In the Name of God

Iran University
of Science and
Technology

Reinforcement Learning in Control

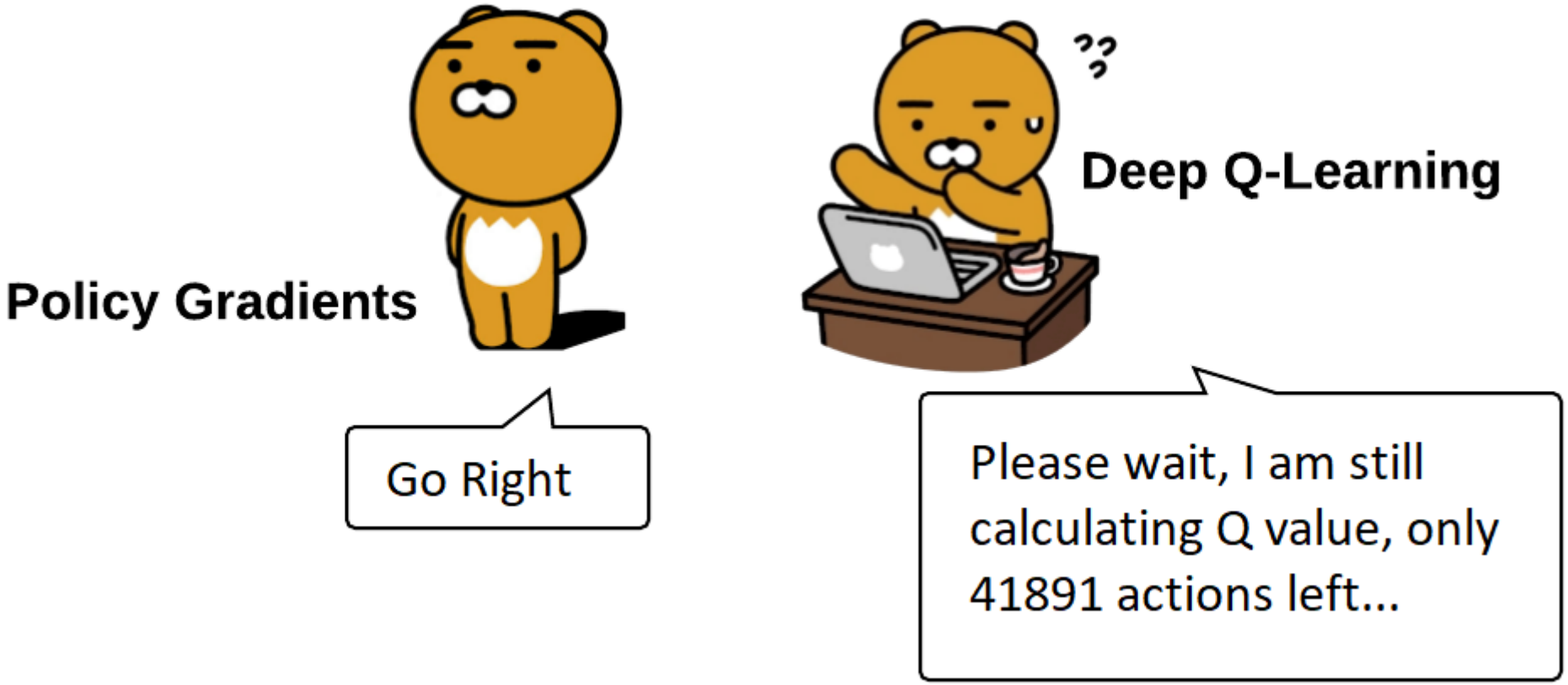
Dr. Saeed Shamaghdari

Electrical Engineering Department
Control Group

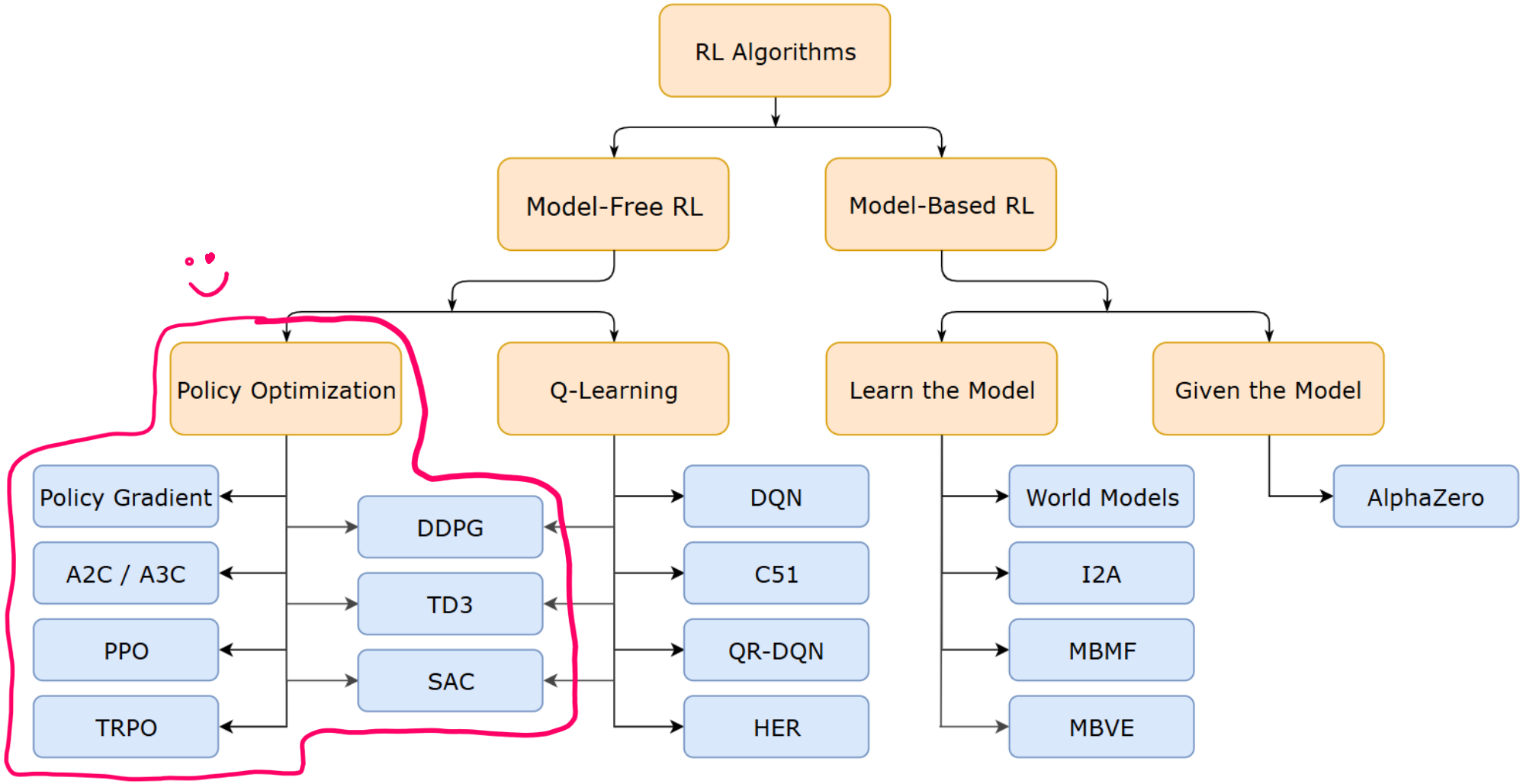
Fall 2025 | 4041

Policy Gradient Methods

