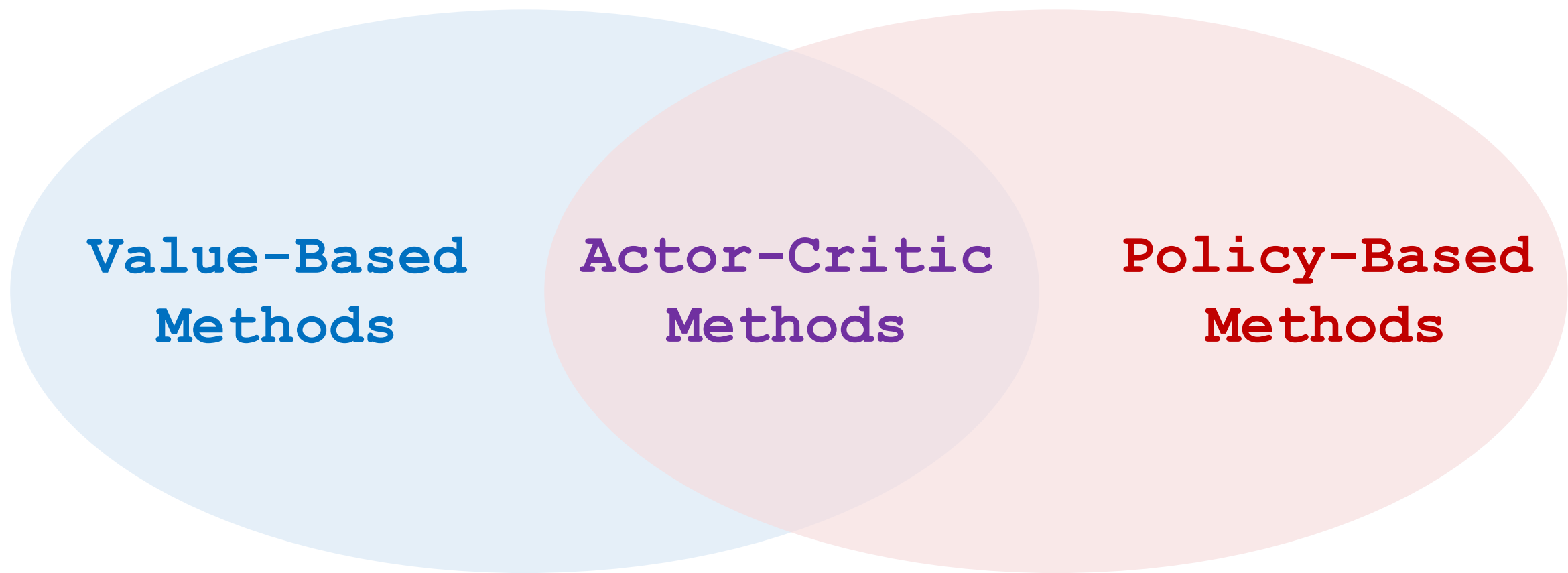


Actor-Critic Methods

Shusen Wang



Actor-Critic Method

State-Value Function Approximation

Definition: State-value function.

- $V_{\pi}(s) = \sum_a \pi(a|s) \cdot Q_{\pi}(s, a).$

Policy network (actor): Use a neural net to approximate $\pi(a|s).$

- Use neural net $\pi(a|s; \theta)$ to approximate $\pi(a|s).$
- θ : trainable parameters of the neural net.

Value network (critic): Use a neural net to approximate $Q_{\pi}(s, a).$

- Use neural net $q(s, a; \mathbf{w})$ to approximate $Q_{\pi}(s, a).$
- \mathbf{w} : trainable parameters of the neural net.

State-Value Function Approximation

Definition: State-value function.

- $V_{\pi}(s) = \sum_a \pi(a|s) \cdot Q_{\pi}(s, a) \approx \sum_a \pi(a|s; \theta) \cdot q(s, a; \mathbf{w}).$

Policy network (actor): Use a neural net to approximate $\pi(a|s)$.

