Policy Gradient

CMPT 729 G100

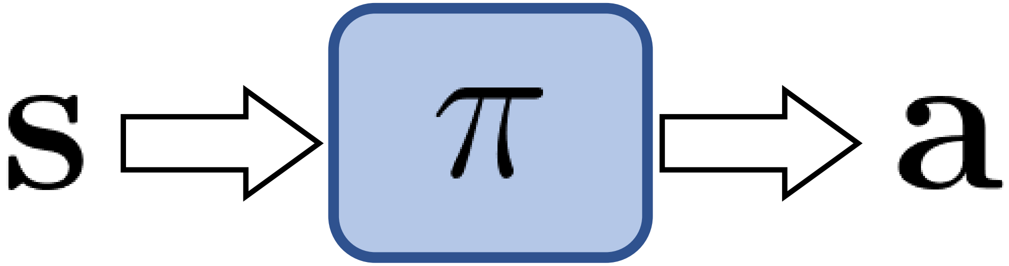
Jason Peng

Overview

- Taxonomy of RL Algorithms
- Policy Gradient
- Derivation
- Variance Reduction
- Applications
- General View of PG

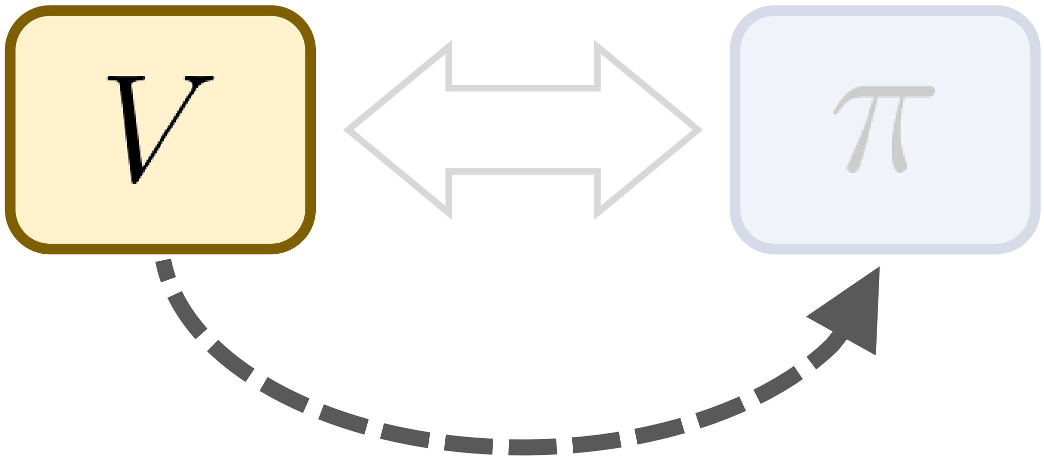
Taxonomy of RL Algorithms

- Policy-Based Methods
- Value-Based Methods
- Actor-Critic Methods
- Model-Based Methods



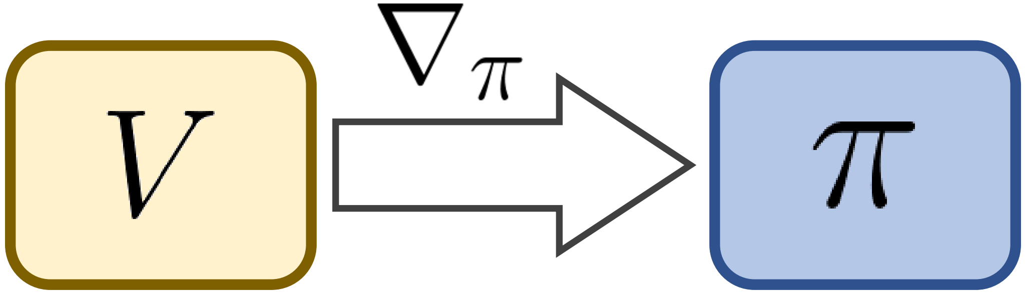
Taxonomy of RL Algorithms

- Policy-Based Methods
- Value-Based Methods
- Actor-Critic Methods
- Model-Based Methods



Taxonomy of RL Algorithms

- Policy-Based Methods
- Value-Based Methods
- **Actor-Critic Methods**
- Model-Based Methods



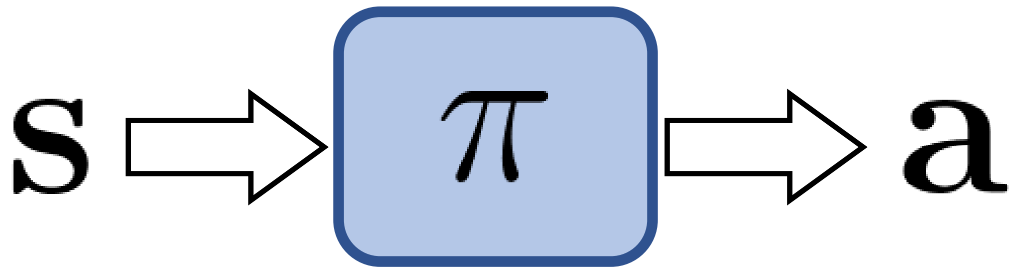
Taxonomy of RL Algorithms

- Policy-Based Methods
- Value-Based Methods
- Actor-Critic Methods
- Model-Based Methods

$$\boxed{p(\mathbf{s}'|\mathbf{s},\mathbf{a})} \Rightarrow \boxed{\pi}$$

Taxonomy of RL Algorithms

- Policy-Based Methods
- Value-Based Methods
- Actor-Critic Methods
- Model-Based Methods



Nondifferentiable Objective

$$\theta^* = \arg\max_{\theta} \; J(\pi_\theta)$$

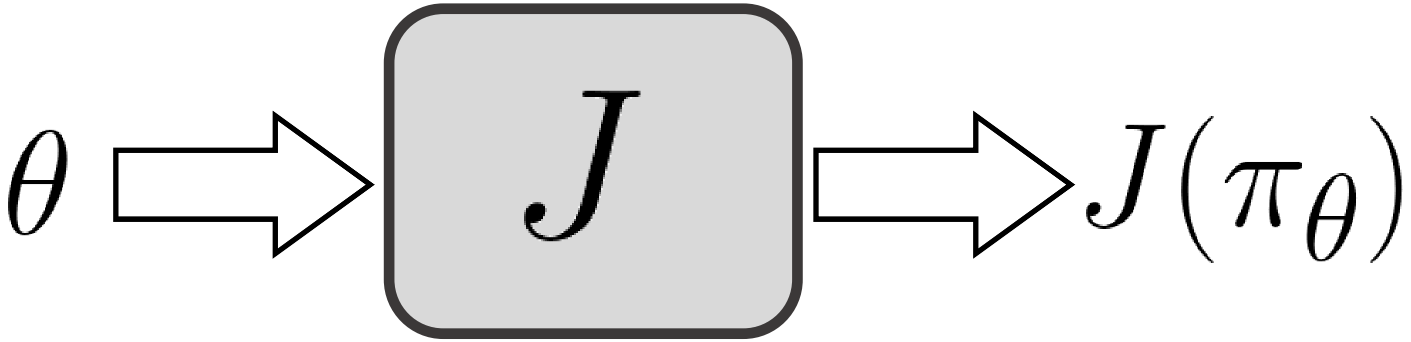
Just use gradient ascent!

Objective is often NOT differentiable

$$\cancel{\nabla_\theta J(\pi_\theta)}$$

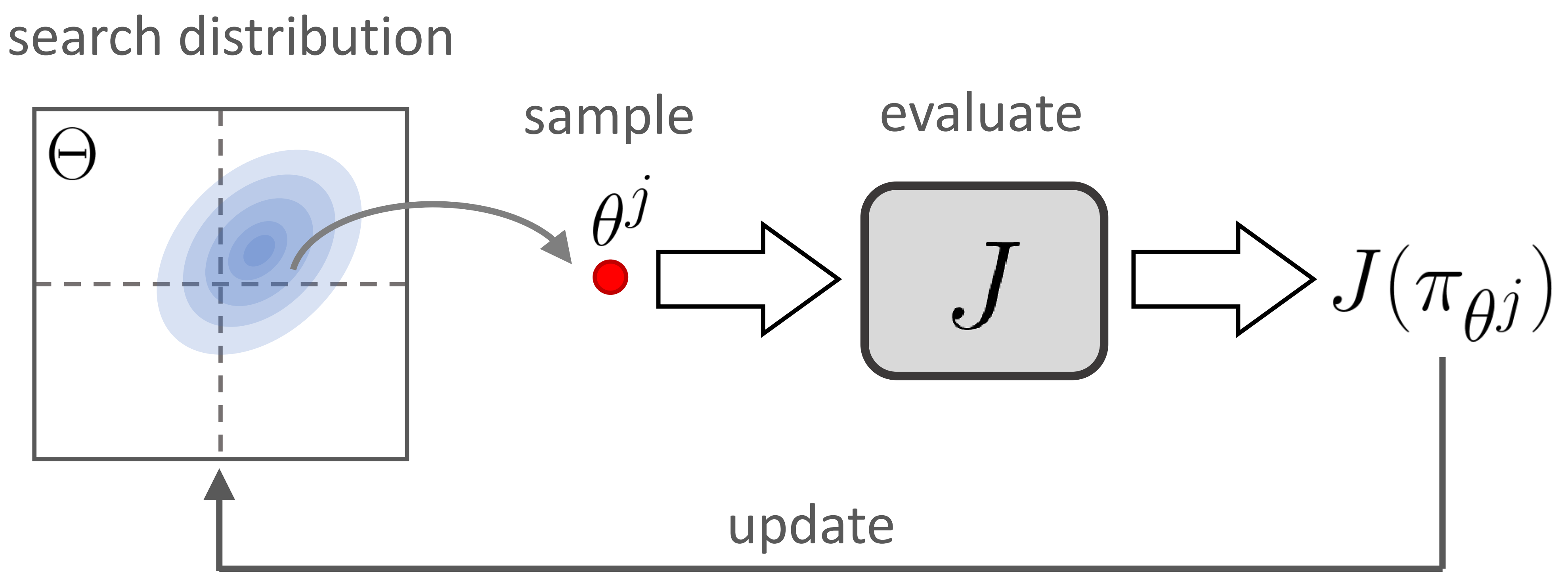
Black Box Optimization

$$\theta^* = \arg\max_{\theta} \underbrace{J(\pi_\theta)}_{\text{black box}}$$



Black Box Optimization

- Adapt search samples base on objective

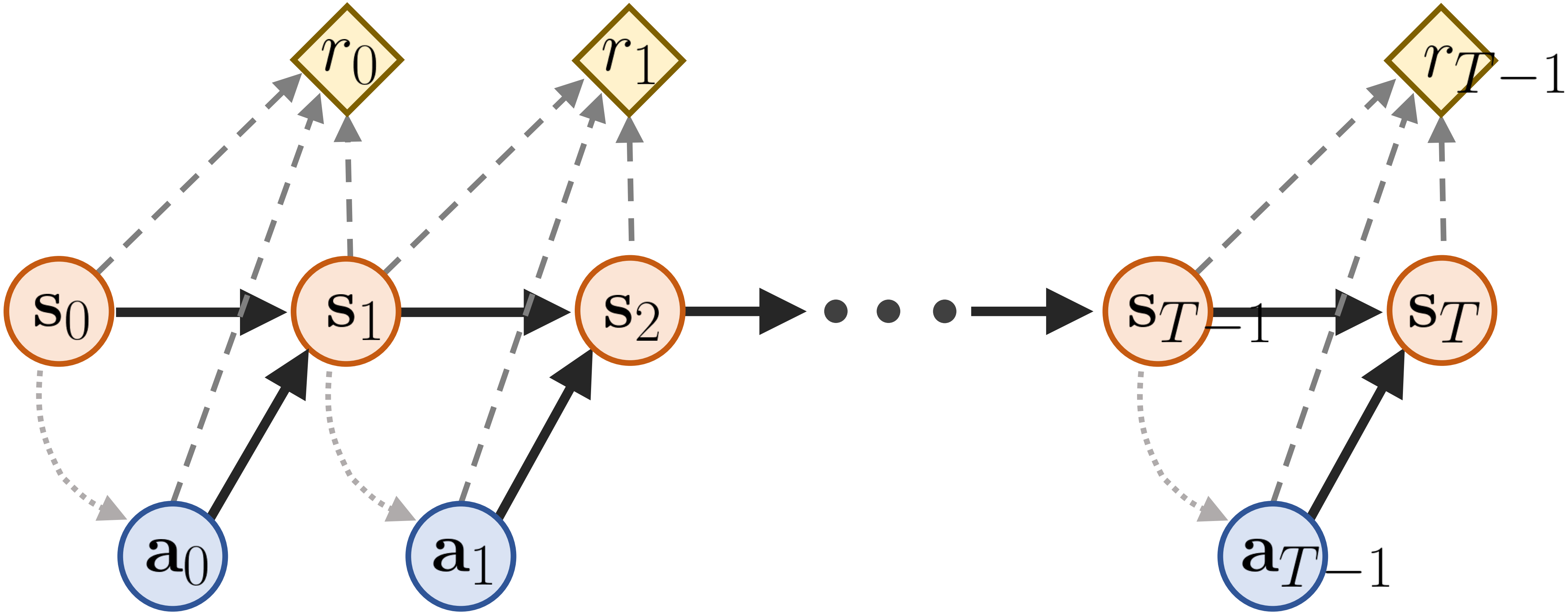


Black Box Optimization

$$\theta^* = \arg\max_{\theta} \; J(\pi_\theta)$$

$$\theta \Rightarrow \boxed{J} \Rightarrow J(\pi_\theta)$$

MDP



r_0
r_1
r_{T-1}
$\mathbf{s}_0$
$\mathbf{s}_1$
$\mathbf{s}_2$
$\mathbf{s}_{T-1}$
$\mathbf{s}_T$
$\mathbf{a}_0$
$\mathbf{a}_1$
$\mathbf{a}_{T-1}$

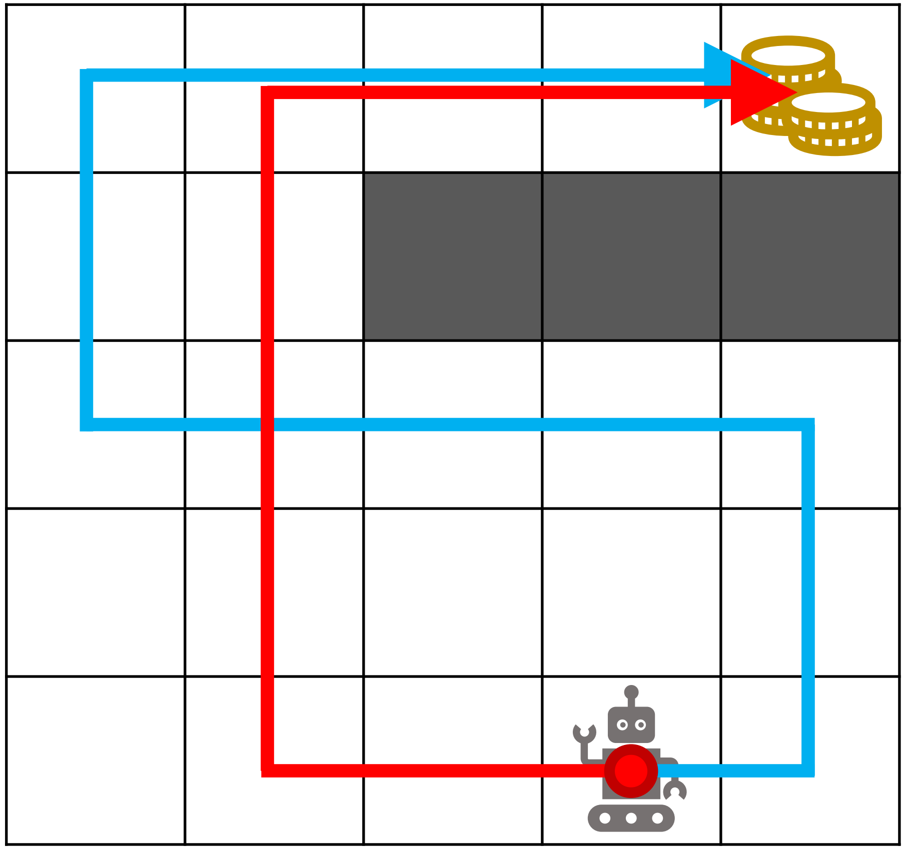
Behavioral Timescales

- Lifetime

Behavioral Timescales

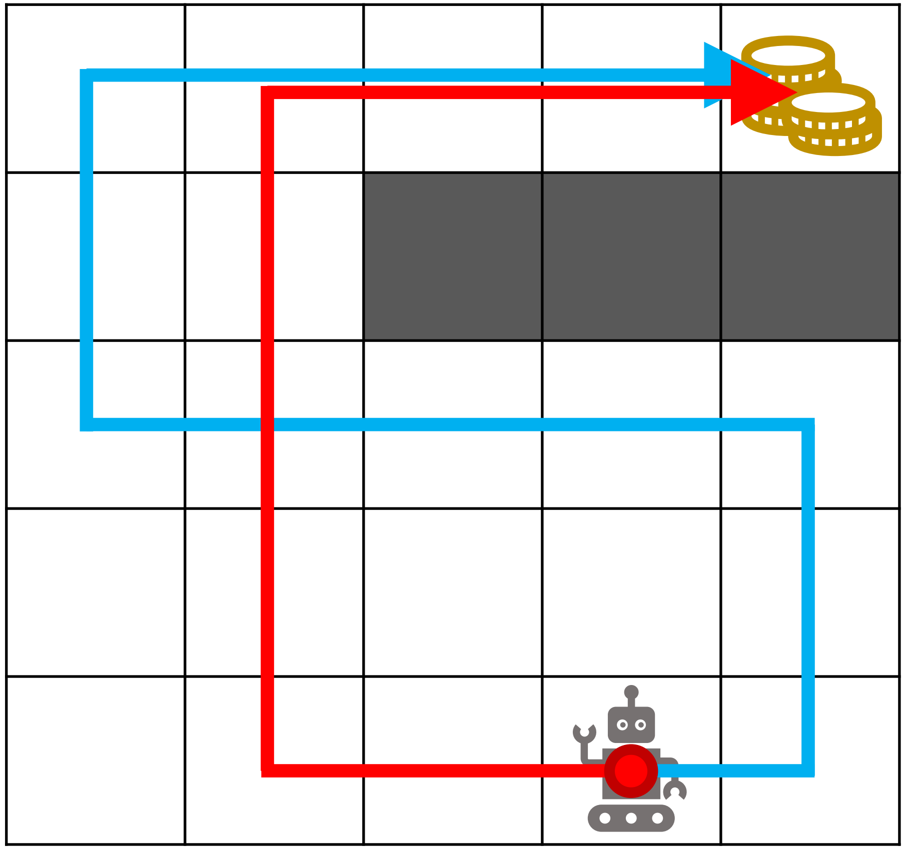
- Lifetime

Evolutionary Methods



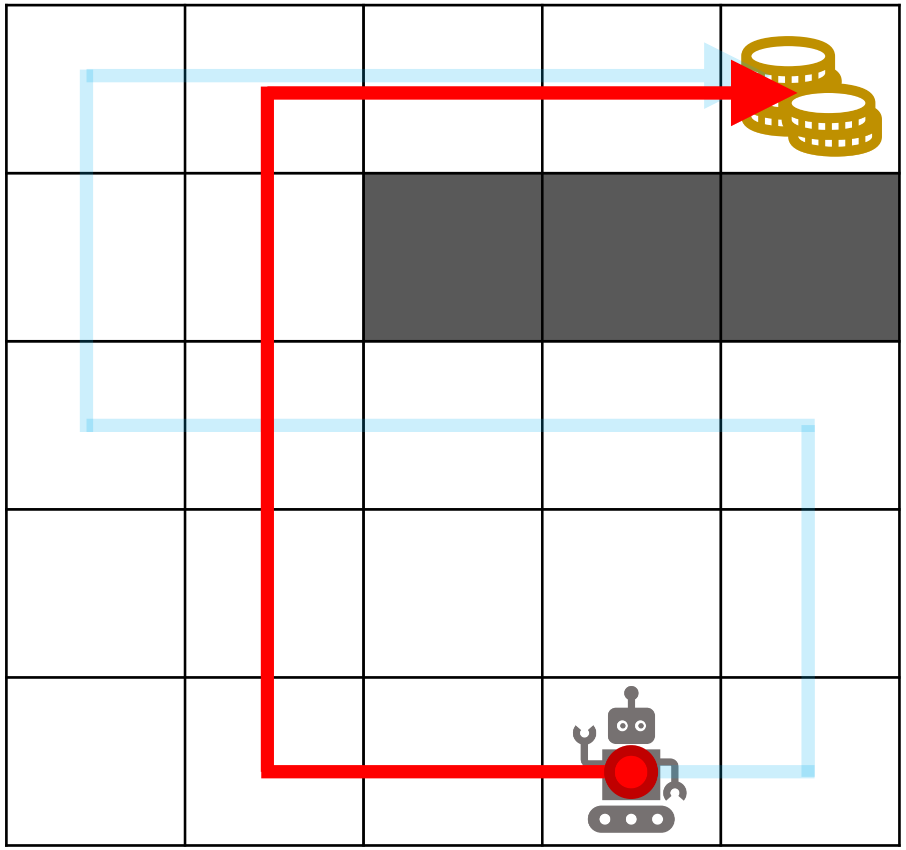
Behavioral Timescales

- Lifetime
- Trajectories



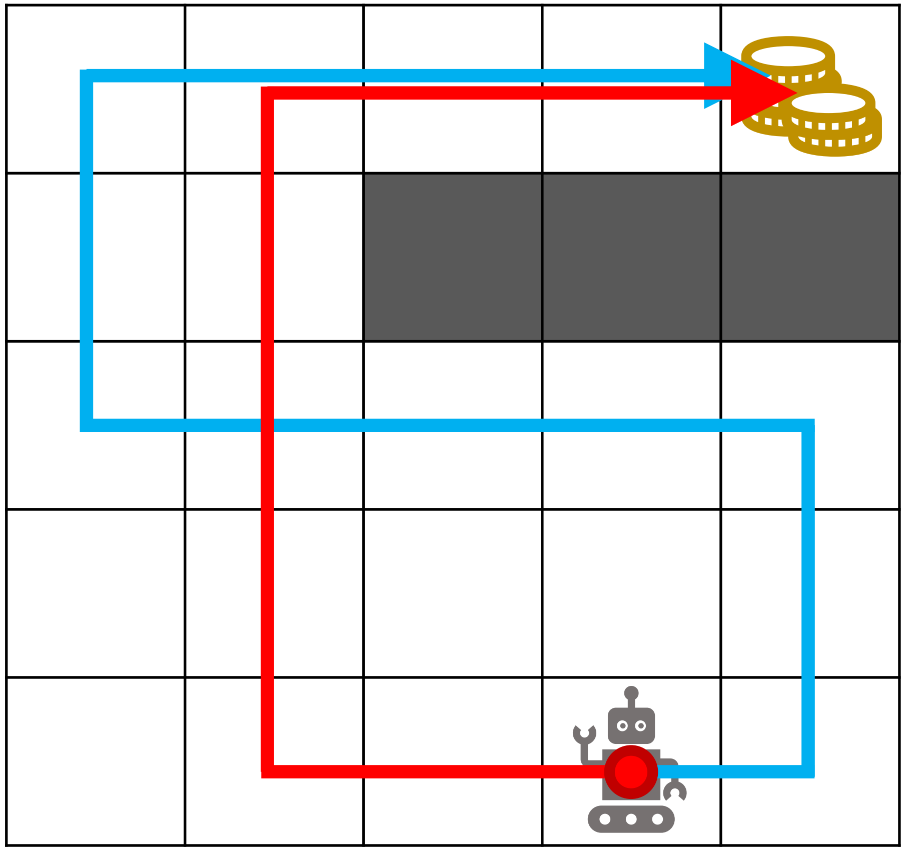
Behavioral Timescales

- Lifetime
- Trajectories



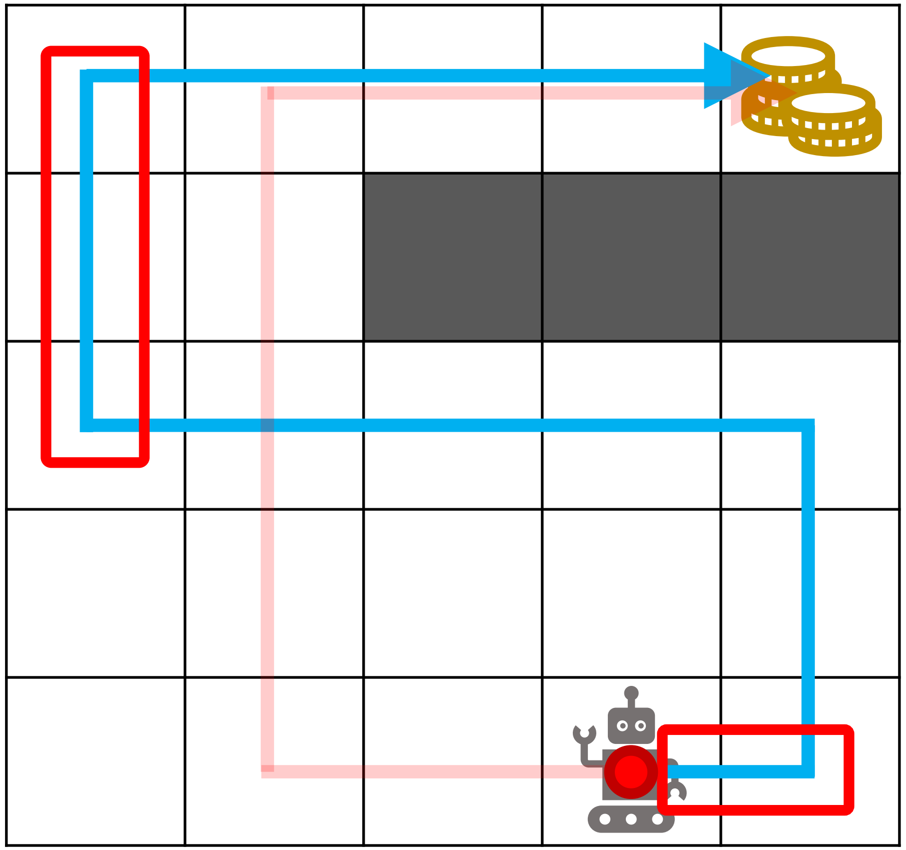
Behavioral Timescales

- Lifetime
- Trajectories
- Actions



Behavioral Timescales

- Lifetime
- Trajectories
- Actions



Nondifferentiable Objective

$$\theta^* = \arg\max_{\theta} \boxed{J(\pi_\theta)}$$

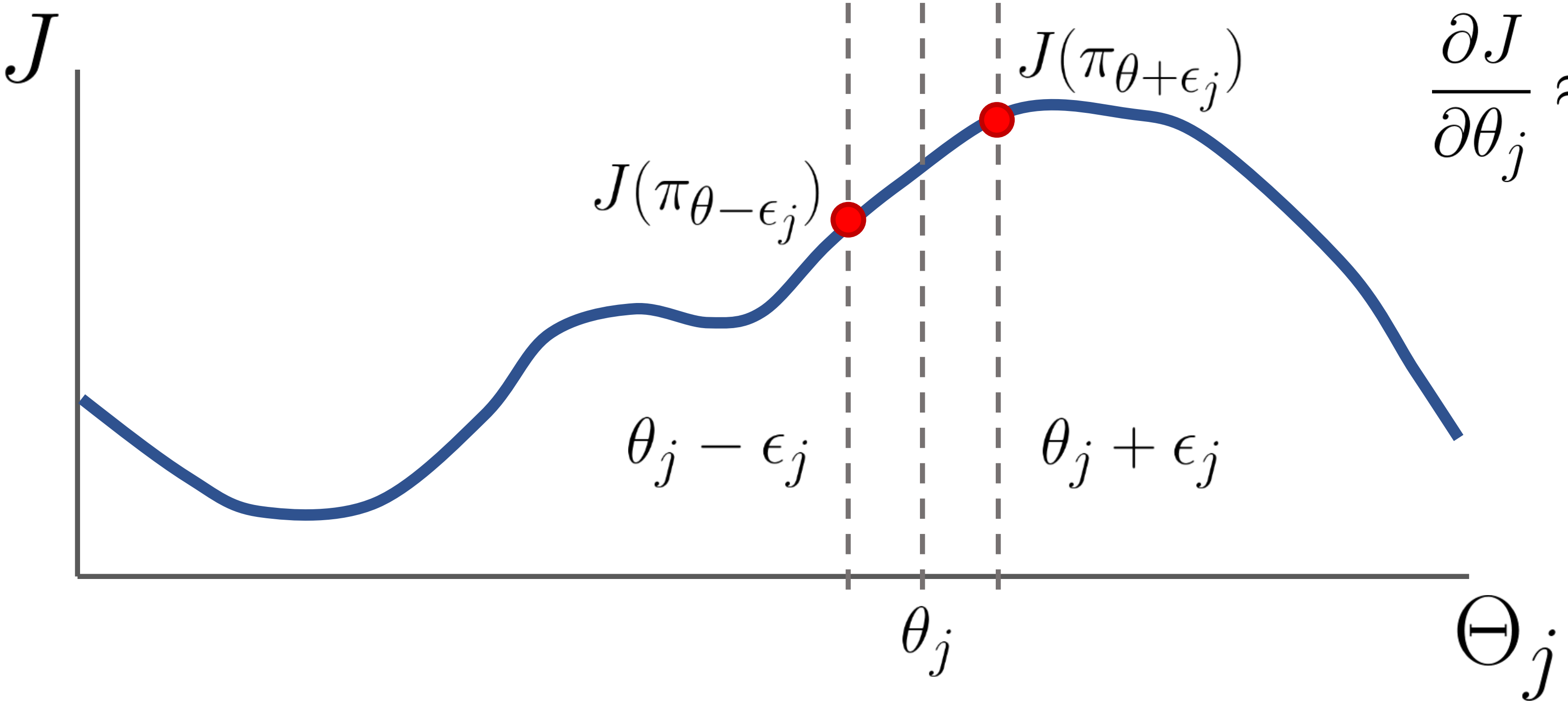
nondifferentiable

$$\nabla_\theta J(\pi_\theta)$$

Can we approximate?

