\|\widehat{Q} - Q^\star\|_\infty \leq \varepsilon$?



if we take $\mu_{\min} \asymp \frac{1}{|\mathcal{S}||\mathcal{A}|}$, $t_{\text{cover}} \asymp \frac{t_{\text{mix}}}{\mu_{\min}}$

Prior art: async Q-learning

Question: how many samples are needed to ensure $\|\widehat{Q} - Q^\star\|_\infty \leq \varepsilon$?

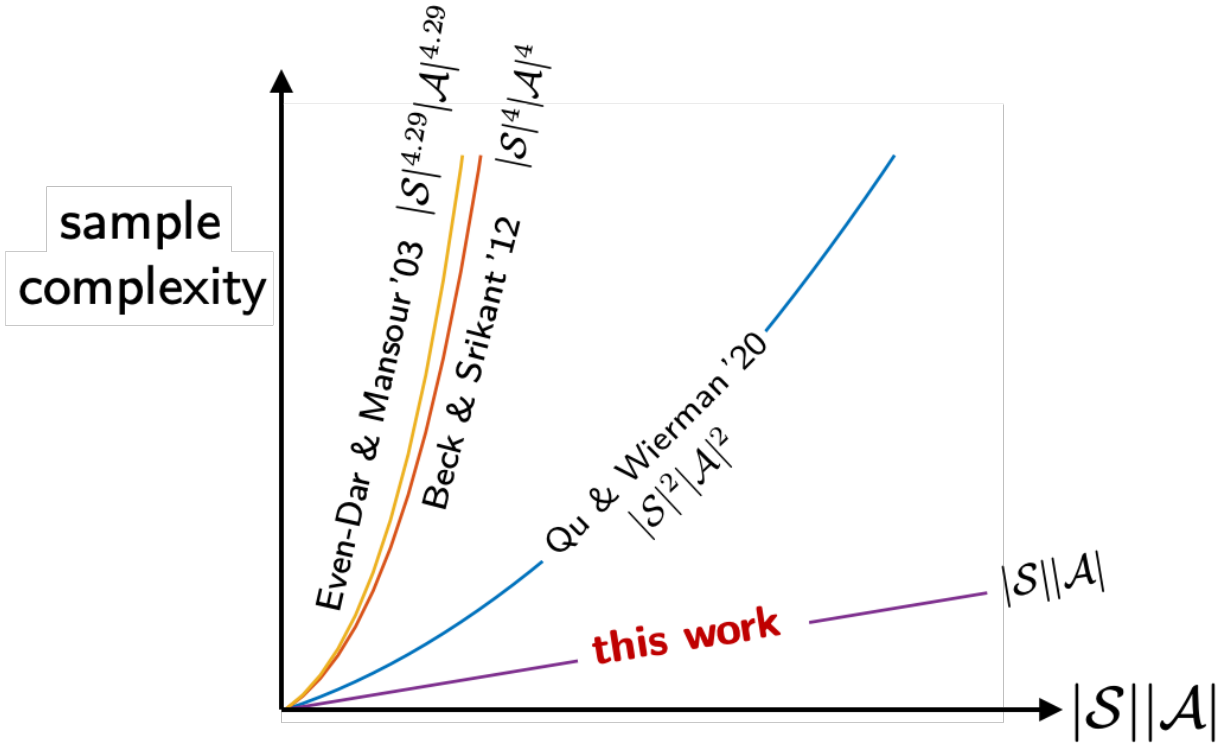


if we take $\mu_{\min} \asymp \frac{1}{|\mathcal{S}||\mathcal{A}|}$, $t_{\text{cover}} \asymp \frac{t_{\text{mix}}}{\mu_{\min}}$

All prior results require sample size of at least $t_{\text{mix}}|\mathcal{S}|^2|\mathcal{A}|^2$!

Prior art: async Q-learning

Question: how many samples are needed to ensure $\|\widehat{Q} - Q^\star\|_\infty \leq \varepsilon$?



if we take $\mu_{\min} \asymp \frac{1}{|\mathcal{S}||\mathcal{A}|}$, $t_{\text{cover}} \asymp \frac{t_{\text{mix}}}{\mu_{\min}}$

All prior results require sample size of at least $t_{\text{mix}}|\mathcal{S}|^2|\mathcal{A}|^2$!