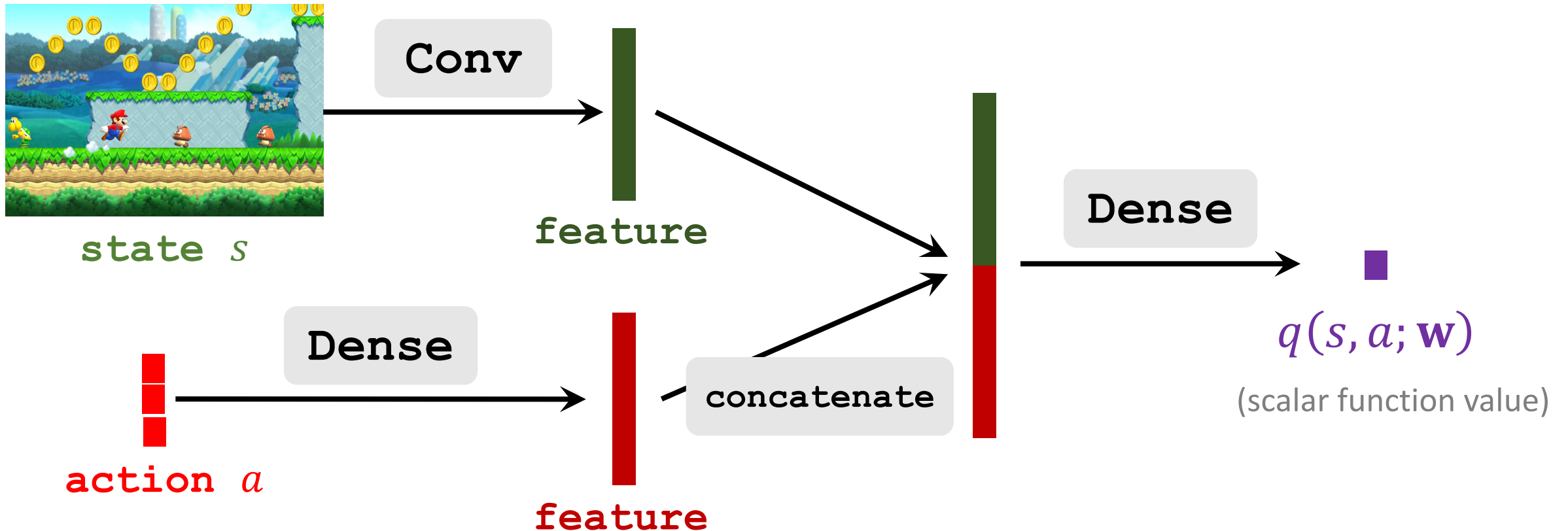
- Use neural net $\pi(a|s; \theta)$ to approximate $\pi(a|s)$.
- θ : trainable parameters of the neural net.

Value network (critic): Use a neural net to approximate $Q_{\pi}(s, a)$.

- Use neural net $q(s, a; \mathbf{w})$ to approximate $Q_{\pi}(s, a)$.
- \mathbf{w} : trainable parameters of the neural net.

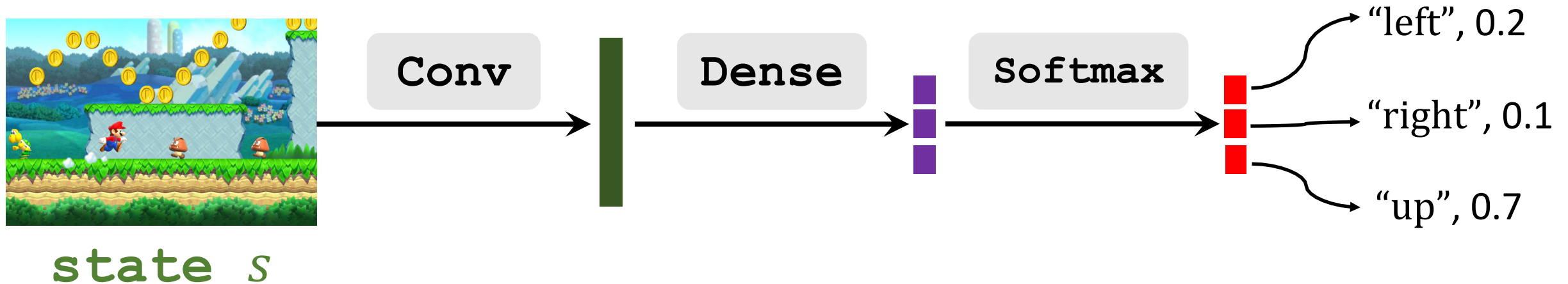
Value Network (Critic): $q(s, a; \mathbf{w})$

- Inputs: **state** s and **action** a .
- Output: approximate action-value (scalar).



Policy Network (Actor): $\pi(a|s, \theta)$

- Input: **state** s , e.g., a screenshot of Super Mario.
- Output: probability distribution over the **actions**.
- Let \mathcal{A} be the set all actions, e.g., $\mathcal{A} = \{\text{"left"}, \text{"right"}, \text{"up"}\}$.
- $\sum_{a \in \mathcal{A}} \pi(a|s, \theta) = 1$. (That is why we use softmax activation.)



State-Value Function Approximation

Definition: State-value function approximated using neural networks.

- $V(\textcolor{green}{s}; \boldsymbol{\theta}, \mathbf{w}) = \sum_{\textcolor{red}{a}} \pi(\textcolor{red}{a} | \textcolor{green}{s}; \boldsymbol{\theta}) \cdot q(\textcolor{green}{s}, \textcolor{red}{a}; \mathbf{w})$.

Training: Update the parameters $\boldsymbol{\theta}$ and \mathbf{w} .

- Update policy network $\pi(\textcolor{red}{a} | \textcolor{green}{s}; \boldsymbol{\theta})$ to increase the state-value $V(\textcolor{green}{s}; \boldsymbol{\theta}, \mathbf{w})$.
 - Actor gradually performs better.
 - (Not actually better; the actor just caters for the critic's taste.)
 - Supervision is purely from the critic.
- Update value network $q(\textcolor{green}{s}, \textcolor{red}{a}; \mathbf{w})$ to better estimate the return.
 - Critic's judgement becomes more accurate.
 - Supervision is purely from the rewards.

State-Value Function Approximation

Definition: State-value function approximated using neural networks.

- $V(\textcolor{green}{s}; \boldsymbol{\theta}, \mathbf{w}) = \sum_{\textcolor{red}{a}} \pi(\textcolor{red}{a} | \textcolor{green}{s}; \boldsymbol{\theta}) \cdot q(\textcolor{green}{s}, \textcolor{red}{a}; \mathbf{w})$.