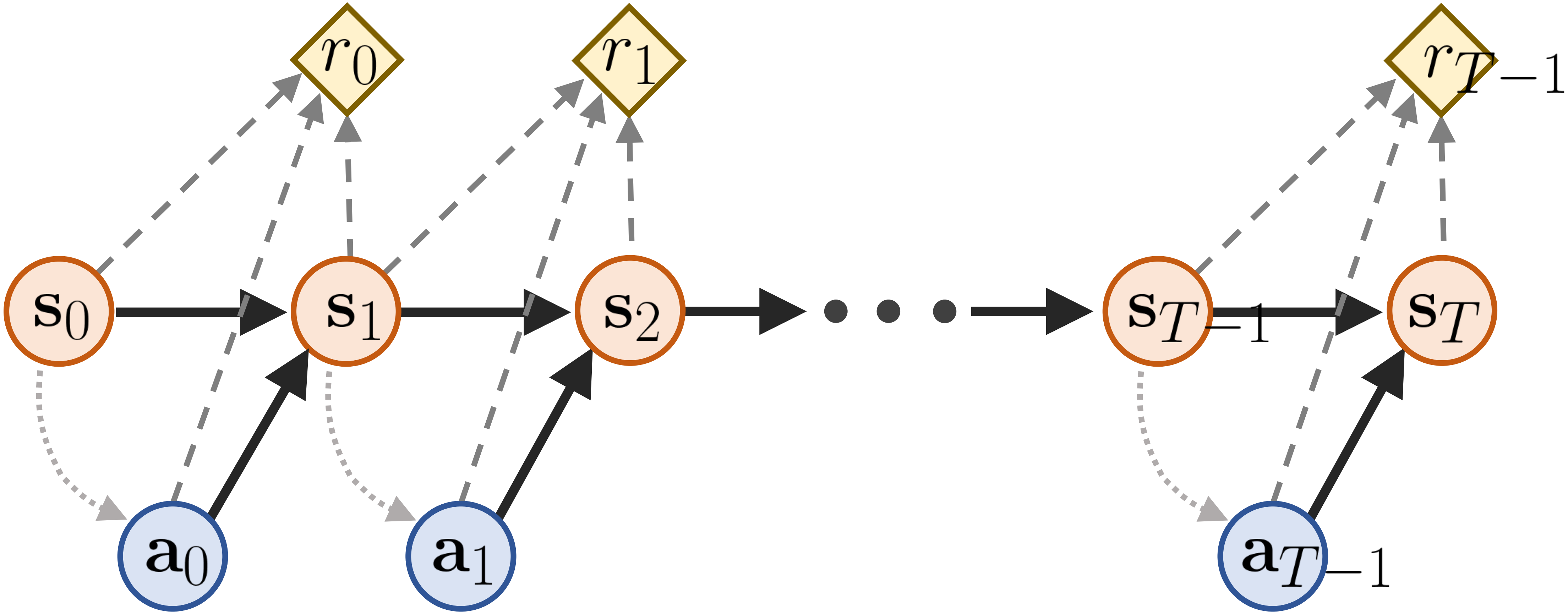
Finite-Differences

- Approximate gradient using finite-differences



$$\frac{\partial J}{\partial \theta_j} \approx \frac{J(\pi_{\theta+\epsilon_j}) - J(\pi_{\theta-\epsilon_j})}{2\epsilon}$$

MDP



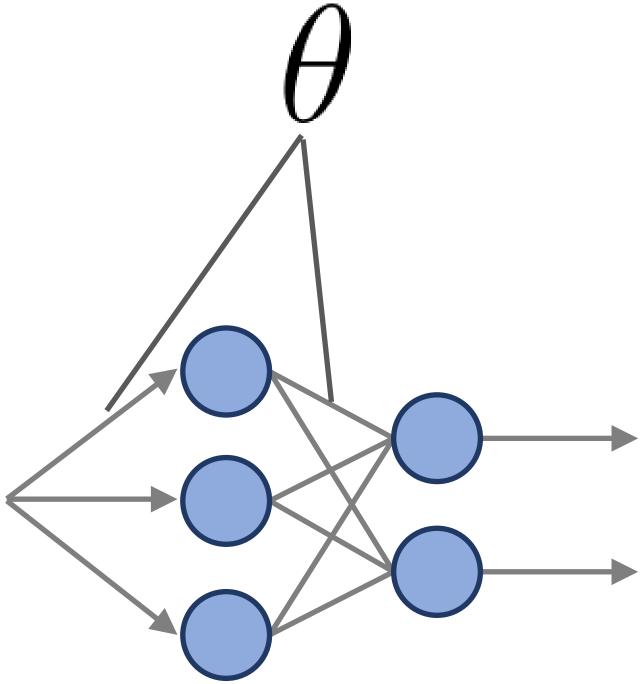
r_0
r_1
r_{T-1}
$\mathbf{s}_0$
$\mathbf{s}_1$
$\mathbf{s}_2$
$\mathbf{s}_{T-1}$
$\mathbf{s}_T$
$\mathbf{a}_0$
$\mathbf{a}_1$
$\mathbf{a}_{T-1}$

Notation

$$\nabla_\theta J(\pi_\theta)$$

$$\Downarrow$$

$$\nabla_\pi J(\pi)$$



Policy Gradients

$$J(\pi) = \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[\sum_{t=0}^{T-1} \gamma^t r_t \right] = \mathbb{E}_{\tau \sim p(\tau|\pi)} \underbrace{[R(\tau)]}_{\text{return of a trajectory}}$$

return of a trajectory

Policy Gradients

$$J(\pi) = \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[\sum_{t=0}^{T-1} \gamma^t r_t \right] = \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[R(\tau) \right]$$

$$= \sum_{\tau} \underline{p(\tau|\pi)} R(\tau)$$

Policy Gradients

$$J(\pi) = \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[\sum_{t=0}^{T-1} \gamma^t r_t \right] = \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[R(\tau) \right]$$

$$= \sum_{\tau} p(\tau|\pi) \underline{R(\tau)}$$

Policy Gradients

$$J(\pi) = \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[\sum_{t=0}^{T-1} \gamma^t r_t \right] = \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[R(\tau) \right]$$

$$= \sum_{\tau} p(\tau|\pi) R(\tau)$$

$$\nabla_\pi J(\pi) = \sum_{\tau} \underline{\nabla_\pi p(\tau|\pi)} R(\tau)$$

completely intractable

Policy Gradients

$$\nabla_\pi J(\pi) = \sum_\tau \nabla_\pi p(\tau|\pi) R(\tau)$$

Score Function

$$\nabla_p p(x) = p(x)\frac{\nabla_p p(x)}{p(x)} = \underline{p(x)}\nabla_p \mathrm{log} p(x)$$

Policy Gradients

$$\nabla_\pi J(\pi) = \sum_\tau \nabla_\pi p(\tau|\pi) R(\tau)$$

$$= \sum_\tau \underline{p(\tau|\pi)} \nabla_\pi \text{log} p(\tau|\pi) R(\tau)$$

Score Function

$$\nabla_p p(x) = p(x) \frac{\nabla_p p(x)}{p(x)} = p(x) \nabla_p \text{log} p(x)$$

Policy Gradients

$$\nabla_\pi J(\pi) = \sum_\tau \nabla_\pi p(\tau|\pi)R(\tau)$$
$$= \sum_\tau p(\tau|\pi)\nabla_\pi \text{log} p(\tau|\pi)R(\tau)$$

Score Function

$$\nabla_p p(x) = p(x)\frac{\nabla_p p(x)}{p(x)} = p(x)\nabla_p \text{log} p(x)$$

Policy Gradients

$$\nabla_\pi J(\pi) = \sum_\tau \nabla_\pi p(\tau|\pi) R(\tau)$$

$$= \sum_\tau p(\tau|\pi) \nabla_\pi \text{log} p(\tau|\pi) R(\tau)$$

$$= \underline{\mathbb{E}_{\tau \sim p(\tau|\pi)}} [\nabla_\pi \text{log} p(\tau|\pi) R(\tau)]$$

Score Function

$$\nabla_p p(x) = p(x) \frac{\nabla_p p(x)}{p(x)} = p(x) \nabla_p \text{log} p(x)$$

Policy Gradients

$$\nabla_\pi J(\pi) = \sum_\tau \nabla_\pi p(\tau|\pi) R(\tau)$$
$$= \sum_\tau p(\tau|\pi) \nabla_\pi \text{log} p(\tau|\pi) R(\tau)$$
$$= \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[\underline{\nabla_\pi \text{log} p(\tau|\pi)} R(\tau)\right]$$

Score Function

$$\nabla_p p(x) = p(x) \frac{\nabla_p p(x)}{p(x)} = p(x) \nabla_p \text{log} p(x)$$

Policy Gradients

$$\nabla_\pi \log p(\tau|\pi) = \nabla_\pi \log \underbrace{\left(p(\mathbf{s}_0) \prod_{t=0}^{T-1} \pi(\mathbf{a}_t|\mathbf{s}_t) p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t) \right)}_{= p(\tau|\pi)}$$

Policy Gradients

$$\nabla_\pi \log p(\tau|\pi) = \nabla_\pi \log \left(\underline{p(\mathbf{s}_0)} \prod_{t=0}^{T-1} \pi(\mathbf{a}_t|\mathbf{s}_t) p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t) \right)$$

Policy Gradients

$$\nabla_\pi \log p(\tau|\pi) = \nabla_\pi \log \left(p(\mathbf{s}_0) \underline{\prod_{t=0}^{T-1} \pi(\mathbf{a}_t|\mathbf{s}_t) p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)} \right)$$

Policy Gradients

$$\nabla_\pi \log p(\tau|\pi) = \nabla_\pi \log \left(p(\mathbf{s}_0) \prod_{t=0}^{T-1} \underline{\pi(\mathbf{a}_t|\mathbf{s}_t)} p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t) \right)$$

Policy Gradients

$$\nabla_\pi \log p(\tau|\pi) = \nabla_\pi \log \left(p(\mathbf{s}_0) \prod_{t=0}^{T-1} \pi(\mathbf{a}_t|\mathbf{s}_t) \underline{p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)} \right)$$

Policy Gradients

$$\nabla_\pi \log p(\tau|\pi) = \nabla_\pi \log \left(p(\mathbf{s}_0) \prod_{t=0}^{T-1} \pi(\mathbf{a}_t|\mathbf{s}_t) p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t) \right)$$

$$= \nabla_\pi \left(\log p(\mathbf{s}_0) + \sum_{t=0}^{T-1} \underline{\log \pi(\mathbf{a}_t|\mathbf{s}_t)} + \log p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t) \right)$$

Policy Gradients

$$\nabla_\pi \log p(\tau|\pi) = \nabla_\pi \log \left(p(\mathbf{s}_0) \prod_{t=0}^{T-1} \pi(\mathbf{a}_t|\mathbf{s}_t) p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t) \right)$$

$$= \nabla_\pi \left(\underline{\log p(\mathbf{s}_0)} + \sum_{t=0}^{T-1} \log \pi(\mathbf{a}_t|\mathbf{s}_t) + \underline{\log p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)} \right)$$

Independent of π

Policy Gradients

$$\nabla_\pi \log p(\tau|\pi) = \nabla_\pi \log \left(p(\mathbf{s}_0) \prod_{t=0}^{T-1} \pi(\mathbf{a}_t|\mathbf{s}_t) p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t) \right)$$

$$= \nabla_\pi \left(\cancel{\log p(\mathbf{s}_0)} + \sum_{t=0}^{T-1} \log \pi(\mathbf{a}_t|\mathbf{s}_t) + \cancel{\log p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)} \right)$$

$$= \sum_{t=0}^{T-1} \nabla_\pi \log \pi(\mathbf{a}_t|\mathbf{s}_t)$$

Policy Gradients

$$\nabla_\pi J(\pi) = \sum_\tau \nabla_\pi p(\tau|\pi) R(\tau)$$
$$= \sum_\tau p(\tau|\pi) \nabla_\pi \text{log} p(\tau|\pi) R(\tau)$$
$$= \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[\nabla_\pi \text{log} p(\tau|\pi) R(\tau) \right]$$
$$= \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[R(\tau) \sum_{t=0}^{T-1} \nabla_\pi \text{log} \pi(\mathbf{a}_t|\mathbf{s}_t) \right]$$

policy gradient
AKA. REINFORCE [Williams 1992]

Score Function

$$\nabla_\pi \pi(\tau) = \pi(\tau) \frac{\nabla_\pi \pi(\tau)}{\pi(\tau)} = \pi(\tau) \nabla_\pi \text{log} \pi(\tau)$$

Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning [Williams 1992]

REINFORCE

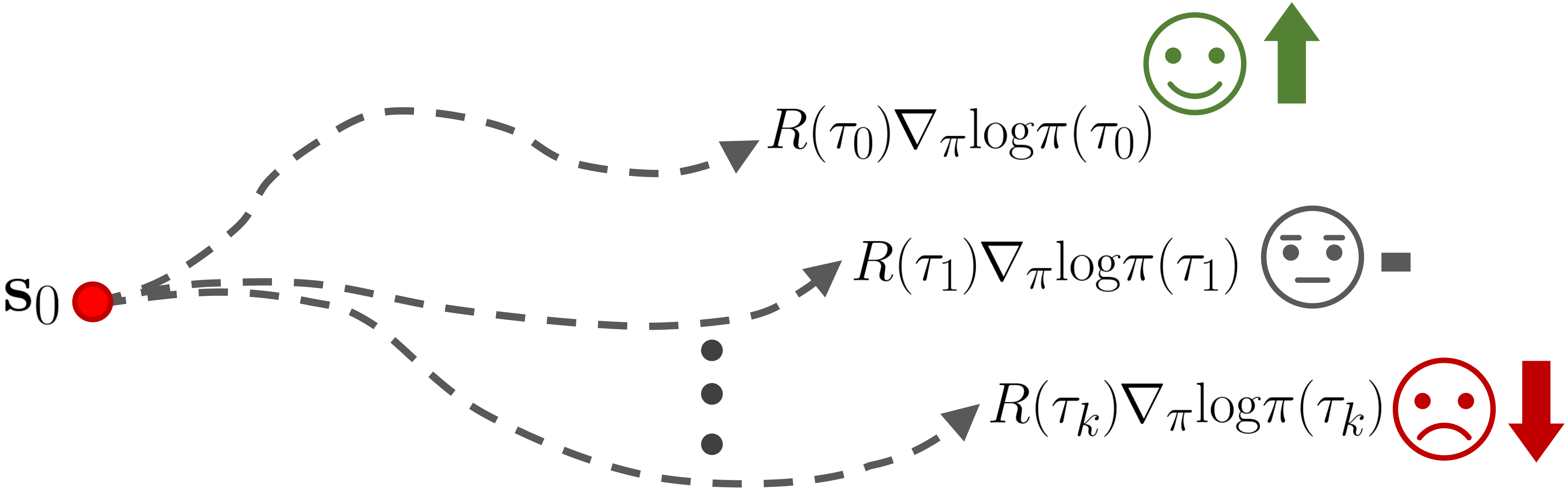
$$\nabla_\pi J(\pi) = \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[R(\tau) \sum_{t=0}^{T-1} \underline{\nabla_\pi \log \pi(\mathbf{a}_t|\mathbf{s}_t)} \right]$$

REINFORCE

$$\nabla_\pi J(\pi) = \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[\underline{R(\tau)} \sum_{t=0}^{T-1} \nabla_\pi \log \pi(\mathbf{a}_t|\mathbf{s}_t) \right]$$

REINFORCE

$$\nabla_\pi J(\pi) = \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[R(\tau) \sum_{t=0}^{T-1} \nabla_\pi \log \pi(\mathbf{a}_t | \mathbf{s}_t) \right]$$



REINFORCE

ALGORITHM: REINFORCE

1: $\theta \leftarrow$ initialize policy parameters

2: **while** not done **do**

3: Sample trajectories $\{\tau^i\}$ from policy $\pi_\theta(\mathbf{a}|\mathbf{s})$

4: Estimate policy gradient

$\nabla_\theta J(\pi_\theta) \approx \frac{1}{N}\sum_i R(\tau^i)\sum_t \nabla_\theta \log\pi_\theta(\mathbf{a}_t^i|\mathbf{s}_t^i)$

5: Update policy $\theta \leftarrow \theta + \alpha\nabla_\theta J(\pi_\theta)$

6: **end while**

7: return policy π_θ

Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning [Williams 1992]

REINFORCE

ALGORITHM: REINFORCE

1: $\theta \leftarrow$ initialize policy parameters

2: **while** not done **do**

3: Sample trajectories $\{\tau^i\}$ from policy $\pi_\theta(\mathbf{a}|\mathbf{s})$

4: Estimate policy gradient

$\nabla_\theta J(\pi_\theta) \approx \frac{1}{N} \sum_i R(\tau^i) \sum_t \nabla_\theta \log \pi_\theta(\mathbf{a}_t^i|\mathbf{s}_t^i)$

5: Update policy $\theta \leftarrow \theta + \alpha \nabla_\theta J(\pi_\theta)$

6: **end while**

7: return policy π_θ

Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning [Williams 1992]

REINFORCE

ALGORITHM: REINFORCE

1: $\theta \leftarrow$ initialize policy parameters

2: **while** not done **do**

3: Sample trajectories $\{\tau^i\}$ from policy $\pi_\theta(\mathbf{a}|\mathbf{s})$

4: Estimate policy gradient
$\nabla_\theta J(\pi_\theta) \approx \frac{1}{N}\sum_i R(\tau^i)\sum_t \nabla_\theta \log\pi_\theta(\mathbf{a}_t^i|\mathbf{s}_t^i)$

5: Update policy $\theta \leftarrow \theta + \alpha\nabla_\theta J(\pi_\theta)$

6: **end while**

7: return policy π_θ

Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning [Williams 1992]

REINFORCE

ALGORITHM: REINFORCE

1: $\theta \leftarrow$ initialize policy parameters

2: **while** not done **do**

3: Sample trajectories $\{\tau^i\}$ from policy $\pi_\theta(\mathbf{a}|\mathbf{s})$

4: Estimate policy gradient

$\nabla_\theta J(\pi_\theta) \approx \frac{1}{N}\sum_i R(\tau^i)\sum_t \nabla_\theta \log \pi_\theta(\mathbf{a}_t^i|\mathbf{s}_t^i)$

5: Update policy $\theta \leftarrow \theta + \alpha \nabla_\theta J(\pi_\theta)$

6: **end while**

7: return policy π_θ

Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning [Williams 1992]

REINFORCE

ALGORITHM: REINFORCE

1: $\theta \leftarrow$ initialize policy parameters

2: **while** not done **do**

3: Sample trajectories $\{\tau^i\}$ from policy $\pi_\theta(\mathbf{a}|\mathbf{s})$

4: Estimate policy gradient

$\nabla_\theta J(\pi_\theta) \approx \frac{1}{N}\sum_i R(\tau^i)\sum_t \nabla_\theta \log\pi_\theta(\mathbf{a}_t^i|\mathbf{s}_t^i)$

5: Update policy $\theta \leftarrow \theta + \alpha\nabla_\theta J(\pi_\theta)$

6: **end while**

7: return policy π_θ

Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning [Williams 1992]

REINFORCE

ALGORITHM: REINFORCE

1: $\theta \leftarrow$ initialize policy parameters

2: **while** not done **do**

3: Sample trajectories $\{\tau^i\}$ from policy $\pi_\theta(\mathbf{a}|\mathbf{s})$

4: Estimate policy gradient

$$\nabla_\theta J(\pi_\theta) \approx \frac{1}{N} \sum_i R(\tau^i) \sum_t \nabla_\theta \log \pi_\theta(\mathbf{a}_t^i|\mathbf{s}_t^i)$$

5: Update policy $\theta \leftarrow \theta + \alpha \nabla_\theta J(\pi_\theta)$

6: **end while**

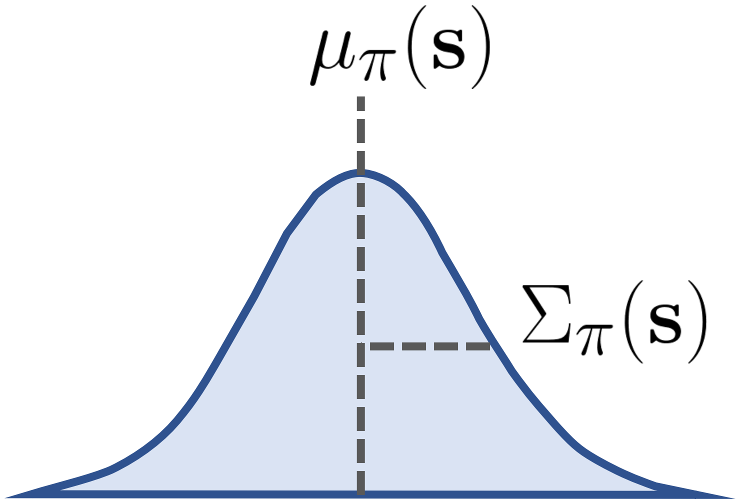
7: return policy π_θ

Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning [Williams 1992]

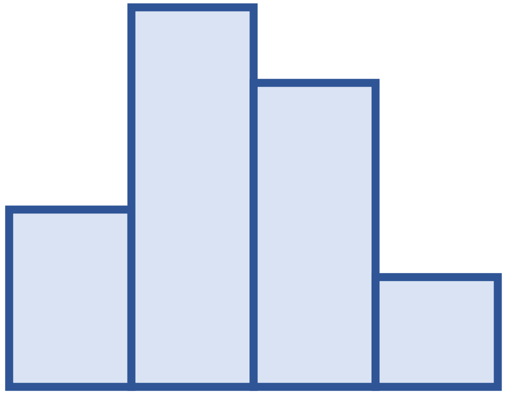
Action Distribution

$$\nabla_{\pi} J(\pi) = \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[R(\tau) \sum_{t=0}^{T-1} \underline{\nabla_{\pi} \log \pi(\mathbf{a}_t | \mathbf{s}_t)} \right]$$

must be differentiable



Gaussian Distribution
(Continuous Actions)

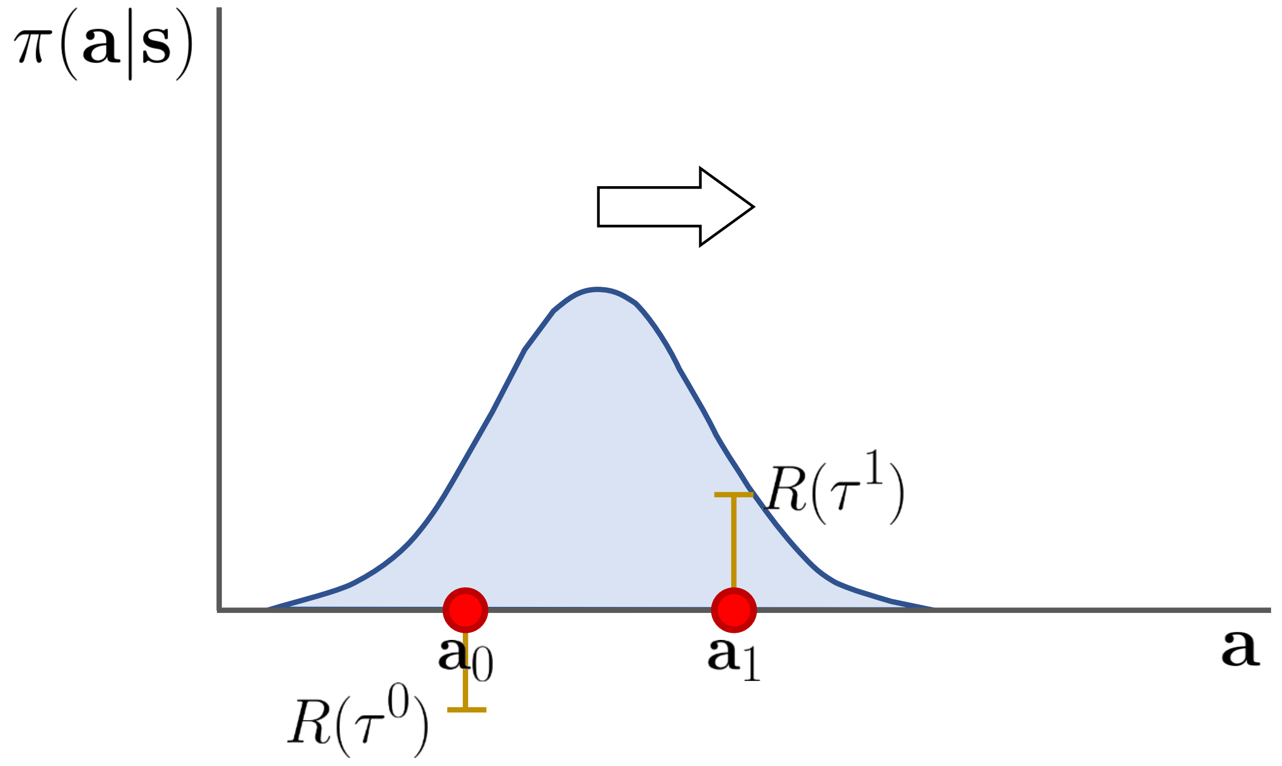


Categorical Distribution
(Discrete Actions)

Etc...

