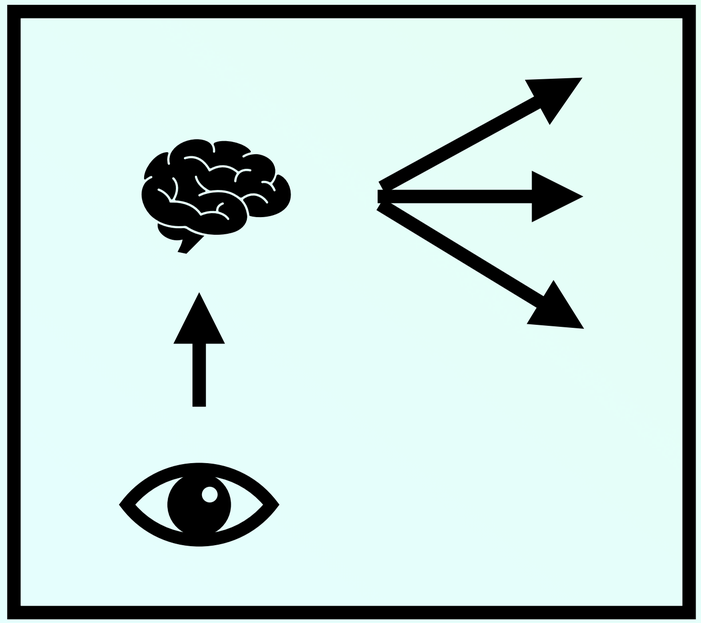
INTRO TO DECISION-MAKING: BANDITS 1

SEQUENTIAL DECISION MAKING

NOTES

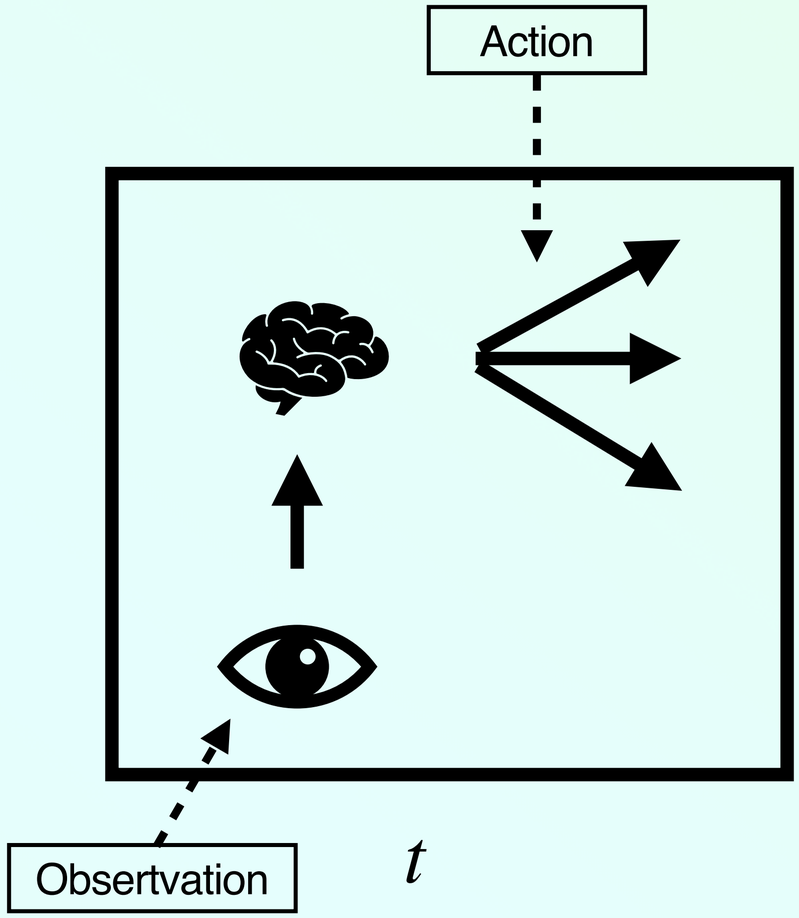
- There is no slack/discord
- Make sure you are enrolled in the private version
  - Deadlines for the first two assignments should be tonight and Jan 17 11:59 pm
- Careful using print statements on Coursera
  - Too many can cause the notebook to crash and need to be restarted.
- Be careful when sampling from random number generators.
  - Autograder needs specific values.
- There is also a python debugger

