Policy Gradients

Sanjiban Choudhury

Switch from costs to rewards

All optimal control / planning literature written as costs

All RL literature written as rewards

Cost = -Reward

All min() become max()

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 $\{\xi^{(i)}\}_{i=1}^{N}$

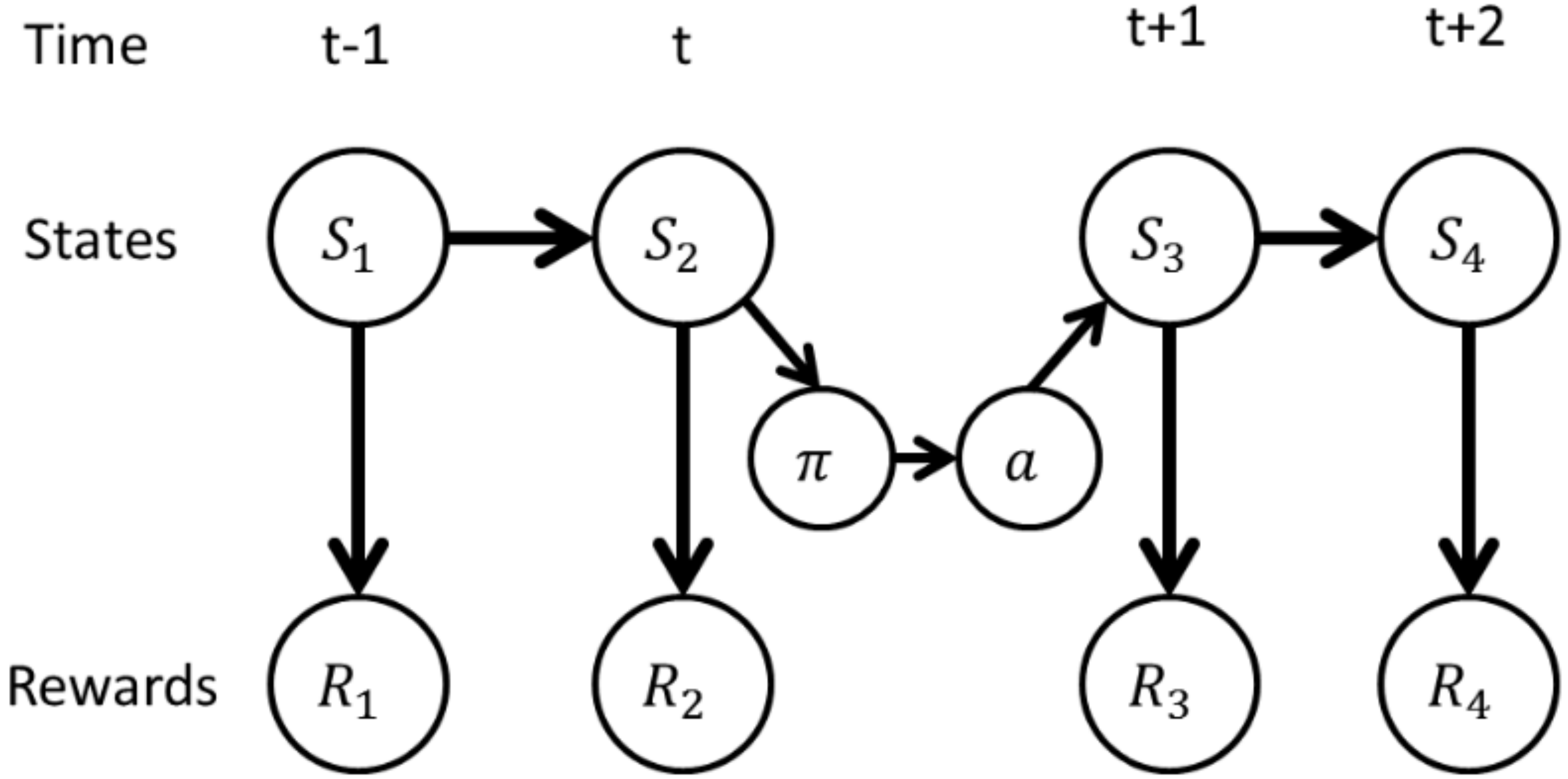
Compute estimated gradient

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

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

return π_θ

Causality: Can actions affect the past?



How can we

$$\nabla_\theta J = \left[\left(\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta \left(a_t^{(i)} \mid s_t^{(i)} \right) \right) \sum_{t=0}^{T-1} r(s_t, a_t)) \right] .$$

The Policy Gradient Theorem

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

The Policy Gradient Theorem

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

$$Q^{\pi_\theta}(s_t,a_t)$$

(The reward to go)

The Policy Gradient Theorem

(Finite Horizon Version)

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

The Policy Gradient Theorem

(Finite Horizon Version)

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

(Infinite Horizon Version)

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

Hardware

D'Claw (9 dof)

Allegro (16 dof)

c) Door opening: This task involves both the arm and the hand working in tandem to open a door. The robot must learn to approach the door, grip the handle, and then pull backwards. This task has more degrees of freedom given the additional arm, and involves the sequence of actions: going to the door, gripping the door, and then pulling away.

Fig. 5: Opening door with flexible handle

The state space is all the joint angles of the hand, the Cartesian position of the arm, the current angle of the door, and last action taken. The action space is the position space of the hand and horizontal position of the wrist of the arm. The reward function is provided as

$$r = -(d\theta)^2 - (x_{\text{arm}} - x_{\text{door}})$$
$$d\theta := \theta_{\text{door}} - \theta_{\text{closed}}$$

We define a trajectory as a success if at any point $d\theta > 30°$.

a) Valve Rotation: This task involves turning a valve or faucet to a target position. The fingers must cooperatively push and move out of the way, posing an exploration challenge. Furthermore the contact forces with the valve complicate the dynamics. For our task, the valve must be rotated from $0°$ to $180°$.