Training: Update the parameters $\boldsymbol{\theta}$ and \mathbf{w} .

1. Observe the state $\textcolor{green}{s}_t$.
2. Randomly sample action $\textcolor{red}{a}_t$ according to $\pi(\cdot | \textcolor{green}{s}_t; \boldsymbol{\theta}_t)$.
3. Observe new state $\textcolor{green}{s}_{t+1}$ and reward $\textcolor{blue}{r}_t$.
4. Update $\boldsymbol{\theta}$ (in policy network) using policy gradient.
5. Update \mathbf{w} (in value network) using temporal difference (TD).

Update value network using TD

- Predicted value: $q(s_t, a_t; \mathbf{w}_t)$.
- Observe new state s_{t+1} and reward r_t .
- TD target: $y_t = r_t + \gamma \cdot q(s_{t+1}, a_{t+1}; \mathbf{w}_t)$.
- Loss: $L_t(\mathbf{w}) = \frac{1}{2} [q(s_t, a_t; \mathbf{w}) - y_t]^2$.
- Gradient: $\mathbf{g}_t(\mathbf{w}) = \frac{\partial L_t(\mathbf{w})}{\partial \mathbf{w}} = [q(s_t, a_t; \mathbf{w}) - y_t] \cdot \frac{\partial q(s_t, a_t; \mathbf{w})}{\partial \mathbf{w}}$.
- Gradient descent: $\mathbf{w}_{t+1} = \mathbf{w}_t - \alpha \cdot \mathbf{g}_t(\mathbf{w}_t)$.

Update policy network using policy gradient

Definition: State-value function approximated using neural networks.

- $V(\textcolor{green}{s}; \boldsymbol{\theta}, \mathbf{w}) = \sum_{\textcolor{red}{a}} \pi(\textcolor{red}{a} | \textcolor{green}{s}; \boldsymbol{\theta}) \cdot q(\textcolor{green}{s}, \textcolor{red}{a}; \mathbf{w})$.

Policy gradient: Derivative of $V(\textcolor{green}{s}_t; \boldsymbol{\theta}, \mathbf{w})$ w.r.t. $\boldsymbol{\theta}$.

- Let $\tilde{\mathbf{g}}_{\boldsymbol{\theta}}(\boldsymbol{\theta}; \textcolor{red}{a}, \textcolor{green}{s}, \mathbf{w}) = \frac{\partial \log \pi(\textcolor{red}{a} | \textcolor{green}{s}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \cdot q(\textcolor{green}{s}_t, \textcolor{red}{a}; \mathbf{w})$.
- $\frac{\partial V(\textcolor{green}{s}; \boldsymbol{\theta}, \mathbf{w}_t)}{\partial \boldsymbol{\theta}} = \mathbb{E}_{\textcolor{red}{a} \sim \pi(\cdot | \textcolor{green}{s}; \boldsymbol{\theta})} [\tilde{\mathbf{g}}_{\boldsymbol{\theta}}(\boldsymbol{\theta}; \textcolor{red}{a}, \textcolor{green}{s}, \mathbf{w})]$.

Update policy network using policy gradient

Definition: State-value function approximated using neural networks.

- $V(\mathbf{s}; \boldsymbol{\theta}, \mathbf{w}) = \sum_{\mathbf{a}} \pi(\mathbf{a}|\mathbf{s}; \boldsymbol{\theta}) \cdot q(\mathbf{s}, \mathbf{a}; \mathbf{w}).$

Policy gradient: Derivative of $V(\mathbf{s}_t; \boldsymbol{\theta}, \mathbf{w})$ w.r.t. $\boldsymbol{\theta}$.

- Let $\tilde{\mathbf{g}}_{\boldsymbol{\theta}}(\boldsymbol{\theta}; \mathbf{a}, \mathbf{s}, \mathbf{w}) = \frac{\partial \log \pi(\mathbf{a}|\mathbf{s}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \cdot q(\mathbf{s}_t, \mathbf{a}; \mathbf{w}).$
- $\frac{\partial V(\mathbf{s}; \boldsymbol{\theta}, \mathbf{w}_t)}{\partial \boldsymbol{\theta}} = \mathbb{E}_{\mathbf{a} \sim \pi(\cdot|\mathbf{s}; \boldsymbol{\theta})} [\tilde{\mathbf{g}}_{\boldsymbol{\theta}}(\boldsymbol{\theta}; \mathbf{a}, \mathbf{s}, \mathbf{w})].$

Algorithm: Update policy network using stochastic policy gradient.

