Reinforcement Learning

Policy Gradients

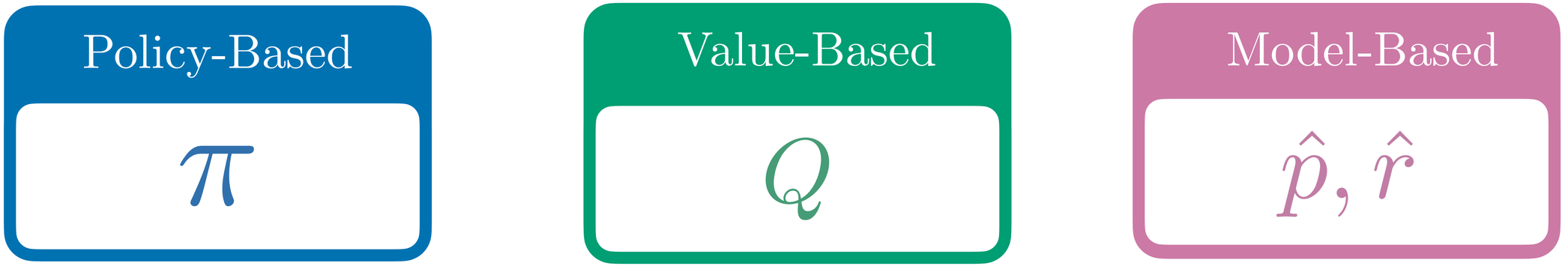
David Abel, Michael Herrmann
Based on slides by Stefano V. Albrecht
28 February, 2025

Lecture Outline

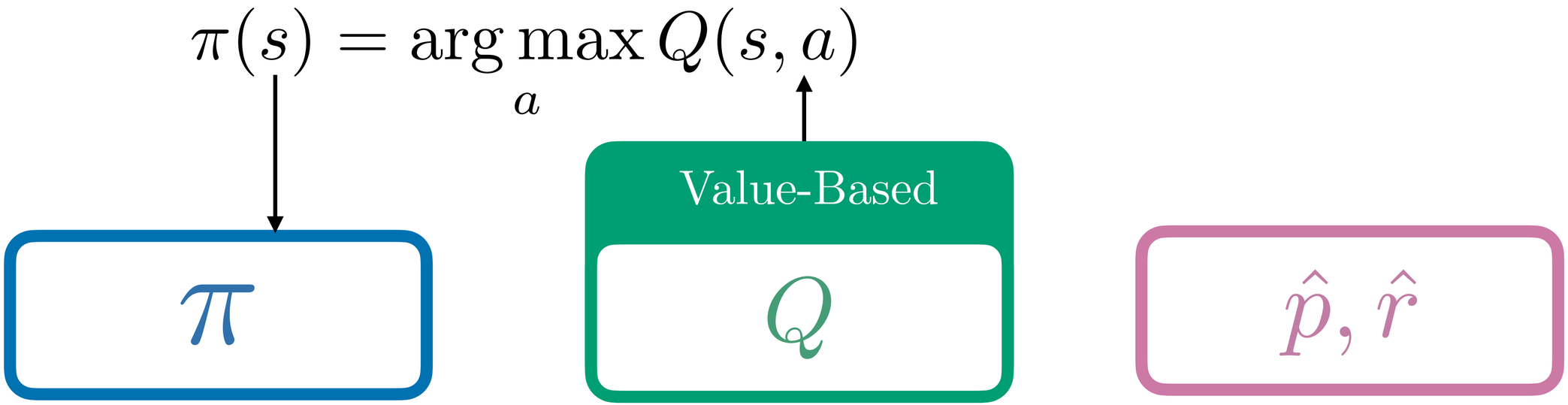
1. RL Algorithms: An Overview
2. Policy Gradients: Main Idea
3. Algorithms: REINFORCE, Actor-Critic

Note: Last examinable material!

