Reinforcement Learning

Multi-Armed Bandits

David Abel, Michael Herrmann
Based on slides by Stefano V. Albrecht
17 January 2025

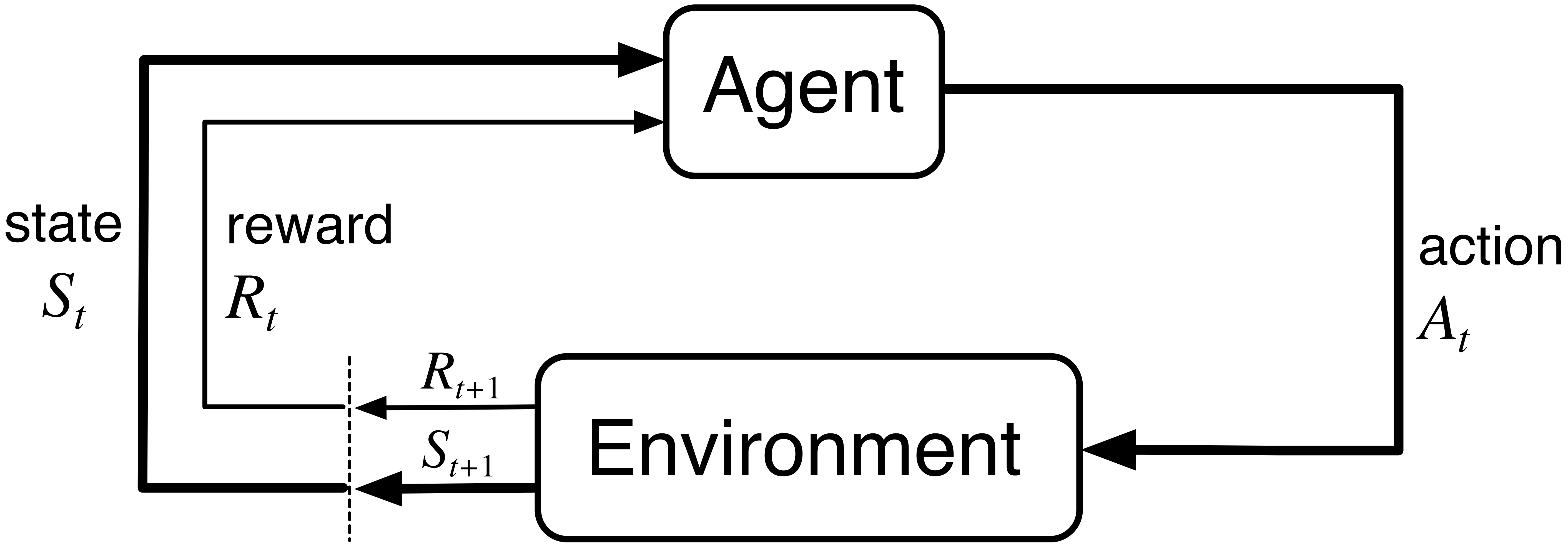
Lecture Outline

1. Recap: What is RL? A Demo
2. Simplest RL problem: Multi-armed bandits
3. Explore-exploit dilemma
4. Algorithms for multi-armed bandits: UCB

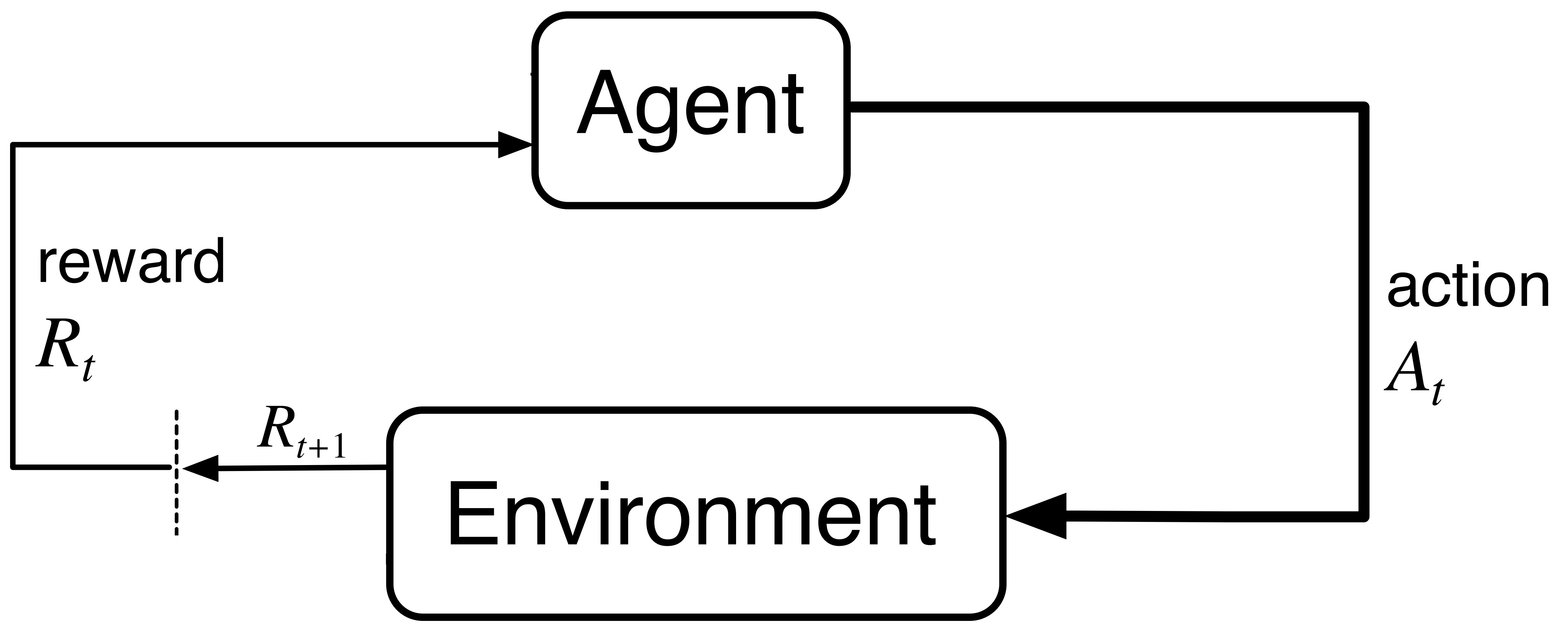
The Big Picture: What is RL?

Learning to act

The Reinforcement Learning Loop



Multi-Armed Bandit: RL Without State



Multi-Armed Bandits: Notation

- Random Variables: capital italics, such as

$$A, R, A_t, R_t$$

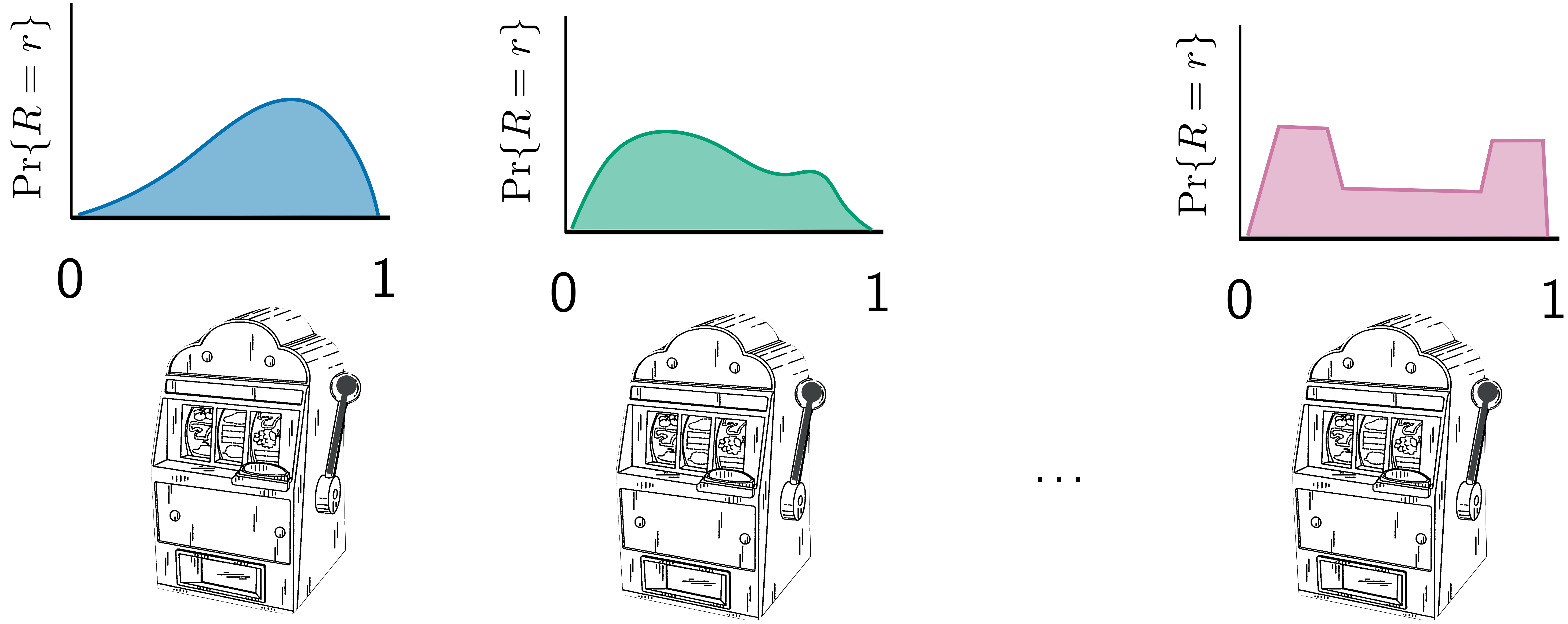
- Realisations of these variables: lower case, such as

$$a, r, a_t, r_t \qquad \Pr\{A_t = a_t\}$$

- Sets: script capitals, intervals, blackboard, such as

$$\mathcal{A}, [0, 1], \mathbb{N}$$

Multi-Armed Bandits



Formal Definition

Definition (Multi-Armed Bandit Problem):

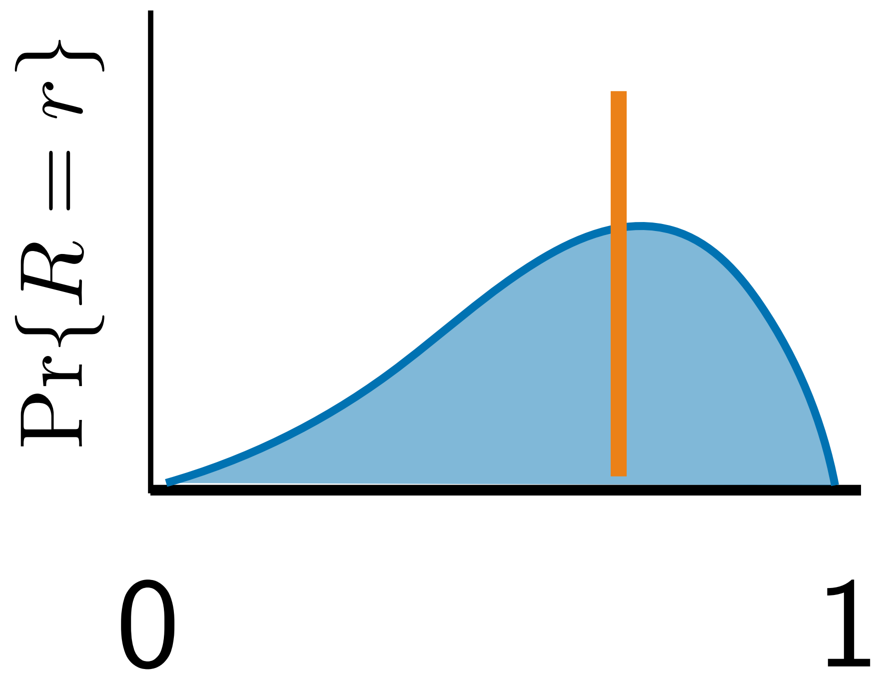
Given: a set of k actions, $\mathcal{A}$, number of rounds T.

