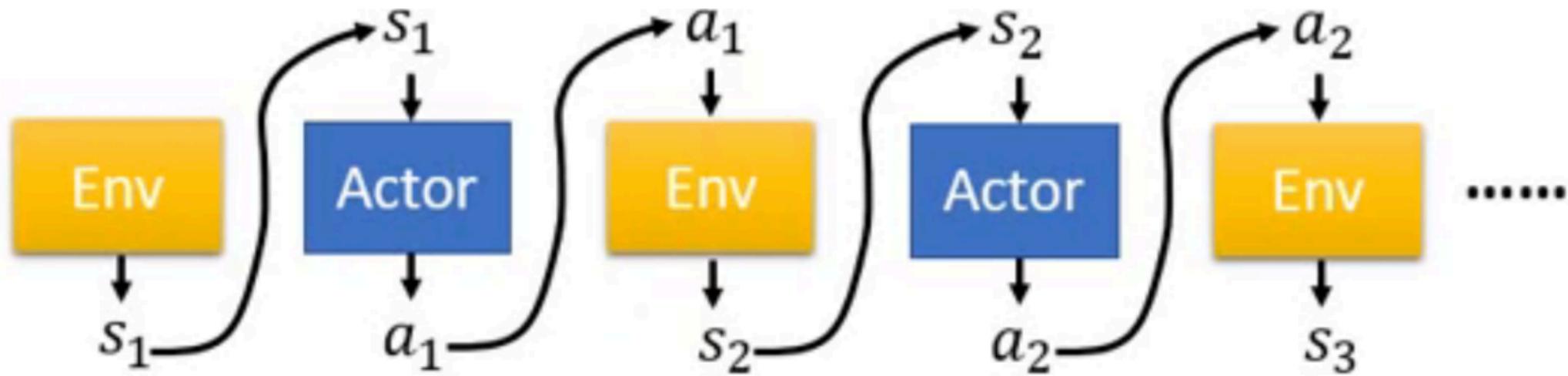
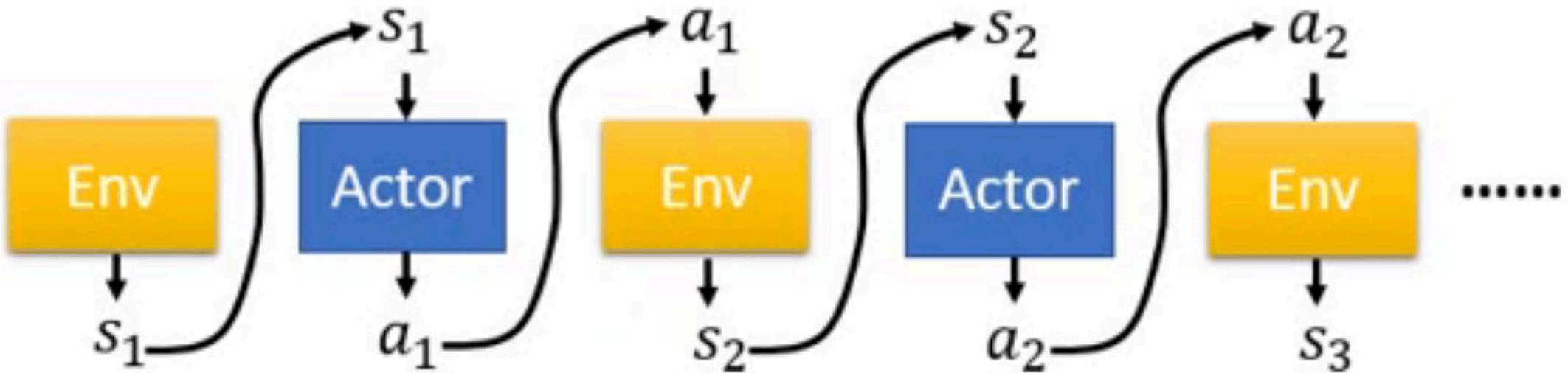