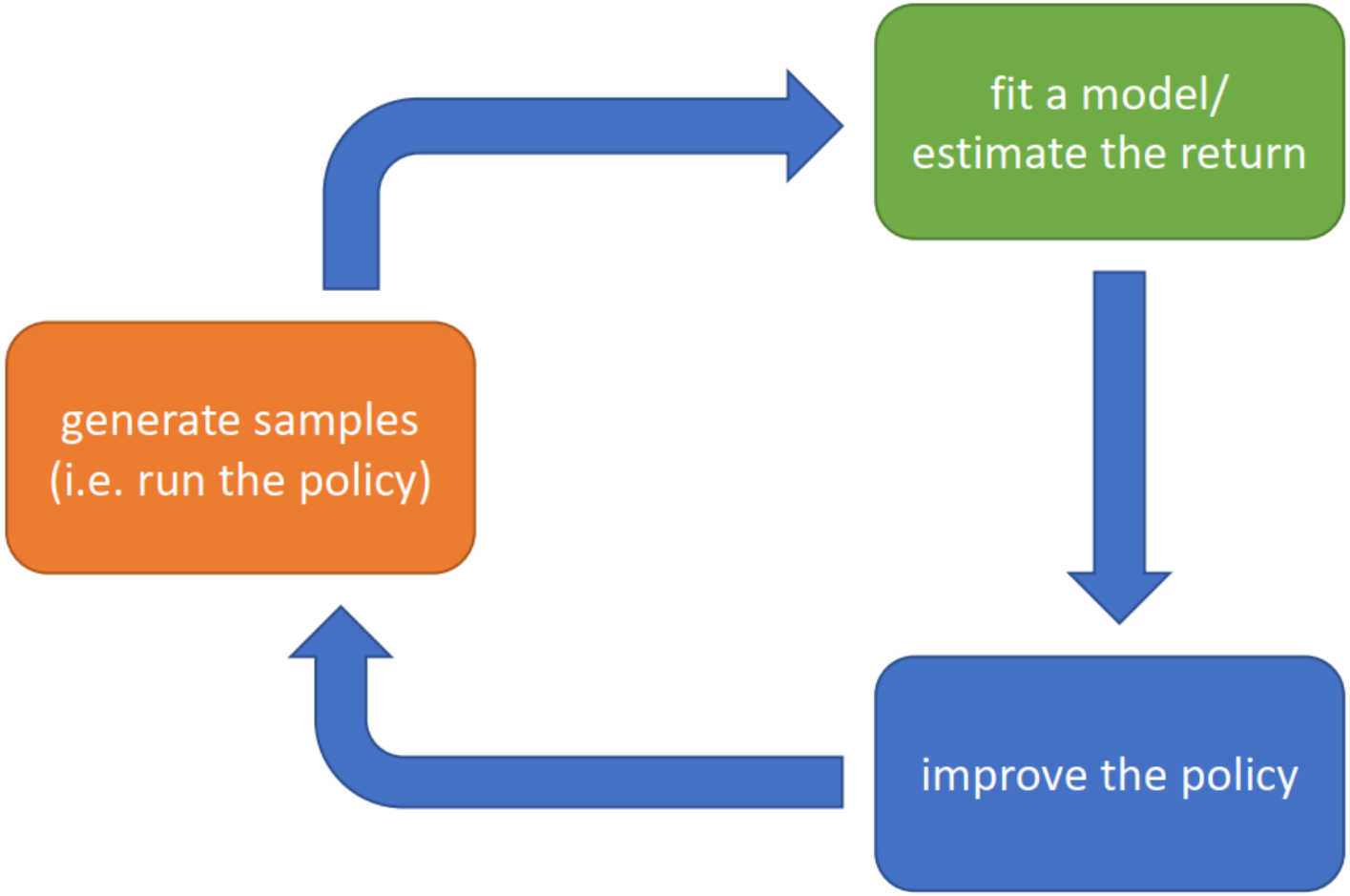
Quick Review

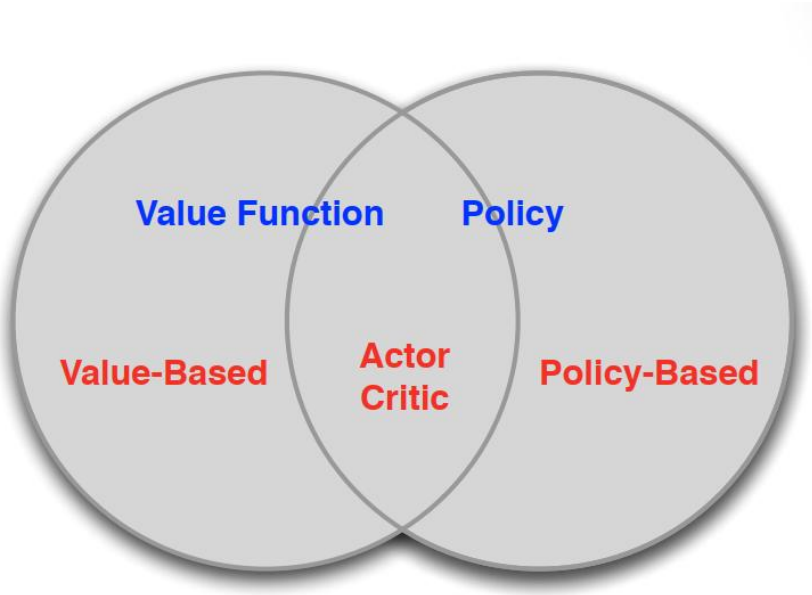


RL Algorithms

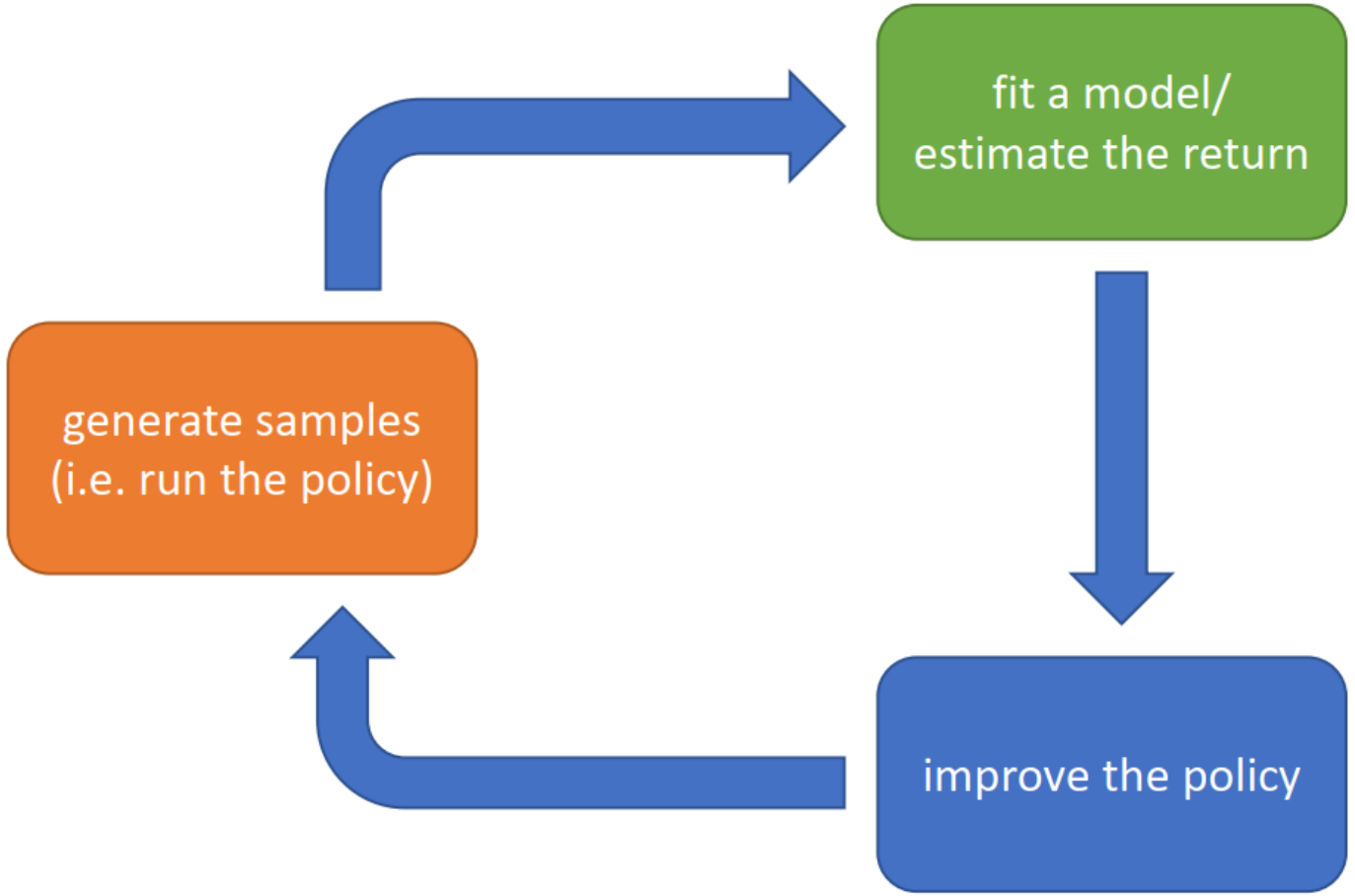


Anatomy of RL Algorithms





Anatomy of RL Algorithms: Value Function Based Methods



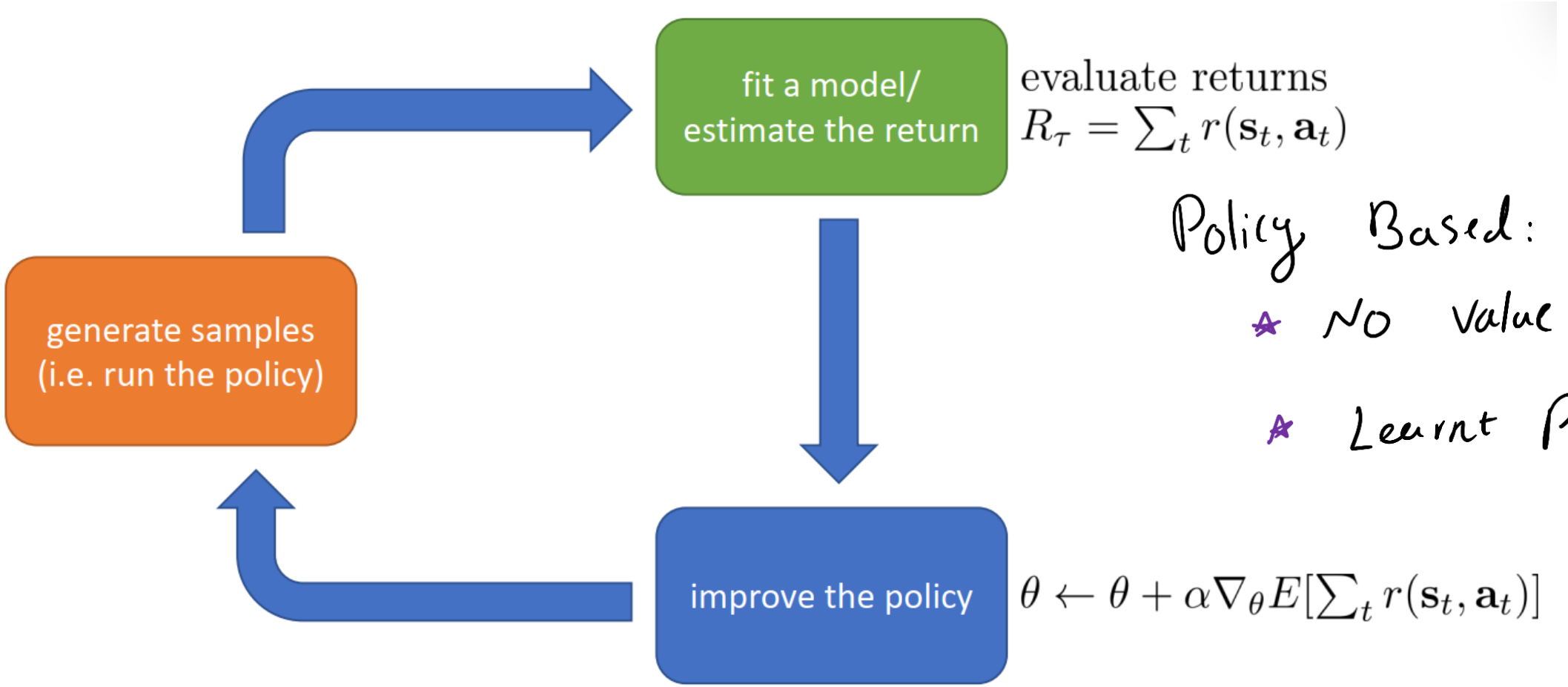