SEQUENTIAL DECISION MAKING

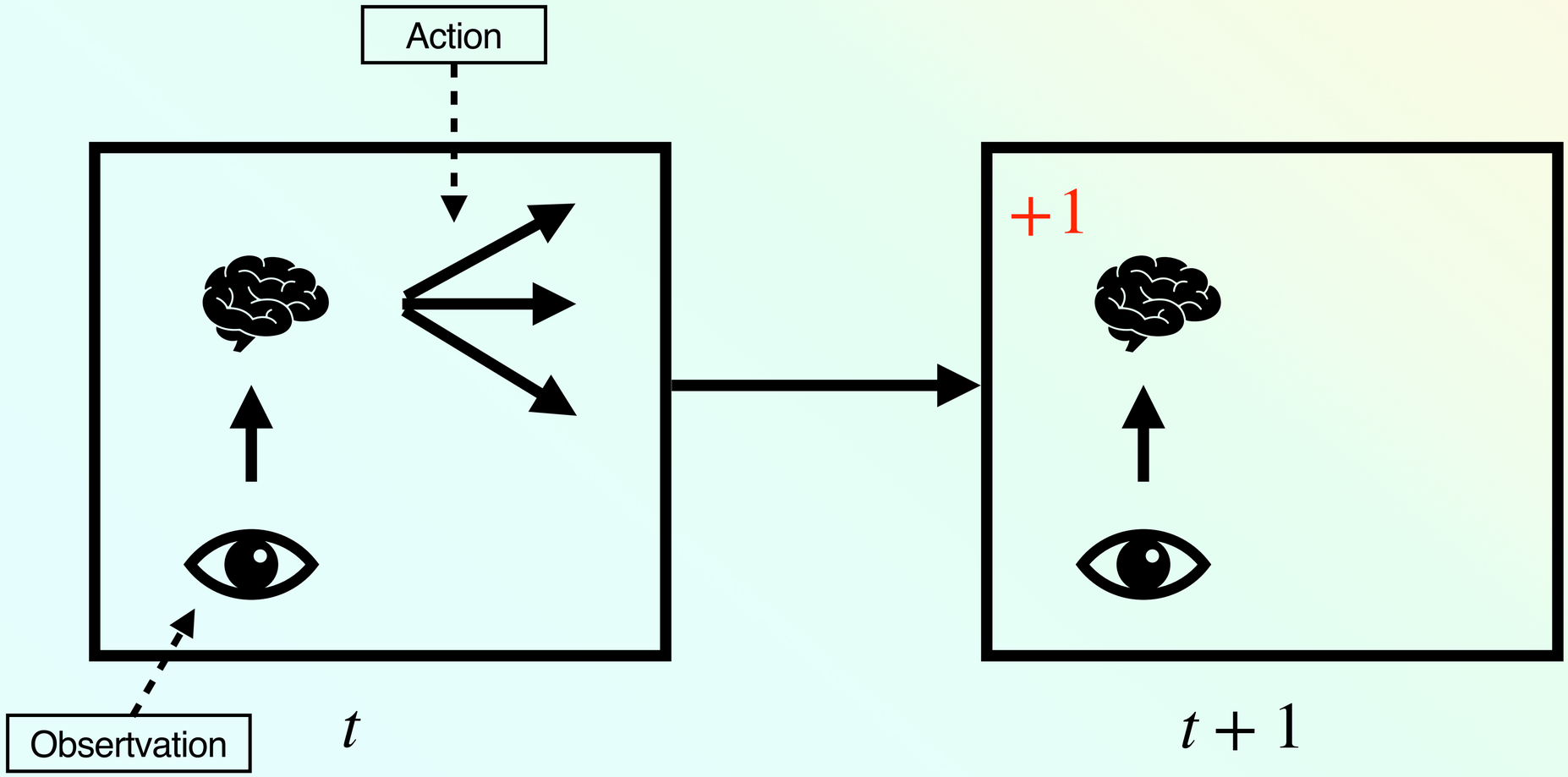


t

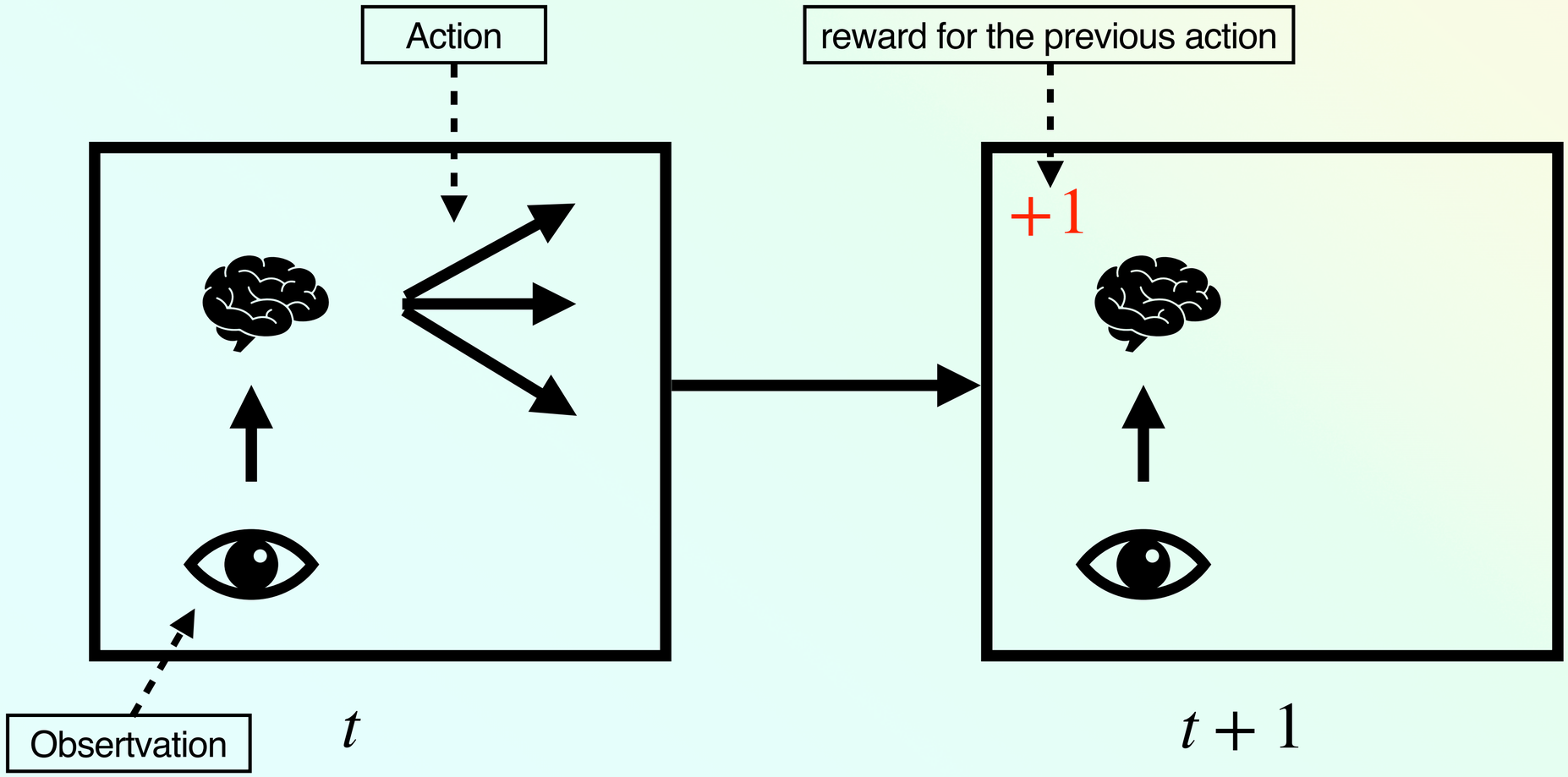
SEQUENTIAL DECISION MAKING



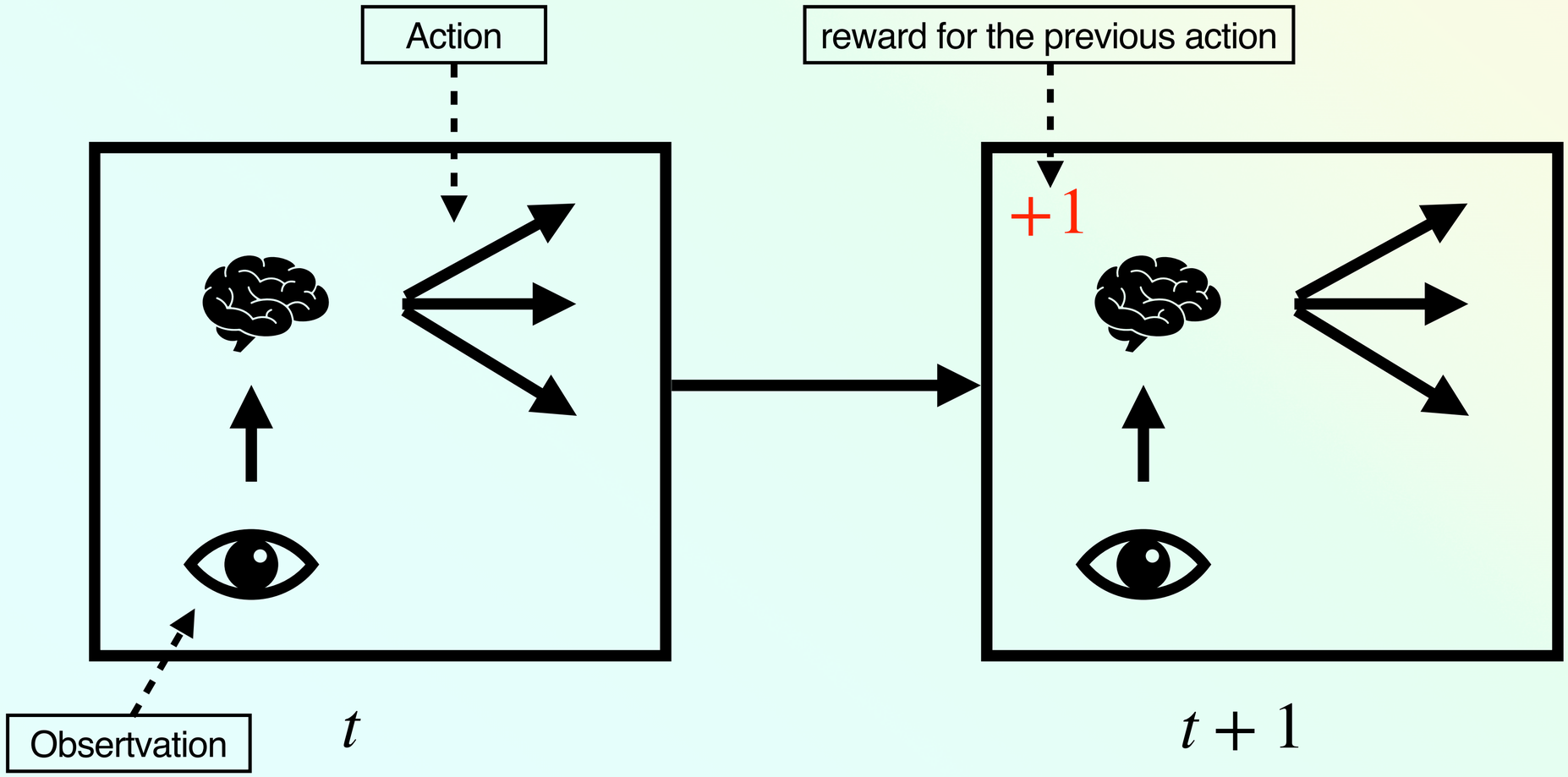
SEQUENTIAL DECISION MAKING



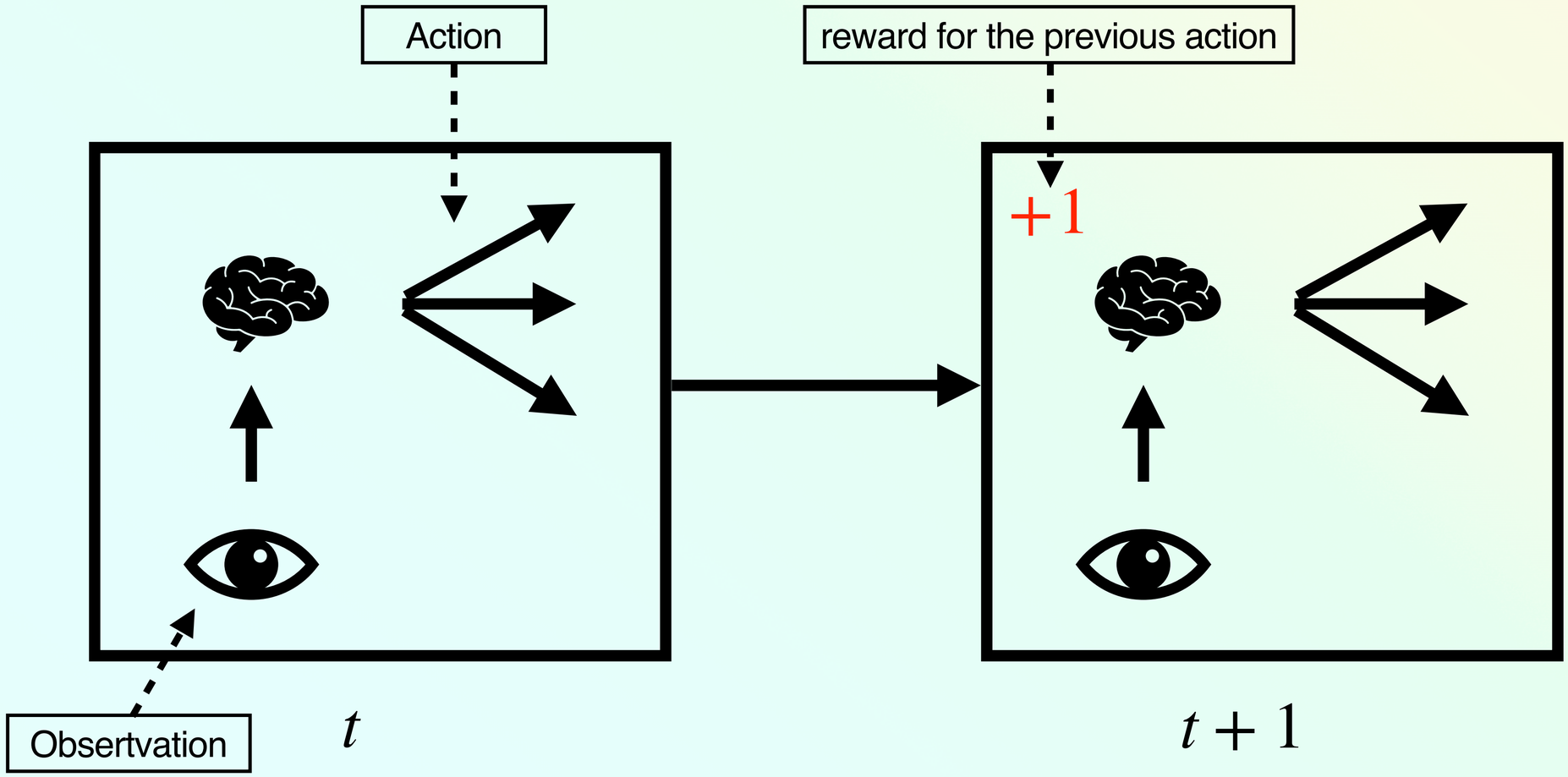
SEQUENTIAL DECISION MAKING



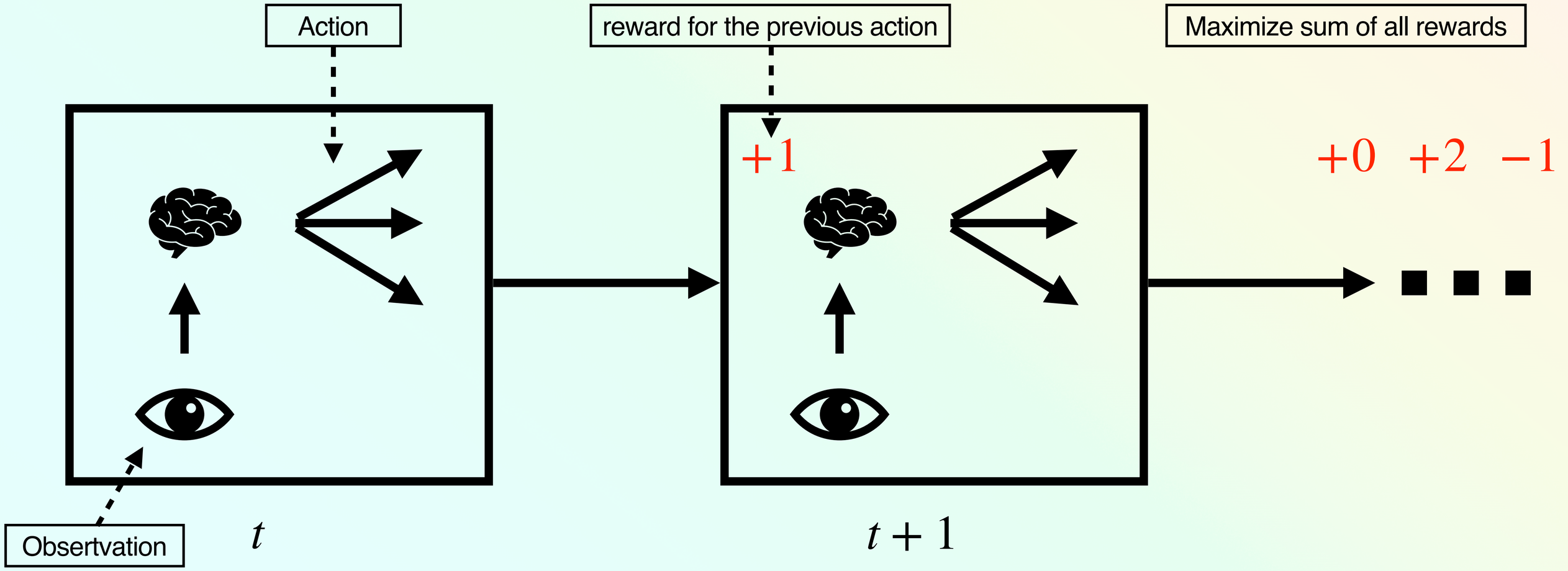
SEQUENTIAL DECISION MAKING



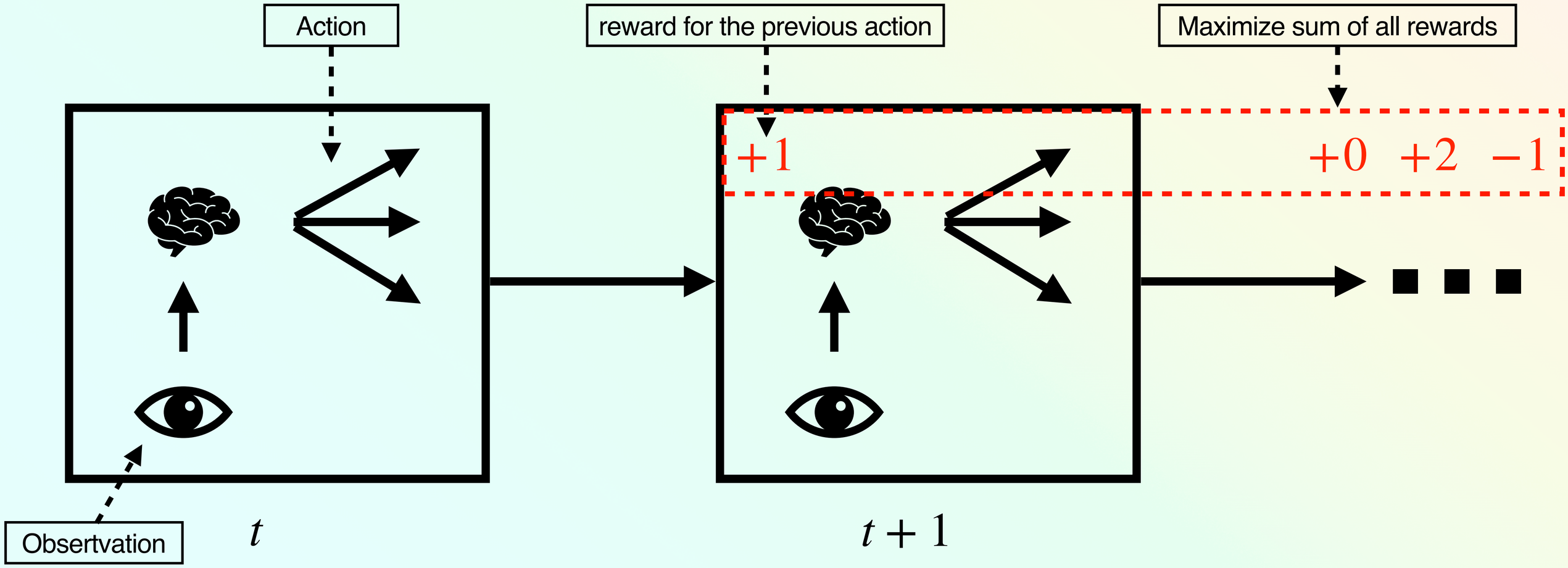
SEQUENTIAL DECISION MAKING



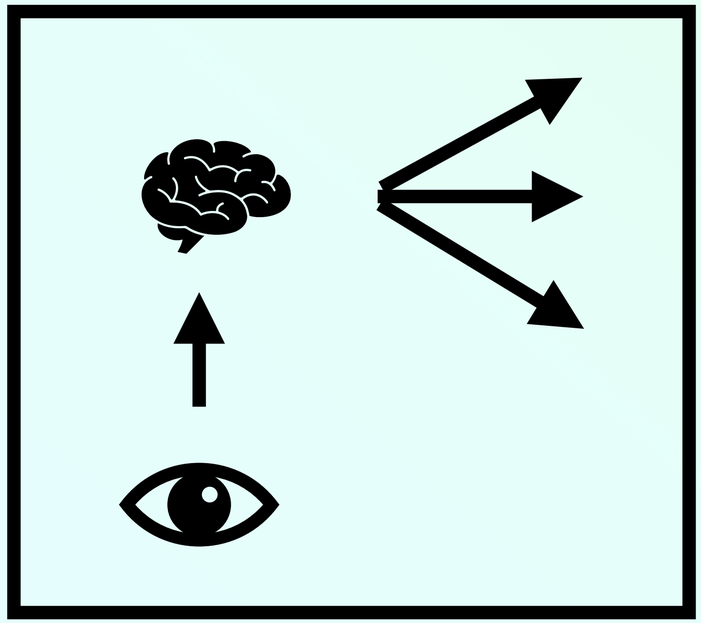
SEQUENTIAL DECISION MAKING



SEQUENTIAL DECISION MAKING



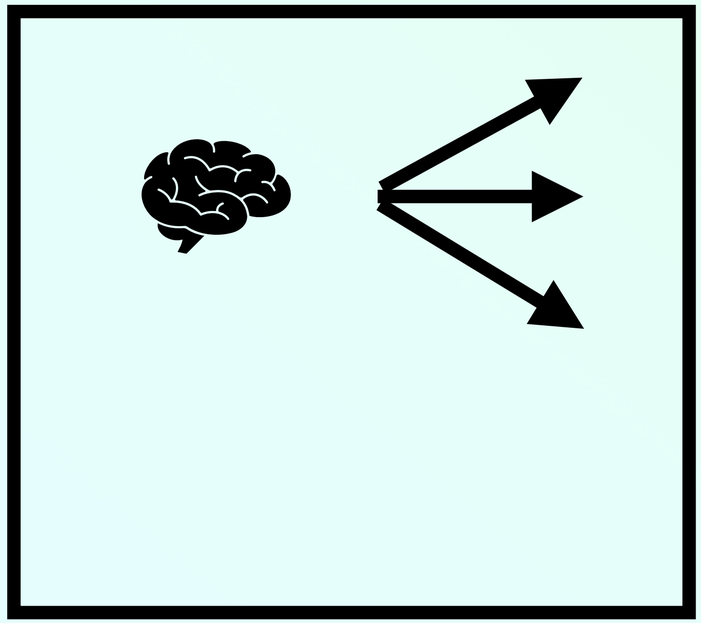
SEQUENTIAL DECISION MAKING



+1

One-step problem with observation

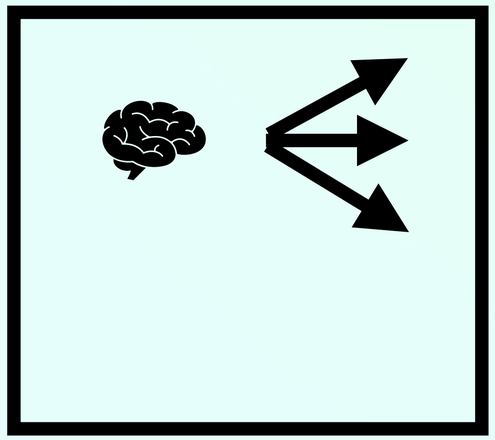
SEQUENTIAL DECISION MAKING



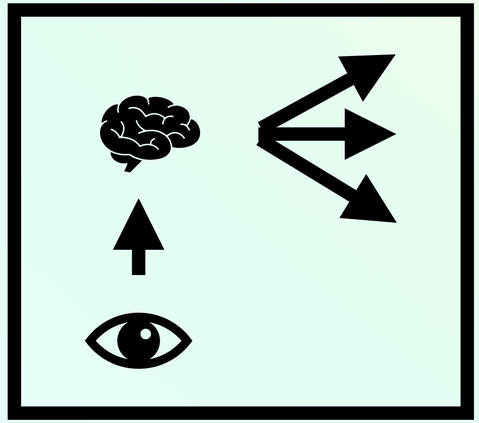
+1

One-step problem no observation

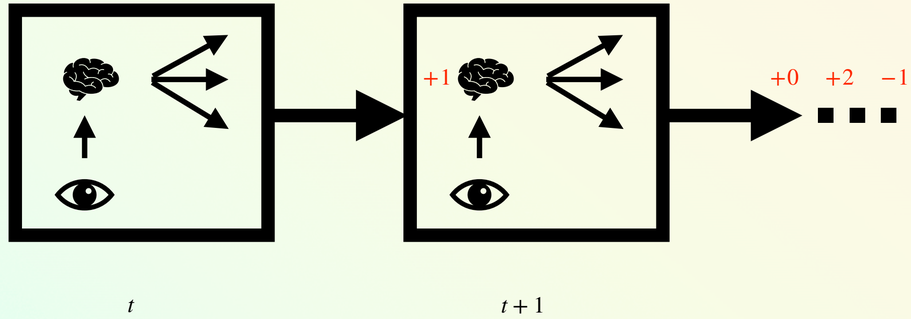
SEQUENTIAL DECISION MAKING



Bandit
1-step
No observation



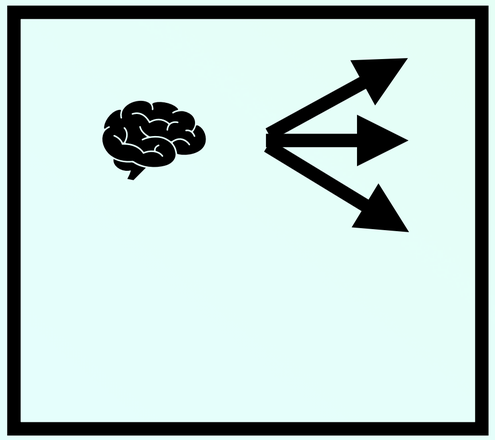
Contextual Bandit
1-step
Observation



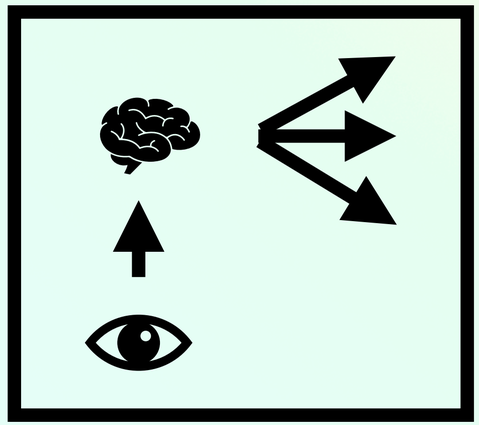
Markov Decision Process
Multi-step
Observation

SEQUENTIAL DECISION MAKING

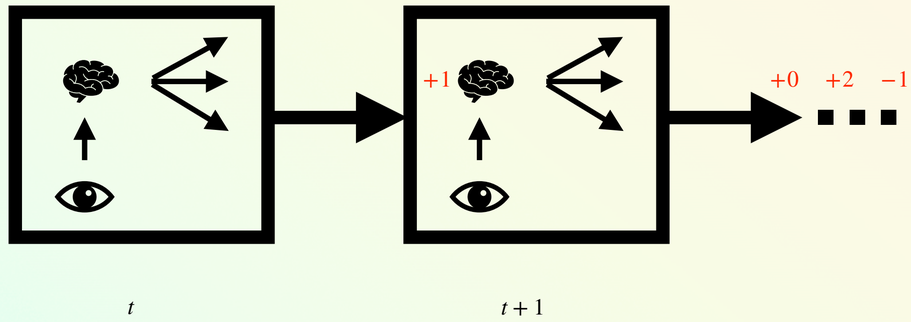
Study this one first



Bandit
1-step
No observation



Contextual Bandit
1-step
Observation



Markov Decision Process
Multi-step
Observation

BANDITS