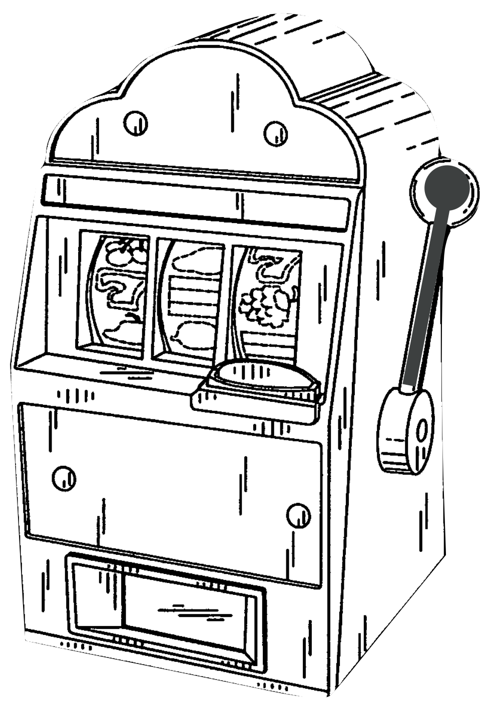
Repeat for t in T rounds:

1. Algorithm selects arm $A_t \in \mathcal{A}$
2. Algorithm observes reward $R_t \in [0, 1]$

Goal: maximise expected total reward.

Value: The Expected Reward

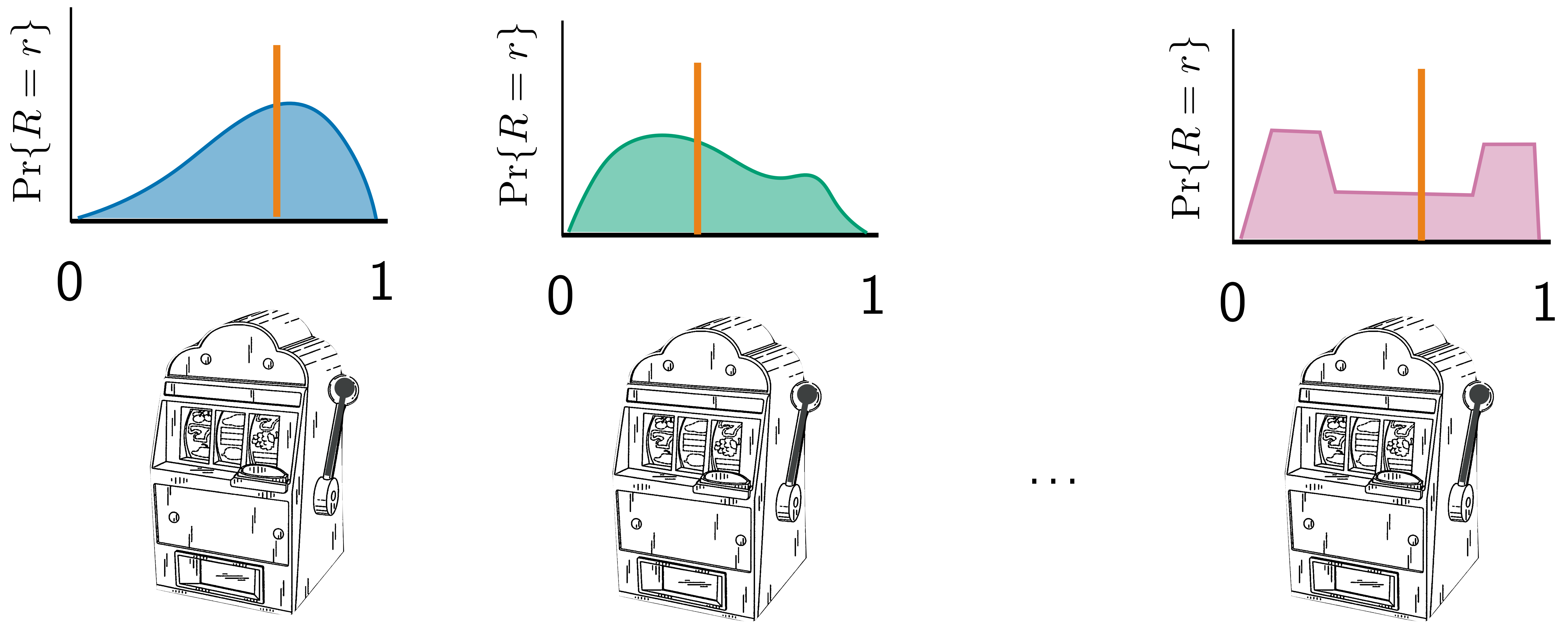




$$\boxed{q_*(a)} = \mathbb{E}[R_t \mid A_t = a]$$

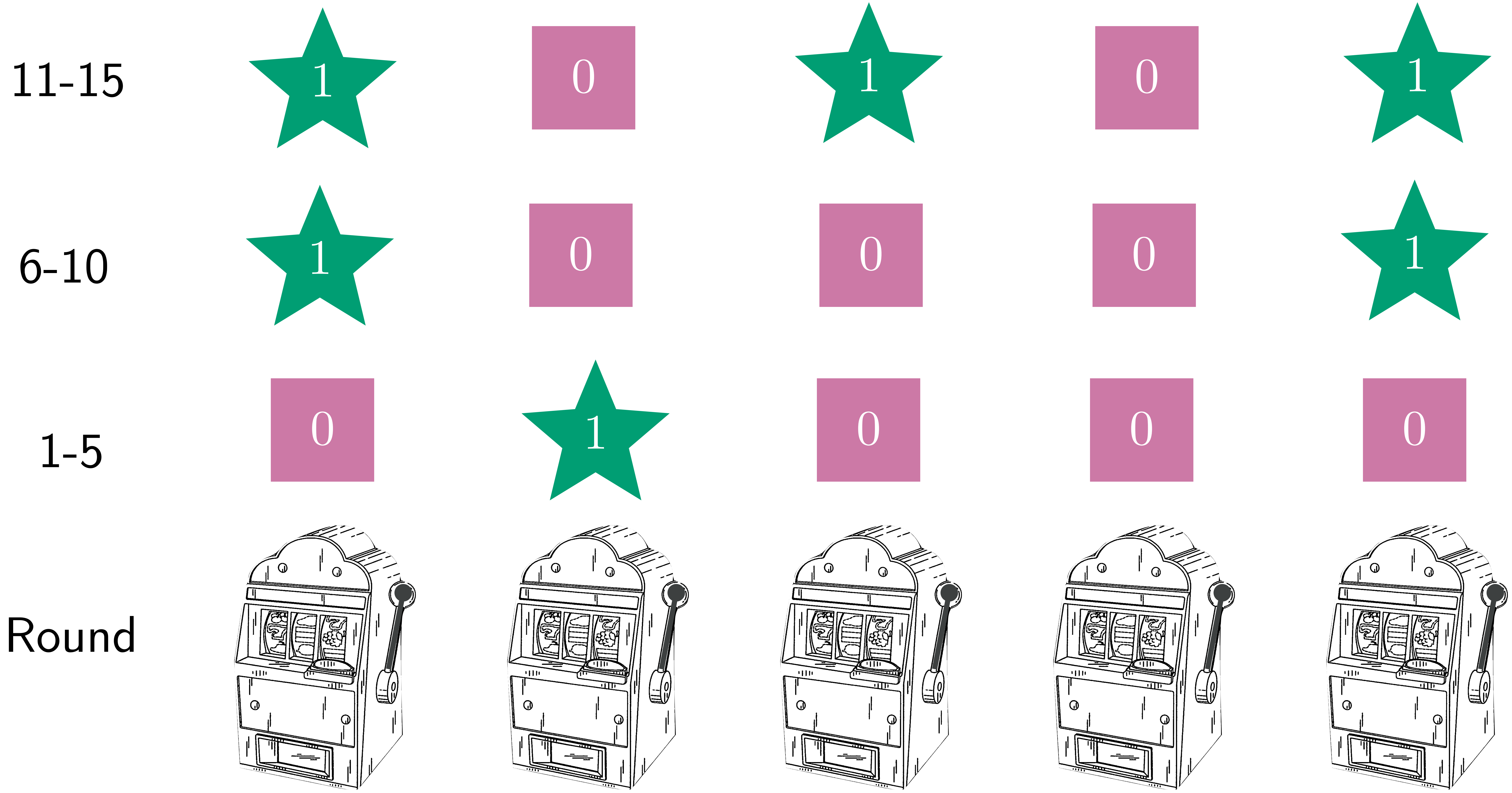
Value of arm

Multi-Armed Bandits

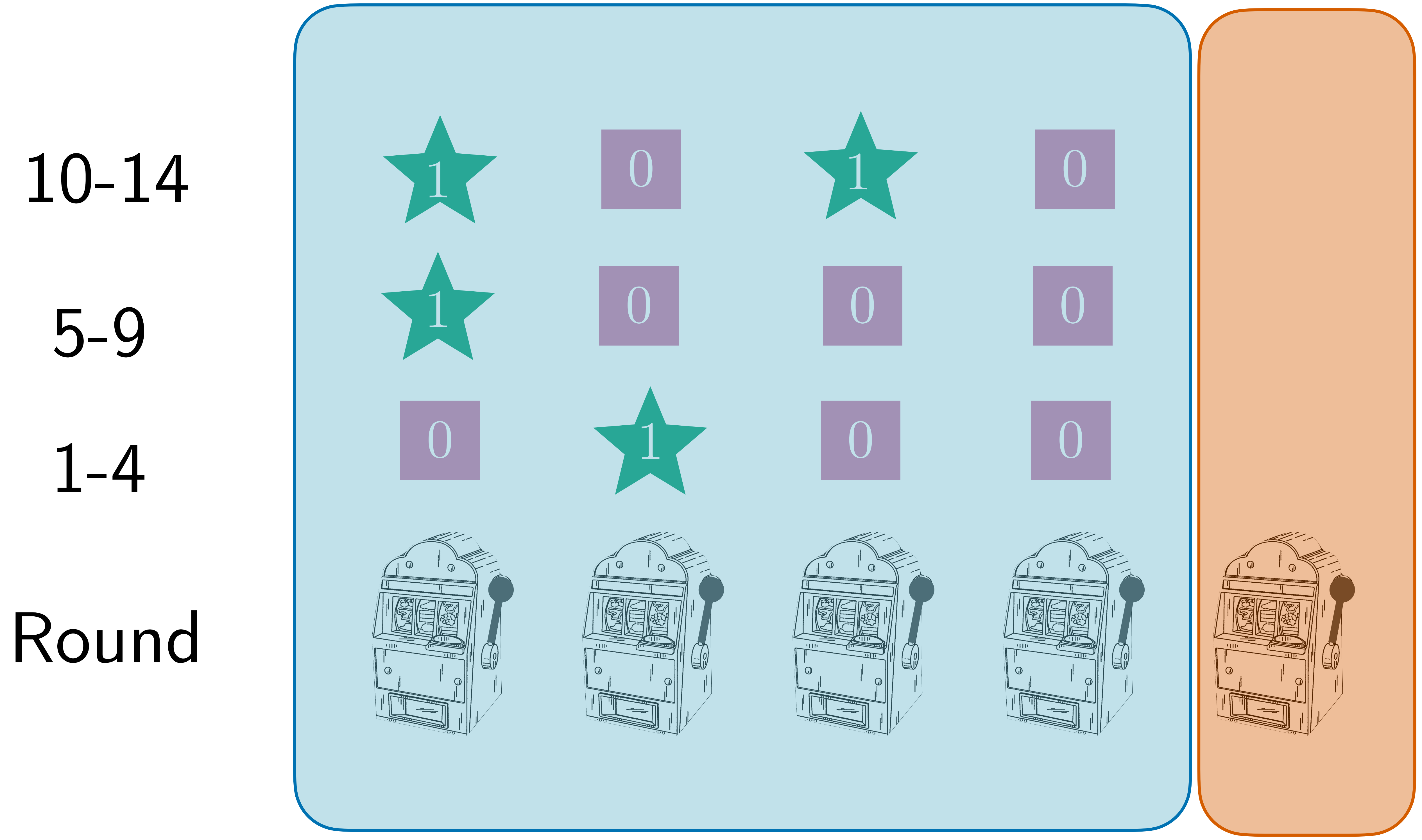


$$q_*(a) = \mathbb{E}[R_t \mid A_t = a]$$

A Typical Run:

