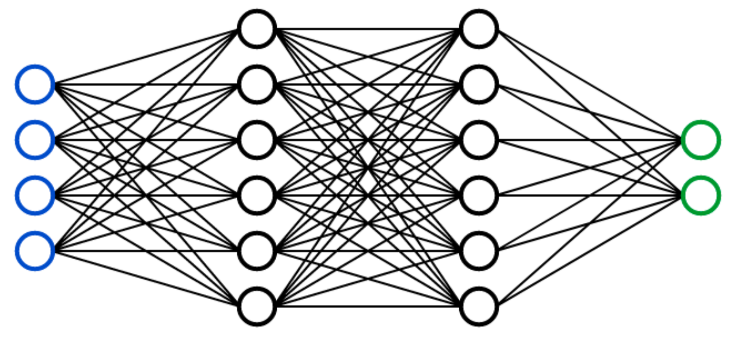
Nightmares of Policy Optimization

Sanjiban Choudhury

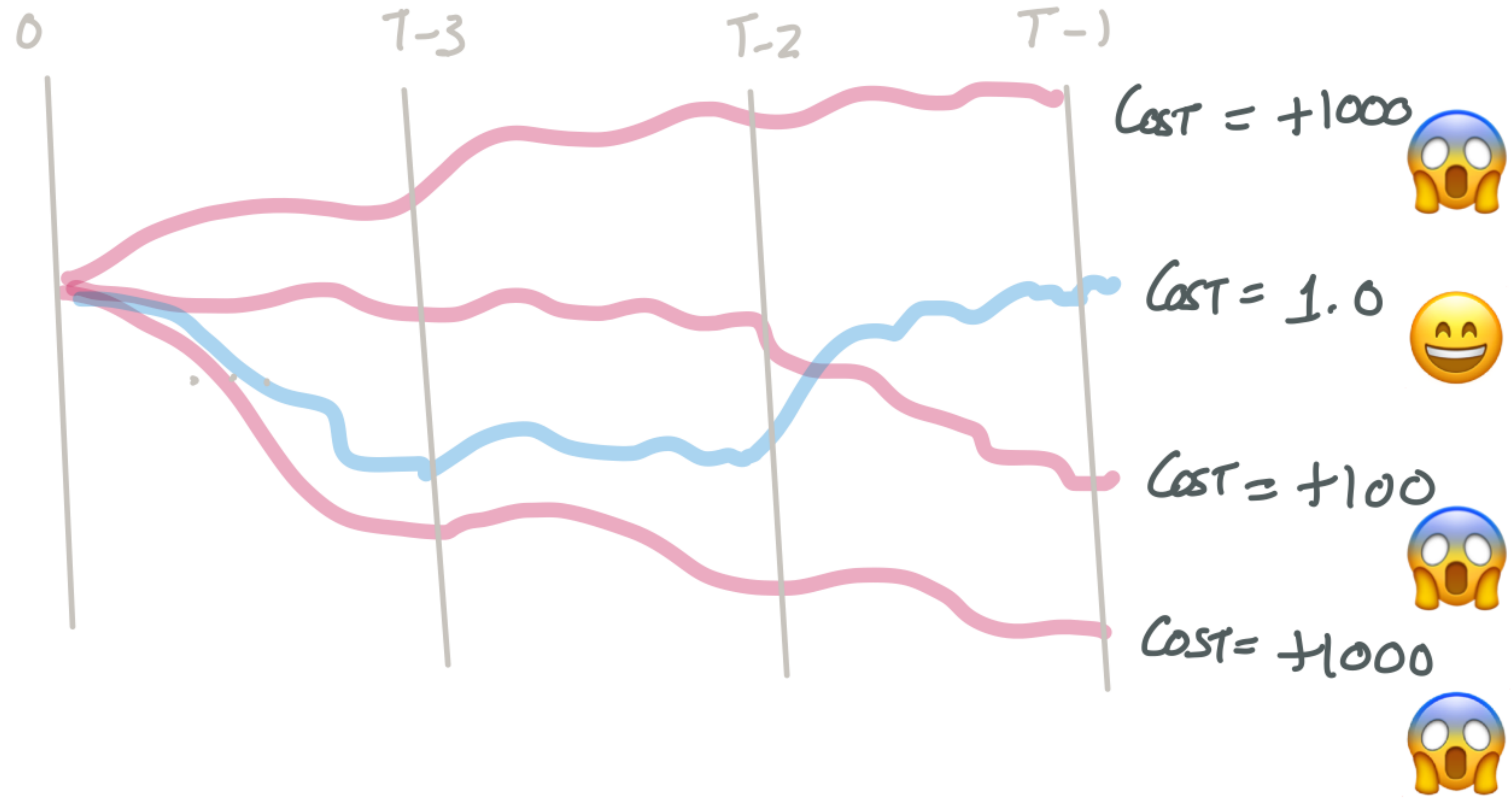
Cornell Bowers CIS
Computer Science

Can we just focus on finding a good policy?

$$\pi_\theta : s_t \rightarrow a_t$$



Learn a mapping from states to actions



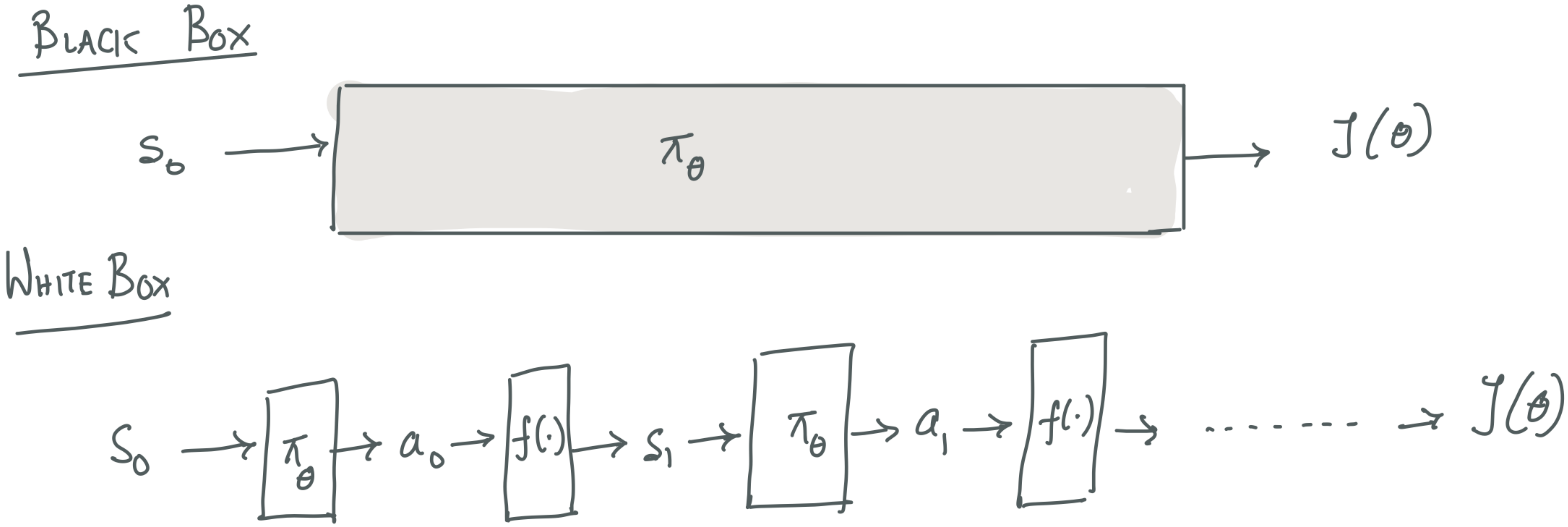
Roll-out policies in the real-world to estimate value

We assumed black-box policies ...