fit $V(\mathbf{s})$ or $Q(\mathbf{s}, \mathbf{a})$

Value Based:

- Learnt Value Function
- Implicit Policy (e.g ε-greedy)

set $\pi(\mathbf{s}) = \arg\max_{\mathbf{a}} Q(\mathbf{s}, \mathbf{a})$

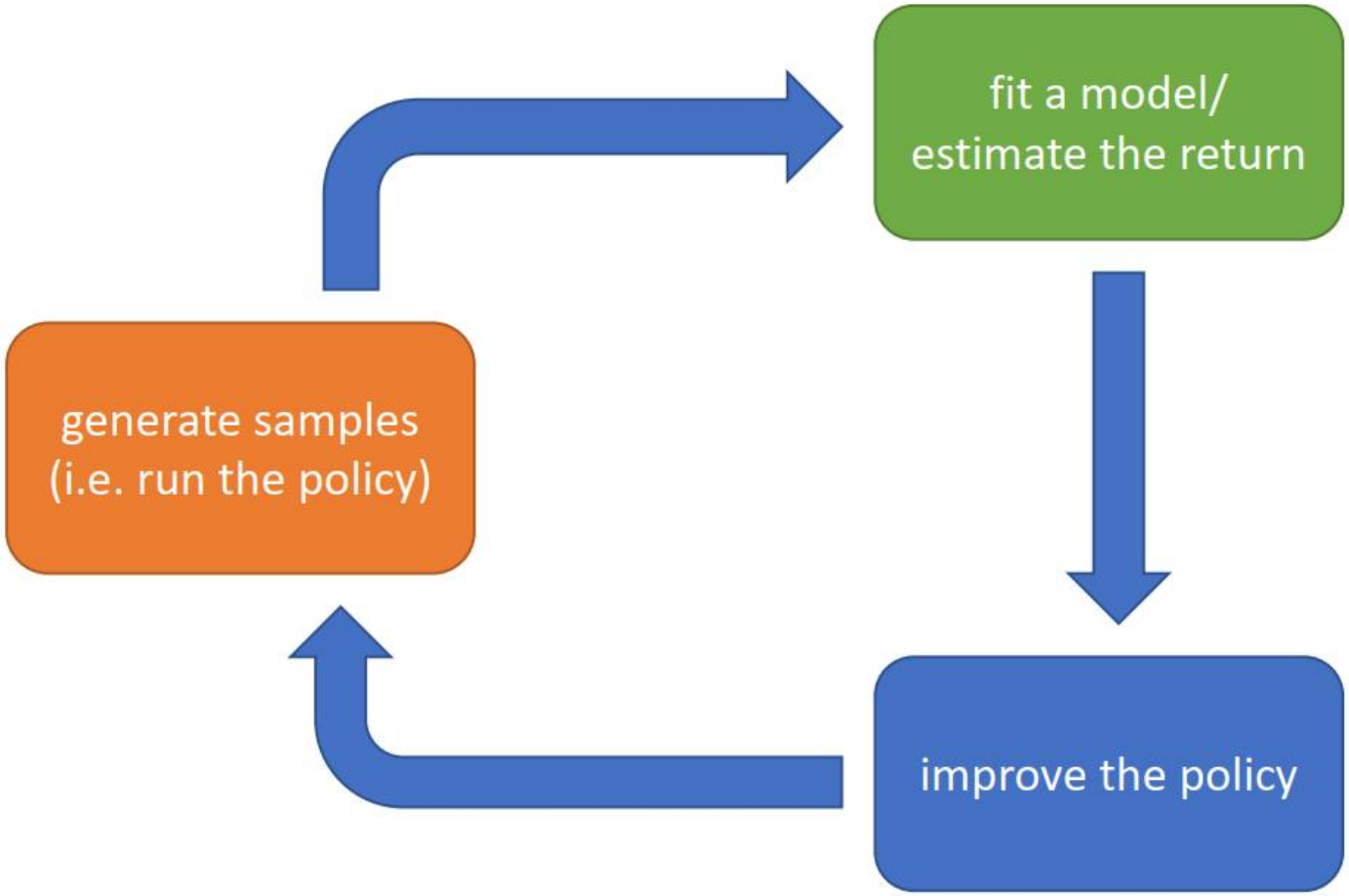
Anatomy of RL Algorithms: Policy Gradients



