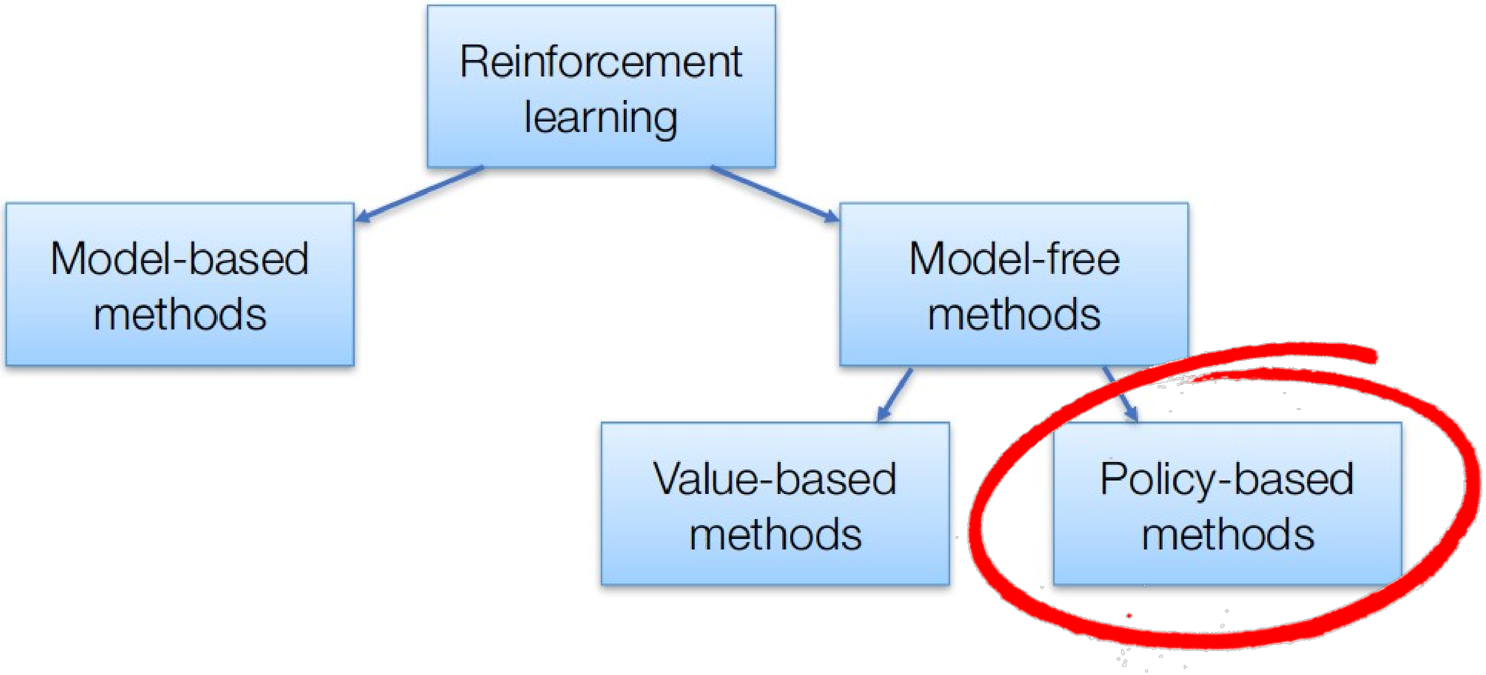
# Policy Gradient methods for RL

**Goal**: Find an optimal policy that maximises expected return in a given environment

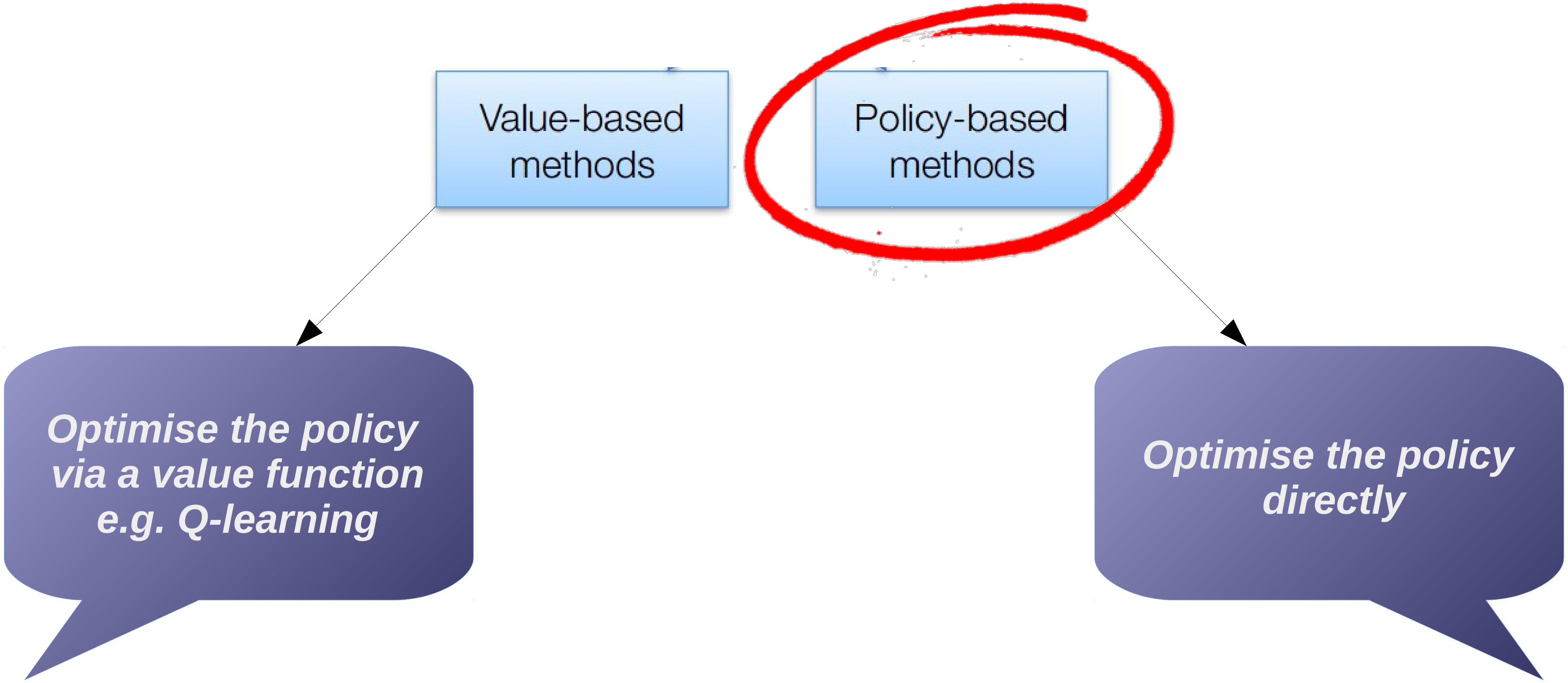
# Policy Gradient methods for RL

**Goal**: Find an optimal policy that maximises expected return in a given environment



# Policy Gradient methods for RL

**Goal**: Find an optimal policy that maximises expected return in a given environment



# Policy Gradient methods for RL

**Goal**: Find an optimal policy that maximises expected return in a given environment

Pros & cons of policy gradient methods (vs Q-learning)

- ➕ Learns policy directly (good when value function is very complex)
- ➕ Can learn stochastic policies
- ➕ Works with continuous action spaces
- ➕ (Often) faster convergence

- ➖ Can converge to local optima
- ➖ High variance of rewards leads to low sample efficiency
- ➖ Less stable performance

# Policy Gradient methods for RL

**Goal**: Find an optimal policy that maximises expected return in a given environment

Definitions

$\tau = (s_0, a_0, s_1, a_1, ..., s_{H-1}, a_{H-1}, s_H)$ is a trajectory

$\pi_\theta : \mathcal{S} \times \mathcal{A} \to [0, 1]$ is a stochastic policy with parameters $\theta$