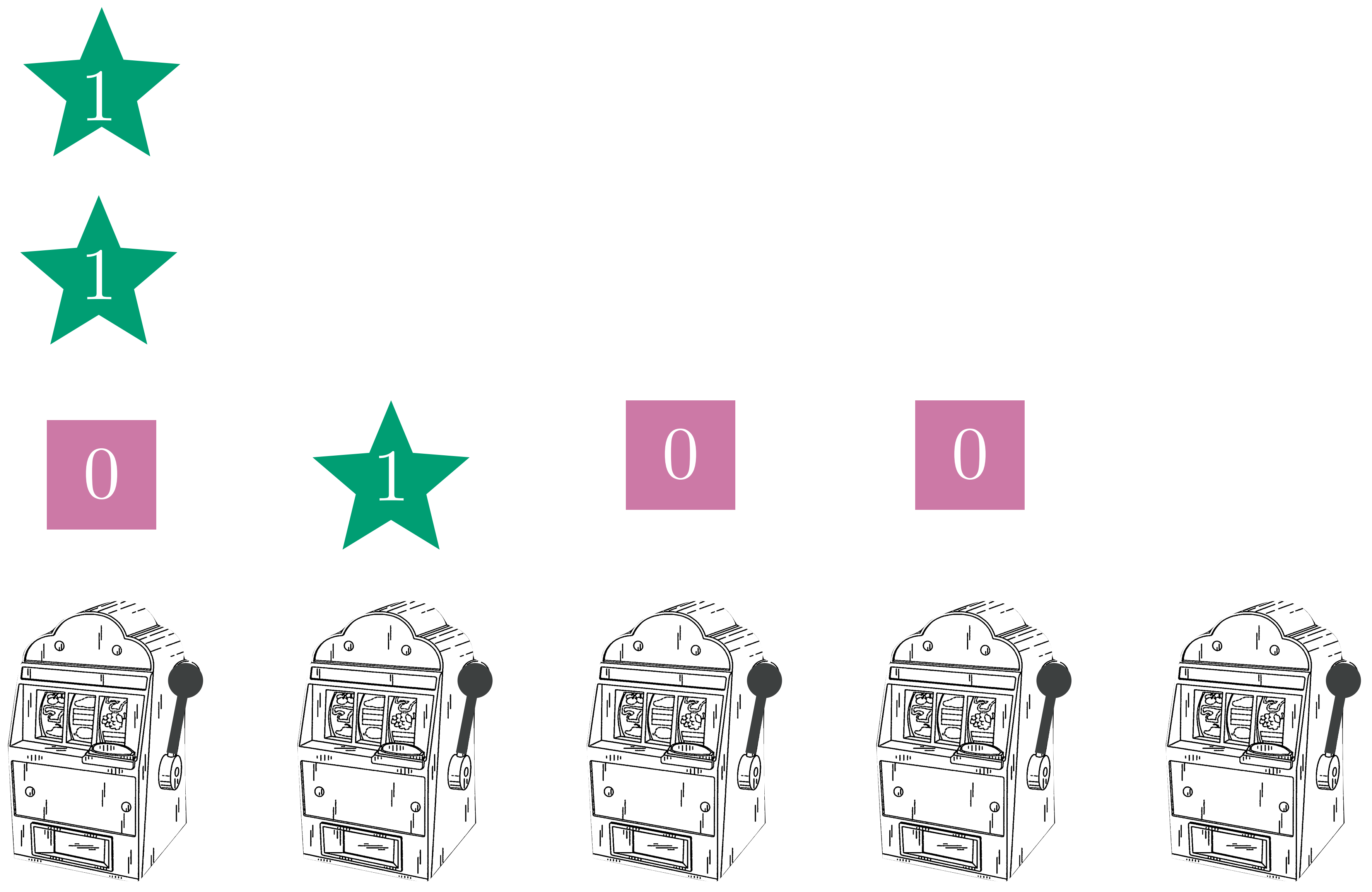


The Explore-Exploit Dilemma



Exploit: Pick best option so far **Explore:** Learn more about other options

The Explore-Exploit Dilemma: Always Present, After t=1



Exploit: Pick best option so far **Explore:** Learn more about other options

The Explore-Exploit Dilemma

Definition (Explore-Exploit Dilemma):

How to balance exploration and exploitation to maximise long-term rewards?

Exploit: Pick best option so far

Explore: Learn more about other options

The Explore-Exploit Dilemma

Definition (Explore-Exploit Dilemma):

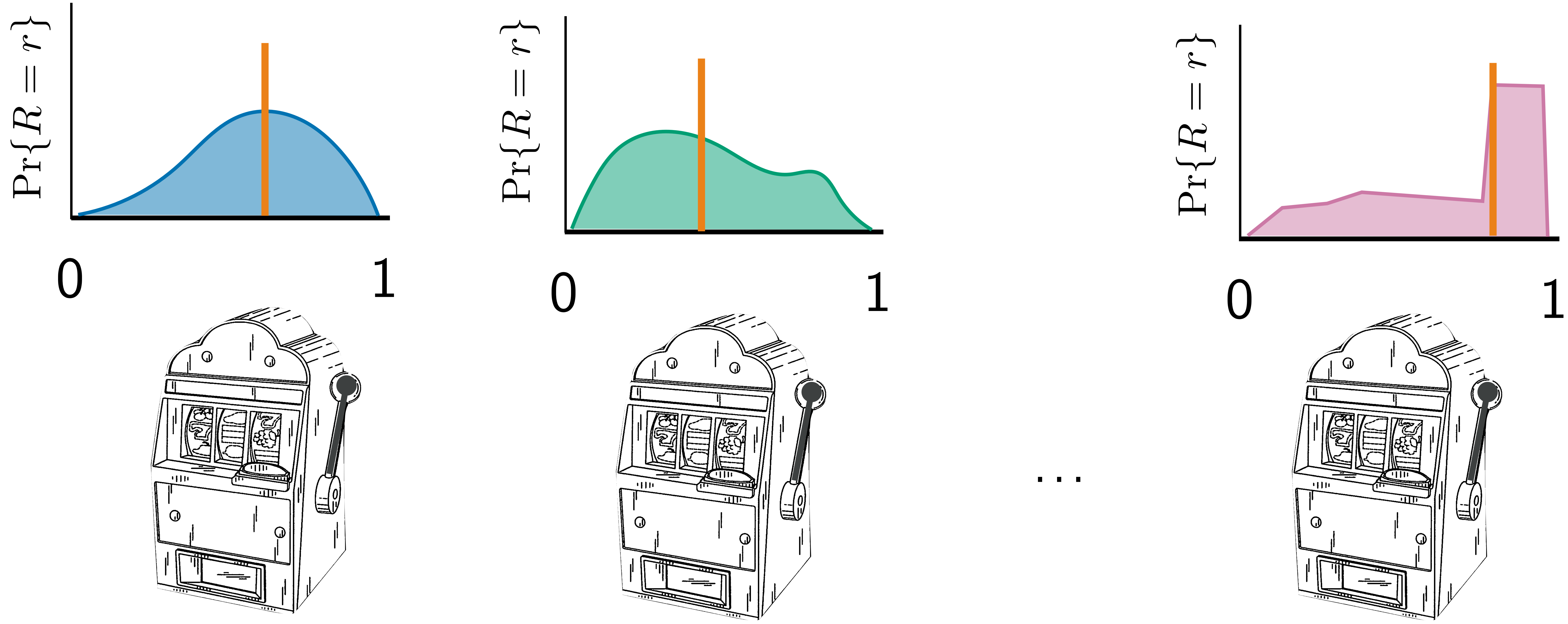
How to balance exploration and exploitation to maximise long-term rewards?

Discussion (2 minutes):

Why will *pure-exploration* or *pure-exploitation* fail?

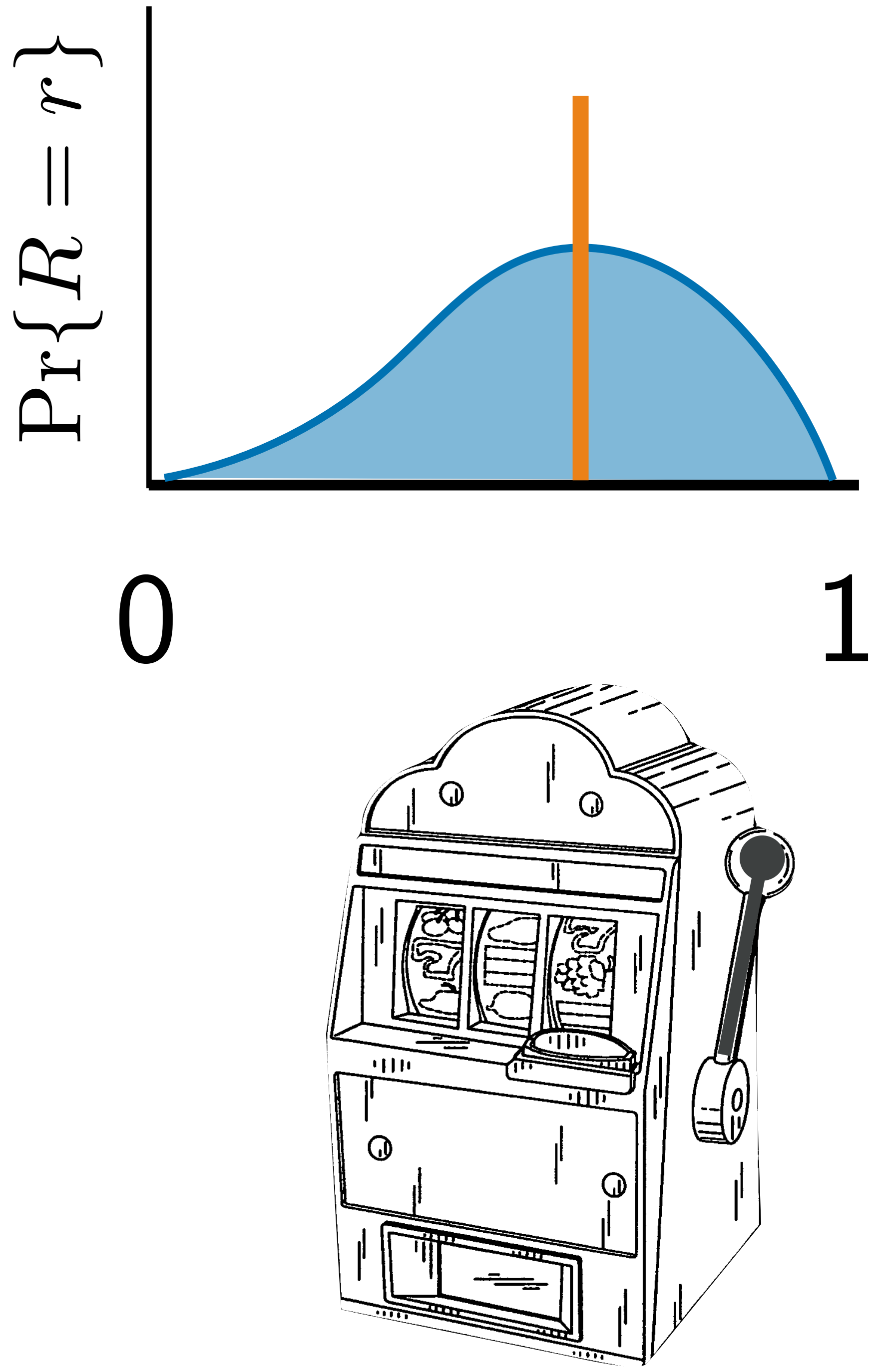
How might you balance between the two?