Policy Based:

- No value function
- Learnt Policy

Anatomy of RL Algorithms: Actor-Critic



fit $V(\mathbf{s})$ or $Q(\mathbf{s}, \mathbf{a})$

$\theta \leftarrow \theta + \alpha \nabla_\theta E[Q(\mathbf{s}_t, \mathbf{a}_t)]$

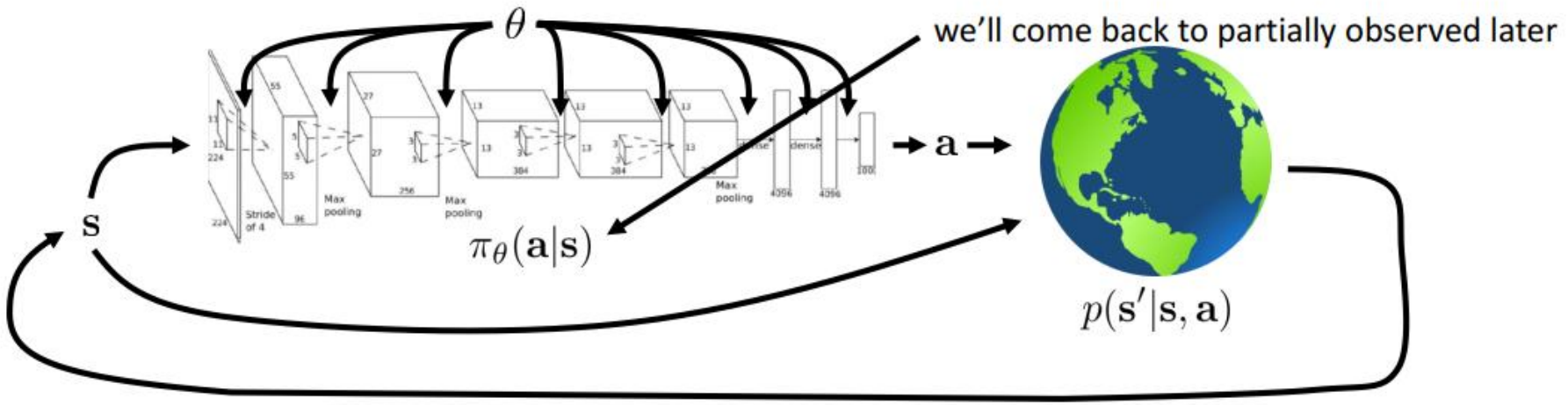
Actor-Critic:

- Learnt Value Function
- Learnt Policy

Policy Gradients

In Action-Value methods $\pi^*(s) = \arg\max_a Q^*(s,a)$

This Chapter: Learn a *parameterized* *policy*



Policy Gradients

Definition

We write

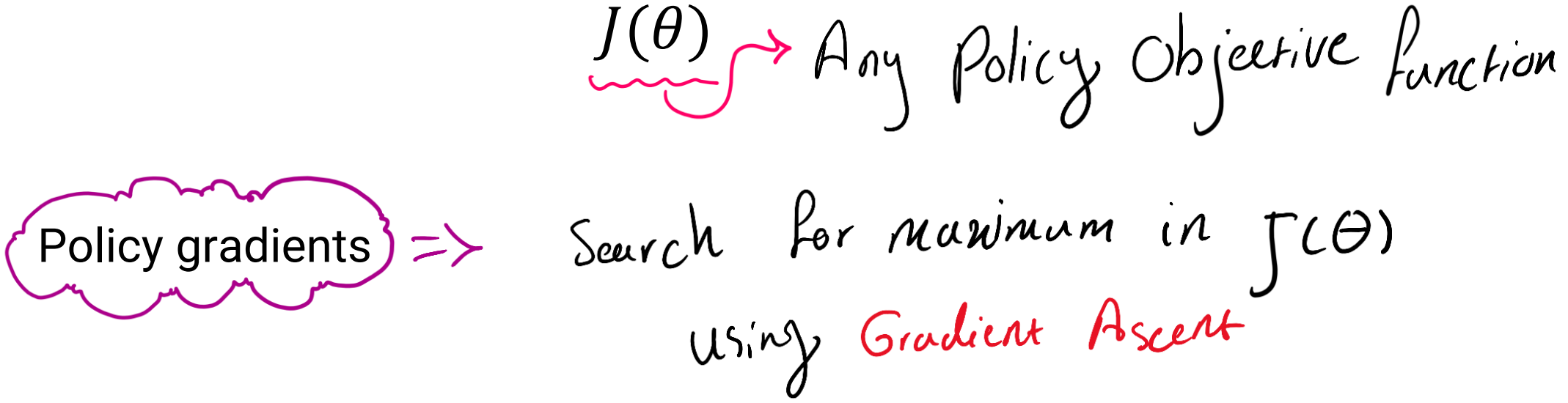
$$\pi(a|s,\boldsymbol{\theta}) = \Pr\{A_t = a \mid S_t = s, \boldsymbol{\theta}_t = \boldsymbol{\theta}\}$$

for the probability that action *a* is taken at time *t* given that the environment is in state *s* at time *t* with parameter θ.

$\boldsymbol{\theta} \in \mathbb{R}^{d'}$ → policy's parameter vector

Policy Gradients

Policy gradients are methods for learning the policy parameter based on the gradient of some scalar performance measure

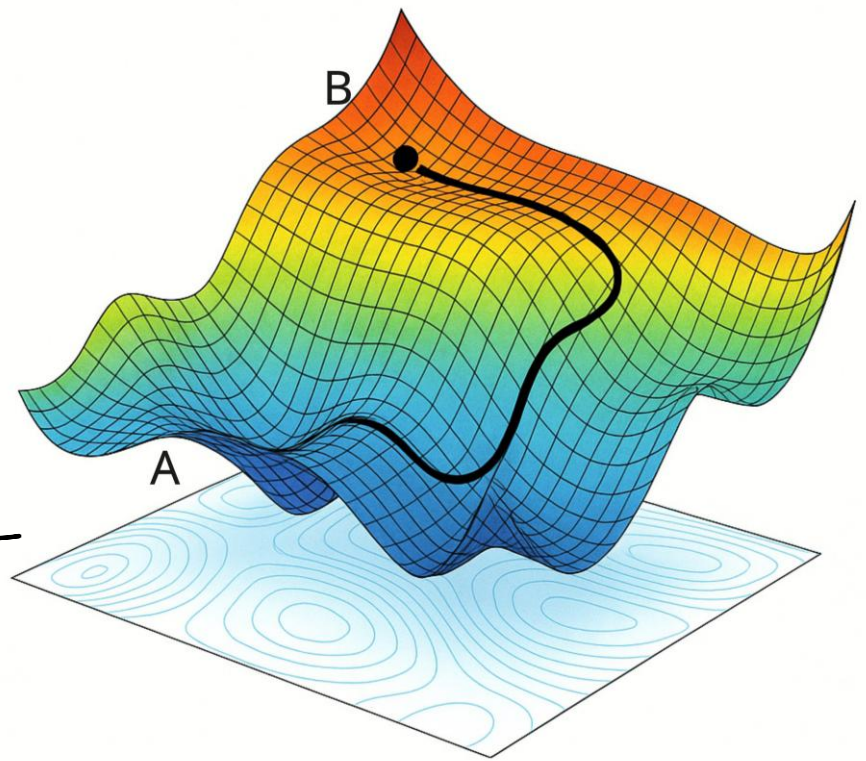


Policy Gradients

Maximize performance using *gradient ascent*

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha \widehat{\nabla J(\boldsymbol{\theta}_t)}$$

Where $\widehat{\nabla J(\theta_t)} \in \mathbb{R}^{d'}$ is a stochastic estimate whose expectation approximates the gradient of the performance measure with respect to its argument θ_t.