$R(s_t, a_t)$ is the expected reward for taking action $a_t$ in state $s_t$

$\mathcal{R}(\tau) = \sum_{t=0}^{T-1} R(s_t, a_t)$ is the expected reward over a trajectory $\tau$

# Policy Gradient methods for RL

**Goal**: Find an optimal policy that maximises expected return in a given environment

# Policy Gradient methods for RL

**Goal**: Find an optimal policy that maximises expected return in a given environment

In other words…

Find a policy π that **maximises** the utility function

$$U(\theta) = \mathbb{E}_{\pi_\theta}\left[\mathcal{R}(\tau)\right]$$

# Policy Gradient methods for RL

**Goal**: Find an optimal policy that maximises expected return in a given environment

In other words…

Find a policy π that **maximises** the utility function

$$U(\theta) = \mathbb{E}_{\pi_\theta}\left[\mathcal{R}(\tau)\right]$$

Use gradient ascent:

$$\theta_{n+1} = \theta_n + \alpha \cdot \nabla_\theta U(\theta_n)$$

# Policy Gradient methods for RL

**Goal**: Find an optimal policy that maximises expected return in a given environment

In other words…

Find a policy π that **maximises** the utility function

$$U(\theta) = \mathbb{E}_{\pi_\theta}\left[\mathcal{R}(\tau)\right]$$

Use gradient ascent:

$$\theta_{n+1} = \theta_n + \alpha \cdot \nabla_\theta U(\theta_n)$$

with

$$\nabla_\theta U(\theta_n) = \mathbb{E}_{\pi_{\theta_n}}\left[\sum_{t=1}^{T-1}\left(\nabla_\theta\left[\log \pi_{\theta_n}(a_t|s_t)\right]\right)\mathcal{R}(\tau)\right]$$

$$\nabla_\theta U(\theta_n) = \mathbb{E}_{\pi_{\theta_n}}\left[\sum_{t=1}^{T-1}\left(\nabla_\theta\left[\log \pi_{\theta_n}(a_t|s_t)\right] Q(s_t, a_t)\right)\right]$$

# Policy Gradient methods for RL

**Goal**: Find an optimal policy that maximises expected return in a given environment

In other words…

Find a policy π that **maximises** the utility function

$$U(\theta) = \mathbb{E}_{\pi_\theta}\left[\mathcal{R}(\tau)\right]$$

Use gradient ascent:

$$\theta_{n+1} = \theta_n + \alpha \cdot \nabla_\theta U(\theta_n)$$

with

$$\nabla_\theta U(\theta_n) = \mathbb{E}_{\pi_{\theta_n}}\left[\sum_{t=1}^{T-1}\left(\nabla_\theta\left[\log \pi_{\theta_n}(a_t|s_t)\right]\right)\mathcal{R}(\tau)\right]$$

$$\nabla_\theta U(\theta_n) = \mathbb{E}_{\pi_{\theta_n}}\left[\sum_{t=1}^{T-1}\left(\nabla_\theta\left[\log \pi_{\theta_n}(a_t|s_t)\right] Q(s_t, a_t)\right)\right]$$

**Note**

$\mathcal{R}(\tau)$ isn't in the sum…

…but $Q(s_t, a_t)$ is

# Policy Gradient methods for RL

**Goal**: Find an optimal policy that maximises expected return in a given environment

Imagine we're performing ascent descent **on the policy directly**

# Policy Gradient methods for RL

**Goal**: Find an optimal policy that maximises expected return in a given environment

Imagine we're performing gradient descent **on the policy directly**

$$\theta_{n+1} = \theta_n + \alpha \cdot \nabla_\theta \pi_{\theta_n}(a|s)$$

# Policy Gradient methods for RL

**Goal**: Find an optimal policy that maximises expected return in a given environment

Imagine we're performing gradient descent **on the policy directly**

$$\cancel{\theta_{n+1} = \theta_n + \alpha \cdot \nabla_\theta \pi_{\theta_n}(a|s)}$$

$$\theta_{n+1} = \theta_n + \alpha \cdot \nabla_\theta \left[\pi_{\theta_n}(a|s)\right] \mathcal{R}(\tau)$$