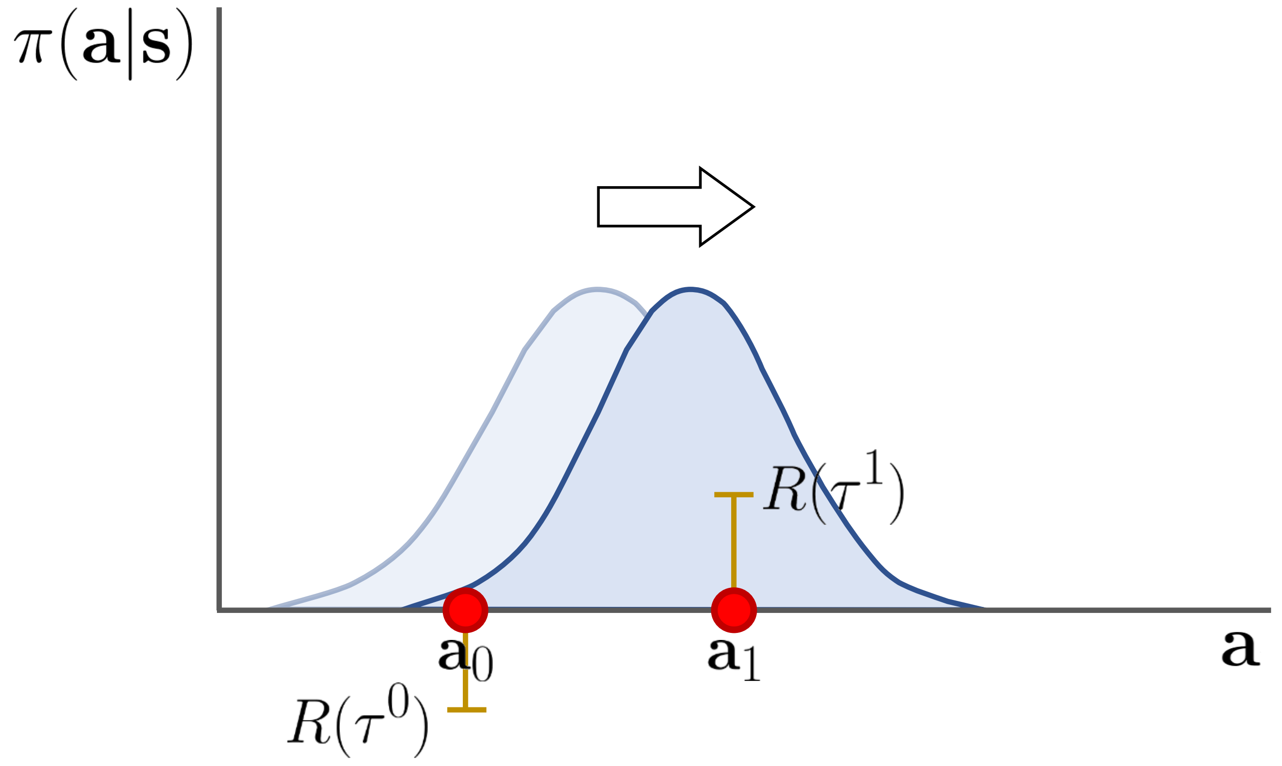
Problems

$$\nabla_\pi J(\pi) = \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[R(\tau) \sum_{t=0}^{T-1} \nabla_\pi \log \pi(\mathbf{a}_t | \mathbf{s}_t) \right]$$



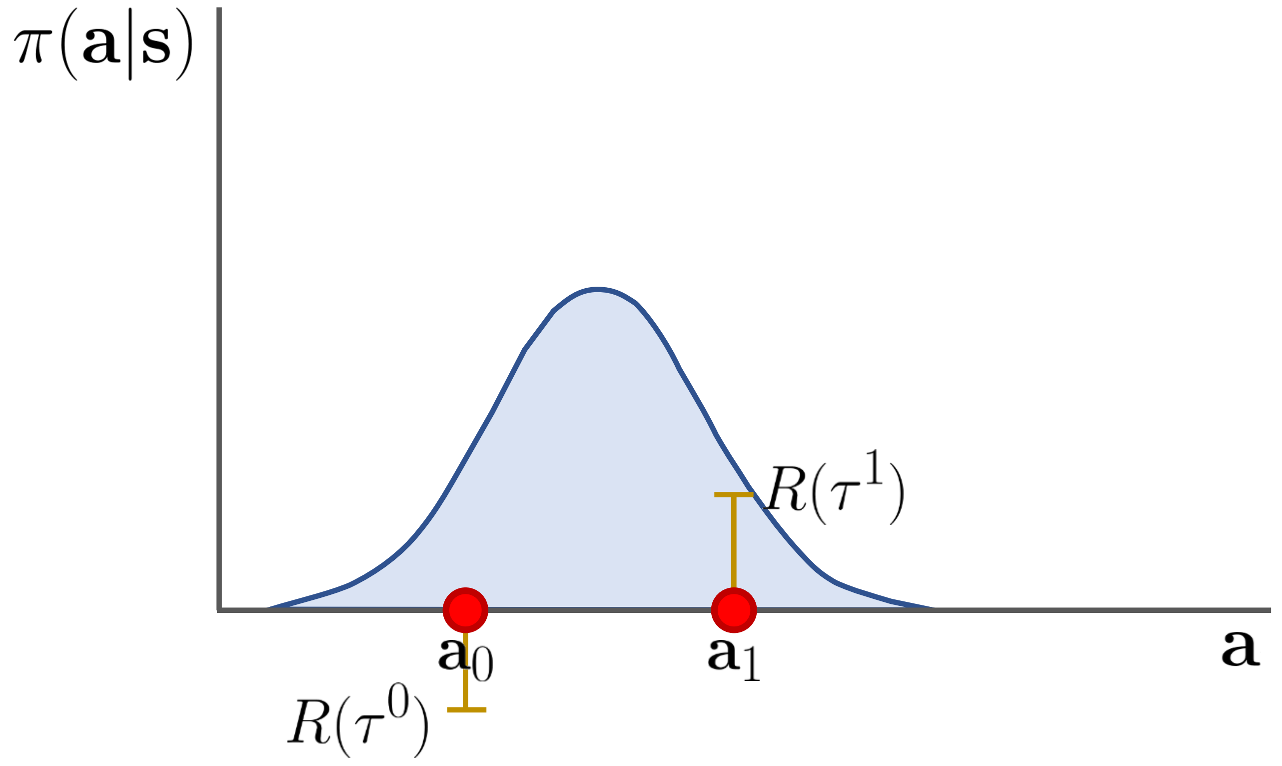
Problems

$$\nabla_\pi J(\pi) = \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[R(\tau) \sum_{t=0}^{T-1} \nabla_\pi \log \pi(\mathbf{a}_t|\mathbf{s}_t) \right]$$



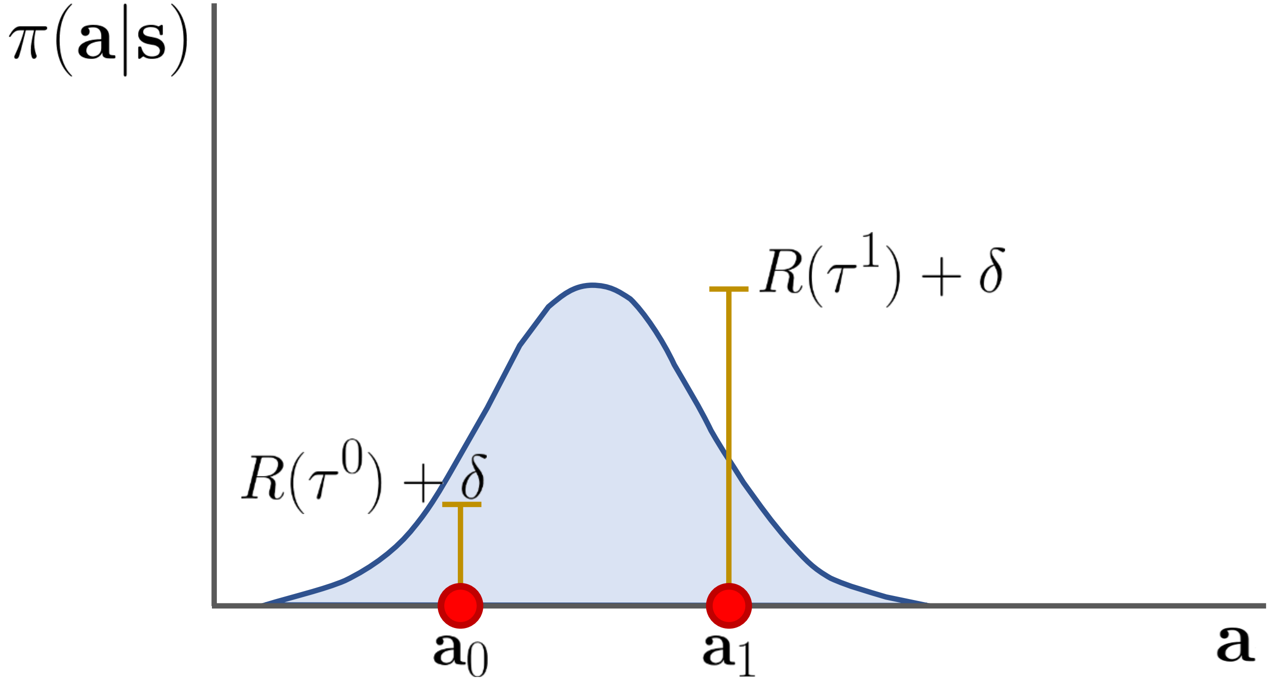
Problems

$$\nabla_\pi J(\pi) = \mathbb{E}_{\tau\sim p(\tau|\pi)}\left[R(\tau)\sum_{t=0}^{T-1}\nabla_\pi \log\pi(\mathbf{a}_t|\mathbf{s}_t)\right]$$



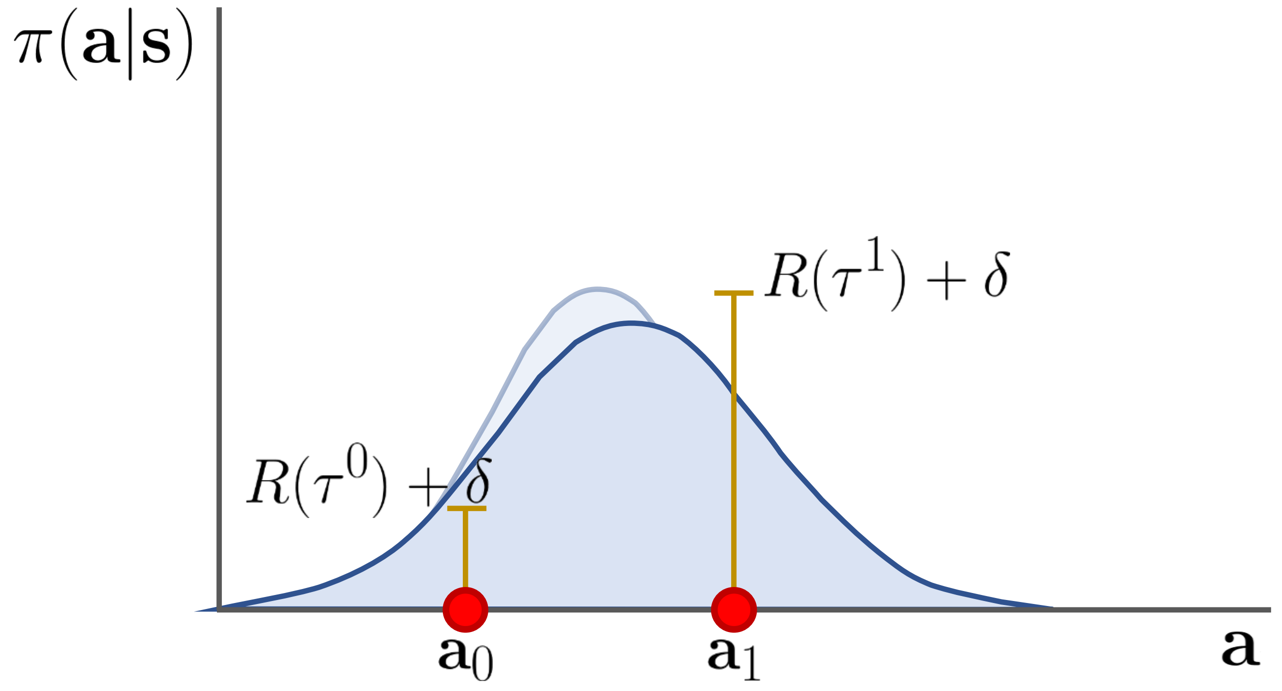
Problems

$$\nabla_\pi J(\pi) = \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[R(\tau) \sum_{t=0}^{T-1} \nabla_\pi \log \pi(\mathbf{a}_t|\mathbf{s}_t) \right]$$



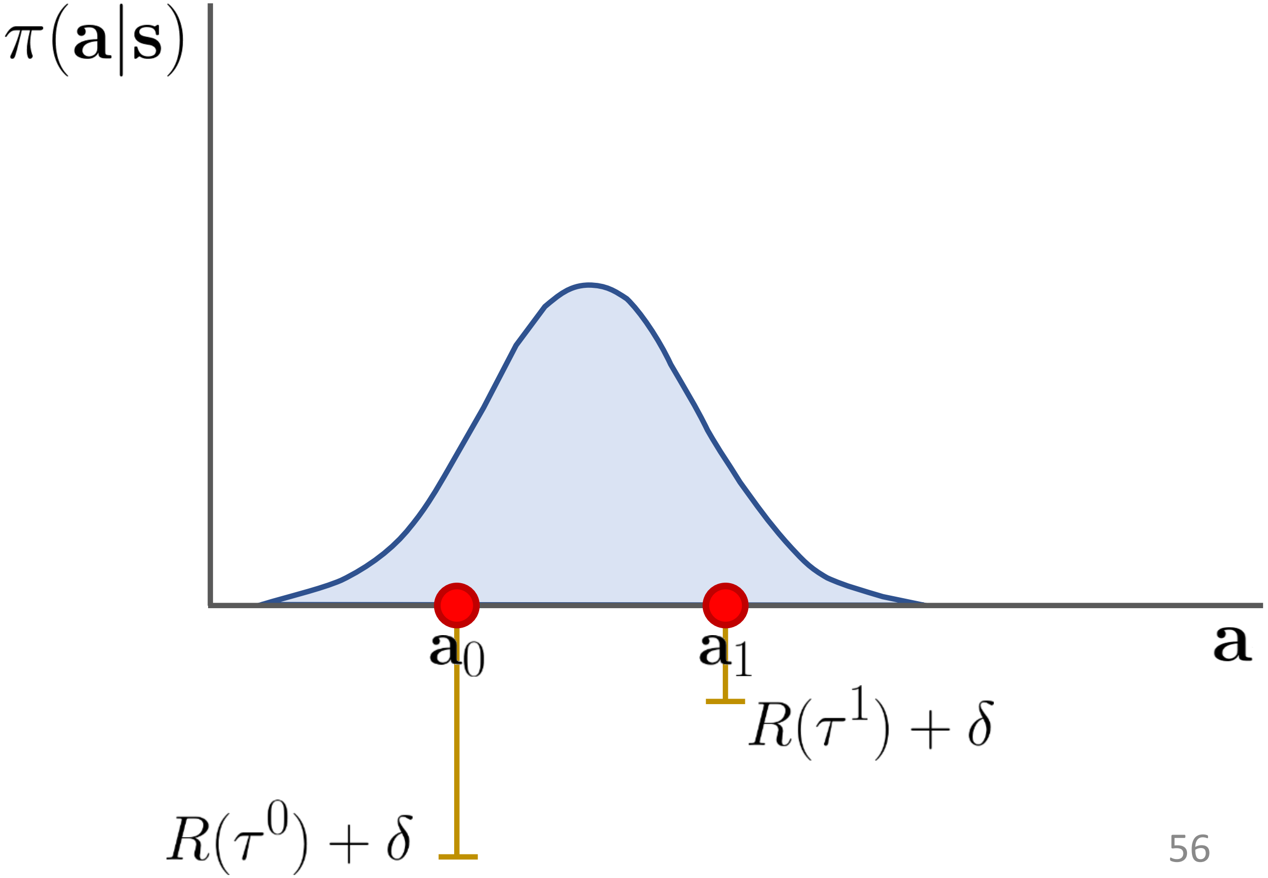
Problems

$$\nabla_\pi J(\pi) = \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[R(\tau) \sum_{t=0}^{T-1} \nabla_\pi \log \pi(\mathbf{a}_t|\mathbf{s}_t) \right]$$



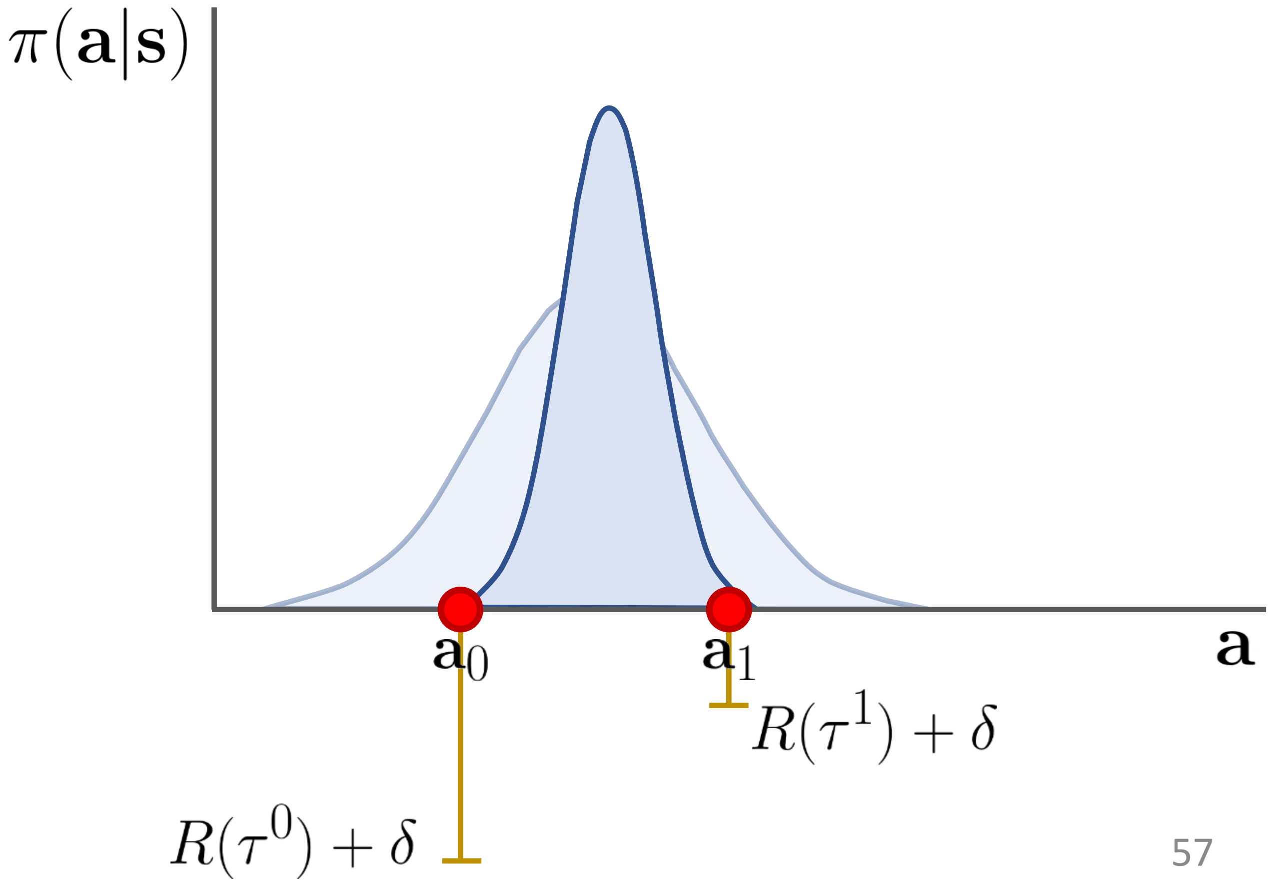
Problems

$$\nabla_\pi J(\pi) = \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[R(\tau) \sum_{t=0}^{T-1} \nabla_\pi \log \pi(\mathbf{a}_t|\mathbf{s}_t) \right]$$



Problems

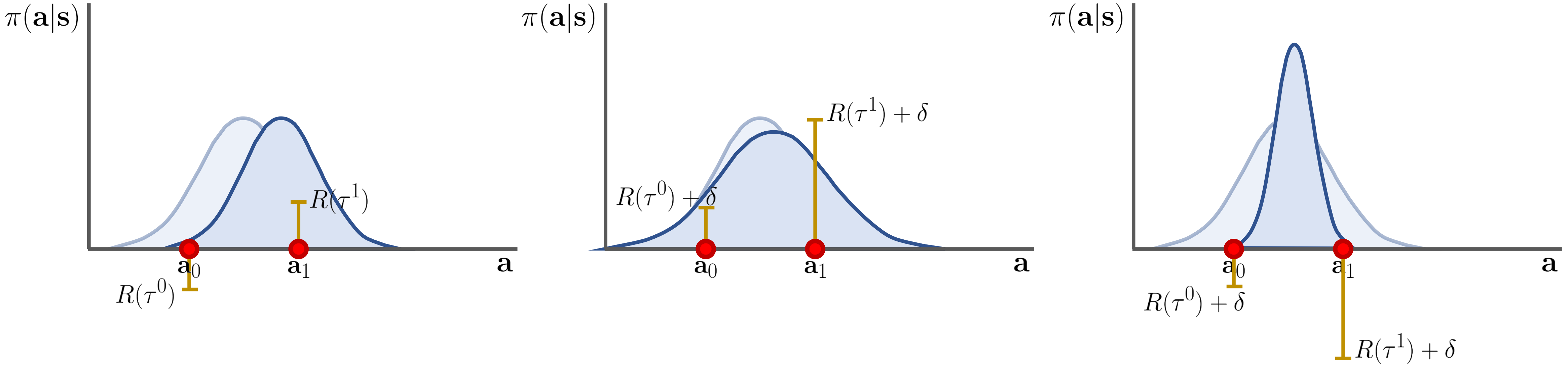
$$\nabla_\pi J(\pi) = \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[R(\tau) \sum_{t=0}^{T-1} \nabla_\pi \log \pi(\mathbf{a}_t|\mathbf{s}_t) \right]$$



Problems

$$\nabla_\pi J(\pi) = \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[R(\tau) \sum_{t=0}^{T-1} \nabla_\pi \log \pi(\mathbf{a}_t|\mathbf{s}_t) \right]$$

Problem: Not invariant to reward translations



Reward Translation

- Optimal policy is invariant to reward translation
- Gradient estimator is *not* invariant to reward translation

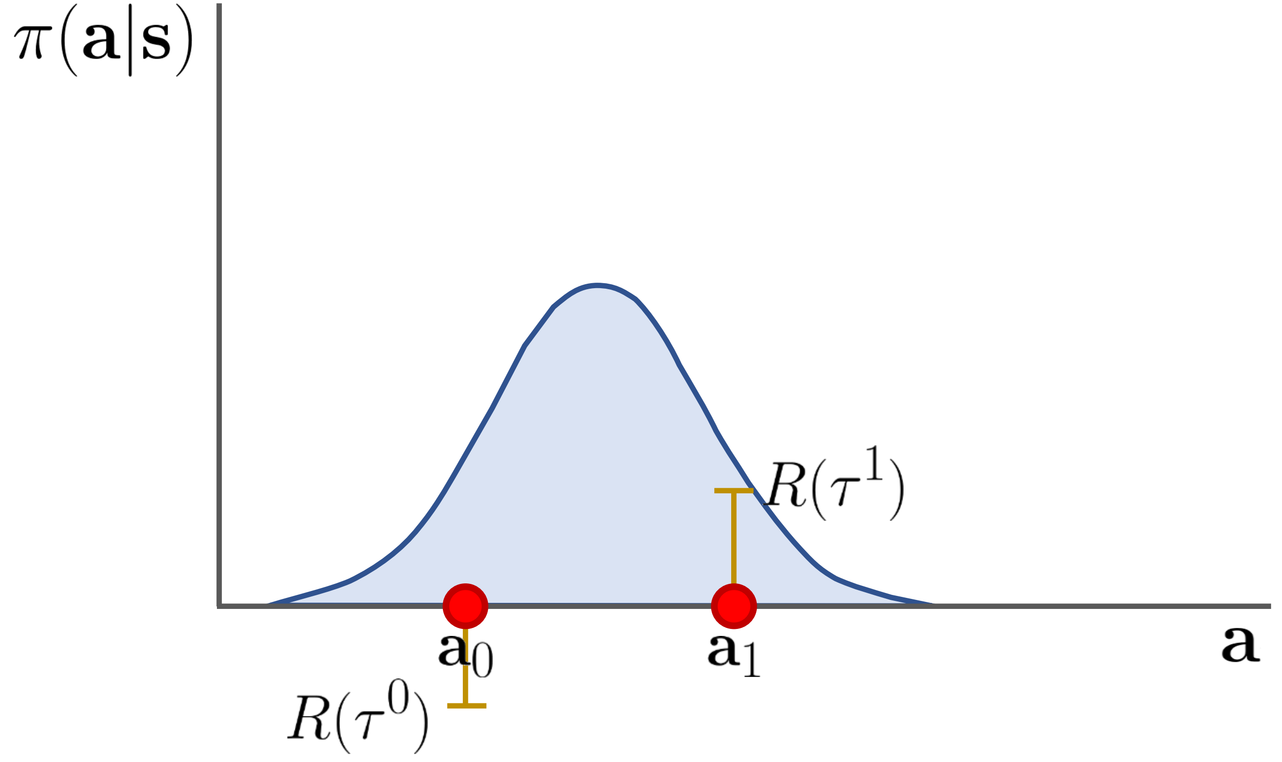
- Problem: Variance
  - Monte-Carlo estimate with finite samples
  - Goes away in expectation with infinite samples

Variance Reduction

- Baselines
- Causality
- Bootstrapping

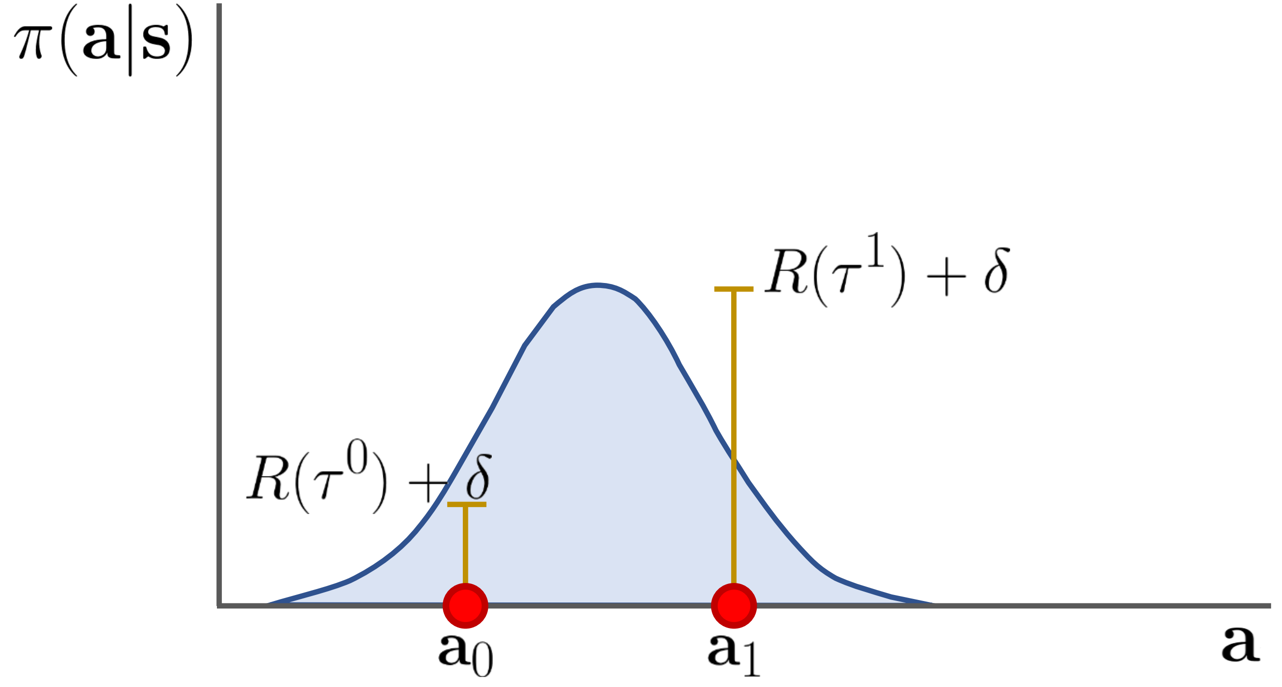
Variance Reduction: Baseline

$$\nabla_\pi J(\pi) = \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[R(\tau) \sum_{t=0}^{T-1} \nabla_\pi \log \pi(\mathbf{a}_t|\mathbf{s}_t) \right]$$



Variance Reduction: Baseline

$$\nabla_\pi J(\pi) = \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[R(\tau) \sum_{t=0}^{T-1} \nabla_\pi \log \pi(\mathbf{a}_t|\mathbf{s}_t) \right]$$