$\Delta\theta = \alpha \nabla_\theta J(\theta)$

$\nabla_\theta J(\theta) = \left[\frac{\partial J}{\partial \theta_1} \cdots \frac{\partial J}{\partial \theta_n} \right]^T$

Policy Gradients: Recap

Objective	Policy Gradient Methods	Actor-Critic Methods
Learn Policy Function	✓	✓
Learn Value Function	∴	✓

Policy Approximation: Soft-max in Action Preferences

Most common parameterization for discrete action spaces(and not too large):

$h(s,a,\theta) \in \mathbb{R}$ (for each action state pairs)

Parametrized numerical preferences

Actions with the ↑ h -> ↑ probabilities of being selected

$$\pi(a|s,\boldsymbol{\theta}) \doteq \frac{e^{h(s,a,\boldsymbol{\theta})}}{\sum_b e^{h(s,b,\boldsymbol{\theta})}}$$

ANN ✓

Linear ✓

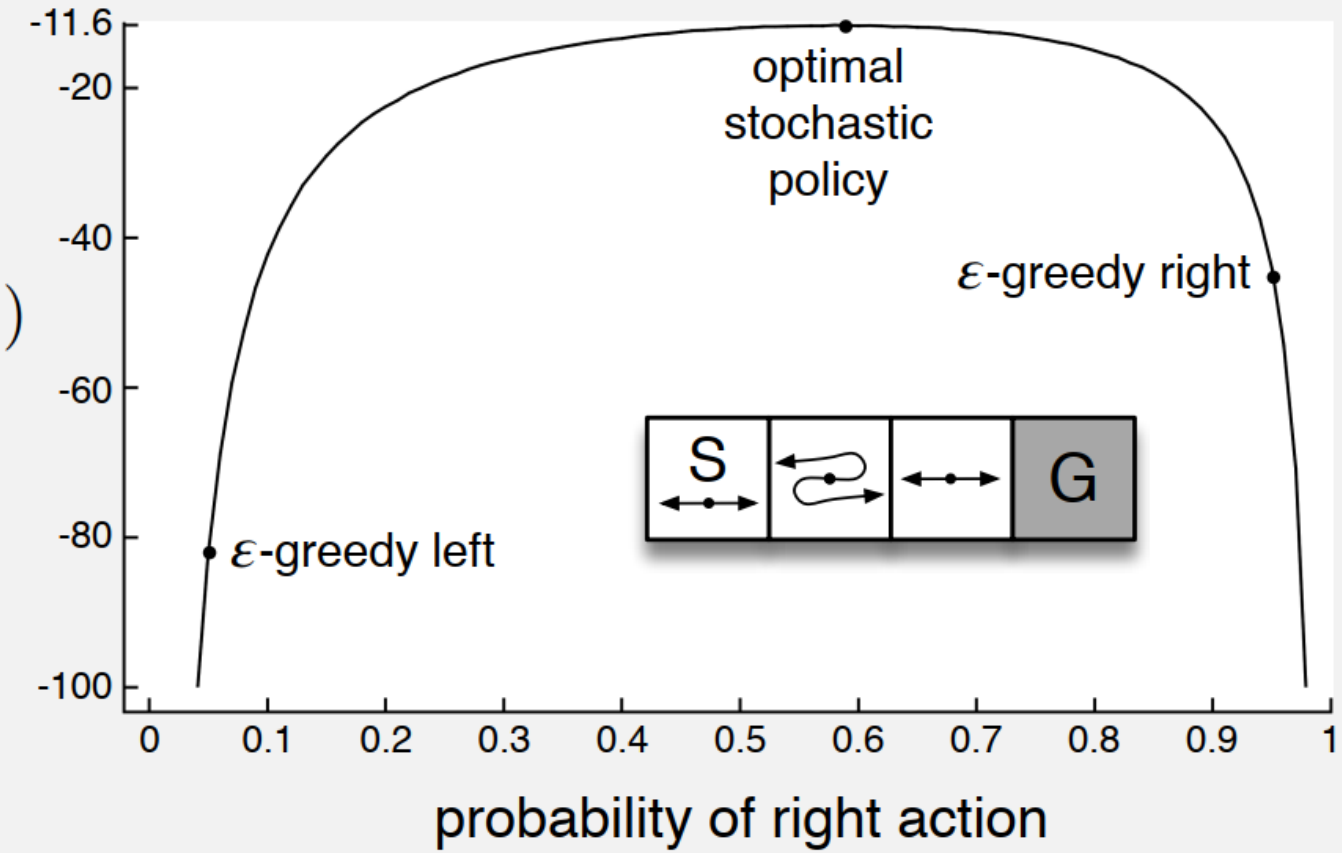
h with linear feature vectors $\mathrm{x}(\mathrm{s},\mathrm{a}) \in \mathbb{R}^{d'}$ => $h(s,a,\boldsymbol{\theta}) = \boldsymbol{\theta}^\top \mathbf{x}(s,a)$

Policy Approximation: Advantages

- ✓ Can become nearly deterministic
- ✓ Can represent arbitrary action probabilities, enabling truly stochastic policies.
- ✓ Simpler function to approximate, learning faster with better final performance.

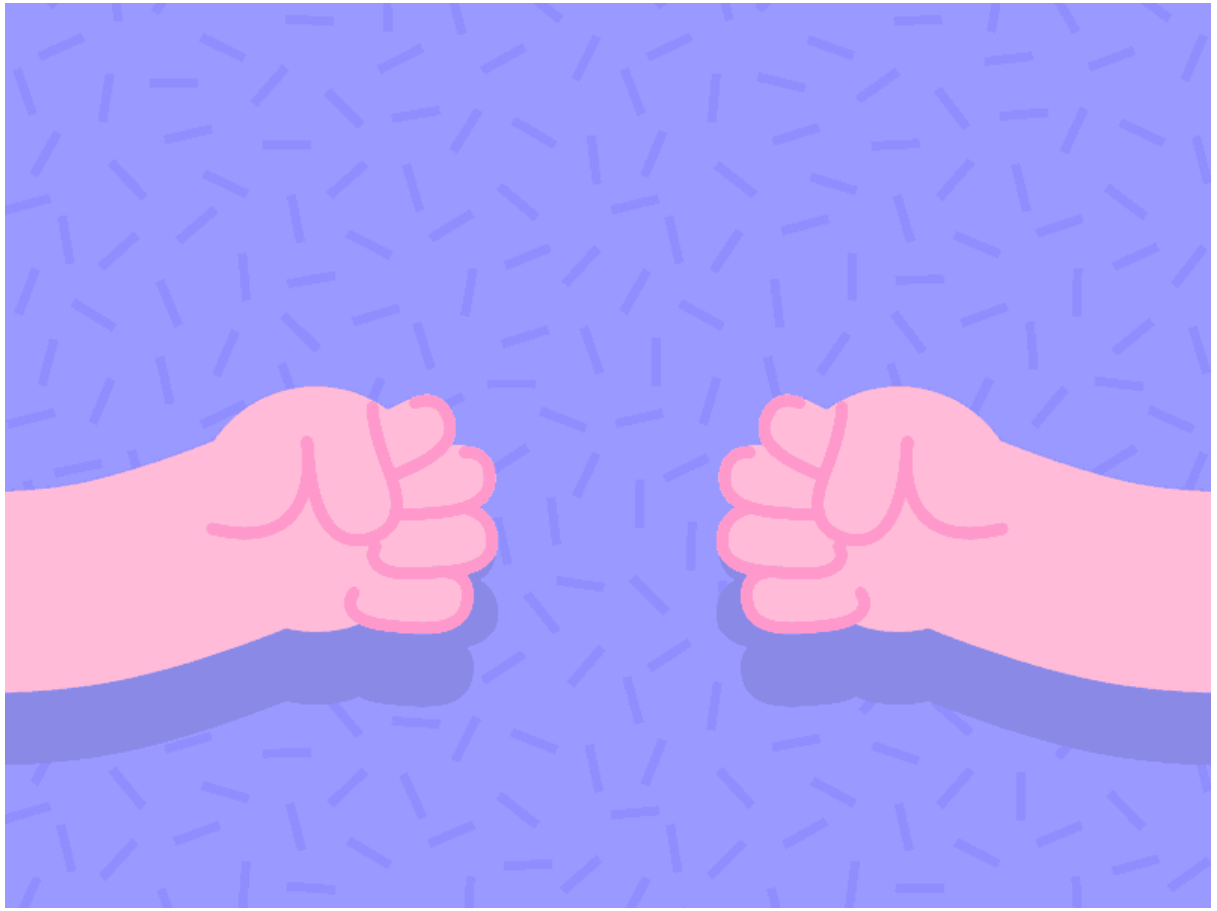
Example: Short Corridor with Switched Actions