# Policy Gradient methods for RL

**Goal**: Find an optimal policy that maximises expected return in a given environment

Imagine we're performing gradient descent **on the policy directly**

$$\cancel{\theta_{n+1} = \theta_n + \alpha \cdot \nabla_\theta \pi_{\theta_n}(a|s)}$$

$$\cancel{\theta_{n+1} = \theta_n + \alpha \cdot \nabla_\theta \left[\pi_{\theta_n}(a|s)\right] \mathcal{R}(\tau)}$$

$$\theta_{n+1} = \theta_n + \alpha \cdot \sum_{t=1}^{T-1} \left(\nabla_\theta \left[\pi_{\theta_n}(a_t|s_t)\right] \mathcal{R}(\tau)\right)$$

# Policy Gradient methods for RL

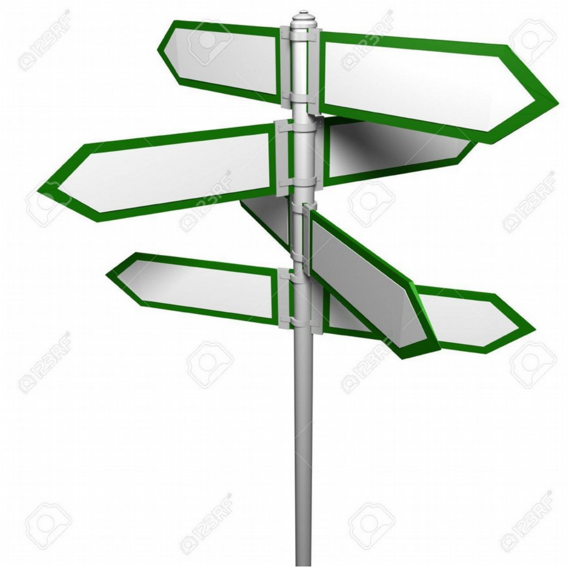
**Goal**: Find an optimal policy that maximises expected return in a given environment

Imagine we're performing gradient descent **on the policy directly**

$$\cancel{\theta_{n+1} = \theta_n + \alpha \cdot \nabla_\theta \pi_{\theta_n}(a|s)}$$

$$\cancel{\theta_{n+1} = \theta_n + \alpha \cdot \nabla_\theta \left[\pi_{\theta_n}(a|s)\right] \mathcal{R}(\tau)}$$

$$\theta_{n+1} = \theta_n + \alpha \cdot \sum_{t=1}^{T-1} \left(\nabla_\theta \left[\pi_{\theta_n}(a_t|s_t)\right] \mathcal{R}(\tau)\right)$$



# Policy Gradient methods for RL

**Goal**: Find an optimal policy that maximises expected return in a given environment

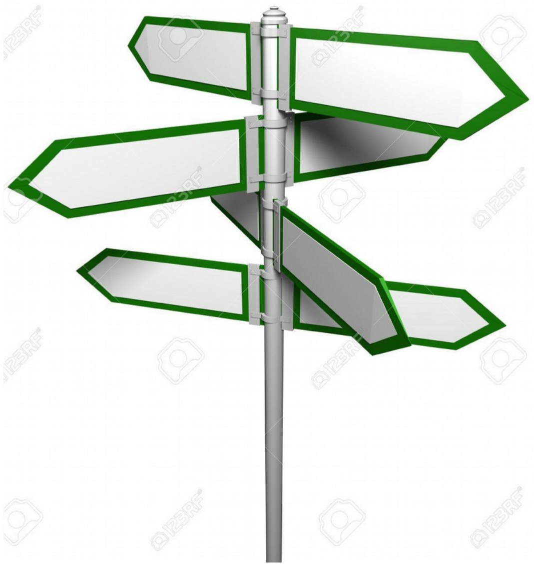
Imagine we're performing gradient descent **on the policy directly**