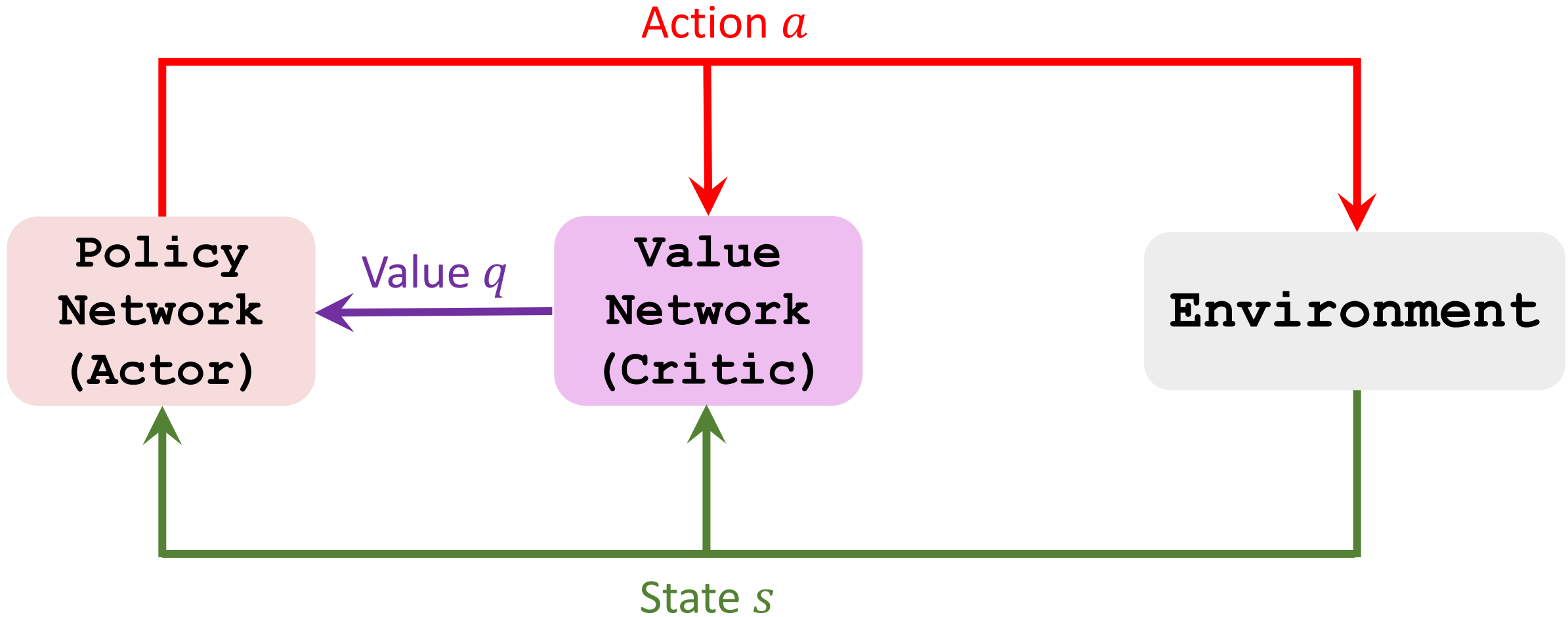
- Random sampling: $\mathbf{a} \sim \pi(\cdot | \mathbf{s}_t; \boldsymbol{\theta}_t)$. (Thus $\tilde{\mathbf{g}}_{\boldsymbol{\theta}}(\boldsymbol{\theta}; \mathbf{a}, \mathbf{s}_t, \mathbf{w}_t)$ is unbiased.)
- Stochastic gradient ascent: $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \beta \cdot \tilde{\mathbf{g}}_{\boldsymbol{\theta}}(\boldsymbol{\theta}_t; \mathbf{a}, \mathbf{s}_t, \mathbf{w}_t).$