$$J(\boldsymbol{\theta}) = v_{\pi_{\boldsymbol{\theta}}}(\mathrm{S})$$

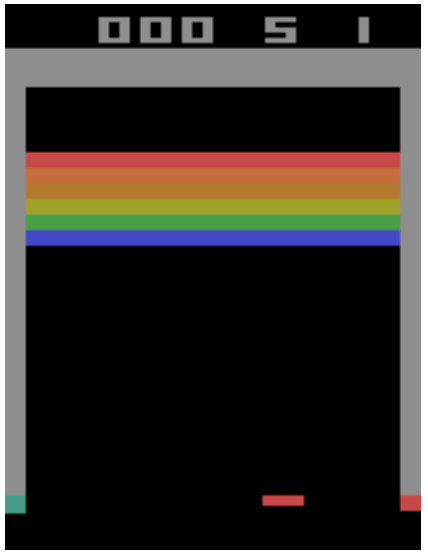


Another Example: Rock-Paper-Scissors

Deterministic Policy vs Stochastic Policy



Advantage of Policy-Based RL: Summary



- Advantages:
  - Better convergence properties
  - Effective in high-dimensional or continuous action spaces
  - Can learn stochastic policies

- Disadvantages:
  - Typically converge to a local rather than global optimum
  - Evaluating a policy is typically inefficient and high variance

The Policy Gradient Theorem

In the episodic case we define performance as ($\gamma = 1$)

$$J(\boldsymbol{\theta}) \doteq v_{\pi_{\boldsymbol{\theta}}}(s_0)$$

true value function

Policy changes affect both actions and the induced state distribution.

Q: How to compute the gradient when policy changes alter an unknown state distribution?

The Policy Gradient Theorem

Proof (Episodic Case)

To keep the notation simple, we leave it implicit in all cases that π is a function of θ, and all gradients are also implicitly with respect to θ. First note that the gradient of the state-value function can be written in terms of the action-value function as

$$\nabla v_\pi(s) = \nabla \left[\sum_a \pi(a|s) q_\pi(s,a) \right], \quad \text{for all } s \in \mathcal{S}$$

The Policy Gradient Theorem

Proof (Episodic Case)

$$
\begin{aligned}
\nabla v_\pi(s) &= \nabla \left[\sum_a \pi(a|s) q_\pi(s,a) \right], \quad \text{for all } s \in \mathcal{S} \\
&= \sum_a \Big[\nabla \pi(a|s) q_\pi(s,a) + \pi(a|s) \nabla q_\pi(s,a) \Big] \\
&= \sum_a \Big[\nabla \pi(a|s) q_\pi(s,a) + \pi(a|s) \nabla \sum_{s',r} p(s',r|s,a)\big(r + v_\pi(s')\big) \Big] \\
&= \sum_a \Big[\nabla \pi(a|s) q_\pi(s,a) + \pi(a|s) \sum_{s'} p(s'|s,a) \nabla v_\pi(s') \Big]
\end{aligned}
$$

The Policy Gradient Theorem

Proof (Episodic Case)

$$
\begin{aligned}
&= \sum_a \Big[\nabla \pi(a|s) q_\pi(s,a) + \pi(a|s) \sum_{s'} p(s'|s,a) \nabla v_\pi(s') \Big] \\
&= \sum_a \Big[\nabla \pi(a|s) q_\pi(s,a) + \pi(a|s) \sum_{s'} p(s'|s,a) && \text{(unrolling)} \\
&\qquad \sum_{a'} \big[\nabla \pi(a'|s') q_\pi(s',a') + \pi(a'|s') \sum_{s''} p(s''|s',a') \nabla v_\pi(s'') \big] \Big] \\
&= \sum_{x \in \mathcal{S}} \sum_{k=0}^{\infty} \Pr(s \to x, k, \pi) \sum_a \nabla \pi(a|x) q_\pi(x,a),
\end{aligned}
$$

The Policy Gradient Theorem

Proof (Episodic Case)

$$= \sum_{x \in \mathcal{S}} \sum_{k=0}^{\infty} \Pr(s \to x, k, \pi) \sum_{a} \nabla \pi(a|x) q_\pi(x, a),$$

after repeated unrolling, where $\Pr(s \to x, k, \pi)$ is the probability of transitioning from state *s* to state *x* in *k* steps under policy π. It is then immediate that

$$\nabla J(\boldsymbol{\theta}) = \nabla v_\pi(s_0)$$

The Policy Gradient Theorem

Proof (Episodic Case)

$$
\begin{aligned}
\nabla J(\boldsymbol{\theta}) &= \nabla v_\pi(s_0) \\
&= \sum_s \left(\sum_{k=0}^{\infty} \Pr(s_0 \to s, k, \pi) \right) \sum_a \nabla \pi(a|s) q_\pi(s,a) \\
&= \sum_s \eta(s) \sum_a \nabla \pi(a|s) q_\pi(s,a) \\
&= \sum_{s'} \eta(s') \sum_s \frac{\eta(s)}{\sum_{s'} \eta(s')} \sum_a \nabla \pi(a|s) q_\pi(s,a)
\end{aligned}
$$

The Policy Gradient Theorem

Proof (Episodic Case)

$$
\begin{aligned}
&= \sum_{s'} \eta(s') \sum_{s} \frac{\eta(s)}{\sum_{s'} \eta(s')} \sum_{a} \nabla \pi(a|s) q_\pi(s,a) \\
&= \sum_{s'} \eta(s') \sum_{s} \mu(s) \sum_{a} \nabla \pi(a|s) q_\pi(s,a) \\
&\propto \sum_{s} \mu(s) \sum_{a} \nabla \pi(a|s) q_\pi(s,a)
\end{aligned}
$$

that does *not* involve the derivative of the state distribution.

The Policy Gradient Theorem

The policy gradient theorem for the episodic case establishes that

$$\nabla J(\boldsymbol{\theta}) \propto \sum_s \mu(s) \sum_a q_\pi(s,a) \nabla \pi(a|s,\boldsymbol{\theta})$$

$$= \mathbb{E}_\pi \left[\sum_a q_\pi(S_t,a) \nabla \pi(a|S_t,\boldsymbol{\theta}) \right].$$

Stochastic gradient-ascent algorithm

$$\boldsymbol{\theta}_{t+1} \doteq \boldsymbol{\theta}_t + \alpha \sum_a \hat{q}(S_t,a,\mathbf{w}) \nabla \pi(a|S_t,\boldsymbol{\theta})$$

Our current interest is the classical REINFORCE algorithm (Willams, 1992) whose update at time *t* involves just A_t, the one action actually taken at time *t*.