Model-free Policy Based Reinforcement Learning



Trajectory $\tau = \{s_1, a_1, s_2, a_2, \dots, s_T, a_T\}$

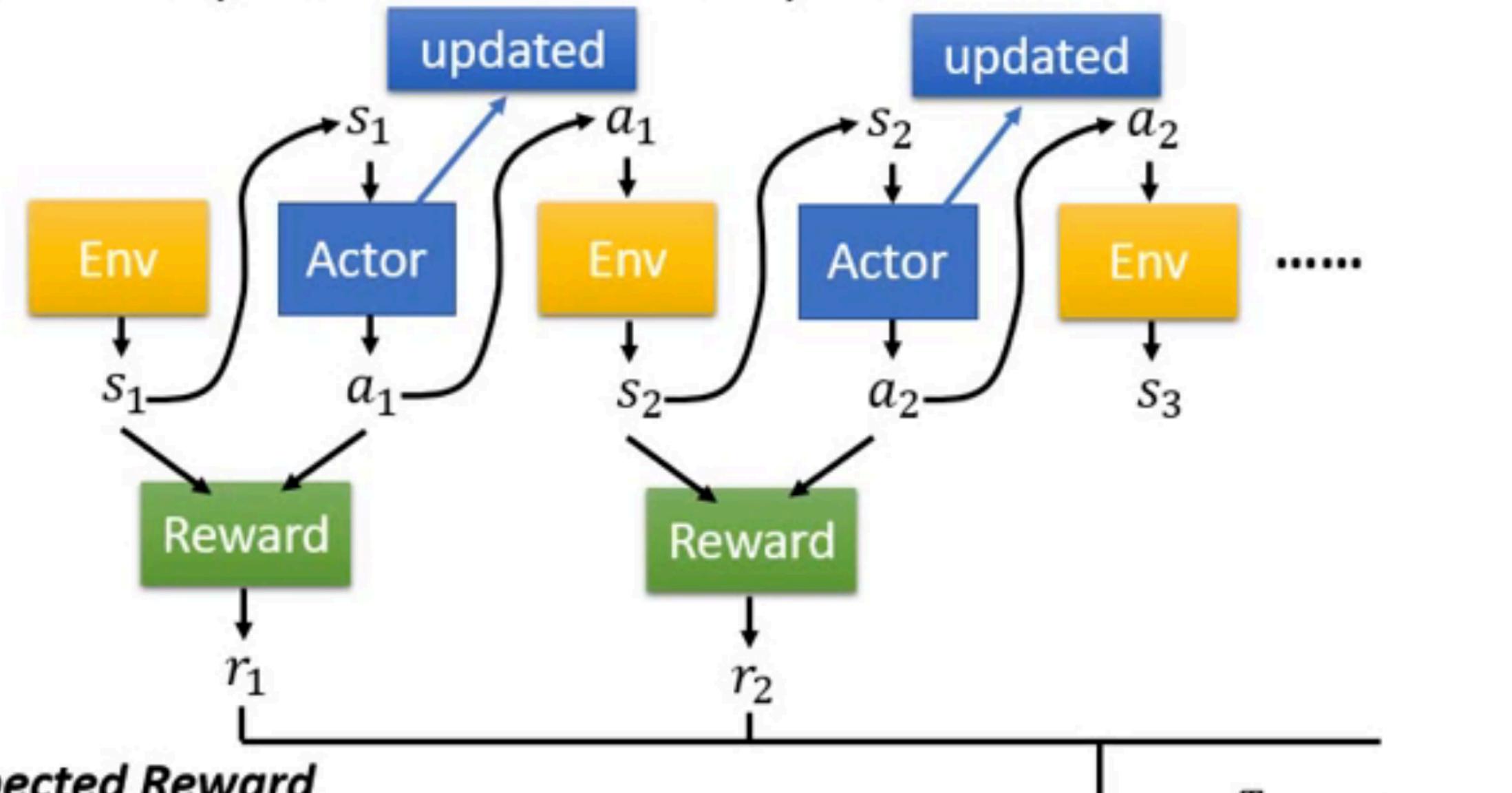


Trajectory $\tau = \{s_1, a_1, s_2, a_2, \dots, s_T, a_T\}$

$$p_\theta(\tau)$$

$$= p(s_1)p_\theta(a_1|s_1)p(s_2|s_1, a_1)p_\theta(a_2|s_2)p(s_3|s_2, a_2)\dots$$