$$\cancel{\theta_{n+1} = \theta_n + \alpha \cdot \nabla_\theta \pi_{\theta_n}(a|s)}$$

$$\cancel{\theta_{n+1} = \theta_n + \alpha \cdot \nabla_\theta \left[\pi_{\theta_n}(a|s)\right] \mathcal{R}(\tau)}$$

$$\cancel{\theta_{n+1} = \theta_n + \alpha \cdot \sum_{t=1}^{T-1} \left(\nabla_\theta \left[\pi_{\theta_n}(a_t|s_t)\right] \mathcal{R}(\tau)\right)}$$

$$\theta_{n+1} = \theta_n + \alpha \cdot \sum_{t=1}^{T-1} \left(\nabla_\theta \left[\pi_{\theta_n}(a_t|s_t)\right] \frac{\mathcal{R}(\tau)}{\pi_{\theta_n}(a_t|s_t)}\right)$$



# Policy Gradient methods for RL

**Goal**: Find an optimal policy that maximises expected return in a given environment

Imagine we're performing gradient descent **on the policy directly**

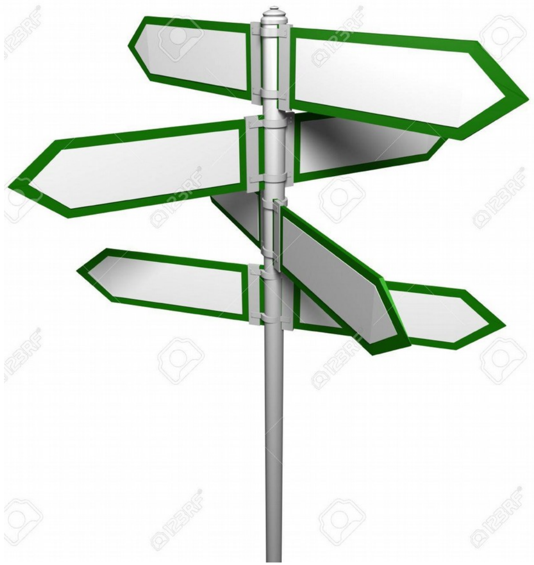
$$\cancel{\theta_{n+1} = \theta_n + \alpha \cdot \nabla_\theta \pi_{\theta_n}(a|s)}$$

$$\cancel{\theta_{n+1} = \theta_n + \alpha \cdot \nabla_\theta \left[\pi_{\theta_n}(a|s)\right] \mathcal{R}(\tau)}$$

$$\cancel{\theta_{n+1} = \theta_n + \alpha \cdot \sum_{t=1}^{T-1} \left(\nabla_\theta \left[\pi_{\theta_n}(a_t|s_t)\right] \mathcal{R}(\tau)\right)}$$

$$\theta_{n+1} = \theta_n + \alpha \cdot \sum_{t=1}^{T-1} \left(\nabla_\theta \left[\pi_{\theta_n}(a_t|s_t)\right] \frac{\mathcal{R}(\tau)}{\pi_{\theta_n}(a_t|s_t)}\right)$$

$$= \theta_n + \alpha \cdot \sum_{t=1}^{T-1} \left(\nabla_\theta \left[\log \pi_{\theta_n}(a_t|s_t)\right]\right) \mathcal{R}(\tau)$$



# Policy Gradient methods for RL

**Goal**: Find an optimal policy that maximises expected return in a given environment

$$\theta_{n+1} = \theta_n + \alpha \cdot \sum_{t=1}^{T-1} \left( \nabla_\theta \left[ \log \pi_{\theta_n} \left( a_t | s_t \right) \right] \right) \mathcal{R}(\tau)$$