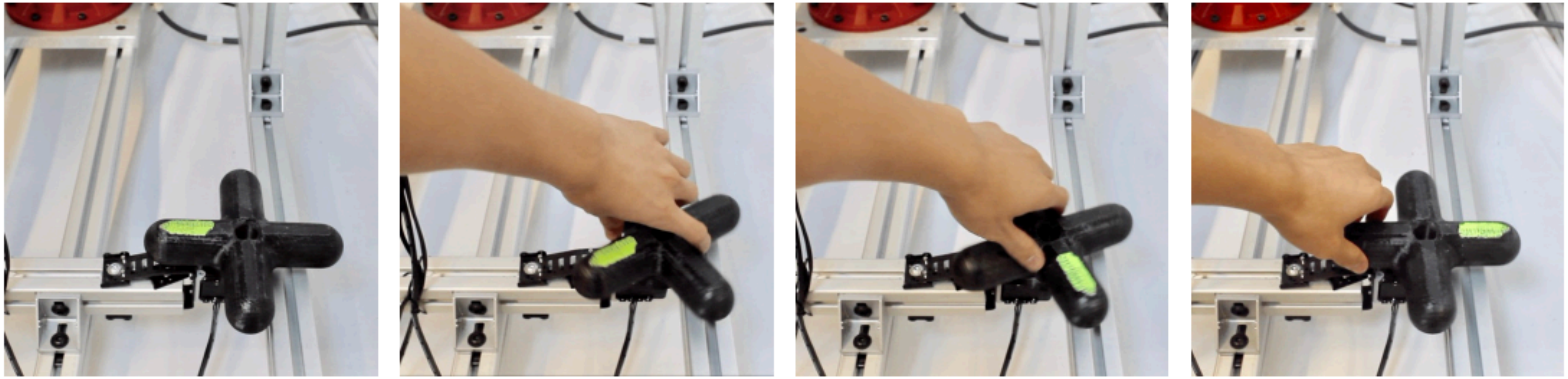


Fig. 3: Illustration of valve rotation

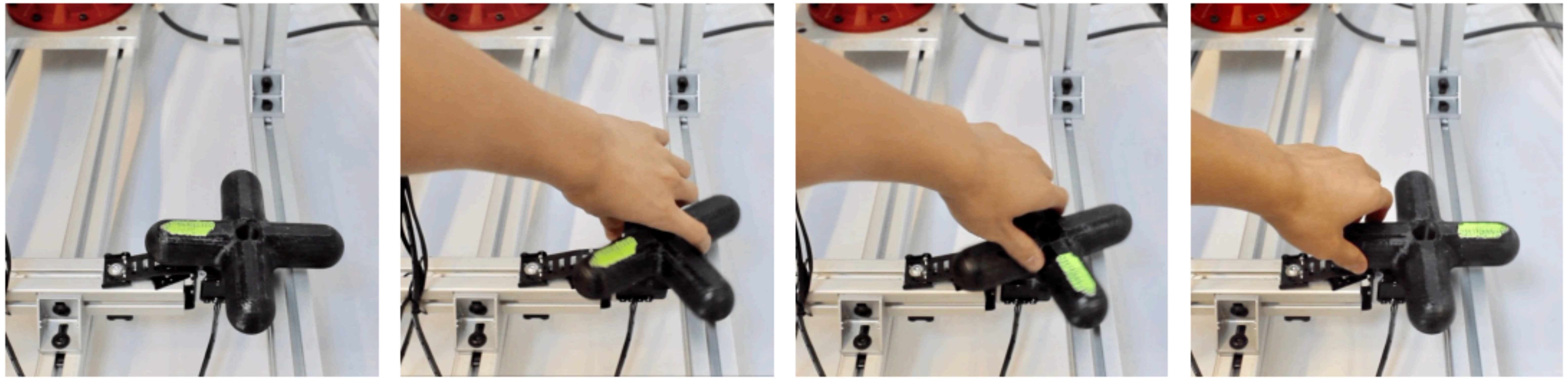


Fig. 3: Illustration of valve rotation

The state space consists of all the joint angles of the hand, the current angle of rotation of the valve [θ_{valve}], the distance to the goal angle [$d\theta$], and the last action taken. The action space is joint angles of the hand and the reward function is

$$r = -|d\theta| + 10 * \mathbb{1}_{\{|d\theta|<0.1\}} + 50 * \mathbb{1}_{\{|d\theta|<0.05\}}$$
$$d\theta := \theta_{\text{valve}} - \theta_{\text{goal}}$$

We define a trajectory as a success if $|d\theta| < 20^\circ$ for at least 20% of the trajectory.

Activity

On-policy vs Off-policy

On-policy RL algorithms:

You must collect data according to your current policy to update learner parameters

Off-policy RL algorithms:

Your learner can learn from data from *any policy*

On-policy vs Off-policy

On-policy RL algorithms:

You must collect data according to your current policy to update learner parameters

Off-policy RL algorithms:

Your learner can learn from data from *any policy*

When poll is active respond at **PollEv.com/sc2582**

Are we done?

No!

Three major nightmares with policy gradients

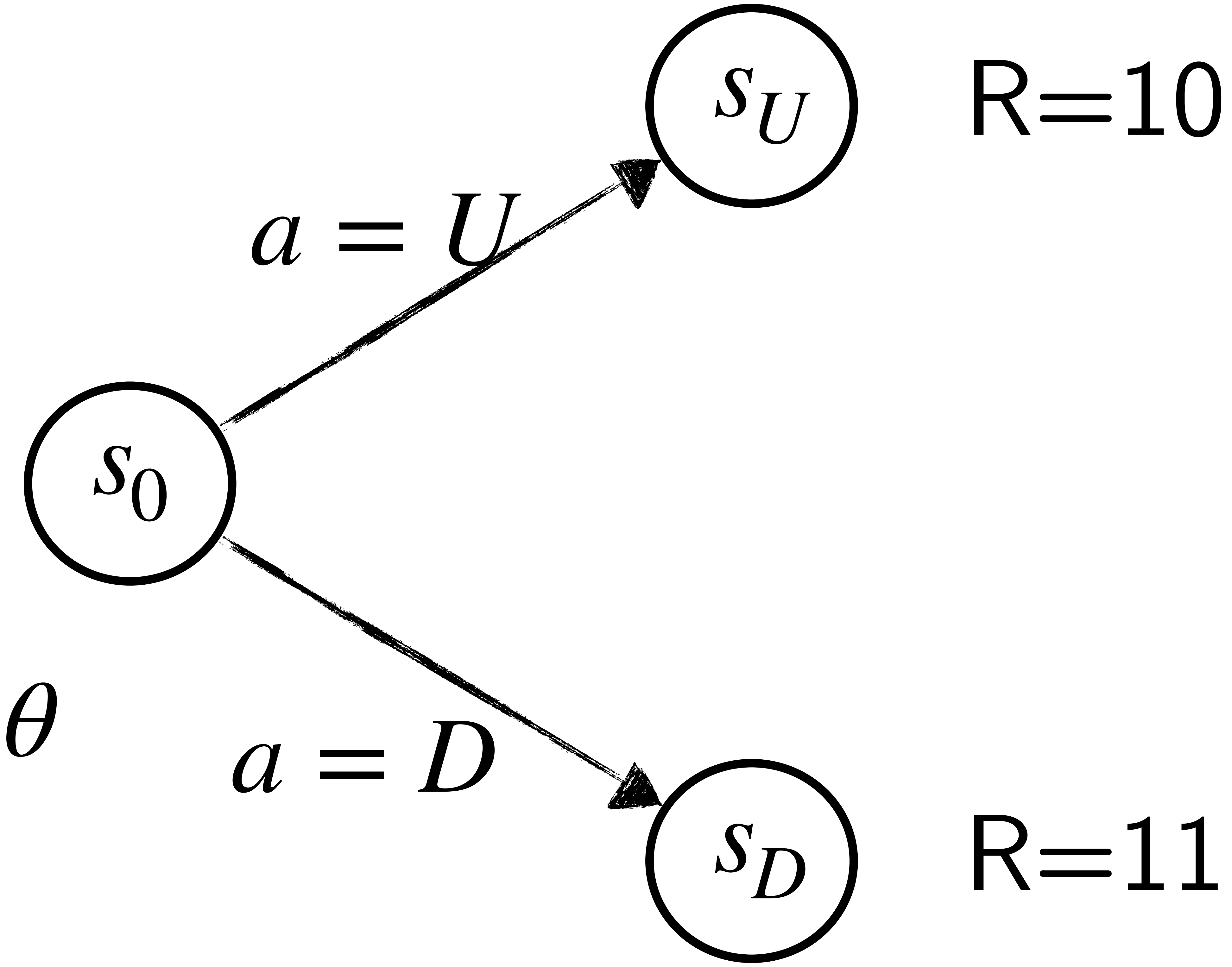
Nightmare 1:

High Variance

Consider the following MDP

$\pi_\theta(a = U \mid s_0) = \theta$

$\pi_\theta(a = D \mid s_0) = 1 - \theta$



Suppose we init $\theta = 0.5$, and draw 4 samples with our policy

And then apply PG

When Q values for all rollouts in a batch are high?