WHY

1. Simple
2. Has many of the properties of multi-step problems
    - Value estimation
    - Requires balancing exploration/exploitation
3. Used in real-world settings
    - Advertising
    - Recommender systems

BANDITS

DEFINITION

k-Armed bandit - $(\mathscr{A}, d_R)$

$\mathscr{A}$ — set of all actions, e.g., $\mathscr{A} = \{a_1, a_2, \dots\}$

$d_R \colon \mathscr{A} \to \Delta(\mathscr{R})$ — function that maps an action to a distribution of rewards

A_t — action taken at time t

$R_t \sim d_R(A_t)$ — reward observed after taking action A_t

$q_* \colon \mathscr{A} \to \mathbb{R}$ — average reward for an action, e.g., $q_*(a) = \mathbb{E}[R_t \,|\, A_t = a]$

BANDITS

DEFINITION

$$a^* \in \arg\max_{a \in \mathcal{A}} \mathbb{E}[R_t \mid A_t = a]$$

$$a^* \in \arg\max_{a \in \mathcal{A}} q_*(a)$$

BANDIT

HOW TO SOLVE

Problems:

- Don’t know q_*
- Only get samples of R_t
- Need to estimate q_*
  - Must try all actions, even bad ones (exploration)

BANDIT

HOW TO SOLVE

Simple Algorithm

1. Sample each arm n times,

$$r_a = \{R_1, R_2, \ldots, R_n\} \text{ where each } R_i \sim d_R(a)$$

2. Compute the sample average of each arm

$$Q(a) = \frac{1}{n} \sum_{i=1}^{n} R_i$$

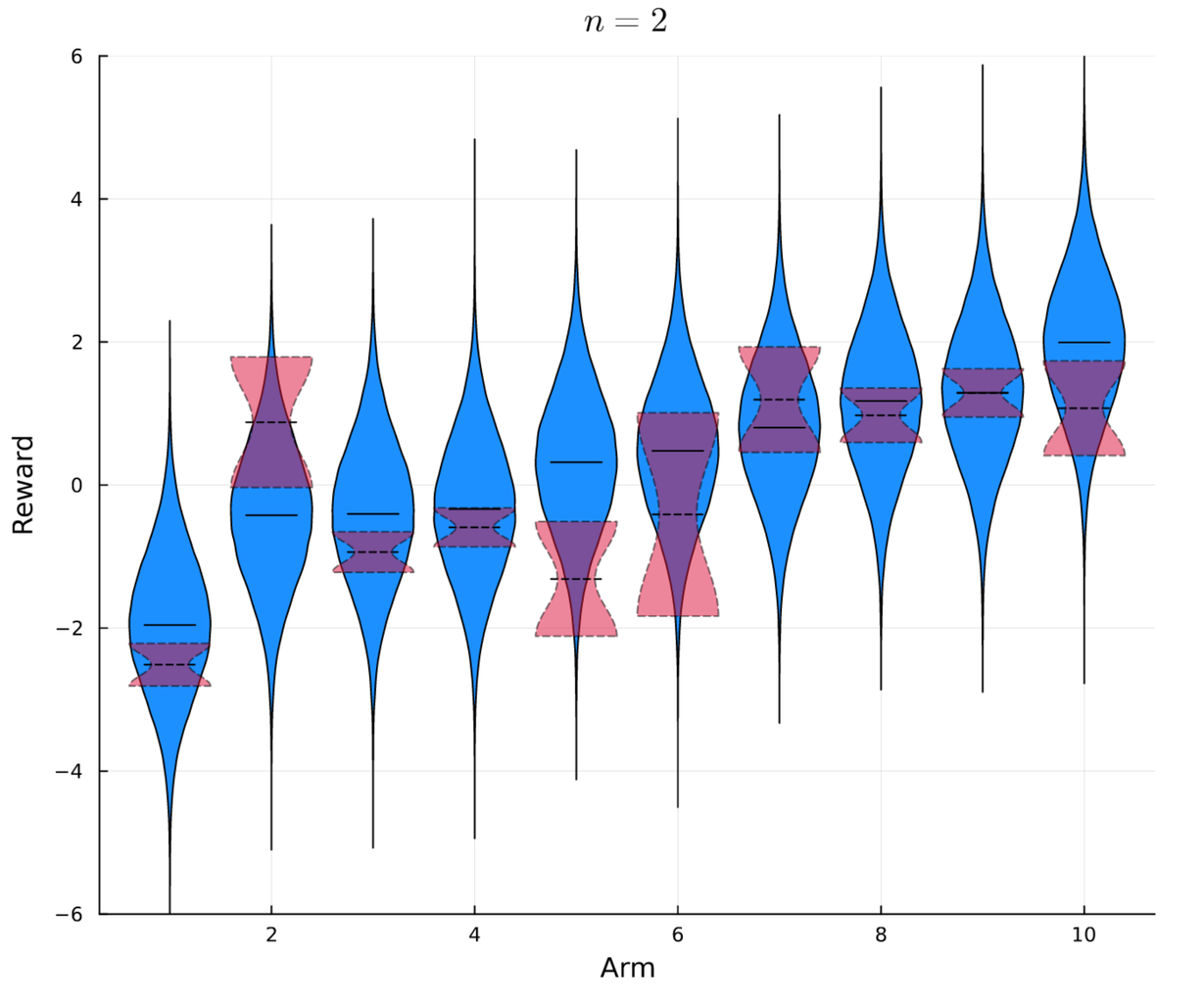
3. Take the best action from the estimate