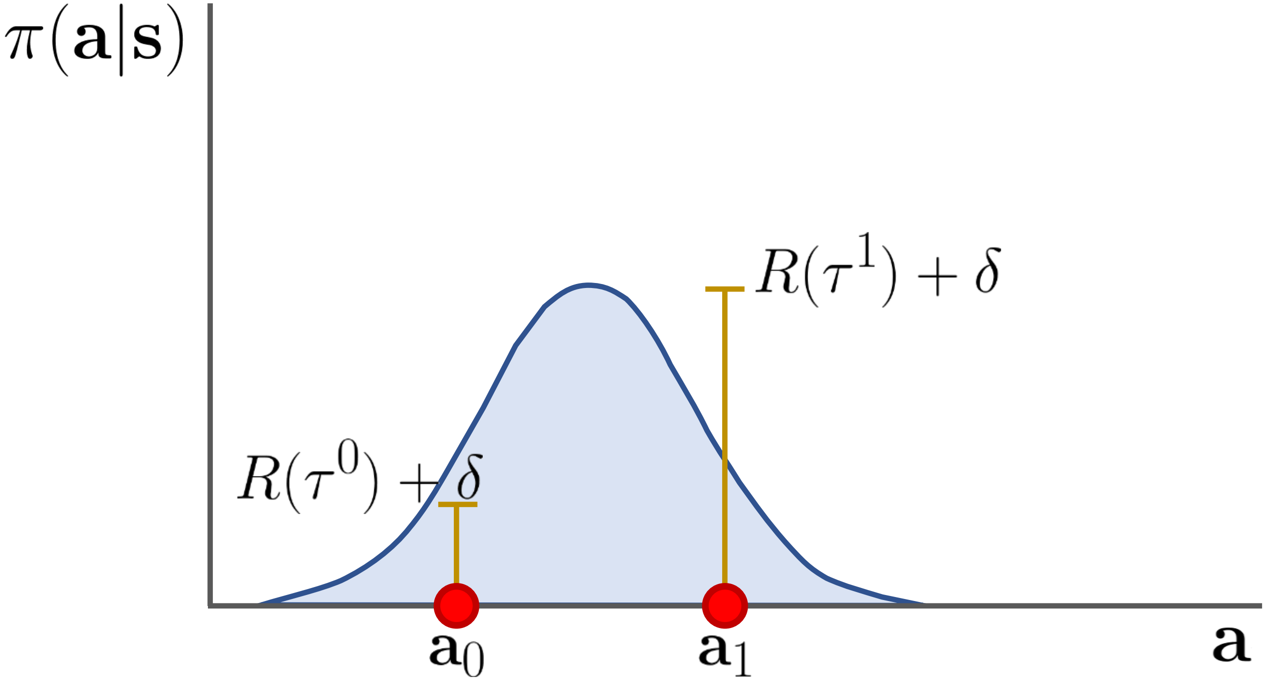


Variance Reduction: Baseline

$$\nabla_\pi J(\pi) = \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[(R(\tau) - b) \sum_{t=0}^{T-1} \nabla_\pi \log \pi(\mathbf{a}_t|\mathbf{s}_t) \right]$$

baseline

e.g. $b = \delta$



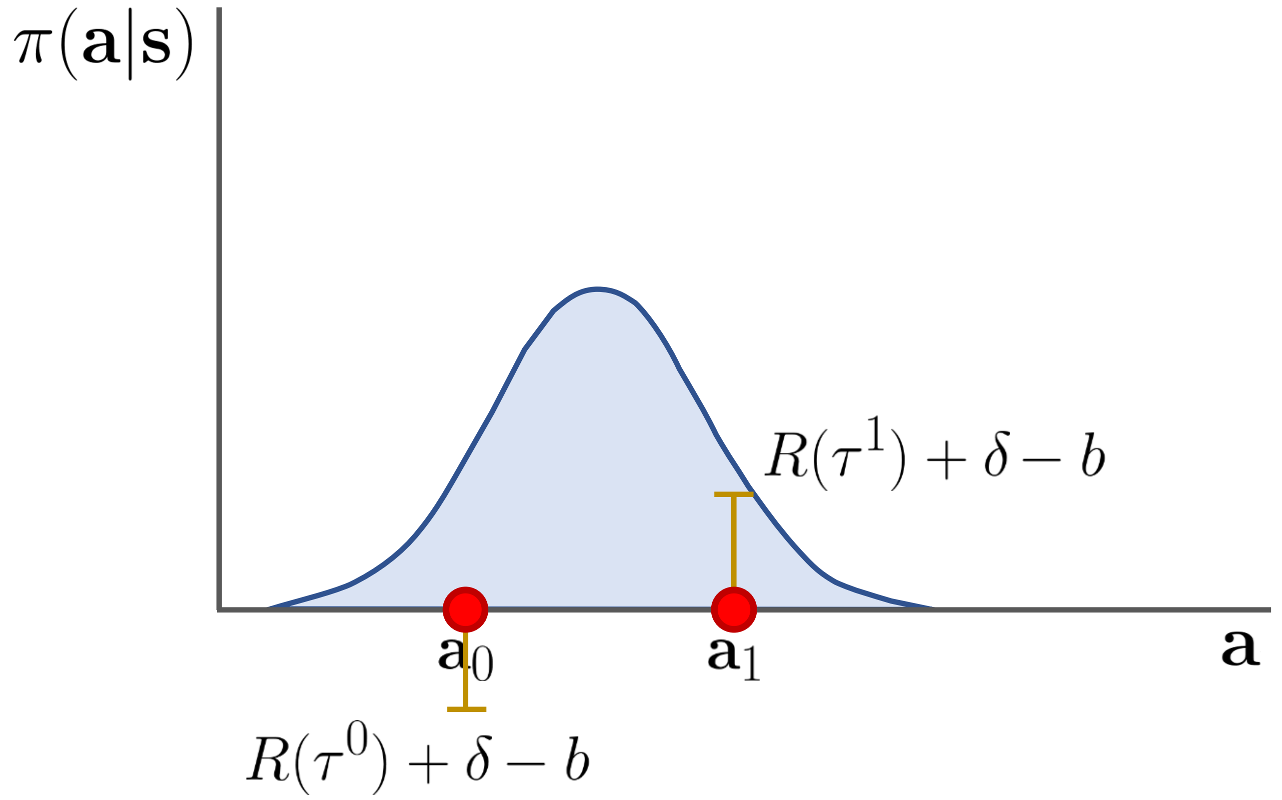
Variance Reduction: Baseline

$$\nabla_\pi J(\pi) = \mathbb{E}_{\tau \sim p(\tau|\pi)}\left[(R(\tau) - b)\sum_{t=0}^{T-1} \nabla_\pi \log \pi(\mathbf{a}_t|\mathbf{s}_t)\right]$$

baseline

e.g. $b = \delta$

- Baseline reduces variance
- Is this allowed?
- What is the optimal baseline?



Variance Reduction: Baseline

- How does the baseline effect the gradient?

$$R(\tau) \Longrightarrow \hat{R}(\tau) = R(\tau) - b$$

$$\nabla_\pi \hat{J}(\pi) = \nabla_\pi \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[\hat{R}(\tau)\right] = \nabla_\pi \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[R(\tau) - b\right]$$

$$= \nabla_\pi \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[R(\tau)\right] - \nabla_\pi \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[b\right]$$

$$= \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[R(\tau) \underbrace{\nabla_\pi \log p(\tau|\pi)}_{\text{score function}}\right] - \nabla_\pi \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[b\right]$$

Variance Reduction: Baseline

- How does the baseline effect the gradient?

$$R(\tau) \Longrightarrow \hat{R}(\tau) = R(\tau) - b$$

$$\nabla_\pi \hat{J}(\pi) = \nabla_\pi \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[\hat{R}(\tau)\right] = \nabla_\pi \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[R(\tau) - b\right]$$

$$= \nabla_\pi \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[R(\tau)\right] - \nabla_\pi \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[b\right]$$

$$= \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[R(\tau) \nabla_\pi \log p(\tau|\pi)\right] - \nabla_\pi \underbrace{\mathbb{E}_{\tau \sim p(\tau|\pi)} \left[b\right]}_{= b}$$

Variance Reduction: Baseline

- How does the baseline effect the gradient?

$$R(\tau) \Longrightarrow \hat{R}(\tau) = R(\tau) - b$$

$$\nabla_\pi \hat{J}(\pi) = \nabla_\pi \mathbb{E}_{\tau\sim p(\tau|\pi)}\left[\hat{R}(\tau)\right] = \nabla_\pi \mathbb{E}_{\tau\sim p(\tau|\pi)}\left[R(\tau) - b\right]$$

$$= \nabla_\pi \mathbb{E}_{\tau\sim p(\tau|\pi)}\left[R(\tau)\right] - \nabla_\pi \mathbb{E}_{\tau\sim p(\tau|\pi)}\left[b\right]$$

$$= \mathbb{E}_{\tau\sim p(\tau|\pi)}\left[R(\tau)\nabla_\pi \log p(\tau|\pi)\right] - \underline{\nabla_\pi \mathbb{E}_{\tau\sim p(\tau|\pi)}\left[b\right]}$$

$$\nabla_\pi b = 0$$

Variance Reduction: Baseline

- How does the baseline effect the gradient?

$$R(\tau) \Longrightarrow \hat{R}(\tau) = R(\tau) - b$$

$$\nabla_\pi \hat{J}(\pi) = \nabla_\pi \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[\hat{R}(\tau)\right] = \nabla_\pi \mathbb{E}_{\tau \sim p(\tau|\pi)} [R(\tau) - b]$$

$$= \nabla_\pi \mathbb{E}_{\tau \sim p(\tau|\pi)} [R(\tau)] - \nabla_\pi \mathbb{E}_{\tau \sim p(\tau|\pi)} [b]$$

$$= \mathbb{E}_{\tau \sim p(\tau|\pi)} [R(\tau) \nabla_\pi \log p(\tau|\pi)] - \cancel{\nabla_\pi \mathbb{E}_{\tau \sim p(\tau|\pi)} [b]} \; 0$$

Variance Reduction: Baseline

- How does the baseline effect the gradient?

$$R(\tau) \Longrightarrow \hat{R}(\tau) = R(\tau) - b$$

$$\nabla_\pi \hat{J}(\pi) = \nabla_\pi \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[\hat{R}(\tau)\right] = \nabla_\pi \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[R(\tau) - b\right]$$

$$= \nabla_\pi \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[R(\tau)\right] - \nabla_\pi \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[b\right]$$

$$= \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[R(\tau) \nabla_\pi \log p(\tau|\pi)\right] - \nabla_\pi \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[b\right]$$

$$= \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[R(\tau) \nabla_\pi \log p(\tau|\pi)\right]$$

$$\boxed{= \nabla_\pi J(\pi)}$$

Variance Reduction: Baseline

- How does the baseline effect the gradient?

$$R(\tau) \Longrightarrow \hat{R}(\tau) = R(\tau) - b$$

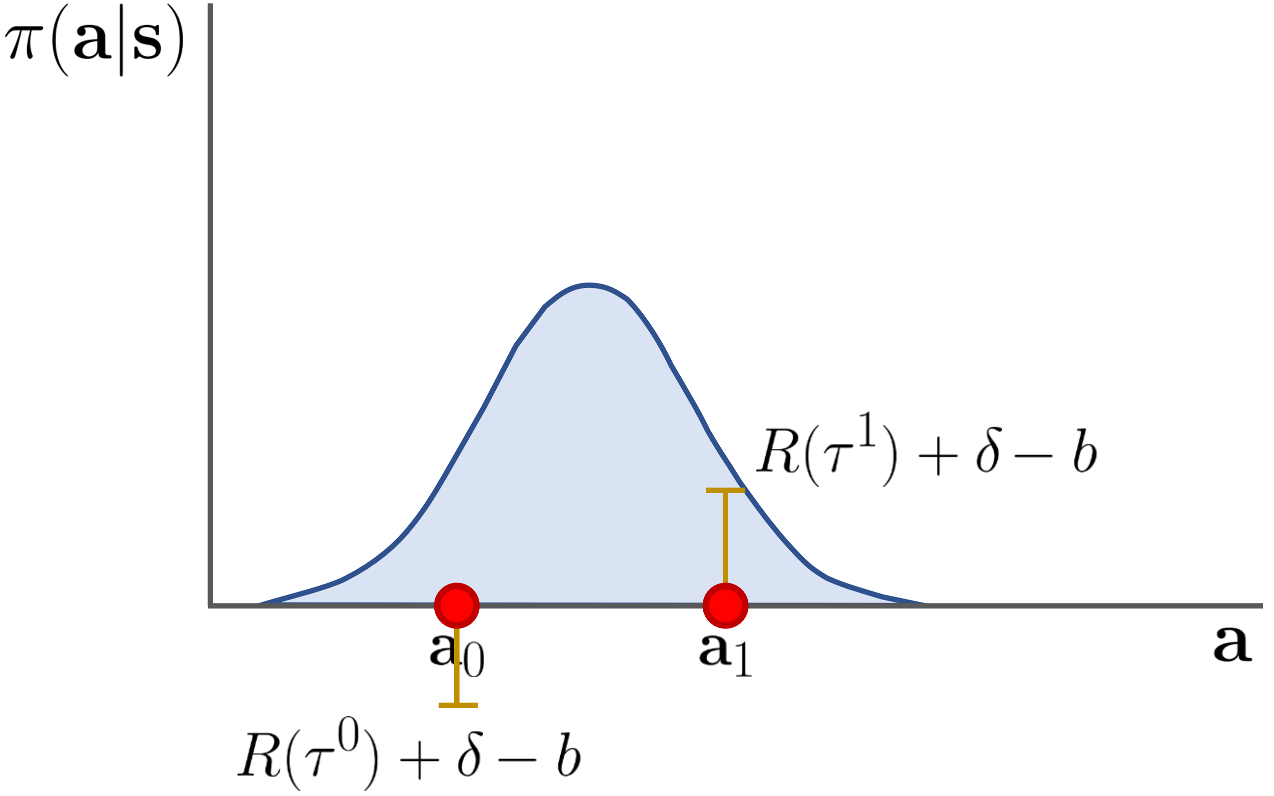
- Baseline does not change the gradient!
- Reduces variance without introducing bias
- Any baseline that is *independent* of the actions will preserve policy gradient

Variance Reduction: Baseline

$$\nabla_\pi J(\pi) = \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[(R(\tau) - b) \sum_{t=0}^{T-1} \nabla_\pi \log \pi(\mathbf{a}_t|\mathbf{s}_t) \right]$$

baseline

e.g. $b = \delta$



- Baseline reduces variance
- Is this allowed?
- What is the optimal baseline?

Optimal Baseline

- Minimize variance of gradient estimator

$$\mathrm{Var}\left[x\right] = \mathbb{E}[x^2] - \left(\mathbb{E}\left[x\right]\right)^2$$

$$\mathrm{Var}\left[\nabla_\pi J(\pi)\right] = \mathrm{Var}\left[\left(R(\tau) - b\right)\nabla_\pi \mathrm{log} p(\tau|\pi)\right]$$

$$= \mathbb{E}_{\tau\sim p(\tau|\pi)}\left[\left(\left(R(\tau) - b\right)\nabla_\pi \log\, p(\tau|\pi)\right)^2\right] - \Big(\underbrace{\mathbb{E}_{\tau\sim p(\tau|\pi)}\left[\left(R(\tau) - b\right)\nabla_\pi \log\, p(\tau|\pi)\right]}_{\substack{= \mathbb{E}_{\tau\sim p(\tau|\pi)}\left[R(\tau)\nabla_\pi \mathrm{log} p(\tau|\pi)\right] \\ = \nabla_\pi J(\pi)}}\Big)^2$$

independent of baseline

Optimal Baseline

- Minimize variance of gradient estimator

$$\mathrm{Var}\left[x\right] = \mathbb{E}[x^2] - \left(\mathbb{E}\left[x\right]\right)^2$$

$$\begin{aligned}\mathrm{Var}\left[\nabla_\pi J(\pi)\right] &= \mathrm{Var}\left[\left(R(\tau) - b\right)\nabla_\pi \mathrm{log} p(\tau|\pi)\right] \\ &= \underline{\mathbb{E}_{\tau\sim p(\tau|\pi)}\left[\left(\left(R(\tau) - b\right)\nabla_\pi \log\, p(\tau|\pi)\right)^2\right]} - \left(\mathbb{E}_{\tau\sim p(\tau|\pi)}\left[\left(R(\tau) - b\right)\nabla_\pi \log\, p(\tau|\pi)\right]\right)^2\end{aligned}$$

Optimal Baseline

$$\frac{d\mathrm{Var}}{db} = \frac{d}{db}\mathbb{E}_{\tau\sim p(\tau|\pi)}\left[(R(\tau)-b)^2\left(\nabla_\pi \log\, p(\tau|\pi)\right)^2\right] = 0$$

$$= \mathbb{E}_{\tau\sim p(\tau|\pi)}\left[-2\left(R(\tau)-b\right)\left(\nabla_\pi \log\, p(\tau|\pi)\right)^2\right]$$

$$= -2\,\mathbb{E}_{\tau\sim p(\tau|\pi)}\left[R(\tau)\left(\nabla_\pi \log\, p(\tau|\pi)\right)^2\right] + 2b\,\mathbb{E}_{\tau\sim p(\tau|\pi)}\left[\left(\nabla_\pi \log\, p(\tau|\pi)\right)^2\right]$$

$$2b\,\mathbb{E}_{\tau\sim p(\tau|\pi)}\left[\left(\nabla_\pi \log\, p(\tau|\pi)\right)^2\right] = 2\,\mathbb{E}_{\tau\sim p(\tau|\pi)}\left[R(\tau)\left(\nabla_\pi \log\, p(\tau|\pi)\right)^2\right]$$

$$b = \frac{\mathbb{E}_{\tau\sim p(\tau|\pi)}\left[R(\tau)\left(\nabla_\pi \log\, p(\tau|\pi)\right)^2\right]}{\mathbb{E}_{\tau\sim p(\tau|\pi)}\left[\left(\nabla_\pi \log\, p(\tau|\pi)\right)^2\right]} \xRightarrow{w(\tau) = \left(\nabla_\pi \log\, p(\tau|\pi)\right)^2} \boxed{b = \frac{\mathbb{E}_{\tau\sim p(\tau|\pi)}\left[R(\tau)w(\tau)\right]}{\mathbb{E}_{\tau\sim p(\tau|\pi)}\left[w(\tau)\right]}}$$

Optimal Baseline

$$R(\tau) \Longrightarrow \hat{R}(\tau) = R(\tau) - b$$

- Optimal baseline:

$$b = \frac{\mathbb{E}_{\tau \sim p(\tau|\pi)}\left[R(\tau)w(\tau)\right]}{\mathbb{E}_{\tau \sim p(\tau|\pi)}\left[w(\tau)\right]} \quad \text{where} \quad w(\tau) = \left(\nabla_\pi \log\, p(\tau|\pi)\right)^2$$

- In practice:

$$b = \mathbb{E}_{\tau \sim p(\tau|\pi)}\left[R(\tau)\right]$$

easier to estimate

Optimal Baseline

$$\nabla_\pi J(\pi) = \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[(R(\tau) - b) \sum_{t=0}^{T-1} \nabla_\pi \log \pi(\mathbf{a}_t | \mathbf{s}_t) \right]$$

where

$$b = \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[R(\tau) \right]$$

- Interpretation:
  - Increase likelihood of trajectories that do *better* than average
  - Decrease likelihood of trajectories that do *worse* than average

Optimal Baseline

$$\nabla_\pi J(\pi) = \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[\underbrace{(R(\tau) - b)}_{>0} \sum_{t=0}^{T-1} \underbrace{\nabla_\pi \log \pi(\mathbf{a}_t|\mathbf{s}_t)}_{\text{increase likelihood}} \right]$$

where

$$b = \mathbb{E}_{\tau \sim p(\tau|\pi)} [R(\tau)]$$

- Interpretation:
  - Increase likelihood of trajectories that do *better* than average
  - Decrease likelihood of trajectories that do *worse* than average

Optimal Baseline

$$\nabla_\pi J(\pi) = \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[\underbrace{(R(\tau) - b)}_{< 0} \sum_{t=0}^{T-1} \underbrace{\nabla_\pi \log \pi(\mathbf{a}_t|\mathbf{s}_t)}_{\text{decrease likelihood}} \right]$$

where

$$b = \mathbb{E}_{\tau \sim p(\tau|\pi)} [R(\tau)]$$

- Interpretation:
  - Increase likelihood of trajectories that do *better* than average
  - Decrease likelihood of trajectories that do *worse* than average

Policy Gradient

ALGORITHM: Policy Gradient

1: $\theta \leftarrow$ initialize policy parameters

2: **while** not done **do**

3: Sample trajectories $\{\tau^i\}$ from policy $\pi_\theta(\mathbf{a}|\mathbf{s})$

4: Estimate baseline $b = \frac{1}{N} R\left(\tau^i\right)$

5: Estimate policy gradient

$\nabla_\theta J(\pi_\theta) \approx \frac{1}{N} \sum_i \left(R(\tau^i) - b\right) \sum_t \nabla_\theta \log \pi_\theta(\mathbf{a}_t^i|\mathbf{s}_t^i)$

6: Update policy $\theta \leftarrow \theta + \alpha \nabla_\theta J(\pi_\theta)$

7: **end while**

8: return policy π_θ

Policy Gradient Methods for Reinforcement Learning with Function Approximation [Sutton et al. 1999]

Policy Gradient

ALGORITHM: Policy Gradient

1: $\theta \leftarrow$ initialize policy parameters

2: **while** not done **do**
3: Sample trajectories $\{\tau^i\}$ from policy $\pi_\theta(\mathbf{a}|\mathbf{s})$
4: Estimate baseline $b = \frac{1}{N} R(\tau^i)$
5: Estimate policy gradient
$\nabla_\theta J(\pi_\theta) \approx \frac{1}{N} \sum_i (R(\tau^i) - b) \sum_t \nabla_\theta \log \pi_\theta(\mathbf{a}_t^i|\mathbf{s}_t^i)$
6: Update policy $\theta \leftarrow \theta + \alpha \nabla_\theta J(\pi_\theta)$
7: **end while**

8: return policy π_θ

Policy Gradient Methods for Reinforcement Learning with Function Approximation [Sutton et al. 1999]

Variance Reduction

- Baselines
- Causality
- Bootstrapping

Variance Reduction: Causality

$$\nabla_\pi J(\pi) = \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[\underline{(R(\tau) - b)} \sum_{t=0}^{T-1} \nabla_\pi \log \pi(\mathbf{a}_t|\mathbf{s}_t) \right]$$

$$= \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[\underline{\left(\sum_{t'=0}^{T-1} \gamma^{t'} r_{t'} - b \right)} \sum_{t=0}^{T-1} \nabla_\pi \log \pi(\mathbf{a}_t|\mathbf{s}_t) \right]$$

rewards across all timesteps

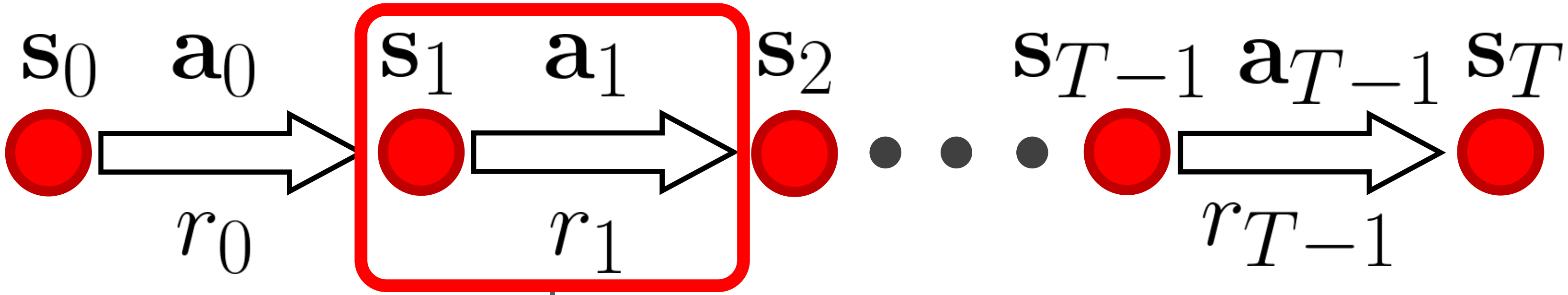
Variance Reduction: Causality

- Gradient at single timestep t

$$\underbrace{\left(\sum_{t'=0}^{T-1} \gamma^{t'} r_{t'} - b\right)}_{\text{sum rewards across all timesteps}} \nabla_\pi \log \pi(\mathbf{a}_t|\mathbf{s}_t)$$

- Current action *does not* affect past rewards
- $r_{t'}$ is independent of $\mathbf{a}_t$ for all $t' < t$

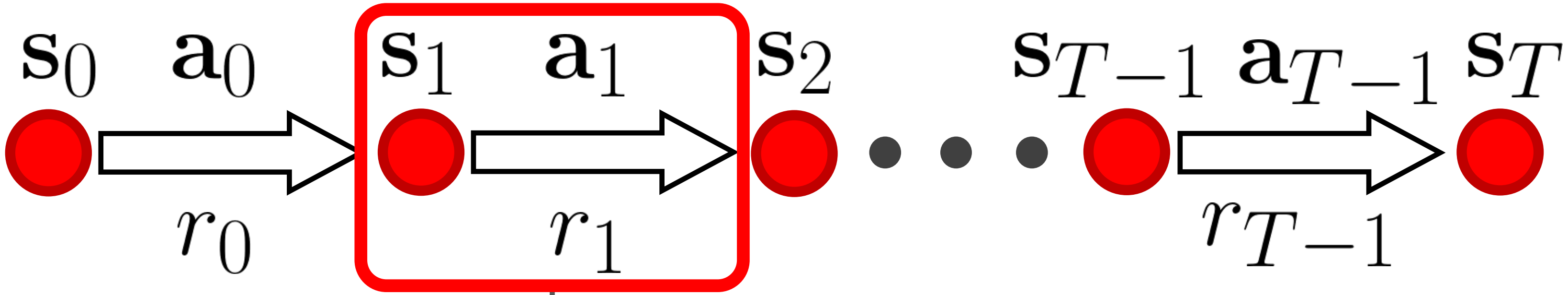
Variance Reduction: Causality



$$\left(\underline{r_0} + r_1 + r_2 + \ldots + r_{T-1}\right) \nabla_\pi \log \pi(\mathbf{a}_1 | \mathbf{s}_1)$$

Not affected by $\mathbf{a}_1$

Variance Reduction: Causality



$$(r_1 + r_2 + \ldots + r_{T-1})\, \nabla_\pi \log \pi(\mathbf{a}_1 | \mathbf{s}_1)$$

Generally:

$$(r_t + r_{t+1} + \ldots + r_{T-1})\, \nabla_\pi \log \pi(\mathbf{a}_t | \mathbf{s}_t)$$

Variance Reduction: Causality

$$\nabla_\pi J(\pi) = \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[\sum_{t=0}^{T-1} \nabla_\pi \log \pi(\mathbf{a}_t|\mathbf{s}_t) \left(\sum_{\underline{t'=0}}^{T-1} \gamma^{t'} r_{t'} - b \right) \right]$$

$$\Downarrow$$

$$\nabla_\pi J(\pi) = \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[\sum_{t=0}^{T-1} \nabla_\pi \log \pi(\mathbf{a}_t|\mathbf{s}_t) \left(\sum_{\underline{t'=t}}^{T-1} \gamma^{t'} r_{t'} - b \right) \right]$$

"reward-to-go"

fewer reward terms → lower variance

Proof: https://spinningup.openai.com/en/latest/spinningup/extra_pg_proof1.html

Variance Reduction: Causality

- Trajectory-based estimator:

$$\nabla_\pi J(\pi) = \underline{\mathbb{E}_{\tau \sim p(\tau|\pi)}} \left[\sum_{t=0}^{T-1} \nabla_\pi \log \pi(\mathbf{a}_t|\mathbf{s}_t) \underline{\left(\sum_{t'=0}^{T-1} \gamma^{t'} r_{t'} - b \right)} \right]$$

- Reward-to-Go estimator:

$$\nabla_\pi J(\pi) = \mathbb{E}_{\mathbf{s} \sim d_\pi(\mathbf{s})} \mathbb{E}_{\mathbf{a} \sim \pi(\mathbf{a}|\mathbf{s})} \mathbb{E}_{\tau \sim p(\tau|\pi, \mathbf{s}_0=\mathbf{s}, \mathbf{a}_0=\mathbf{a})} \left[\nabla_\pi \log \pi(\mathbf{a}|\mathbf{s}) \left(\sum_{t'=0}^{T-1} \gamma^{t'} r_{t'} - b \right) \right]$$