REINFORCE: Monte Carlo Policy Gradient

$$\nabla J(\boldsymbol{\theta}) \propto \mathbb{E}_\pi \left[\sum_a q_\pi(S_t, a) \nabla \pi(a|S_t, \boldsymbol{\theta}) \right]$$

$$\propto \mathbb{E}_\pi \left[\sum_a \pi(a|S_t, \boldsymbol{\theta}) q_\pi(S_t, a) \frac{\nabla \pi(a|S_t, \boldsymbol{\theta})}{\pi(a|S_t, \boldsymbol{\theta})} \right]$$

$$= \mathbb{E}_\pi \left[q_\pi(S_t, A_t) \frac{\nabla \pi(A_t|S_t, \boldsymbol{\theta})}{\pi(A_t|S_t, \boldsymbol{\theta})} \right] \qquad \text{(replacing } a \text{ by the sample } A_t \sim \pi\text{)}$$

$$= \mathbb{E}_\pi \left[G_t \frac{\nabla \pi(A_t|S_t, \boldsymbol{\theta})}{\pi(A_t|S_t, \boldsymbol{\theta})} \right], \qquad \text{(because } \mathbb{E}_\pi[G_t|S_t, A_t] = q_\pi(S_t, A_t)\text{)}$$

The REINFORCE Update:

$$\boldsymbol{\theta}_{t+1} \doteq \boldsymbol{\theta}_t + \alpha G_t \frac{\nabla \pi(A_t|S_t, \boldsymbol{\theta}_t)}{\pi(A_t|S_t, \boldsymbol{\theta}_t)}$$

REINFORCE: Monte Carlo Policy Gradient

The REINFORCE Update:

$$\boldsymbol{\theta}_{t+1} \doteq \boldsymbol{\theta}_t + \alpha G_t \frac{\nabla \pi(A_t|S_t, \boldsymbol{\theta}_t)}{\pi(A_t|S_t, \boldsymbol{\theta}_t)}$$

$$\nabla \ln x = \frac{\nabla x}{x}$$

$$\nabla \ln \pi(A_t|S_t, \boldsymbol{\theta}_t) = \frac{\nabla \pi(A_t|S_t, \boldsymbol{\theta}_t)}{\pi(A_t|S_t, \boldsymbol{\theta}_t)}$$

$$\boldsymbol{\theta}_{t+1} \doteq \boldsymbol{\theta}_t + \alpha G_t \nabla \ln \pi(A_t|S_t, \boldsymbol{\theta}_t)$$

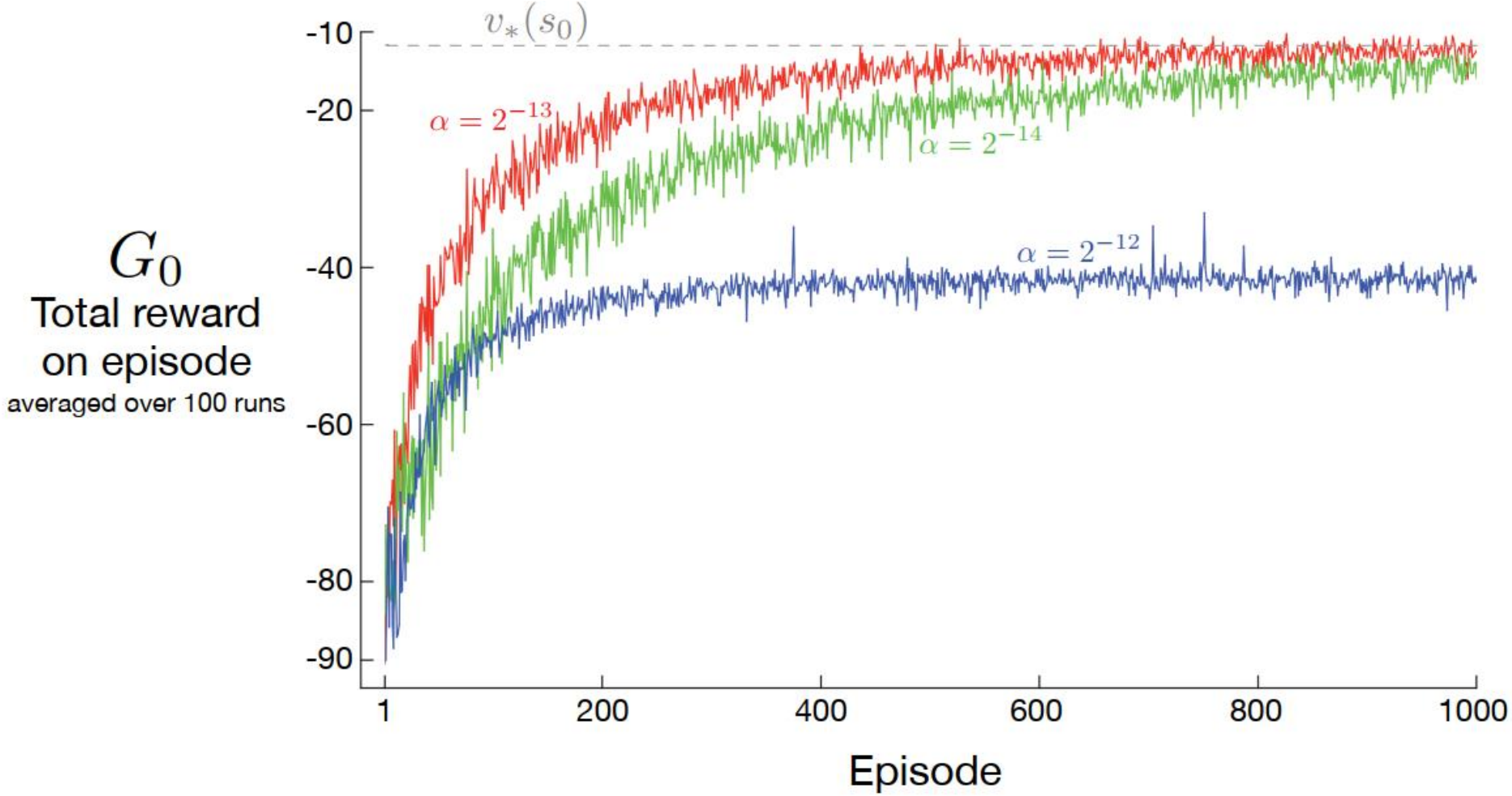
REINFORCE: Algorithm (Discounted)

REINFORCE: Monte-Carlo Policy-Gradient Control (episodic) for π_*

Input: a differentiable policy parameterization $\pi(a|s, \boldsymbol{\theta})$
Algorithm parameter: step size $\alpha > 0$
Initialize policy parameter $\boldsymbol{\theta} \in \mathbb{R}^{d'}$ (e.g., to $\mathbf{0}$)

Loop forever (for each episode):
  Generate an episode $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$, following $\pi(\cdot|\cdot, \boldsymbol{\theta})$
  Loop for each step of the episode $t = 0, 1, \ldots, T-1$:
    $G \leftarrow \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k$ (G_t)
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \gamma^t G \nabla \ln \pi(A_t|S_t, \boldsymbol{\theta})$

REINFORCE: Short Corridor Example



REINFORCE: Problem!

Good convergence properties, However …

Q: What problem does the REINFORCE method have as a Monte Carlo approach?

REINFORCE with Baseline

The policy gradient theorem can be generalized to include a comparison of the action value to an arbitrary *baseline* $b(s)$:

$$\nabla J(\boldsymbol{\theta}) \propto \sum_s \mu(s) \sum_a \Big(q_\pi(s,a) - b(s) \Big) \nabla \pi(a|s,\boldsymbol{\theta})$$

The baseline can be any function, even a random variable, as long as it does not vary with a; the equation remains valid because the subtracted quantity is zero:

$$\sum_a b(s) \nabla \pi(a|s,\boldsymbol{\theta}) \;=\; b(s) \nabla \sum_a \pi(a|s,\boldsymbol{\theta}) \;=\; b(s) \nabla 1 \;=\; 0$$

REINFORCE with Baseline: Update

$$\boldsymbol{\theta}_{t+1} \doteq \boldsymbol{\theta}_t + \alpha\Big(G_t - b(S_t)\Big)\frac{\nabla\pi(A_t|S_t,\boldsymbol{\theta}_t)}{\pi(A_t|S_t,\boldsymbol{\theta}_t)}$$