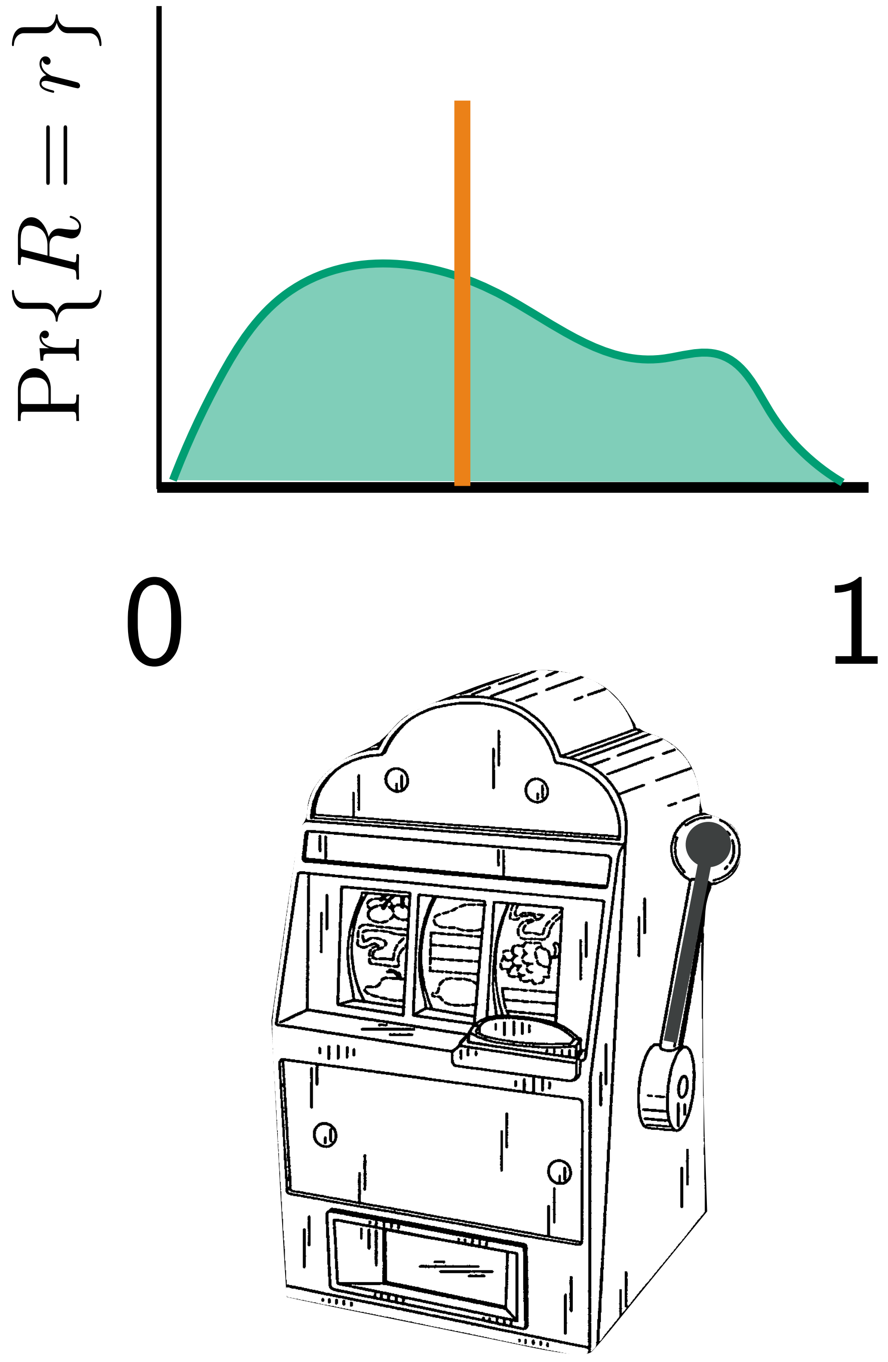
MAB Approach 1: Action-Value Methods



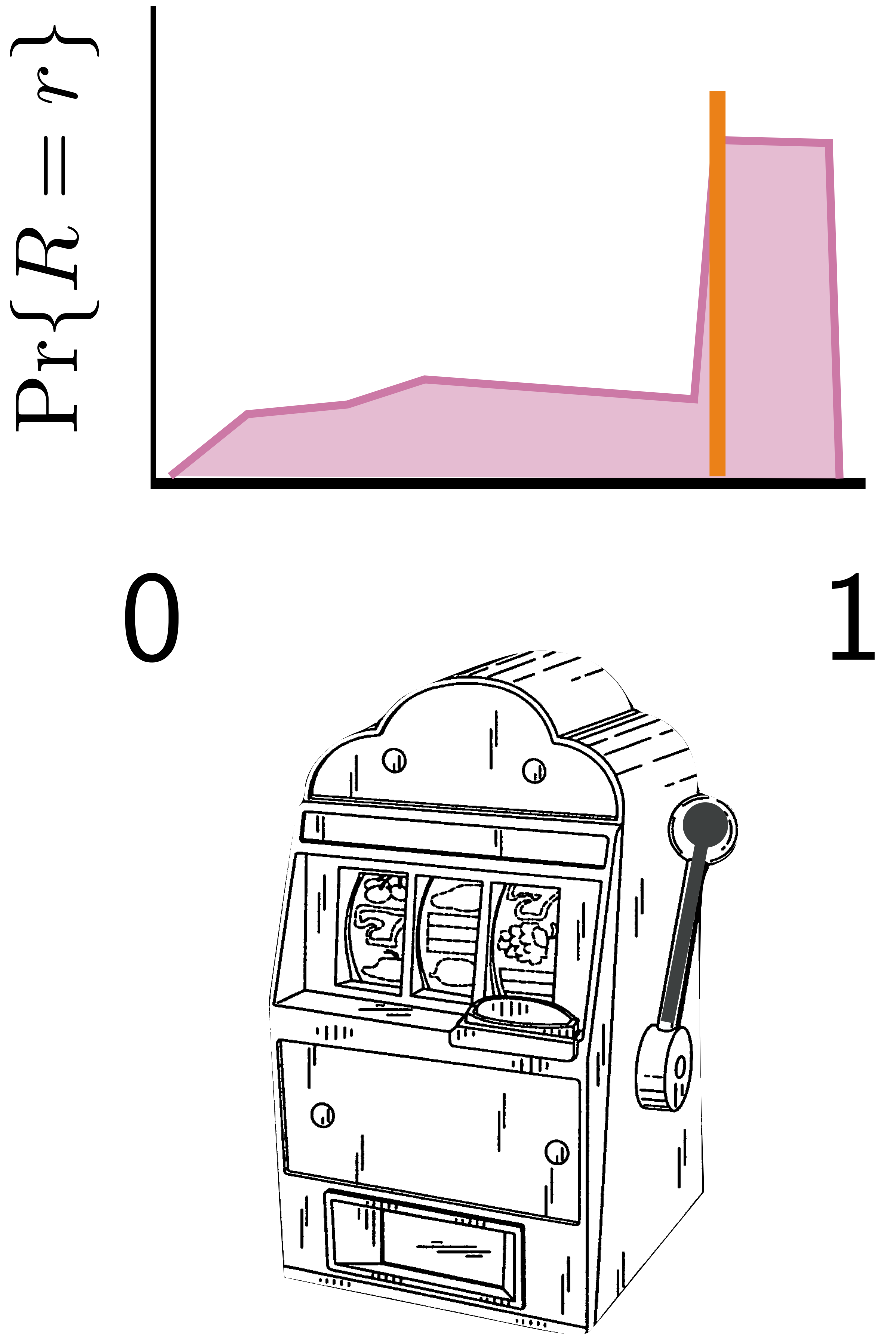
$$q_*(a) = \mathbb{E}[R_t \mid A_t = a]$$

MAB Approach 1: Action-Value Methods



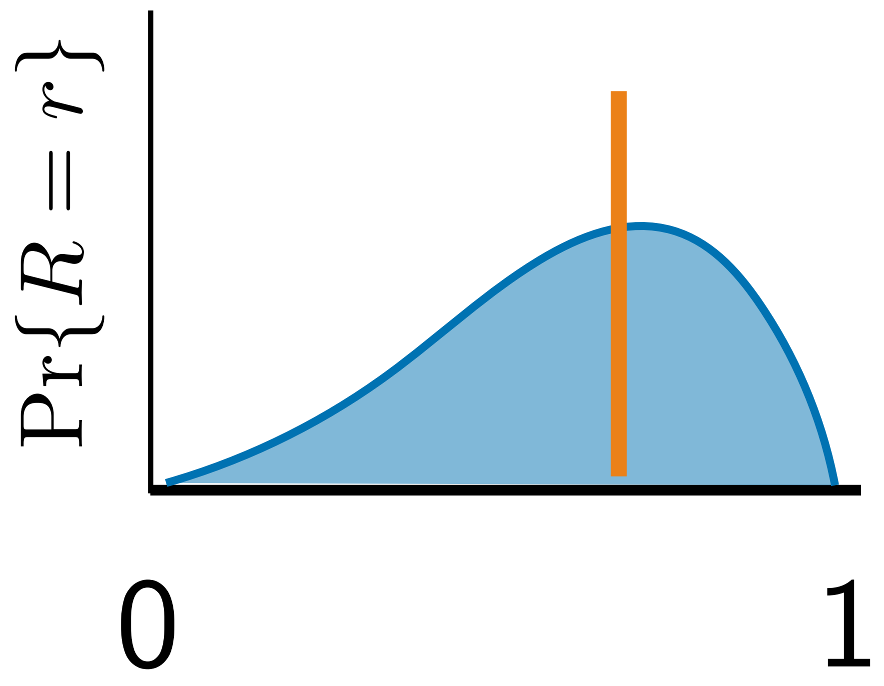


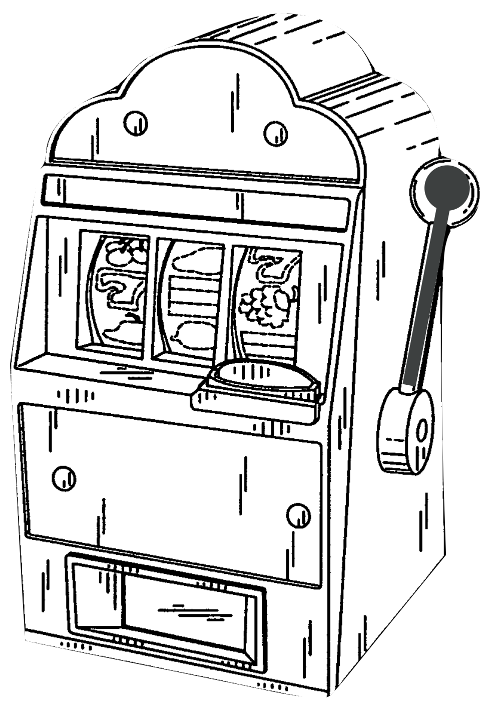
...