$$= p(s_1) \prod_{t=1}^T p_\theta(a_t|s_t)p(s_{t+1}|s_t, a_t)$$



Expected Reward

$$\bar{R}_\theta = \sum_{\tau} R(\tau)p_\theta(\tau) = E_{\tau \sim p_\theta(\tau)}[R(\tau)]$$

$$R(\tau) = \sum_t^T r_t$$

REINFORCE method

Policy gradient methods for reinforcement learning with function approximation

$$\nabla \bar{R}_\theta = \sum_{\tau} R(\tau) \nabla p_\theta(\tau) = \sum_{\tau} R(\tau) p_\theta(\tau) \frac{\nabla p_\theta(\tau)}{p_\theta(\tau)}$$

$$= E_{\tau \sim p_\theta(\tau)} [R(\tau) \nabla \log p_\theta(\tau)] \approx \frac{1}{N} \sum_{n=1}^N R(\tau^n) \nabla \log p_\theta(\tau^n)$$

$$= \frac{1}{N} \sum_{n=1}^N \sum_{t=1}^{T_n} R(\tau^n) \nabla \log p_\theta(a_t^n | s_t^n)$$

Update
Model

Given policy π_θ

$$\tau^1: (s_1^1, a_1^1) \quad R(\tau^1)$$

$$(s_2^1, a_2^1) \quad R(\tau^1)$$

\vdots

\vdots

$$\tau^2: (s_1^2, a_1^2) \quad R(\tau^2)$$

$$(s_2^2, a_2^2) \quad R(\tau^2)$$

\vdots

\vdots

$$\theta \leftarrow \theta + \eta \nabla \bar{R}_\theta$$

$$\nabla \bar{R}_\theta =$$

$$\frac{1}{N} \sum_{n=1}^N \sum_{t=1}^{T_n} R(\tau^n) \nabla \log p_\theta(a_t^n | s_t^n)$$

Data
Collection

only used once

On-policy -> Off-policy

- Important sampling:

$$E_{x \sim p}[f(x)] = E_{x \sim q}[f(x) \frac{p(x)}{q(x)}]$$

$$\nabla \bar{R}_\theta = E_{\tau \sim p_\theta(\tau)}[R(\tau) \nabla \log p_\theta(\tau)]$$



$$\nabla \bar{R}_\theta = E_{\tau \sim \underline{p_{\theta'}(\tau)}} \left[\frac{p_\theta(\tau)}{\underline{p_{\theta'}(\tau)}} R(\tau) \nabla \log p_\theta(\tau) \right]$$

定义 G_t 为长期回报期望 (Return) :

$$G_t = \sum_{n=0}^N \gamma^n r_{t+n}$$

状态s的value为:

$$V_\pi(s) = E_\pi[G_t | S_t = s]$$

状态s下采取动作a的Q值为:

$$Q_\pi(s, a) = E_\pi[G_t | S_t = s, A_t = a]$$

其中 $0 < \gamma < 1$ 是长期收益的折扣因子，类似于金融中的折现率。

PPO:

Proximal policy optimization algorithms

$$\begin{aligned} & \underset{\theta}{\text{maximize}} \quad \hat{\mathbb{E}}_t \left[\frac{\pi_{\theta}(a_t | s_t)}{\pi_{\theta_{\text{old}}}(a_t | s_t)} \hat{A}_t \right] \\ & \text{subject to} \quad \hat{\mathbb{E}}_t [\text{KL}[\pi_{\theta_{\text{old}}}(\cdot | s_t), \pi_{\theta}(\cdot | s_t)]] \leq \delta. \end{aligned}$$