Actor Critic Method

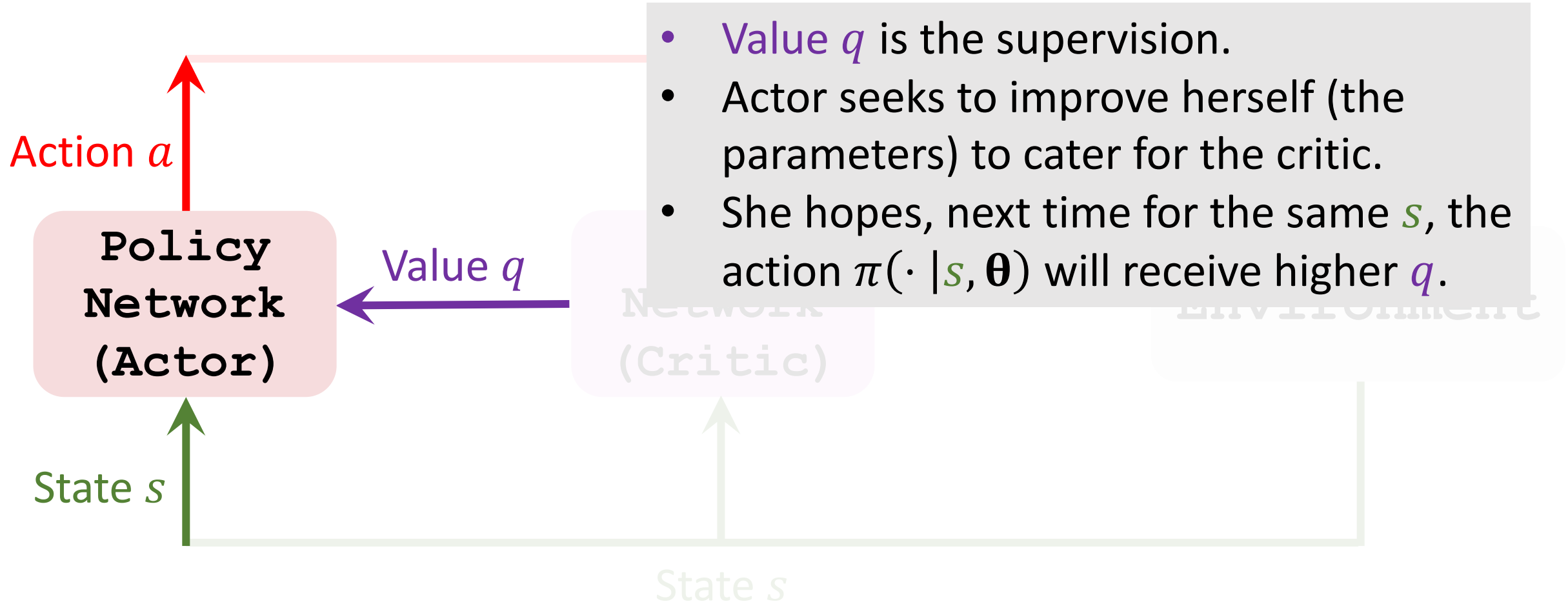
Action a



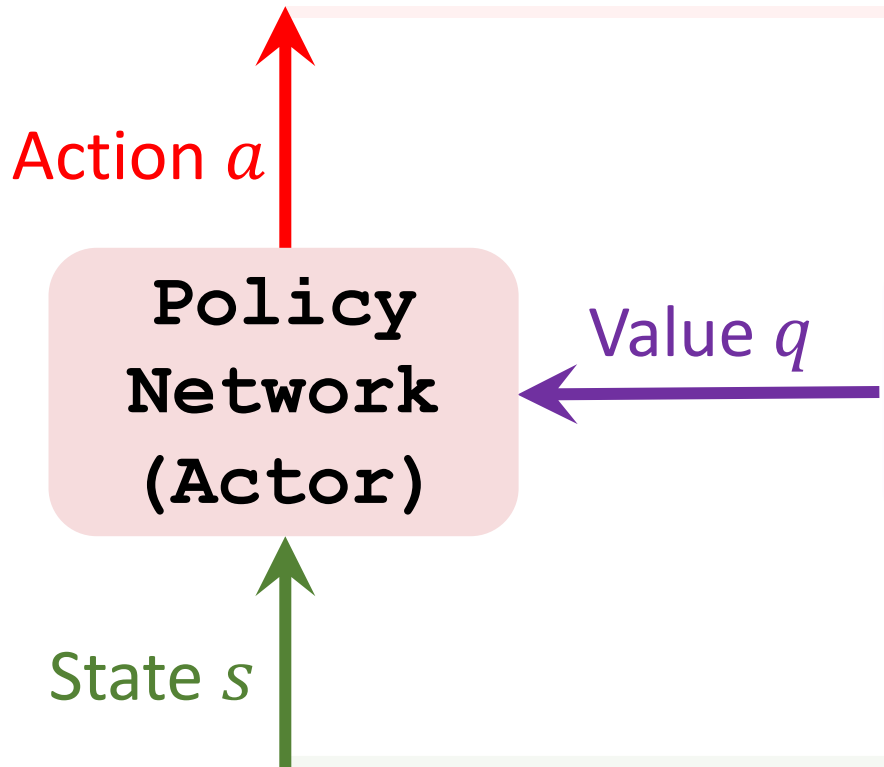
Actor Critic Method



Actor Critic Method: **Update Actor**



Actor Critic Method: **Update Actor**

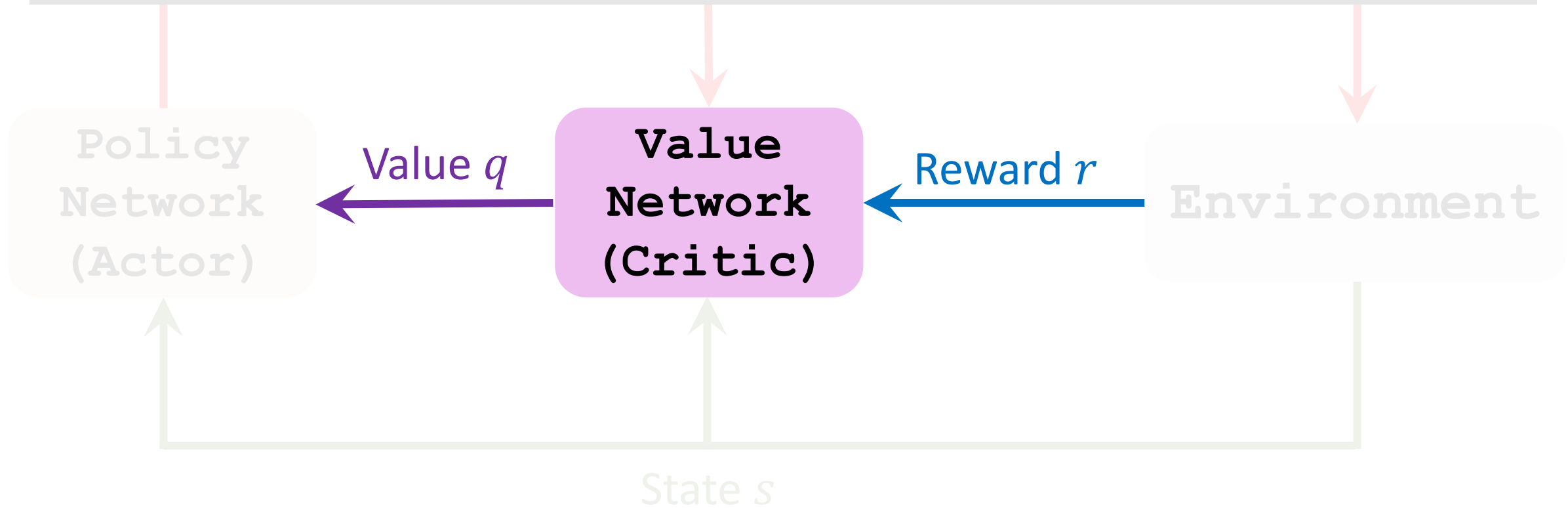


- Value q is the supervision.
- Actor seeks to improve herself (the parameters) to cater for the critic.
- She hopes, next time for the same s , the action $\pi(\cdot | s, \theta)$ will receive higher q .