$q_*(a_1) = 0.55$ $q_*(a_2) = 0.4$ $q_*(a_k) = 0.8$

Estimating the Action-Value



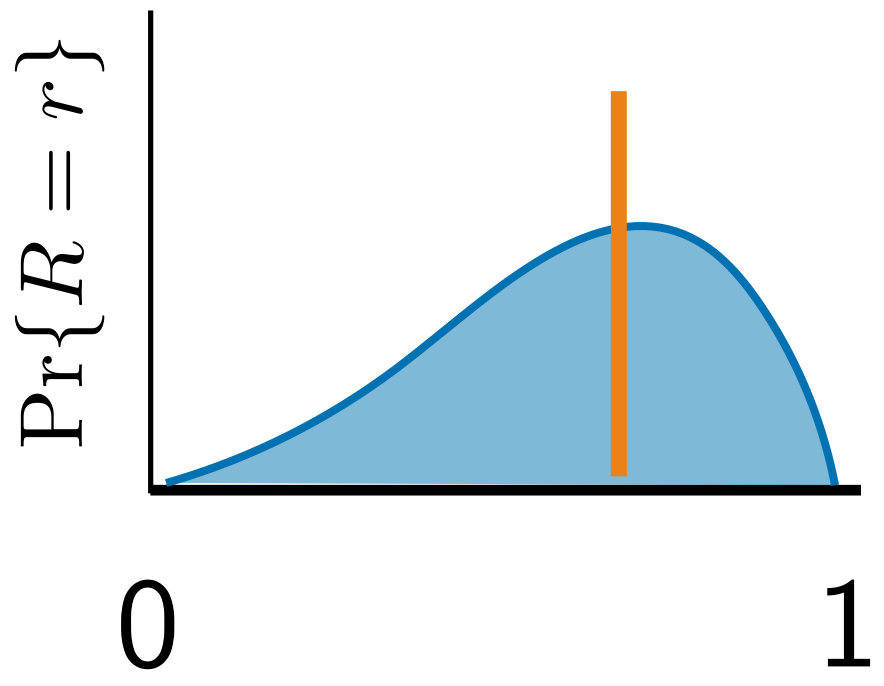


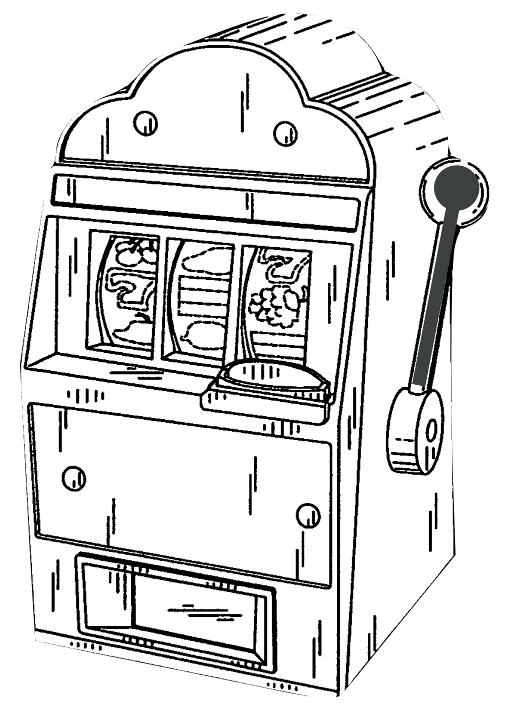
Main Idea: Estimate the value of each arm!

$$Q_t(a) = \frac{\textit{Sum of rewards when taken a so far}}{\textit{Number of times taken a so far}}$$

$$Q_t(a) = \frac{1}{N_t(a)} \sum_{\tau=1}^{t-1} R_\tau \cdot \mathbb{1}_{A_t = a}$$

Estimating the Action-Value



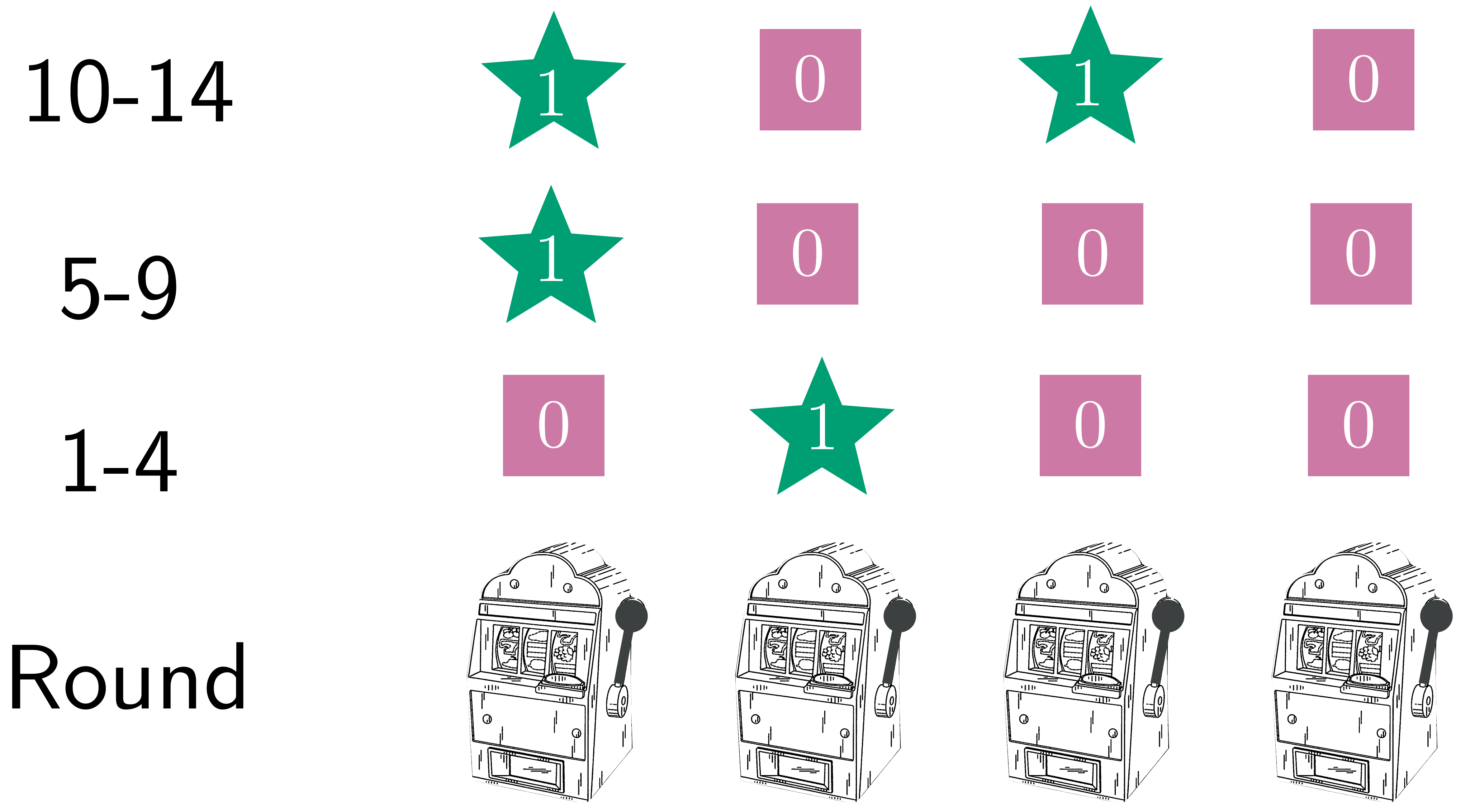


Sample average converges in the limit

$$\lim_{N_t(a)\to\infty} Q_t(a) = q_*(a)$$

$$Q_t(a) = \frac{1}{N_t(a)} \sum_{\tau=1}^{t-1} R_\tau \cdot \mathbb{1}_{A_t=a}$$

How to Explore, Exploit



Exploit: Pick best option so far

$$A_t = A_t^* = \arg\max_a Q_t(a)$$

Greedy action selection

Explore: Learn more about other options

$$A_t \sim \mathrm{Unif}(\mathcal{A})$$

Random action selection

MAB Algorithm 1: ϵ - greedy Action Selection

Algorithm: ϵ-greedy

0 $Q_1(a), N_1(a) = 0, \forall a \in \mathcal{A}$

1 For each round t in T:

2 $A_t = \begin{cases} A_t^* & \text{Pr } 1-\epsilon \\ \text{Unif}(\mathcal{A}) & \text{otherwise} \end{cases}$