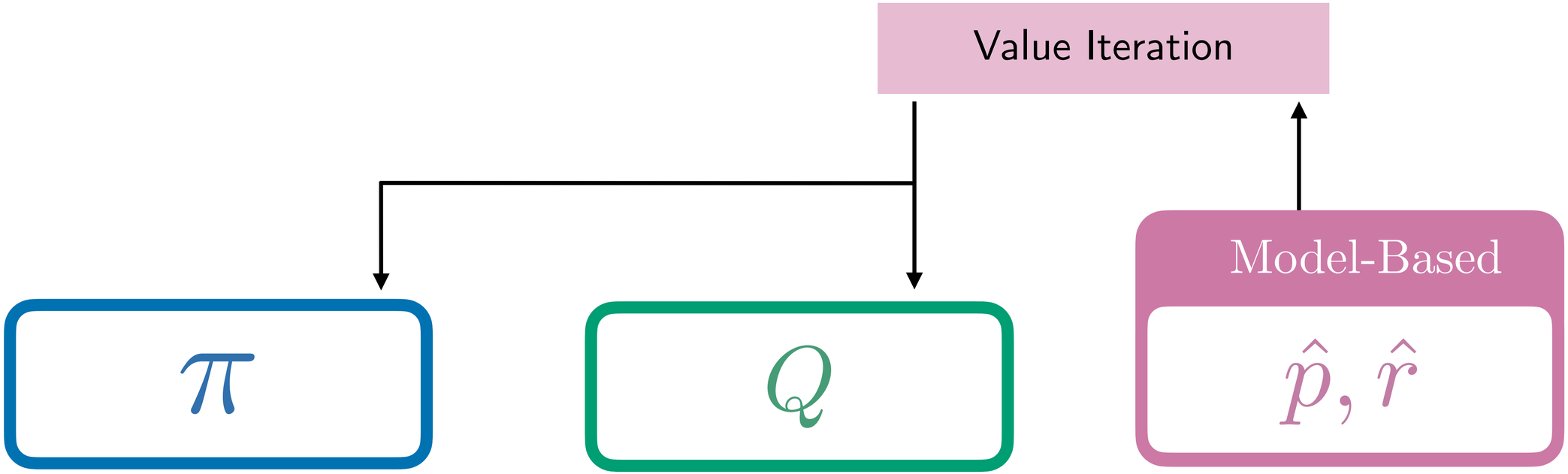
RL Algorithms: Three Kinds



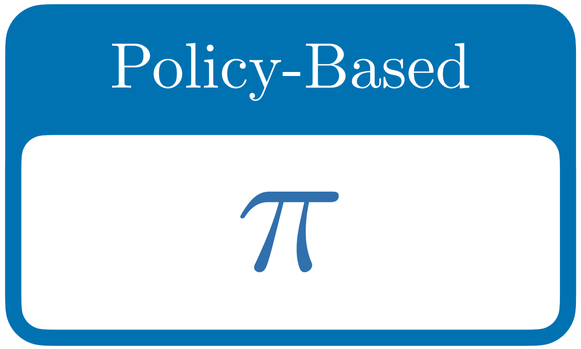
RL Algorithms: Three Kinds

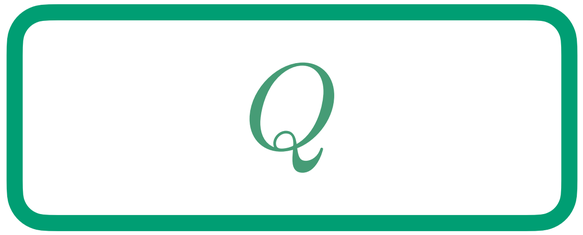


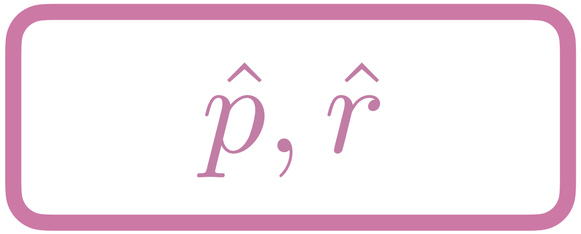
RL Algorithms: Three Kinds



RL Algorithms: Three Kinds





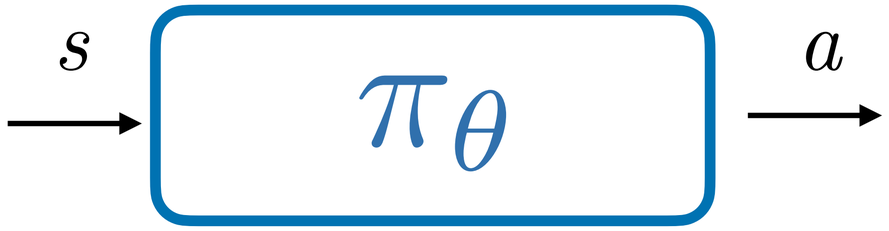


*Learn policy **directly**!*

Policy-Based Methods

Gradient-based optimization

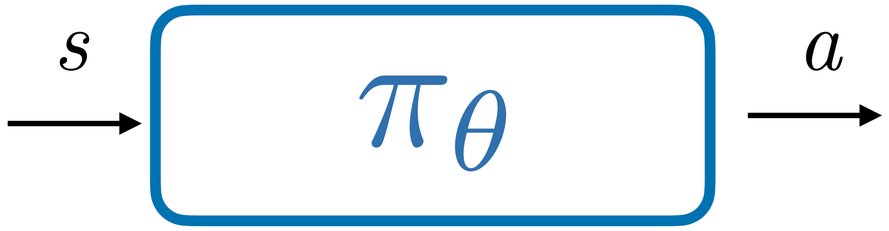
$$\theta_{t+1} = \theta_t + \alpha \widehat{\nabla J(\theta_t)}$$



$$\theta \in \mathbb{R}^d$$

Policy-Based Methods

Q: But how do we represent the policy?