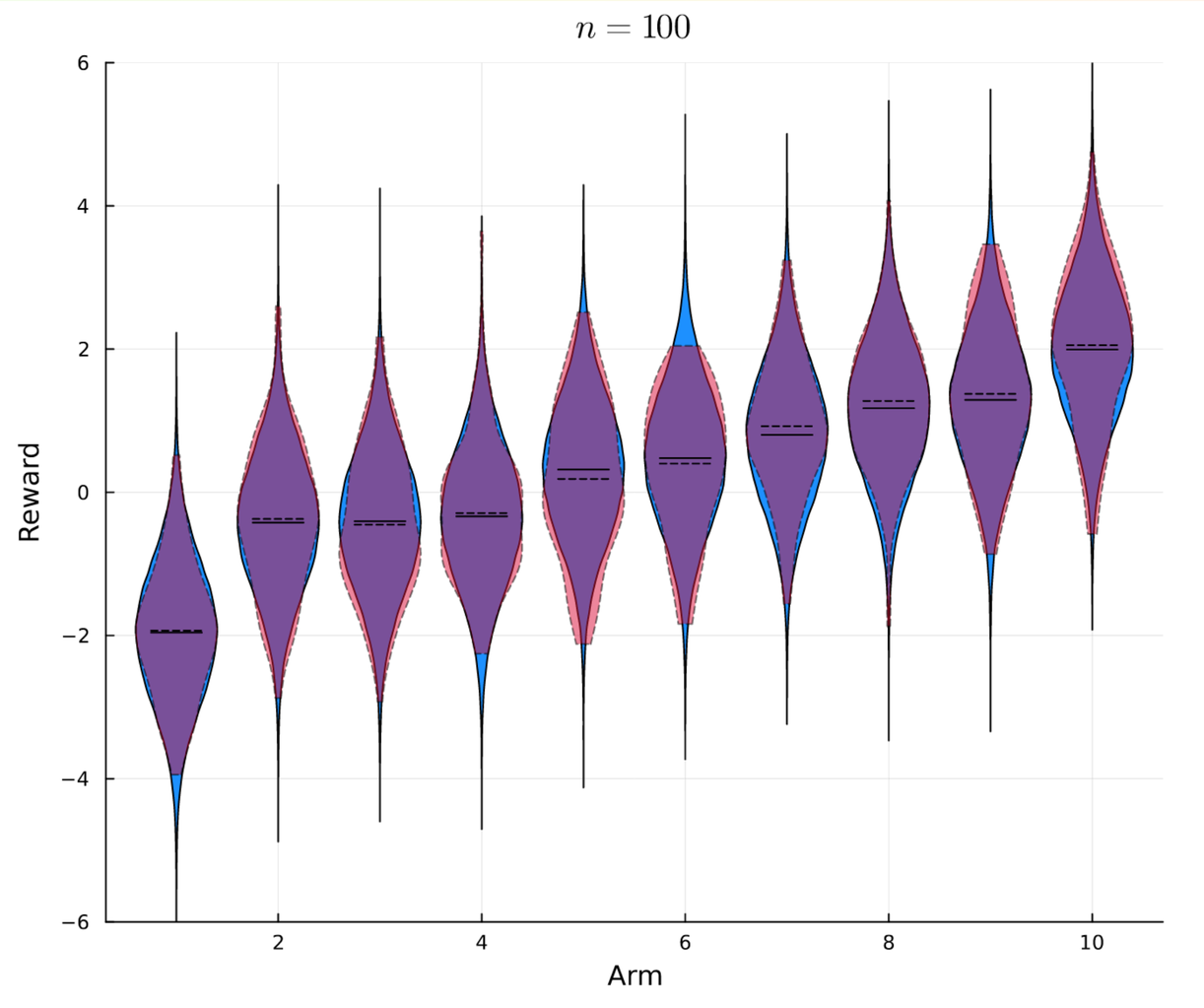
$$\arg\max_{a \in \mathcal{A}} Q(a)$$

BANDIT

CODE

BANDIT - SIMPLE ALGORITHM





BANDIT - HOW MANY SAMPLES

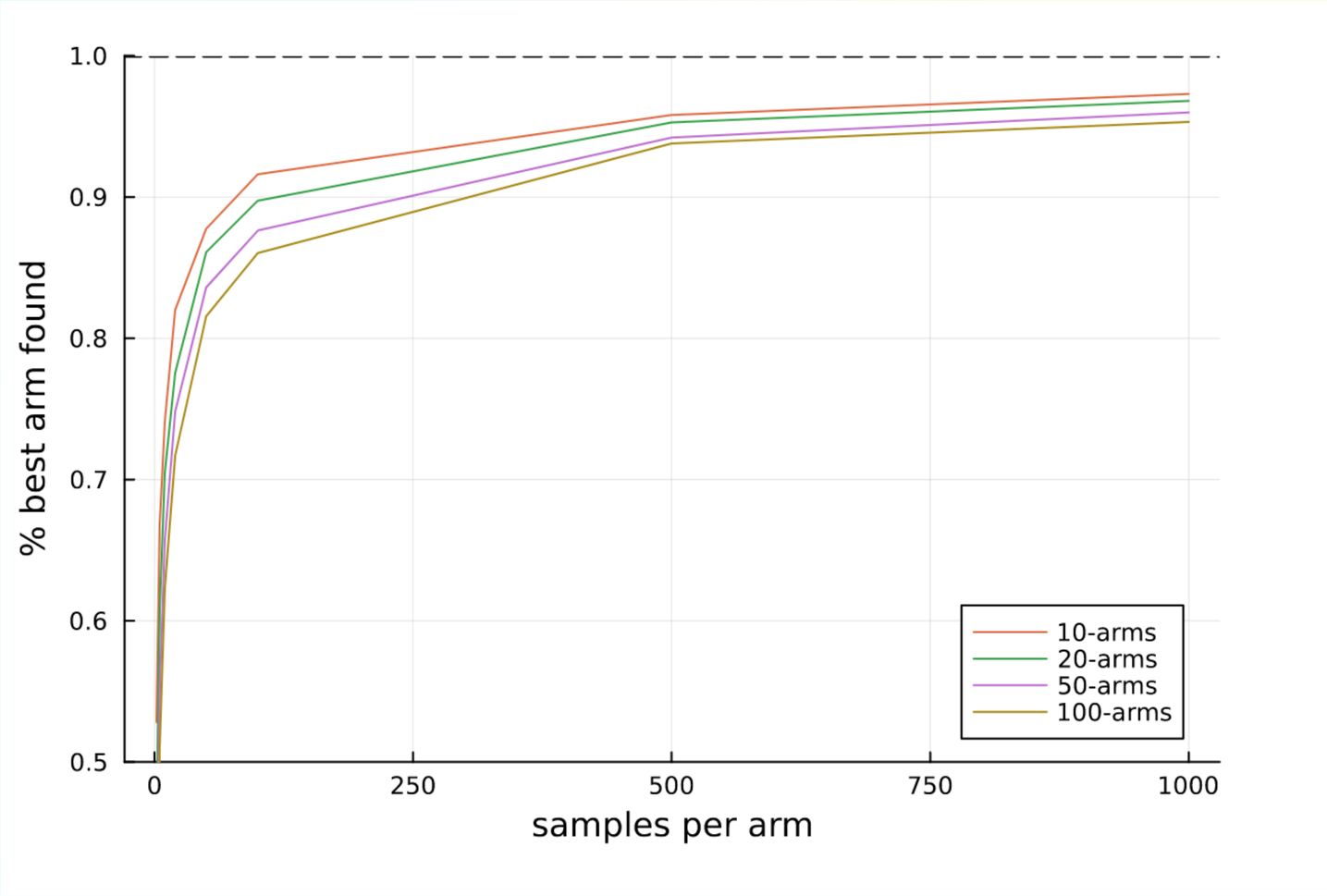
EXPERIMENT

Goal: Figure out how often the best arm is chosen after n samples

1. Create new reward distributions
2. Run algorithm
3. Determine if the best arm was found
4. Repeat 1-3 10,000 times
5. Compute % of the time the algorithm found the best arm
6. Repeat 1-5 for different values of n

BANDIT - HOW MANY SAMPLES

EXPERIMENT



BANDIT - HOW MANY SAMPLES

EXPERIMENT

Result:

1. Even with 1,000 samples, the best arm was often missed (2-5%)
2. More arms —> need more data

In general, we cannot have 100% accuracy without infinite data

ESTIMATING q_*

METHODS

Sample average:

$$Q_{n+1}(a) = \frac{1}{n}\sum_{i=1}^{n} R_i$$ — Requires storing n samples per arm

Iterative method:

$$Q_{n+1} = Q_n + \frac{1}{n}\left(R_n - Q_n\right)$$ — Stores only single scalar

Equivalent mathematically (in practice, small floating point errors)

ESTIMATING q_*

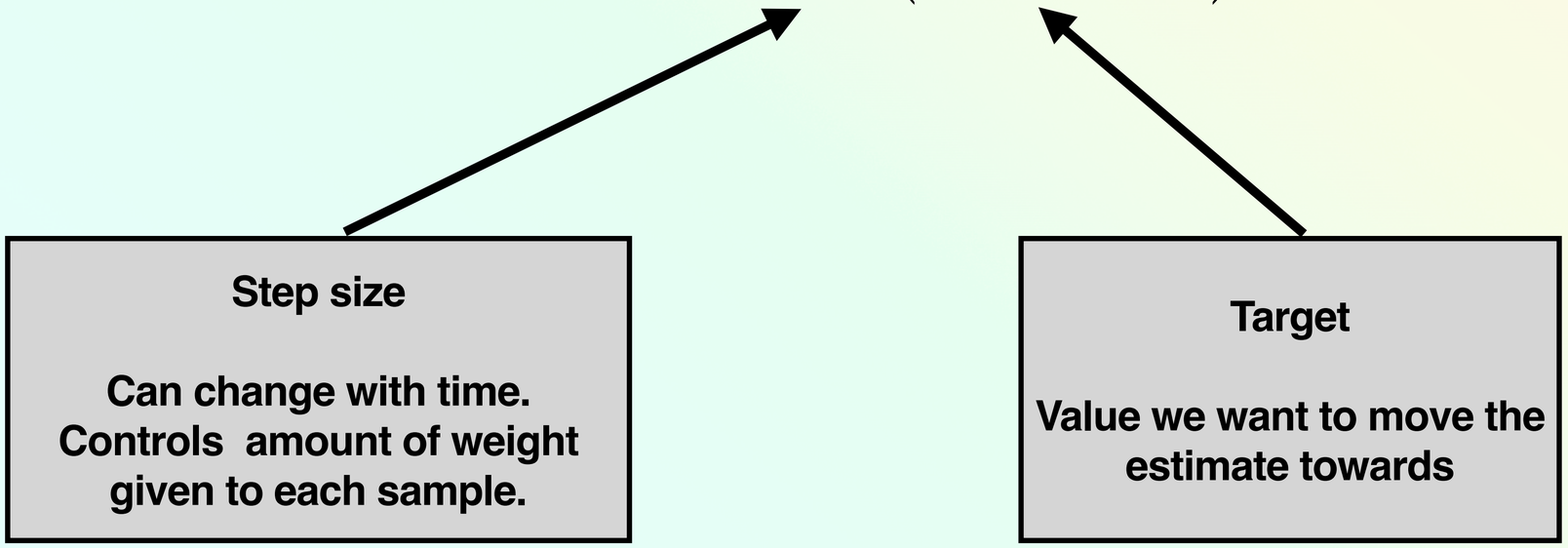
ITERATIVE UPDATE

$$
\begin{aligned}
Q_{n+1} &= \frac{1}{n}\sum_{i=1}^{n} R_i \\
&= \frac{1}{n}\left(R_n + \sum_{i=1}^{n-1} R_i\right) \\
&= \frac{1}{n}\left(R_n + (n-1)\frac{1}{n-1}\sum_{i=1}^{n-1} R_i\right) \\
&= \frac{1}{n}\left(R_n + (n-1)Q_n\right) \\
&= \frac{1}{n}\left(R_n + nQ_n - Q_n\right) \\
&= Q_n + \frac{1}{n}\left(R_n - Q_n\right)
\end{aligned}
$$

ESTIMATING q_*

ITERATIVE UPDATE

$$Q_{n+1} = Q_n + \alpha_t \left(R_n - Q_n \right)$$



ESTIMATING q_*

NONSTATIONARY

$$Q_{n+1} = Q_n + \alpha \left(R_n - Q_n \right)$$

- Constant step size allows for learning to keep happening
- Never converges to the mean
- It is essential when the reward distribution changes over time
  - It is also useful when we switch to multi-step problems

ESTIMATING q_*

CHOOSING A CONSTANT STEP SIZE

$$Q_{n+1} = Q_n + \alpha\left(R_n - Q_n\right)$$

- Constant step size puts a larger weight on more recent samples

$$\begin{aligned}
Q_{n+1} &= Q_n + \alpha\left(R_n - Q_n\right) \\
&= \alpha R_n - (1-\alpha)Q_n \\
&= \alpha R_n - (1-\alpha)\left(\alpha R_{n-1} + (1-\alpha)Q_{n-1}\right) \\
&= \alpha R_n - (1-\alpha)\alpha R_{n-1} + (1-\alpha)^2 Q_{n-1} \\
&= \alpha R_n - (1-\alpha)\alpha R_{n-1} + (1-\alpha)^2\alpha R_{n-2} + \ldots (1-\alpha)^{n-1}\alpha R_1 + (1-\alpha)^n Q_1 \\
&= (1-\alpha)^n Q_1 + \sum_{i=1}^{n} \alpha(1-\alpha)^{n-i} R_i
\end{aligned}$$