3 Execute A_t, observe R_t

4 Update $N_t(a)$, $Q_t(a)$

Exploit: Pick best option so far

$$A_t = A_t^* = \arg\max_a Q_t(a)$$

Greedy action selection

Explore: Learn more about other options

$$A_t \sim \text{Unif}(\mathcal{A})$$

Random action selection

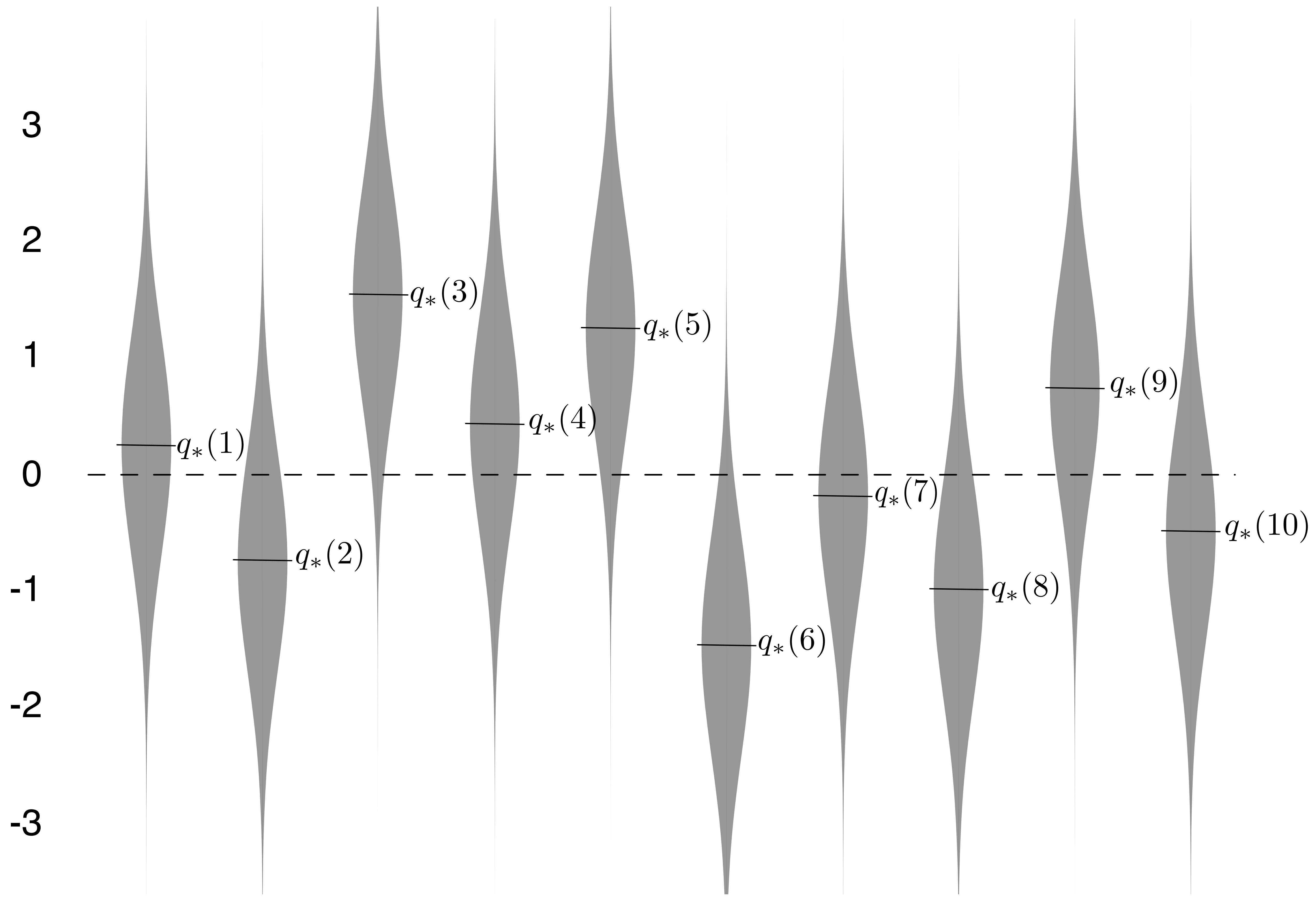
MAB Experiment: Setup

2000 random MABs

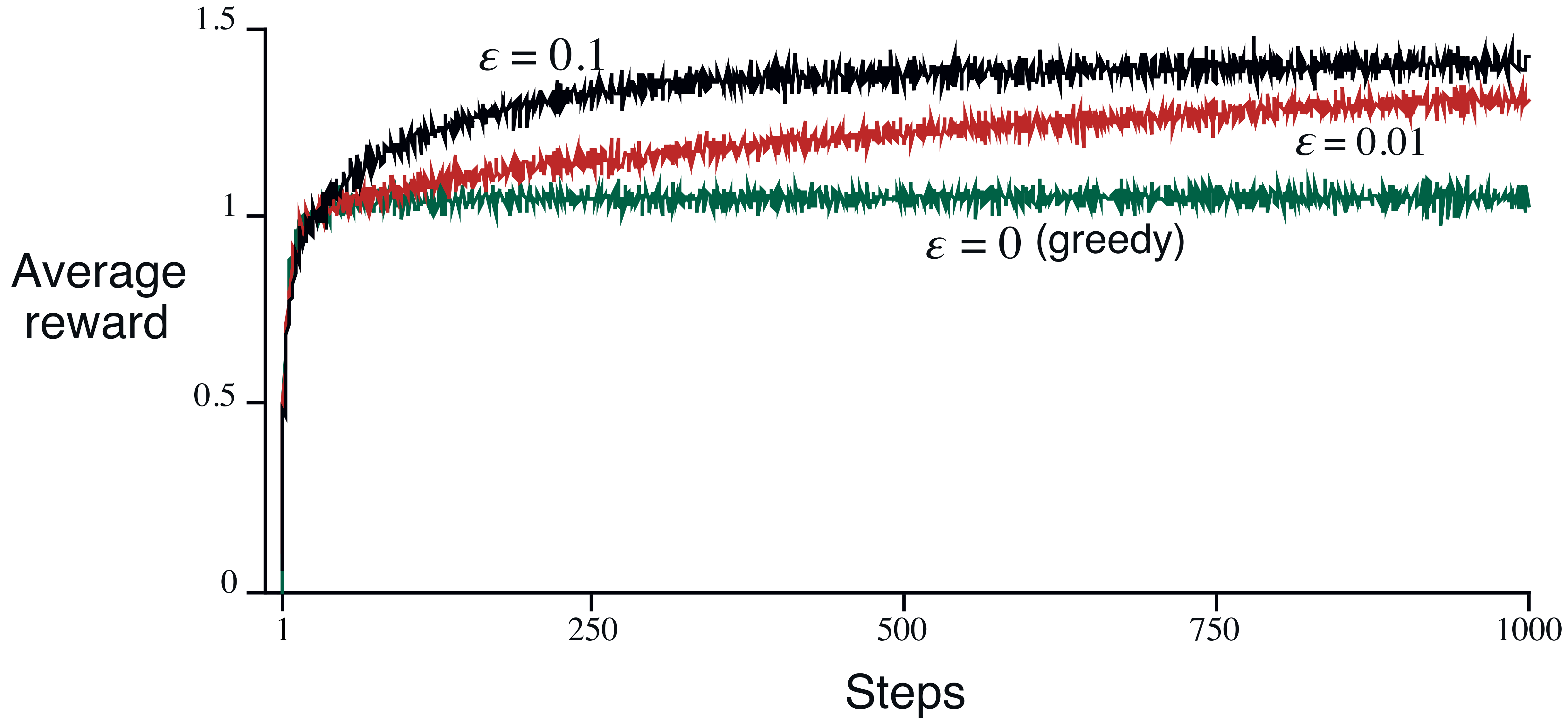
each with 10 arms

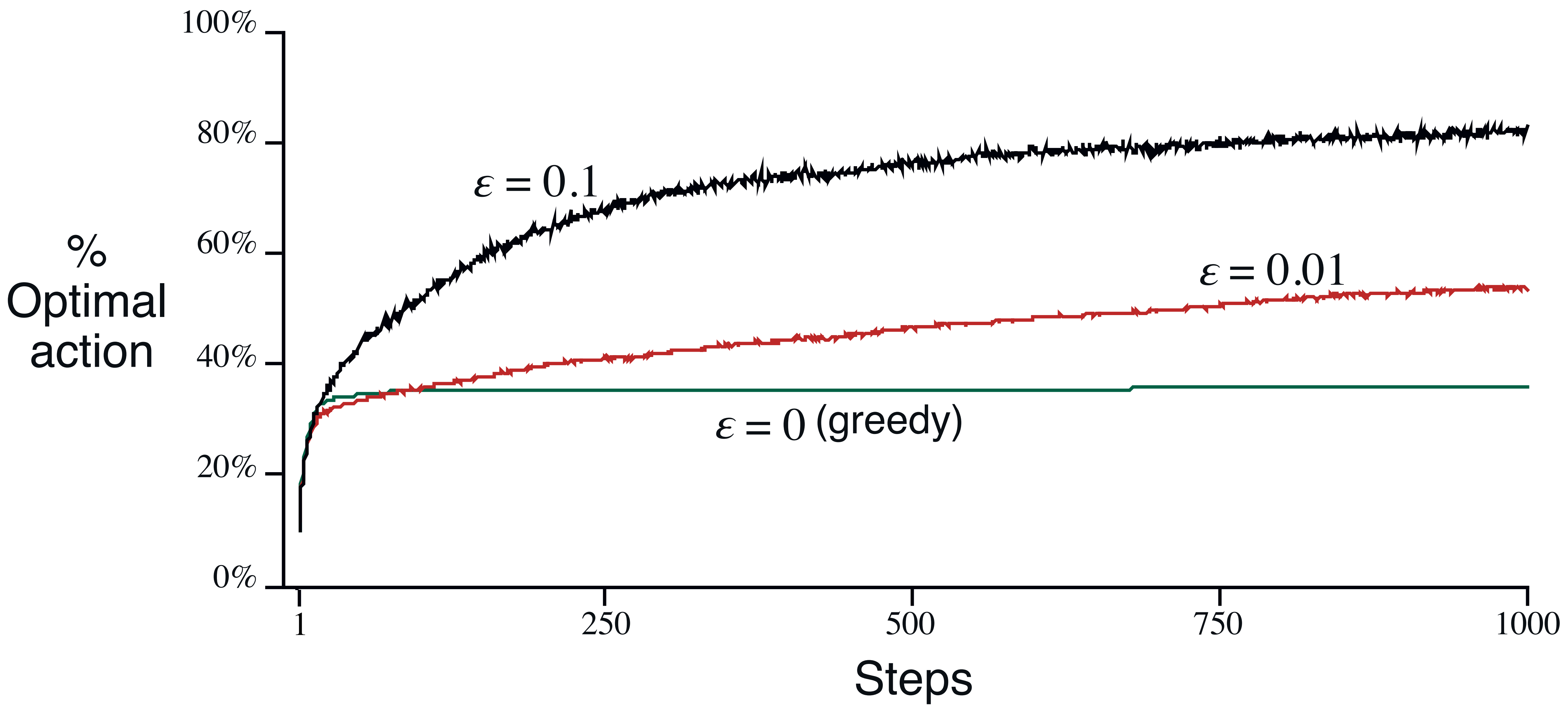
normal reward dist.

each 1000 rounds



MAB Experiment: Results





Where is $\epsilon = 0.1$ after

10,000 time steps?

Incremental Learning Rule

Sample average (focusing on a single action):

$$Q_n = \frac{R_1 + R_2 + \ldots + R_{n-1}}{n-1}$$

Can compute *incrementally* to avoid recomputing:

$$Q_{n+1} = Q_n + \frac{1}{n}[R_n - Q_n]$$

Learning Rules

Standard form for update rules in RL

NewEstimate <— OldEstimate + StepSize[Target - OldEstimate]

Can compute *incrementally* to avoid recomputing:

$$Q_{n+1} = Q_n + \frac{1}{n}[R_n - Q_n]$$

Simple Bandit Algorithm

A simple bandit algorithm

Initialize, for $a = 1$ to k:

$$Q(a) \leftarrow 0$$
$$N(a) \leftarrow 0$$

Loop forever:

$$A \leftarrow \begin{cases} \arg\max_a Q(a) & \text{with probability } 1 - \varepsilon \quad \text{(breaking ties randomly)} \\ \text{a random action} & \text{with probability } \varepsilon \end{cases}$$
$$R \leftarrow bandit(A)$$
$$N(A) \leftarrow N(A) + 1$$
$$Q(A) \leftarrow Q(A) + \frac{1}{N(A)}\big[R - Q(A)\big]$$

Non-Stationary Problems

Issue: Suppose the true action values *shift over time*:

- This problem is then called *non-stationary*
- Sample average alone is no longer appropriate (why?)
- Very common issue in RL!