$$\theta \in \mathbb{R}^d$$

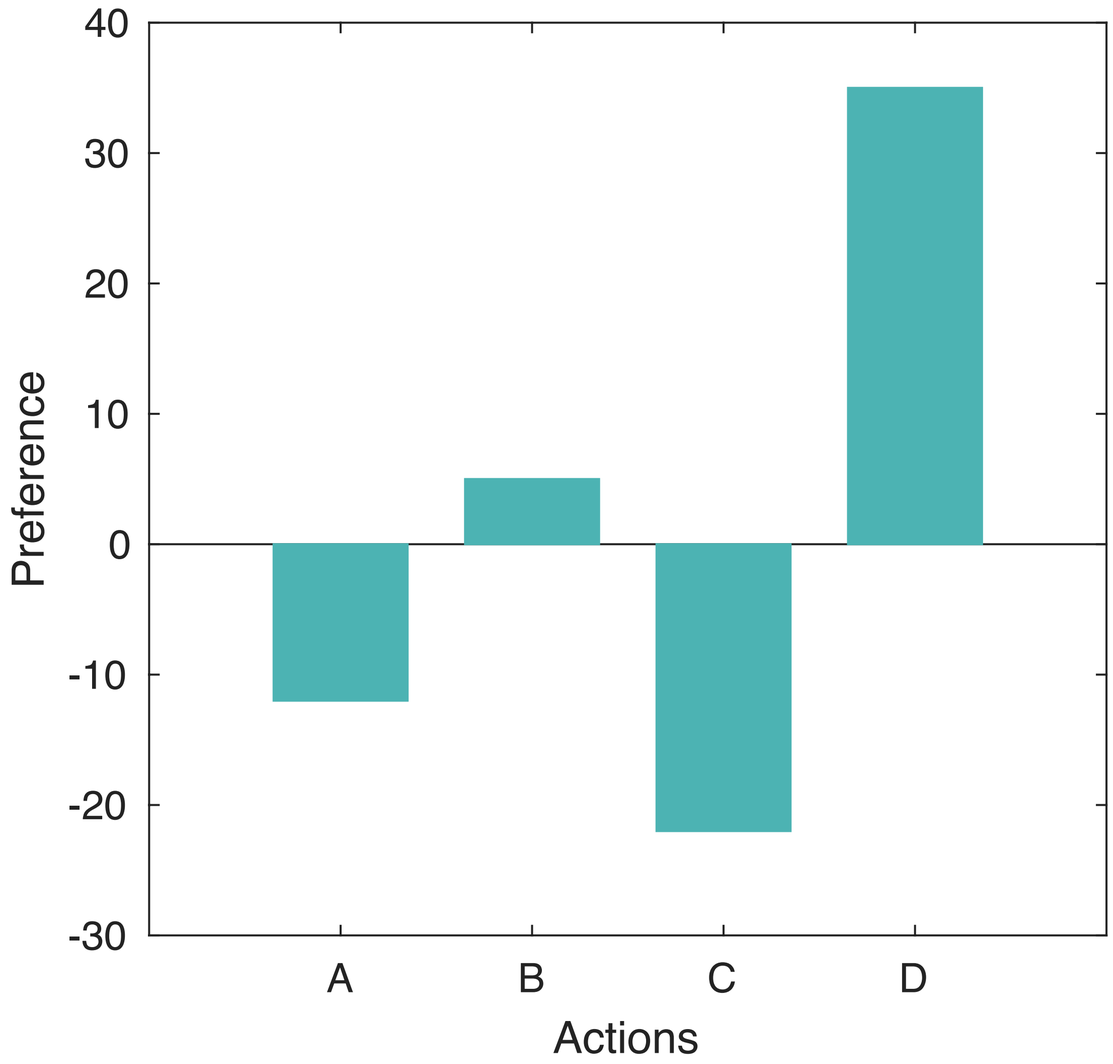
Example 1: Softmax Policies

Softmax Policy

$$\pi(a \mid s, \theta) = \frac{e^{h(s,a,\theta)}}{\sum_{b \in \mathcal{A}} e^{h(s,b,\theta)}}$$