ESTIMATING q_*

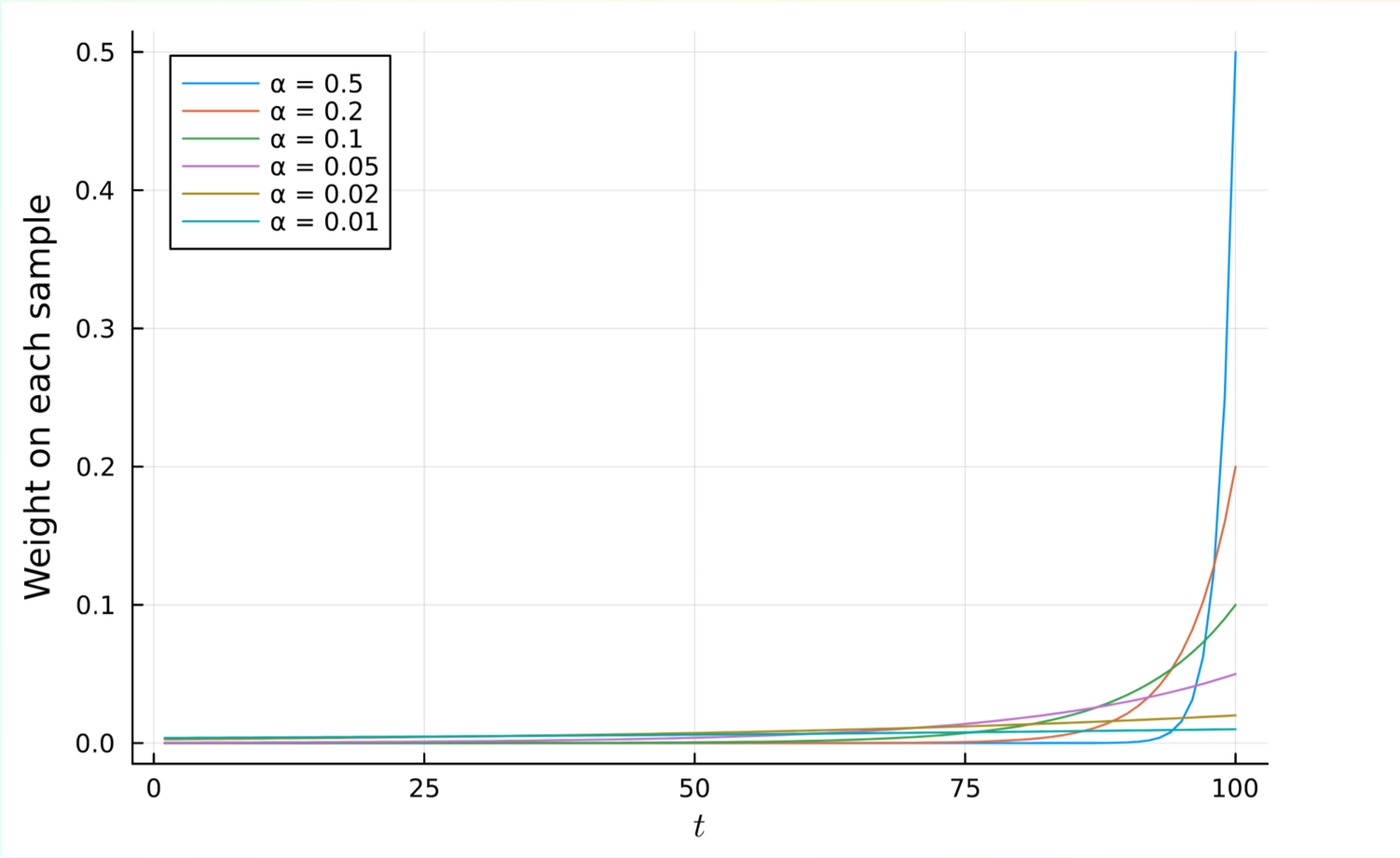
CHOOSING A CONSTANT STEP SIZE

$$(1-\alpha)^n Q_1 + \sum_{i=1}^{n} \alpha(1-\alpha)^{n-i} R_i$$

$(1-\alpha)^n \to 0$ as $n \to \infty$

Weight on each reward

- Smaller step sizes average over more samples



ESTIMATING q_*

CHOOSING A CONSTANT STEP SIZE

How do we choose a step size in practice?