Solution: track action values using a **step-size parameter,** $\alpha \in (0, 1]$

$$Q_{n+1} = Q_n + \alpha[R_n - Q_n]$$

Stochastic Approximation Convergence Conditions

Estimates Q_n will converge with probability 1 to q_* if:

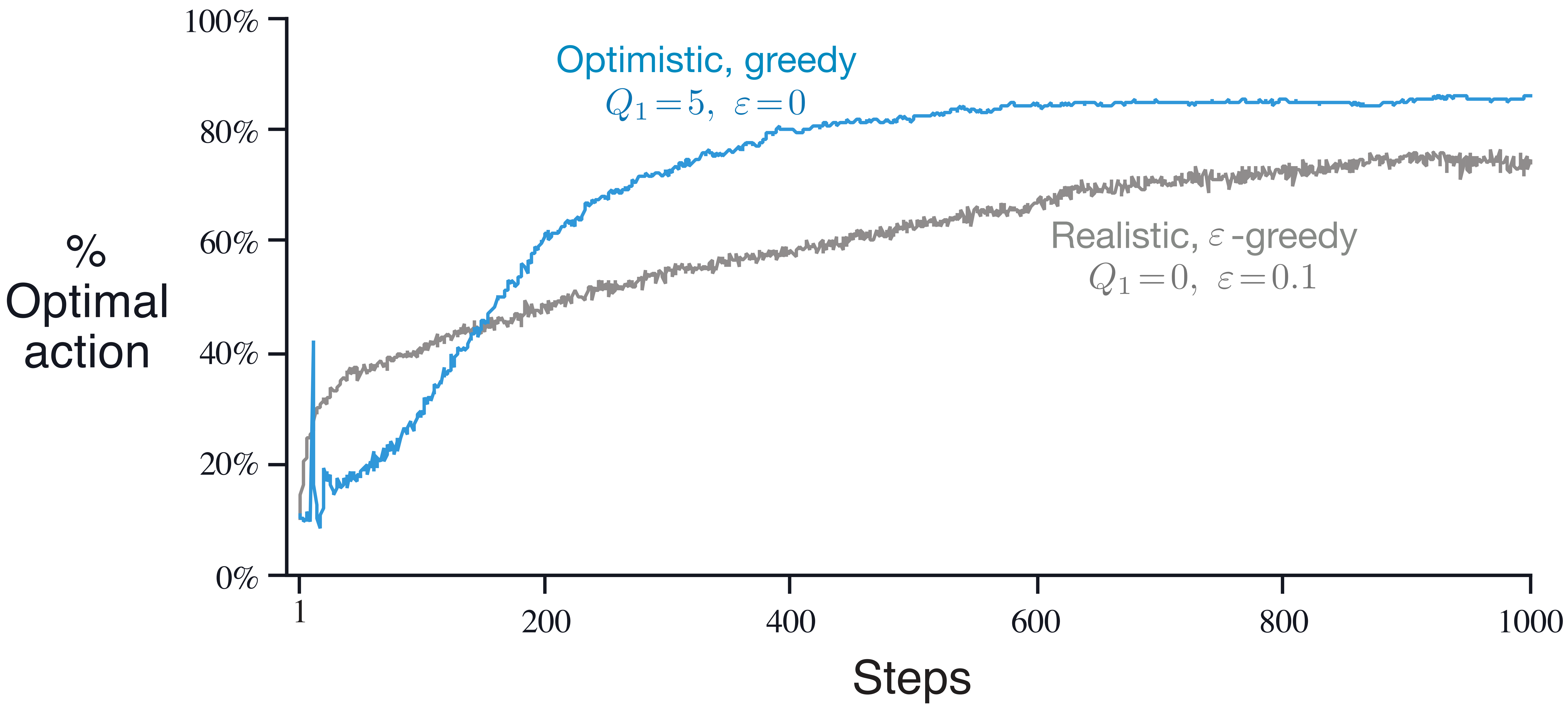
$$\sum_{n=1}^{\infty} \alpha_n(a) \to \infty \qquad \text{and} \qquad \sum_{n=1}^{\infty} \alpha_n^2(a) < \infty$$

Based on a classical result by Robbins and Monro (1951)

Works: $\alpha_n = \frac{1}{n}$

Not: $\alpha_n = c$, $\alpha_n = \frac{1}{n^2}$

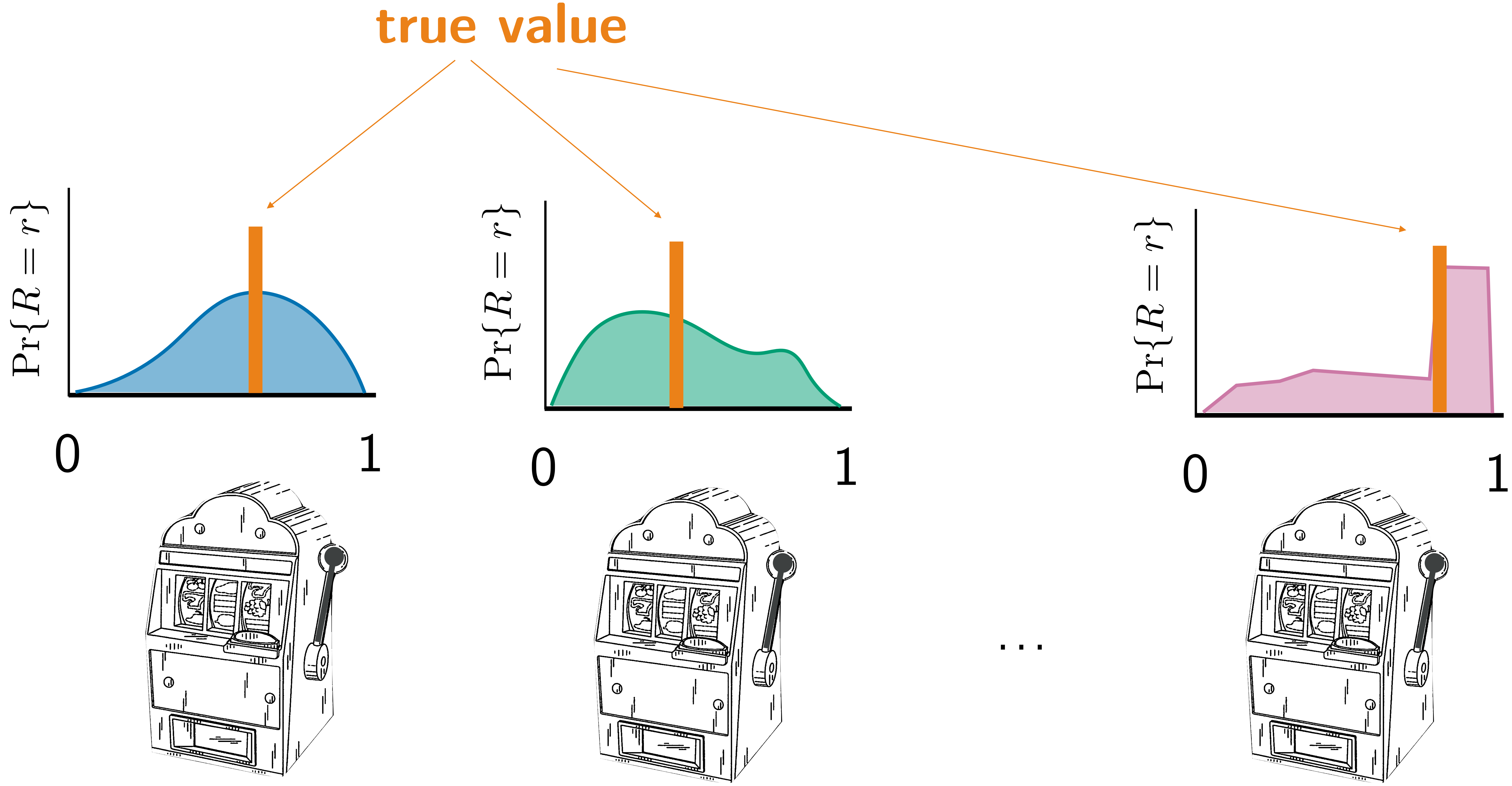
Explore-Exploit Principle: *Optimism Under Uncertainty*



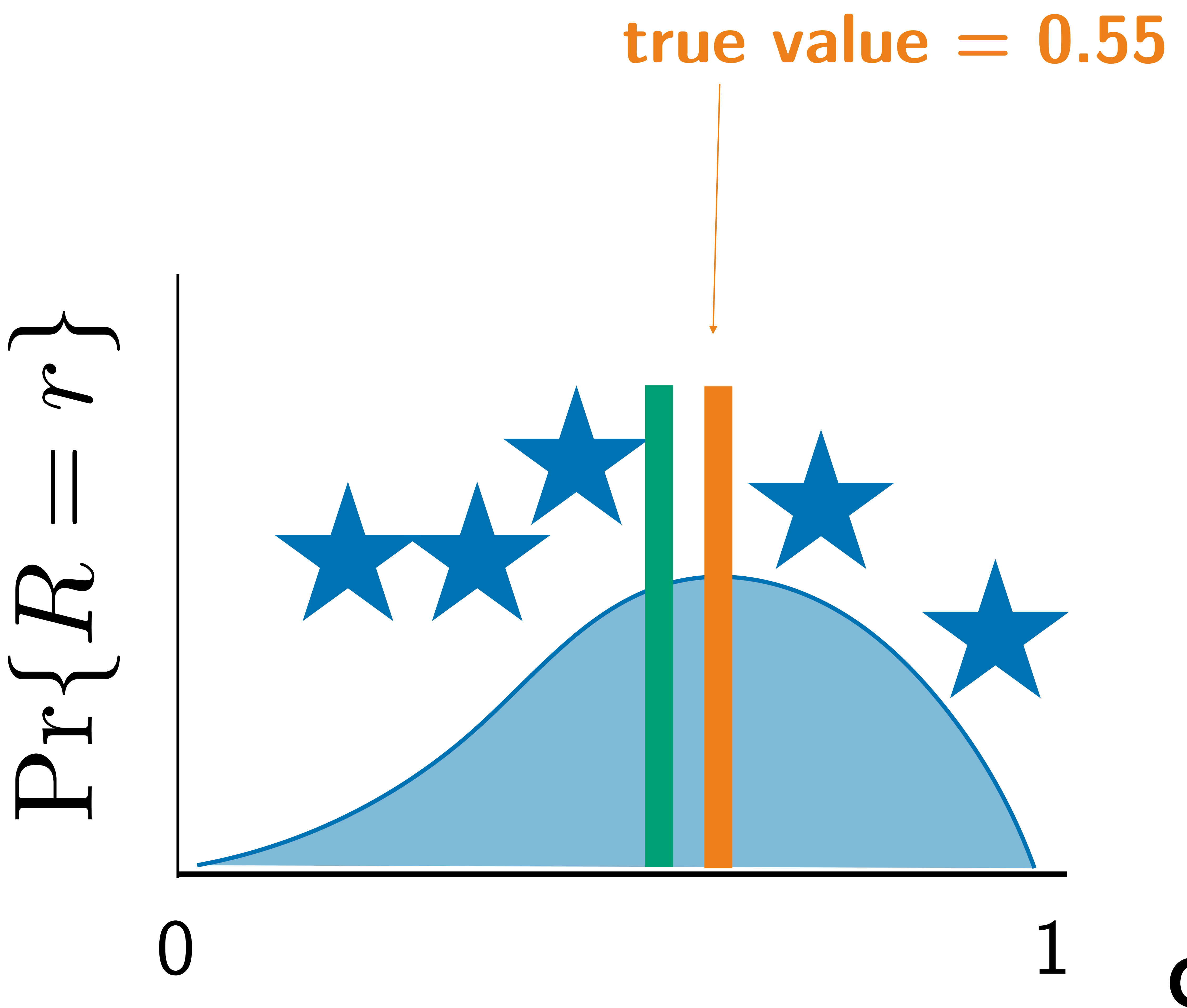
Optimism: Set Q_1 to be high!