$$h(s, a, \theta) = \theta^\top \boxed{x(s, a)}$$ *State-action features*

Example 1: Softmax Policies



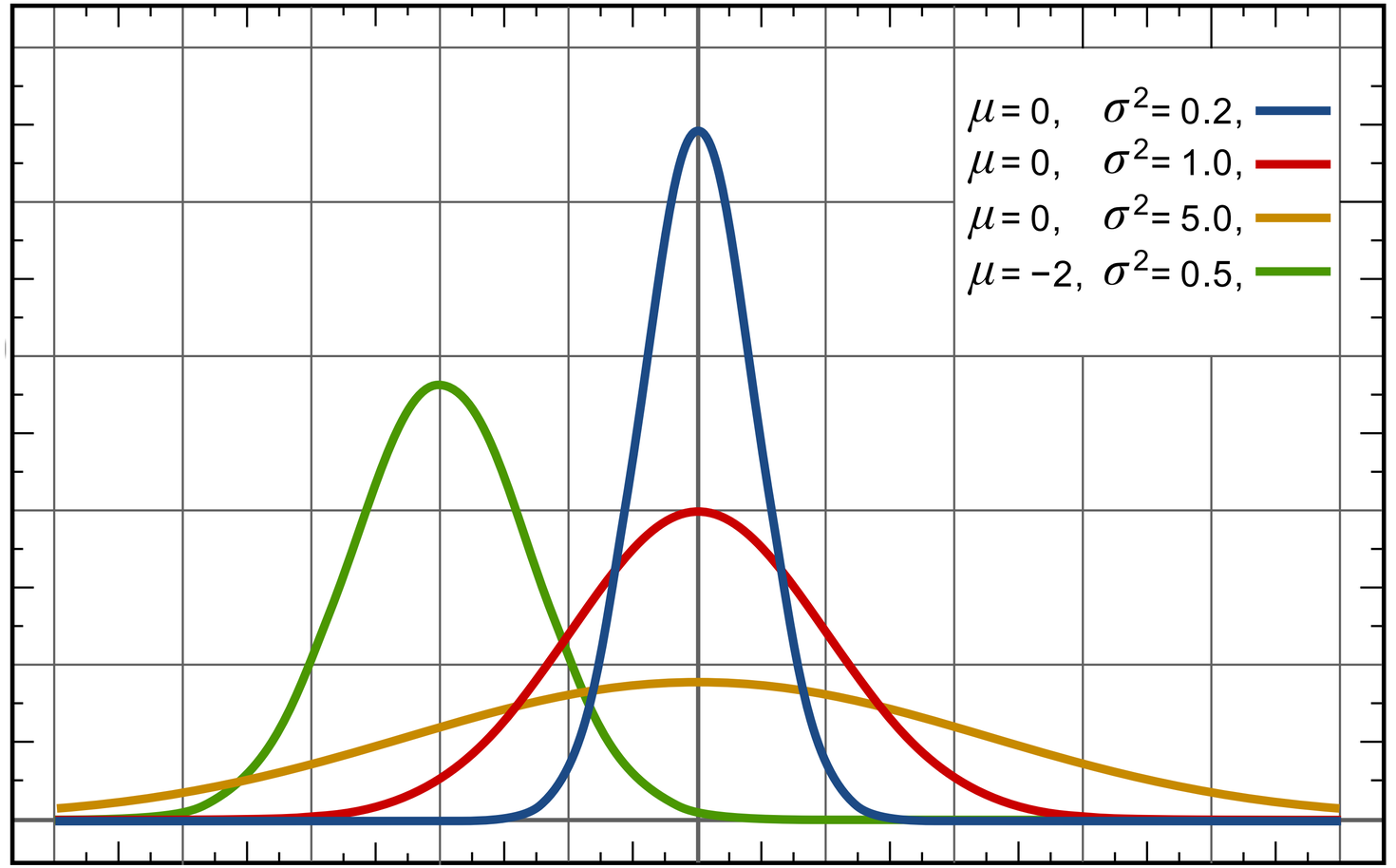
Example 2: Gaussian Policies

Gaussian Policy

$$\pi(a \mid s, \theta) \sim \mathcal{N}(\mu(s, \theta), \sigma^2)$$

For example: $\mu(s, \theta) = \theta^\top x(s)$ *State features*

Example 2: Gaussian Policies



RL Algorithms: Three Kinds

Model-Free

Policy-Based

π

Learn policy, use to act

Value-Based

Q

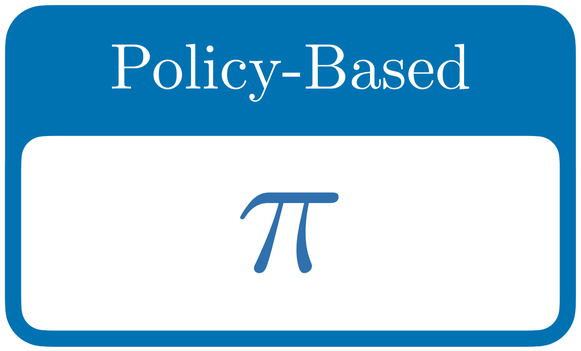
Learn value, use to get policy

Model-Based

$\hat{p}, \hat{r}$

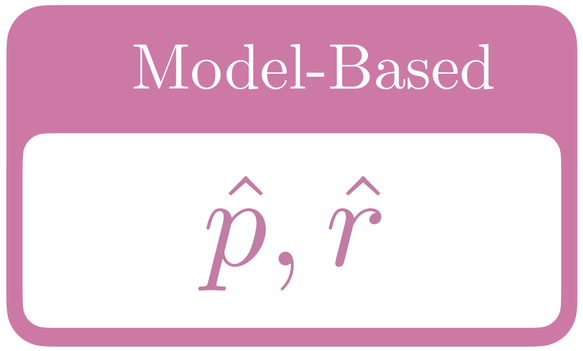
*Learn model, then **plan** to get policy*

Discussion



Learn policy, use to act

Learn value, use to get policy



*Learn model, then **plan** to get policy*

Discussion (2 minutes):

What is one advantage or disadvantage of any of the above three classes?

What is one setting you can think of where we should clearly use one of the above over the other two?

Policy Optimisation

Definition: Policy Optimisation Problem

Given: $\pi(a \mid s, \theta)$, interaction with MDP m

Find: optimal choice of θ

Q: How do we measure the quality of a given θ?

Policy Optimisation

Definition: Policy Optimisation Problem

Given: $\pi(a \mid s, \theta)$, interaction with MDP m

Find: optimal choice of θ

Q: How do we measure the quality of a given θ? → $J(\theta) = v_{\pi_\theta}(s_0)$

Episodic

Policy Gradient Algorithms: Main Idea

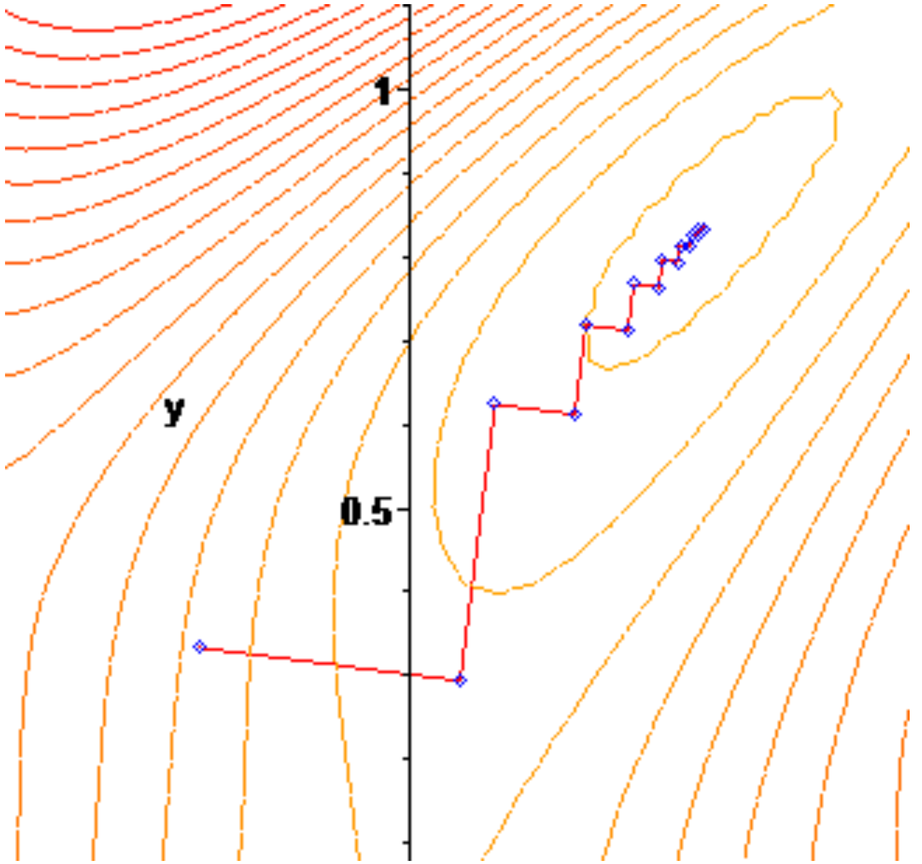
Sketch: Policy Gradient Algorithms

initialise θ_0

for $t = 0, 1, \ldots$

collect data using π_{θ_t}

$\theta_{t+1} = \theta_t + \alpha \nabla J(\theta_t)$



Q: How do we compute this..?

Policy Gradient Theorem

Policy Gradient Theorem:

For any *differentiable* policy π, the policy gradient is

$$\nabla J(\theta) = \sum_s d_\pi(s) \sum_a q_\pi(s,a) \nabla \pi(a|s,\theta)$$

$d_\pi(s)$ is the *on-policy distribution* under π:

Policy Gradient Theorem

Policy Gradient Theorem:

For any *differentiable* policy π, the policy gradient is

$$\nabla J(\theta) \;=\; \sum_s d_\pi(s) \sum_a q_\pi(s,a)\, \nabla \pi(a|s,\theta)$$

$d_\pi(s)$ is the *on-policy distribution* under π:

- For start-state value: $d_\pi(s) = \sum_{t=0}^{\infty} \gamma^t \Pr\{S_t = s \mid s_0, \pi\}$
- For average reward: $d_\pi(s) = \lim_{t\to\infty} \Pr\{S_t = s \mid \pi\}$ (steady-state dist.)

