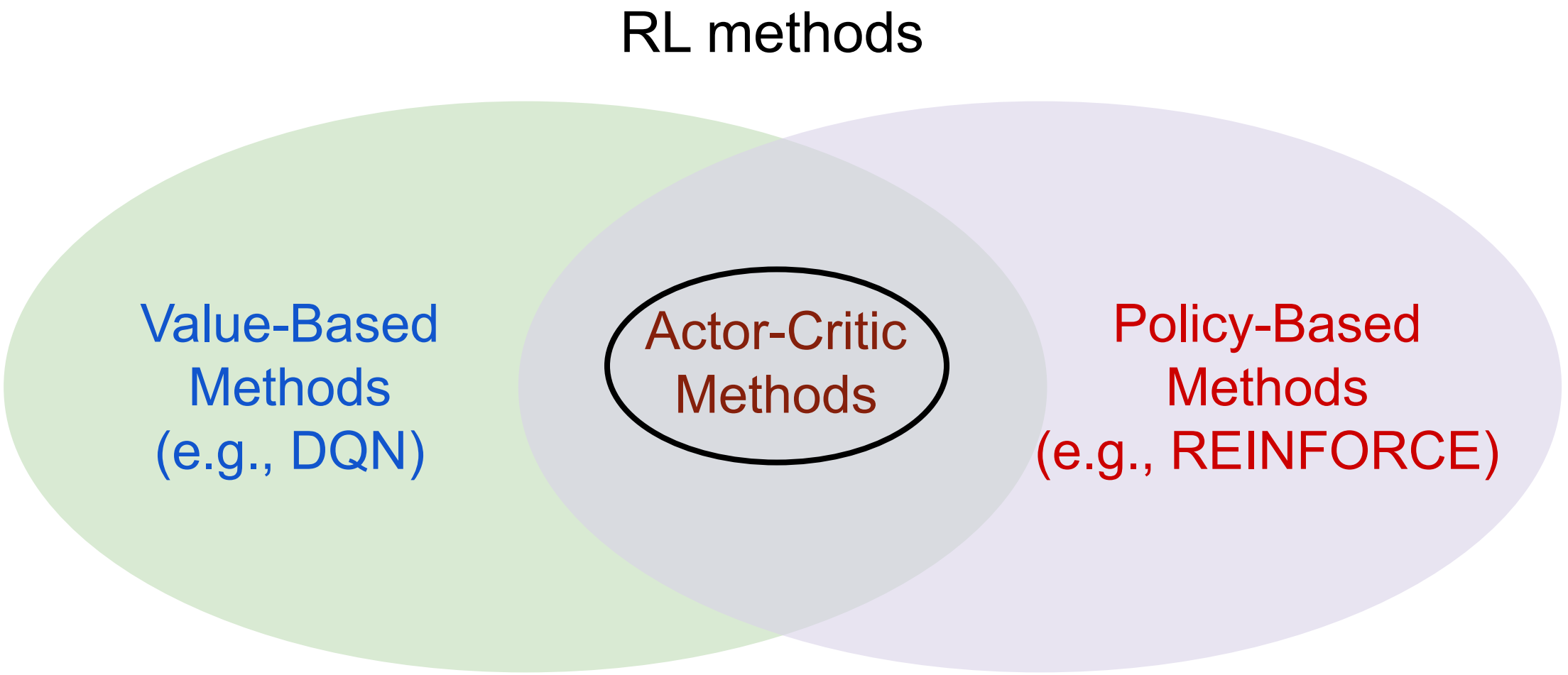
384.195 Robot Learning
Deep Reinforcement Learning Part 2

Dongheui Lee, Yashuai Yan

Overview


RL methods
Value-Based Methods (e.g., DQN)
Actor-Critic Methods
Policy-Based Methods (e.g., REINFORCE)

Actor-Critic Methods

Definition: State-value function

- $V^{\pi}(s) = \sum_{a} \underbrace{\pi(a \mid s)}_{\text{unknown}} \cdot \underbrace{Q^{\pi}(s, a)}_{\text{unknown}}$

Value-based RL:

(approx. G fct. eg. with NN)

$$Q^{\pi}(s, a) \simeq q(s, a; \mathbf{w})$$

Policy-based RL:

$$\pi(a \mid s) \simeq \pi(a \mid s; \theta)$$

Actor-Critic Methods

Definition: State-value function

- $V^{\pi}(s) = \sum_{a} \underbrace{\pi(a \mid s)}_{\text{unknown}} \cdot \underbrace{Q^{\pi}(s, a)}_{\text{unknown}} \simeq \sum_{a} \pi(a \mid s; \theta) \cdot q(s, a; \mathbf{w})$

Actor-Critic Methods

Definition: State-value function

- $V^{\pi}(s) = \sum_{a} \underbrace{\pi(a \mid s)}_{\text{unknown}} \cdot \underbrace{Q^{\pi}(s, a)}_{\text{unknown}} \simeq \sum_{a} \pi(a \mid s; \theta) \cdot q(s, a; \mathbf{w})$

Policy network (actor):

- Use neural network $\pi(a \mid s; \theta)$ to approximate $\pi(a \mid s)$.
- θ: trainable parameters.

Value network (critic):

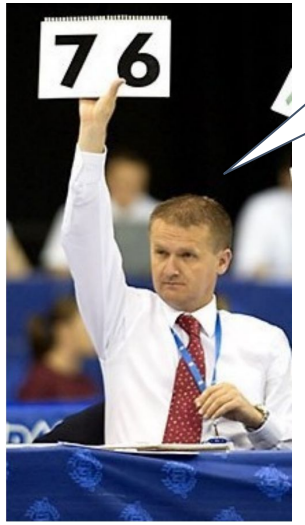
- Use neural network $q(s, a; \mathbf{w})$ to approximate $Q^{\pi}(s, a)$.
- $\mathbf{w}$: trainable parameters.

Actor-Critic in real World

Player (actor)

I learn how to act to get good scores.

Referee (critic)



I learn to judge the player with scores.

Training Actor-Critic Methods

Definition: Approximate state-value function

- $V(s;\theta,\mathbf{w}) = \sum_a \pi(a \mid s;\theta) \cdot q(s,\ a;\mathbf{w})$

We know how to train each network separately.

- Update policy network $\pi(a \mid s;\theta)$ → policy gradient ascent.

$$\theta_{t+1} = \theta_t + \beta \cdot \frac{\partial V(s;\theta)}{\partial \theta}\Big|_{\theta=\theta_t}$$

- Update value network $q(s,\ a;\mathbf{w})$ → value gradient descent.

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \alpha \cdot \delta \cdot \frac{\partial q(s,\ a;\mathbf{w})}{\partial \mathbf{w}}\Big|_{\mathbf{w}=\mathbf{w_t}}$$

Training Actor-Critic

Definition: Approximate state-value function

- $V(s; \theta, \mathbf{w}) = \sum_a \pi(a \mid s; \theta) \cdot q(s, a; \mathbf{w})$

Training: Update the parameter θ and $\mathbf{w}$

1. Observe the state s_t.
2. Randomly sample action $a_t \sim \pi(\cdot \mid s_t; \theta_t)$.
3. Perform a_t and observe new state s_{t+1} and reward r_t.
4. Update $\mathbf{w}$ (in value network).
5. Update θ (in policy network).

Update Value Network $q(s, a; \mathbf{w})$ using TD

Recall from last lecture:

- compute $q(s_t, a_t; \mathbf{w}_t)$ and $q(s_{t+1}, a_{t+1}; \mathbf{w}_t)$
- TD target: $y_t = r_t + \gamma \cdot q(s_{t+1}, a_{t+1}; \mathbf{w}_t)$

used to minimize

- TD error: $\delta = r_t + \gamma q(s_{t+1}, a_{t+1}; \mathbf{w}) - q(s_t, a_t; \mathbf{w})$
- Value gradient: $\frac{\partial \mathcal{L}(\mathbf{w})}{\partial \mathbf{w}} = \delta \cdot \frac{\partial q(s_t, a_t; \mathbf{w})}{\partial \mathbf{w}}$
- Gradient descent: $\mathbf{w}_{t+1} = \mathbf{w}_t - \alpha \cdot \frac{\partial \mathcal{L}(\mathbf{w})}{\partial \mathbf{w}}\Big|_{\mathbf{w}=\mathbf{w}_t}$

Update Policy Network $\pi(a \mid s; \theta)$ using TD

Recall from **policy gradient**:

$$\frac{\partial V(s;\theta)}{\partial \theta} = \mathbb{E}_{A \sim \pi(\cdot \mid s;\theta)}\left[\frac{\partial \log \pi(A \mid s;\theta)}{\partial \theta} \cdot q(s, A; \mathbf{w})\right]$$

Compute the expectation is infeasible, we need to estimate it

Expectation estimate using Monte Carlo:

- Random sampling: $a \sim \pi(\cdot \mid s; \theta_t)$

by performing random actions

- Let $g(a) = \frac{\partial \log \pi(a \mid s;\theta)}{\partial \theta} \cdot q(s, a; \mathbf{w})$; $g(a)$ is an estimate of policy gradient

- Gradient ascent: $\theta_{t+1} = \theta_t + \beta \cdot g(a)$

Role of Actor and Critic

- What does the actor learn from?
  - Policy gradient: $\frac{\partial V(s;\theta)}{\partial \theta} \simeq \frac{\partial \log \pi(a \mid s;\theta)}{\partial \theta} \cdot q(s,\, a;\mathbf{w})$

Role of Actor and Critic

- What does the actor learn from? The critic supervises the actor to update
  - Policy gradient: $\frac{\partial V(s;\theta)}{\partial \theta} \simeq \frac{\partial \log \pi(a \mid s;\theta)}{\partial \theta} \cdot q(s, a; \mathbf{w})$

Role of Actor and Critic

➔ What does the actor learn from? The critic supervises the actor to update

- Policy gradient: $\frac{\partial V(s;\theta)}{\partial \theta} \simeq \frac{\partial \log \pi(a \mid s;\theta)}{\partial \theta} \cdot \boxed{q(s,\, a;\mathbf{w})}$

➔ What does the critic learn from?