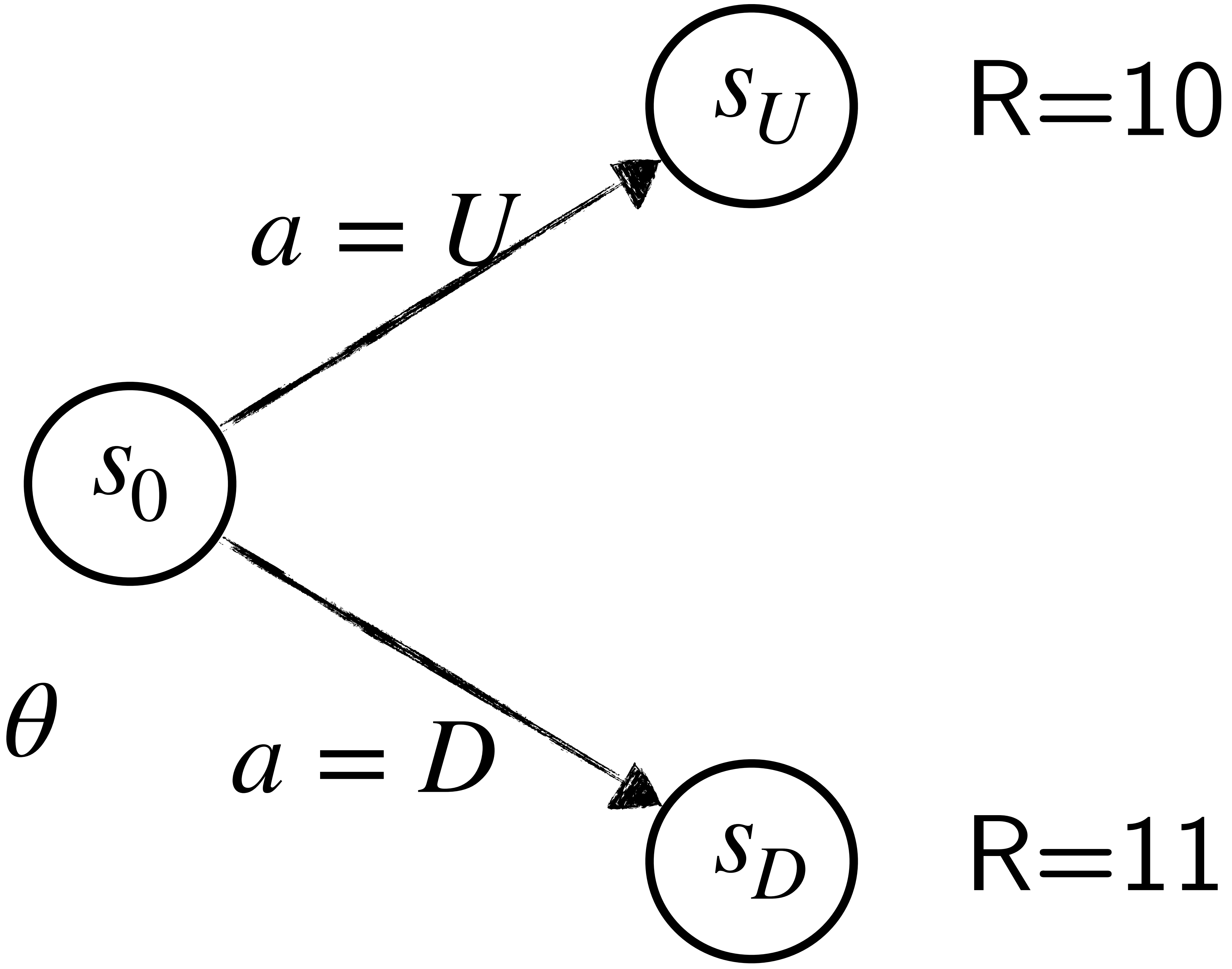
$$\boxed{\nabla_\theta J = E_{s\sim d^{\pi_\theta}(s), a\sim\pi_\theta(a|s)}\left[\nabla_\theta \log \pi_\theta(a|s) Q^{\pi_\theta}(s,a)\right]}$$

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!

$\pi_\theta(a = U \mid s_0) = \theta$

$\pi_\theta(a = D \mid s_0) = 1 - \theta$



Suppose we subtracted of $V^\pi(s_0) = 10.5$ from the reward to go

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

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

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

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

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

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

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

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

Can we justify this move using the PDL?

Nightmare 2:

Distribution Shift

What happens if your step-size is large?

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

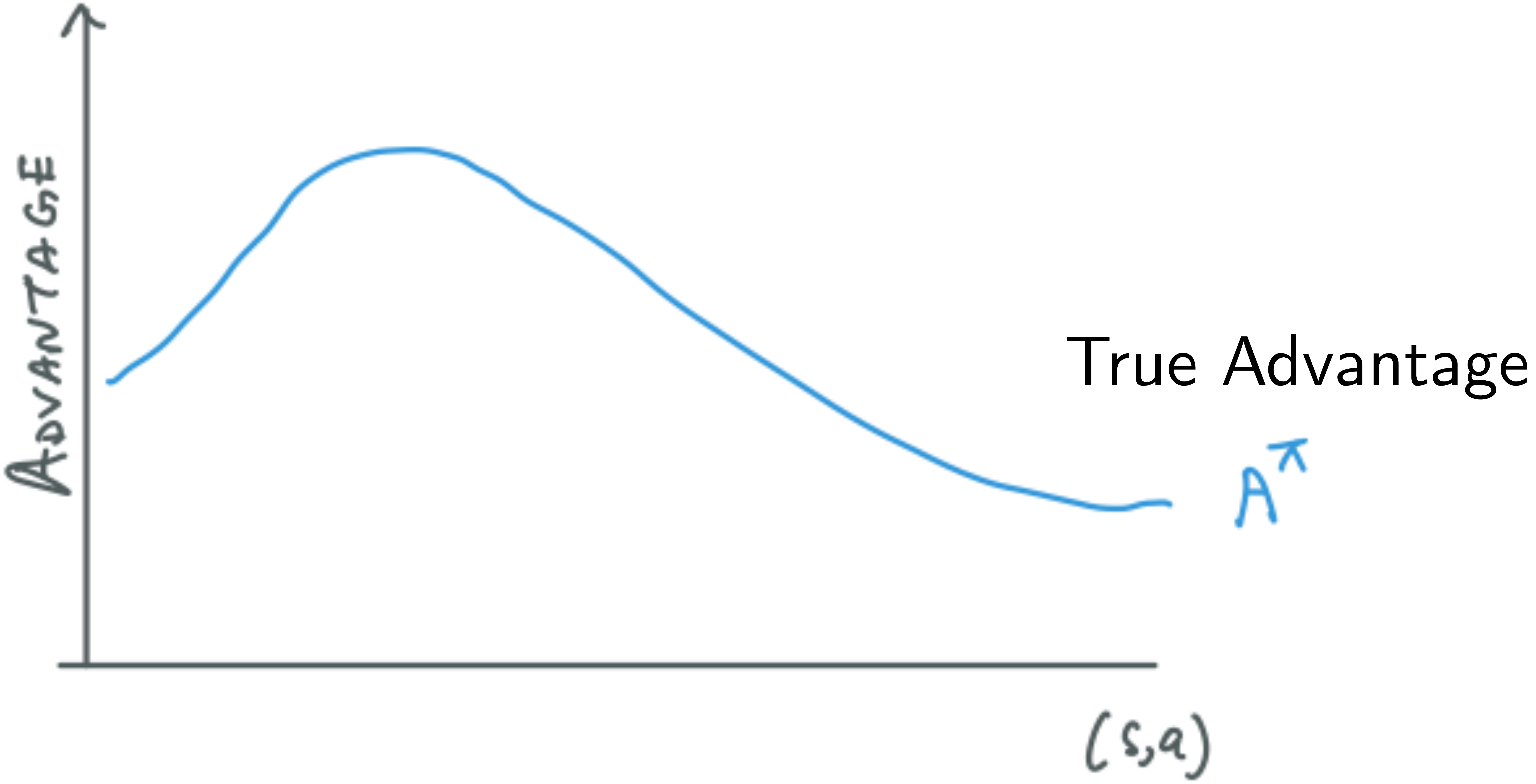
What happens if your step-size is large?

$$\nabla_\theta J = E_{d^{\pi_\theta}(s)} E_{\pi_\theta(a|s)} \left[\nabla_\theta \log(\pi_\theta(a|s)\cancel{A^{\pi_\theta}(s,a)})\right]$$