Trial and error 

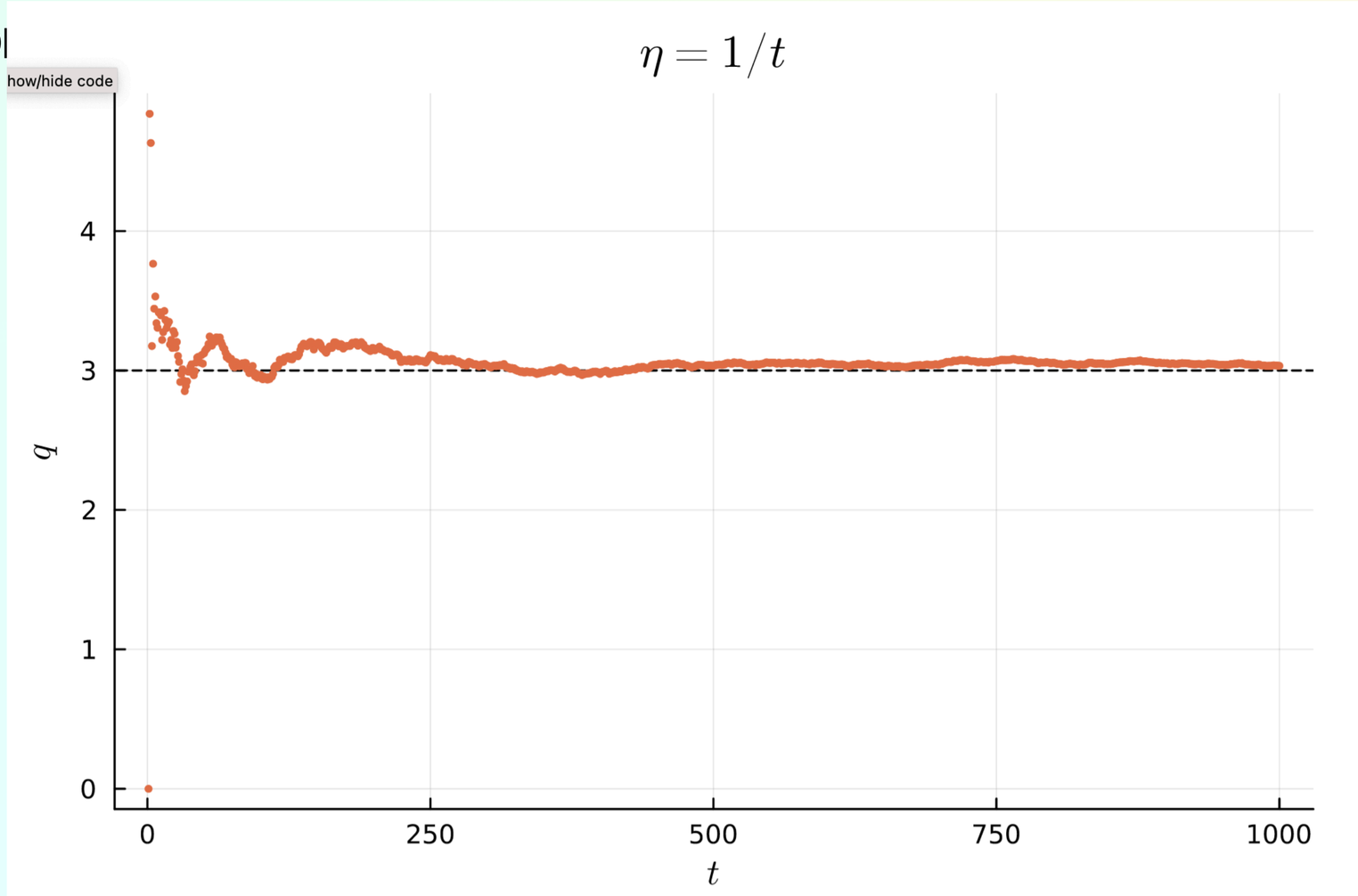
ESTIMATING q_*

CHOOSING A CONSTANT STEP SIZE

Code screen

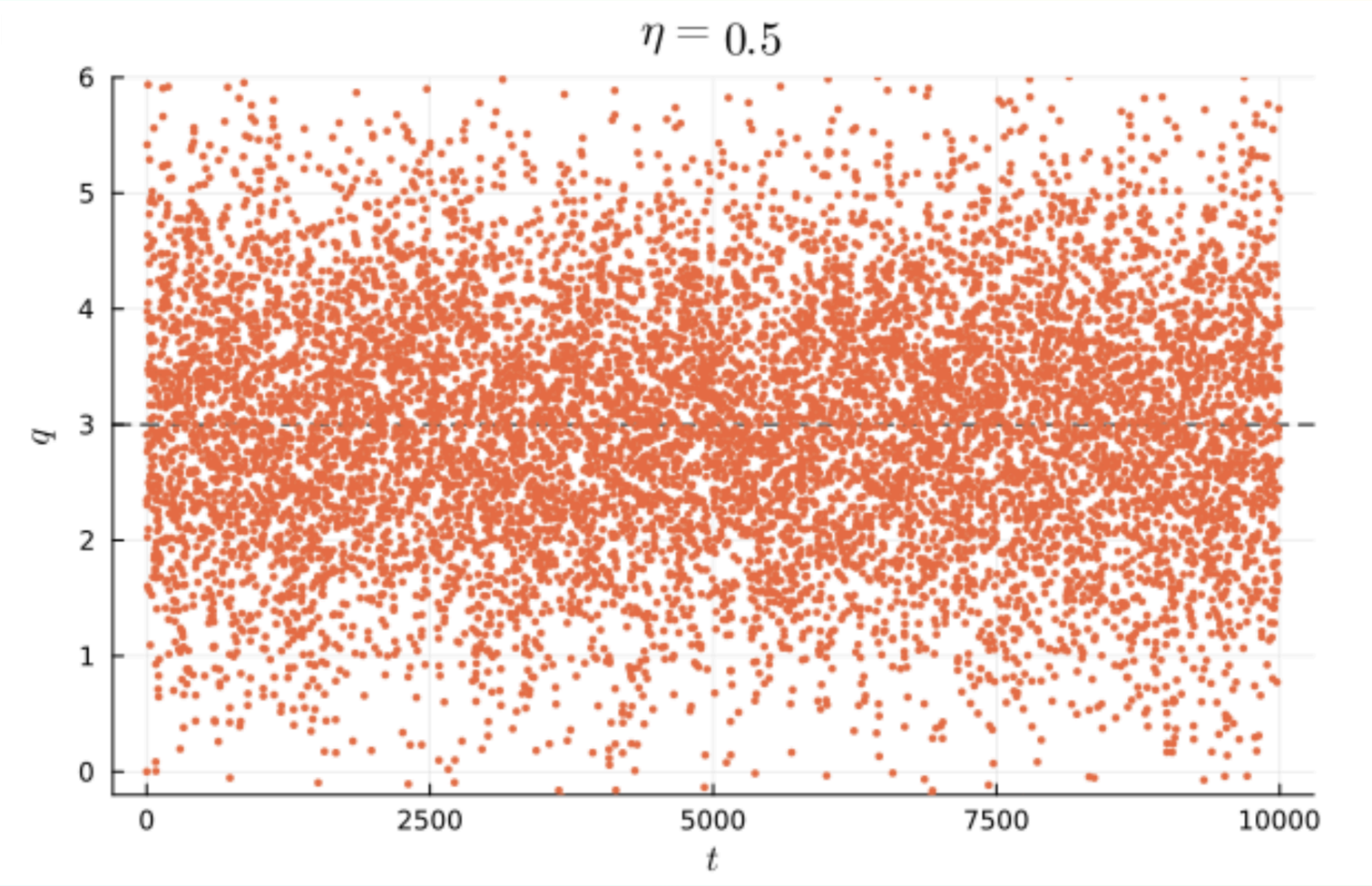
ESTIMATING q_*

CHOOSING A CO



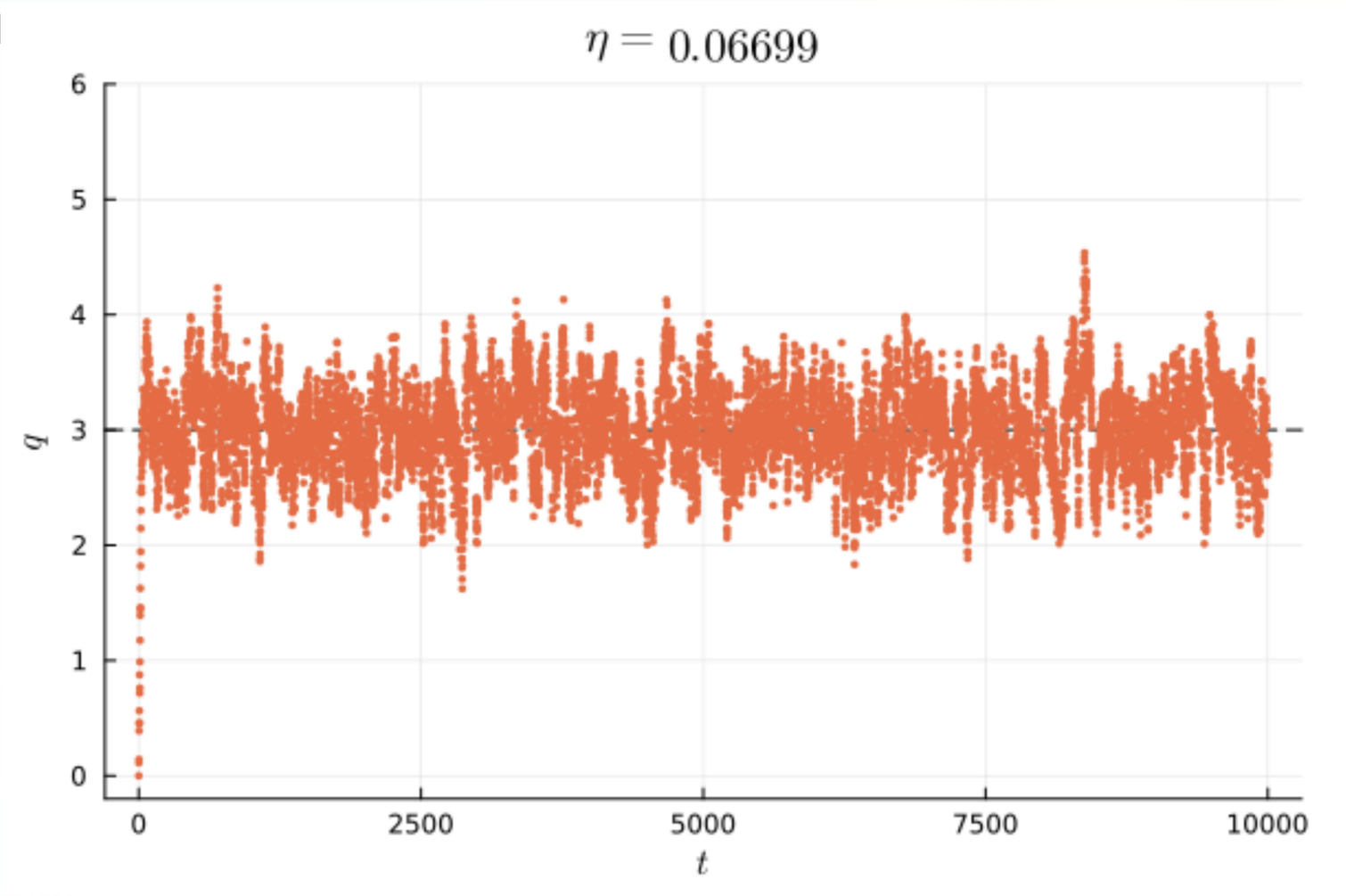
ESTIMATING q_*

CHOOSING A CO



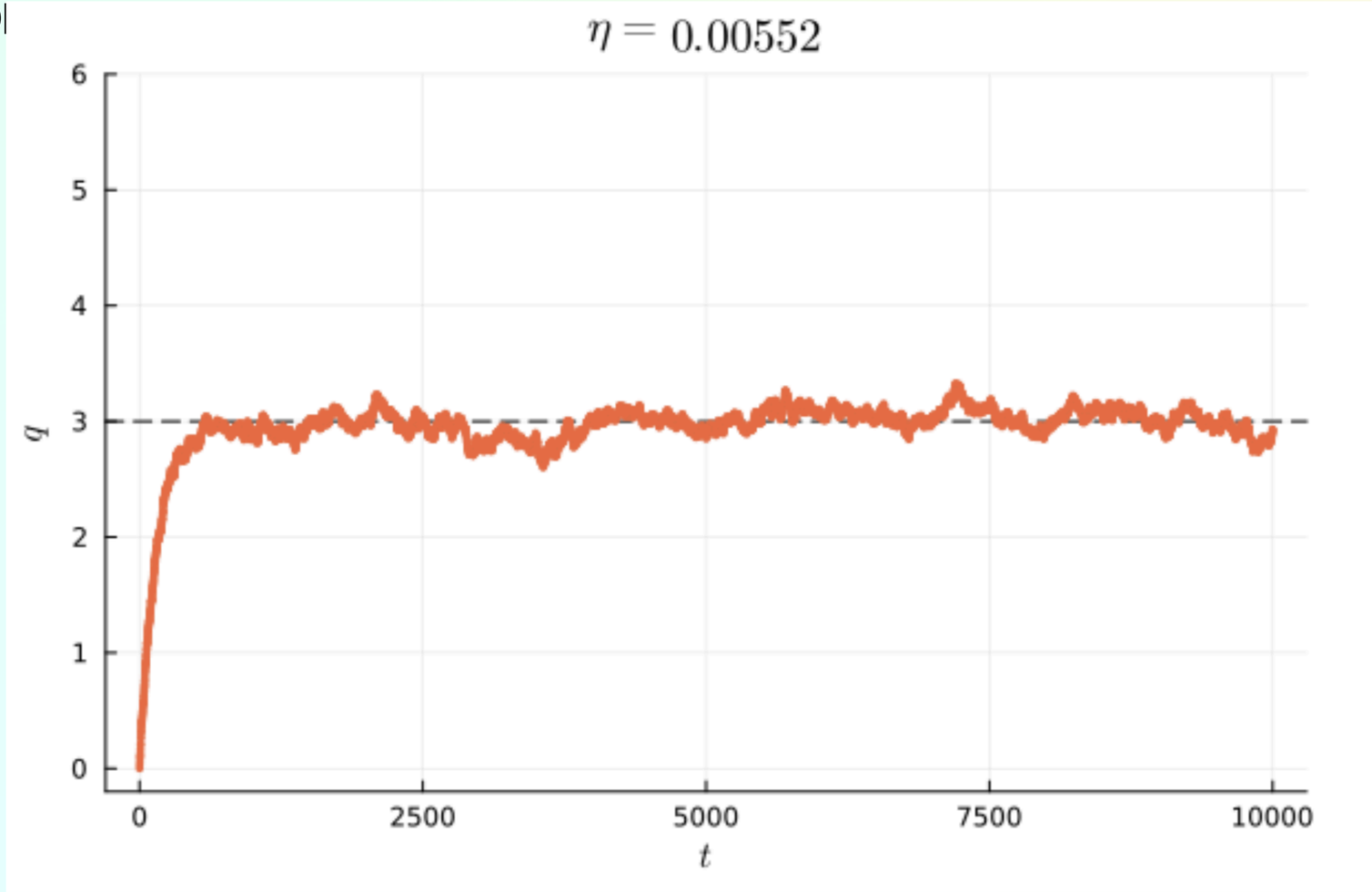
ESTIMATING q_*

CHOOSING A COI



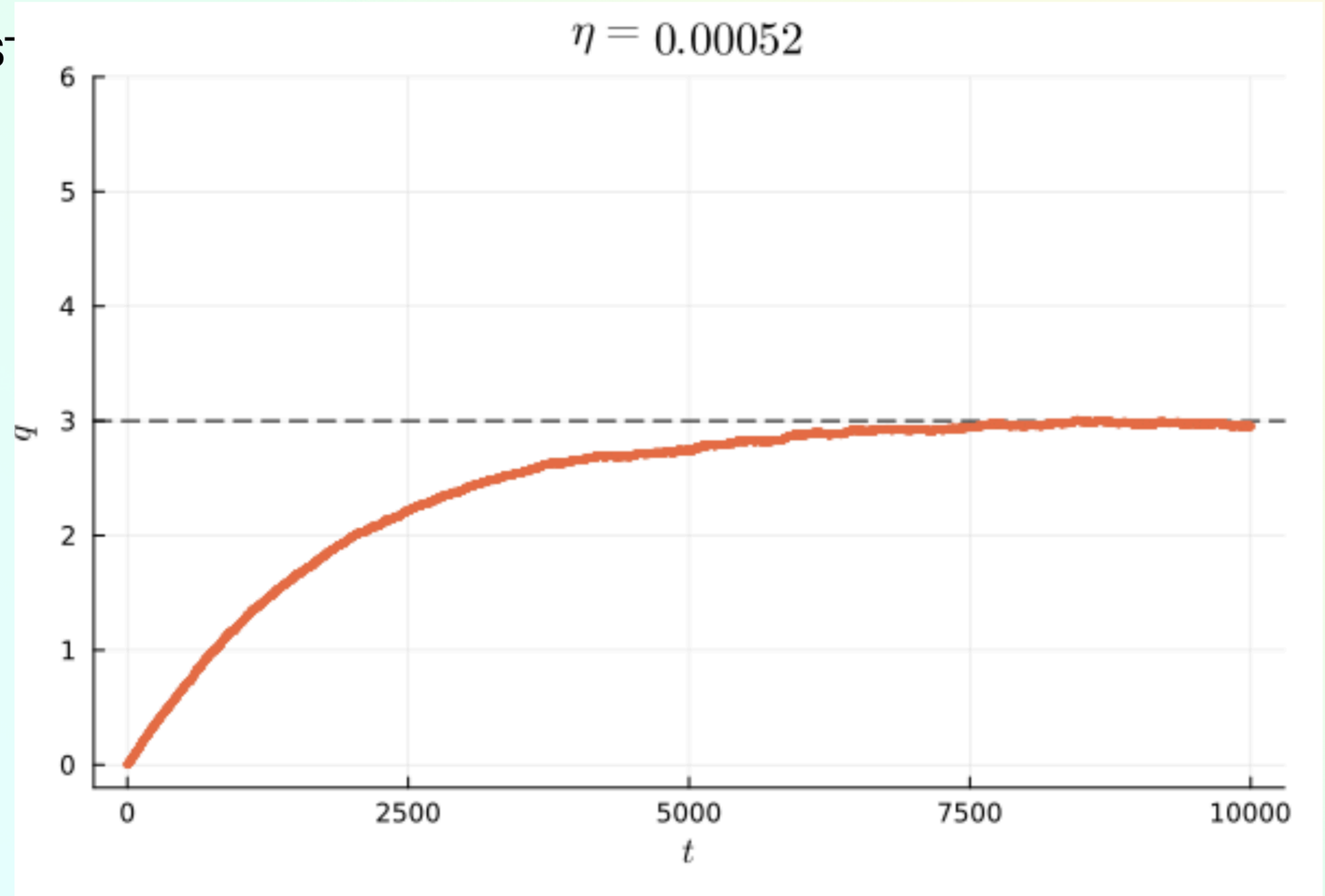
ESTIMATING q_*

CHOOSING A CO

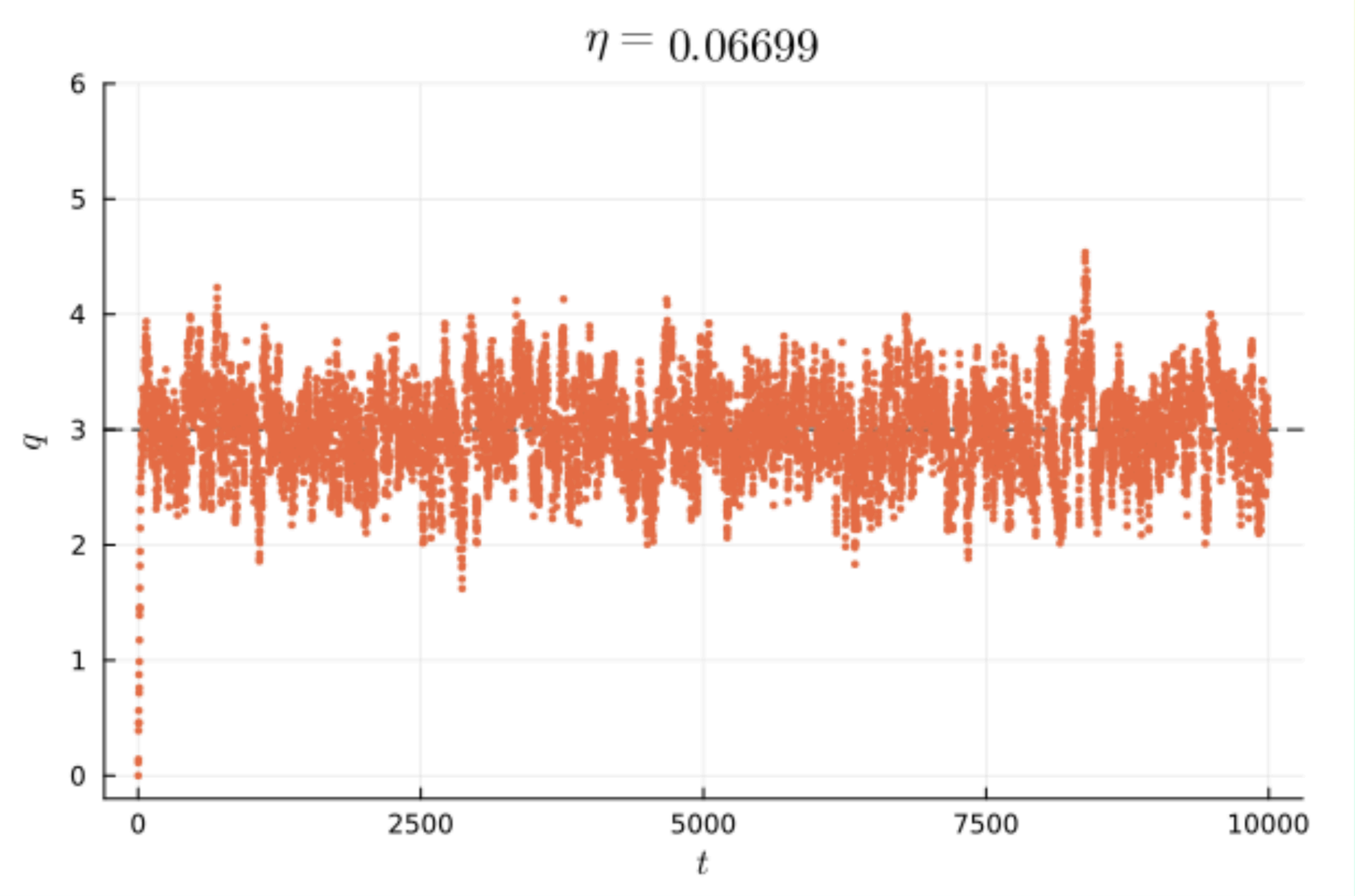


ESTIMATING q_*

CHOOSING A CONST



ESTIMATING q_*

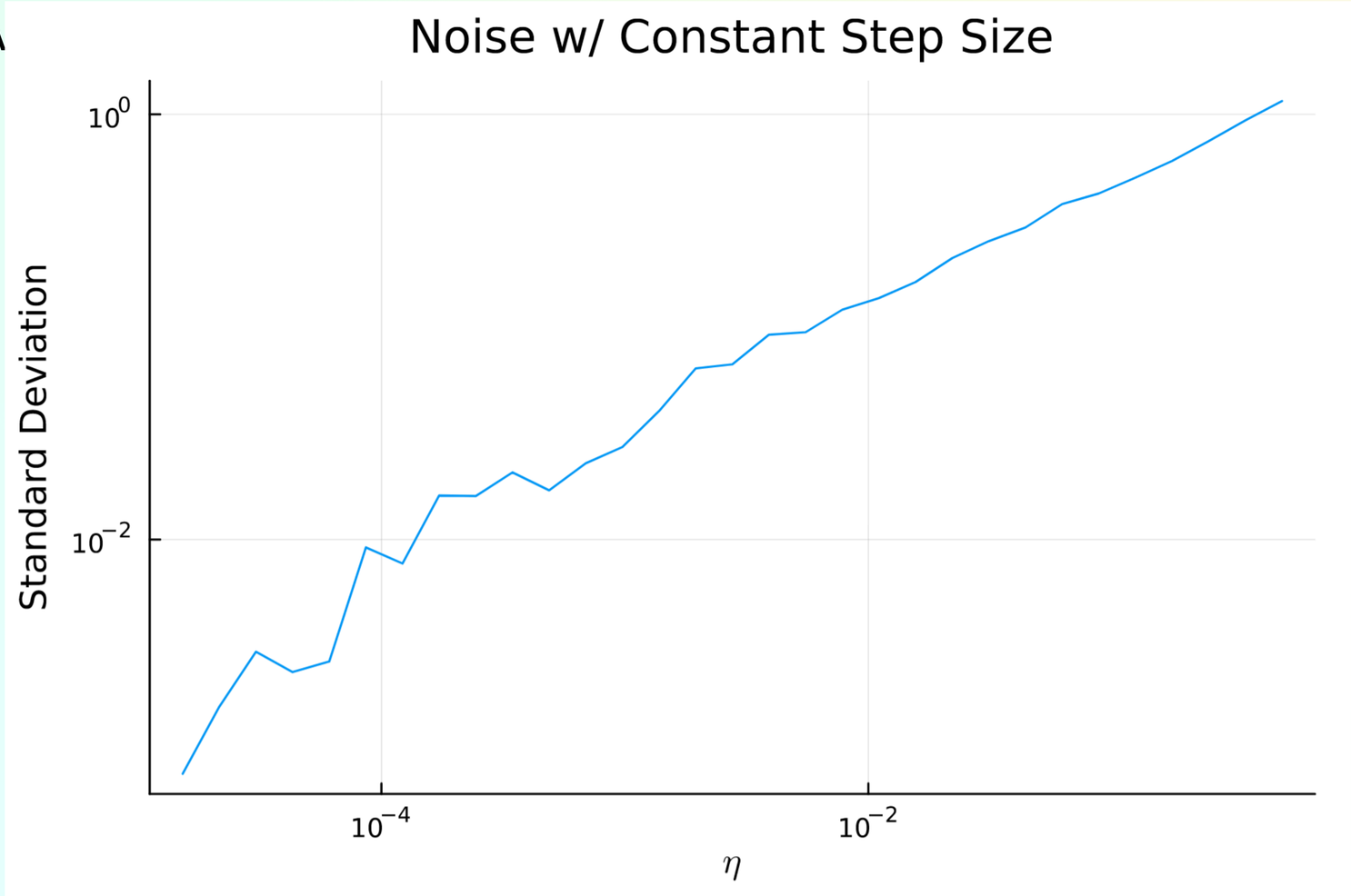