# Policy Gradient methods for RL

**Goal**: Find an optimal policy that maximises expected return in a given environment

$$\theta_{n+1} = \theta_n + \alpha \cdot \sum_{t=1}^{T-1} \left( \nabla_\theta \left[ \log \pi_{\theta_n} \left( a_t | s_t \right) \right] \right) \mathcal{R}(\tau)$$

**Intuition**: Update the policy parameters using trajectories they generate:

- *Weight updates by the 'good-ness' of the trajectory*
- *Scale updates down by the likelihood of that trajectory*

# Policy Gradient methods for RL

**Goal**: Find an optimal policy that maximises expected return in a given environment

$$\theta_{n+1} = \theta_n + \alpha \cdot \sum_{t=1}^{T-1} \left( \nabla_\theta \left[ \log \pi_{\theta_n} \left( a_t | s_t \right) \right] \right) \mathcal{R}(\tau)$$

**Intuition**: Update the policy parameters using trajectories they generate:

- *Weight updates by the 'good-ness' of the trajectory*
- *Scale updates down by the likelihood of that trajectory*

$$\boxed{\theta_{n+1} = \theta_n + \alpha \cdot \nabla_\theta U(\theta_n)}$$

# Policy Gradient methods for RL

**Goal**: Find an optimal policy that maximises expected return in a given environment

$$\theta_{n+1} = \theta_n + \alpha \cdot \sum_{t=1}^{T-1} \left(\nabla_\theta \left[\log \pi_{\theta_n}(a_t|s_t)\right]\right)\mathcal{R}(\tau)$$

**Intuition**: Update the policy parameters using trajectories they generate:

- *Weight updates by the ‘good-ness’ of the trajectory*
- *Scale updates down by the likelihood of that trajectory*

$$\boxed{\theta_{n+1} = \theta_n + \alpha \cdot \nabla_\theta U(\theta_n)}$$

$$\theta_{n+1} = \theta_n + \alpha \cdot \nabla_\theta \left[\sum_{t=1}^{T-1} \left(\log \pi_{\theta_n}(a_t|s_t)\right)\mathcal{R}(\tau)\right]$$

# Policy Gradient methods for RL

**Goal**: Find an optimal policy that maximises expected return in a given environment

$$\theta_{n+1} = \theta_n + \alpha \cdot \sum_{t=1}^{T-1} \left( \nabla_\theta \left[ \log \pi_{\theta_n} \left( a_t | s_t \right) \right] \right) \mathcal{R}(\tau)$$

**Intuition**: Update the policy parameters using trajectories they generate:

- *Weight updates by the 'good-ness' of the trajectory*
- *Scale updates down by the likelihood of that trajectory*

$$\boxed{\theta_{n+1} = \theta_n + \alpha \cdot \nabla_\theta U(\theta_n)}$$

$$\theta_{n+1} = \theta_n + \alpha \cdot \nabla_\theta \left[ \sum_{t=1}^{T-1} \left( \log \pi_{\theta_n} \left( a_t | s_t \right) \right) \mathcal{R}(\tau) \right]$$

$$\nabla_\theta U(\theta_n) = \sum_{t=1}^{T-1} \left( \nabla_\theta \left[ \log \pi_{\theta_n} \left( a_t | s_t \right) \right] \right) \mathcal{R}(\tau)$$

# Policy Gradient methods for RL

**Goal**: Find an optimal policy that maximises expected return in a given environment

$$\nabla_\theta U(\theta_n) = \sum_{t=1}^{T-1} \left(\nabla_\theta \left[\log \pi_{\theta_n} \left(a_t | s_t\right)\right]\right) \mathcal{R}(\tau)$$

# Policy Gradient methods for RL

**Goal**: Find an optimal policy that maximises expected return in a given environment

$$\nabla_\theta U(\theta_n) = \sum_{t=1}^{T-1} \left( \nabla_\theta \left[ \log \pi_{\theta_n} \left( a_t | s_t \right) \right] \right) \mathcal{R}(\tau)$$

$$\nabla_\theta U(\theta_n) = \mathbb{E}_{\pi_{\theta_n}} \left[ \sum_{t=1}^{T-1} \left( \nabla_\theta \left[ \log \pi_{\theta_n} \left( a_t | s_t \right) \right] \right) \mathcal{R}(\tau) \right]$$