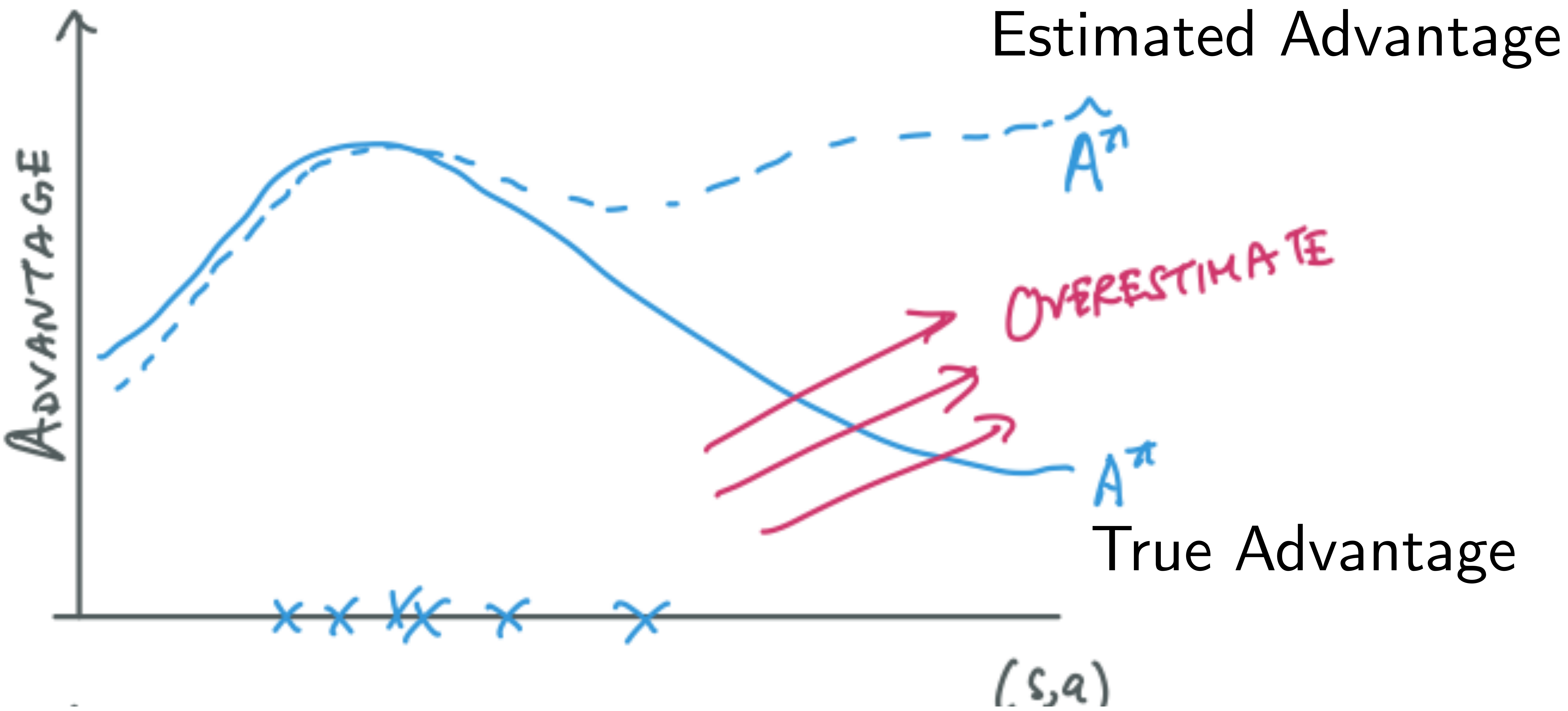
$$\hat{A}^{\pi_\theta}(s,a)$$

We are *estimating* the advantage from roll-outs

The problem of distribution shift

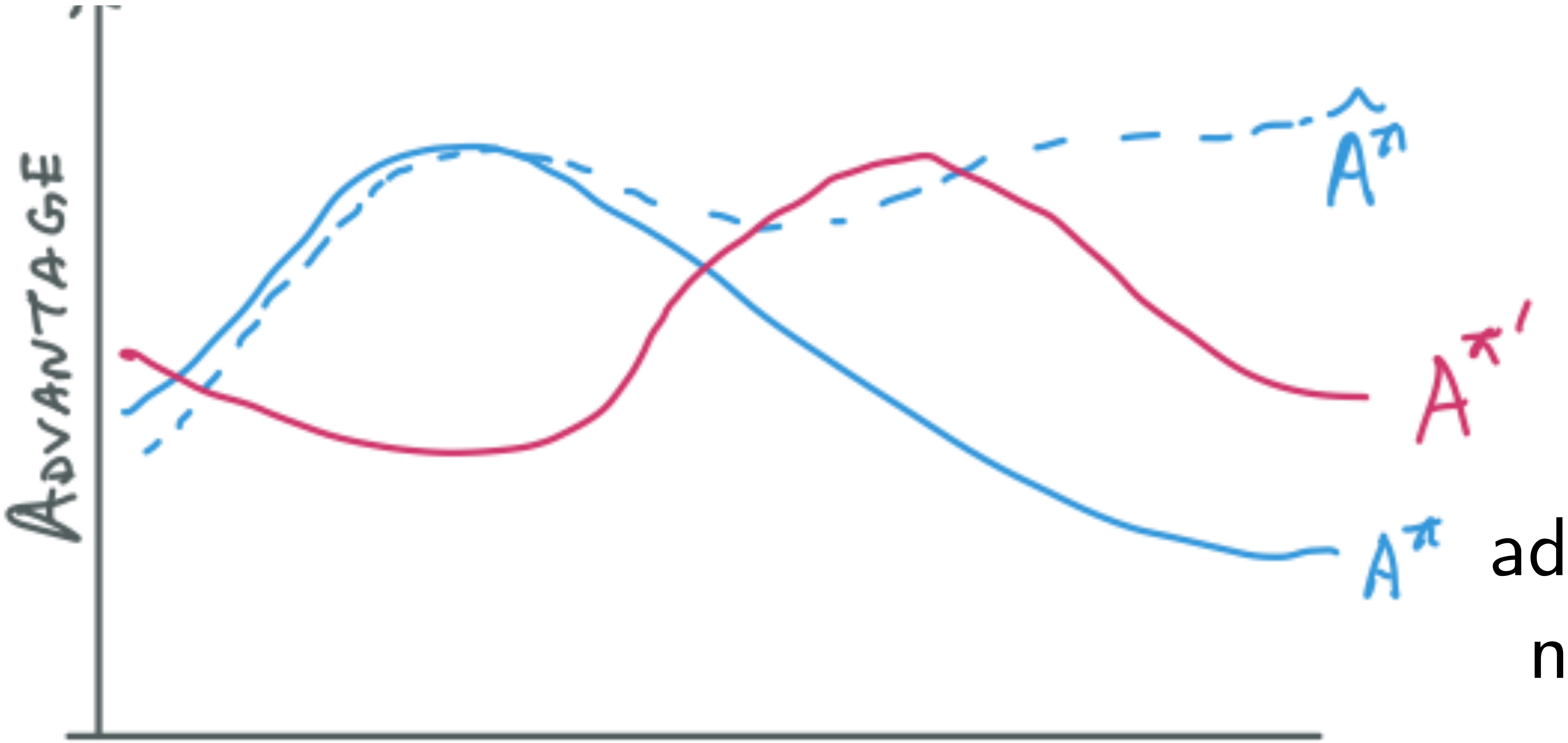


The problem of distribution shift



Our new policy wants to go all the way to the RIGHT

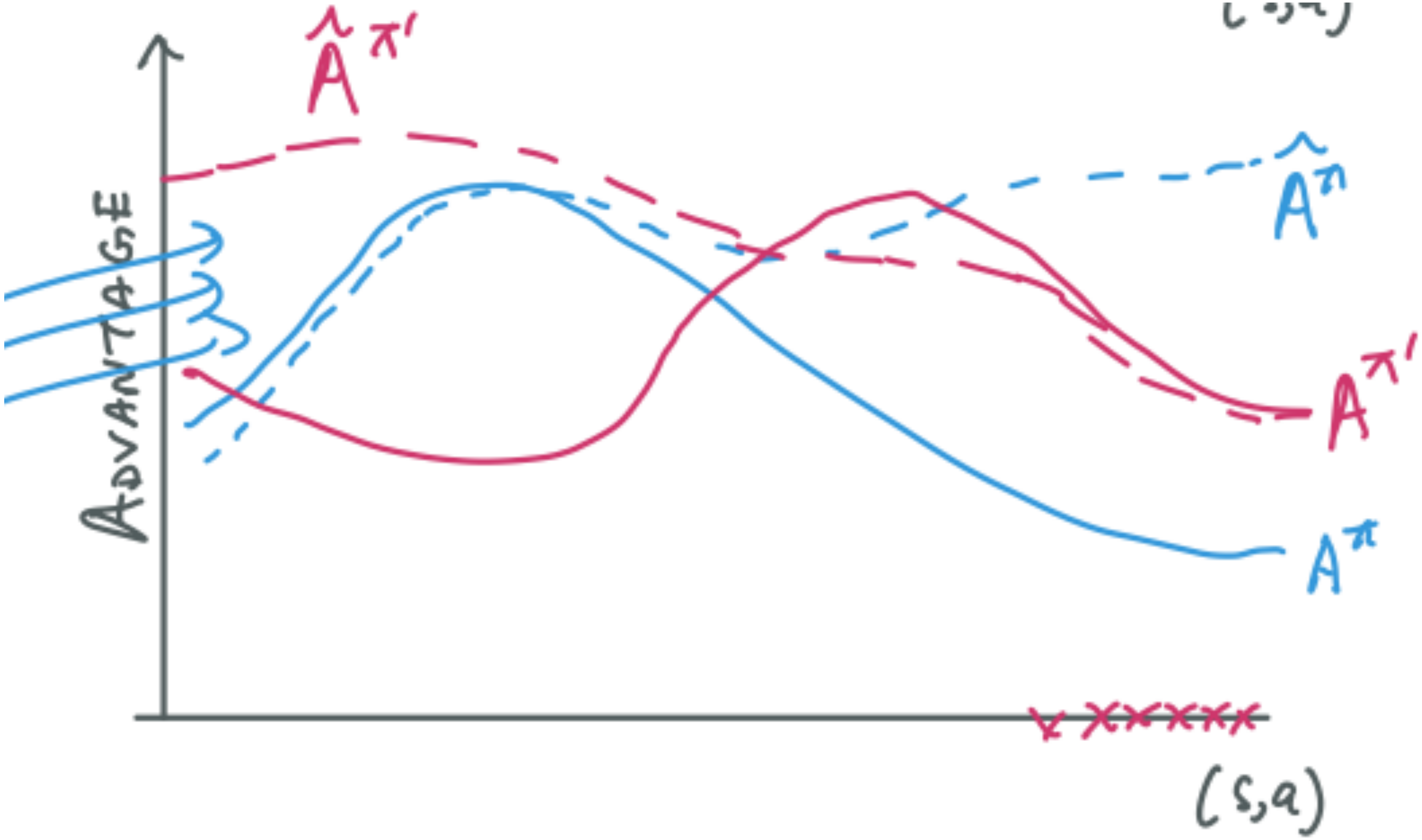
The problem of distribution shift



True advantage of new policy

The problem of distribution shift

Estimated Advantage



True advantage of new policy

Our new policy wants to go all the way to the LEFT

Recap: Problem with Approximate Policy Iteration

$$V^{\pi^+}(s_0) - V^{\pi}(s_0) = \sum_{t=0}^{T-1} \mathbb{E}_{s_t \sim d_t^{\pi^+}} A^{\pi}(s_t, \pi^+)$$

PDL requires accurate Q_θ^π on states that π^+ will visit! ($d_t^{\pi^+}$)

But we only have states that π visits (d_t^π)

If π^+ changes drastically from π, then $|d_t^{\pi^+} - d_t^\pi|$ is big!

Be stable

Slowly change policies

Keep $d_t^{\pi^+}$ close to d_t^{π}

Goal: Change distributions slowly

$$\max_{\Delta\theta} J(\theta + \Delta\theta)$$

s.t. $d^{\pi_{\theta+\Delta\theta}}$ is close to d^{π_θ}

How do we measure distance between distributions?

Goal: Change distributions slowly