- TD error: $\delta = r_t + \gamma q(s_{t+1},\, a_{t+1};\mathbf{w}) - q(s_t,\, a_t;\mathbf{w})$
- Value gradient: $\frac{\partial \mathcal{L}(\mathbf{w})}{\partial \mathbf{w}} = \delta \cdot \frac{\partial q(s_t,\, a_t;\mathbf{w})}{\partial \mathbf{w}}$

Role of Actor and Critic

➔ What does the actor learn from? The critic supervises the actor to update

- Policy gradient: $\frac{\partial V(s;\theta)}{\partial \theta} \simeq \frac{\partial \log \pi(a \mid s;\theta)}{\partial \theta} \cdot q(s, a; \mathbf{w})$

➔ What does the critic learn from? The critic learns from reward signal

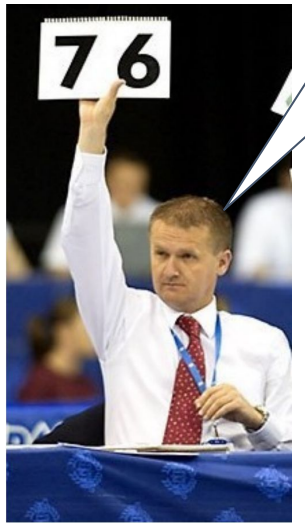
- TD error: $\delta = r_t + \gamma q(s_{t+1}, a_{t+1}; \mathbf{w}) - q(s_t, a_t; \mathbf{w})$
- Value gradient: $\frac{\partial \mathcal{L}(\mathbf{w})}{\partial \mathbf{w}} = \delta \cdot \frac{\partial q(s_t, a_t; \mathbf{w})}{\partial \mathbf{w}}$

Role of Actor and Critic

Player (actor)

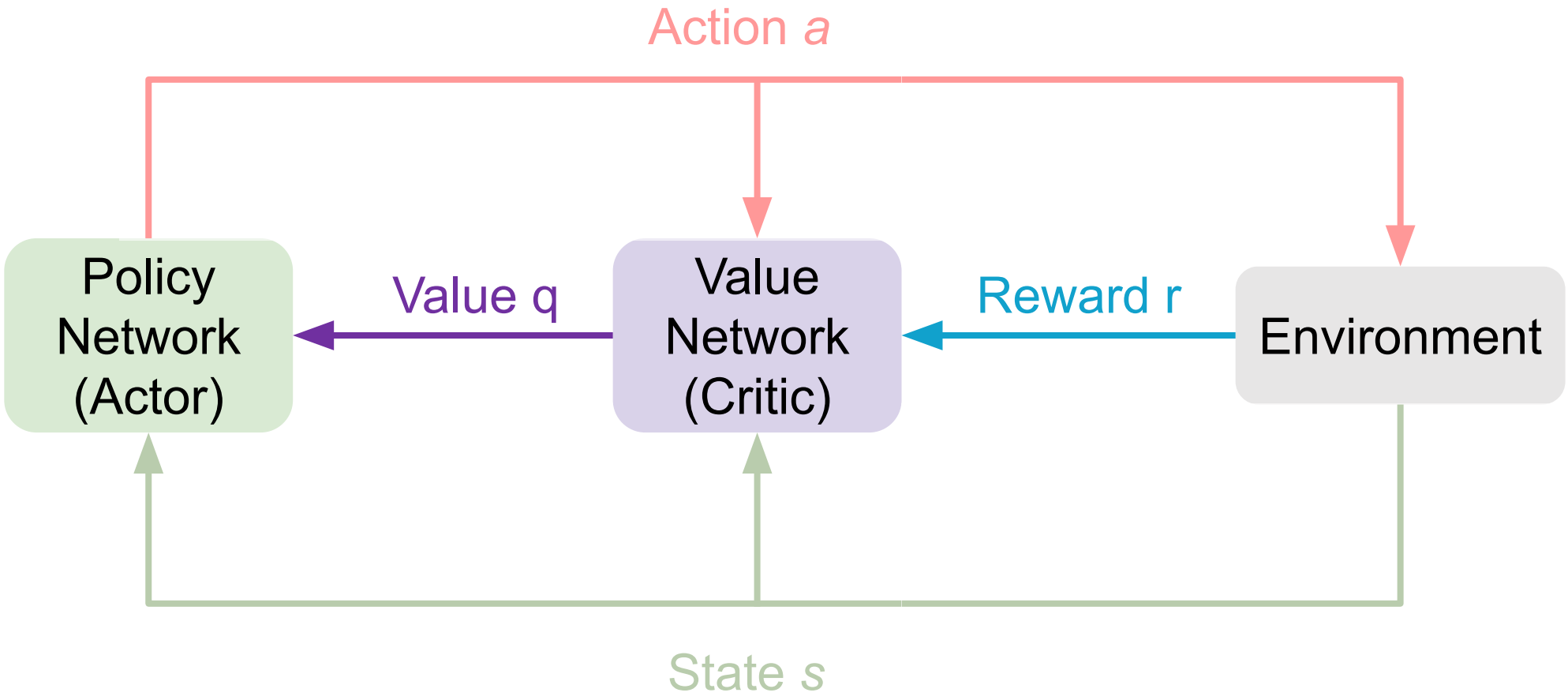
I learn to act based on the scores from the referee (critic).

Referee (critic)

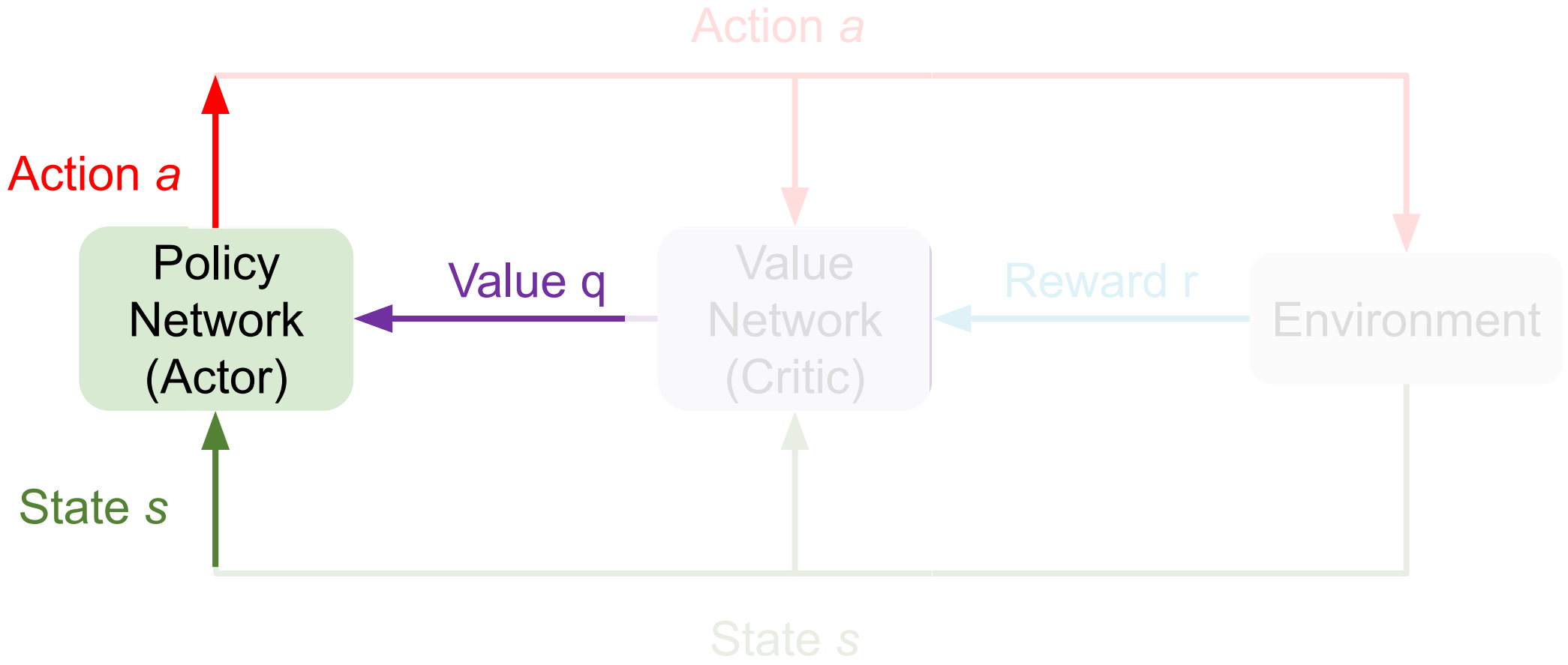


I learn to judge the player (actor) based on rewards from environment.