Iterations are centered around the mean.

Distance from the mean depends on the step size

ESTIMATING q_*

CHOOSING A CONSTA



ESTIMATING q_*

CHOOSING A CONSTANT STEP SIZE

What about rare events?

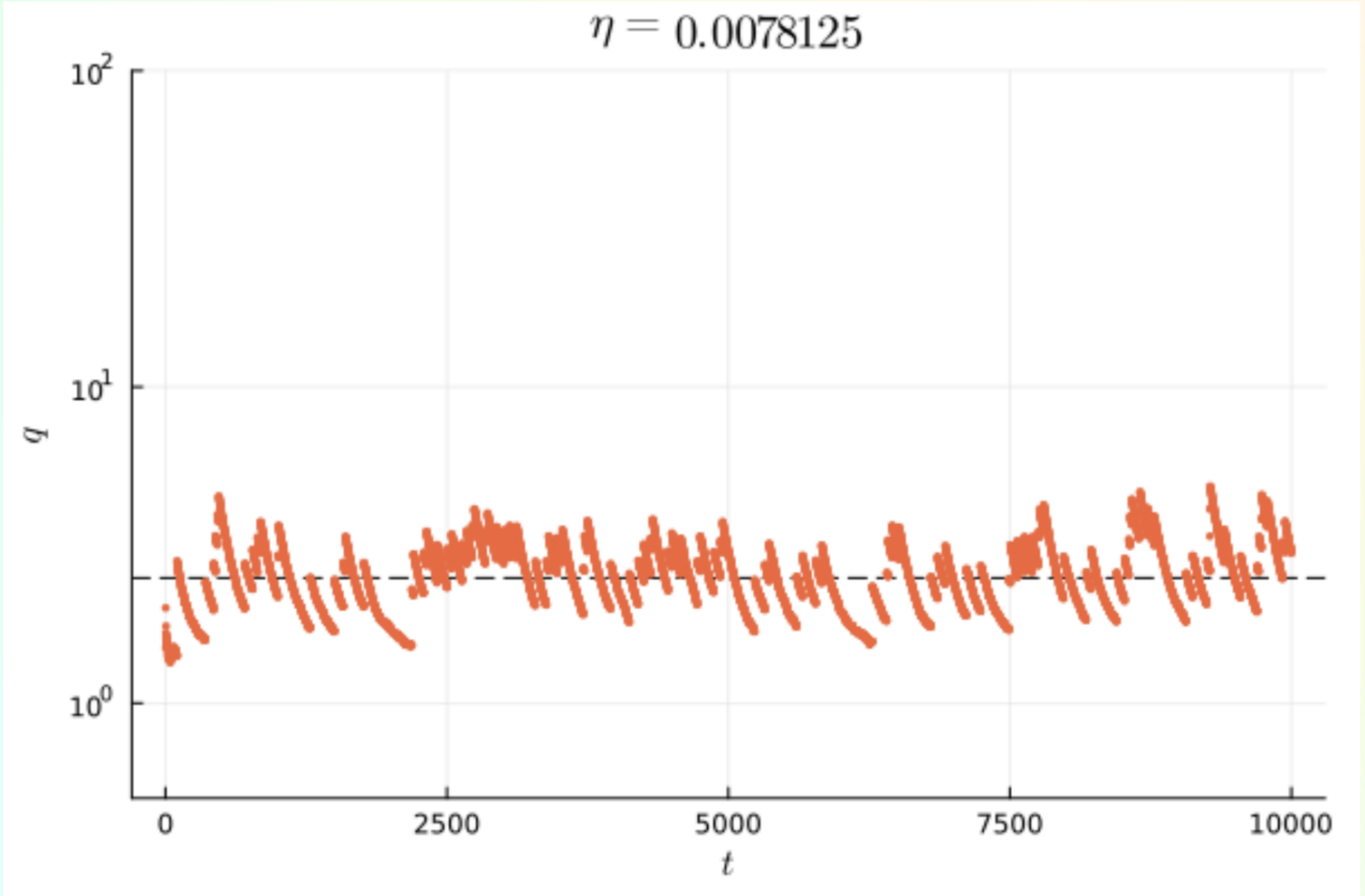
$$R_t \sim \begin{cases} 100 & p \\ 1 & (1-p)/2 \\ 2 & (1-p)/2 \end{cases}$$

ESTIMATING q_*

CHOOSING A CONSTANT STEP SIZE

What about rare events?

$$R_t \sim \begin{cases} 100 & p \\ 1 & (1-p)/2 \\ 2 & (1-p)/2 \end{cases}$$

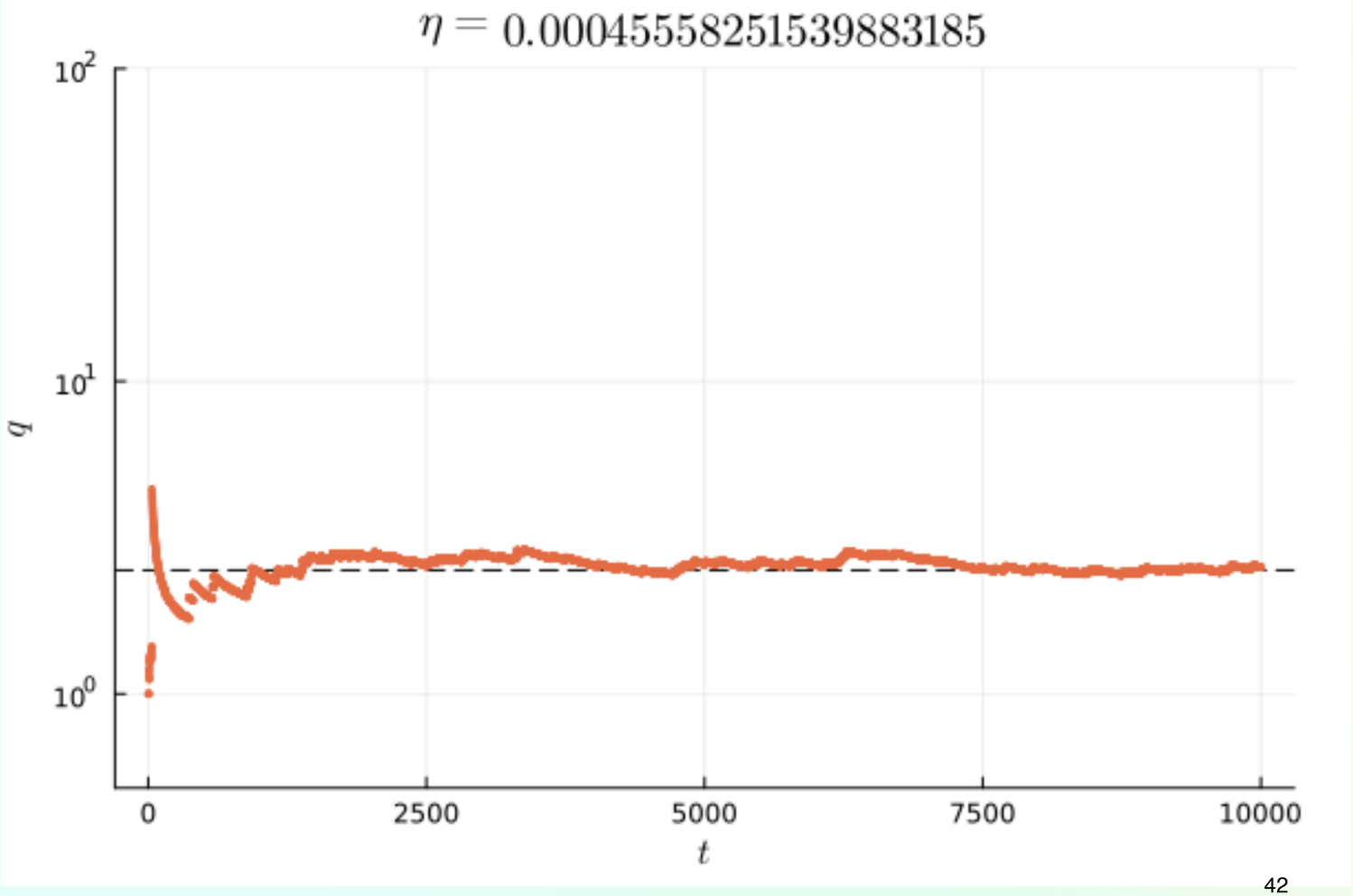


ESTIMATING q_*

CHOOSING A CONSTANT STEP SIZE

What about rare events?