$$\max_{\Delta\theta} J(\theta + \Delta\theta)$$

s.t. $KL(d^{\pi_{\theta+\Delta\theta}} || d^{\pi_\theta}) \leq \epsilon$

This gives us a new type of gradient descent

$$\max_{\Delta\theta} J(\theta + \Delta\theta)$$

s.t. $KL(d^{\pi_{\theta+\Delta\theta}} || d^{\pi_\theta}) \leq \epsilon$

$$\theta \leftarrow \theta + \eta \boxed{G^{-1}(\theta)} \nabla_\theta J(\theta)$$

Where $G(\theta)$ is the Fischer Information Matrix

$$G(\theta) = \mathbb{E}_{s,a \sim d^\pi_\theta}[\nabla_\theta \log \pi_\theta(a \,|\, s) \nabla_\theta \log \pi_\theta(a \,|\, s)^T]$$

This is also called a "natural" gradient

$$\theta \leftarrow \theta + \eta \boxed{G^{-1}(\theta)} \nabla_\theta J(\theta)$$

Where $G(\theta)$ is the Fischer Information Matrix

$$G(\theta) = \mathbb{E}_{s,a \sim d_\theta^\pi}[\nabla_\theta \log \pi_\theta(a \mid s) \nabla_\theta \log \pi_\theta(a \mid s)^T]$$

"Natural" Gradient Descent

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

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

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

Update parameters $\theta \leftarrow \theta + \alpha \boxed{\tilde{G}^{-1}(\theta)}\widetilde{\nabla}_\theta J$.