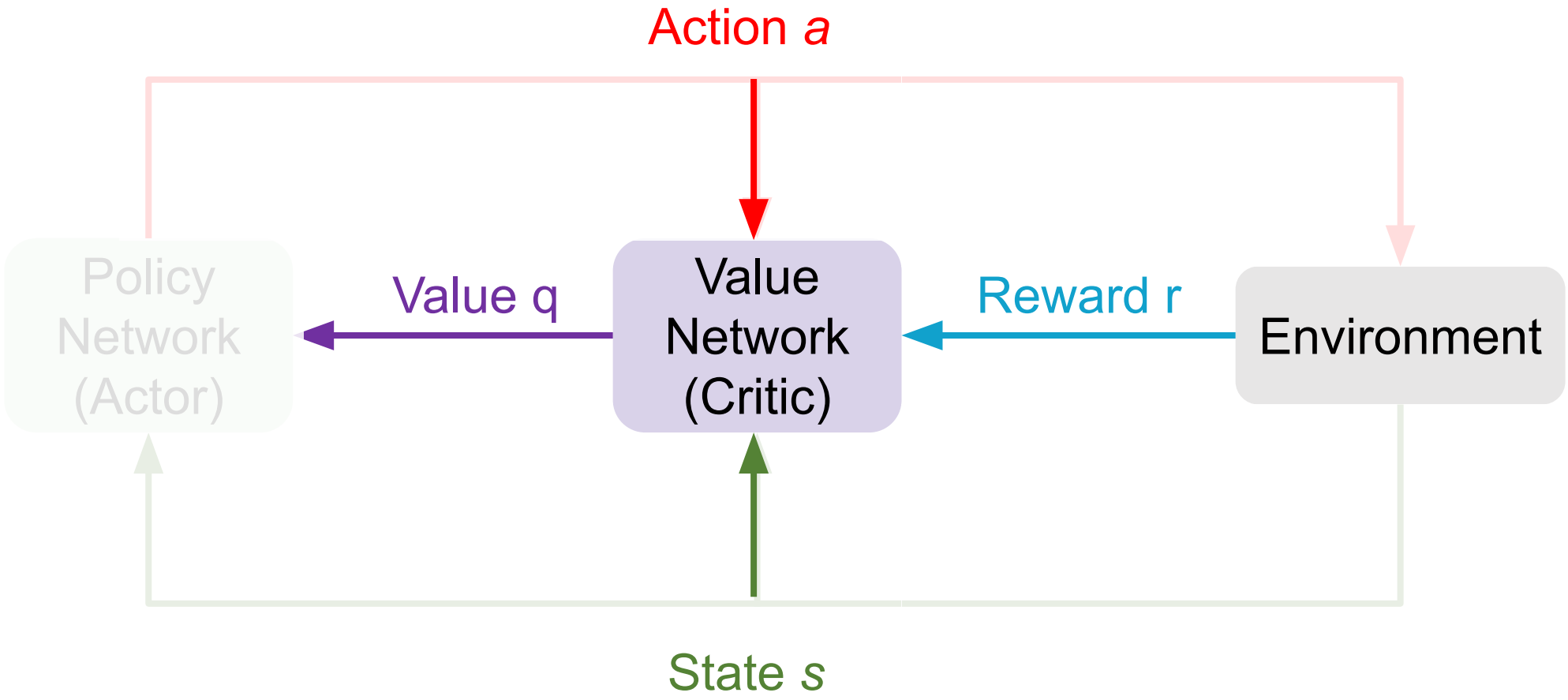
Actor, Critic and Environment



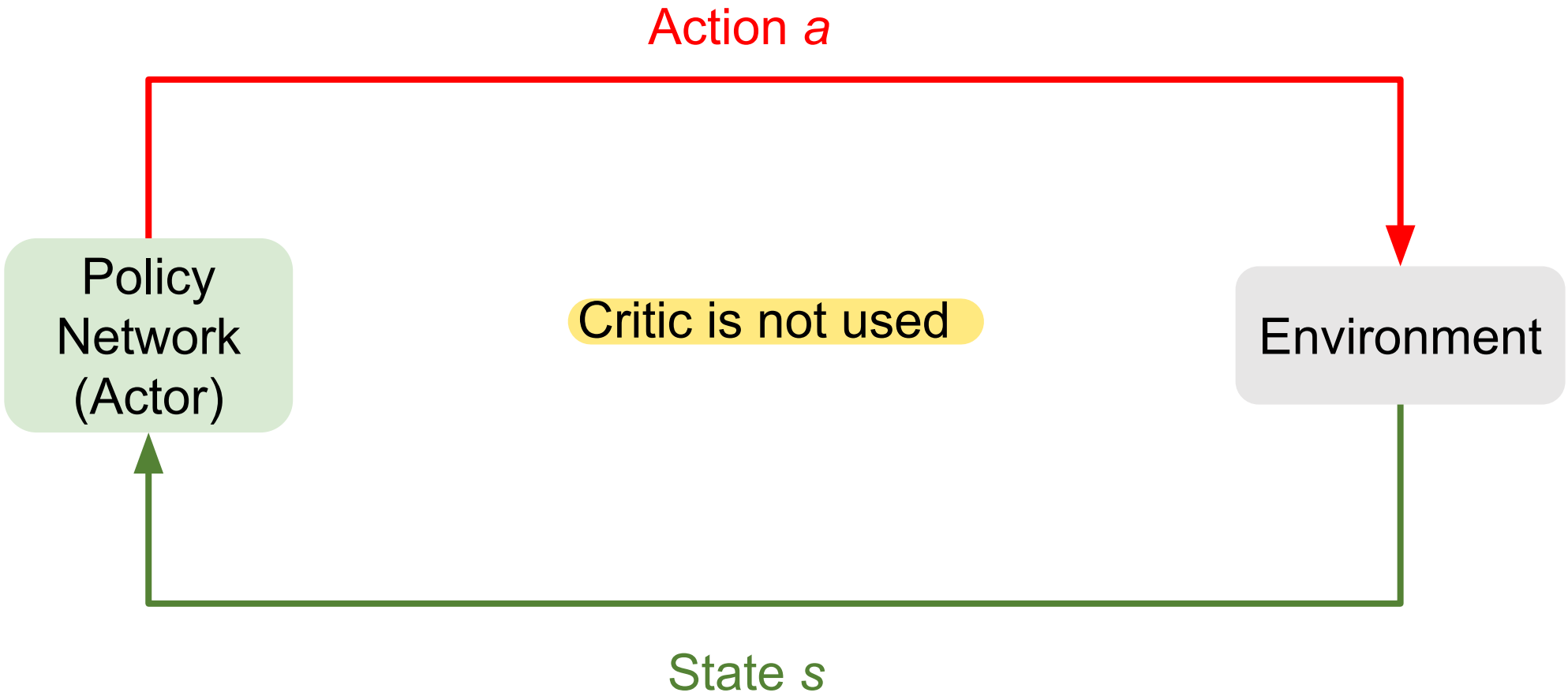
Actor and Critic: Update Actor



Actor and Critic: Update Critic



Actor and Critic: After Training



Summary of Actor-Critic Methods

1. Observe the state s_t and randomly sample $a_t \sim \pi(\cdot \mid s_t; \theta_t)$.
2. Perform a_t then environment gives new state s_{t+1} and the reward r_t.
3. Randomly sample $\tilde{a}_{t+1} \sim \pi(\cdot \mid s_{t+1}; \theta_t)$. (Do not perform a_{t+1})
4. Evaluate value network: $q_t = q(s_t, a_t; \mathbf{w}_t)$ and $q_{t+1} = q(s_{t+1}, \tilde{a}_{t+1}; \mathbf{w}_t)$.
5. Compute TD error: $\delta_t = \underbrace{q_t - (r_t + \gamma \cdot q_{t+1})}_{\text{TD target}}$.

Summary of Actor-Critic Methods

1. Observe the state s_t and randomly sample $a_t \sim \pi(\cdot \mid s_t; \theta_t)$.
2. Perform a_t then environment gives new state s_{t+1} and the reward r_t.
3. Randomly sample $\tilde{a}_{t+1} \sim \pi(\cdot \mid s_{t+1}; \theta_t)$. (Do not perform a_{t+1})
4. Evaluate value network: $q_t = q(s_t, a_t; \mathbf{w}_t)$ and $q_{t+1} = q(s_{t+1}, \tilde{a}_{t+1}; \mathbf{w}_t)$.
5. Compute TD error: $\delta_t = q_t - (r_t + \gamma \cdot q_{t+1})$.

---

6. Differentiate value network: $\nabla \mathbf{w}_t = \left. \frac{\partial q(s_t, a_t; \mathbf{w})}{\partial \mathbf{w}} \right|_{\mathbf{w}=\mathbf{w}_t}$.
7. Update value network: $\mathbf{w}_{t+1} = \mathbf{w}_t - \alpha \cdot \delta_t \cdot \nabla \mathbf{w}_t$.

Summary of Actor-Critic Methods

1. Observe the state s_t and randomly sample $a_t \sim \pi(\cdot \mid s_t; \theta_t)$.
2. Perform a_t then environment gives new state s_{t+1} and the reward r_t.
3. Randomly sample $\tilde{a}_{t+1} \sim \pi(\cdot \mid s_{t+1}; \theta_t)$. (Do not perform a_{t+1})
4. Evaluate value network: $q_t = q(s_t, a_t; \mathbf{w}_t)$ and $q_{t+1} = q(s_{t+1}, \tilde{a}_{t+1}; \mathbf{w}_t)$.
5. Compute TD error: $\delta_t = q_t - (r_t + \gamma \cdot q_{t+1})$.
6. Differentiate value network: $\nabla \mathbf{w}_t = \left. \frac{\partial q(s_t, a_t; \mathbf{w})}{\partial \mathbf{w}} \right|_{\mathbf{w}=\mathbf{w}_t}$.
7. Update value network: $\mathbf{w}_{t+1} = \mathbf{w}_t - \alpha \cdot \delta_t \cdot \nabla \mathbf{w}_t$.

---

8. Differentiate policy network: $\nabla \theta_t = \left. \frac{\partial \log \pi(a_t \mid s_t; \theta)}{\partial \theta} \right|_{\theta=\theta_t}$.
9. Update policy network: $\theta_{t+1} = \theta_t + \beta \cdot q_t \cdot \nabla \theta_t$

Problems of Training Actor-Critic Method

Problem 1: **Training Stability**.

At the beginning of training, the critic has no experience on how to judge actor's actions, and just gives random scores.

Bad feedback from critic can lead to unstable training !!!

Problem 2: **Sample Efficiency**.

During training, the collected data are usually used to update the neural network once, then they are dropped, which is inefficient of training RL agents.

Actor-Critic Method with Baseline

Policy gradient: $$\frac{\partial V^{\pi}(s)}{\partial \theta} = \mathbb{E}_{A \sim \pi}\left[\frac{\partial \log \pi(A \mid s; \theta)}{\partial \theta}(q(s, A; \mathbf{w}) - b)\right]$$

using a baseline to reduce variance from critic

Actor-Critic Method with Baseline

Policy gradient: $$\frac{\partial V^{\pi}(s)}{\partial \theta} = \mathbb{E}_{A\sim\pi}\Big[\frac{\partial \log \pi(A \mid s;\theta)}{\partial \theta}\cdot(q(s,A;\mathbf{w}) - b)\Big]$$

- Let the baseline, b, be anything independent of A.
- $$\mathbb{E}_{A\sim\pi}\Big[b\cdot\frac{\partial \log \pi(A \mid s;\theta)}{\partial \theta}\Big] = b\cdot\mathbb{E}_{A\sim\pi}\Big[\frac{\partial \log \pi(A \mid s;\theta)}{\partial \theta}\Big]$$

$$= b\cdot\sum_a \pi(a \mid s;\theta)\cdot\Big[\frac{1}{\pi(a \mid s;\theta)}\cdot\frac{\partial \pi(a \mid s;\theta)}{\partial \theta}\Big]$$

$$= b\cdot\sum_a \frac{\partial \pi(a \mid s;\theta)}{\partial \theta}$$

$$= b\cdot\frac{\partial \sum_a \pi(a \mid s;\theta)}{\partial \theta} \qquad \sum_a \pi(a \mid s;\theta) = 1$$

$$= b\cdot\frac{\partial 1}{\partial \theta} = 0$$

Actor-Critic Method with Baseline

Policy gradient: $\frac{\partial V_\pi(s)}{\partial \theta} = \mathbb{E}_{A\sim\pi}\left[\frac{\partial \log \pi(A \mid s;\theta)}{\partial \theta} \cdot (q(s, A; \mathbf{w}) - b)\right]$

- Let the baseline, b, be anything independent of A.
- $\mathbb{E}_{A\sim\pi}\left[b \cdot \frac{\partial \log \pi(A \mid s;\theta)}{\partial \theta}\right] = b \cdot \mathbb{E}_{A\sim\pi}\left[\frac{\partial \log \pi(A \mid s;\theta)}{\partial \theta}\right]$

$$= b \cdot \sum_a \pi(a \mid s;\theta) \cdot \left[\frac{1}{\pi(a \mid s;\theta)} \cdot \frac{\partial \pi(a \mid s;\theta)}{\partial \theta}\right]$$

$$= b \cdot \sum_a \frac{\partial \pi(a \mid s;\theta)}{\partial \theta}$$

$$= b \cdot \frac{\partial \sum_a \pi(a \mid s;\theta)}{\partial \theta}$$

$$= b \cdot \frac{\partial 1}{\partial \theta} = 0$$

Chain rule:

$$\frac{\partial \log \pi(\theta)}{\partial \theta} = \frac{1}{\pi(\theta)} \cdot \frac{\partial \pi(\theta)}{\partial \theta}$$

$$\sum_a \pi(a \mid s;\theta) = 1$$

Actor-Critic Method with Baseline

Policy gradient: $$\frac{\partial V_\pi(s)}{\partial \theta} = \mathbb{E}_{A\sim\pi}\Big[\frac{\partial \log \pi(A \mid s;\theta)}{\partial \theta}(q(s, A; \mathbf{w}) - b)\Big]$$