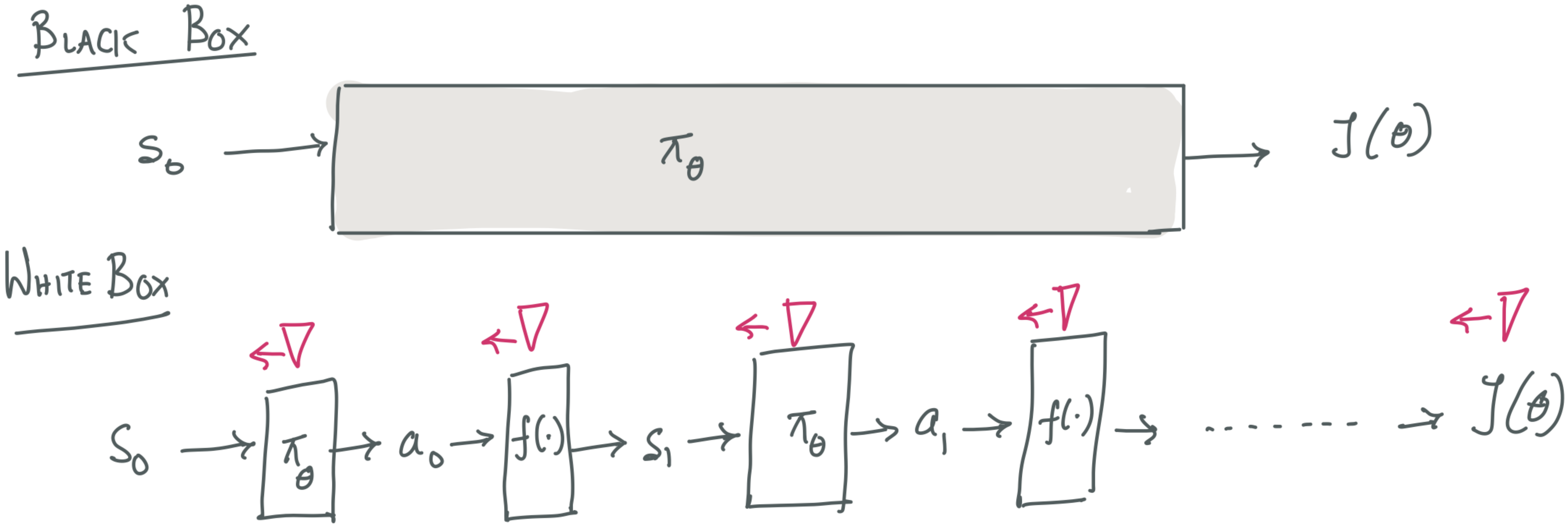
BLACK BOX


Have we redacted too much?
Form No. I
THIS CASE ORIGINATED A
REPORT MADE AT
TITLE
Mr. [redacted]
SYNOPSIS OF FACTS:
Subject
employment as a worke
chief [redacted] for t
[redacted]
DETAILS:
BACKGROUND
Birth
SUBJECT: [redacted]
1. [redacted]
of [redacted] is being s [redacted]
[redacted] and biological study at [redacted] in the areas of [redacted] work
until [redacted], [redacted]. The pr [redacted]
2. This [redacted]
c [redacted] d
into the [redacted] l effects of [redacted]
continue to be D [redacted] areas, [redacted] load
[redacted], [redacted] ors will
[redacted] and [redacted]
3. The [redacted]
[redacted] The [redacted] estimated to in the
[redacted] equired, under [redacted]

Black-box vs White-box vs Gray-box



Black-box vs White-box vs Gray-box



How can we take gradients if we don't know the dynamics?

The Likelihood Ratio Trick!

REINFORCE

Algorithm 20: The REINFORCE algorithm.

Start with an arbitrary initial policy π_θ

while *not converged* **do**

Run simulator with π_θ to collect $\{\xi^{(i)}\}_{i=1}^{N}$

Compute estimated gradient

$$\widetilde{\nabla}_\theta J = \frac{1}{N} \sum_{i=1}^{N} \left[\left(\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta \left(a_t^{(i)} | s_t^{(i)} \right) \right) R(\xi^{(i)}) \right]$$

Update parameters $\theta \leftarrow \theta + \alpha \, \widetilde{\nabla}_\theta J$

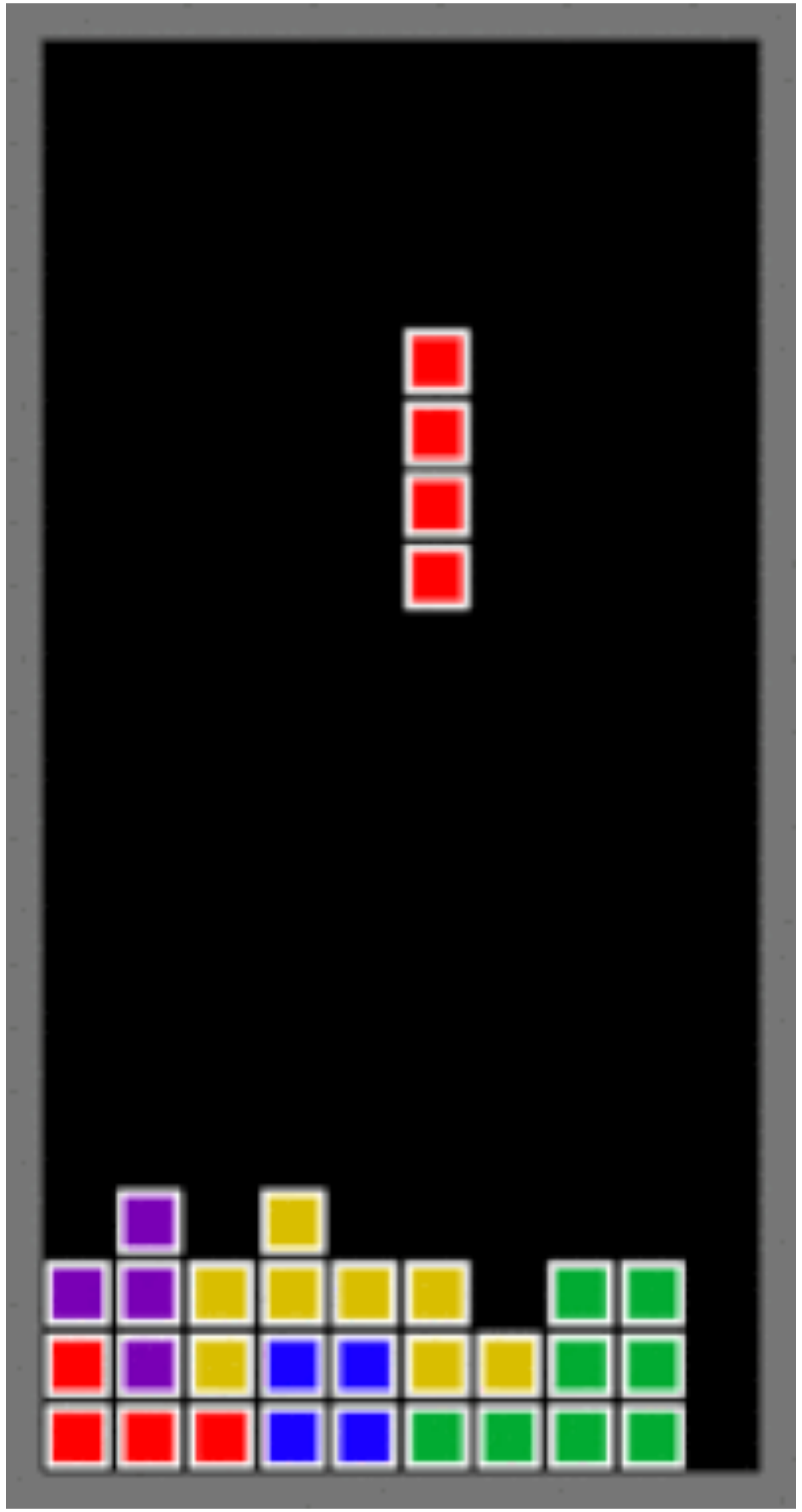
return π_θ

Tetris Policy

$$\pi_\theta(a|s) = \frac{\exp\left(\theta^\top f(s,a)\right)}{\sum_{a'} \exp\left(\theta^\top f(s,a')\right)}$$

$f_1(s,a) = \#$ number of holes

$f_2(s,a) = \#$ max height

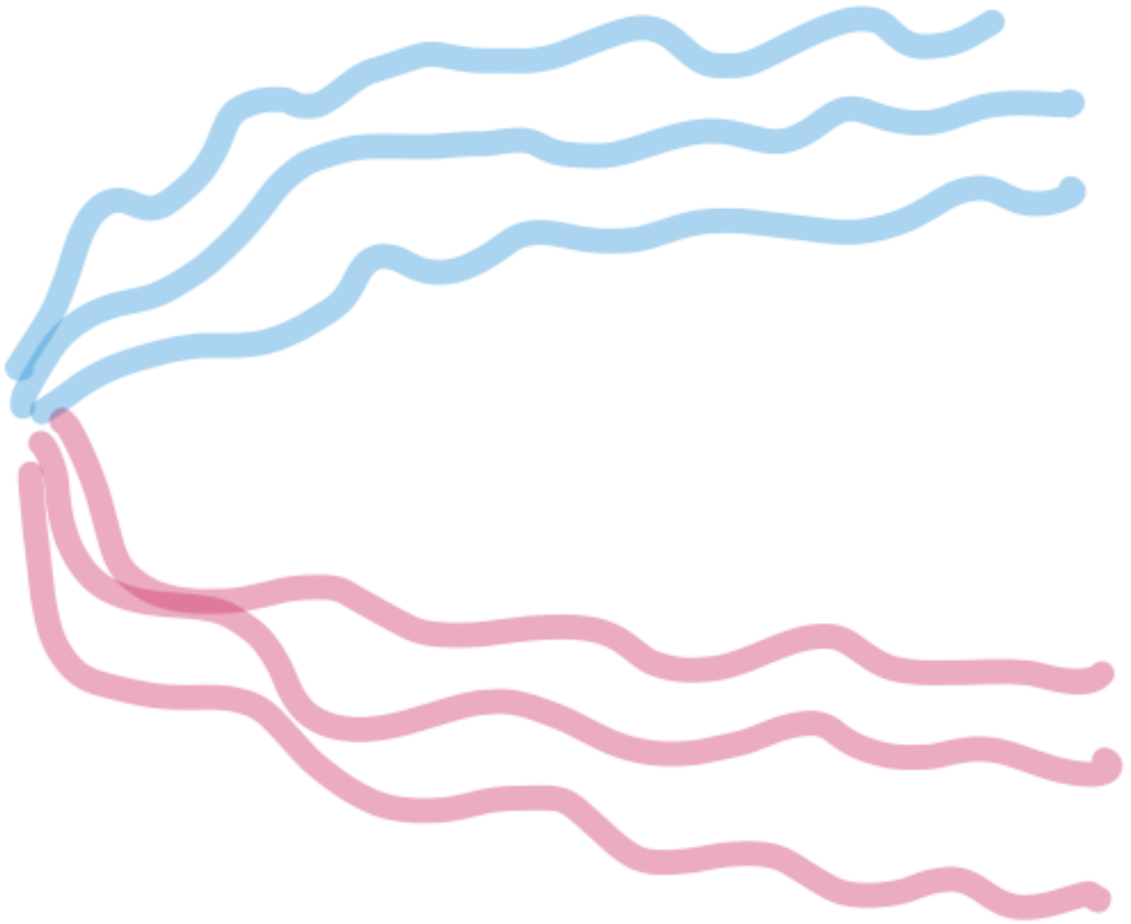


Chugging through the gradient ..