- Compute policy gradient using s, a, q .
- Update the actor's parameters using "stochastic gradient ascent".

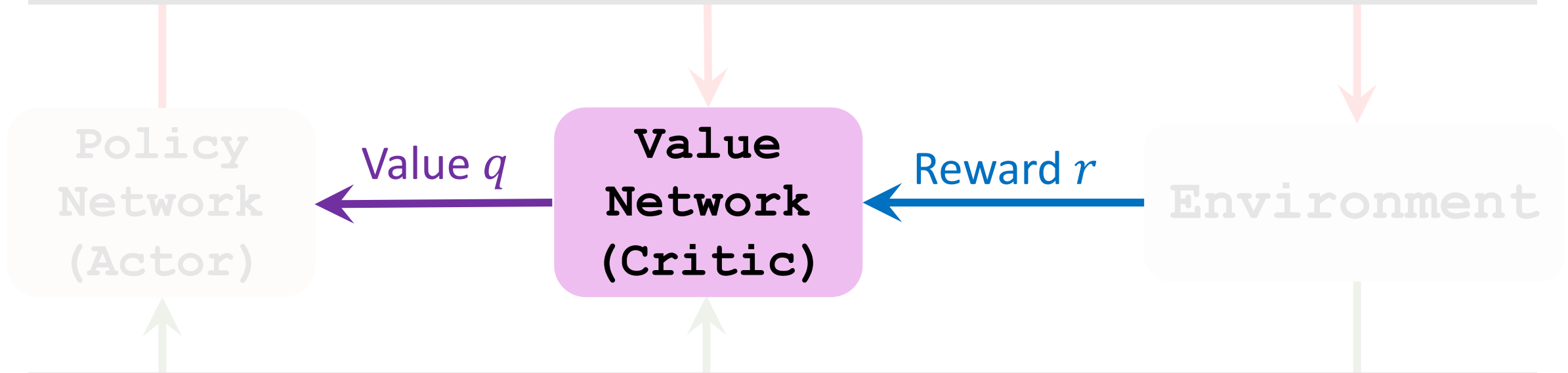
Actor Critic Method: **Update Critic**

- In the beginning, the critic makes random judgement.
- How to improve the critic?



Actor Critic Method: Update Critic

- In the beginning, the critic makes random judgement.
- How to improve the critic?



- Make use the fact that q_t should be close to $r_t + \gamma \cdot q_{t+1}$; if not, a loss.
- Compute q_t using (s_t, a_t) and q_{t+1} using (s_{t+1}, a_{t+1}) .
- Update the critic's parameters using TD learning.

Summary of Algorithm

1. Observe the state s_t ; randomly sample action a_t according to $\pi(\cdot | s_t; \theta_t)$.
2. Perform a_t ; observe new state s_{t+1} and reward r_t .
3. Randomly sample a_{t+1} according to $\pi(\cdot | s_{t+1}; \theta_t)$. (Do not perform a_{t+1} .)
4. Evaluate value network: $q_t = q(s_t, a_t; \mathbf{w}_t)$ and $q_{t+1} = q(s_{t+1}, a_{t+1}; \mathbf{w}_t)$.
5. Compute the TD error: $\delta_t = q_t - (r_t + \gamma \cdot q_{t+1})$.
6. Differentiate value network: $\mathbf{d}_{w,t} = \frac{\partial q(s_t, a_t; \mathbf{w})}{\partial \mathbf{w}} \Big|_{\mathbf{w}=\mathbf{w}_t}$.
7. Update value network: $\mathbf{w}_{t+1} = \mathbf{w}_t - \alpha \cdot \delta_t \cdot \mathbf{d}_{w,t}$.
8. Differentiate policy network: $\mathbf{d}_{\theta,t} = \frac{\partial \log \pi(a_t | s_t, \theta)}{\partial \theta} \Big|_{\theta=\theta_t}$.
9. Update policy network: $\theta_{t+1} = \theta_t + \beta \cdot q_t \cdot \mathbf{d}_{\theta,t}$.

Policy Gradient with Baseline

Policy Gradient with Baseline

Definition: Approximated state-value function.

- $V(s; \theta) - b = \sum_a \pi(a|s; \theta) \cdot [Q_\pi(s, a) - b]$.
- Here, the baseline b must be independent of θ and a .

Policy Gradient with Baseline

Definition: Approximated **state-value function**.

- $V(\mathbf{s}; \boldsymbol{\theta}) - b = \sum_a \pi(a|\mathbf{s}; \boldsymbol{\theta}) \cdot [Q_\pi(\mathbf{s}, a) - b]$.
- Here, the baseline b must be independent of $\boldsymbol{\theta}$ and a .