- Let the baseline, b, be anything independent of A.
- $$\mathbb{E}_{A\sim\pi}\Big[b \cdot \frac{\partial \log \pi(A \mid s;\theta)}{\partial \theta}\Big] = b \cdot \mathbb{E}_{A\sim\pi}\Big[\frac{\partial \log \pi(A \mid s;\theta)}{\partial \theta}\Big]$$
$$= b \cdot \sum_a \cancel{\pi(a \mid s;\theta)} \cdot \Big[\frac{1}{\cancel{\pi(a \mid s;\theta)}} \cdot \frac{\partial \pi(a \mid s;\theta)}{\partial \theta}\Big]$$
$$= b \cdot \sum_a \frac{\partial \pi(a \mid s;\theta)}{\partial \theta}$$
$$= b \cdot \frac{\partial \sum_a \pi(a \mid s;\theta)}{\partial \theta} \qquad \sum_a \pi(a \mid s;\theta) = 1$$
$$= b \cdot \frac{\partial 1}{\partial \theta} = 0$$

Actor-Critic Method with Baseline

Policy gradient: $$\frac{\partial V_\pi(s)}{\partial \theta} = \mathbb{E}_{A\sim\pi}\Big[\frac{\partial \log \pi(A \mid s;\theta)}{\partial \theta}(q(s, A; \mathbf{w}) - b)\Big]$$

- Let the baseline, b, be anything independent of A.
- $$\mathbb{E}_{A\sim\pi}\Big[b \cdot \frac{\partial \log \pi(A \mid s;\theta)}{\partial \theta}\Big] = b \cdot \mathbb{E}_{A\sim\pi}\Big[\frac{\partial \log \pi(A \mid s;\theta)}{\partial \theta}\Big]$$
$$= b \cdot \sum_a \pi(a \mid s;\theta) \cdot \Big[\frac{1}{\pi(a \mid s;\theta)} \cdot \frac{\partial \pi(a \mid s;\theta)}{\partial \theta}\Big]$$
$$= b \cdot \sum_a \frac{\partial \pi(a \mid s;\theta)}{\partial \theta}$$
$$= b \cdot \frac{\partial \sum_a \pi(a \mid s;\theta)}{\partial \theta} \qquad \sum_a \pi(a \mid s;\theta) = 1$$
$$= b \cdot \frac{\partial 1}{\partial \theta} = 0$$

Actor-Critic Method with Baseline

Policy gradient: $$\frac{\partial V_\pi(s)}{\partial \theta} = \mathbb{E}_{A\sim\pi}\Big[\frac{\partial \log \pi(A \mid s;\theta)}{\partial \theta}(q(s,A;\mathbf{w}) - b)\Big]$$

- Let the baseline, b, be anything independent of A.
- $$\mathbb{E}_{A\sim\pi}\Big[b \cdot \frac{\partial \log \pi(A \mid s;\theta)}{\partial \theta}\Big] = b \cdot \mathbb{E}_{A\sim\pi}\Big[\frac{\partial \log \pi(A \mid s;\theta)}{\partial \theta}\Big]$$
$$= b \cdot \sum_a \pi(a \mid s;\theta) \cdot \Big[\frac{1}{\pi(a \mid s;\theta)} \cdot \frac{\partial \pi(a \mid s;\theta)}{\partial \theta}\Big]$$
$$= b \cdot \sum_a \frac{\partial \pi(a \mid s;\theta)}{\partial \theta}$$
$$= b \cdot \frac{\partial \sum_a \pi(a \mid s;\theta)}{\partial \theta} \qquad \sum_a \pi(a \mid s;\theta) = 1$$
$$= b \cdot \frac{\partial 1}{\partial \theta} = 0$$

Actor-Critic Method with Baseline

If b is independent of A, then $\mathbb{E}_{A\sim\pi}\left[b \cdot \frac{\partial \log \pi(A \mid s;\theta)}{\partial\theta}\right] = 0$.

- Policy gradient with baseline:

$$\frac{\partial V(s)}{\partial\theta} = \mathbb{E}_{A\sim\pi}\left[\frac{\partial \log \pi(A \mid s;\theta)}{\partial\theta} \cdot q(s, A; \mathbf{w})\right] - \underbrace{\mathbb{E}_{A\sim\pi}\left[\frac{\partial \log \pi(A \mid s;\theta)}{\partial\theta} \cdot b\right]}_{= 0}$$

$$= \underbrace{\mathbb{E}_{A\sim\pi}\left[\frac{\partial \log \pi(A \mid s;\theta)}{\partial\theta}(q(s, A; \mathbf{w}) - b)\right]}_{\text{policy gradient with baseline}}$$

must be valid

valid if baseline is independet of A

Actor-Critic Method with Baseline

Policy gradient: $$\frac{\partial V^{\pi}(s)}{\partial \theta} = \mathbb{E}_{A \sim \pi}\left[\frac{\partial \log \pi(A \mid s; \theta)}{\partial \theta} (q(s, A; \mathbf{w}) - b)\right]$$

Actor-Critic Method with Baseline

Policy gradient: $$\frac{\partial V^\pi(s)}{\partial \theta} = \mathbb{E}_{A\sim\pi}\Big[\frac{\partial \log \pi(A \mid s;\theta)}{\partial \theta}(q(s,A;\mathbf{w}) - b)\Big]$$

$$= g(a_t)$$

- Whatever b (independent of A_t) is, the policy gradient $\mathbb{E}_{A_t\sim\pi}\big[g(A_t)\big]$ remains the same.
- However, b affects $g(a_t)$.
- A good b leads to small variance and speeds up convergence.

Question: How to choose a baseline?

State-value Function as Baseline

Policy gradient: $$\frac{\partial V^{\pi}(s)}{\partial \theta} = \mathbb{E}_{A\sim\pi}\Big[\frac{\partial \log \pi(A \mid s;\theta)}{\partial \theta}(q(s,A;\mathbf{w}) - b)\Big]$$

b is the state-value function: $b = V_{\pi}(s_t)$ (average performance for this state)

- By definition, $V^{\pi}(s_t)$ is close to $Q^{\pi}(s_t, A_t)$:

$$V^{\pi}(s_t) = \mathbb{E}_{A_t}\big[Q^{\pi}(s_t, A_t)\big]$$

↳ $(q - b) = (q - V)$

see difference from q to average

Advantage Actor Critic (A2C)

Policy gradient: $\frac{\partial V^{\pi}(s)}{\partial \theta} = \mathbb{E}_{A \sim \pi}\left[\frac{\partial \log \pi(A \mid s; \theta)}{\partial \theta}(Q^{\pi}(s, A) - V^{\pi}(s))\right]$

Advantage Actor Critic (A2C)

Policy gradient: $\frac{\partial V^{\pi}(s)}{\partial \theta} = \mathbb{E}_{A \sim \pi}\left[\frac{\partial \log \pi(A \mid s; \theta)}{\partial \theta}\left(Q^{\pi}(s, A) - V^{\pi}(s)\right)\right]$

Advantage function

Advantage function: $\mathcal{A}(s, a) = Q^{\pi}(s, a) - V^{\pi}(s)$

q value of action a in state s

average value of state s

Advantage Actor Critic (A2C)

Policy gradient: $\frac{\partial V^{\pi}(s)}{\partial \theta} = \mathbb{E}_{A \sim \pi}\left[\frac{\partial \log \pi(A \mid s; \theta)}{\partial \theta}\left(Q^{\pi}(s, A) - V^{\pi}(s)\right)\right]$

Advantage function

Advantage function: $\mathcal{A}(s, a) = Q^{\pi}(s, a) - V^{\pi}(s)$

q value of action a in state s

average value of state s

- If $\mathcal{A}(s, a) > 0$, action a is better than average actions, gradient (+) is pushed in that direction.
- If $\mathcal{A}(s, a) < 0$, action a is worse than average actions, gradient (-) is pushed in the opposite direction.

Advantage Actor Critic (A2C)

Policy gradient: $\frac{\partial V^{\pi}(s)}{\partial \theta} = \mathbb{E}_{A \sim \pi}\left[\frac{\partial \log \pi(A \mid s; \theta)}{\partial \theta}\left(Q^{\pi}(s, A) - V^{\pi}(s)\right)\right]$

Advantage function

Advantage function: $\mathcal{A}(s, a) = Q^{\pi}(s, a) - V^{\pi}(s)$

Unknown

Advantage Actor Critic (A2C)

Policy gradient: $\frac{\partial V^\pi(s)}{\partial \theta} = \mathbb{E}_{A\sim\pi}\left[\frac{\partial \log \pi(A \mid s;\theta)}{\partial \theta}\left(Q^\pi(s,A) - V^\pi(s)\right)\right]$

Advantage function

Advantage function: $\mathcal{A}(s,\, a) = Q^\pi(s,\, a) - V^\pi(s)$

Unknown

Fortunately, we know

$$Q^\pi(s_t, a_t) = \mathbb{E}[R_t + \gamma V^\pi(S_{t+1})]$$

Advantage Actor Critic (A2C)

Policy gradient: $\frac{\partial V^\pi(s)}{\partial \theta} = \mathbb{E}_{A\sim\pi}\left[\frac{\partial \log \pi(A \mid s;\theta)}{\partial \theta}\left(Q^\pi(s,A) - V^\pi(s)\right)\right]$

Advantage function

Advantage function: $\mathcal{A}(s,\, a) = Q^\pi(s,\, a) - V^\pi(s)$

Unknown

Fortunately, we know

$$Q^\pi(s_t, a_t) = \mathbb{E}[R_t + \gamma V^\pi(S_{t+1})]$$

$$\simeq r_t + \gamma V^\pi(s_{t+1})$$ an estimate

$$\mathcal{A}(s_t,\, a_t) = r_t + \gamma V^\pi(s_{t+1}) - V^\pi(s_t)$$ TD Error

Advantage Actor Critic (A2C)

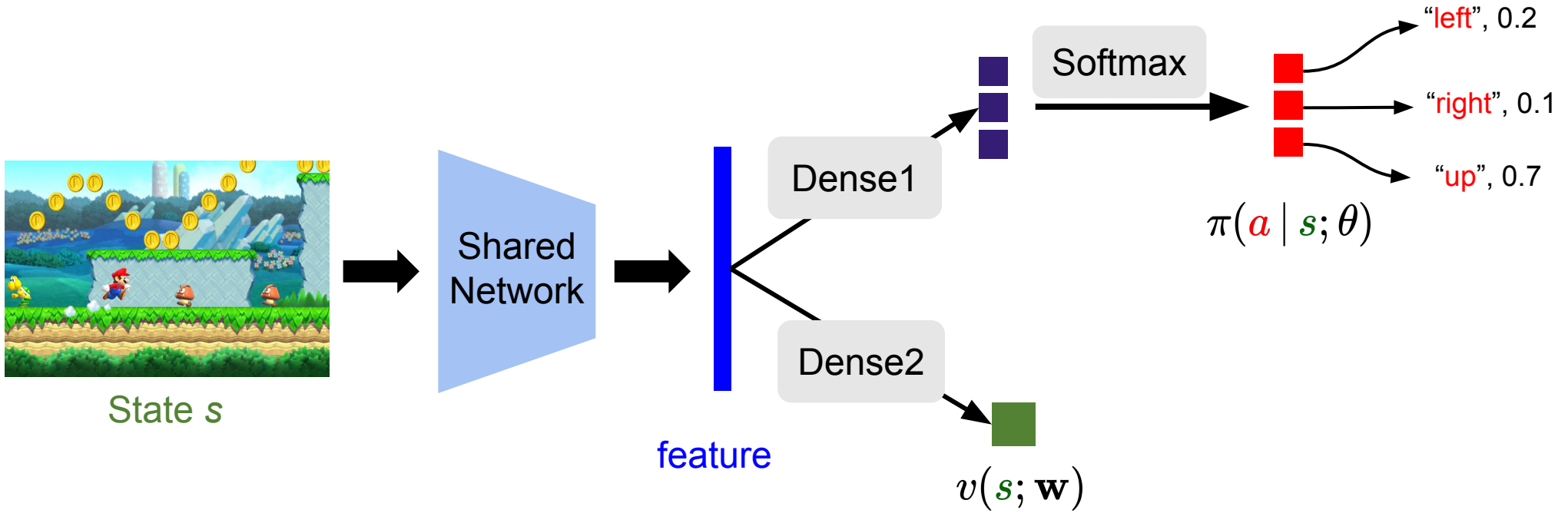
Policy gradient: $\frac{\partial V^{\pi}(s_t)}{\partial \theta} \simeq \mathbb{E}_{A_t \sim \pi}\left[\frac{\partial \log \pi(A_t \mid s_t; \theta)}{\partial \theta}(r_t + \gamma V^{\pi}(s_{t+1}) - V^{\pi}(s_t))\right]$