$$
\begin{aligned}
\nabla_\theta \log \pi_\theta(a|s) &= \nabla_\theta \left[\theta^\top f(s,a) - \log \sum_{a'} \exp\left(\theta^\top f(s,a')\right) \right] \\
&= f(s,a) - \frac{\sum_{a'} f(s,a') \exp\left(\theta^\top f(s,a')\right)}{\sum_{a'} \exp\left(\theta^\top f(s,a')\right)} \\
&= f(s,a) - \sum_{a'} f(s,a')\, \pi_\theta\left(a'|s\right) \\
&= f(s,a) - E_{\pi_\theta(a'|s)}\left[f(s,a')\right]
\end{aligned}
$$

Understanding the REINFORCE update

LET $f_1(s,a) = \#$ holes.



$$\left.\begin{array}{l} R=+1 \\ R=+1 \\ R=+1 \end{array}\right\} \Rightarrow \sum_{(s,a)\in\tau} \underbrace{f_1(s,a)}_{\text{Feature}} - \underbrace{\mathop{\mathbb{E}}_{a'\sim\pi_\theta} f_1(s,a')}_{\text{Average feature}} = -5$$

$$\left.\begin{array}{l} R=-1 \\ R=-1 \\ R=-1 \end{array}\right\} \Rightarrow \sum_{(s,a)\in\tau} \underbrace{f_1(s,a)}_{\text{Feature}} - \underbrace{\mathop{\mathbb{E}}_{a'\sim\pi_\theta} f_1(s,a')}_{\text{Average feature}} = +3$$

$$\theta_1 = \theta_1 + \sum_{(s,a)\in\tau}\left(\nabla_\theta \log \pi_\theta(a|s)\right) R(\tau) = \theta_1 + \alpha\left(-5\times(+1) + 3(-1)\right)$$

$$= \theta_1 - \alpha\, 8 \quad \left(\text{Bump down this feature}\right)$$

REINFORCE

Algorithm 20: The REINFORCE algorithm.

Start with an arbitrary initial policy π_θ

while *not converged* **do**

  Run simulator with π_θ to collect $\{\xi^{(i)}\}_{i=1}^{N}$

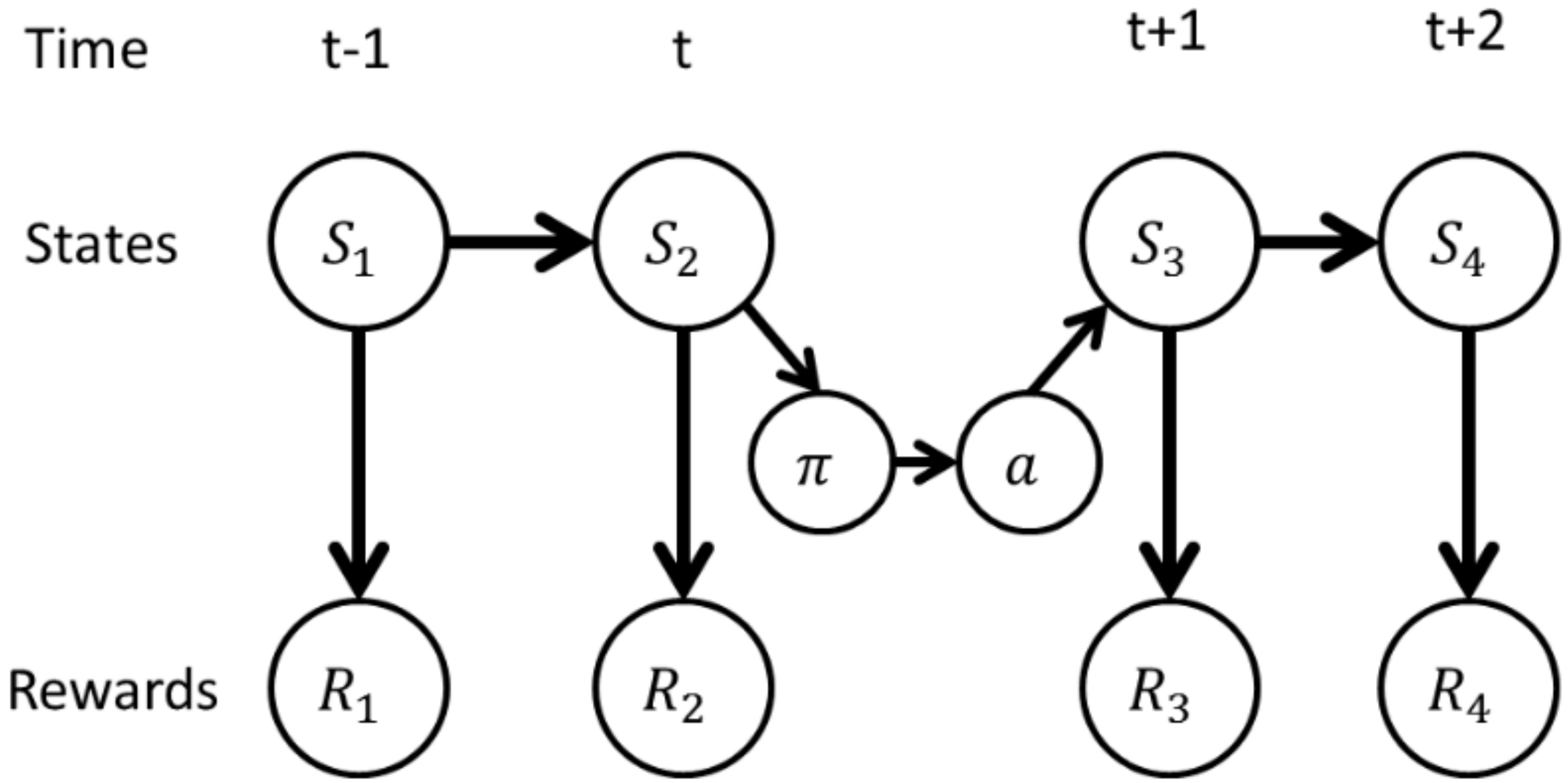
  Compute estimated gradient

$$\widetilde{\nabla}_\theta J = \frac{1}{N} \sum_{i=1}^{N} \left[\left(\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta \left(a_t^{(i)} | s_t^{(i)} \right) \right) R(\xi^{(i)}) \right]$$

  Update parameters $\theta \leftarrow \theta + \alpha \, \widetilde{\nabla}_\theta J$

return π_θ

Causality: Can actions affect the past?



The Policy Gradient Theorem

$$\begin{aligned}\nabla_\theta J &= E_{p(\xi|\theta)}\left[\sum_{t=0}^{T-1}\left(\nabla_\theta \log \pi_\theta(a_t|s_t)\left(\sum_{t'=0}^{t-1} r(s_{t'},a_{t'}) + \sum_{t'=t}^{T-1} r(s_{t'},a_{t'})\right)\right)\right]\\ &= E_{p(\xi|\theta)}\left[\sum_{t=0}^{T-1}\left(\nabla_\theta \log \pi_\theta(a_t|s_t)\sum_{t'=t}^{T-1} r(s_{t'},a_{t'})\right)\right],\end{aligned}$$

$$\nabla_\theta J = E_{p(\xi|\theta)}\left[\sum_{t=0}^{T-1}\nabla_\theta \log \pi_\theta(a_t|s_t)\, Q^{\pi_\theta}(s_t,a_t)\right]$$

Life is good!

This solves everything ...