Policy gradient: Derivative of $V(\mathbf{s}; \boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$.

- $\frac{\partial V(\mathbf{s}; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \frac{\partial [V(\mathbf{s}; \boldsymbol{\theta}) - b]}{\partial \boldsymbol{\theta}} = \mathbb{E}_{a \sim \pi(\cdot|\mathbf{s}; \boldsymbol{\theta})} \left[\frac{\partial \log \pi(a|\mathbf{s}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \cdot [Q_\pi(\mathbf{s}, a) - b] \right]$.
- The baseline b does not affect correctness.
- A good baseline b can reduce variance.
- We can use $b = r_t + \gamma \cdot q_{t+1}$ (TD target) as the baseline.

Actor Critic Update (**without baseline**)

1. Observe the state s_t ; randomly sample action a_t according to $\pi(\cdot | s_t; \boldsymbol{\theta}_t)$.
2. Perform a_t ; observe new state s_{t+1} and reward r_t .
3. Randomly sample a_{t+1} according to $\pi(\cdot | s_{t+1}; \boldsymbol{\theta}_t)$. (Do not perform a_{t+1} .)
4. Evaluate value network: $q_t = q(s_t, a_t; \mathbf{w}_t)$ and $q_{t+1} = q(s_{t+1}, a_{t+1}; \mathbf{w}_t)$.
5. Compute the TD error: $\delta_t = q_t - (r_t + \gamma \cdot q_{t+1})$.
6. Differentiate value network: $\mathbf{d}_{w,t} = \frac{\partial q(s_t, a_t; \mathbf{w})}{\partial \mathbf{w}} \Big|_{\mathbf{w}=\mathbf{w}_t}$.
7. Update value network: $\mathbf{w}_{t+1} = \mathbf{w}_t - \alpha \cdot \delta_t \cdot \mathbf{d}_{w,t}$.
8. Differentiate policy network: $\mathbf{d}_{\theta,t} = \frac{\partial \log \pi(a_t | s_t, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_t}$.
9. Update policy network: $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \beta \cdot q_t \cdot \mathbf{d}_{\theta,t}$.

Actor Critic Update (**with baseline**)

1. Observe the state s_t ; randomly sample action a_t according to $\pi(\cdot | s_t; \theta_t)$.
2. Perform a_t ; observe new state s_{t+1} and reward r_t .
3. Randomly sample a_{t+1} according to $\pi(\cdot | s_{t+1}; \theta_t)$. (Do not perform a_{t+1} .)
4. Evaluate value network: $q_t = q(s_t, a_t; \mathbf{w}_t)$ and $q_{t+1} = q(s_{t+1}, a_{t+1}; \mathbf{w}_t)$.
5. Compute the TD error: $\delta_t = q_t - (r_t + \gamma \cdot q_{t+1})$.
6. Differentiate value network: $\mathbf{d}_{w,t} = \frac{\partial q(s_t, a_t; \mathbf{w})}{\partial \mathbf{w}} \Big|_{\mathbf{w}=\mathbf{w}_t}$.
Baseline
7. Update value network: $\mathbf{w}_{t+1} = \mathbf{w}_t - \alpha \cdot \delta_t \cdot \mathbf{d}_{w,t}$.
8. Differentiate policy network: $\mathbf{d}_{\theta,t} = \frac{\partial \log \pi(a_t | s_t, \theta)}{\partial \theta} \Big|_{\theta=\theta_t}$.
9. Update policy network: $\theta_{t+1} = \theta_t + \beta \cdot \delta_t \cdot \mathbf{d}_{\theta,t}$.

Summary

Actor-Critic Method

Definition: State-value function.

- $V_{\pi}(s) = \sum_a \pi(a|s) \cdot Q_{\pi}(s, a).$

Definition: State-value function approximation using neural networks.

- Approximate policy function $\pi(a|s)$ by $\pi(a|s; \theta)$ (actor).
- Approximate value function $Q_{\pi}(s, a)$ by $q(s, a; \mathbf{w})$ (critic).

Actor-Critic Method

- Value network (critic) is useful in training; it provides the policy network (actor) with supervision.
- After training, the value network (critic) will not be used.
- Agent is controlled by policy network (actor):

$$a_t \sim \pi(\cdot | s_t; \theta).$$

Training Policy and Value Networks

Learning: Update the **policy network (actor)** by **policy gradient**.

- Seek to increase state-value: $V(\mathbf{s}; \boldsymbol{\theta}, \mathbf{w}) = \sum_{\mathbf{a}} \pi(\mathbf{a}|\mathbf{s}; \boldsymbol{\theta}) \cdot q(\mathbf{s}, \mathbf{a}; \mathbf{w})$.
- Compute policy gradient: $\frac{\partial V(\mathbf{s}; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \mathbb{E}_{\mathbf{a} \sim \pi(\cdot|\mathbf{s}; \boldsymbol{\theta})} \left[\frac{\partial \log \pi(\mathbf{a}|\mathbf{s}; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \cdot q(\mathbf{s}, \mathbf{a}; \mathbf{w}) \right]$.
- Perform gradient ascent.

Learning: Update the **value network (critic)** by **TD learning**.