Main result: ℓ_∞-based sample complexity

Theorem 1 (Li, Wei, Chi, Gu, Chen '20)

For any $0 < \varepsilon \leq \frac{1}{1-\gamma}$, *sample complexity of async Q-learning to yield* $\|\widehat{Q} - Q^\star\|_\infty \leq \varepsilon$ *is at most (up to some log factor)*

$$\frac{1}{\mu_{\mathsf{min}}(1-\gamma)^5\varepsilon^2} + \frac{t_{\mathsf{mix}}}{\mu_{\mathsf{min}}(1-\gamma)}$$

Main result: ℓ_∞-based sample complexity

Theorem 1 (Li, Wei, Chi, Gu, Chen '20)

For any $0 < \varepsilon \leq \frac{1}{1-\gamma}$, *sample complexity of async Q-learning to yield* $\|\widehat{Q} - Q^\star\|_\infty \leq \varepsilon$ *is at most (up to some log factor)*

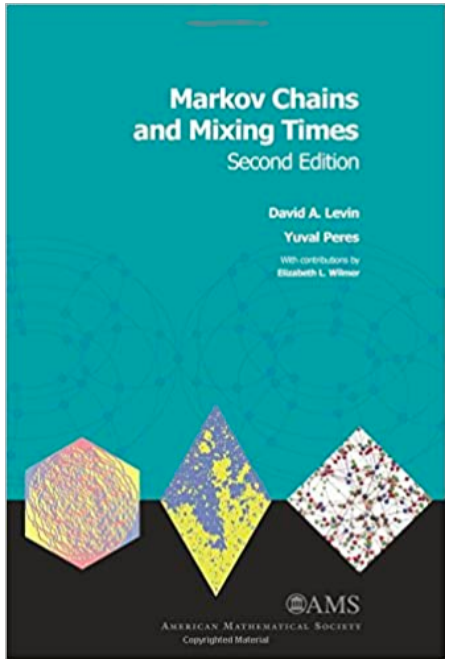
$$\frac{1}{\mu_{\mathsf{min}}(1-\gamma)^5\varepsilon^2} + \frac{t_{\mathsf{mix}}}{\mu_{\mathsf{min}}(1-\gamma)}$$

- Improves upon prior art by **at least** $|\mathcal{S}||\mathcal{A}|$!

— *prior art:* $\frac{t_{\mathsf{mix}}}{\mu_{\mathsf{min}}^2(1-\gamma)^5\varepsilon^2}$ (Qu & Wierman '20)

Effect of mixing time on sample complexity

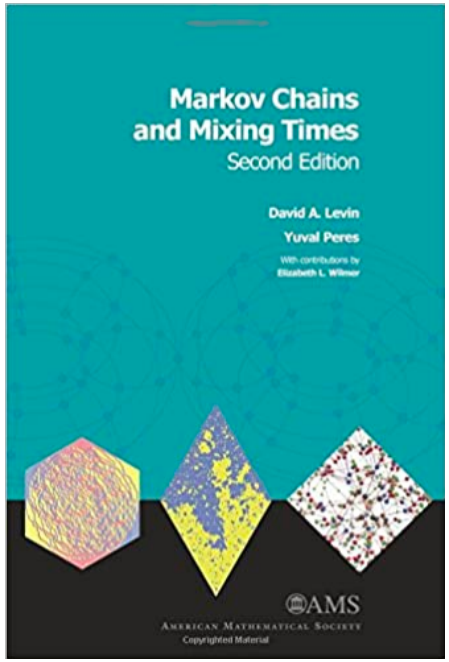
$$\frac{1}{\mu_{\mathsf{min}}(1-\gamma)^5\varepsilon^2} + \frac{t_{\mathsf{mix}}}{\mu_{\mathsf{min}}(1-\gamma)}$$



- reflects cost taken to reach steady state
- one-time expense (almost independent of ε)
  — it becomes amortized as algorithm runs

Effect of mixing time on sample complexity

$$\frac{1}{\mu_{\mathsf{min}}(1-\gamma)^5\varepsilon^2} + \frac{t_{\mathsf{mix}}}{\mu_{\mathsf{min}}(1-\gamma)}$$



- reflects cost taken to reach steady state
- one-time expense (almost independent of ε)
  — it becomes amortized as algorithm runs

— *prior art:* $\frac{t_{\mathsf{mix}}}{\mu_{\mathsf{min}}^2(1-\gamma)^5\varepsilon^2}$ (Qu & Wierman '20)

Learning rates

Our choice: constant stepsize $\eta_t \equiv \min\left\{\frac{(1-\gamma)^4\varepsilon^2}{\gamma^2}, \frac{1}{t_{\mathsf{mix}}}\right\}$

- Qu & Wierman '20: rescaled linear $\eta_t = \frac{\frac{1}{\mu_{\min}(1-\gamma)}}{t+\max\{\frac{1}{\mu_{\min}(1-\gamma)}, t_{\mathsf{mix}}\}}$
- Beck & Srikant '12: constant $\underbrace{\eta_t \equiv \frac{(1-\gamma)^4\varepsilon^2}{|\mathcal{S}||\mathcal{A}|t_{\mathsf{cover}}^2}}_{\text{too conservative}}$
- Even-Dar & Mansour '03: polynomial $\eta_t = t^{-\omega}$ $(\omega \in (\frac{1}{2}, 1])$