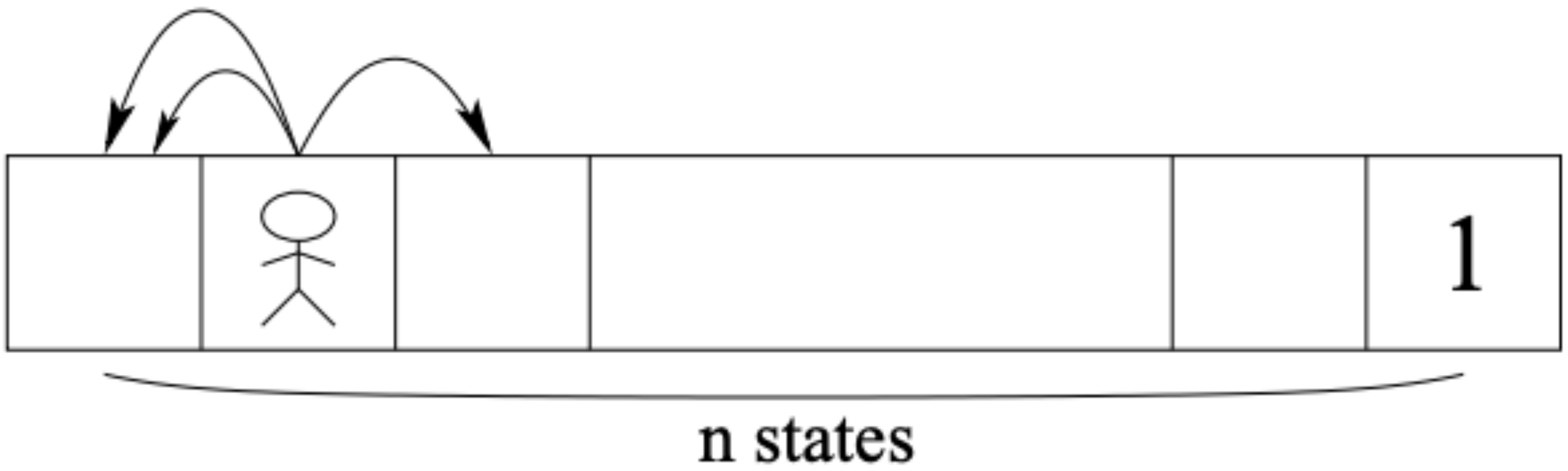
The Three Nightmares of Policy Optimization

Nightmare 1:

Local Optima

Activity!

Consider the following MDP



Let's say I picked actions uniformly.

How long would it take me to get to the state with reward=1?

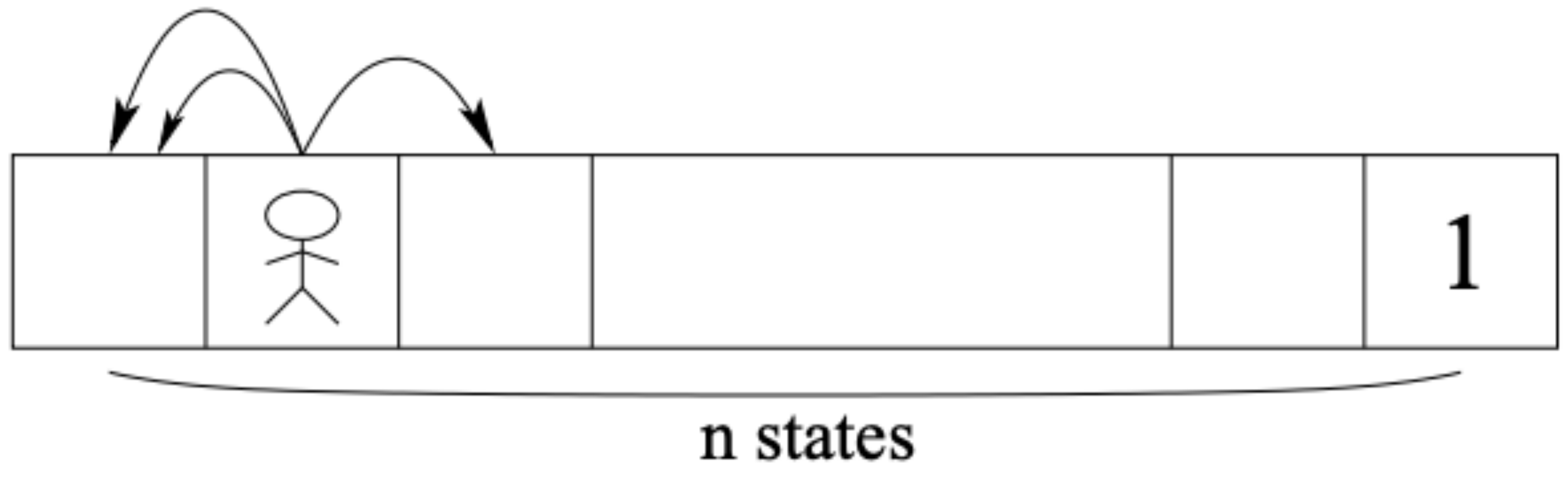
From Kakade and Langford

Think-Pair-Share

Think (30 sec): How long would it take me to get to the state with reward = 1? What does this imply if I run policy gradients?