Unknown

Advantage Actor Critic (A2C)

Policy gradient: $\frac{\partial V^{\pi}(s_t)}{\partial \theta} \simeq \mathbb{E}_{A_t \sim \pi}\left[\frac{\partial \log \pi(A_t \mid s_t; \theta)}{\partial \theta}(r_t + \gamma v(s_{t+1}; \mathbf{w}) - v(s_t; \mathbf{w}))\right]$

- Approximate $V^{\pi}(s)$ by the value network $v(s; \mathbf{w})$ (critic).



A2C v.s. Vanilla Actor Critic

- Vanilla Actor Critic: (default)

$$\frac{\partial V^{\pi}(s_t)}{\partial \theta} \simeq \mathbb{E}_{A_t \sim \pi}\Big[\frac{\partial \log \pi(A_t \mid s_t; \theta)}{\partial \theta} \cdot q(s, A; \mathbf{w})\Big]$$

$\pi(a \mid s; \theta)$ and $q(s, A; \mathbf{w})$ are coupled by action, which is harder to train.

- Advantage Actor Critic (A2C):

$$\frac{\partial V^{\pi}(s_t)}{\partial \theta} \simeq \mathbb{E}_{A_t \sim \pi}\Big[\frac{\partial \log \pi(A_t \mid s_t; \theta)}{\partial \theta} \cdot (r_t + \gamma v(s_{t+1}; \mathbf{w}) - v(s_t; \mathbf{w}))\Big]$$

$\pi(a \mid s; \theta)$ and $v(s; \mathbf{w})$ are decoupled, which is easier to train.

Training A2C

1. Observe a transition (s_t, a_t, r_t, s_{t+1})
2. TD target: $y_t = r_t + \gamma v(s_{t+1}; \mathbf{w})$
3. TD error: $\delta_t = y_t - v(s_t; \mathbf{w})$
4. Update the policy network (actor) by:

$$\theta \leftarrow \theta + \beta \cdot \delta_t \cdot \frac{\partial \log \pi(a_t \mid s_t; \theta)}{\partial \theta}$$

5. Update the value network (critic) by:

$$\mathbf{w} \leftarrow \mathbf{w} - \alpha \cdot \delta_t \cdot \frac{\partial v(s_t; \mathbf{w})}{\partial \mathbf{w}}$$

Training A2C

1. Observe a transition (s_t, a_t, r_t, s_{t+1})

2. TD target: $y_t = r_t + \gamma v(s_{t+1}; \mathbf{w})$

3. TD error: $\delta_t = y_t - v(s_t; \mathbf{w})$

difference

4. Update the policy network (actor) by:

$$\theta \leftarrow \theta + \beta \cdot \delta_t \cdot \frac{\partial \log \pi(a_t \mid s_t; \theta)}{\partial \theta}$$

5. Update the value network (critic) by:

$$\mathbf{w} \leftarrow \mathbf{w} - \alpha \cdot \delta_t \cdot \frac{\partial v(s_t; \mathbf{w})}{\partial \mathbf{w}}$$

Generalized Advantage Estimation (GAE)

Policy gradient: $\nabla V_\theta(s) = \mathbb{E}_{A\sim\pi_\theta}\left[\mathcal{A}(s, a)\nabla \log \pi_\theta(A \mid s)\right]$

Advantage function: $\mathcal{A}(s, a) = Q^\pi(s, a) - V^\pi(s)$

By comparing to the average, $\mathcal{A}(s, a)$ tells us how good the action a is,

very important for training agents

In reality:

$$\mathcal{A}(s_t, a_t) \simeq \underbrace{r_t}_{\text{real}} + \gamma \underbrace{V^\pi(s_{t+1})}_{\text{guess}} - \underbrace{V^\pi(s_t)}_{\text{guess}} = \delta_t$$

(from env.)

Generalized Advantage Estimation (GAE)

Policy gradient: $\nabla V_\theta(s) = \mathbb{E}_{A\sim\pi_\theta}\left[\mathcal{A}(s,\, a)\nabla \log \pi_\theta(A \mid s)\right]$

$$\mathcal{A}_t^{(1)} = r_t + \gamma V^\pi(s_{t+1}) - V^\pi(s_t)$$

$$\mathcal{A}_t^{(2)} = r_t + \gamma r_{t+1} + \gamma^2 V^\pi(s_{t+2}) - V^\pi(s_t)$$

$$\cdots$$

$$\mathcal{A}_t^{(k)} = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots - V^\pi(s_t)$$

- k is small, low variance but high bias;
- k is large, low bias but high variance.

↑ depend on more steps in the future

↳ Trade-off needed

Generalized Advantage Estimation (GAE)

Policy gradient: $\nabla V_\theta(s) = \mathbb{E}_{A\sim\pi_\theta}\left[\mathcal{A}(s, a)\nabla \log \pi_\theta(A \mid s)\right]$

$$\mathcal{A}_t^{(1)} = r_t + \gamma V^\pi(s_{t+1}) - V^\pi(s_t)$$

$$\mathcal{A}_t^{(2)} = r_t + \gamma r_{t+1} + \gamma^2 V^\pi(s_{t+2}) - V^\pi(s_t)$$

$$\cdots$$

$$\mathcal{A}_t^{(k)} = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots - V^\pi(s_t)$$

- k is small, low variance but high bias;
- k is large, low bias but high variance.

Bias/Variance trade-off:

$$\begin{aligned}\mathcal{A}_t^{GAE(\gamma,\lambda)} &= (1-\lambda)\left(\mathcal{A}_t^{(1)} + \lambda\mathcal{A}_t^{(2)} + \lambda^2\mathcal{A}_t^{(3)} + \cdots\right)\\ &= (1-\lambda)\left(\delta_t + \lambda(\delta_t + \gamma\delta_{t+1}) + \lambda^2\left(\delta_t + \gamma\delta_{t+1} + \gamma^2\delta_{t+2}\right) + \cdots\right)\\ &\cdots \quad [0,\gamma] \quad [0,1]\\ &= \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+l}\end{aligned}$$

Tradeoff-variables decide less bias or less variance

Generalized Advantage Estimation (GAE)

Policy gradient: $\nabla V_\theta(s) = \mathbb{E}_{A\sim\pi_\theta}\left[\mathcal{A}(s, a)\nabla \log \pi_\theta(A \mid s)\right]$

Bias/Variance trade-off:

$$\mathcal{A}_t^{GAE(\gamma,\lambda)} = \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+l}$$

γ and λ are hyperparameters during training

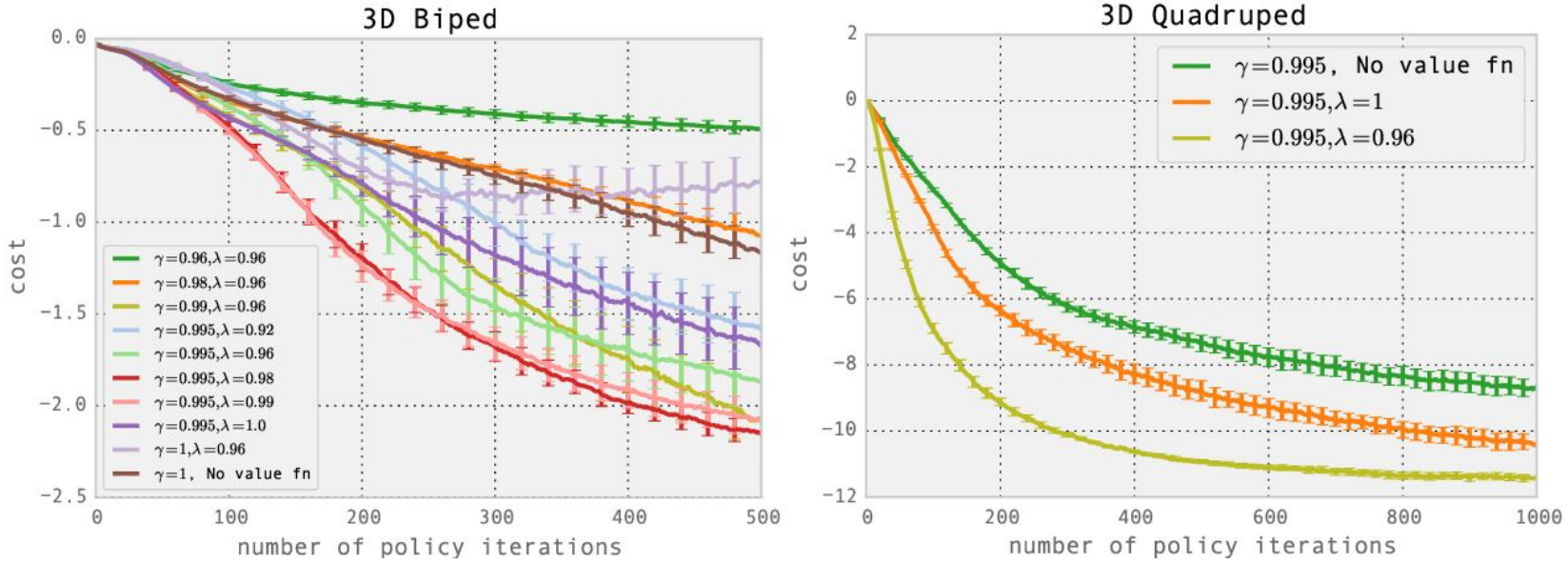
Generalized Advantage Estimation (GAE)

Policy gradient: $\nabla V_\theta(s) = \mathbb{E}_{A\sim\pi_\theta}\left[\mathcal{A}(s, a)\nabla \log \pi_\theta(A \mid s)\right]$

Bias/Variance trade-off:

$$\mathcal{A}_t^{GAE(\gamma,\lambda)} = \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+l}$$