# Policy Gradient methods for RL

**Goal**: Find an optimal policy that maximises expected return in a given environment

$$\nabla_\theta U(\theta_n) = \sum_{t=1}^{T-1} \left(\nabla_\theta \left[\log \pi_{\theta_n} \left(a_t|s_t\right)\right]\right) \mathcal{R}(\tau)$$

$$\nabla_\theta U(\theta_n) = \mathbb{E}_{\pi_{\theta_n}} \left[\sum_{t=1}^{T-1} \left(\nabla_\theta \left[\log \pi_{\theta_n} \left(a_t|s_t\right)\right]\right) \mathcal{R}(\tau)\right]$$

$$\boxed{\nabla_\theta U(\theta_n) = \mathbb{E}_{\pi_{\theta_n}} \left[\sum_{t=1}^{T-1} \left(\nabla_\theta \left[\log \pi_{\theta_n} \left(a_t|s_t\right)\right]\right) \mathcal{R}(\tau)\right]}$$

# Policy Gradient methods for RL

**Goal**: Find an optimal policy that maximises expected return in a given environment

$$\nabla_\theta U(\theta_n) = \sum_{t=1}^{T-1} \left(\nabla_\theta \left[\log \pi_{\theta_n}\left(a_t|s_t\right)\right]\right)\mathcal{R}(\tau)$$

$$\nabla_\theta U(\theta_n) = \mathbb{E}_{\pi_{\theta_n}}\left[\sum_{t=1}^{T-1} \left(\nabla_\theta \left[\log \pi_{\theta_n}\left(a_t|s_t\right)\right]\right)\mathcal{R}(\tau)\right]$$

$$\boxed{\nabla_\theta U(\theta_n) = \mathbb{E}_{\pi_{\theta_n}}\left[\sum_{t=1}^{T-1} \left(\nabla_\theta \left[\log \pi_{\theta_n}\left(a_t|s_t\right)\right]\right)\mathcal{R}(\tau)\right]}$$

$$\nabla_\theta U(\theta_n) = \frac{1}{m}\sum_{i=1}^{m}\left[\sum_{t=1}^{T-1} \left(\nabla_\theta \left[\log \pi_{\theta_n}\left(a_t^{(i)}|s_t^{(i)}\right)\right]\right)\mathcal{R}(\tau^{(i)})\right]$$

# Policy Gradient methods for RL

**Goal**: Find an optimal policy that maximises expected return in a given environment

Looking back, if we'd had access to $Q(s, a)$...

# Policy Gradient methods for RL

**Goal**: Find an optimal policy that maximises expected return in a given environment

Looking back, if we'd had access to $Q(s,a)$...

$$\cancel{\theta_{n+1} = \theta_n + \alpha \cdot \nabla_\theta \pi_{\theta_n}(a|s)}$$

$$\theta_{n+1} = \theta_n + \alpha \cdot \nabla_\theta \left[\pi_{\theta_n}(a|s)\right] Q(s,a)$$

# Policy Gradient methods for RL

**Goal**: Find an optimal policy that maximises expected return in a given environment

Looking back, if we'd had access to $Q(s,a)$...

$$\cancel{\theta_{n+1} = \theta_n + \alpha \cdot \nabla_\theta \pi_{\theta_n}(a|s)}$$

$$\theta_{n+1} = \theta_n + \alpha \cdot \nabla_\theta \left[\pi_{\theta_n}(a|s)\right] Q(s,a)$$

$$\vdots$$

$$\nabla_\theta U(\theta_n) = \mathbb{E}_{\pi_{\theta_n}} \left[\sum_{t=1}^{T-1} \left(\nabla_\theta \left[\log \pi_{\theta_n}(a_t|s_t)\right] Q(s_t, a_t)\right)\right]$$