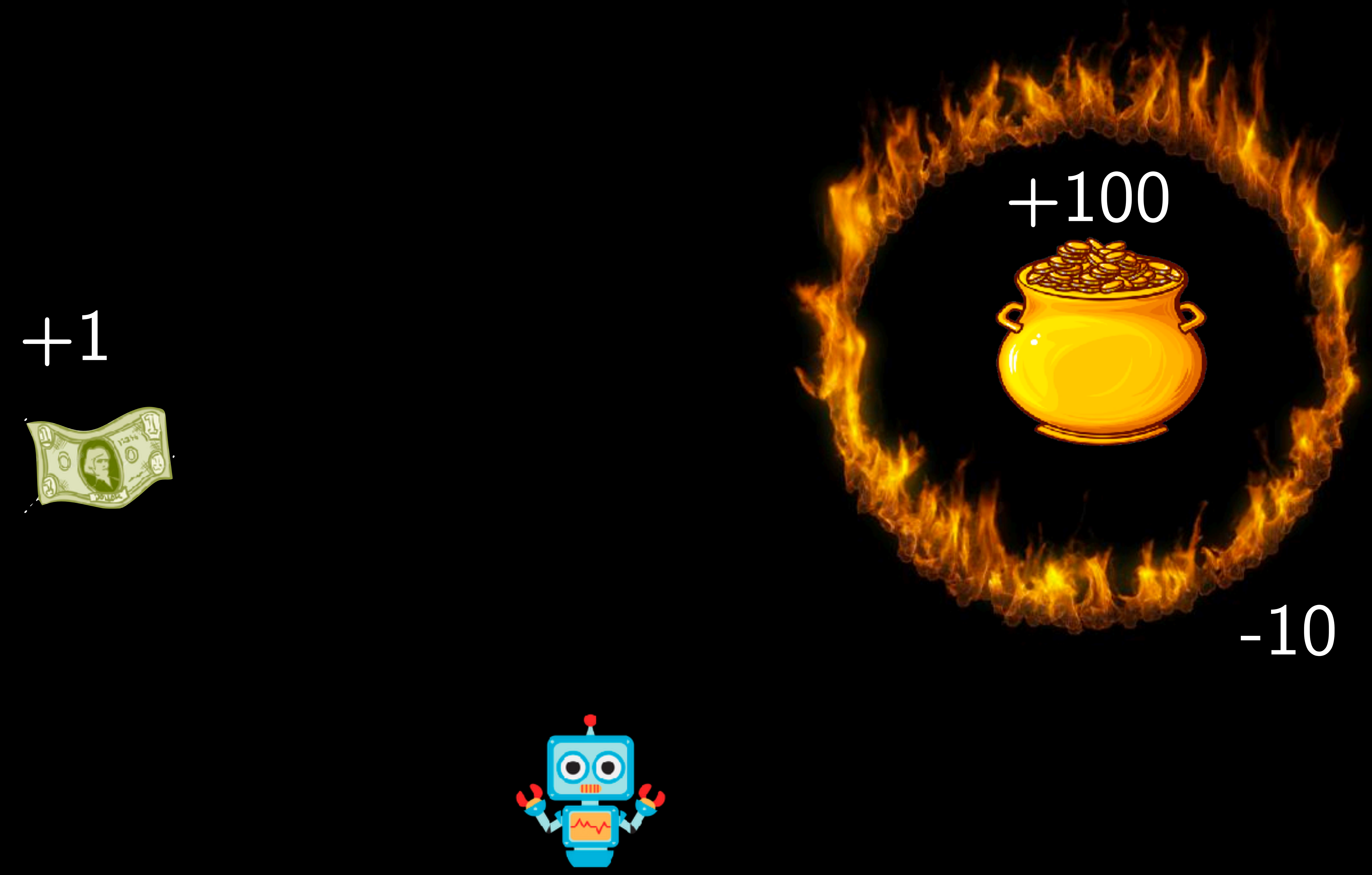
return π_θ

Modern variants are TRPO, PPO, etc

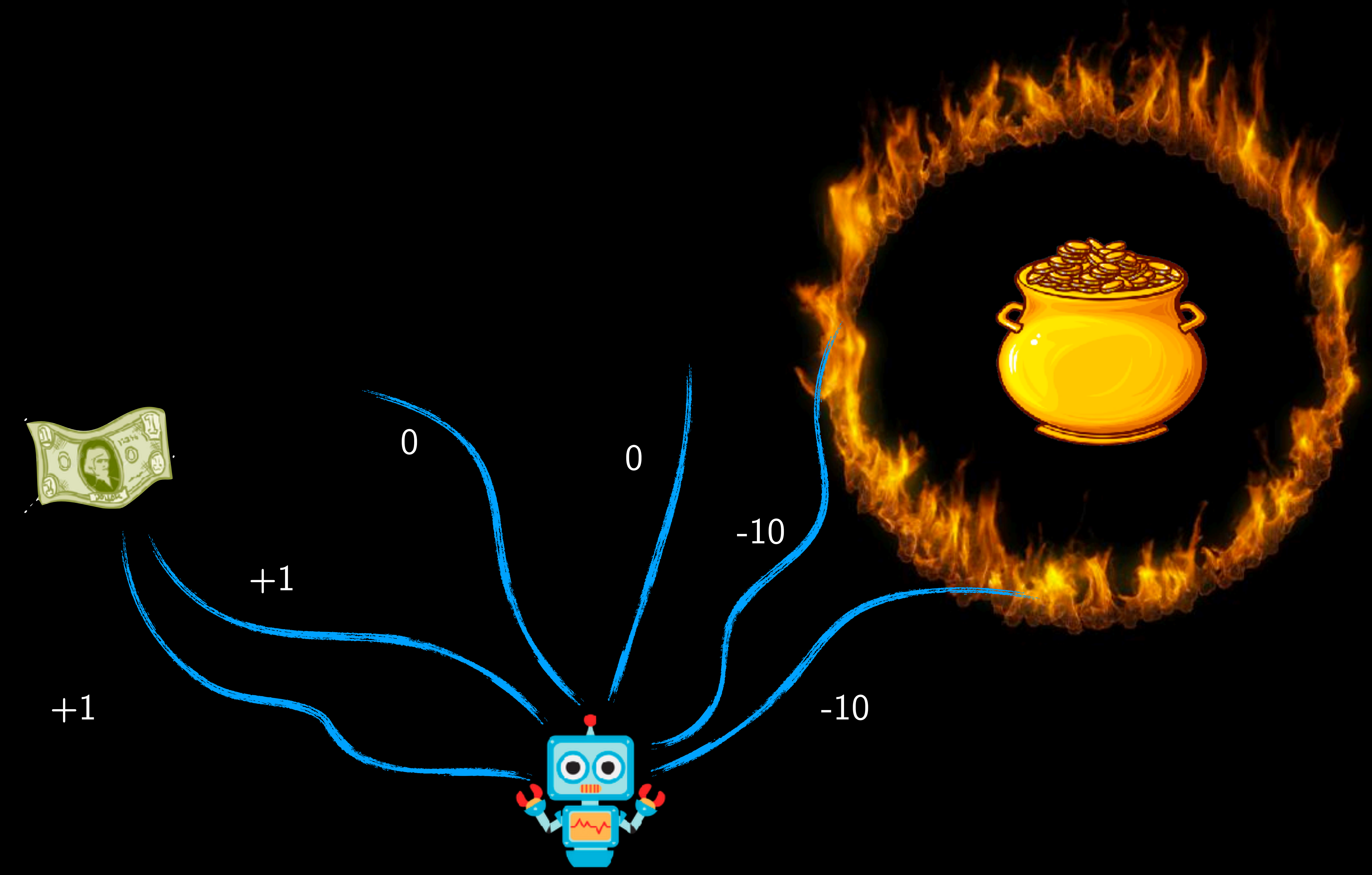
Nightmare 3:

Local Optima

The Ring of Fire

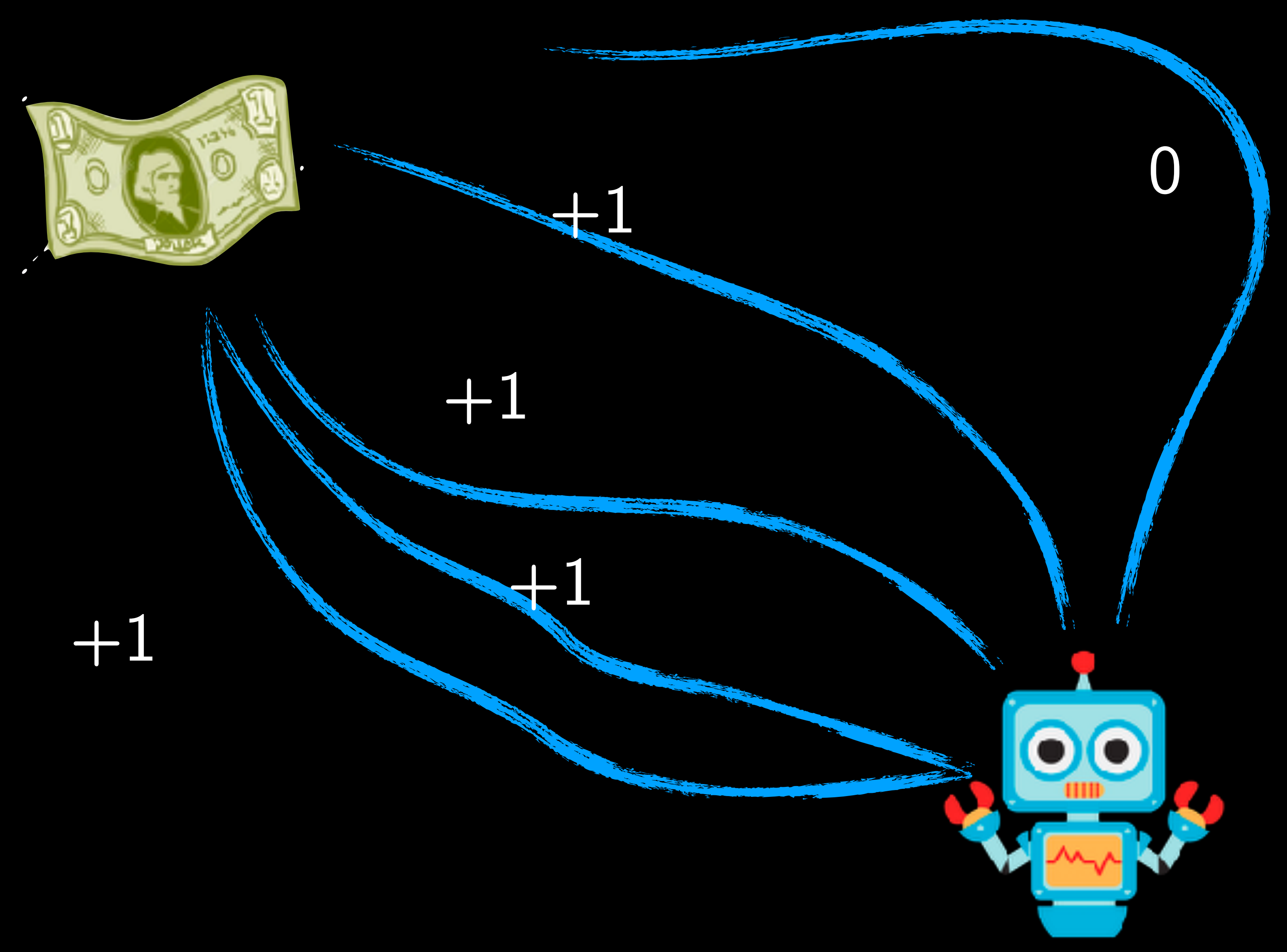


The Ring of Fire



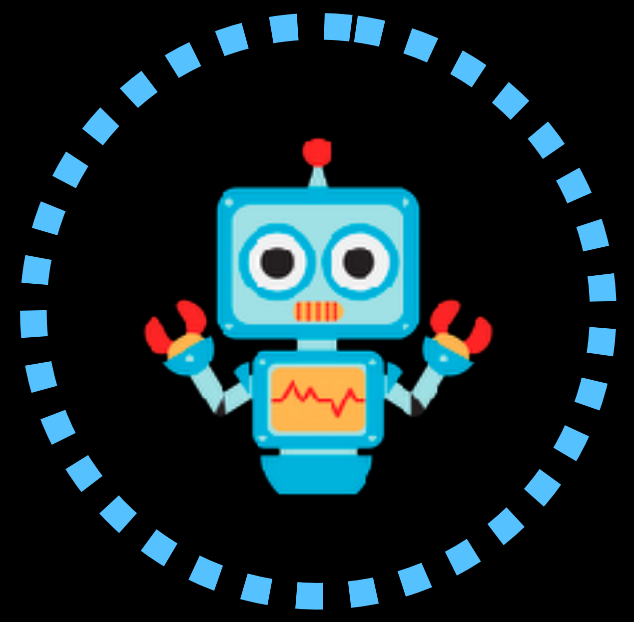
The Ring of Fire

Get's sucked into a local optima!!