Minimax lower bound

minimax lower bound
(Azar et al. '13)

$$\frac{1}{\mu_{\min}(1-\gamma)^3\varepsilon^2}$$

asyn Q-learning
(ignoring dependency on t_{mix})

$$\frac{1}{\mu_{\min}(1-\gamma)^5\varepsilon^2}$$

Minimax lower bound

minimax lower bound
(Azar et al. '13)

$$\frac{1}{\mu_{\min}(1-\gamma)^3\varepsilon^2}$$

asyn Q-learning
(ignoring dependency on t_{mix})

$$\frac{1}{\mu_{\min}(1-\gamma)^5\varepsilon^2}$$

Can we improve dependency on **discount complexity** $\frac{1}{1-\gamma}$?

One strategy: variance reduction

— inspired by Johnson & Zhang '13, Wainwright '19

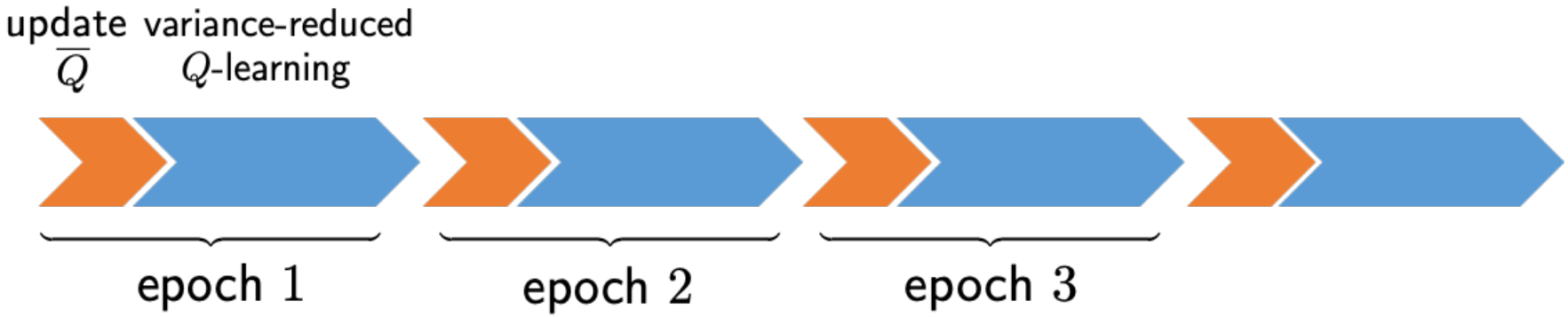
Variance-reduced Q-learning updates

$$Q_t(s_t, a_t) = (1-\eta) Q_{t-1}(s_t, a_t) + \eta \Big(\mathcal{T}_t(Q_{t-1}) \underbrace{- \mathcal{T}_t(\overline{Q}) + \widetilde{\mathcal{T}}(\overline{Q})}_{\text{use } \overline{Q} \text{ to help reduce variability}} \Big)(s_t, a_t)$$

- $\overline{Q}$: some reference Q-estimate
- $\widetilde{\mathcal{T}}$: empirical Bellman operator (using a batch of samples)

Variance-reduced Q-learning

— inspired by Johnson & Zhang '13, Wainwright '19



for each epoch

1. update $\overline{Q}$ and $\widetilde{T}(\overline{Q})$
2. run variance-reduced Q-learning updates

Main result: ℓ_∞-based sample complexity

Theorem 2 (Li, Wei, Chi, Gu, Chen '20)

For any $0 < \varepsilon \leq 1$, sample complexity for ***(async) variance-reduced Q-learning*** *to yield $\|\widehat{Q} - Q^\star\|_\infty \leq \varepsilon$ is at most on the order of*

$$\frac{1}{\mu_{\min}(1-\gamma)^3\varepsilon^2} + \frac{t_{\mathsf{mix}}}{\mu_{\min}(1-\gamma)}$$

- more aggressive learning rates: $\eta_t \equiv \min\left\{\frac{\cancel{(1-\gamma)^4}(1-\gamma)^2}{\gamma^2}, \frac{1}{t_{\mathsf{mix}}}\right\}$

Main result: ℓ_∞-based sample complexity

Theorem 2 (Li, Wei, Chi, Gu, Chen '20)