Pair: Find a partner

Share (45 sec): Partners exchange ideas

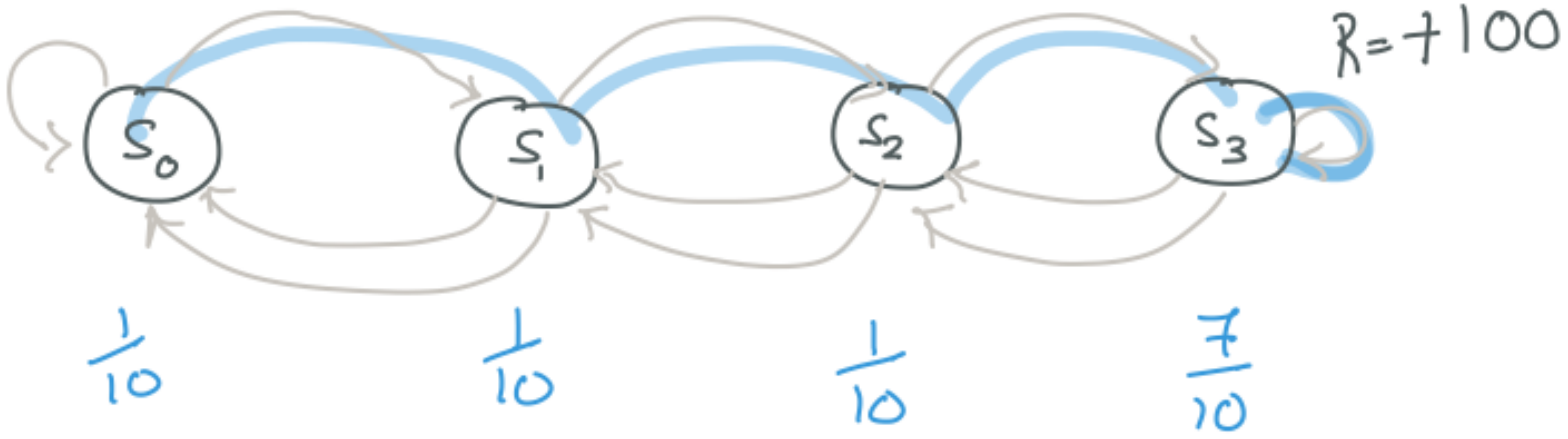


Problem: Lack of exploration

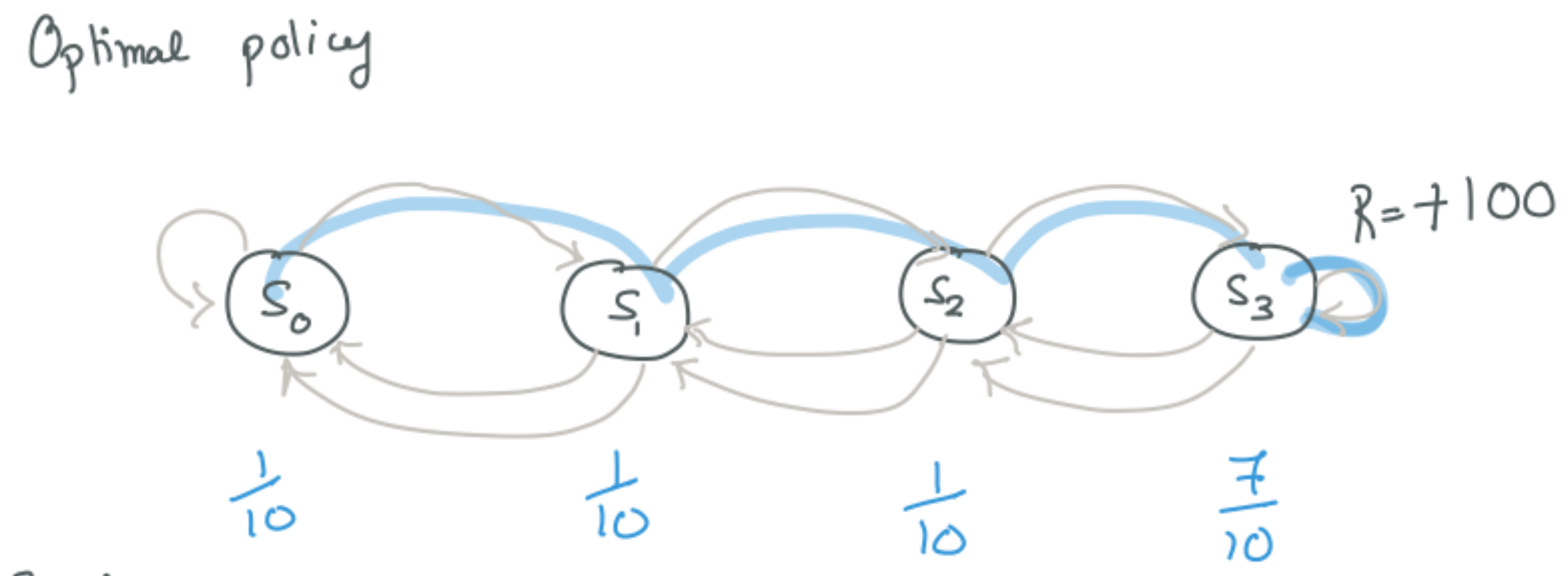
Optimal policy

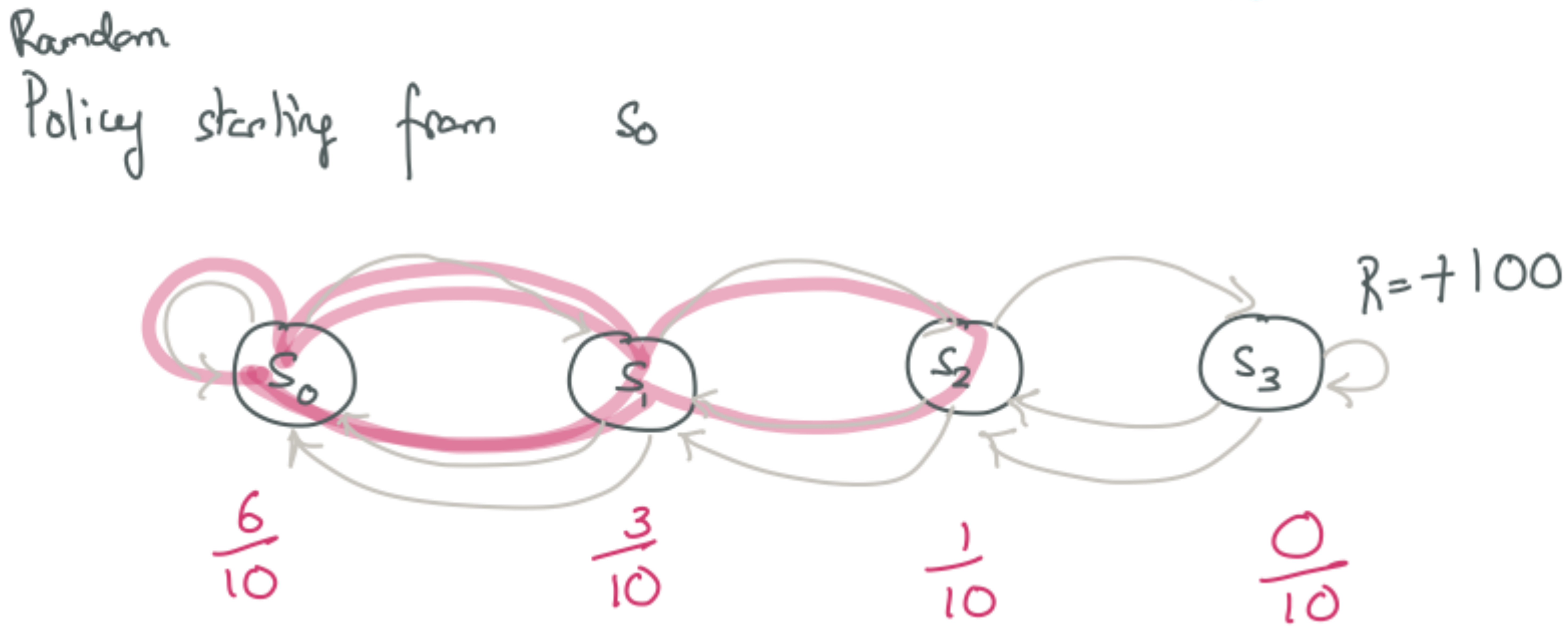
Problem: Lack of exploration

Optimal policy

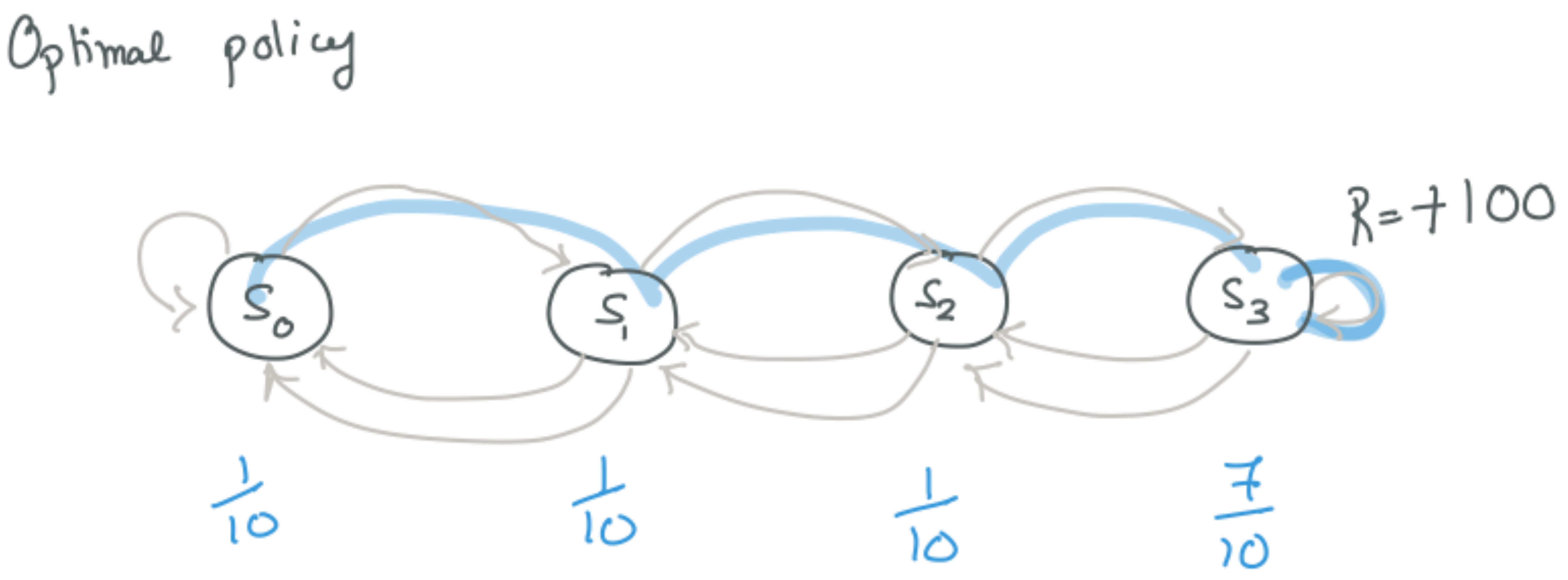


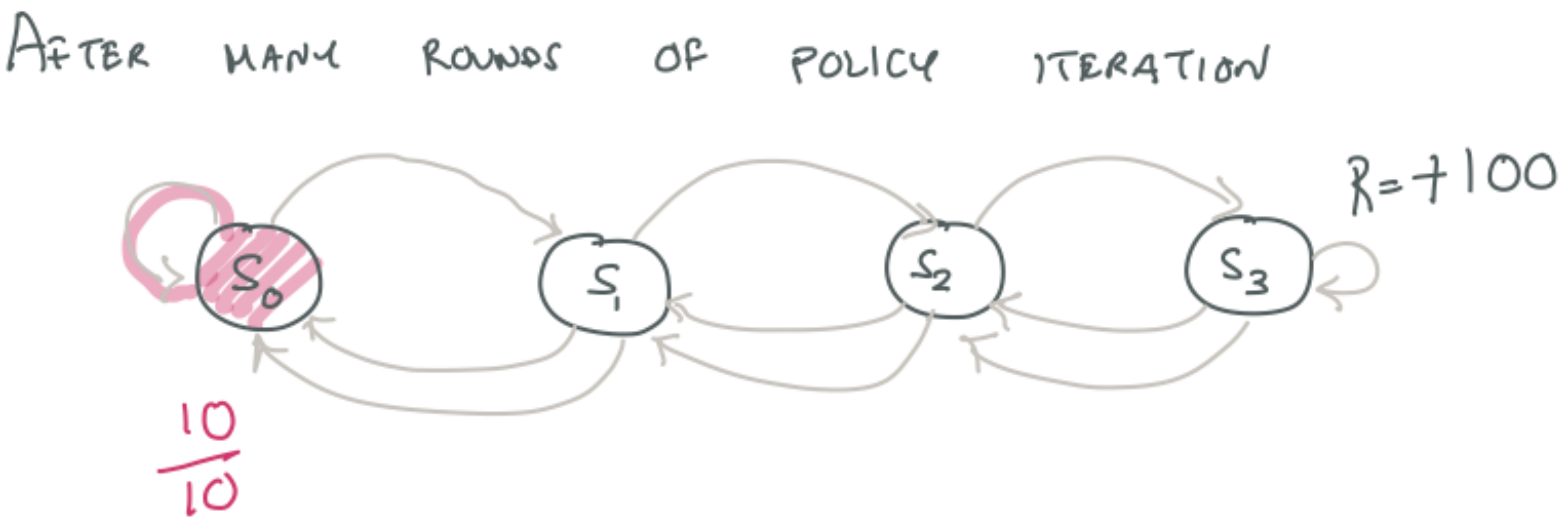
Problem: Lack of exploration





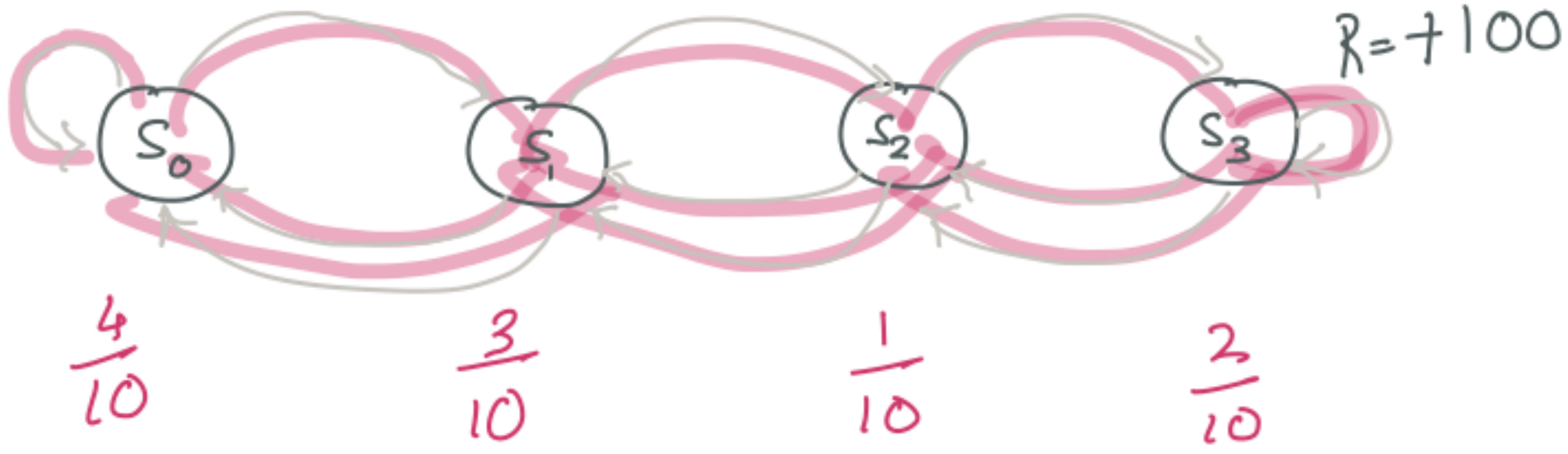