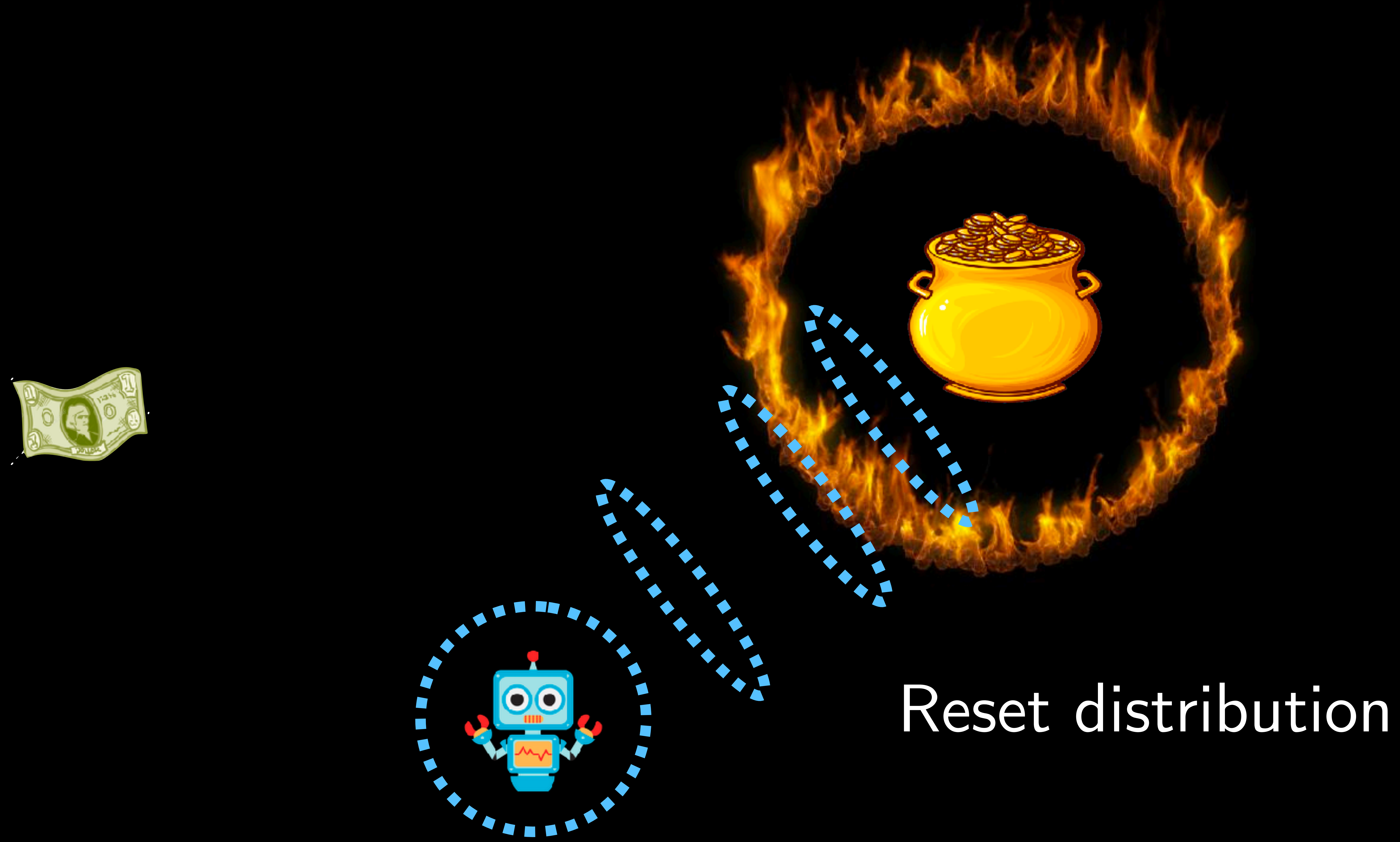


Idea: What if we had a "good reset distribution?"

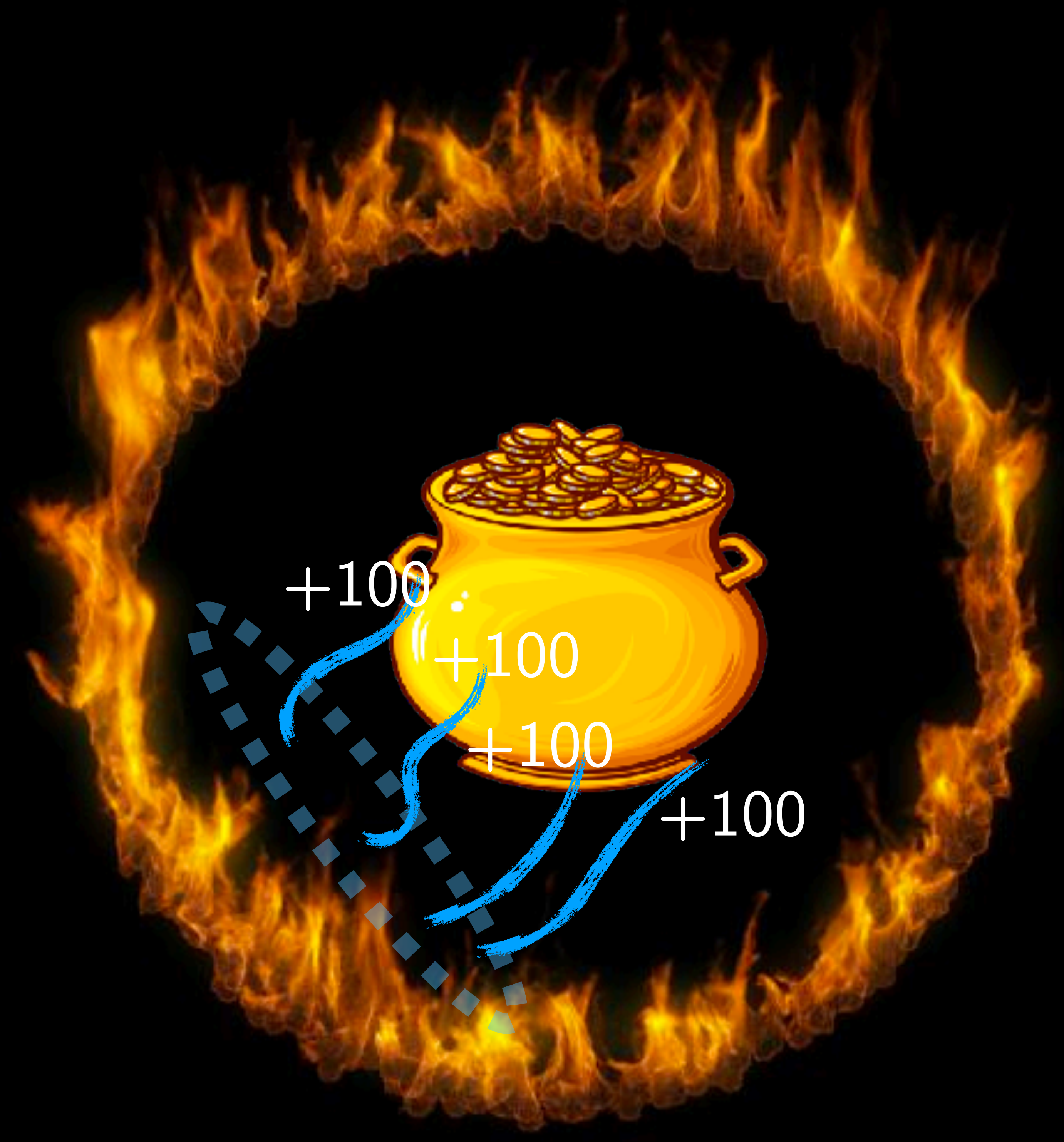
Start distribution



Idea: What if we had a "good reset distribution?"