γ and λ are hyperparameters during training



On-Policy vs. Off-Policy

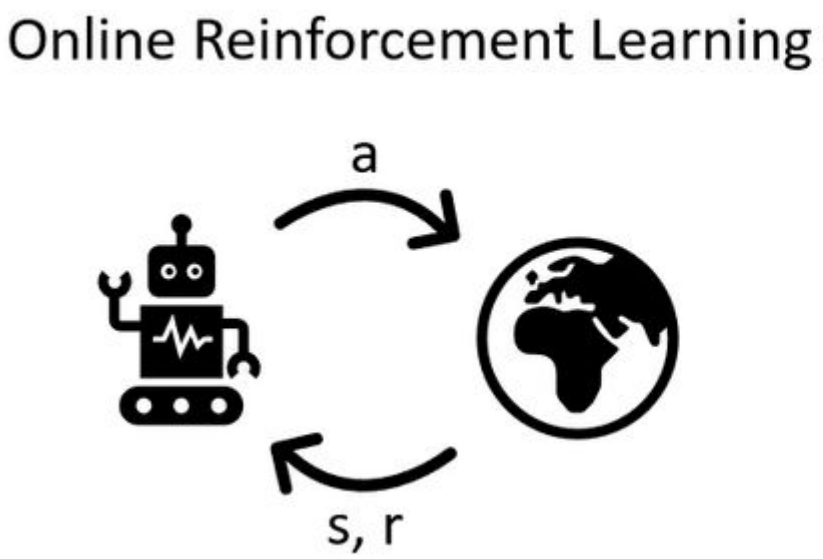
On-Policy: the RL agent interacts with environment directly (online), and improve its policy with collected data.

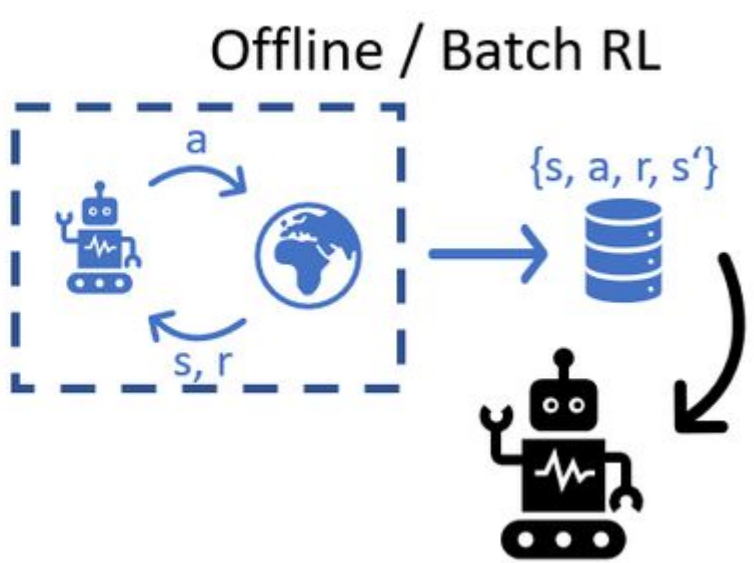
Dynamic behaviour with environment

Real-time Adaptation, Exploration, Sample inefficiency (data not reused)

Off-Policy: other agents (usually unknown) interact with environment to collect dataset. The agent improve its policy from the dataset.

Sample efficiency, Lack of Exploration, Distribution Mismatch





From on-policy to off-policy

Policy gradient: $\nabla V_\theta(s) = \mathbb{E}_{A\sim\pi_\theta}\left[\mathcal{A}(s, A)\nabla \log \pi_\theta(A \mid s)\right]$

$= g(a)$

$g(a)$ is an unbiased estimate because a is sampled from π_θ;
but after the first update: $\pi_{\theta_{old}} \rightarrow \pi_{\theta_{new}}$

$g(a)$ is not an unbiased estimate of $\nabla V_\theta(s) = \mathbb{E}_{A\sim\pi_{\theta_{old}}}\left[\mathcal{A}(s, A)\nabla \log \pi_{\theta_{new}}(A \mid s)\right]$

Question: How to reuse data that is sampled from old policy?

Importance Sampling

Question: How to reuse data that is sampled from $\pi_{\theta_{old}}$ to update $\pi_{\theta_{new}}$?

$$
\begin{aligned}
\nabla V_\theta(s) &= \mathbb{E}_{A\sim\pi_{\theta_{old}}}\left[f(\cdot)\right] \\
&= \sum_{a\in A} \pi_{\theta_{old}} \cdot f(\cdot) \\
&= \sum_{a\in A} \pi_{\theta_{new}} \cdot \frac{\pi_{\theta_{old}}}{\pi_{\theta_{new}}} \cdot f(\cdot) \\
&= \mathbb{E}_{A\sim\pi_{\theta_{new}}}\left[\frac{\pi_{\theta_{old}}}{\pi_{\theta_{new}}} \cdot f(\cdot)\right]
\end{aligned}
$$

- Sample data from $\pi_{\theta_{old}}$
- Use data to train $\pi_{\theta_{new}}$ many times

Importance Sampling

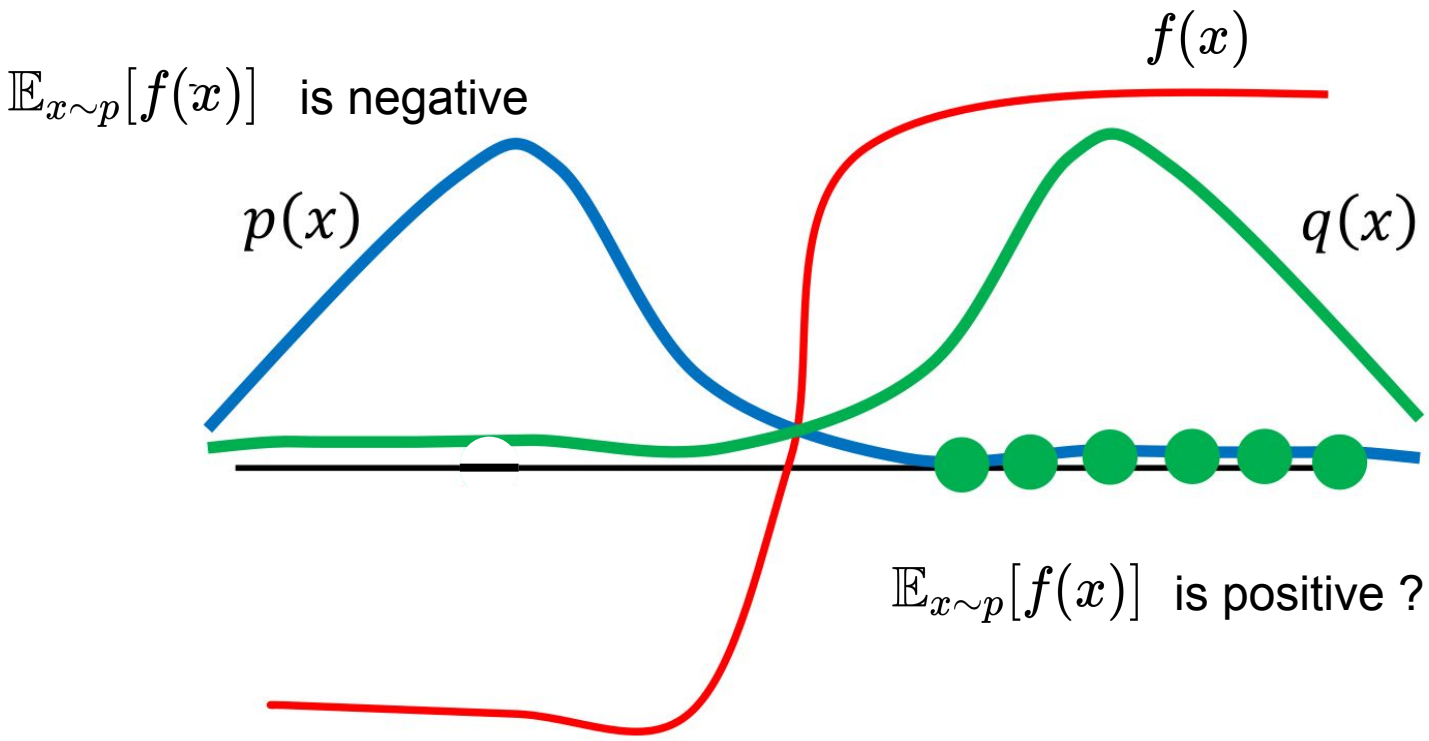
Question: How to reuse data that is sampled from $\pi_{\theta_{old}}$ to update $\pi_{\theta_{new}}$?

$$
\begin{aligned}
\nabla V_\theta(s) &= \mathbb{E}_{A\sim\pi_{\theta_{old}}}\left[f(\cdot)\right] \\
&= \sum_{a\in A} \pi_{\theta_{old}} \cdot f(\cdot) \\
&= \sum_{a\in A} \pi_{\theta_{new}} \cdot \frac{\pi_{\theta_{old}}}{\pi_{\theta_{new}}} \cdot f(\cdot) \\
&= \mathbb{E}_{A\sim\pi_{\theta_{new}}}\left[\frac{\pi_{\theta_{old}}}{\pi_{\theta_{new}}} \cdot f(\cdot)\right]
\end{aligned}
$$

- Sample data from $\pi_{\theta_{old}}$
- Use data to train $\pi_{\theta_{new}}$ many times