Problem: Lack of exploration



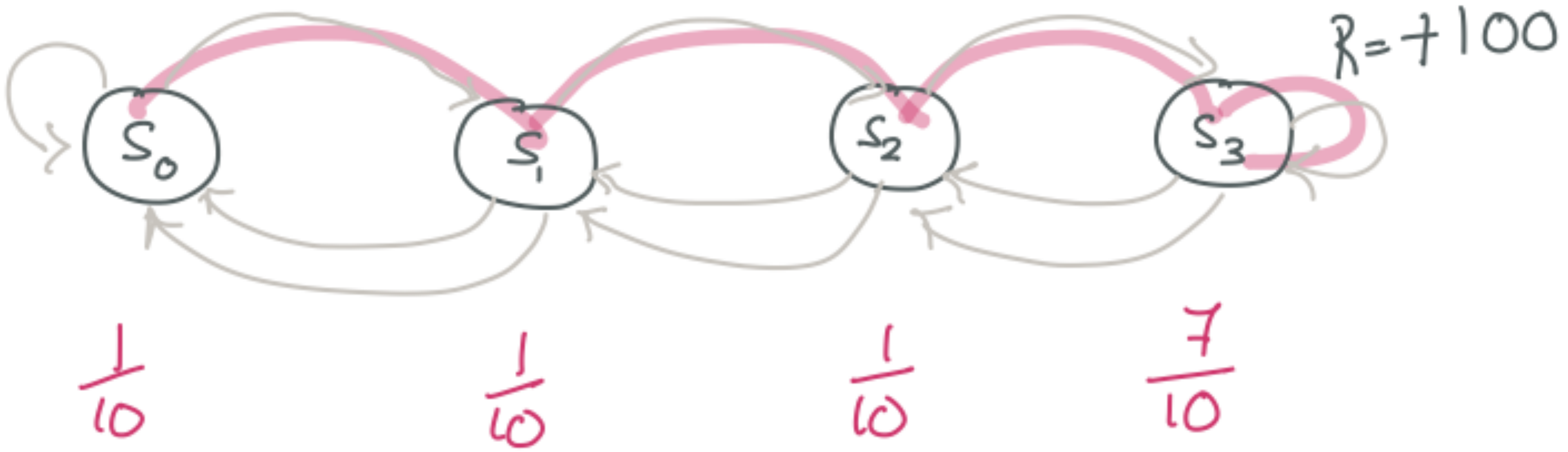


Solution: Demand improvement from *all states*

CHOOSING AN UNIFORM START STATE DISTRIBUTION



AFTER POLICY ITERATION



Key Idea: Use a good "restart" distribution

Choose a restart distribution $\mu(s)$ instead of start state distribution

Try your best to "cover" states the expert will visit

Suffer at most a penalty of $\|\frac{d_{\pi^*}}{\mu}\|_\infty$

Nightmare 2:

Distribution Shift

Approximate Policy Iteration

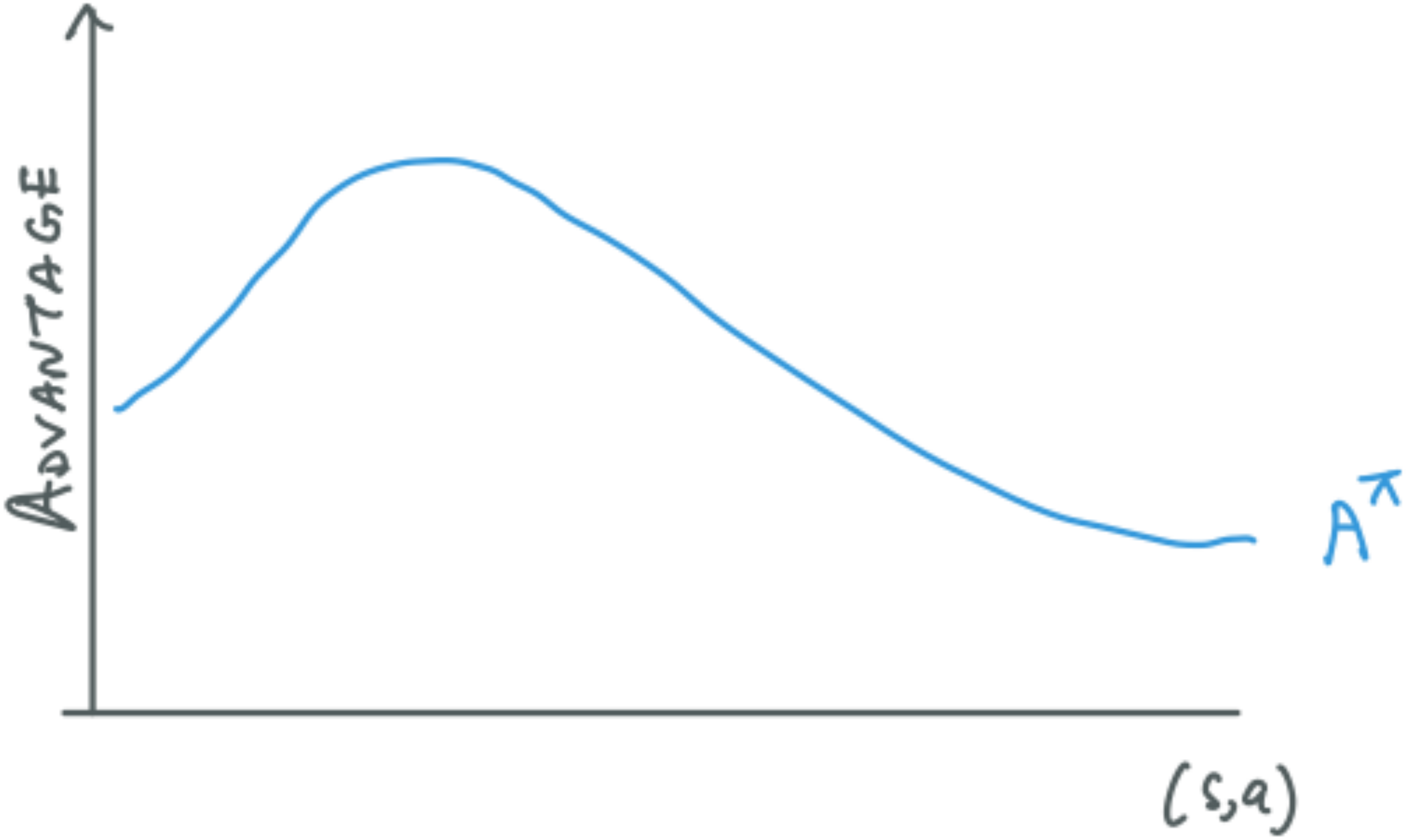
Estimate advantage

$$A^{\pi}(s, a)$$

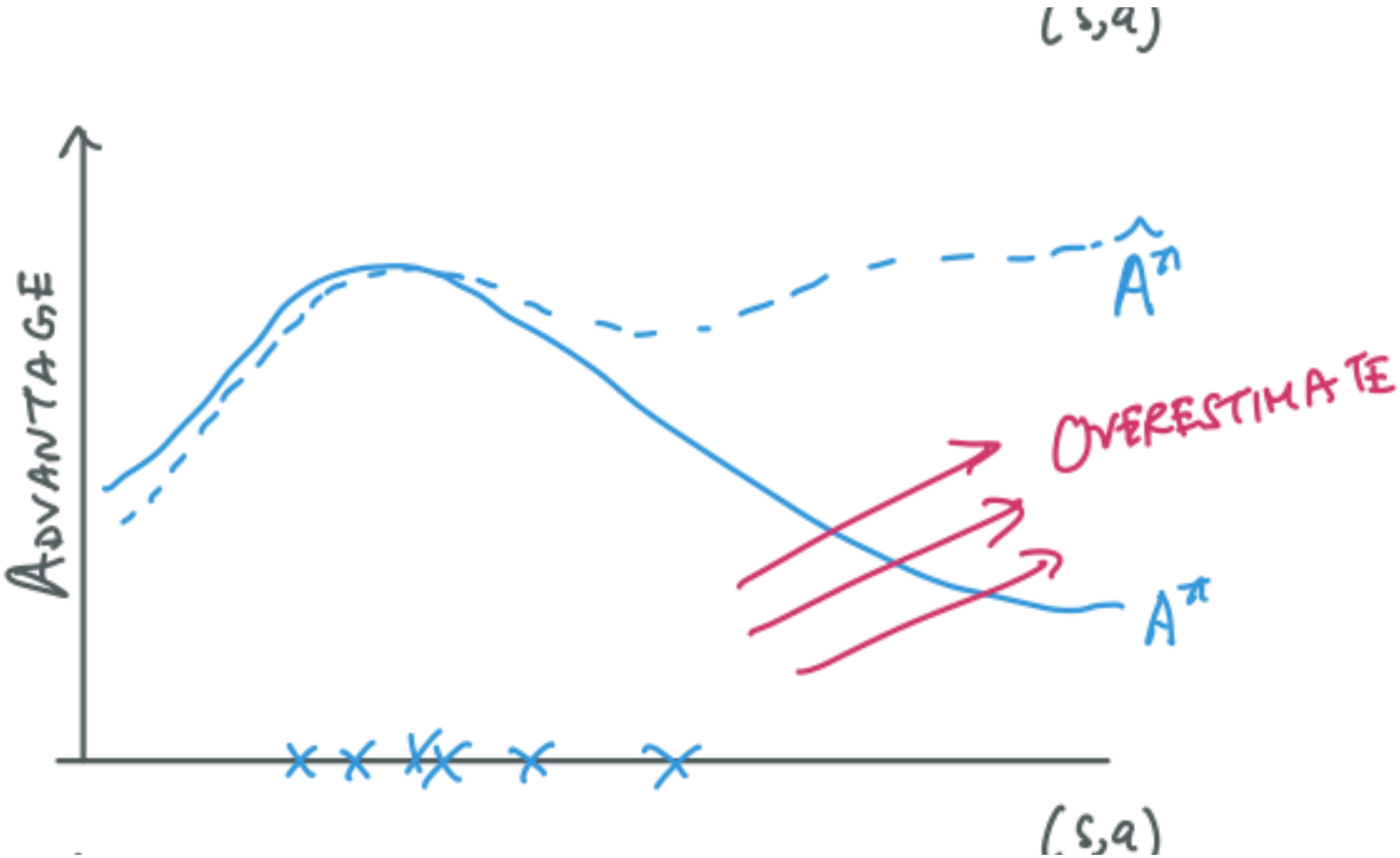
Greedily improve policy

$$\pi' = \arg\min_{\pi'} A^{\pi}(s, \pi'(s))$$

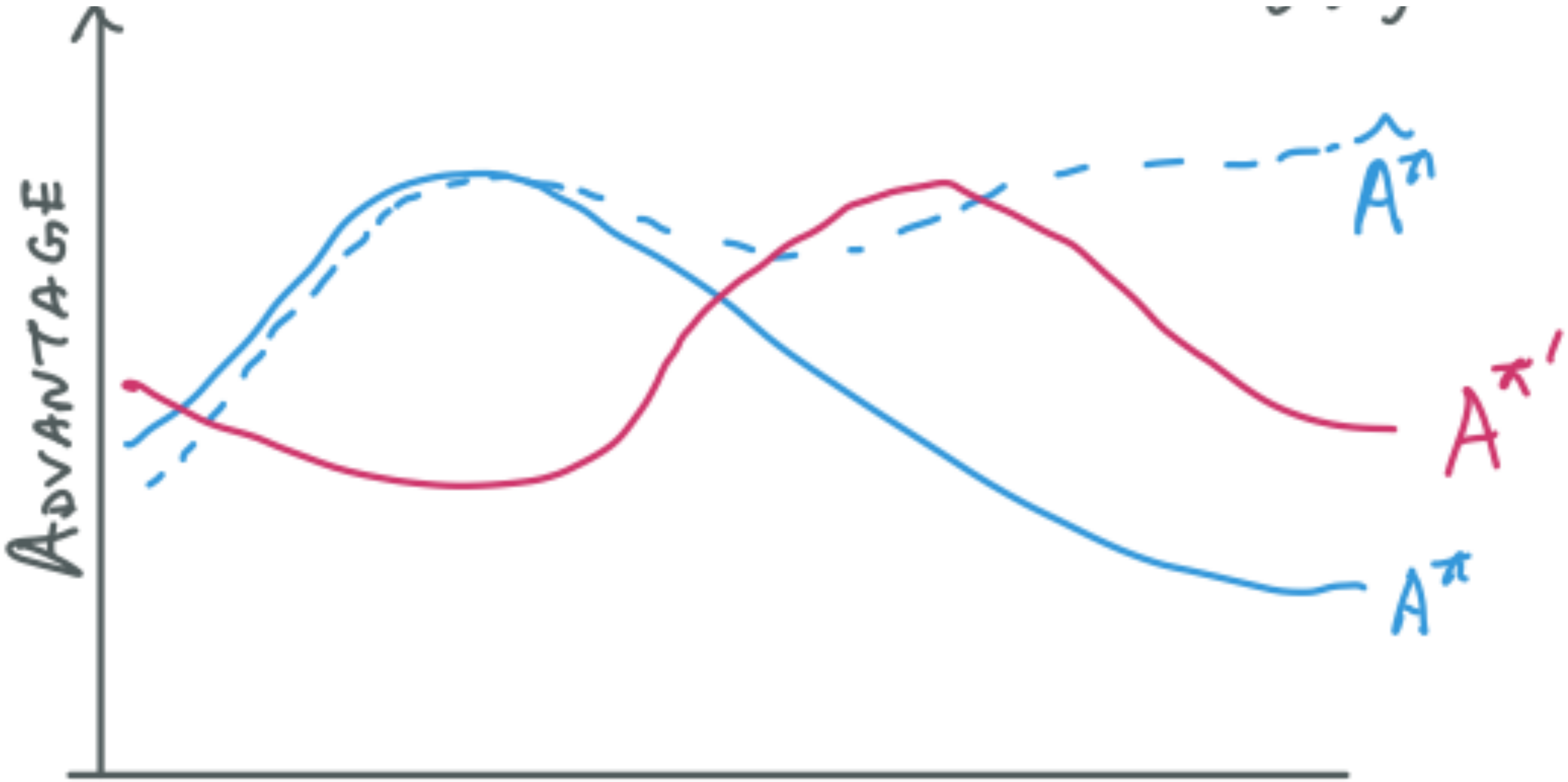
The problem of distribution shift



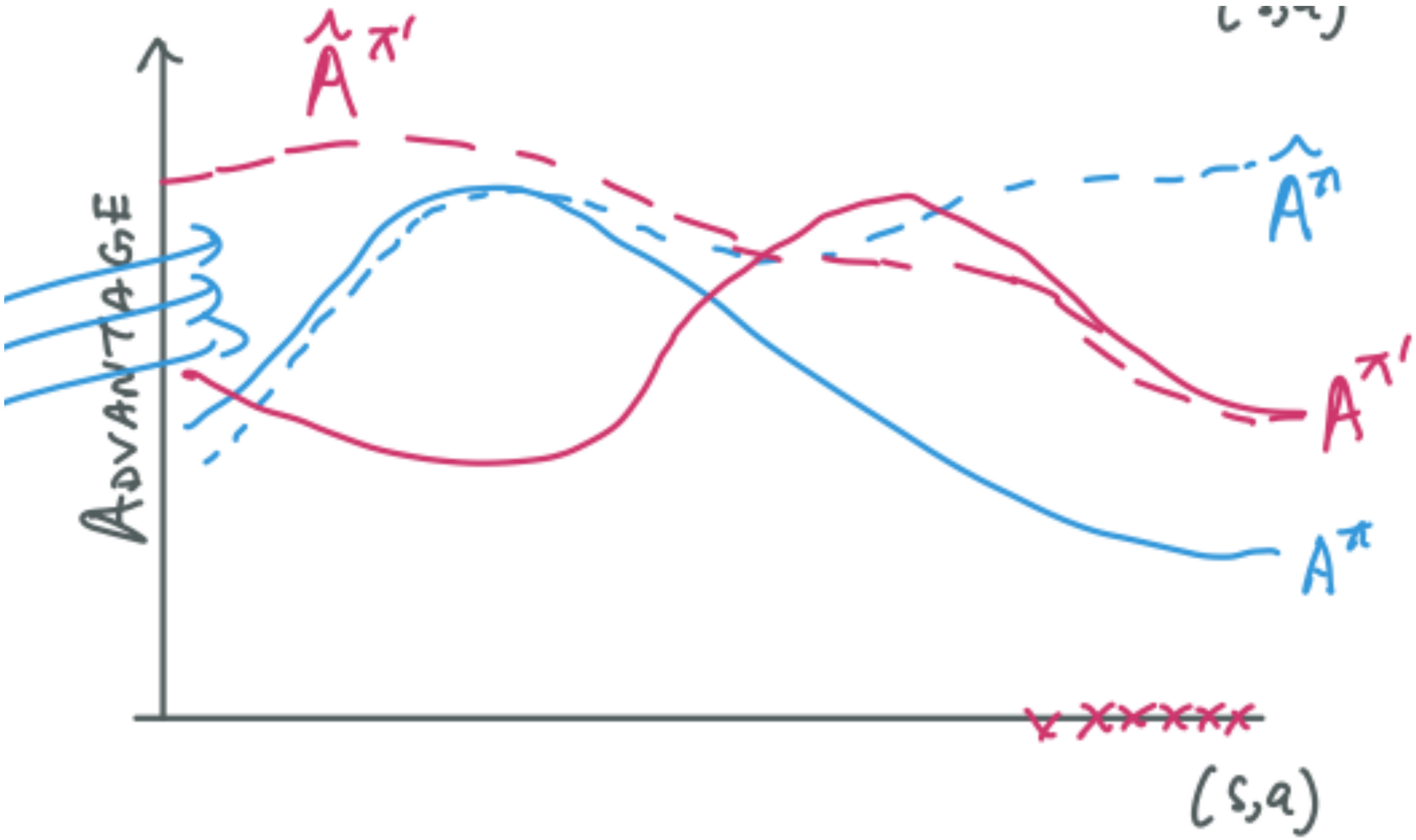
The problem of distribution shift



The problem of distribution shift



The problem of distribution shift



How does distribution shift manifest?

The true performance difference

$$V^{\pi'}(s) - V^{\pi}(s) = \frac{1}{1-\gamma}\mathbb{E}_{\boxed{s \sim d^t_{\pi'}}} A^{\pi}(s, \pi'(s))$$

(New) (Old)

What our estimator currently approximates

$$\frac{1}{1-\gamma}\mathbb{E}_{\boxed{s \sim d^t_{\pi}}} A^{\pi}(s, \pi'(s))$$

Be stable

Slowly change policies

Keep d^t_π close to $d^t_{\pi'}$

Idea 1: Conservative Policy Iteration (CPI)

$$\pi' = (1 - \alpha)\pi + \alpha\pi_{greedy}$$

Approximately Optimal Approximate Reinforcement Learning

Sham Kakade SHAM@GATSBY.UCL.AC.UK
Gatsby Computational Neuroscience Unit, UCL, London WC1N 3AR, UK

John Langford JCL@CS.CMU.EDU
Computer Science Department, Carnegie-Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15217

Mix in old policy and greedy policy

Can prove that performance difference is bounded by

$$V^{\pi'}(s) - V^{\pi}(s) \geq \alpha A_{greedy} - 2\alpha^2 \frac{\gamma}{1-\gamma}$$

How much greedy policy improves based on estimate

How much distribution shift hurts!

Idea 2: Update distributions slowly

CPI requires keeping around *all* the policies you have seen thus far, which is not scalable ...

Instead can we change policies slowly?

Does this simply mean do gradient descent with a small step size?

Nightmare 2:

~~Distribution Shift~~
Correlated Features

Activity!

What happens if we have correlated features?

Parameterization 1: $f_1 =$ # of Holes after the placement, $f_2 =$ Height after the placement. We use θ to denote the parameter for this parameterization.

Parameterization 2: $g_1 = \ldots = g_{100} =$ # of Holes after the placement, $g_{101} =$ Height after the placement. We use ϕ to denote the parameter for this parameterization

Then, for Parameterization 1, we have,

$$\theta^\top f(x, a) = \theta_1 \times \text{\# of Holes}(x, a) + \theta_2 \times \text{Height}(x, a).$$

While for Parameterization 2, we have,

$$\phi^\top g = \left(\sum_{i=1}^{100} \phi_i \right) \times \text{\# of Holes}(x, a) + \phi_{101} \times \text{Height}(x, a).$$

Think-Pair-Share

Think (30 sec): What would happen if we ran policy gradient with Feature Set 1 vs Feature Set 2? How can we fix it?

Pair: Find a partner

Then, for Parameterization 1, we have,