Variance Reduction: Causality

- Trajectory-based estimator:

$$\nabla_\pi J(\pi) = \mathbb{E}_{\tau\sim p(\tau|\pi)}\left[\sum_{t=0}^{T-1} \nabla_\pi \log\pi(\mathbf{a}_t|\mathbf{s}_t)\left(\sum_{t'=0}^{T-1}\gamma^{t'} r_{t'} - b\right)\right]$$

- Reward-to-Go estimator:

$$\nabla_\pi J(\pi) = \underline{\mathbb{E}_{\mathbf{s}\sim d_\pi(\mathbf{s})}}\mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\underline{\mathbb{E}_{\tau\sim p(\tau|\pi,\mathbf{s}_0=\mathbf{s},\mathbf{a}_0=\mathbf{a})}}\left[\nabla_\pi \log\pi(\mathbf{a}|\mathbf{s})\left(\sum_{t'=0}^{T-1}\gamma^{t'} r_{t'} - b\right)\right]$$

treat every state as start of
a new trajectory

Variance Reduction: Causality

- Trajectory-based estimator:

$$\nabla_\pi J(\pi) = \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[\sum_{t=0}^{T-1} \nabla_\pi \log \pi(\mathbf{a}_t|\mathbf{s}_t) \left(\sum_{t'=0}^{T-1} \gamma^{t'} r_{t'} - b \right) \right]$$

- Reward-to-Go estimator:

$$\nabla_\pi J(\pi) = \underbrace{\mathbb{E}_{\mathbf{s} \sim d_\pi(\mathbf{s})}}_{\text{“discounted” state distribution of the policy } \pi} \mathbb{E}_{\mathbf{a} \sim \pi(\mathbf{a}|\mathbf{s})} \mathbb{E}_{\tau \sim p(\tau|\pi, \mathbf{s}_0 = \mathbf{s}, \mathbf{a}_0 = \mathbf{a})} \left[\nabla_\pi \log \pi(\mathbf{a}|\mathbf{s}) \underbrace{\left(\sum_{t'=0}^{T-1} \gamma^{t'} r_{t'} - b \right)}_{\text{sum future rewards}} \right]$$

Discounted State Distribution

$$\nabla_\pi J(\pi) = \mathbb{E}_{\mathbf{s}\sim d_\pi(\mathbf{s})}\mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\mathbb{E}_{\tau\sim p(\tau|\pi,\mathbf{s}_0=\mathbf{s},\mathbf{a}_0=\mathbf{a})}\left[\nabla_\pi \log\pi(\mathbf{a}|\mathbf{s})\left(\sum_{t'=0}^{T-1}\gamma^{t'}r_{t'} - b\right)\right]$$

"discounted" state distribution

$$d_\pi(\mathbf{s}) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t p(\mathbf{s}_t = \mathbf{s}|\pi)$$

Discounted State Distribution

$$\nabla_\pi J(\pi) = \underline{\mathbb{E}_{\mathbf{s}\sim d_\pi(\mathbf{s})}}\mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\mathbb{E}_{\tau\sim p(\tau|\pi,\mathbf{s}_0=\mathbf{s},\mathbf{a}_0=\mathbf{a})}\left[\nabla_\pi \log\pi(\mathbf{a}|\mathbf{s})\left(\sum_{t'=0}^{T-1}\gamma^{t'}r_{t'} - b\right)\right]$$

"discounted" state distribution

$$d_\pi(\mathbf{s}) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t \underline{p(\mathbf{s}_t = \mathbf{s}|\pi)}$$

probability of being in $\mathbf{s}$ after following π for t timesteps

Discounted State Distribution

$$\nabla_\pi J(\pi) = \underbrace{\mathbb{E}_{\mathbf{s}\sim d_\pi(\mathbf{s})}}_{\text{“discounted” state distribution}} \mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})} \mathbb{E}_{\tau\sim p(\tau|\pi,\mathbf{s}_0=\mathbf{s},\mathbf{a}_0=\mathbf{a})} \left[\nabla_\pi \log\pi(\mathbf{a}|\mathbf{s}) \left(\sum_{t'=0}^{T-1} \gamma^{t'} r_{t'} - b \right) \right]$$

$$d_\pi(\mathbf{s}) = (1-\gamma) \sum_{t=0}^{\infty} \underline{\gamma^t} p(\mathbf{s}_t = \mathbf{s}|\pi)$$

Discounted State Distribution

$$\nabla_\pi J(\pi) = \underline{\mathbb{E}_{\mathbf{s}\sim d_\pi(\mathbf{s})}}\mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\mathbb{E}_{\tau\sim p(\tau|\pi,\mathbf{s}_0=\mathbf{s},\mathbf{a}_0=\mathbf{a})}\left[\nabla_\pi \log\pi(\mathbf{a}|\mathbf{s})\left(\sum_{t'=0}^{T-1}\gamma^{t'}r_{t'} - b\right)\right]$$

"discounted" state distribution

$$d_\pi(\mathbf{s}) = \underline{(1-\gamma)}\sum_{t=0}^{\infty}\gamma^t p(\mathbf{s}_t = \mathbf{s}|\pi)$$

Discounted State Distribution

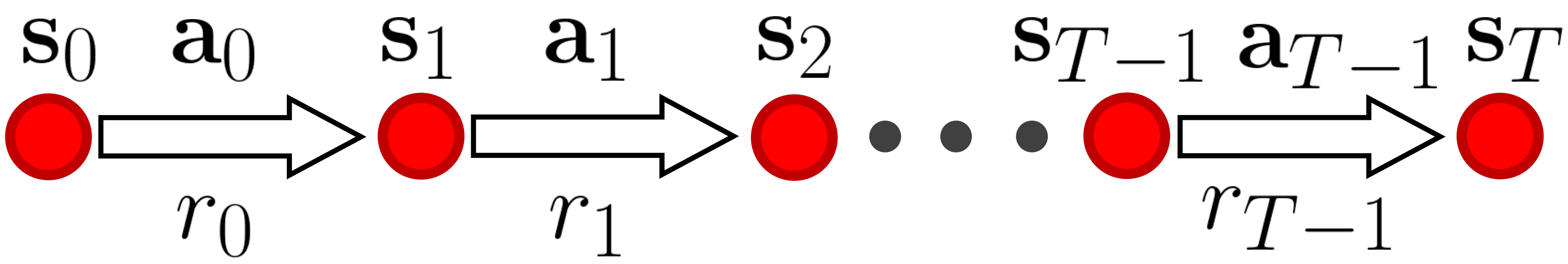
$$\nabla_\pi J(\pi) = \mathbb{E}_{\mathbf{s}\sim d_\pi(\mathbf{s})} \mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})} \mathbb{E}_{\tau\sim p(\tau|\pi,\mathbf{s}_0=\mathbf{s},\mathbf{a}_0=\mathbf{a})} \left[\nabla_\pi \log \pi(\mathbf{a}|\mathbf{s}) \left(\sum_{t'=0}^{T-1} \gamma^{t'} r_{t'} - b \right) \right]$$

"discounted" state distribution

$$d_\pi(\mathbf{s}) = (1-\gamma) \sum_{t=0}^{\infty} \gamma^t p(\mathbf{s}_t = \mathbf{s}|\pi) \Longrightarrow p(\mathbf{s}|\pi)$$

In practice, just use the marginal state distribution instead

Reward-to-Go Gradient Estimator



$$\nabla_0 = \left(r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots + \gamma^{T-1} r_{T-1}\right) \nabla_\pi \log \pi(\mathbf{a}_0|\mathbf{s}_0)$$

$$\nabla_1 = \left(r_1 + \gamma r_2 + \gamma^2 r_3 + \ldots + \gamma^{T-2} r_{T-1}\right) \nabla_\pi \log \pi(\mathbf{a}_1|\mathbf{s}_1)$$

$$\vdots$$

$$\nabla_{T-1} = (r_{T-1}) \nabla_\pi \log \pi(\mathbf{a}_{T-1}|\mathbf{s}_{T-1})$$

average grads $\approx \nabla_\pi J(\pi)$

State-Based Baseline

- Reward-to-Go estimator:

$$\nabla_\pi J(\pi) = \mathbb{E}_{\mathbf{s}\sim d_\pi(\mathbf{s})}\mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\mathbb{E}_{\tau\sim p(\tau|\pi,\mathbf{s}_0=\mathbf{s},\mathbf{a}_0=\mathbf{a})}\left[\nabla_\pi \log\pi(\mathbf{a}|\mathbf{s})\left(\sum_{t'=0}^{T-1}\gamma^{t'} r_{t'} - \underline{b}\right)\right]$$

Can use a better baseline for
even lower variance

State-Based Baseline

- Reward-to-Go estimator:

$$\nabla_\pi J(\pi) = \mathbb{E}_{\mathbf{s}\sim d_\pi(\mathbf{s})}\mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\mathbb{E}_{\tau\sim p(\tau|\pi,\mathbf{s}_0=\mathbf{s},\mathbf{a}_0=\mathbf{a})}\left[\nabla_\pi \log\pi(\mathbf{a}|\mathbf{s})\left(\sum_{t'=0}^{T-1}\gamma^{t'}r_{t'} - b\right)\right]$$

$$\nabla_\pi J(\pi) = \mathbb{E}_{\mathbf{s}\sim d_\pi(\mathbf{s})}\mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\mathbb{E}_{\tau\sim p(\tau|\pi,\mathbf{s}_0=\mathbf{s},\mathbf{a}_0=\mathbf{a})}\left[\nabla_\pi \log\pi(\mathbf{a}|\mathbf{s})\left(\sum_{t'=0}^{T-1}\gamma^{t'}r_{t'} - V^\pi(\mathbf{s})\right)\right]$$

Value function baseline

State-Based Baseline

- Reward-to-Go estimator:

$$\nabla_\pi J(\pi) = \mathbb{E}_{\mathbf{s}\sim d_\pi(\mathbf{s})}\mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\mathbb{E}_{\tau\sim p(\tau|\pi,\mathbf{s}_0=\mathbf{s},\mathbf{a}_0=\mathbf{a})}\left[\nabla_\pi \log\pi(\mathbf{a}|\mathbf{s})\left(\sum_{t'=0}^{T-1}\gamma^{t'} r_{t'} - b\right)\right]$$

$$\nabla_\pi J(\pi) = \mathbb{E}_{\mathbf{s}\sim d_\pi(\mathbf{s})}\mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\mathbb{E}_{\tau\sim p(\tau|\pi,\mathbf{s}_0=\mathbf{s},\mathbf{a}_0=\mathbf{a})}\left[\nabla_\pi \log\pi(\mathbf{a}|\mathbf{s})\underbrace{\left(\sum_{t'=0}^{T-1}\gamma^{t'} r_{t'} - V^\pi(\mathbf{s})\right)}_{\text{“Advantage”}}\right]$$

- Advantage > 0: Action is better than average
- Advantage < 0: Action is worse than average

Value Function Baseline

$$\mathbb{E}_{\mathbf{s}\sim d_\pi(\mathbf{s})}\mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\mathbb{E}_{\tau\sim p(\tau|\pi,\mathbf{s}_0=\mathbf{s},\mathbf{a}_0=\mathbf{a})}\left[\nabla_\pi \log\pi(\mathbf{a}|\mathbf{s})\left(\sum_{t'=0}^{T-1}\gamma^{t'}r_{t'} - V^\pi(\mathbf{s})\right)\right]$$

$$= \mathbb{E}_{\mathbf{s}\sim d_\pi(\mathbf{s})}\mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\mathbb{E}_{\tau\sim p(\tau|\pi,\mathbf{s}_0=\mathbf{s},\mathbf{a}_0=\mathbf{a})}\left[\nabla_\pi \log\pi(\mathbf{a}|\mathbf{s})\sum_{t'=0}^{T-1}\gamma^{t'}r_{t'}\right]$$

$$\underline{-\mathbb{E}_{\mathbf{s}\sim d_\pi(\mathbf{s})}\mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\mathbb{E}_{\tau\sim \pi(\tau|\pi,\mathbf{s}_0=\mathbf{s},\mathbf{a}_0=\mathbf{a})}\left[\nabla_\pi \log\pi(\mathbf{a}|\mathbf{s})V^\pi(\mathbf{s})\right]}$$

Value Function Baseline

$$\mathbb{E}_{\mathbf{s}\sim d_\pi(\mathbf{s})}\mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\mathbb{E}_{\tau\sim p(\tau|\pi,\mathbf{s}_0=\mathbf{s},\mathbf{a}_0=\mathbf{a})}\left[\nabla_\pi \log\pi(\mathbf{a}|\mathbf{s})\left(\sum_{t'=0}^{T-1}\gamma^{t'}r_{t'} - V^\pi(\mathbf{s})\right)\right]$$

$$= \mathbb{E}_{\mathbf{s}\sim d_\pi(\mathbf{s})}\mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\mathbb{E}_{\tau\sim p(\tau|\pi,\mathbf{s}_0=\mathbf{s},\mathbf{a}_0=\mathbf{a})}\left[\nabla_\pi \log\pi(\mathbf{a}|\mathbf{s})\sum_{t'=0}^{T-1}\gamma^{t'}r_{t'}\right]$$

$$-\mathbb{E}_{\mathbf{s}\sim d_\pi(\mathbf{s})}\mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\mathbb{E}_{\tau\sim \pi(\tau|\pi,\mathbf{s}_0=\mathbf{s},\mathbf{a}_0=\mathbf{a})}\left[\nabla_\pi \log\pi(\mathbf{a}|\underline{\mathbf{s}})V^\pi(\underline{\mathbf{s}})\right]$$

Value Function Baseline

$$\mathbb{E}_{\mathbf{s}\sim d_\pi(\mathbf{s})}\mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\mathbb{E}_{\tau\sim p(\tau|\pi,\mathbf{s}_0=\mathbf{s},\mathbf{a}_0=\mathbf{a})}\left[\nabla_\pi \log\pi(\mathbf{a}|\mathbf{s})\left(\sum_{t'=0}^{T-1}\gamma^{t'}r_{t'} - V^\pi(\mathbf{s})\right)\right]$$

$$= \mathbb{E}_{\mathbf{s}\sim d_\pi(\mathbf{s})}\mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\mathbb{E}_{\tau\sim p(\tau|\pi,\mathbf{s}_0=\mathbf{s},\mathbf{a}_0=\mathbf{a})}\left[\nabla_\pi \log\pi(\mathbf{a}|\mathbf{s})\sum_{t'=0}^{T-1}\gamma^{t'}r_{t'}\right]$$

$$-\mathbb{E}_{\mathbf{s}\sim d_\pi(\mathbf{s})}\mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\cancel{\mathbb{E}_{\tau\sim\pi(\tau|\pi,\mathbf{s}_0=\mathbf{s},\mathbf{a}_0=\mathbf{a})}}\left[\nabla_\pi \log\pi(\mathbf{a}|\mathbf{s})\underline{V^\pi(\mathbf{s})}\right]$$

Value Function Baseline

$$\mathbb{E}_{\mathbf{s}\sim d_\pi(\mathbf{s})}\mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\mathbb{E}_{\tau\sim p(\tau|\pi,\mathbf{s}_0=\mathbf{s},\mathbf{a}_0=\mathbf{a})}\left[\nabla_\pi \log\pi(\mathbf{a}|\mathbf{s})\left(\sum_{t'=0}^{T-1}\gamma^{t'}r_{t'} - V^\pi(\mathbf{s})\right)\right]$$

$$= \mathbb{E}_{\mathbf{s}\sim d_\pi(\mathbf{s})}\mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\mathbb{E}_{\tau\sim p(\tau|\pi,\mathbf{s}_0=\mathbf{s},\mathbf{a}_0=\mathbf{a})}\left[\nabla_\pi \log\pi(\mathbf{a}|\mathbf{s})\sum_{t'=0}^{T-1}\gamma^{t'}r_{t'}\right]$$

$$-\mathbb{E}_{\mathbf{s}\sim d_\pi(\mathbf{s})}\mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\mathbb{E}_{\tau\sim \pi(\tau|\pi,\mathbf{s}_0=\mathbf{s},\mathbf{a}_0=\mathbf{a})}\left[\nabla_\pi \log\pi(\mathbf{a}|\mathbf{s})V^\pi(\mathbf{s})\right]$$

$$= \mathbb{E}_{\mathbf{s}\sim d_\pi(\mathbf{s})}\mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\mathbb{E}_{\tau\sim p(\tau|\pi,\mathbf{s}_0=\mathbf{s},\mathbf{a}_0=\mathbf{a})}\left[\nabla_\pi \log\pi(\mathbf{a}|\mathbf{s})\sum_{t'=0}^{T-1}\gamma^{t'}r_{t'}\right]$$

$$-\mathbb{E}_{\mathbf{s}\sim d_\pi(\mathbf{s})}\left[\underline{V^\pi(\mathbf{s})}\ \mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\left[\nabla_\pi \log\pi(\mathbf{a}|\mathbf{s})\right]\right]$$

Value Function Baseline

$$\mathbb{E}_{\mathbf{s}\sim d_\pi(\mathbf{s})}\mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\mathbb{E}_{\tau\sim p(\tau|\pi,\mathbf{s}_0=\mathbf{s},\mathbf{a}_0=\mathbf{a})}\left[\nabla_\pi \log\pi(\mathbf{a}|\mathbf{s})\left(\sum_{t'=0}^{T-1}\gamma^{t'}r_{t'} - V^\pi(\mathbf{s})\right)\right]$$

$$= \mathbb{E}_{\mathbf{s}\sim d_\pi(\mathbf{s})}\mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\mathbb{E}_{\tau\sim p(\tau|\pi,\mathbf{s}_0=\mathbf{s},\mathbf{a}_0=\mathbf{a})}\left[\nabla_\pi \log\pi(\mathbf{a}|\mathbf{s})\sum_{t'=0}^{T-1}\gamma^{t'}r_{t'}\right]$$

$$-\mathbb{E}_{\mathbf{s}\sim d_\pi(\mathbf{s})}\mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\mathbb{E}_{\tau\sim \pi(\tau|\pi,\mathbf{s}_0=\mathbf{s},\mathbf{a}_0=\mathbf{a})}\left[\nabla_\pi \log\pi(\mathbf{a}|\mathbf{s})V^\pi(\mathbf{s})\right]$$

$$= \mathbb{E}_{\mathbf{s}\sim d_\pi(\mathbf{s})}\mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\mathbb{E}_{\tau\sim p(\tau|\pi,\mathbf{s}_0=\mathbf{s},\mathbf{a}_0=\mathbf{a})}\left[\nabla_\pi \log\pi(\mathbf{a}|\mathbf{s})\sum_{t'=0}^{T-1}\gamma^{t'}r_{t'}\right]$$

$$-\mathbb{E}_{\mathbf{s}\sim d_\pi(\mathbf{s})}\left[V^\pi(\mathbf{s})\ \underline{\mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\left[\nabla_\pi \log\pi(\mathbf{a}|\mathbf{s})\right]}\right]$$

$$= \nabla_\pi \sum_{\mathbf{a}} \pi(\mathbf{a}|\mathbf{s}) = \nabla_\pi 1$$

Value Function Baseline

$$\mathbb{E}_{\mathbf{s}\sim d_\pi(\mathbf{s})}\mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\mathbb{E}_{\tau\sim p(\tau|\pi,\mathbf{s}_0=\mathbf{s},\mathbf{a}_0=\mathbf{a})}\left[\nabla_\pi \log\pi(\mathbf{a}|\mathbf{s})\left(\sum_{t'=0}^{T-1}\gamma^{t'}r_{t'} - V^\pi(\mathbf{s})\right)\right]$$

$$= \mathbb{E}_{\mathbf{s}\sim d_\pi(\mathbf{s})}\mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\mathbb{E}_{\tau\sim p(\tau|\pi,\mathbf{s}_0=\mathbf{s},\mathbf{a}_0=\mathbf{a})}\left[\nabla_\pi \log\pi(\mathbf{a}|\mathbf{s})\sum_{t'=0}^{T-1}\gamma^{t'}r_{t'}\right]$$

$$-\mathbb{E}_{\mathbf{s}\sim d_\pi(\mathbf{s})}\mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\mathbb{E}_{\tau\sim\pi(\tau|\pi,\mathbf{s}_0=\mathbf{s},\mathbf{a}_0=\mathbf{a})}\left[\nabla_\pi \log\pi(\mathbf{a}|\mathbf{s})V^\pi(\mathbf{s})\right]$$

$$= \mathbb{E}_{\mathbf{s}\sim d_\pi(\mathbf{s})}\mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\mathbb{E}_{\tau\sim p(\tau|\pi,\mathbf{s}_0=\mathbf{s},\mathbf{a}_0=\mathbf{a})}\left[\nabla_\pi \log\pi(\mathbf{a}|\mathbf{s})\sum_{t'=0}^{T-1}\gamma^{t'}r_{t'}\right]$$

$$-\mathbb{E}_{\mathbf{s}\sim d_\pi(\mathbf{s})}\left[V^\pi(\mathbf{s})\ \cancel{\mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\left[\nabla_\pi \log\pi(\mathbf{a}|\mathbf{s})\right]}^{\,0}\right]$$

Value function baseline is unbiased!

Value Function Baseline

$$\nabla_\pi J(\pi) = \mathbb{E}_{\mathbf{s}\sim d_\pi(\mathbf{s})}\mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\mathbb{E}_{\tau\sim p(\tau|\pi,\mathbf{s}_0=\mathbf{s},\mathbf{a}_0=\mathbf{a})}\left[\nabla_\pi \log\pi(\mathbf{a}|\mathbf{s})\left(\sum_{t'=0}^{T-1}\gamma^{t'}r_{t'} - V^\pi(\mathbf{s})\right)\right]$$

- Value function baseline is unbiased!
- Substantial variance reduction
- Any baseline that is only a function of the state is unbiased [Sutton et al. 1990]

Policy Gradient Methods for Reinforcement Learning with Function Approximation [Sutton et al. 1999]

Reward-to-Go Policy Gradient

ALGORITHM: Reward-to-Go Policy Gradient

1: $\theta \leftarrow$ initialize policy parameters
2: $V \leftarrow$ initialize value function parameters

3: **while** not done **do**
4: Sample trajectory τ from policy $\pi_\theta(\mathbf{a}|\mathbf{s})$
5: Fit value function $V(\mathbf{s})$

6: **for** every timestep t **do**
7: $\nabla_t \leftarrow \left(\sum_{t'=t}^{T-1} \gamma^{t'-t} r_{t'} - V(\mathbf{s})\right) \nabla_\theta \log \pi_\theta(\mathbf{a}_t|\mathbf{s}_t)$
8: **end for**

9: $\nabla_\theta J(\pi_\theta) \approx \frac{1}{T} \sum_{t=0}^{T-1} \nabla_t$
10: Update policy $\theta \leftarrow \theta + \alpha \nabla_\theta J(\pi_\theta)$
11: **end while**

12: return policy π_θ

Reward-to-Go Policy Gradient

ALGORITHM: Reward-to-Go Policy Gradient

$\theta \leftarrow$ initialize policy parameters
$V \leftarrow$ initialize value function parameters

while not done **do**
    Sample trajectory τ from policy $\pi_\theta(\mathbf{a}|\mathbf{s})$
    Fit value function $V(\mathbf{s})$

    for every timestep t **do**
        $\nabla_t \leftarrow \left(\sum_{t'=t}^{T-1} \gamma^{t'-t} r_{t'} - V(\mathbf{s})\right) \nabla_\theta \log \pi_\theta(\mathbf{a}_t|\mathbf{s}_t)$
    end for

    $\nabla_\theta J(\pi_\theta) \approx \frac{1}{T} \sum_{t=0}^{T-1} \nabla_t$
    Update policy $\theta \leftarrow \theta + \alpha \nabla_\theta J(\pi_\theta)$
end while

return policy π_θ

Reward-to-Go Policy Gradient

ALGORITHM: Reward-to-Go Policy Gradient

$\theta \leftarrow$ initialize policy parameters
$V \leftarrow$ initialize value function parameters

while not done **do**
    Sample trajectory τ from policy $\pi_\theta(\mathbf{a}|\mathbf{s})$
    Fit value function $V(\mathbf{s})$

    for every timestep t **do**
        $\nabla_t \leftarrow \left(\sum_{t'=t}^{T-1} \gamma^{t'-t} r_{t'} - V(\mathbf{s})\right) \nabla_\theta \log\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)$
    end for

    $\nabla_\theta J(\pi_\theta) \approx \frac{1}{T}\sum_{t=0}^{T-1} \nabla_t$
    Update policy $\theta \leftarrow \theta + \alpha \nabla_\theta J(\pi_\theta)$
end while

return policy π_θ

Reward-to-Go Policy Gradient

ALGORITHM: Reward-to-Go Policy Gradient

$\theta \leftarrow$ initialize policy parameters
$V \leftarrow$ initialize value function parameters

while not done **do**
    Sample trajectory τ from policy $\pi_\theta(\mathbf{a}|\mathbf{s})$
    Fit value function $V(\mathbf{s})$

    for every timestep t **do**
        $\nabla_t \leftarrow \left(\sum_{t'=t}^{T-1} \gamma^{t'-t} r_{t'} - V(\mathbf{s})\right) \nabla_\theta \log \pi_\theta(\mathbf{a}_t|\mathbf{s}_t)$
    end for

    $\nabla_\theta J(\pi_\theta) \approx \frac{1}{T} \sum_{t=0}^{T-1} \nabla_t$
    Update policy $\theta \leftarrow \theta + \alpha \nabla_\theta J(\pi_\theta)$
end while

return policy π_θ

Reward-to-Go Policy Gradient

ALGORITHM: Reward-to-Go Policy Gradient

$\theta \leftarrow$ initialize policy parameters
$V \leftarrow$ initialize value function parameters

while not done **do**
  Sample trajectory τ from policy $\pi_\theta(\mathbf{a}|\mathbf{s})$
  Fit value function $V(\mathbf{s})$

  for every timestep t **do**
    $\nabla_t \leftarrow \left(\sum_{t'=t}^{T-1} \gamma^{t'-t} r_{t'} - V(\mathbf{s})\right) \nabla_\theta \log \pi_\theta(\mathbf{a}_t|\mathbf{s}_t)$
  end for

  $\nabla_\theta J(\pi_\theta) \approx \frac{1}{T} \sum_{t=0}^{T-1} \nabla_t$
  Update policy $\theta \leftarrow \theta + \alpha \nabla_\theta J(\pi_\theta)$
end while

return policy π_θ

Reward-to-Go Policy Gradient

ALGORITHM: Reward-to-Go Policy Gradient

$\theta \leftarrow$ initialize policy parameters
$V \leftarrow$ initialize value function parameters

while not done **do**
    Sample trajectory τ from policy $\pi_\theta(\mathbf{a}|\mathbf{s})$
    Fit value function $V(\mathbf{s})$

    for every timestep t **do**
        $\nabla_t \leftarrow \left(\sum_{t'=t}^{T-1} \gamma^{t'-t} r_{t'} - V(\mathbf{s})\right) \nabla_\theta \log \pi_\theta(\mathbf{a}_t|\mathbf{s}_t)$
    end for

    $\nabla_\theta J(\pi_\theta) \approx \frac{1}{T} \sum_{t=0}^{T-1} \nabla_t$
    Update policy $\theta \leftarrow \theta + \alpha \nabla_\theta J(\pi_\theta)$
end while

return policy π_θ

Reward-to-Go Policy Gradient

ALGORITHM: Reward-to-Go Policy Gradient

$\theta \leftarrow$ initialize policy parameters
$V \leftarrow$ initialize value function parameters

while not done **do**
  Sample trajectory τ from policy $\pi_\theta(\mathbf{a}|\mathbf{s})$
  Fit value function $V(\mathbf{s})$

  for every timestep t **do**
    $\nabla_t \leftarrow \left(\sum_{t'=t}^{T-1} \gamma^{t'-t} r_{t'} - V(\mathbf{s}) \right) \nabla_\theta \log \pi_\theta(\mathbf{a}_t|\mathbf{s}_t)$
  end for

  $\nabla_\theta J(\pi_\theta) \approx \frac{1}{T} \sum_{t=0}^{T-1} \nabla_t$
  Update policy $\theta \leftarrow \theta + \alpha \nabla_\theta J(\pi_\theta)$
end while

return policy π_θ

Reward-to-Go Policy Gradient

ALGORITHM: Reward-to-Go Policy Gradient

$\theta \leftarrow$ initialize policy parameters
$V \leftarrow$ initialize value function parameters

while not done **do**
  Sample trajectory τ from policy $\pi_\theta(\mathbf{a}|\mathbf{s})$
  Fit value function $V(\mathbf{s})$

  for every timestep t **do**
    $\nabla_t \leftarrow \left(\sum_{t'=t}^{T-1} \gamma^{t'-t} r_{t'} - V(\mathbf{s})\right) \underline{\nabla_\theta \log \pi_\theta(\mathbf{a}_t|\mathbf{s}_t)}$
  end for