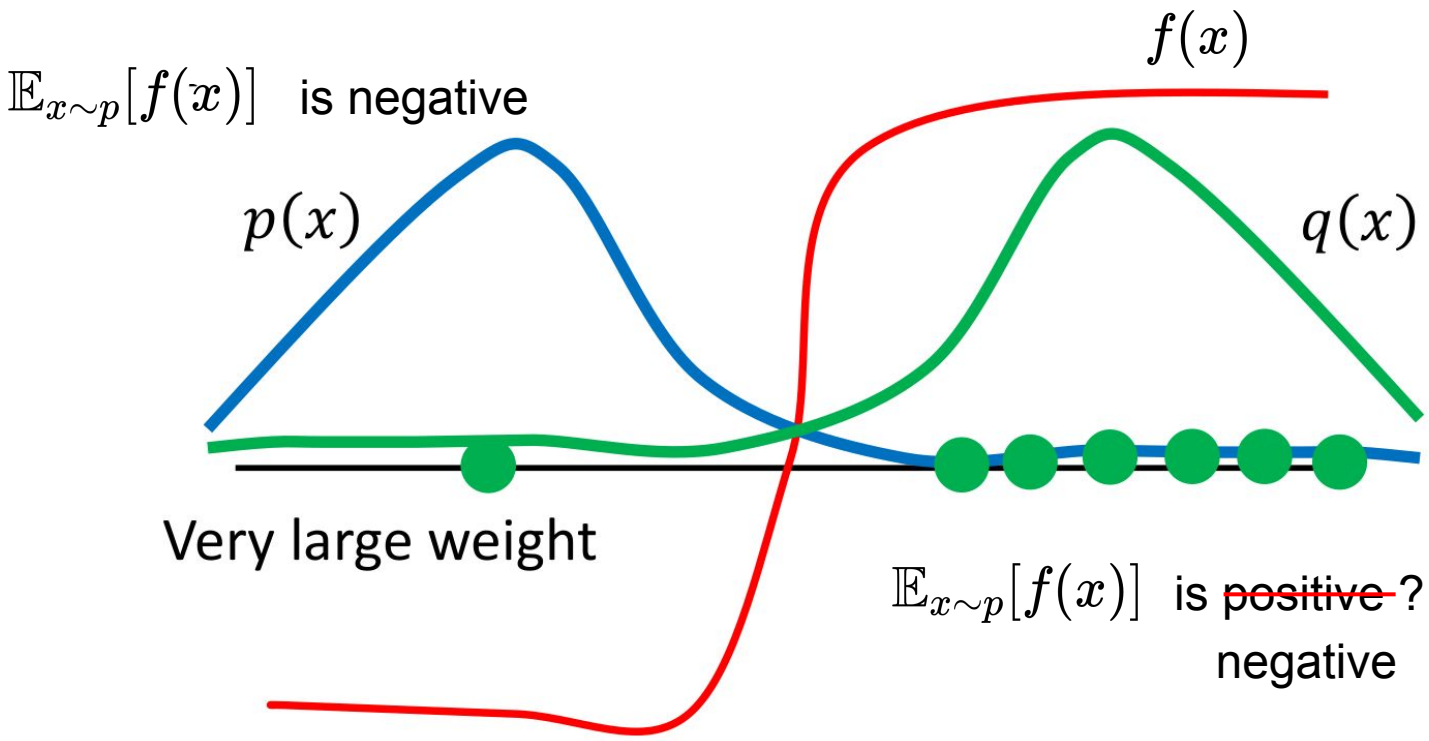
Issue of Importance Sampling

$$\mathbb{E}_{x\sim p}[f(x)] = \mathbb{E}_{x\sim q}\left[\frac{p(x)}{q(x)}f(x)\right]$$



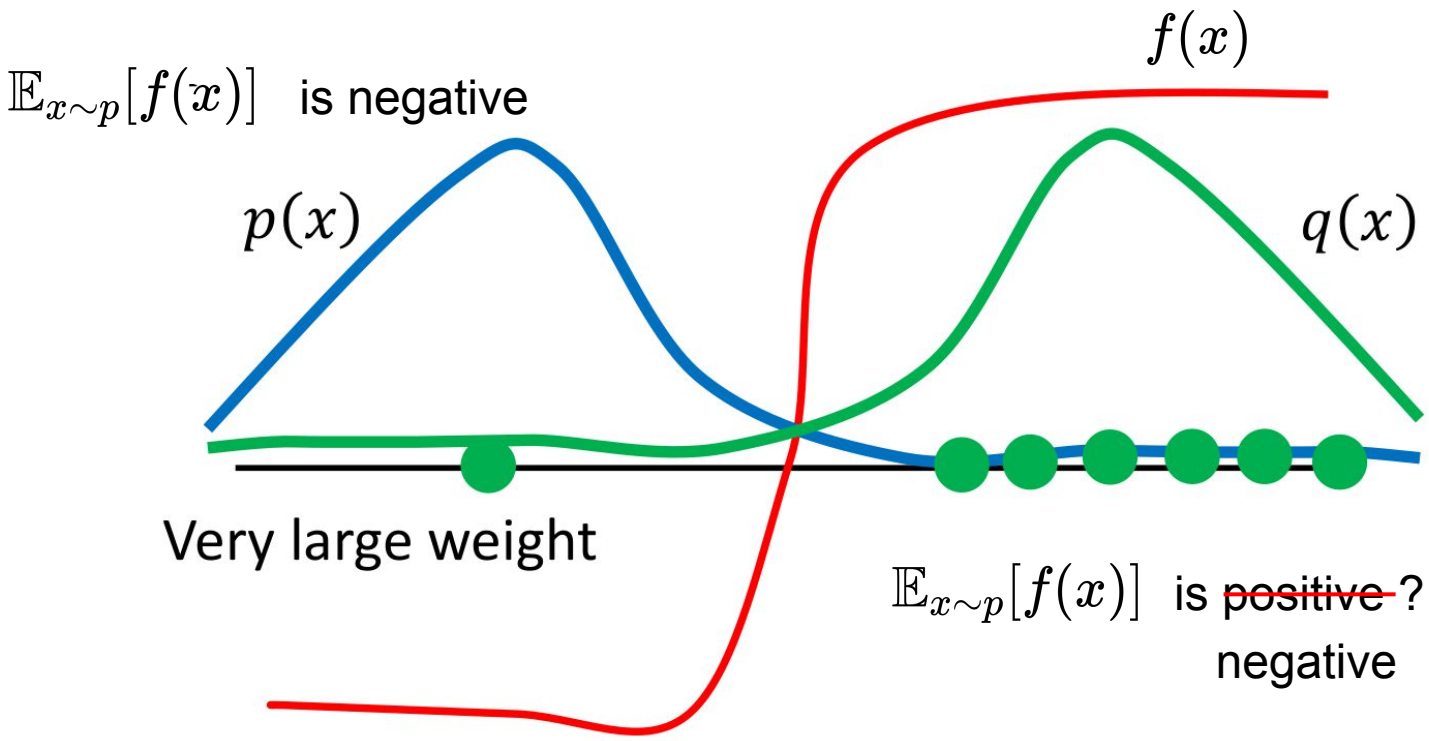
Issue of Importance Sampling

$$\mathbb{E}_{x\sim p}[f(x)] = \mathbb{E}_{x\sim q}\left[\frac{p(x)}{q(x)}f(x)\right]$$



Issue of Importance Sampling

$$\mathbb{E}_{x\sim p}[f(x)] = \mathbb{E}_{x\sim q}\left[\frac{p(x)}{q(x)}f(x)\right]$$



high variance if two distribution are very different from each other.

Add Constraints

PPO-v1:

$$J_{PPO}^{\theta'}(\theta) = J^{\theta'}(\theta) - \beta KL(\pi_\theta, \pi_{\theta'})$$ → punish if two distributions are very different

- $If\ KL(\pi_\theta, \pi_{\theta'}) > KL_{\max}$, increase β
- $If\ KL(\pi_\theta, \pi_{\theta'}) < KL_{\min}$, decrease β

It is hard to choose a single value of β that performs well across different problems, or even within a single problem.

Add Constraints

PPO-v1:

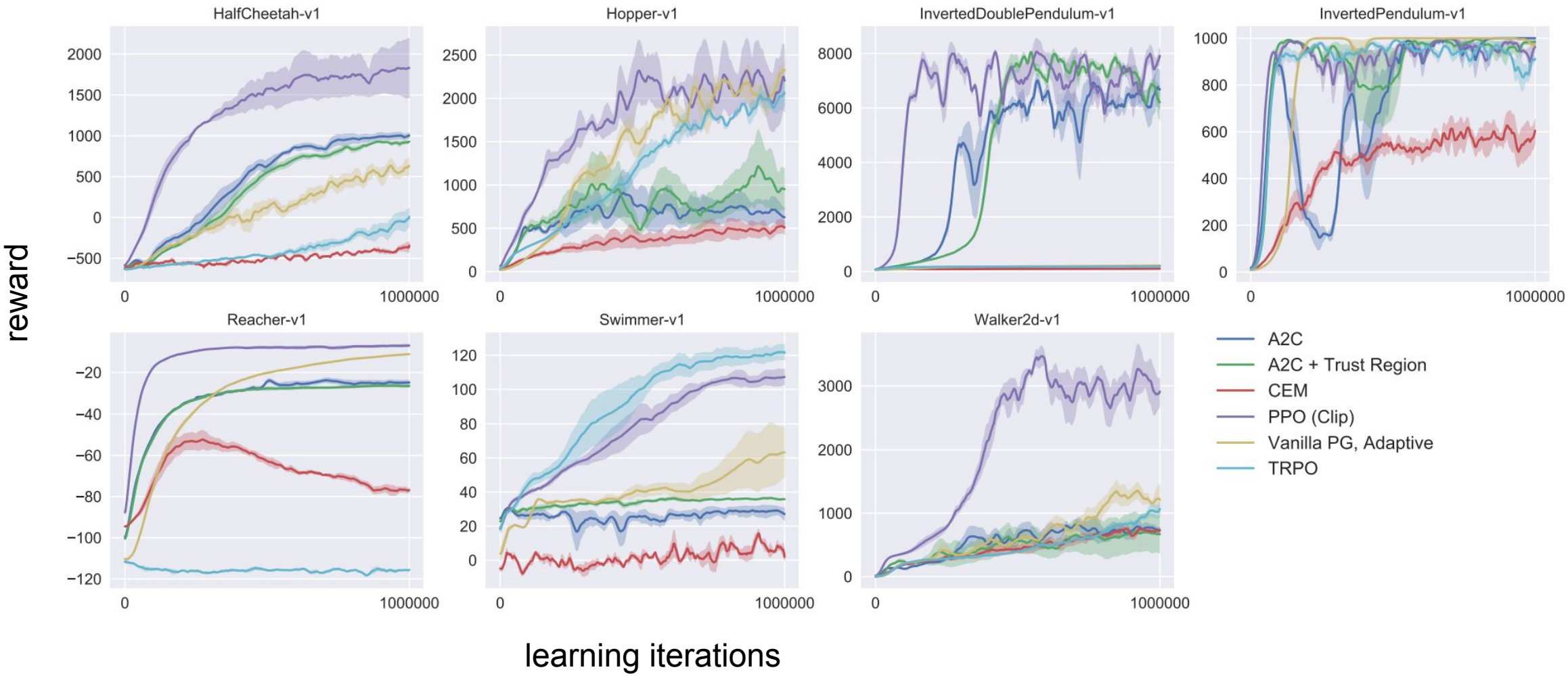
$$J_{PPO}^{\theta'}(\theta) = J^{\theta'}(\theta) - \beta KL(\pi_\theta, \pi_{\theta'})$$ → punish if two distributions are very different

- $If\ KL(\pi_\theta, \pi_{\theta'}) > KL_{\max}$, increase β
- $If\ KL(\pi_\theta, \pi_{\theta'}) < KL_{\min}$, decrease β

PPO-v2:

$$J_{PPO2}^{\theta'}(\theta) \simeq \mathbb{E}\Big[\min\Big(\frac{\pi_\theta}{\pi_{\theta'}}\mathcal{A}(s, a), \underline{\text{clip}}\Big(\frac{\pi_\theta}{\pi_{\theta'}}, 1-\epsilon, 1+\epsilon\Big)\mathcal{A}(s, a)\Big)\Big]$$