$$\theta^\top f(x,a) = \theta_1 \times \#\text{ of Holes}(x,a) + \theta_2 \times \text{Height}(x,a).$$

While for Parameterization 2, we have,

$$\phi^\top g = \left(\sum_{i=1}^{100} \phi_i\right) \times \#\text{ of Holes}(x,a) + \phi_{101} \times \text{Height}(x,a).$$

Share (45 sec): Partners exchange ideas

Gradient Descent as Steepest Descent

Gradient Descent is simply Steepest Descent with L2 norm

$$\max_{\Delta\theta} J(\theta + \Delta\theta) \qquad s.t. \qquad \|\Delta\theta\| \leq \epsilon$$

An alternative norm: KL Divergence! Gives rise to Fisher Information Matrix

$$G(\theta) = E_{p_\theta}\left[\nabla_\theta \log(p_\theta)\, \nabla_\theta \log(p_\theta)^\top\right] \qquad\qquad \Delta\theta = \frac{1}{2\lambda}\,\tilde{G}^{-1}(\theta)\,\tilde{\nabla}_\theta J.$$

Natural Gradient Descent

Estimate Fisher Information Matrix

$$\tilde{G}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \left[\nabla_\theta \log \pi_\theta(a_i|s_i) \nabla_\theta \log \pi_\theta(a_i|s_i)^\top \right]$$

Parameter update:

$$\Delta\theta = \frac{1}{2\lambda} \tilde{G}^{-1}(\theta)\, \widetilde{\nabla}_\theta J.$$

Modern variants known as TRPO, PPO

Nightmare 3:

Variance

What happens when Q values for all rollouts are similar?

$$\nabla_\theta J = E_{p(\xi|\theta)} \left[\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t|s_t) \, Q^{\pi_\theta}(s_t, a_t) \right]$$

Recall that one of the reasons for the high variance is that the algorithm does not know how well the trajectories perform compared to other trajectories. Therefore, by introducing a baseline for the total reward (or reward to go), we can update the policy based on how well the policy performs compared to a baseline

Solution: Subtract a baseline!

$$\nabla_\theta J = E_{d^{\pi_\theta}(s)} E_{\pi_\theta(a|s)} \left[\nabla_\theta \log(\pi_\theta(a|s)) \left(Q^{\pi_\theta}(s,a) - V^{\pi_\theta}(s) \right) \right].$$

We can prove that this does not change the gradient

But turns Q values into advantage (which is lower magnitude)

tl;dr

The Policy Gradient Theorem

$$\begin{aligned}\nabla_\theta J &= E_{p(\xi|\theta)}\left[\sum_{t=0}^{T-1}\left(\nabla_\theta \log \pi_\theta(a_t|s_t)\left(\sum_{t'=0}^{t-1} r(s_{t'},a_{t'}) + \sum_{t'=t}^{T-1} r(s_{t'},a_{t'})\right)\right)\right] \\ &= E_{p(\xi|\theta)}\left[\sum_{t=0}^{T-1}\left(\nabla_\theta \log \pi_\theta(a_t|s_t) \sum_{t'=t}^{T-1} r(s_{t'},a_{t'})\right)\right],\end{aligned}$$

$$\nabla_\theta J = E_{p(\xi|\theta)}\left[\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t|s_t)\, Q^{\pi_\theta}(s_t,a_t)\right]$$

15

1. Local Optima: Use Exploration Distribution
2. Distribution Shift: *Natural* Gradient Descent
3. High Variance: Subtract baseline