$$R_t \sim \begin{cases} 100 & p \\ 1 & (1-p)/2 \\ 2 & (1-p)/2 \end{cases}$$



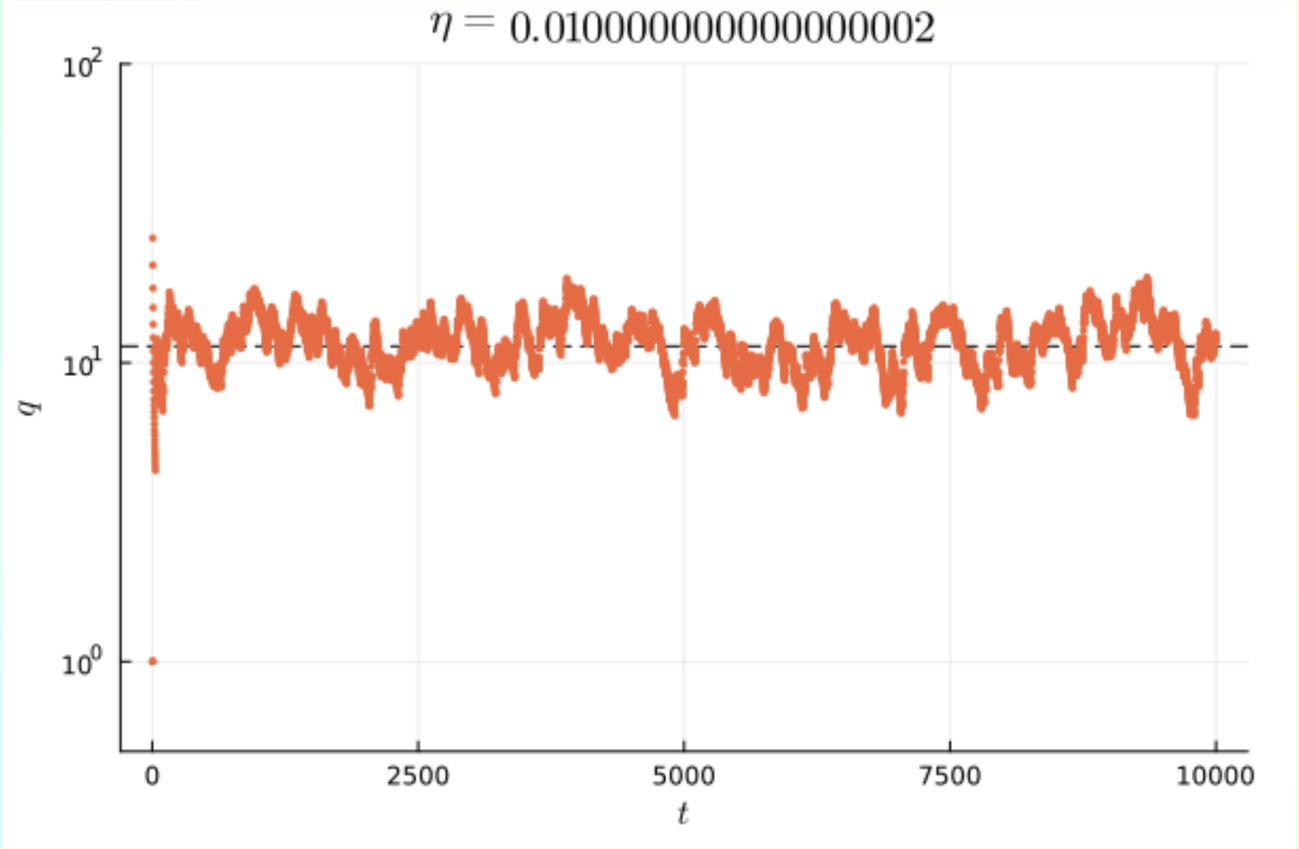
ESTIMATING q_*

CHOOSING A CONSTANT STEP SIZE

A heuristic:

$\eta = p/10$

$p = 0.1$



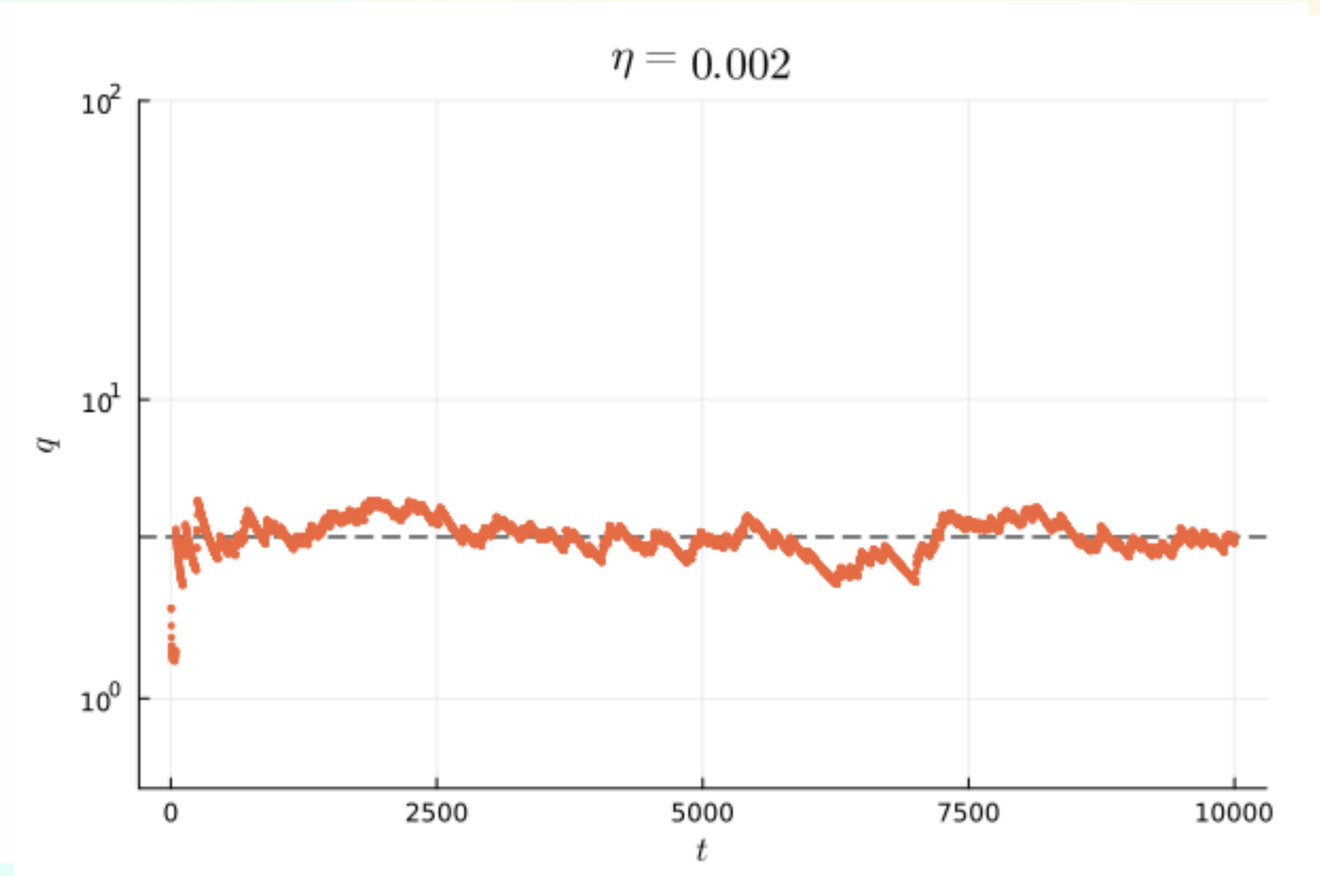
ESTIMATING q_*

CHOOSING A CONSTANT STEP SIZE

A heuristic:

$$\eta = p/10$$

$$p = 0.02$$



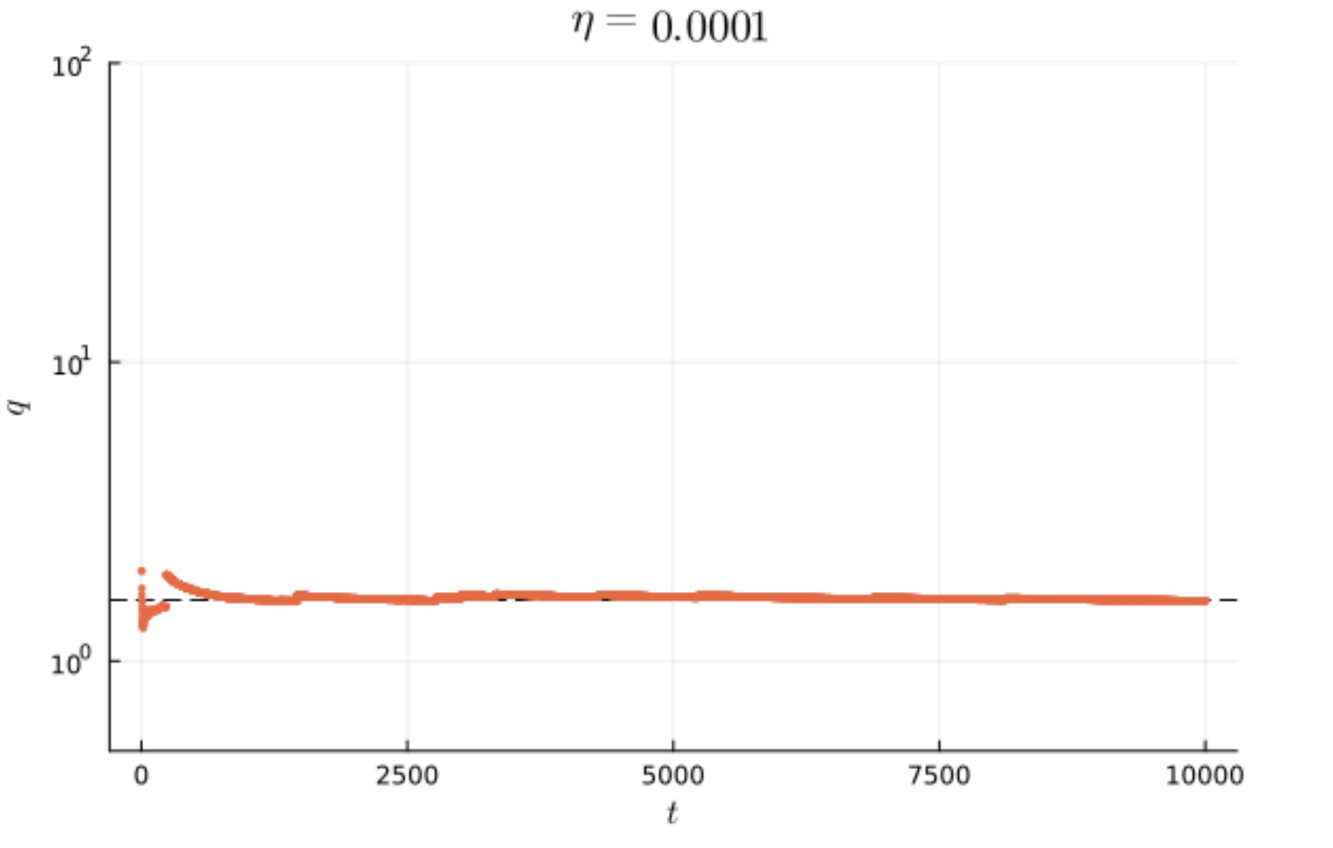
ESTIMATING q_*

CHOOSING A CONSTANT STEP SIZE

A heuristic:

$\eta = p/10$

$p = 0.001$



EXPLORATION

METHODS

Need to sample all actions to estimate their values.

Don’t have to sample all actions the same number of times.

How to choose actions:

- Random — all actions treated as equally interesting
- ϵ-greedy — mostly take the believed best action. All nongreedy actions are treated as equal
- Uncertainty/UCB — Takes the action that could be the best until it is shown not to be.
- Novelty / Surprise — Rewards the agent for less frequent actions. Similar to uncertainty.

NEXT CLASS

WHAT YOU SHOULD DO

1. Bring paper & pen (or something to write on) — in-class exercises
2. The quiz is due by 11:59 pm after today's class
3. Start the programming assignment (Due 11:59pm Wednesday)

Monday: In-class exercises, Q&A