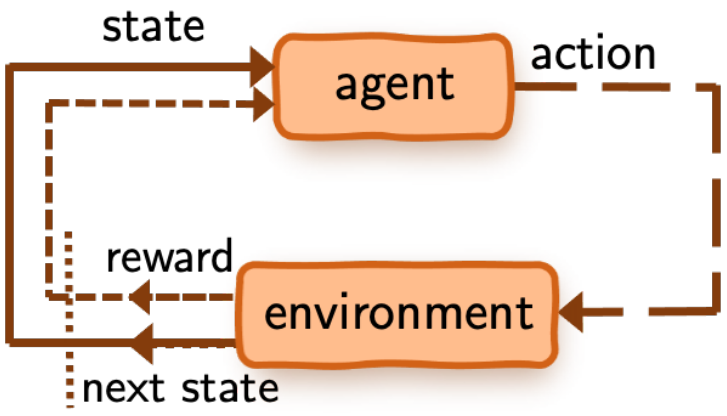
For any $0 < \varepsilon \leq 1$, sample complexity for ***(async) variance-reduced Q-learning*** *to yield $\|\widehat{Q} - Q^\star\|_\infty \leq \varepsilon$ is at most on the order of*

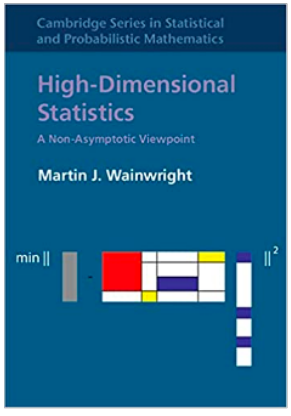
$$\frac{1}{\mu_{\min}(1-\gamma)^3\varepsilon^2} + \frac{t_{\mathsf{mix}}}{\mu_{\min}(1-\gamma)}$$

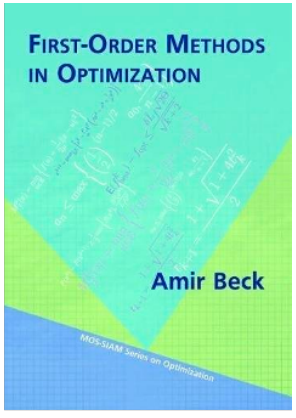
- more aggressive learning rates: $\eta_t \equiv \min\left\{\frac{(1-\gamma)^4(1-\gamma)^2}{\gamma^2}, \frac{1}{t_{\mathsf{mix}}}\right\}$
- minimax-optimal for $0 < \varepsilon \leq 1$

Concluding remarks

Understanding RL requires modern statistics and optimization

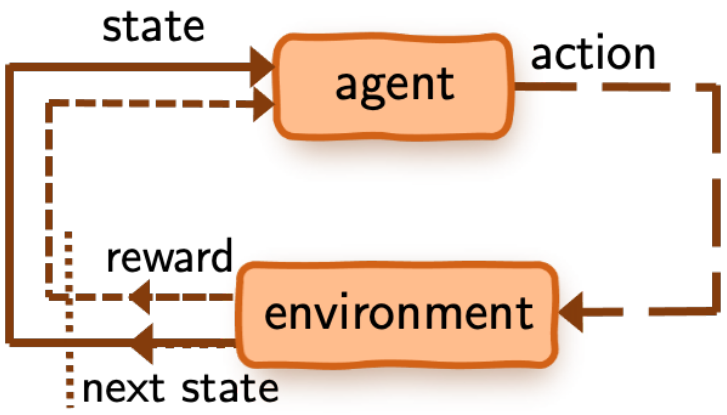


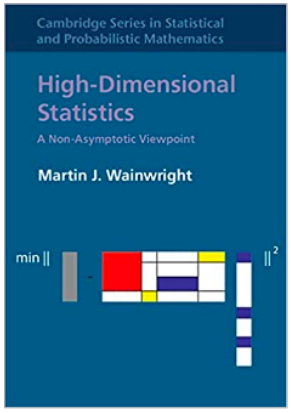


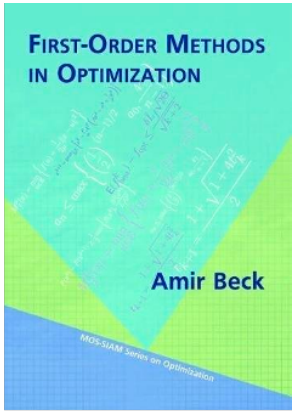


Concluding remarks

Understanding RL requires modern statistics and optimization







future directions

- function approximation
- finite-horizon episodic MDPs
- on-policy algorithms like SARSA
- general Markov-chain-based optimization algorithms

Paper:

"Sample complexity of asynchronous Q-learning: sharper analysis and variance reduction," G. Li, Y. Wei, Y. Chi, Y. Gu, Y. Chen, arxiv:2006.03041, NeurIPS 2020