- Predicted action-value: $q_t = q(\mathbf{s}_t, \mathbf{a}_t; \mathbf{w})$.
- TD target: $y_t = r_t + \gamma \cdot q(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}; \mathbf{w})$
- Gradient: $\frac{\partial (q_t - y_t)^2 / 2}{\partial \mathbf{w}} = (q_t - y_t) \cdot \frac{\partial q(\mathbf{s}_t, \mathbf{a}_t; \mathbf{w})}{\partial \mathbf{w}}$.
- Perform gradient descent.

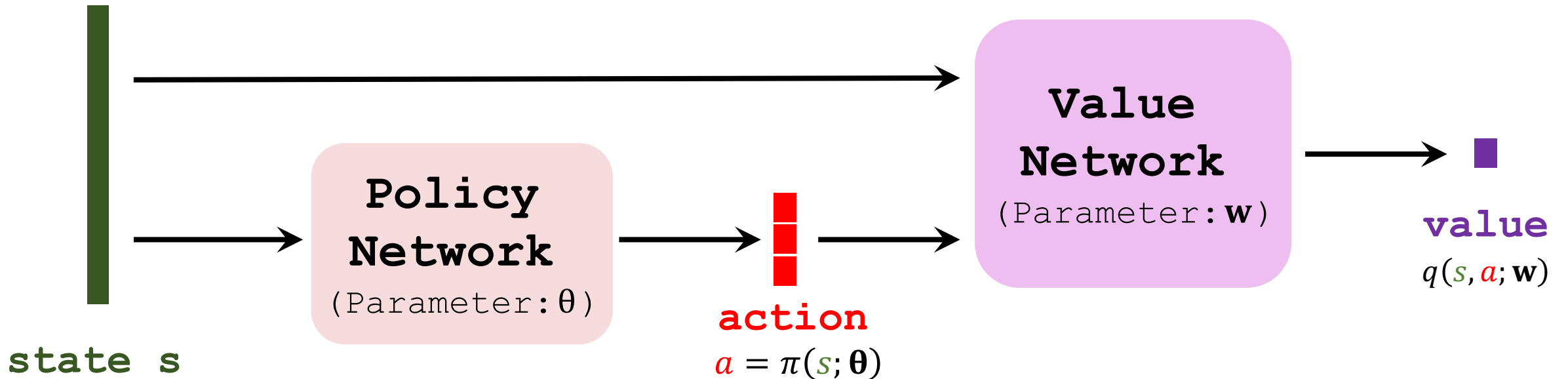
Deterministic Policy Gradient (DPG)

Reference

- Lillicrap and others: [Continuous control with deep reinforcement learning](#). arXiv:1509.02971. 2015.

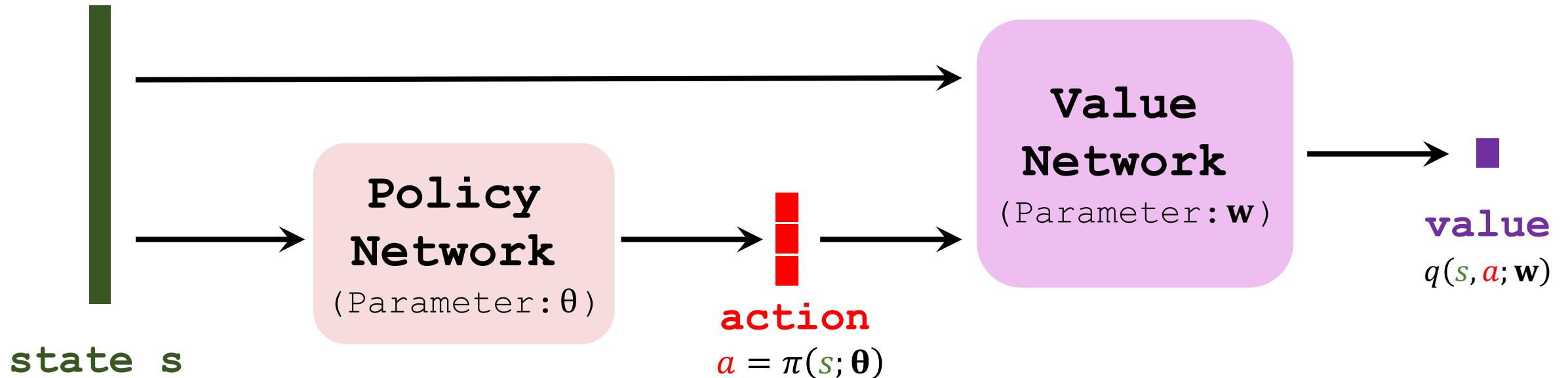
Deterministic Policy Gradient (DPG)

- DPG is an actor-critic method.
- The policy network is deterministic: $a = \pi(s; \theta)$.



Deterministic Policy Gradient (DPG)

- DPG is a actor-critic method.
- The policy network is deterministic: $a = \pi(s; \theta)$.
- Trained value network by TD learning.
- Train policy network to maximize the value $q(s, a; w)$.



Train Policy Network

- Train policy network to maximize the value $q(s, a; \mathbf{w})$.
- Gradient: $\frac{\partial q(s, a; \mathbf{w})}{\partial \theta} = \frac{\partial a}{\partial \theta} \cdot \frac{\partial q(s, a; \mathbf{w})}{\partial a}$.
- Update θ using gradient ascent.