See RL book: Section 2.6

Algorithm 2: Upper Confidence Bound (UCB)



Algorithm 2: Upper Confidence Bound (UCB)



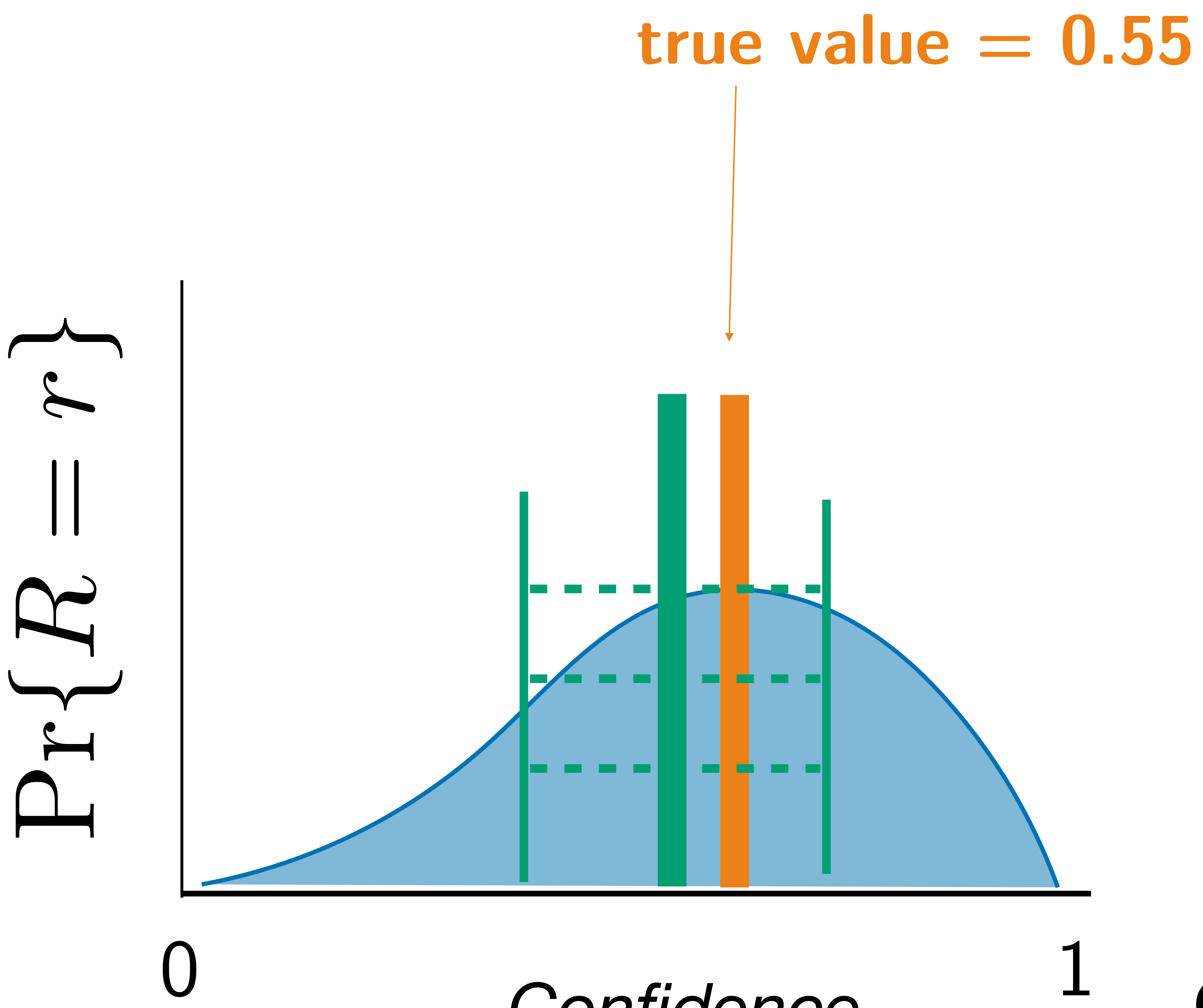
Samples after five rounds:

0.2, 0.4, 0.45, 0.6, 0.9

—> Sample average = 0.51

Q: How much more optimistic should we be?

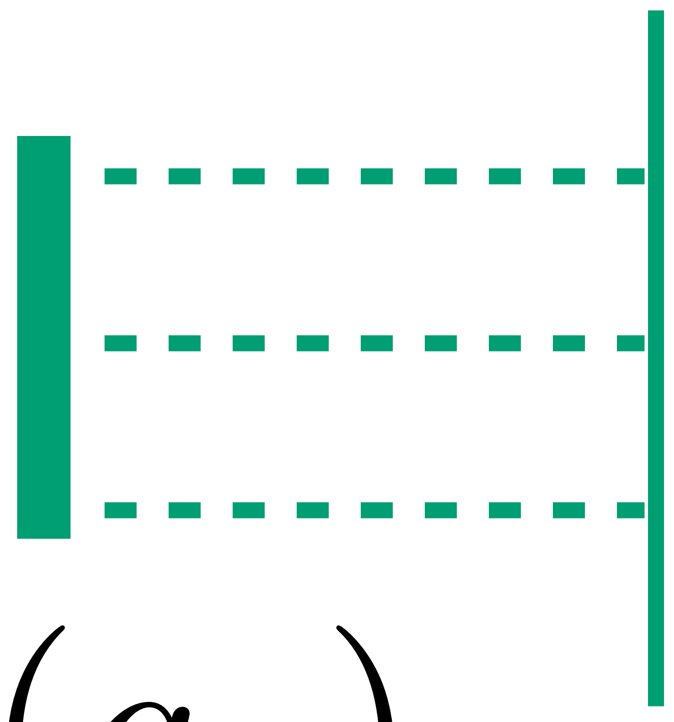
Confidence Intervals



Confidence intervals

Samples after five rounds:

0.2, 0.4, 0.45, 0.6, 0.9



$$Q_t(a_1) + c\sqrt{\frac{\log t}{N_t(a)}}$$

Q: How much more optimistic should we be?

Algorithm 2: Upper Confidence Bound (UCB)

Algorithm: UCB

$Q_1(a), N_1(a) = 0, \forall a \in \mathcal{A}$

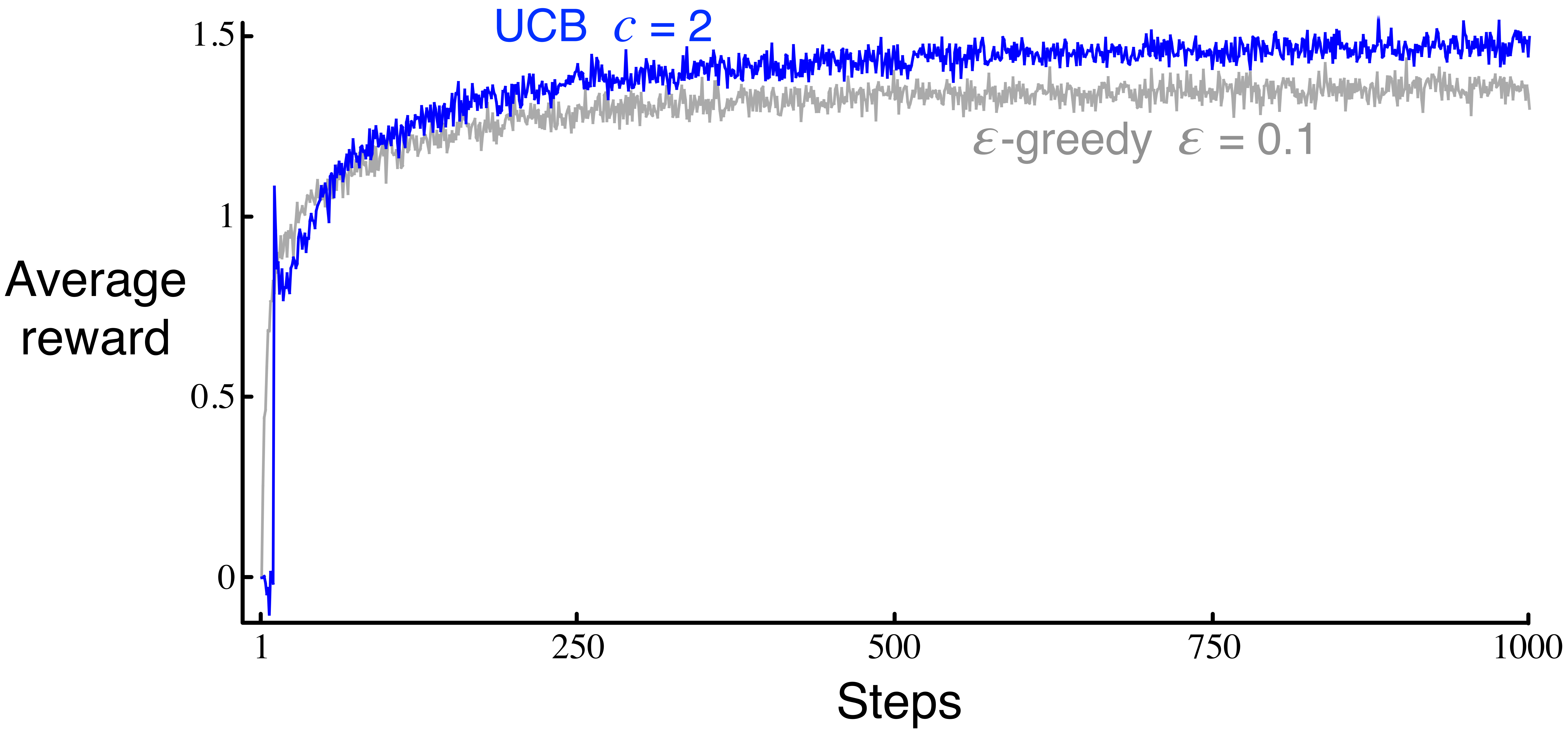
For each round t in T:

$$A_t = \begin{cases} \text{Unif}(\mathcal{A}) & \max_a N_t(a) = 0 \\ \arg\max_a [Q_t(a) + c\sqrt{\frac{\log t}{N_t(a)}}] & otherwise \end{cases}$$

Execute A_t, observe R_t

Update $N_t(a)$, $Q_t(a)$

Experiments: UCB vs. ϵ - greedy Action Selection



Gradient-Based Algorithms

We will return to this!

RL Book: Section 2.8

Recap

- Simplest RL problem: Multi-armed bandit (MAB)
- MAB: k actions, no state. Goal: maximise long term reward
- Dilemma: balance exploration and exploitation
- Two basic algorithms: greedy and UCB

Reading

- **RL Book, Chapter 2 (2.1-2.8)**

Box "The Bandit Gradient Algorithm as Stochastic Gradient Ascent" in Sec 2.8 not examined

Optional

- UCB paper: P. Auer, N. Cesa-Bianchi, P. Fischer (2002). *Finite-time analysis of the multi-armed bandit problem.* Machine Learning, 47(2-3), 235-256.
- Book: *Bandit Algorithms* by Tor Lattimore and Csaba Szepesvári. Free download: https://tor-lattimore.com/downloads/book/book.pdf 28