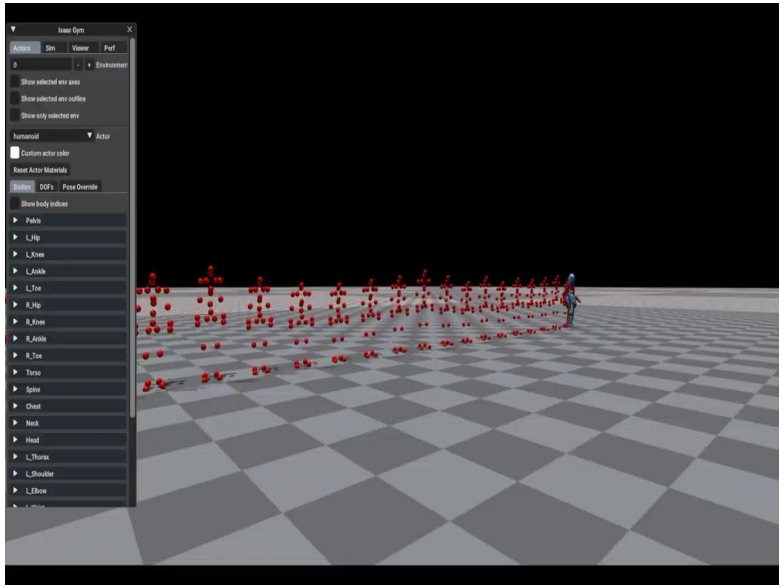
Comparison of PPO with others



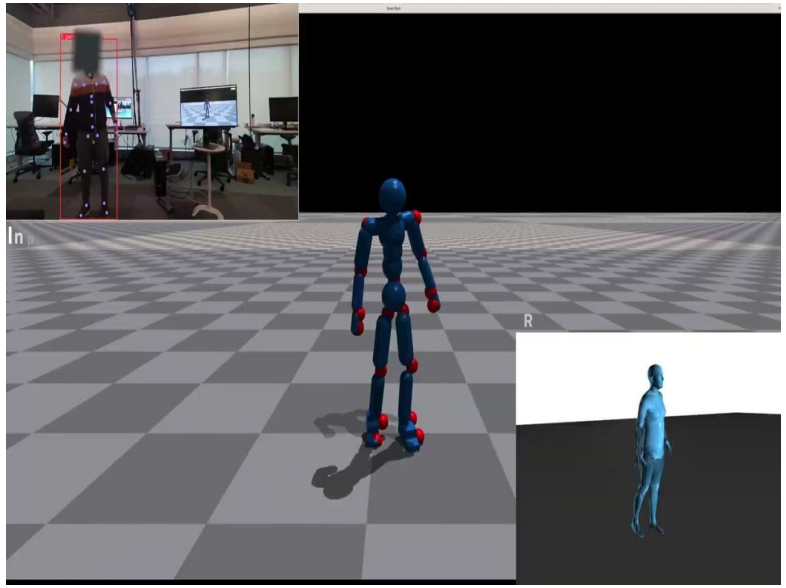
Schulman, Wolski, Dhariwal,Radford and Klimov. Proximal Policy Optimization algorithms. 2017

RL for Locomotion

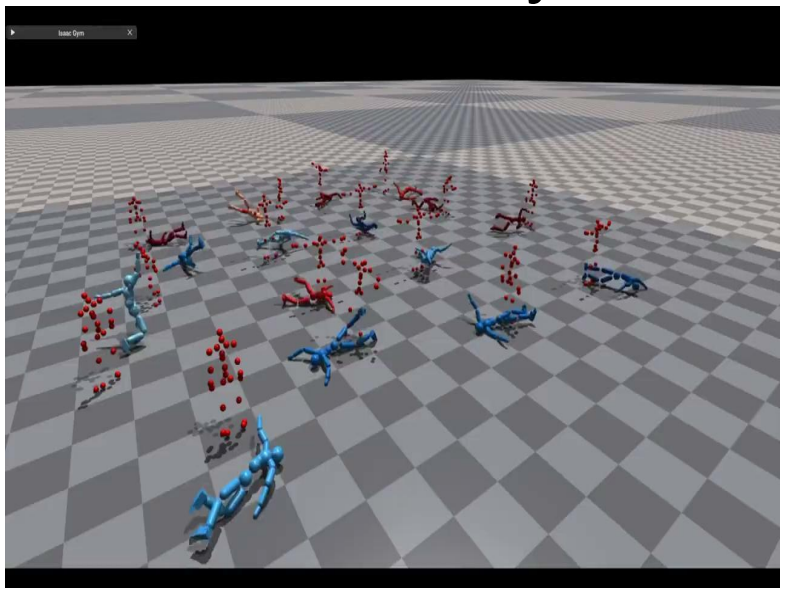
fair walk



real-time control



fall recovery



Summary

Actor-Critic Method

- $V(s;\theta,\mathbf{w}) = \sum_a \pi(a \mid s;\theta) \cdot q(s,\ a;\mathbf{w})$
- policy network (actor): $\pi(a \mid s;\theta)$
- value network (critic): $q(s,\ a;\mathbf{w})$

Update the parameter θ and $\mathbf{w}$

1. Observe a transition (s_t, a_t, r_t, s_{t+1})
2. Evaluate the value network:

   $q_t = q(s_t,\ a_t;\mathbf{w}_t)$ & $q_{t+1} = q(s_{t+1},\ \tilde{a}_{t+1};\mathbf{w}_t)$
3. Compute TD error:

   $$\delta_t = q_t - (r_t + \gamma \cdot q_{t+1})$$
4. Update value network:

   $$\mathbf{w}_{t+1} = \mathbf{w}_t - \alpha \cdot \delta_t \cdot \nabla \mathbf{w}_t$$
5. Update policy network.

   $$\theta_{t+1} = \theta_t + \beta \cdot q_t \cdot \nabla \theta_t$$

Summary

Policy gradient: $\frac{\partial V_\pi(s)}{\partial \theta} = \mathbb{E}_{A\sim\pi}\left[\frac{\partial \log \pi(A \mid s;\theta)}{\partial \theta}(Q_\pi(s,A) - V_\pi(s))\right]$

Advantage function

Advantage function: $\mathcal{A}(s, a) = Q_\pi(s, a) - V_\pi(s)$

How good the action a comparing with the average.

In reality, we estimate advantage function by:

$$\mathcal{A}(s_t, a_t) \simeq \underbrace{r_t}_{\text{real}} + \gamma \underbrace{V_\pi(s_{t+1})}_{\text{guess}} - \underbrace{V_\pi(s_t)}_{\text{guess}} = \delta_t$$

Summary

Policy gradient with advantage function: $\nabla V_\theta(s) = \mathbb{E}_{A\sim\pi_\theta}\left[\mathcal{A}(s, a)\nabla \log \pi_\theta(A \mid s)\right]$

$$\mathcal{A}_t^{(1)} = r_t + \gamma V_\pi(s_{t+1}) - V_\pi(s_t)$$

$$\mathcal{A}_t^{(2)} = r_t + \gamma r_{t+1} + \gamma^2 V(s_{t+2}) - V_\pi(s_t)$$

$$\cdots$$

$$\mathcal{A}_t^{(k)} = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots - V_\pi(s_t)$$

- k is small, low variance but high bias;
- k is large, low bias but high variance.

Summary

Policy gradient with advantage function: $\nabla V_\theta(s) = \mathbb{E}_{A\sim\pi_\theta}\left[\mathcal{A}(s, a)\nabla \log \pi_\theta(A \mid s)\right]$

$$\mathcal{A}_t^{(1)} = r_t + \gamma V_\pi(s_{t+1}) - V_\pi(s_t)$$

$$\mathcal{A}_t^{(2)} = r_t + \gamma r_{t+1} + \gamma^2 V(s_{t+2}) - V_\pi(s_t)$$

$$\cdots$$

$$\mathcal{A}_t^{(k)} = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots - V_\pi(s_t)$$

- k is small, low variance but high bias;
- k is large, low bias but high variance.

Bias/Variance trade-off:

$$\begin{aligned}\mathcal{A}_t^{GAE(\gamma,\lambda)} &= (1-\lambda)\left(\mathcal{A}_t^{(1)} + \lambda\mathcal{A}_t^{(2)} + \lambda^2\mathcal{A}_t^{(3)} + \cdots\right)\\ &= (1-\lambda)\left(\delta_t + \lambda(\delta_t + \gamma\delta_{t+1}) + \lambda^2\left(\delta_t + \gamma\delta_{t+1} + \gamma^2\delta_{t+2}\right) + \cdots\right)\\ &\cdots\\ &= \sum_{l=0}^{\infty}(\gamma\lambda)^l\delta_{t+l}\end{aligned}$$

Generalized Advantage Estimation (GAE)

Summary

PPO-v1:

$$J_{PPO}^{\theta'}(\theta) = J^{\theta'}(\theta) - \beta KL(\pi_\theta, \pi_{\theta'})$$ → punish if two distributions are very different

- $If\ KL(\pi_\theta, \pi_{\theta'}) > KL_{\max}$, increase β
- $If\ KL(\pi_\theta, \pi_{\theta'}) < KL_{\min}$, decrease β

PPO-v2:

$$J_{PPO2}^{\theta'}(\theta) \simeq \mathbb{E}\Big[\min\left(\frac{\pi_\theta}{\pi_{\theta'}}\mathcal{A}(s,\ a),\ \underline{\text{clip}}\left(\frac{\pi_\theta}{\pi_{\theta'}},\ 1-\epsilon,\ 1+\epsilon\right)\mathcal{A}(s,\ a)\right)\Big]$$

Quiz

1. High variance in policy gradient can be a problem. What are some techniques used to reduce the variance in policy gradient methods?
   A. Increasing the learning rate
   B. Baseline subtraction and advantage estimation
   C. Using larger neural networks
   D. Using Monte Carlo to estimate value function

2. How does actor-critic architecture combine elements of both value-based and policy-based methods?
   A. It uses only a value function for decision-making
   B. It combine a policy with a separate Q-value network
   C. It relies solely on a fixed policy
   D. It doesn't involve value functions

Quiz

3. What is the primary role of the critic in Actor-Critic methods concerning policy improvement?
   A. The critic is responsible for directly determining the optimal policy.
   B. The critic guides the actor by providing an estimate of the advantage of different actions.
   C. The critic is only used for value function approximation and does not influence policy improvement.
   D. The critic is used exclusively for exploration purposes.
4. What is a key difference between on-policy and off-policy algorithms?
   A. On-policy methods require more training steps than off-policy methods
   B. Off-policy methods are less sample-efficient than on-policy methods
   C. On-policy methods update the policy using the current policy, while off-policy methods use data generated by a different policy
   D. Off-policy methods can adapt better to dynamic environments than on-policy methods

Quiz

5. In the context of actor-critic methods, what is the role of the advantage function?
   A. determines the learning rate for the actor
   B. measures the uncertainty in the policy
   C. represents the difference between the estimated value and the baseline
   D. regulates the exploration rate during learning

6. What is the primary motivation behind the introduction of the Proximal Policy Optimization (PPO) algorithm?
   A. To minimize the bias in policy gradient estimates
   B. To maximize entropy in the policy for better exploration
   C. To ensure stability during policy updates and prevent large policy changes
   D. To reduce the computational complexity of value function estimation

Thank you for listening

Autonomous
Systems Lab