  $\nabla_\theta J(\pi_\theta) \approx \frac{1}{T} \sum_{t=0}^{T-1} \nabla_t$
  Update policy $\theta \leftarrow \theta + \alpha \nabla_\theta J(\pi_\theta)$
end while

return policy π_θ

Reward-to-Go Policy Gradient

ALGORITHM: Reward-to-Go Policy Gradient

```
θ ← initialize policy parameters
V ← initialize value function parameters

while not done do
    Sample trajectory τ from policy π_θ(a|s)
    Fit value function V(s)

    for every timestep t do
        ∇_t ← ( Σ_{t'=t}^{T-1} γ^{t'-t} r_{t'} − V(s) ) ∇_θ log π_θ(a_t|s_t)
    end for

    ∇_θ J(π_θ) ≈ (1/T) Σ_{t=0}^{T-1} ∇_t
    Update policy θ ← θ + α ∇_θ J(π_θ)
end while

return policy π_θ
```

$$\nabla_t \leftarrow \left(\underline{\sum_{t'=t}^{T-1} \gamma^{t'-t} r_{t'}} - V(\mathbf{s})\right) \nabla_\theta \log \pi_\theta(\mathbf{a}_t|\mathbf{s}_t)$$

$$\nabla_\theta J(\pi_\theta) \approx \frac{1}{T}\sum_{t=0}^{T-1} \nabla_t$$

Reward-to-Go Policy Gradient

ALGORITHM: Reward-to-Go Policy Gradient

$\theta \leftarrow$ initialize policy parameters
$V \leftarrow$ initialize value function parameters

while not done **do**
  Sample trajectory τ from policy $\pi_\theta(\mathbf{a}|\mathbf{s})$
  Fit value function $V(\mathbf{s})$

  for every timestep t **do**
    $\nabla_t \leftarrow \left(\sum_{t'=t}^{T-1} \gamma^{t'-t} r_{t'} - V(\mathbf{s})\right) \nabla_\theta \log \pi_\theta(\mathbf{a}_t|\mathbf{s}_t)$
  end for

  $\nabla_\theta J(\pi_\theta) \approx \frac{1}{T} \sum_{t=0}^{T-1} \nabla_t$
  Update policy $\theta \leftarrow \theta + \alpha \nabla_\theta J(\pi_\theta)$
end while

return policy π_θ

Reward-to-Go Policy Gradient

ALGORITHM: Reward-to-Go Policy Gradient

$\theta \leftarrow$ initialize policy parameters
$V \leftarrow$ initialize value function parameters

while not done **do**
  Sample trajectory τ from policy $\pi_\theta(\mathbf{a}|\mathbf{s})$
  Fit value function $V(\mathbf{s})$

  for every timestep t **do**
    $\nabla_t \leftarrow \left(\sum_{t'=t}^{T-1} \gamma^{t'-t} r_{t'} - V(\mathbf{s})\right) \nabla_\theta \log \pi_\theta(\mathbf{a}_t|\mathbf{s}_t)$
  end for

  $\nabla_\theta J(\pi_\theta) \approx \frac{1}{T} \sum_{t=0}^{T-1} \nabla_t$
  Update policy $\theta \leftarrow \theta + \alpha \nabla_\theta J(\pi_\theta)$
end while

return policy π_θ

Reward-to-Go Policy Gradient

ALGORITHM: Reward-to-Go Policy Gradient

$\theta \leftarrow$ initialize policy parameters
$V \leftarrow$ initialize value function parameters

while not done **do**
  Sample trajectory τ from policy $\pi_\theta(\mathbf{a}|\mathbf{s})$
  Fit value function $V(\mathbf{s})$

  for every timestep t **do**
    $\nabla_t \leftarrow \left(\sum_{t'=t}^{T-1} \gamma^{t'-t} r_{t'} - V(\mathbf{s})\right) \nabla_\theta \log \pi_\theta(\mathbf{a}_t|\mathbf{s}_t)$
  end for

  $\nabla_\theta J(\pi_\theta) \approx \frac{1}{T} \sum_{t=0}^{T-1} \nabla_t$
  Update policy $\theta \leftarrow \theta + \alpha \nabla_\theta J(\pi_\theta)$
end while

return policy π_θ

Reward-to-Go Policy Gradient

ALGORITHM: Reward-to-Go Policy Gradient

1: $\theta \leftarrow$ initialize policy parameters
2: $V \leftarrow$ initialize value function parameters

3: **while** not done **do**
4: Sample trajectory τ from policy $\pi_\theta(\mathbf{a}|\mathbf{s})$
5: Fit value function $V(\mathbf{s})$

6: **for** every timestep t **do**
7: $\nabla_t \leftarrow \left(\sum_{t'=t}^{T-1} \gamma^{t'-t} r_{t'} - V(\mathbf{s})\right) \nabla_\theta \log \pi_\theta(\mathbf{a}_t|\mathbf{s}_t)$
8: **end for**

9: $\nabla_\theta J(\pi_\theta) \approx \frac{1}{T} \sum_{t=0}^{T-1} \nabla_t$
10: Update policy $\theta \leftarrow \theta + \alpha \nabla_\theta J(\pi_\theta)$
11: **end while**

12: return policy π_θ

Variance Reduction

- Baselines
- Causality
- <u>Bootstrapping</u>

Variance Reduction: Bootstrapping

$$\nabla_\pi J(\pi) = \mathbb{E}_{\mathbf{s}\sim d_\pi(\mathbf{s})}\mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\mathbb{E}_{\tau\sim p(\tau|\pi,\mathbf{s}_0=\mathbf{s},\mathbf{a}_0=\mathbf{a})}\left[\nabla_\pi \log\pi(\mathbf{a}|\mathbf{s})\left(\sum_{t'=0}^{T-1}\gamma^{t'} r_{t'} - V^\pi(\mathbf{s})\right)\right]$$

sum of random variables → high variance

$$r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots$$

Variance Reduction: Bootstrapping

$$\nabla_\pi J(\pi) = \mathbb{E}_{\mathbf{s}\sim d_\pi(\mathbf{s})}\mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\mathbb{E}_{\tau\sim p(\tau|\pi,\mathbf{s}_0=\mathbf{s},\mathbf{a}_0=\mathbf{a})}\left[\nabla_\pi \log\pi(\mathbf{a}|\mathbf{s})\left(\underline{\sum_{t'=0}^{T-1}\gamma^{t'}r_{t'}} - V^\pi(\mathbf{s})\right)\right]$$

sum of random variables → high variance

n-step return: $r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots + \gamma^{k-1} r_{k-1}$

Variance Reduction: Bootstrapping

$$\nabla_\pi J(\pi) = \mathbb{E}_{\mathbf{s}\sim d_\pi(\mathbf{s})}\mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\mathbb{E}_{\tau\sim p(\tau|\pi,\mathbf{s}_0=\mathbf{s},\mathbf{a}_0=\mathbf{a})}\left[\nabla_\pi \log\pi(\mathbf{a}|\mathbf{s})\left(\underline{\sum_{t'=0}^{T-1}\gamma^{t'}r_{t'}} - V^\pi(\mathbf{s})\right)\right]$$

sum of random variables → high variance

n-step return: $r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots + \gamma^{k-1} r_{k-1} \underbrace{+ \gamma^k V^\pi(\mathbf{s}_k)}_{\text{bootstrap}}$

N-Step Bootstrapping

1-step bootstrap: $y = r_0 + \gamma \hat{V}^{\pi}(\mathbf{s}_1)$

2-step bootstrap: $y = r_0 + \gamma r_1 + \gamma^2 \hat{V}^{\pi}(\mathbf{s}_2)$

3-step bootstrap: $y = r_0 + \gamma r_1 + \gamma^2 r_2 + \gamma^3 \hat{V}^{\pi}(\mathbf{s}_3)$

⋮

n-step bootstrap: $y = \underbrace{\sum_{t=0}^{n-1} \gamma^t r_t}_{\text{High variance}} + \gamma^n \underbrace{\hat{V}^{\pi}(\mathbf{s}_n)}_{\text{Biased}}$

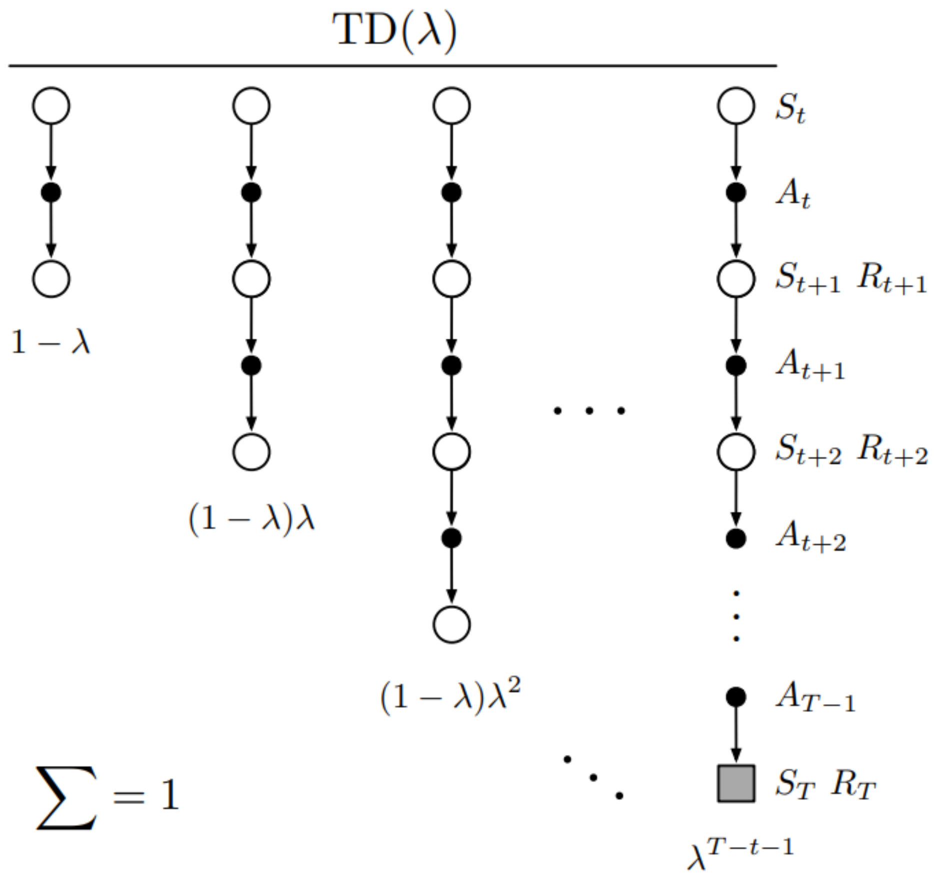
TD(λ)

- Use TD(λ) to estimate return

$$\nabla_\pi \log \pi(\mathbf{a}|\mathbf{s}) \left(\underline{\sum_{t=0} \gamma^t r_t} - V^\pi(\mathbf{s}) \right)$$

$$\Downarrow$$

$$\nabla_\pi \log \pi(\mathbf{a}|\mathbf{s}) \left(\underbrace{R^\lambda(\mathbf{s}, \mathbf{a})}_{\lambda\text{-return}} - V^\pi(\mathbf{s}) \right)$$



Reinforcement Learning: An Introduction
[Sutton and Barto 1998]

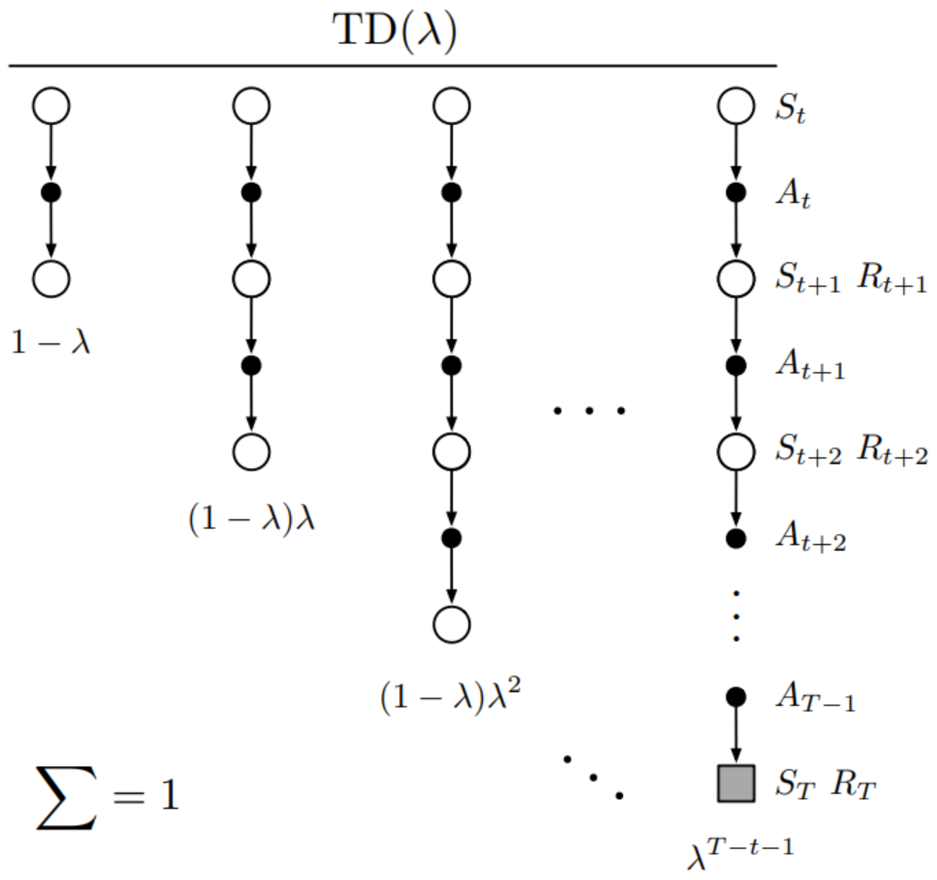
TD(λ)

- Use TD(λ) to estimate return

$$\nabla_\pi \log \pi(\mathbf{a}|\mathbf{s}) \left(\sum_{t=0} \gamma^t r_t - V^\pi(\mathbf{s}) \right)$$

$$\nabla_\pi \log \pi(\mathbf{a}|\mathbf{s}) \underbrace{\left(R^\lambda(\mathbf{s}, \mathbf{a}) - V^\pi(\mathbf{s}) \right)}_{\text{Generalized Advantage Estimation (GAE)}}$$

High-Dimensional Continuous Control Using Generalized Advantage Estimation [Schulman et al. 2016]



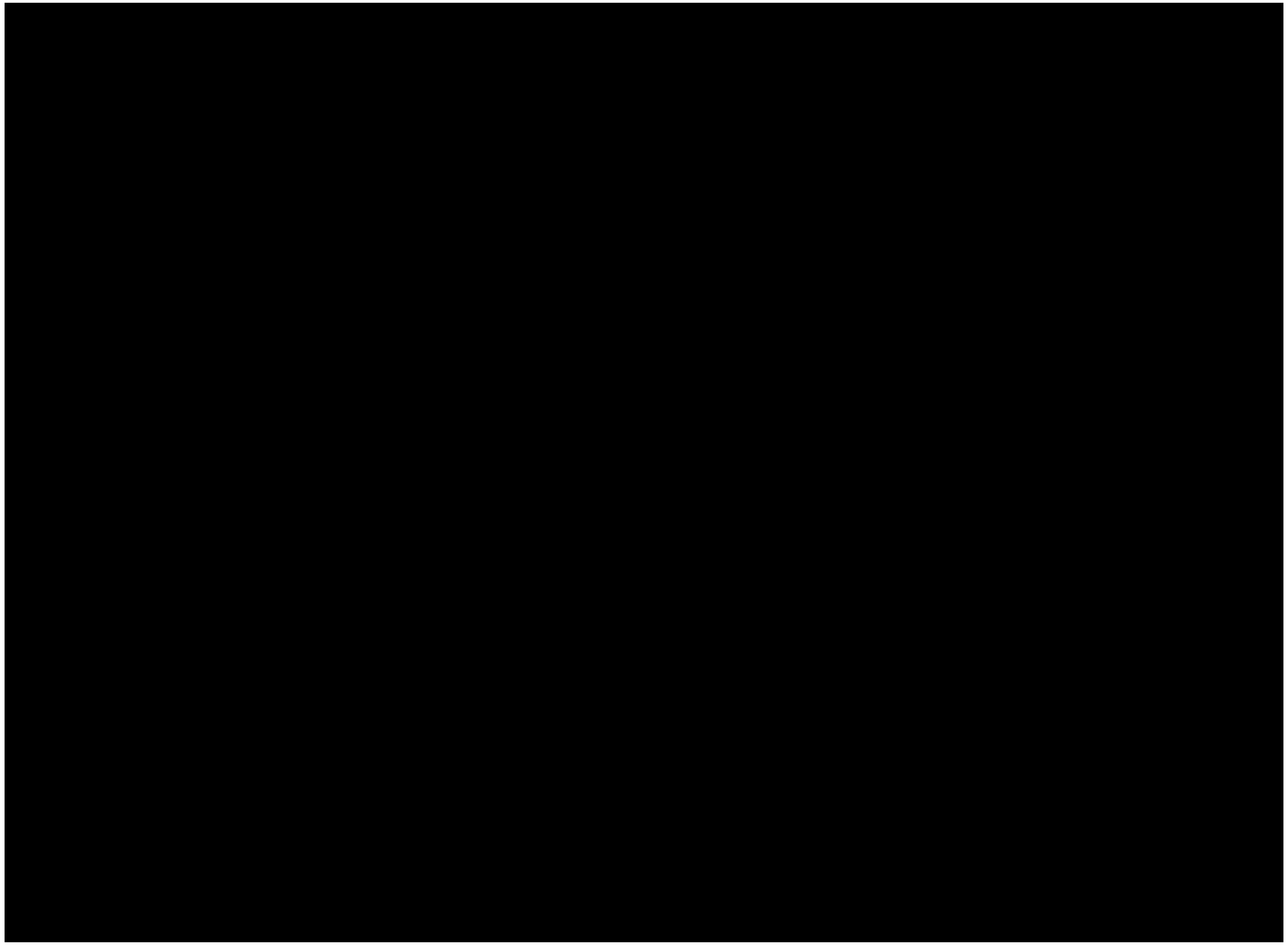
Reinforcement Learning: An Introduction [Sutton and Barto 1998]

Variance Reduction

- Baselines
- Causality
- Bootstrapping

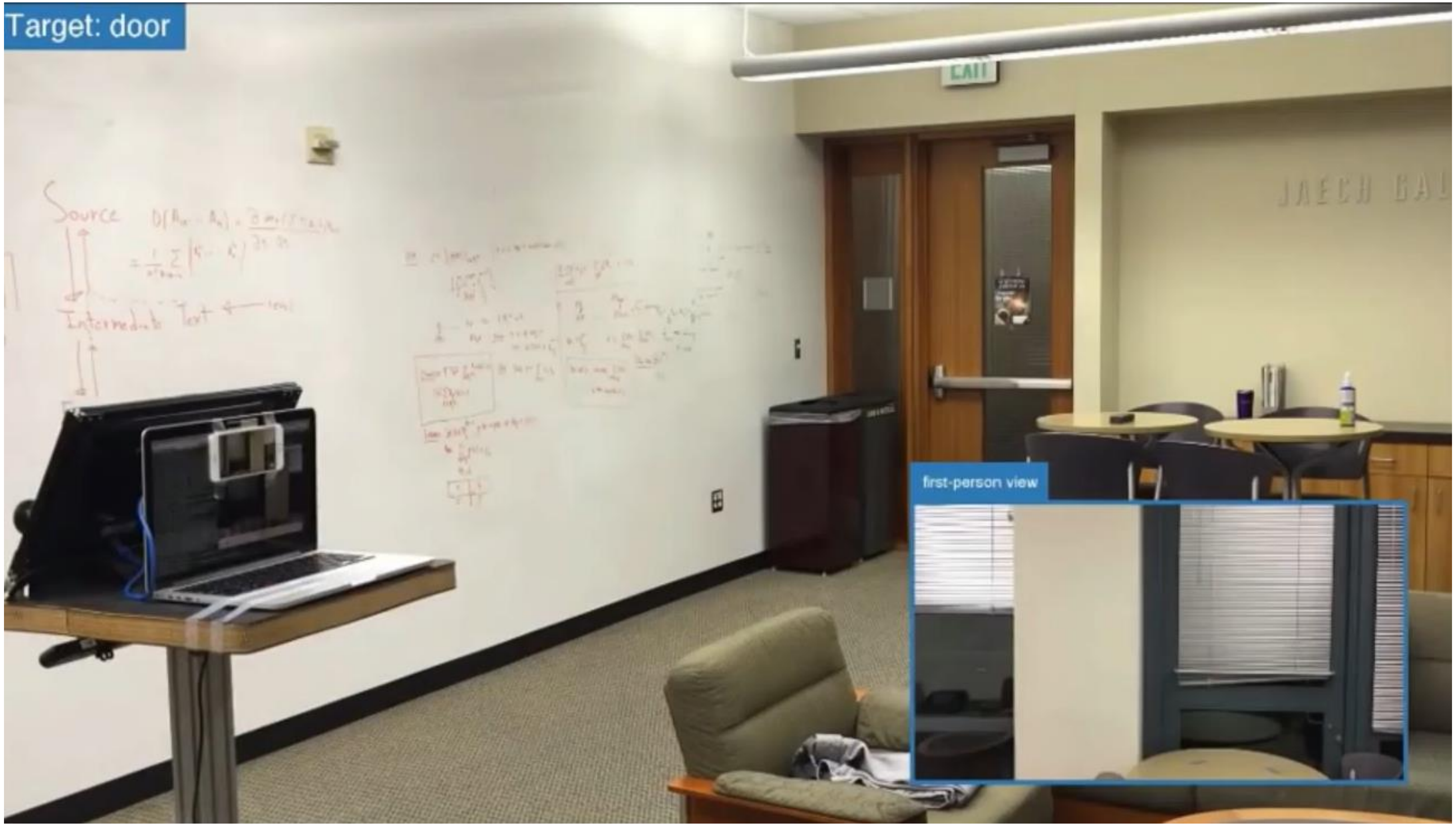
Applications

Visual Navigation



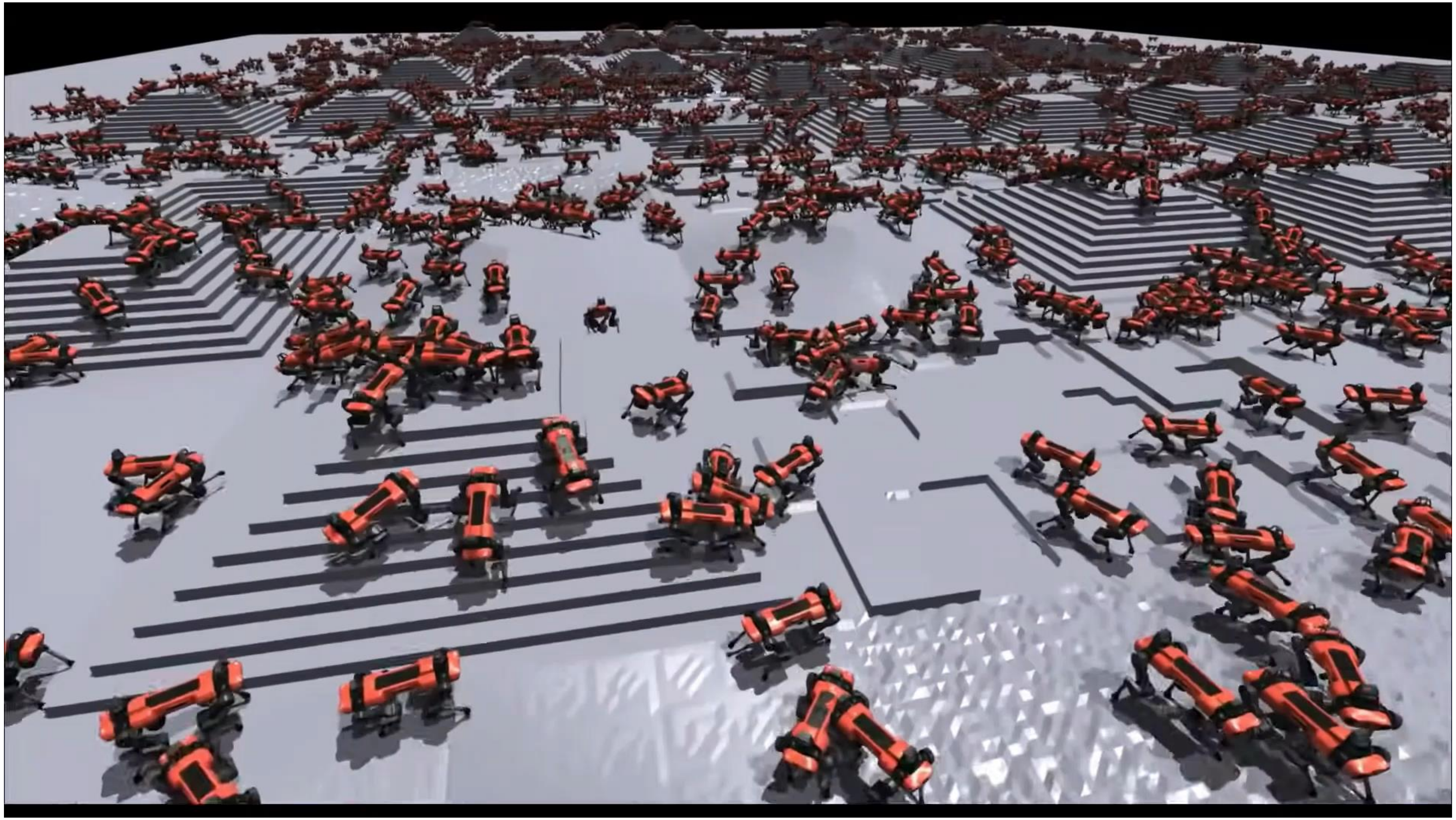
Asynchronous Methods for Deep Reinforcement Learning
[Mnih et al. 2016]

Visual Navigation



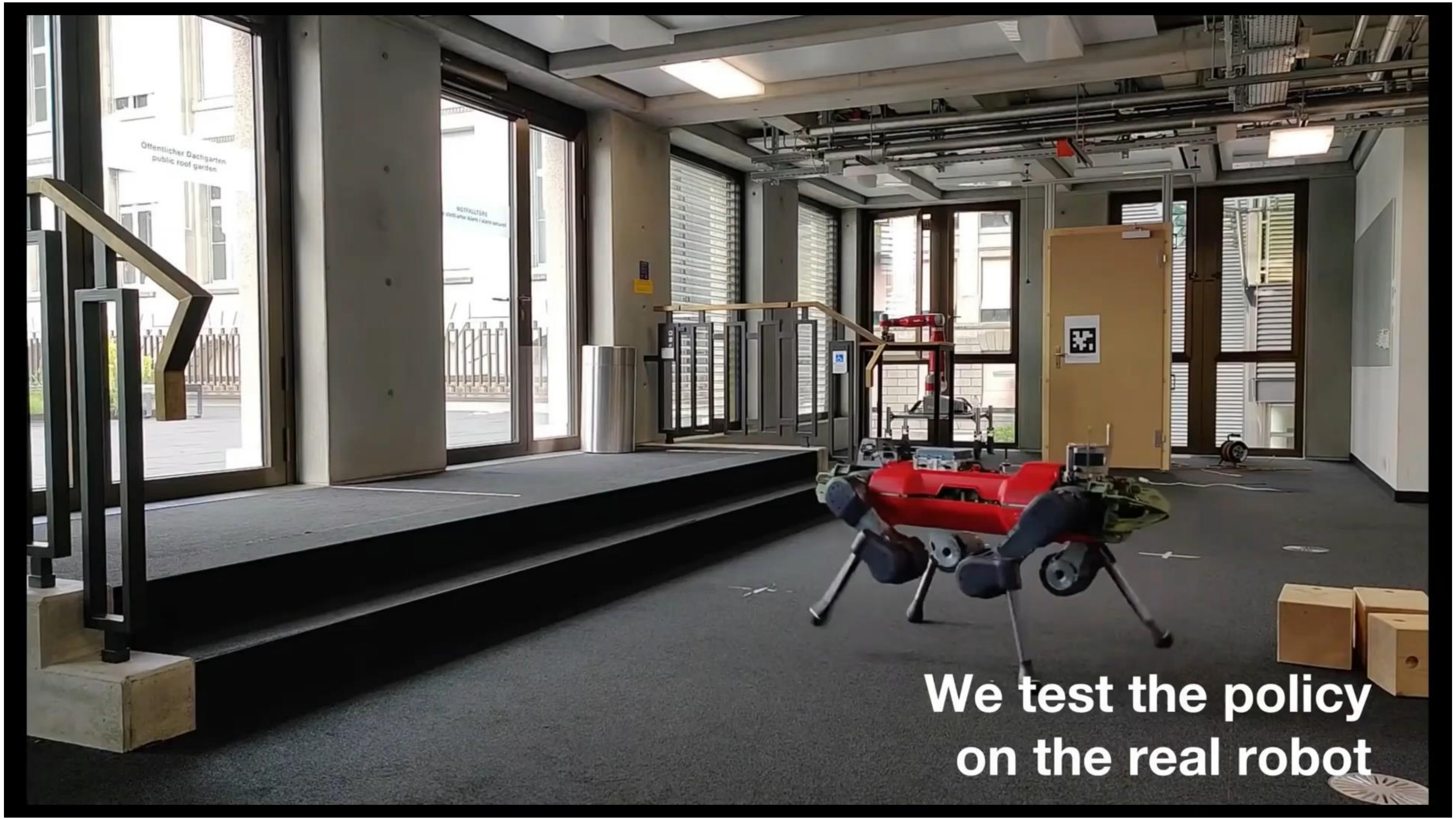
Target-driven Visual Navigation in Indoor Scenes using Deep Reinforcement Learning
[Zhu et al. 2017]