Note: does not require derivative of environment dynamics $p(s', r|s, a)$!

Policy Gradient Theorem: Breakdown

$$\nabla J(\theta) = \sum_{s} d_\pi(s) \sum_{a} q_\pi(s,a) \nabla \pi(a|s,\theta)$$

Policy Gradient Theorem: Breakdown

$$\begin{aligned}\nabla J(\theta) &= \sum_s d_\pi(s) \sum_a q_\pi(s,a)\, \nabla \pi(a|s,\theta) \\ &= \mathbb{E}_\pi \left[\sum_a q_\pi(S_t,a)\, \nabla \pi(a|S_t,\theta) \right]\end{aligned}$$

Policy Gradient Theorem: Breakdown

$$
\begin{aligned}
\nabla J(\theta) &= \sum_{s} d_\pi(s) \sum_{a} q_\pi(s,a)\, \nabla \pi(a|s,\theta) \\
&= \mathbb{E}_\pi \left[\sum_{a} q_\pi(S_t,a)\, \nabla \pi(a|S_t,\theta) \right] \\
&= \mathbb{E}_\pi \left[\sum_{a} \pi(a|S_t,\theta)\, q_\pi(S_t,a)\, \frac{\nabla \pi(a|S_t,\theta)}{\pi(a|S_t,\theta)} \right]
\end{aligned}
$$

Policy Gradient Theorem: Breakdown

$$
\begin{aligned}
\nabla J(\theta) &= \sum_s d_\pi(s) \sum_a q_\pi(s,a)\, \nabla \pi(a|s,\theta) \\
&= \mathbb{E}_\pi \left[\sum_a q_\pi(S_t,a)\, \nabla \pi(a|S_t,\theta) \right] \\
&= \mathbb{E}_\pi \left[\sum_a \pi(a|S_t,\theta)\, q_\pi(S_t,a)\, \frac{\nabla \pi(a|S_t,\theta)}{\pi(a|S_t,\theta)} \right] \\
&= \mathbb{E}_\pi \left[q_\pi(S_t,A_t)\, \frac{\nabla \pi(A_t|S_t,\theta)}{\pi(A_t|S_t,\theta)} \right]
\end{aligned}
$$

Policy Gradient Theorem: Breakdown

$$
\begin{aligned}
\nabla J(\theta) &= \sum_s d_\pi(s) \sum_a q_\pi(s,a)\, \nabla \pi(a|s,\theta) \\
&= \mathbb{E}_\pi \left[\sum_a q_\pi(S_t,a)\, \nabla \pi(a|S_t,\theta) \right] \\
&= \mathbb{E}_\pi \left[\sum_a \pi(a|S_t,\theta)\, q_\pi(S_t,a)\, \frac{\nabla \pi(a|S_t,\theta)}{\pi(a|S_t,\theta)} \right] \\
&= \mathbb{E}_\pi \left[q_\pi(S_t,A_t)\, \frac{\nabla \pi(A_t|S_t,\theta)}{\pi(A_t|S_t,\theta)} \right] \\
&= \mathbb{E}_\pi [q_\pi(S_t,A_t)\, \nabla \ln \pi(A_t|S_t,\theta)]
\end{aligned}
$$

Policy Gradient: General Form

Sketch: Policy Gradient Algorithms

initialise θ_0

for $t = 0, 1, \ldots$

collect data using π_{θ_t}

Need to approximate!

$$\theta_{t+1} = \theta_t + \alpha(q_\pi(S_t, A_t)\nabla \ln \pi(A_t \mid S_t, \theta_t))$$

Computing / Approximating Two Key Quantities

$$\theta_{t+1} = \theta_t + \alpha(q_\pi(S_t, A_t)\nabla \ln \pi(A_t \mid S_t, \theta_t))$$