One natural choice for the baseline is an estimate of the state value

$$\hat{v}(S_t,\mathbf{w})$$

$\mathbf{w} \in \mathbb{R}^d$ is a weight vector

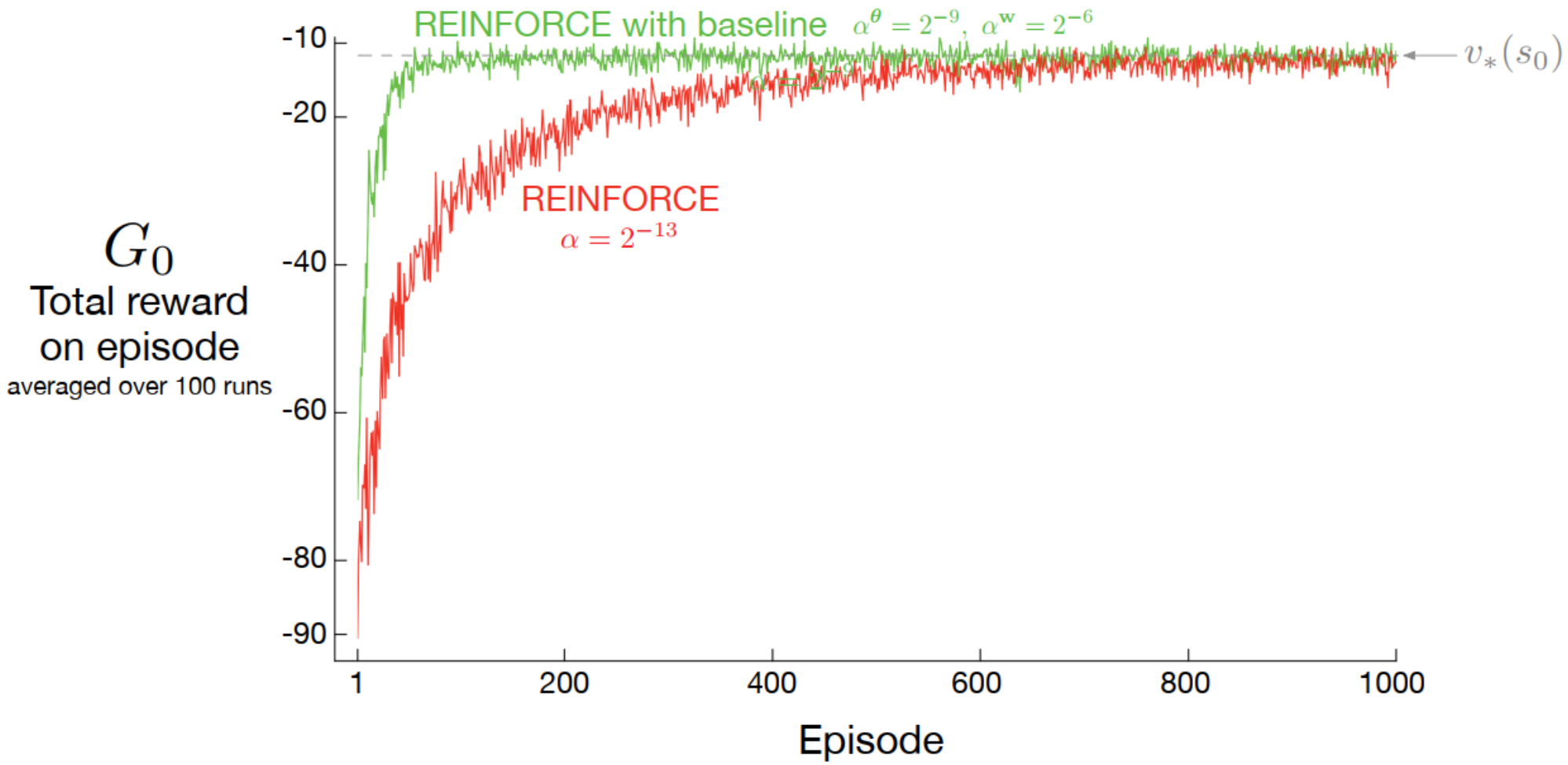
REINFORCE with Baseline: Algorithm

REINFORCE with Baseline (episodic), for estimating $\pi_{\boldsymbol{\theta}} \approx \pi_*$

Input: a differentiable policy parameterization $\pi(a|s, \boldsymbol{\theta})$
Input: a differentiable state-value function parameterization $\hat{v}(s,\mathbf{w})$
Algorithm parameters: step sizes $\alpha^{\boldsymbol{\theta}} > 0$, $\alpha^{\mathbf{w}} > 0$
Initialize policy parameter $\boldsymbol{\theta} \in \mathbb{R}^{d'}$ and state-value weights $\mathbf{w} \in \mathbb{R}^{d}$ (e.g., to $\mathbf{0}$)

Loop forever (for each episode):
  Generate an episode $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$, following $\pi(\cdot|\cdot, \boldsymbol{\theta})$
  Loop for each step of the episode $t = 0, 1, \ldots, T-1$:
    $G \leftarrow \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k$ (G_t)
    $\delta \leftarrow G - \hat{v}(S_t,\mathbf{w})$
    $\mathbf{w} \leftarrow \mathbf{w} + \alpha^{\mathbf{w}} \delta \nabla \hat{v}(S_t,\mathbf{w})$
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha^{\boldsymbol{\theta}} \gamma^t \delta \nabla \ln \pi(A_t|S_t, \boldsymbol{\theta})$

REINFORCE with Baseline: Short Corridor Example



Actor-Critic Methods

REINFORCE with baseline:

$$\boldsymbol{\theta}_{t+1} \doteq \boldsymbol{\theta}_t + \alpha\Big(G_t - b(S_t)\Big)\frac{\nabla\pi(A_t|S_t,\boldsymbol{\theta}_t)}{\pi(A_t|S_t,\boldsymbol{\theta}_t)}$$

$b(S_t) \rightarrow \hat{v}(s,\omega)$

$$\Rightarrow \theta_{t+1} \doteq \theta_t + \alpha\left(G_t - \hat{v}(s_t,\omega)\right)\frac{\nabla\pi(A_t|S_t,\theta_t)}{\pi(A_t|S_t,\theta_t)}$$

Replace full return G_t with one-step return $G_{t:t+1}$

Actor-Critic Methods

$$\nabla \ln \pi(A_t|S_t,\theta_t)$$

REINFORCE with baseline:

$$\Rightarrow \theta_{t+1} \doteq \theta_t + \alpha\left(G_t - \hat{v}(S_t,\omega)\right)\frac{\nabla \pi(A_t|S_t,\theta_t)}{\pi(A_t|S_t,\theta_t)}$$

Replace full return G_t with one-step return $G_{t:t+1}$

$$G_{t:t+1} \doteq R_{t+1} + \gamma \hat{v}(S_{t+1},\omega)$$

$$\theta_{t+1} \doteq \theta_t + \alpha\left[R_{t+1} + \gamma \hat{v}(S_{t+1},\omega) - \hat{v}(S_t,\omega)\right]\nabla \ln \pi(A_t|S_t,\theta_t)$$

TD Error δ_t

Actor-Critic Methods

Actor Critic Update

$$\theta_{t+1} \doteq \theta_t + \alpha \underbrace{\left[R_{t+1} + \gamma \hat{v}(s_{t+1}, w) - \hat{v}(s_t, w) \right]}_{\text{TD Error } \delta_t} \nabla \ln \pi(A_t | s_t, \theta_t)$$

$$\Rightarrow \theta_{t+1} \doteq \theta_t + \alpha \delta_t \nabla \ln \pi(A_t | s_t, \theta_t)$$

Actor-Critic Methods: Algorithm

One-step Actor–Critic (episodic), for estimating $\pi_{\boldsymbol{\theta}} \approx \pi_*$

Input: a differentiable policy parameterization $\pi(a|s,\boldsymbol{\theta})$
Input: a differentiable state-value function parameterization $\hat{v}(s,\mathbf{w})$
Parameters: step sizes $\alpha^{\boldsymbol{\theta}} > 0$, $\alpha^{\mathbf{w}} > 0$
Initialize policy parameter $\boldsymbol{\theta} \in \mathbb{R}^{d'}$ and state-value weights $\mathbf{w} \in \mathbb{R}^{d}$ (e.g., to $\mathbf{0}$)
Loop forever (for each episode):
  Initialize S (first state of episode)
  $I \leftarrow 1$
  Loop while S is not terminal (for each time step):
    $A \sim \pi(\cdot|S,\boldsymbol{\theta})$
    Take action A, observe S', R
    $\delta \leftarrow R + \gamma \hat{v}(S',\mathbf{w}) - \hat{v}(S,\mathbf{w})$ (if S' is terminal, then $\hat{v}(S',\mathbf{w}) \doteq 0$)
    $\mathbf{w} \leftarrow \mathbf{w} + \alpha^{\mathbf{w}} \delta \nabla \hat{v}(S,\mathbf{w})$
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha^{\boldsymbol{\theta}} I \delta \nabla \ln \pi(A|S,\boldsymbol{\theta})$
    $I \leftarrow \gamma I$
    $S \leftarrow S'$