Robotic Locomotion



Learning to Walk in Minutes Using Massively Parallel Deep Reinforcement Learning
[Rudin et al. 2022]

Robotic Locomotion



Learning to Walk in Minutes Using Massively Parallel Deep Reinforcement Learning
[Rudin et al. 2022]

Policy Gradient

- ✓ Directly optimize $J(\pi)$ by estimating gradient $\nabla_\pi J(\pi)$
- ✓ General: can be applied to continuous and discrete states and actions
- ✘ High-variance gradient estimator → unstable/slow convergence
- ✘ Very sample inefficient

General View of PG

Nondifferentiable Functions

- Why does PG allow us to calculate gradients for a nondifferentiable function?
  - Gradient exists but unknown
  - Gradient does not exist

$$\underline{\nabla_\pi J(\pi)} = \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[R(\tau) \sum_{t=0}^{T-1} \nabla_\pi \log \pi(\mathbf{a}_t | \mathbf{s}_t) \right]$$

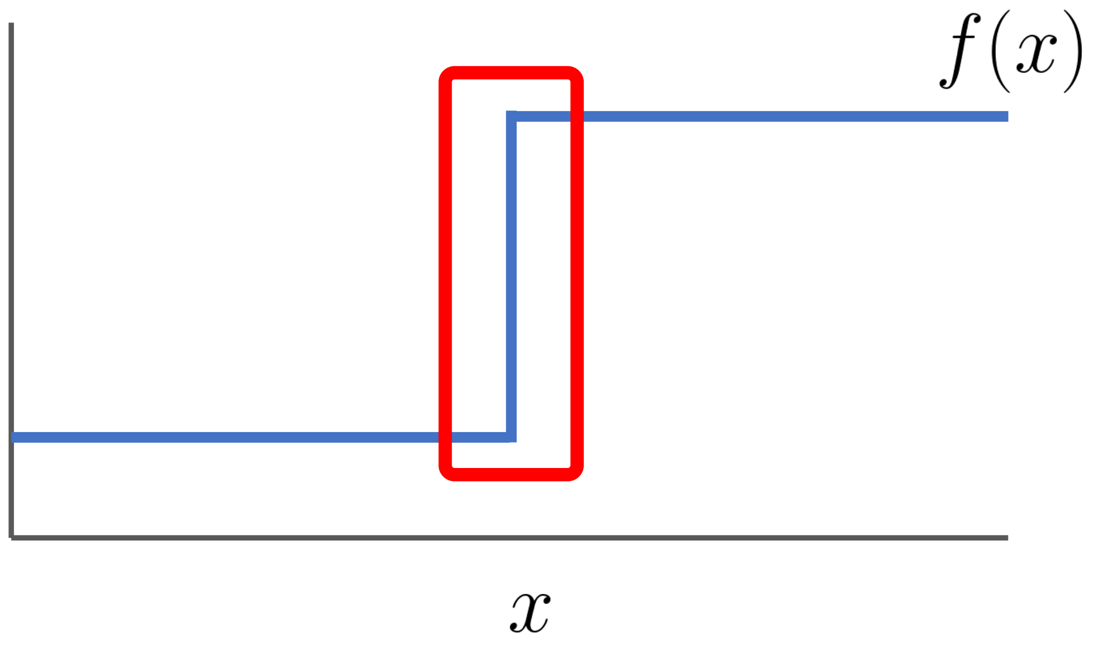
Nondifferentiable Functions

$$\underset{x}{\arg\max}\ \underline{f(x)}$$

$$\cancel{\nabla_x f(x)}$$

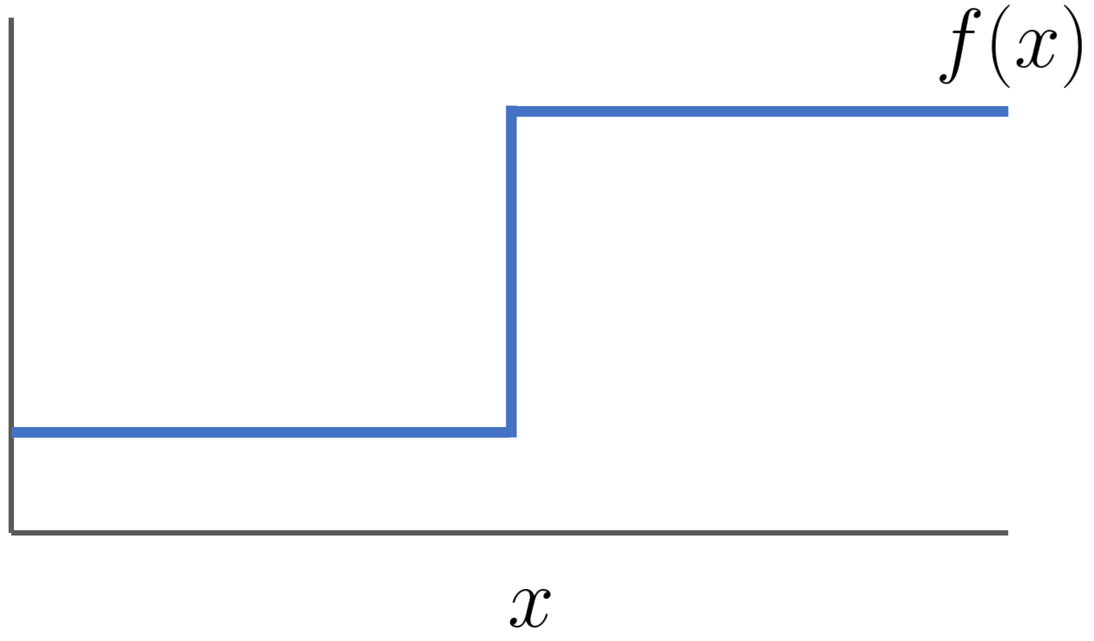
Nondifferentiable Functions

$$\arg\max_x \; f(x)$$



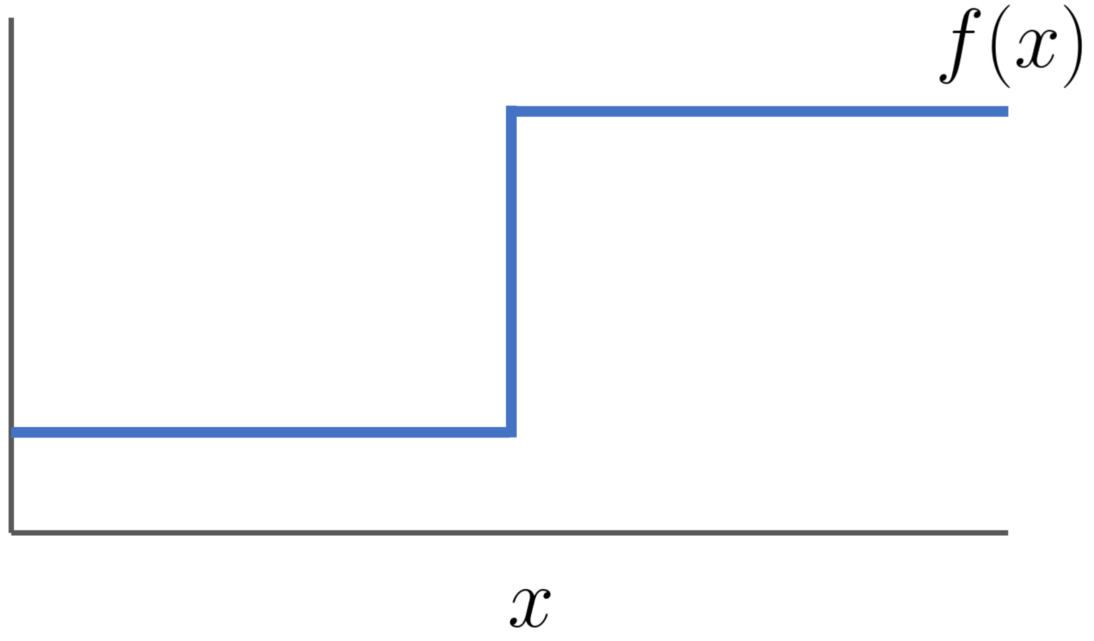
Nondifferentiable Functions

$$\arg\max_x \; f(x) \Longrightarrow \arg\max_{\underline{p}} \; \mathbb{E}_{x \sim p(x)}\left[f(x)\right]$$



Nondifferentiable Functions

$$\arg\max_x \; f(x) \Longrightarrow \arg\max_p \; \mathbb{E}_{x \sim p(x)}\left[f(x)\right]$$

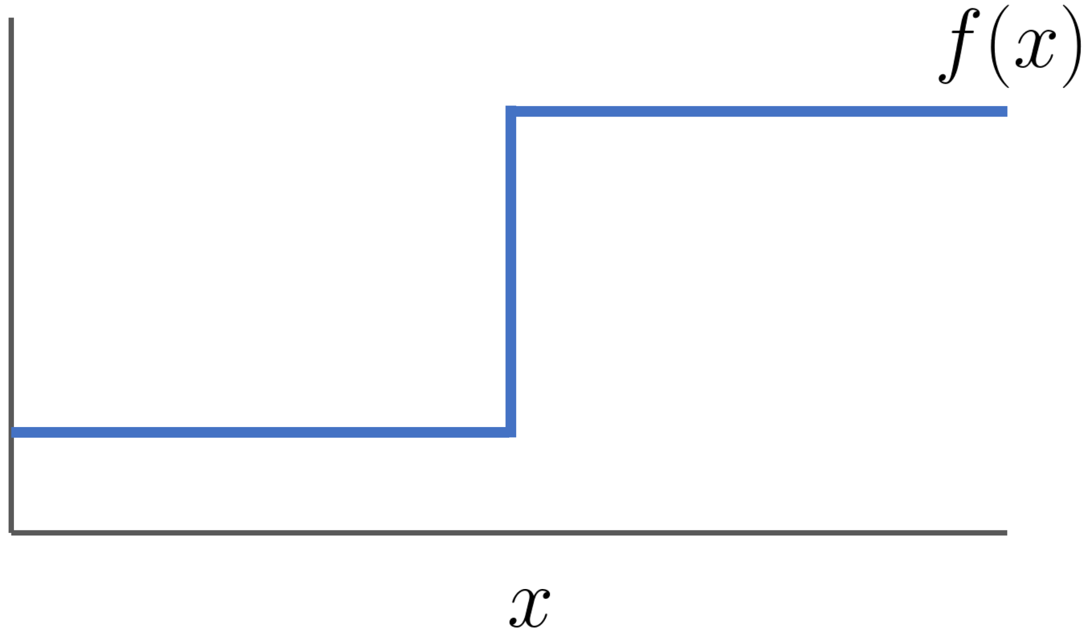


Nondifferentiable Functions

$$\arg\max_x f(x) \Longrightarrow \arg\max_p \mathbb{E}_{x\sim p(x)}[f(x)]$$

Score Function

$$\nabla_p = \mathbb{E}_{x\sim p(x)}\left[\underline{f(x)}\nabla_p \log p(x)\right]$$

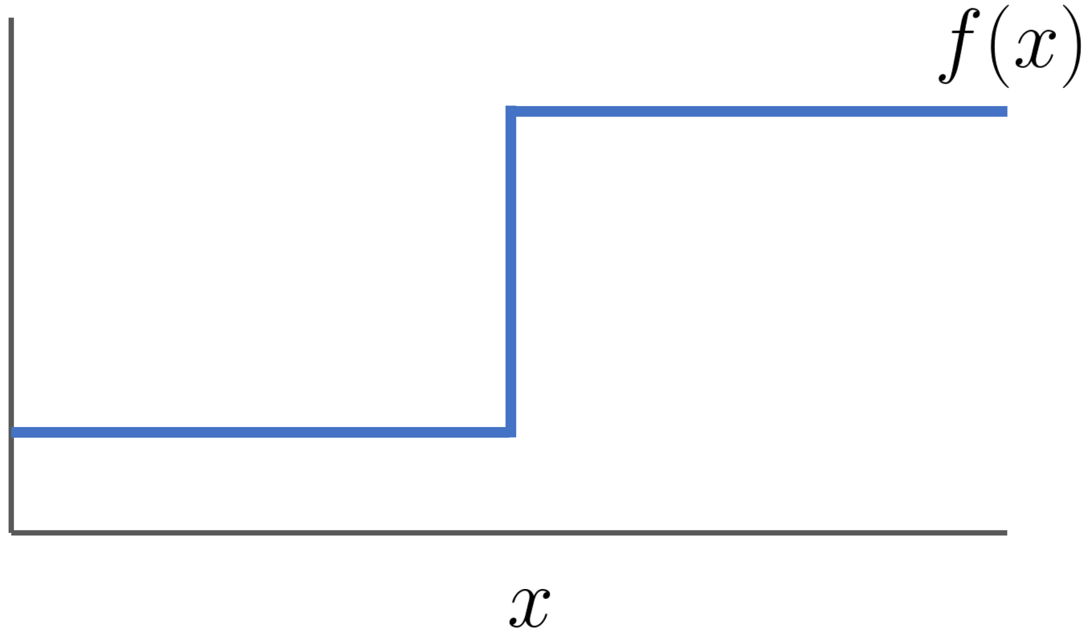


Nondifferentiable Functions

$$\arg\max_{x}\ f(x) \quad \Longrightarrow \quad \arg\max_{p}\ \mathbb{E}_{x\sim p(x)}\left[f(x)\right]$$

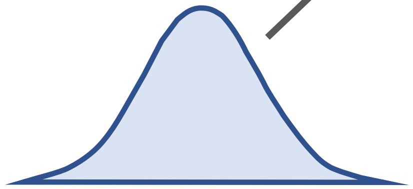
Score Function

$$\nabla_p = \mathbb{E}_{x\sim p(x)}\left[f(x)\underline{\nabla_p \mathrm{log} p(x)}\right]$$



Nondifferentiable Functions

$$\arg\max_{p} \; \mathbb{E}_{x \sim p(x)} \left[f(x) \right]$$



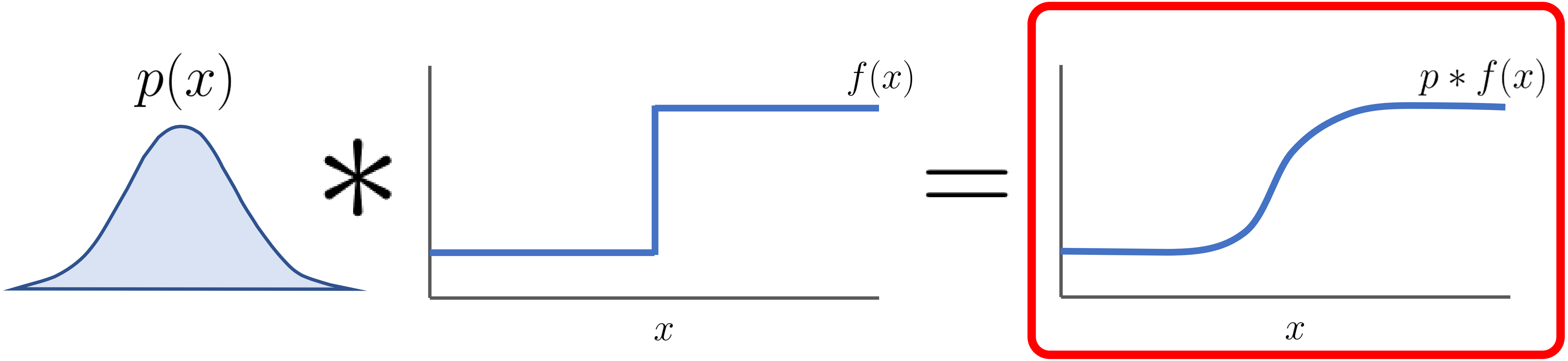
Nondifferentiable Functions

$$\arg\max_{p} \underline{\mathbb{E}_{x\sim p(x)}\left[f(x)\right]}$$

$$= \underbrace{\sum_{x} p(x) f(x)}_{\text{This is a convolution!}}$$

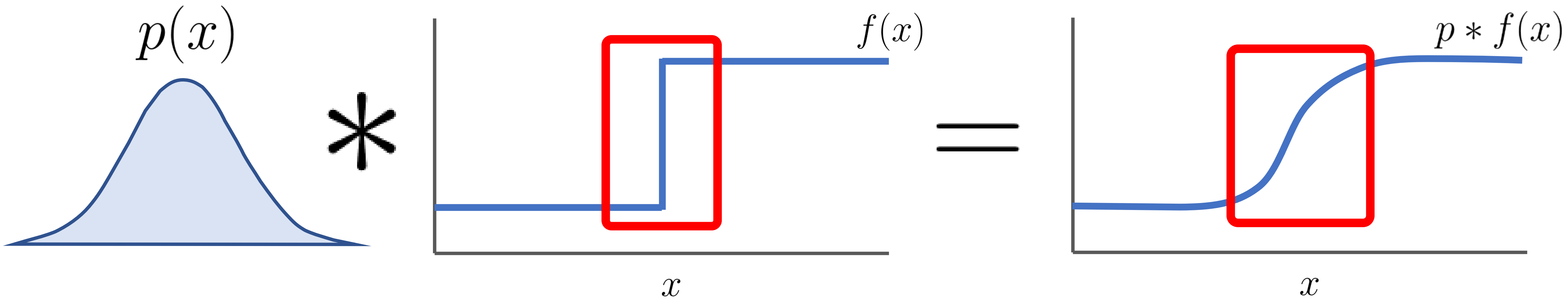
Nondifferentiable Functions

$$\arg\max_p \; \mathbb{E}_{x \sim p(x)} \left[f(x) \right]$$



Nondifferentiable Functions

$$\arg\max_{p} \; \mathbb{E}_{x \sim p(x)} \left[f(x) \right]$$



Score Function

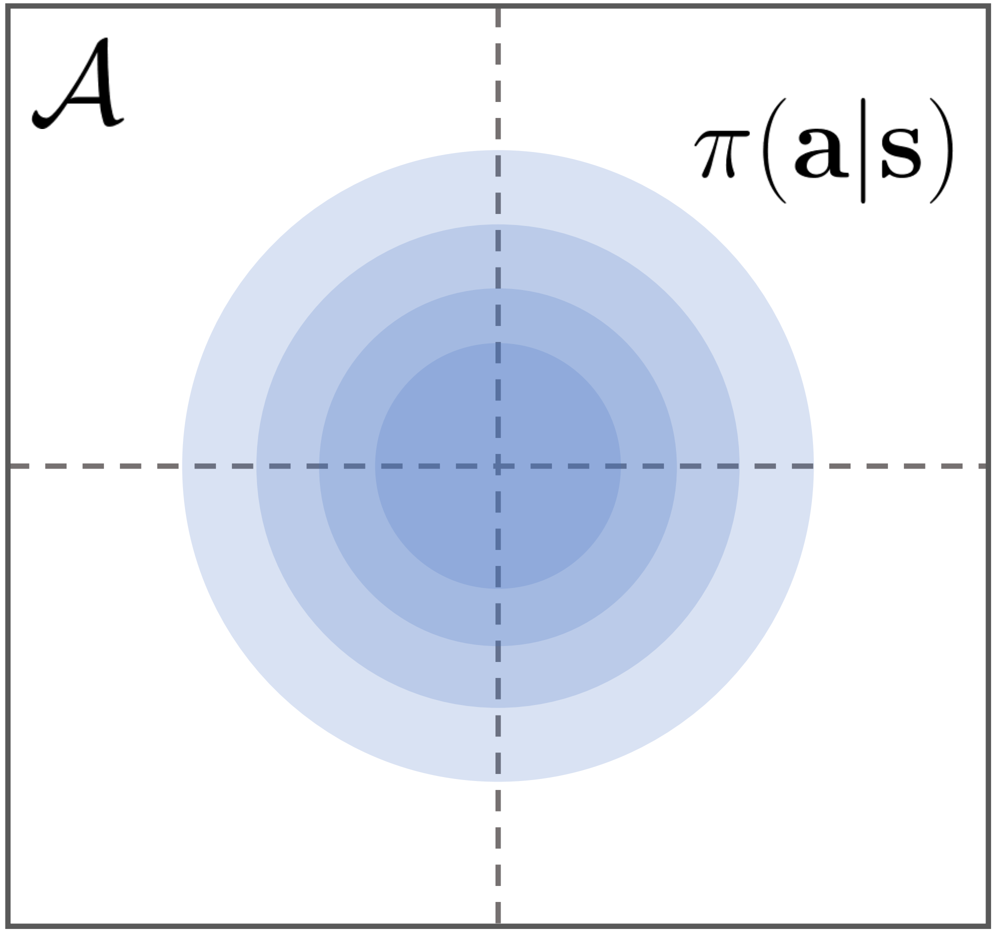
- Score function can be applied to calculate gradients for ***any*** nondifferentiable function
  - Converts an optimization of a ***deterministic*** variable into an optimization of a ***stochastic*** distribution

- Policy gradient only works for stochastic policies

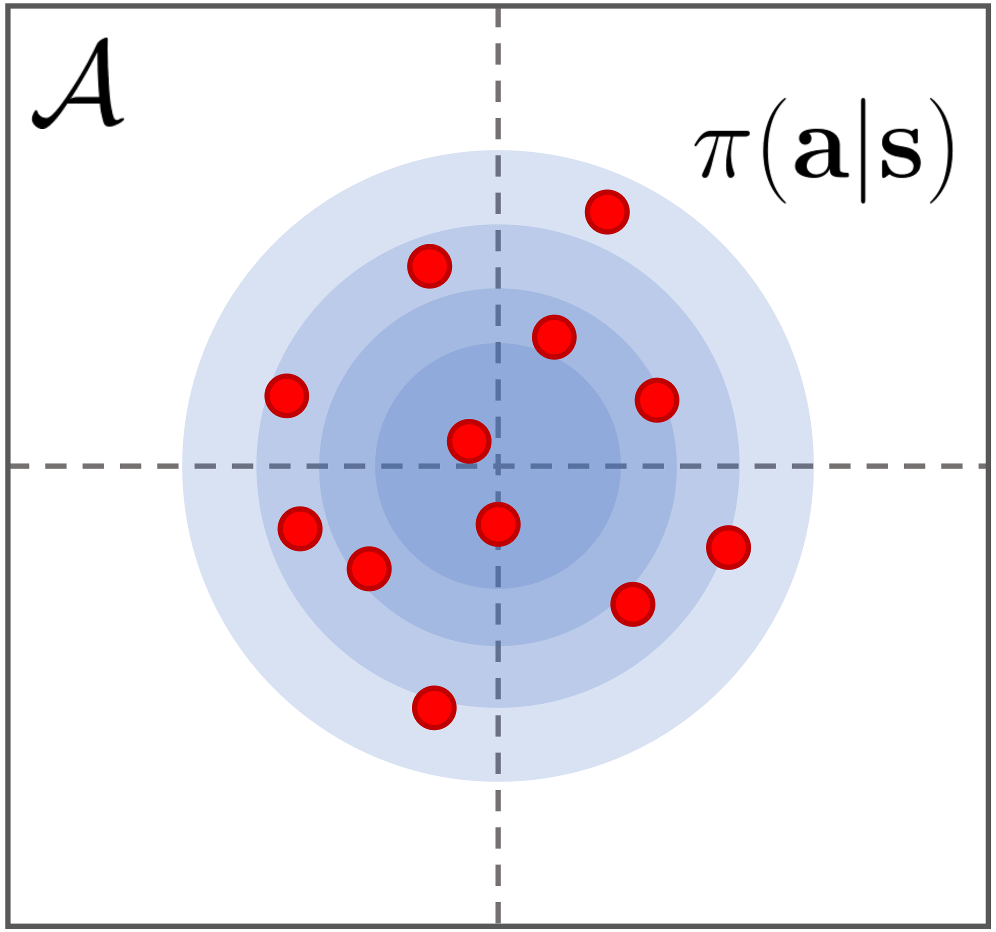
$$\mathbf{a} \sim \pi(\mathbf{a}|\mathbf{s}) \qquad\qquad \mathbf{a} \cancel{=} \pi(\mathbf{s})$$

Gradient Estimation Using Stochastic Computation Graphs
[Schulman et al. 2015]