$q_\pi(S_t, A_t)$

Monte Carlo estimate of G_t

since,

$$\mathbb{E}_\pi[G_t \mid S_t, A_t] = q_\pi(S_t, A_t)$$

REINFORCE Algorithm

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

softmax

$$= x(s,a) - \sum_{a'} \pi(a' \mid s, \theta) x(s, a')$$

Gaussian

$$= (a - \mu(s,\theta))x(s)/\sigma^2$$

Algorithm 1: REINFORCE — Pseudocode

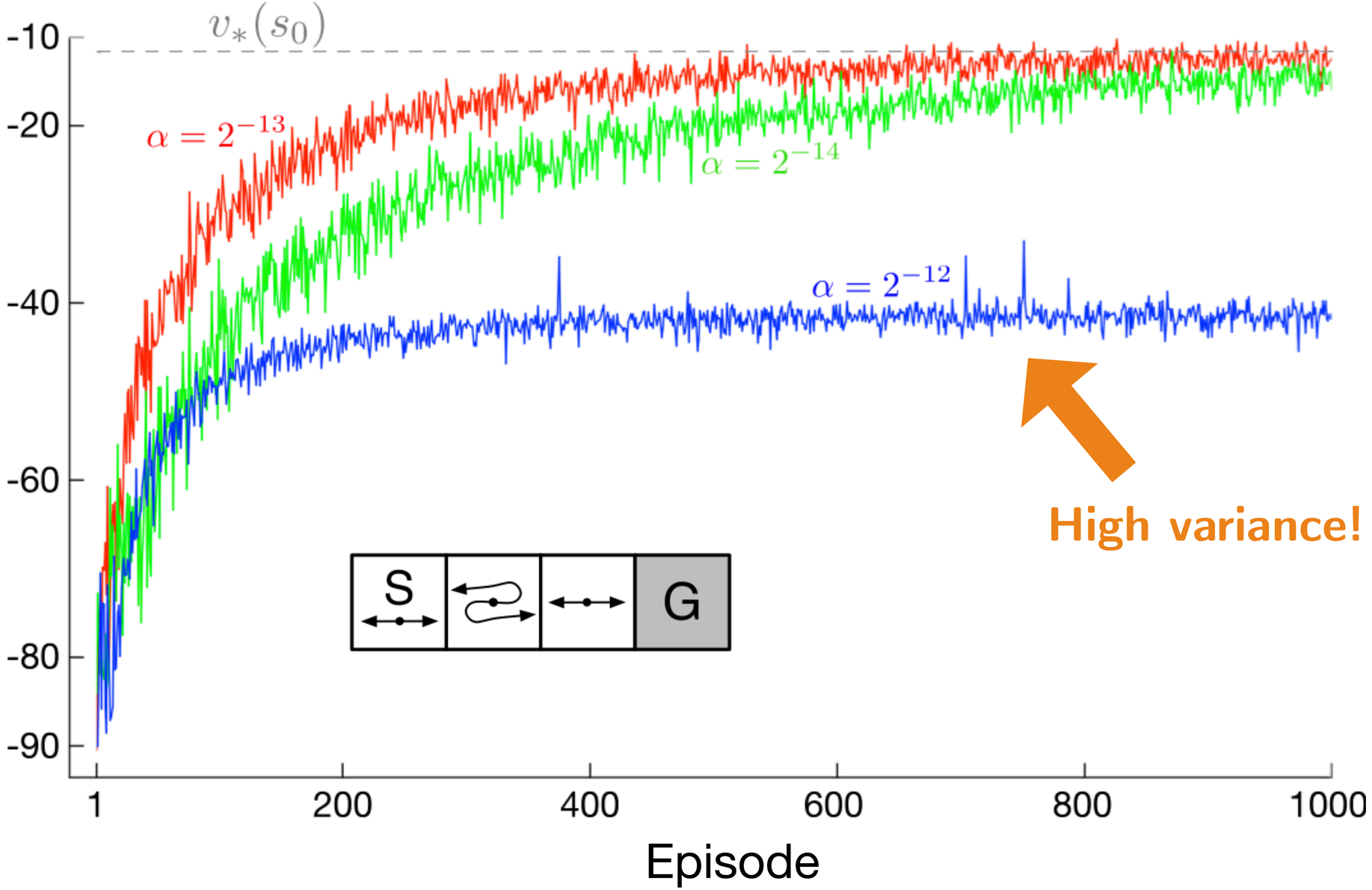
REINFORCE

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 $t = 0, 1, \ldots, T-1$:
        $G \leftarrow \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k$
        $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \gamma^t G \nabla \ln \pi(A_t|S_t, \boldsymbol{\theta})$

Example: REINFORCE in Corridor

G_0
Total reward on episode
averaged over 100 runs



High variance!

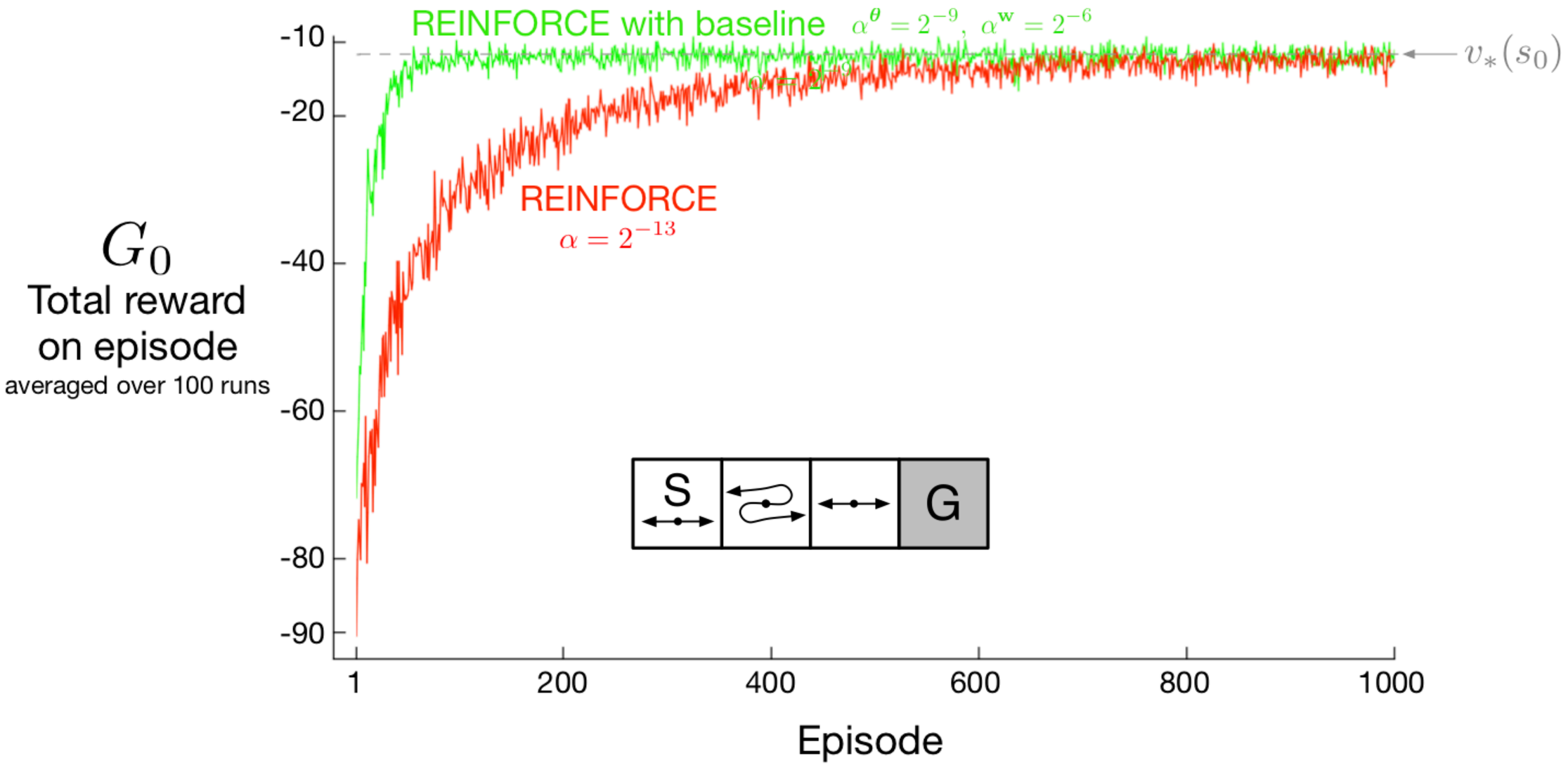
Mitigating Variance: Fix 1 — Add a Baseline