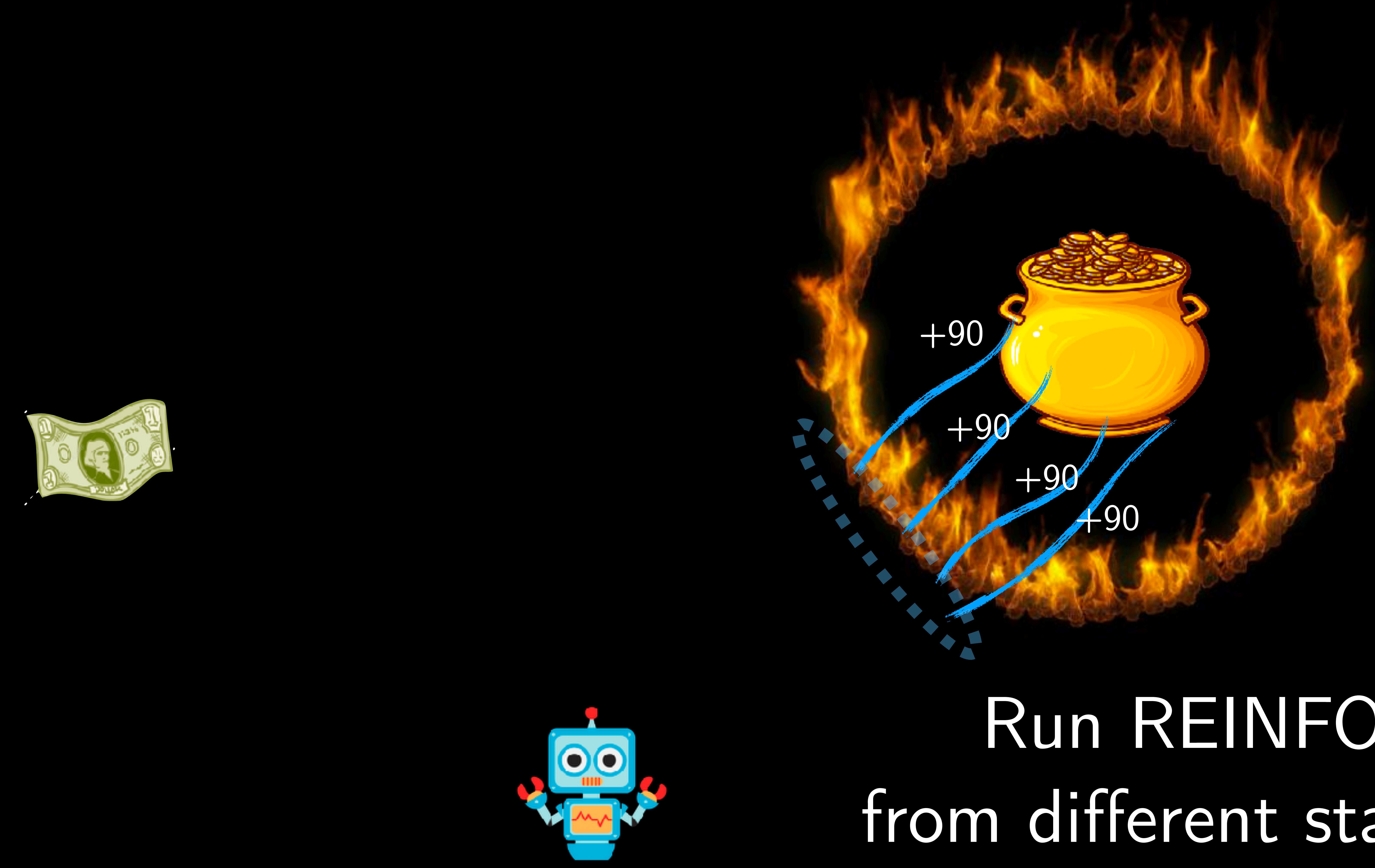


Idea: What if we had a "good reset distribution?"



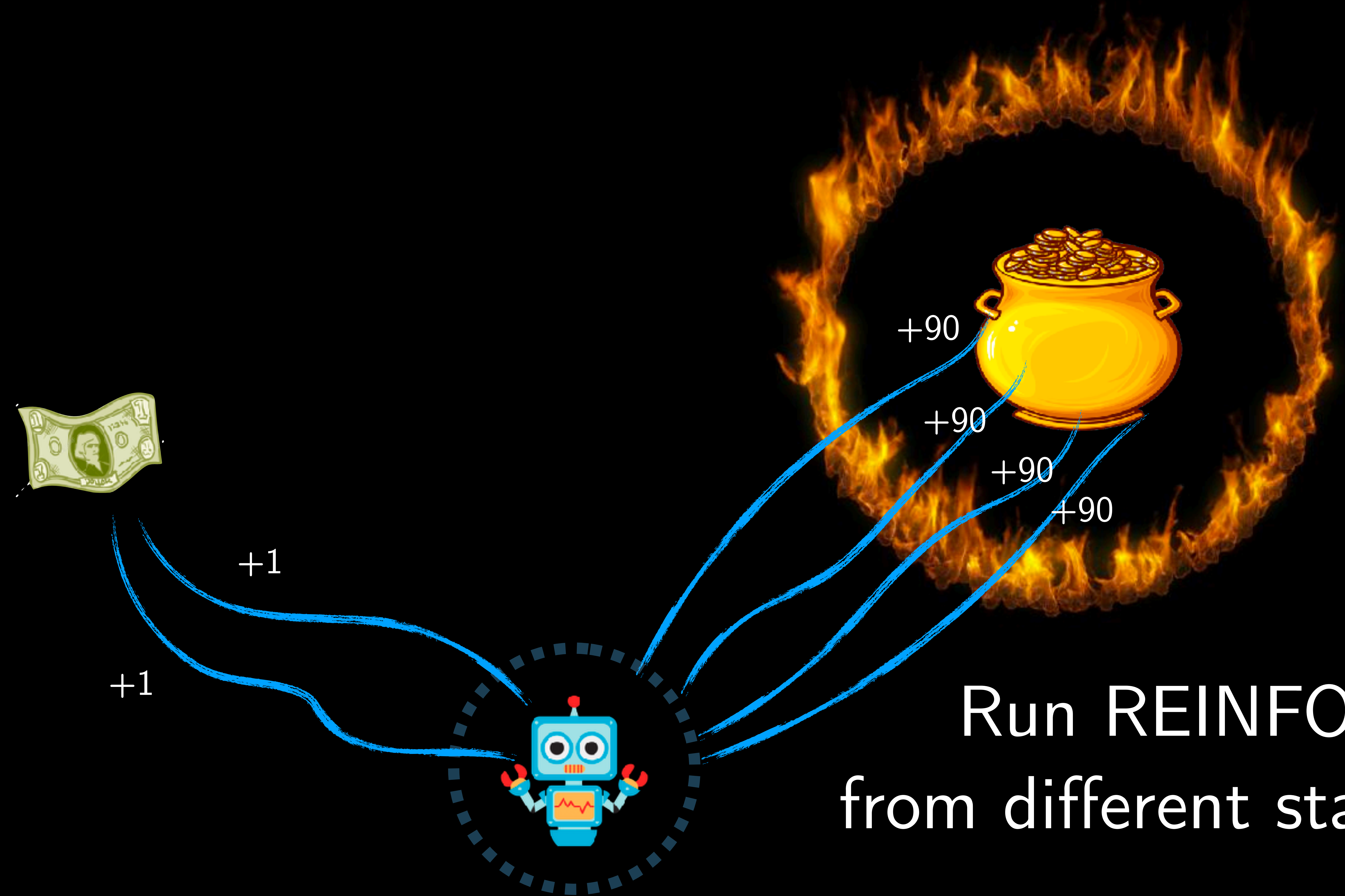
Run REINFORCE
from different start states

Idea: What if we had a "good reset distribution?"



Run REINFORCE
from different start states

Idea: What if we had a "good reset distribution?"



Run REINFORCE
from different start states