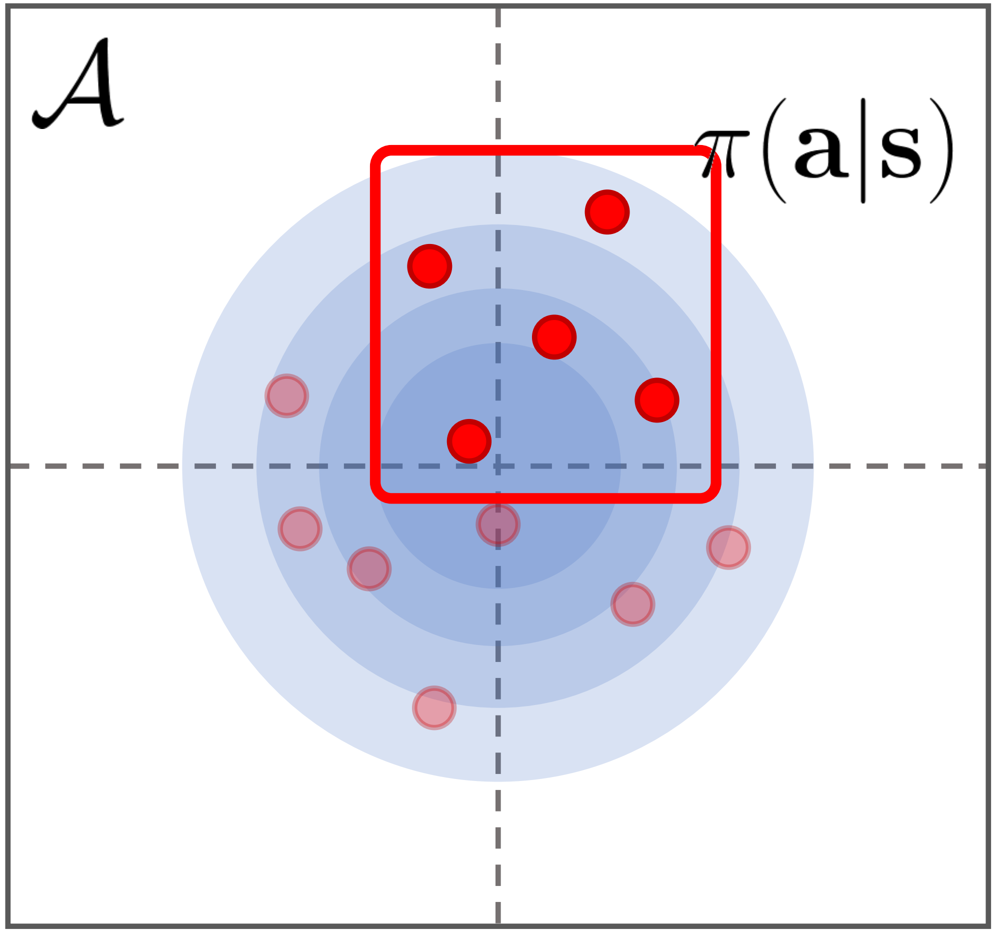
Evolutionary Strategies



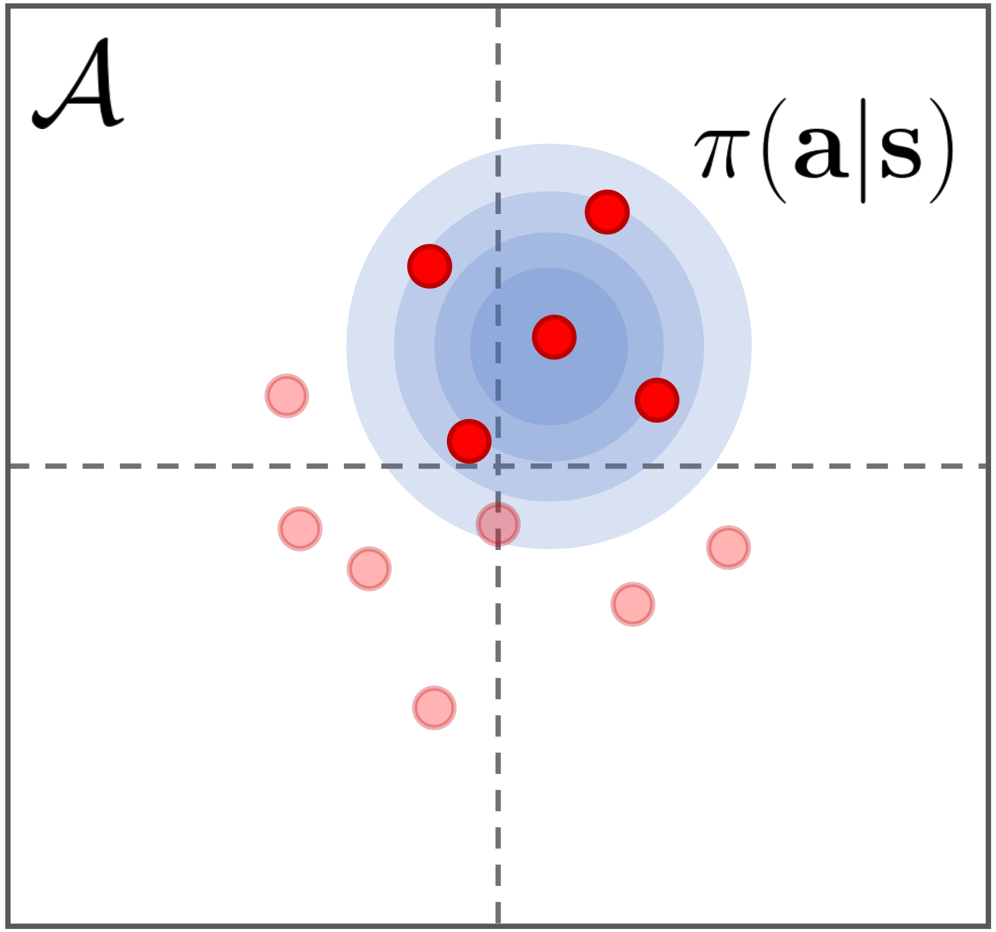
Evolutionary Strategies



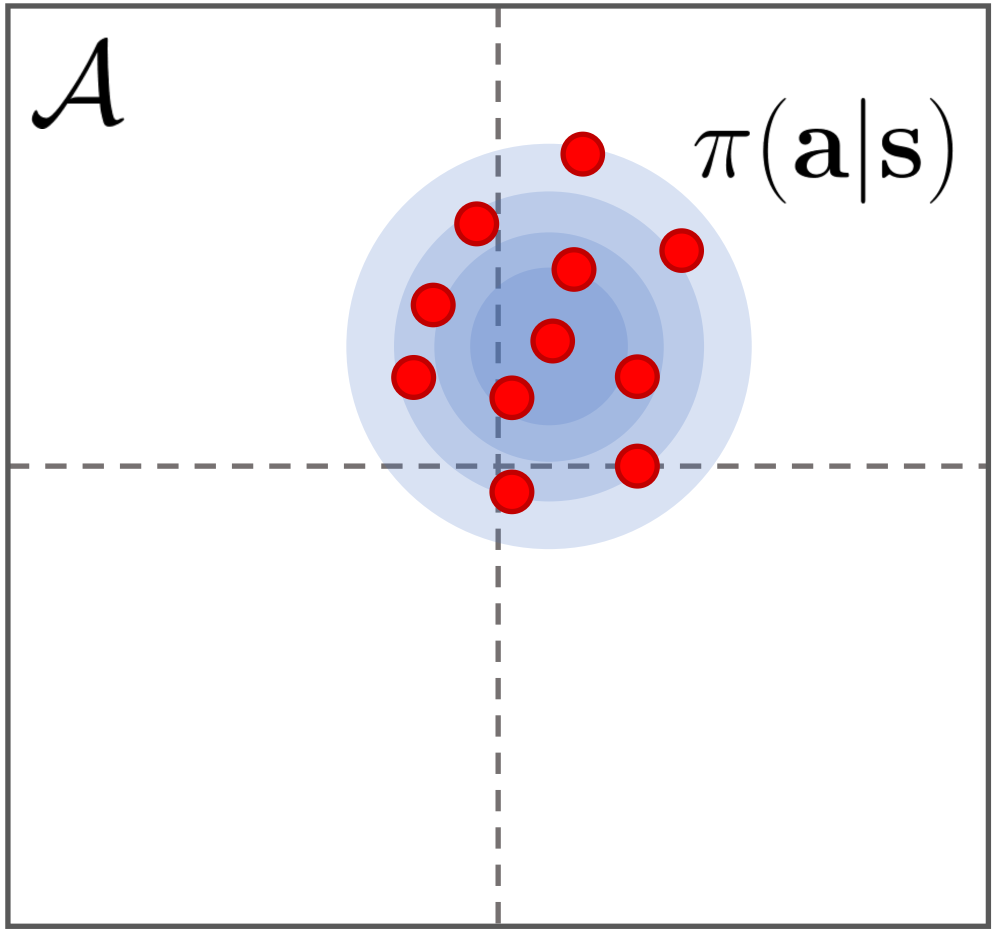
Evolutionary Strategies



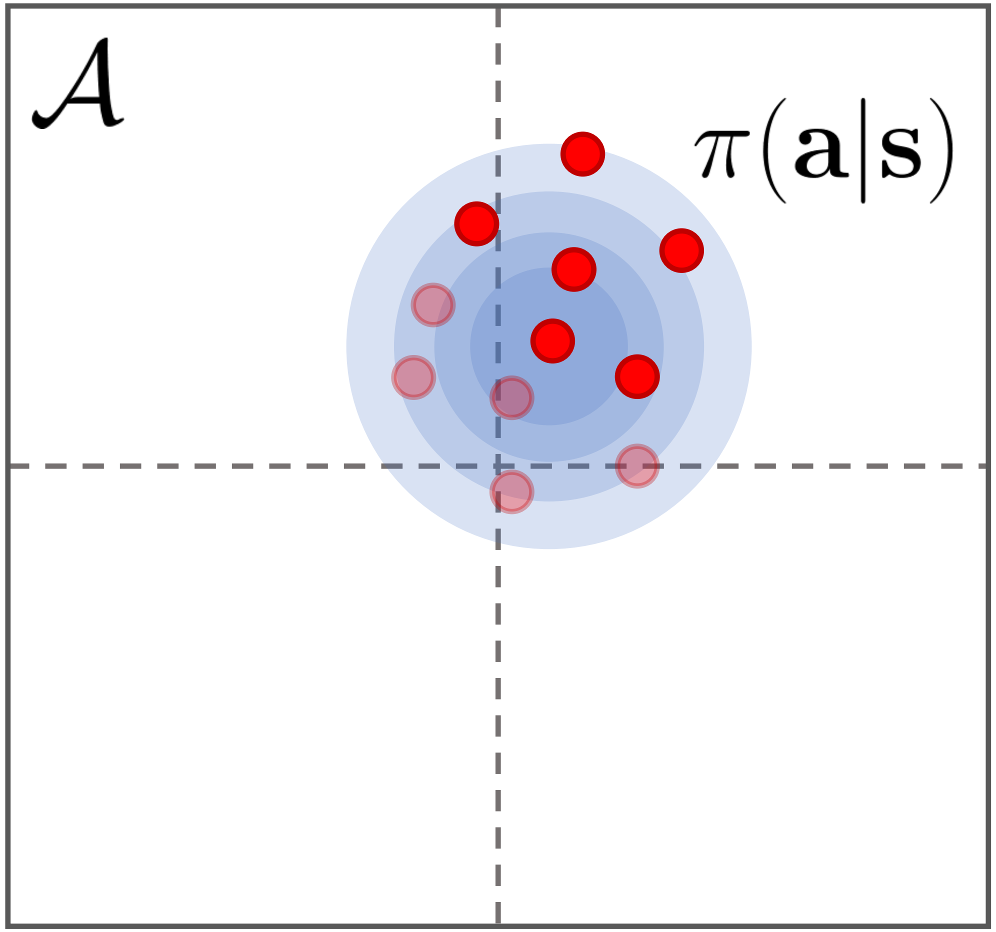
Evolutionary Strategies



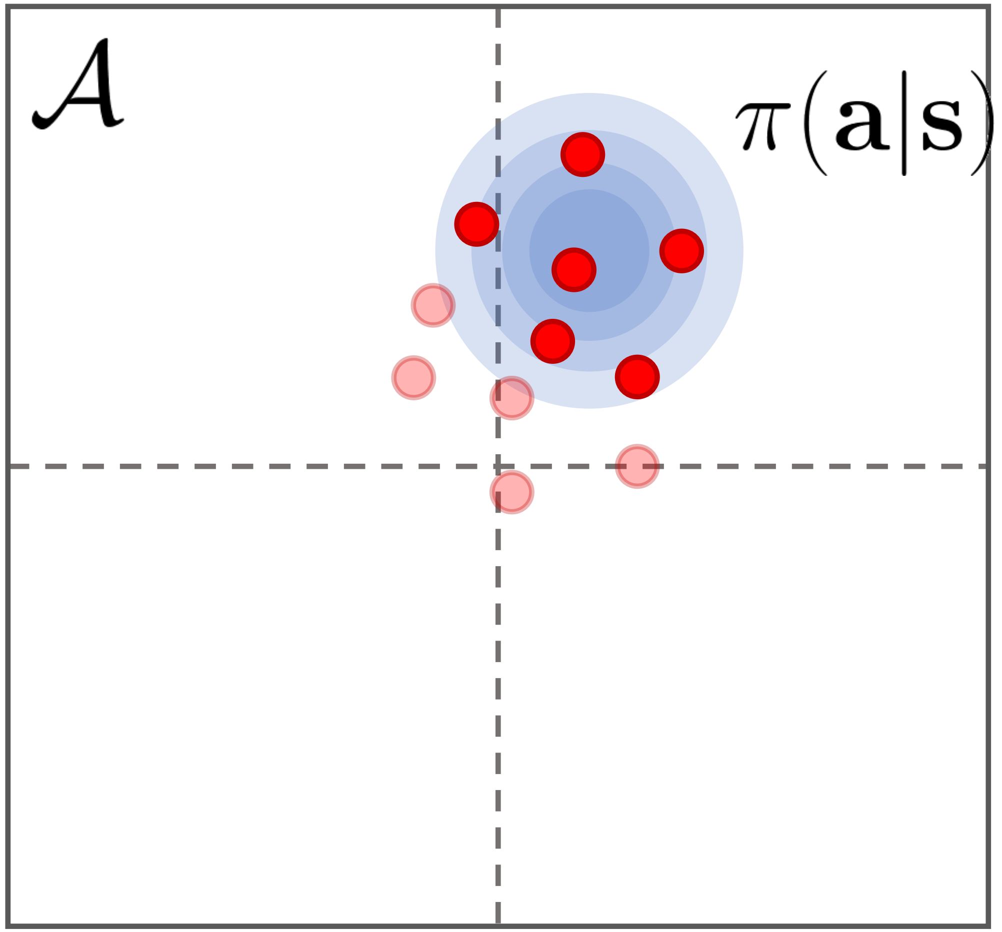
Evolutionary Strategies



Evolutionary Strategies

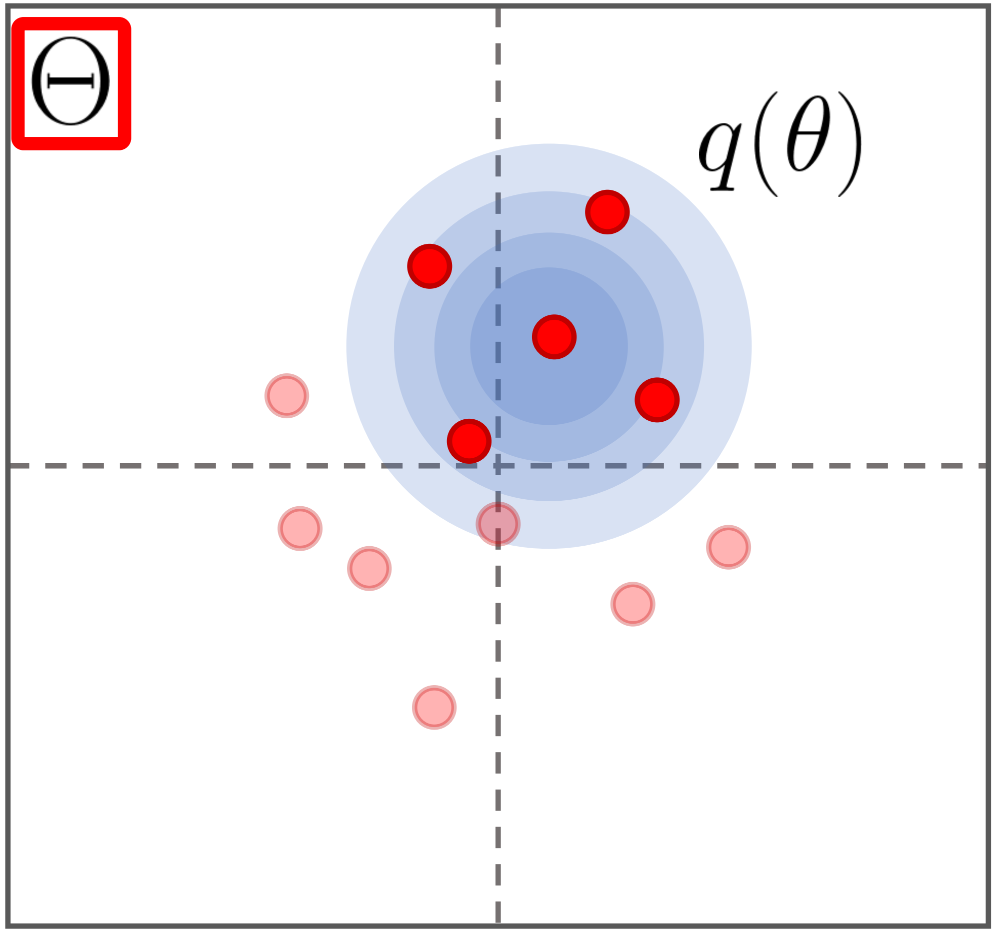


Evolutionary Strategies

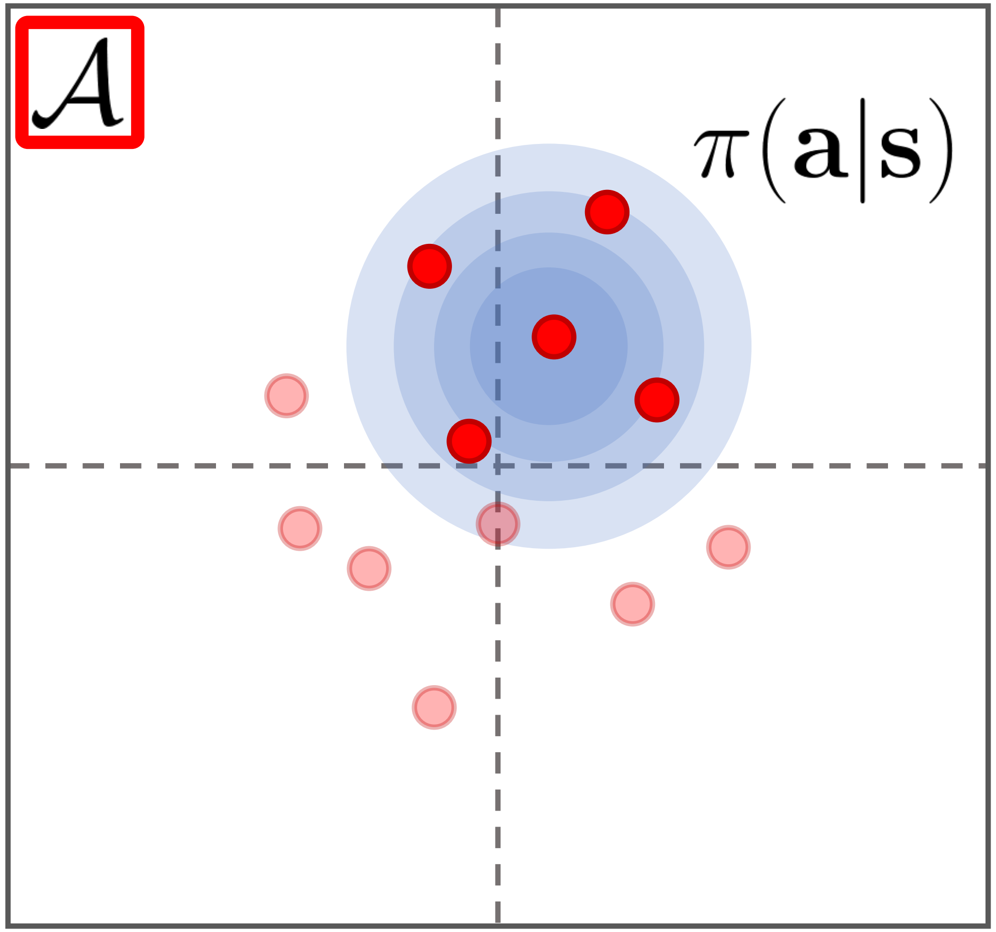


Evolutionary Strategies

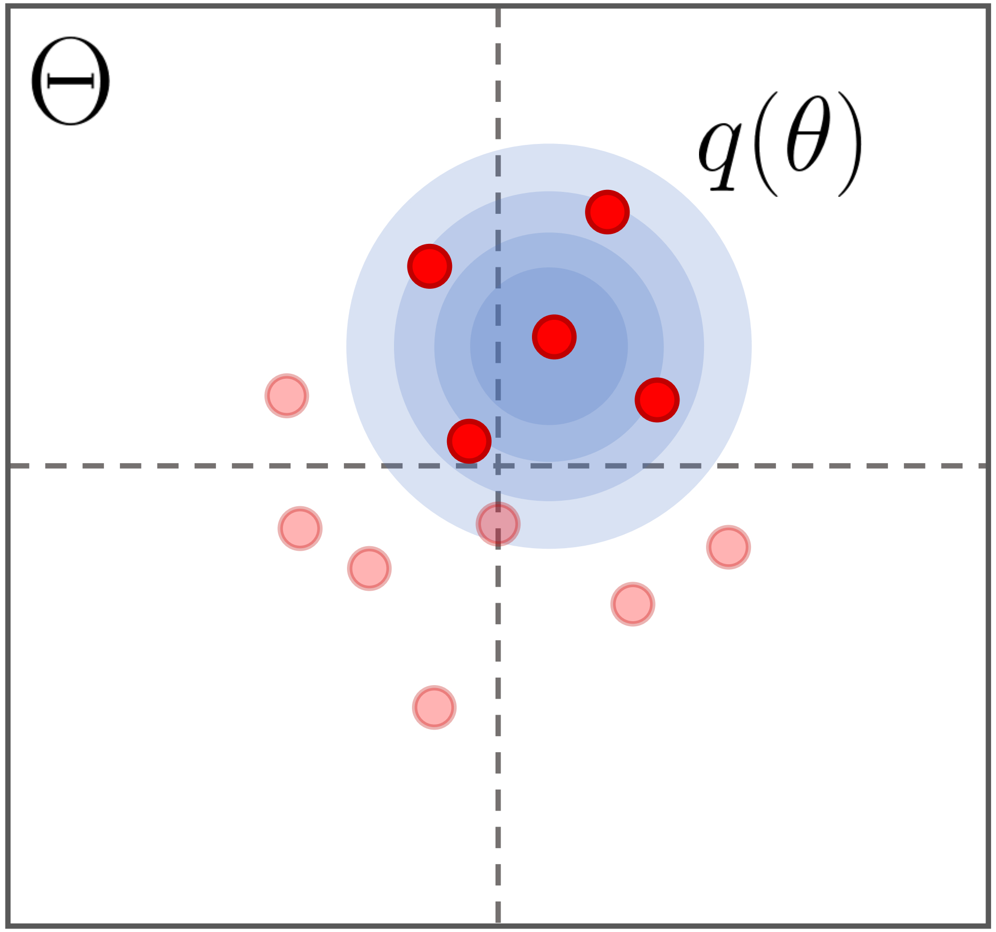
Cross-Entropy Method



Policy Gradient

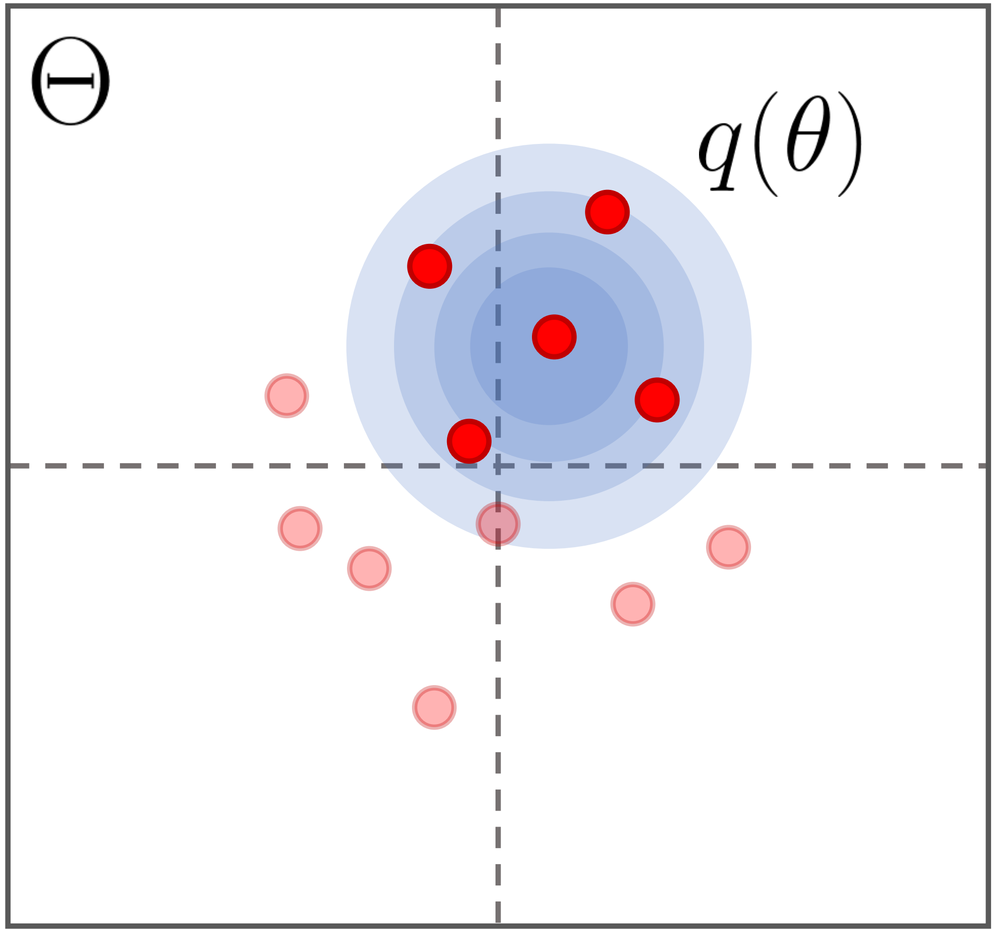


Evolutionary Strategies



$$J(q) = \underline{\mathbb{E}_{\theta \sim q(\theta)}} \left[J(\pi_\theta) \right]$$

Evolutionary Strategies



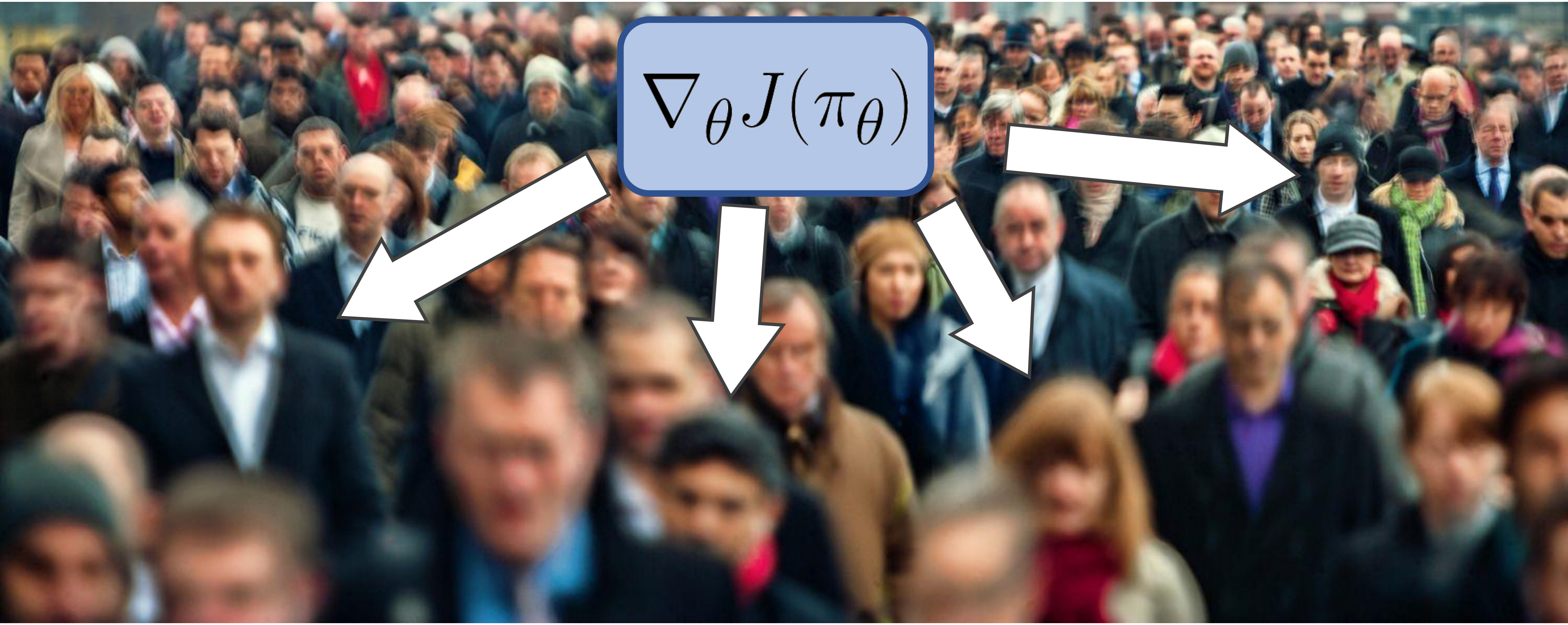
$$J(q) = \mathbb{E}_{\theta \sim q(\theta)}\left[J(\pi_\theta)\right]$$

$$\nabla_q J(q) = \mathbb{E}_{\theta \sim q(\theta)}\left[J(\pi_\theta)\nabla_q \log q(\theta)\right]$$

evolutionary strategy

Evolution is doing gradient ascent!

Evolutionary Strategies



Evolutionary Strategies

Cross-Entropy Method:

- Optimize distribution over parameters

Policy Gradient:

- Optimize distribution over actions

Summary

- Policy Gradient
- Derivation
- Variance Reduction
- Applications
- General View of PG