$$\theta_{t+1} = \theta_t + \alpha[(q_\pi(S_t, A_t) - b(S_t))\nabla \ln \pi(A_t \mid S_t, \theta_t)]$$

Typically: $b(S_t) = \hat{v}(S_t)$

Does not change expectation, but can reduce variance

Mitigating Variance: Fix 1 — Add a Baseline



REINFORCE is Episodic

REINFORCE

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** ★ → **Only episodic!**

  Generate an episode $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$, following $\pi(\cdot|\cdot, \boldsymbol{\theta})$

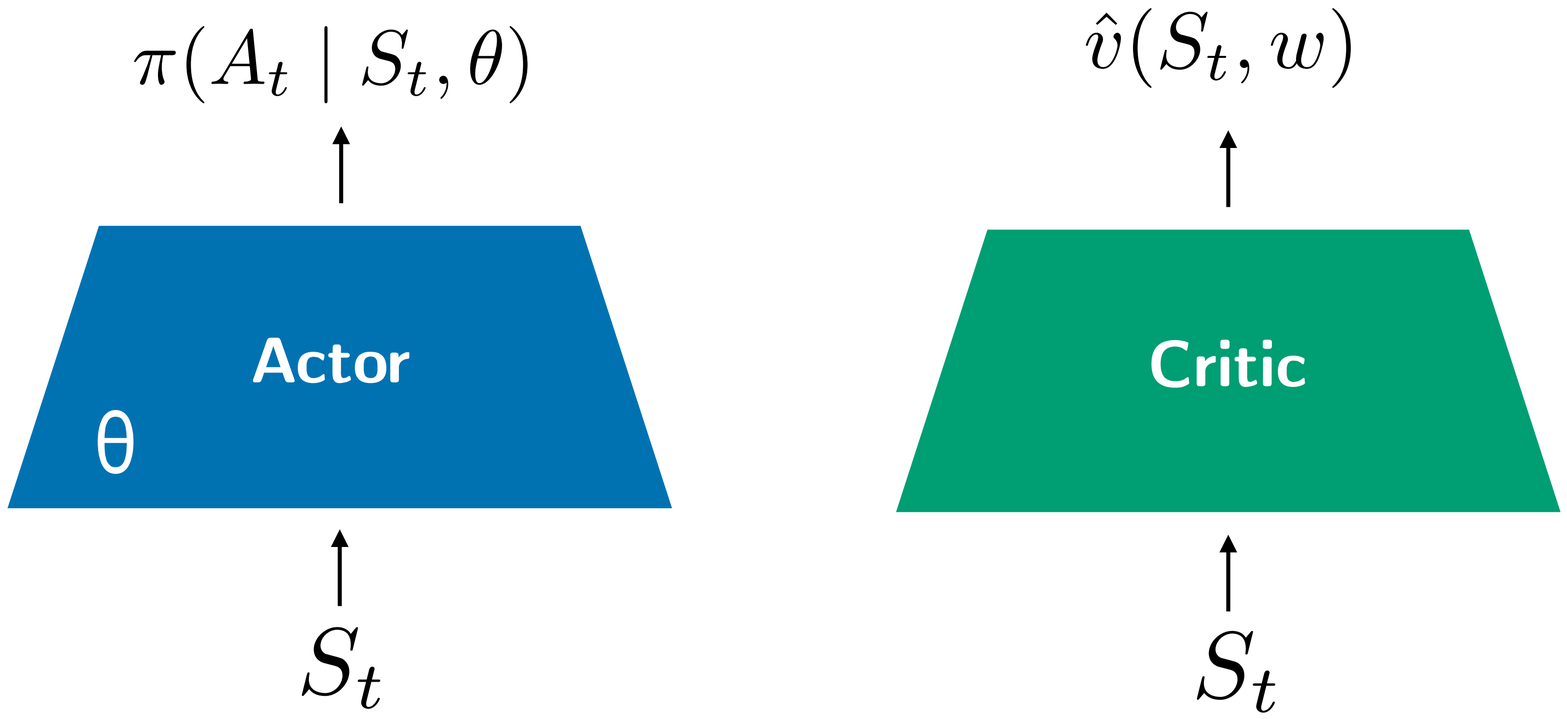
  Loop for each step of the episode $t = 0, 1, \ldots, T-1$:

    $G \leftarrow \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k$

    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \gamma^t G \nabla \ln \pi(A_t|S_t, \boldsymbol{\theta})$

Fix: Actor-Critic

$$\theta_{t+1} = \theta_t + \alpha[(R_{t+1} + \gamma\hat{v}(S_{t+1}, \mathrm{w}) - \hat{v}(S_t, \mathrm{w}))\nabla \ln \pi(A_t \mid S_t, \theta_t)]$$

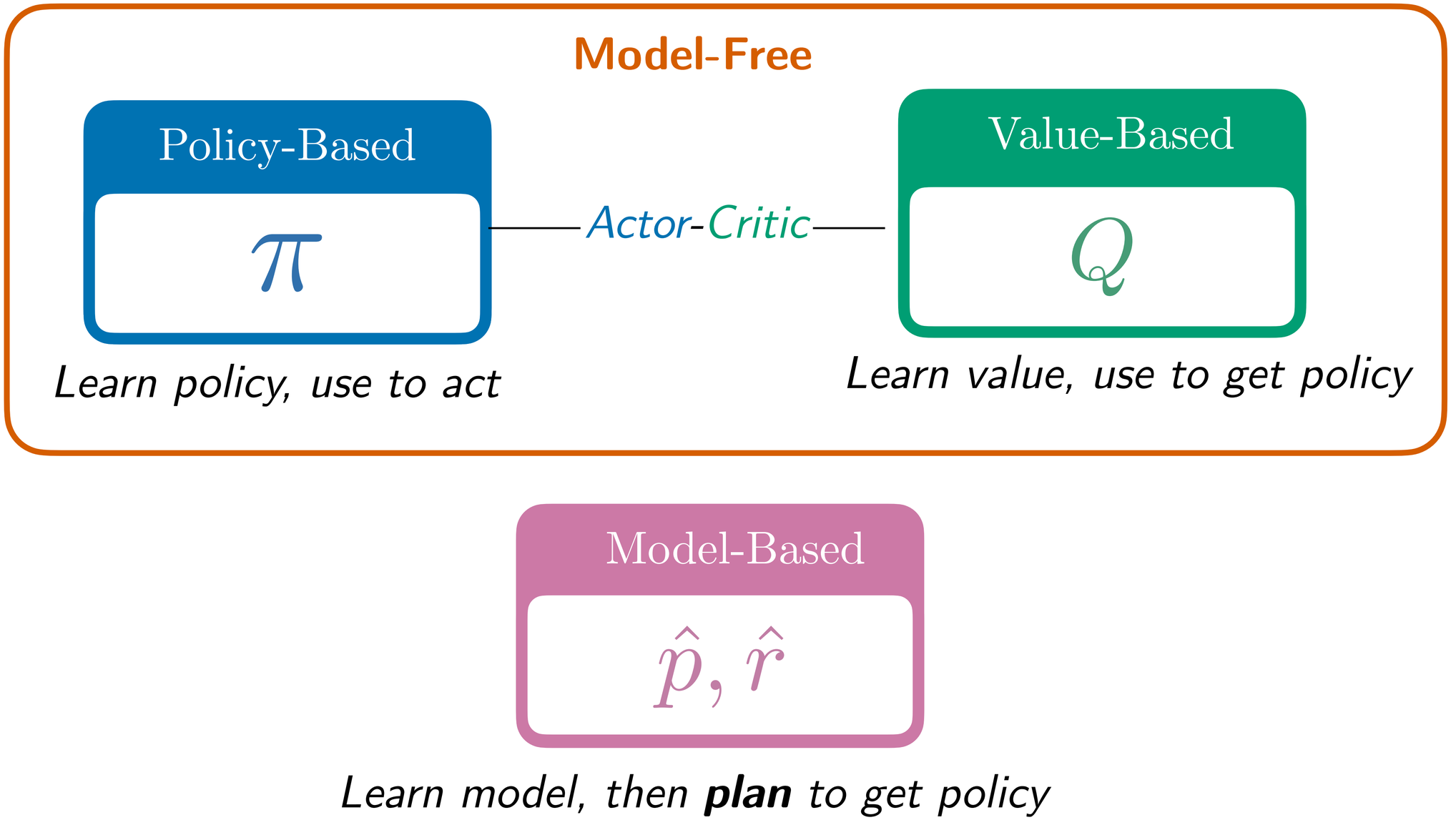


Algorithm 2: Actor-Critic w/ TD(0)

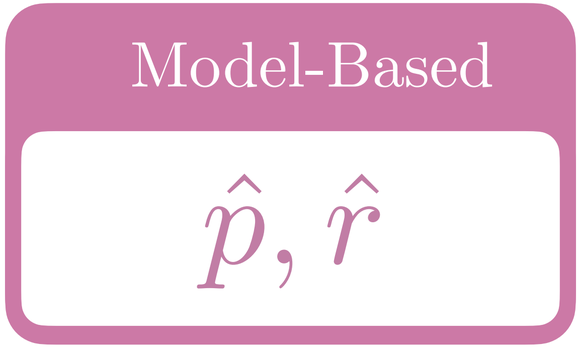
Input: a differentiable policy parameterization $\pi(a|s,\boldsymbol{\theta})$
Input: a differentiable state-value function parameterization $\hat{v}(s,\mathbf{w})$
Parameters: step sizes $\alpha^{\boldsymbol{\theta}} > 0$, $\alpha^{\mathbf{w}} > 0$
Initialize policy parameter $\boldsymbol{\theta} \in \mathbb{R}^{d'}$ 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 $\hat{v}(S',\mathbf{w}) \doteq 0$)
    $\mathbf{w} \leftarrow \mathbf{w} + \alpha^{\mathbf{w}} \delta \nabla \hat{v}(S,\mathbf{w})$
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha^{\boldsymbol{\theta}} I \delta \nabla \ln \pi(A|S,\boldsymbol{\theta})$
    $I \leftarrow \gamma I$
    $S \leftarrow S'$

RL Algorithms: Three Kinds



Not Examined: A Simple Model-Based Algorithm



*Learn model, then **plan** to get policy*

Coming Up...

Deep Reinforcement Learning

The Reward Hypothesis, RLHF

RL beyond MDPs

MARL / Cognitive RL

Not examined

Reading

- **RL book, Chapter 13 (13.1–13.5)**
- **Note: End of examinable material. For extra exam revision, see Tutorials 8 & 9.**

- Optional:

  Policy Gradient Methods for Reinforcement Learning with Function Approximation by Sutton et al. (1999)

  Simple statistical gradient-following algorithms for connectionist reinforcement learning by Williams (1992)

  Proximal Policy Optimization Algorithms by Schulman et al. (2017)

  https://arxiv.org/abs/1707.06347