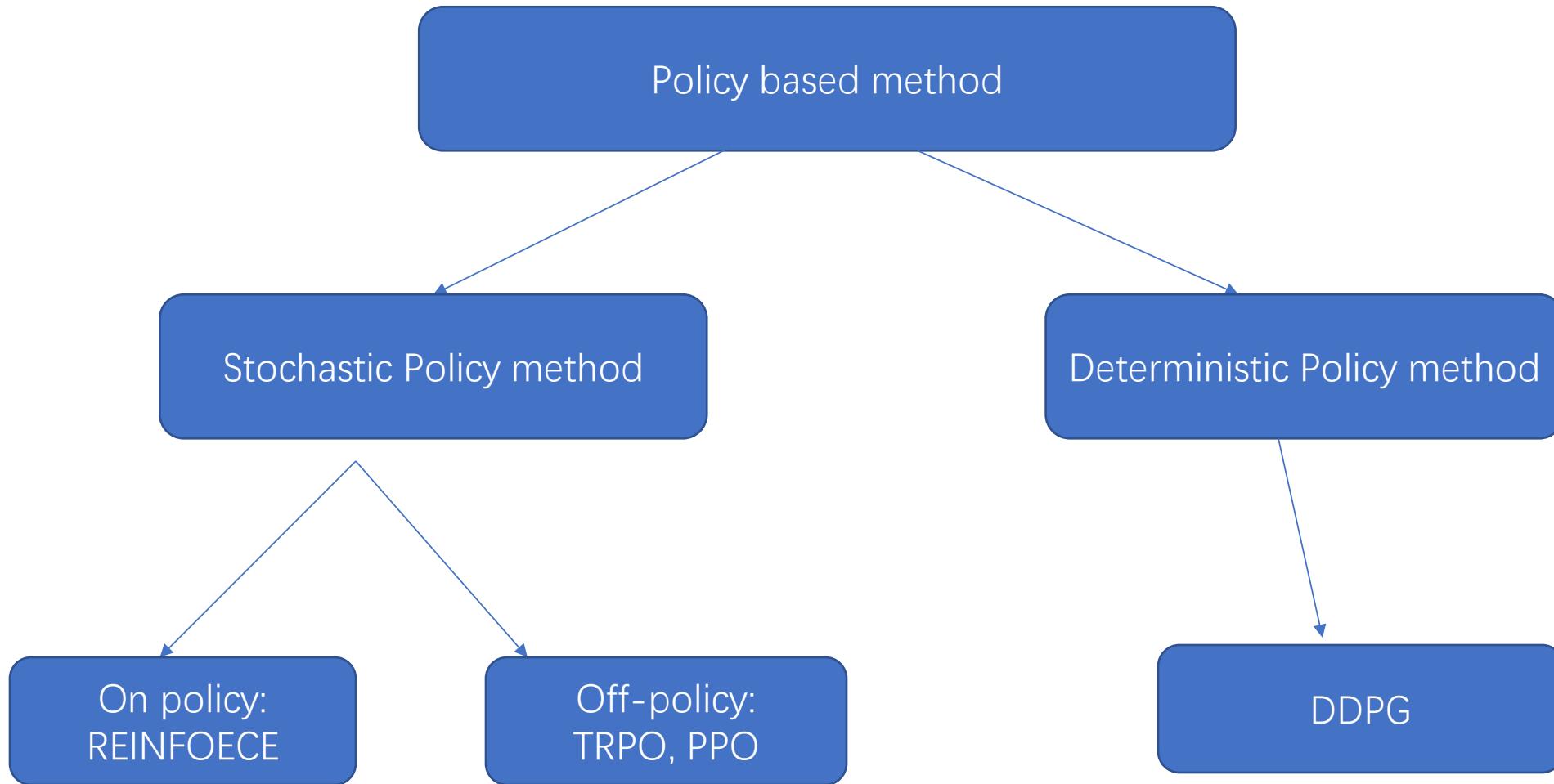
$$\underset{\theta}{\text{maximize}} \hat{\mathbb{E}}_t \left[\frac{\pi_{\theta}(a_t | s_t)}{\pi_{\theta_{\text{old}}}(a_t | s_t)} \hat{A}_t - \beta \text{KL}[\pi_{\theta_{\text{old}}}(\cdot | s_t), \pi_{\theta}(\cdot | s_t)] \right]$$

- Initial policy parameters θ^0
- In each iteration
 - Using θ^k to interact with the environment to collect $\{s_t, a_t\}$ and compute advantage $A^{\theta^k}(s_t, a_t)$
 - Find θ optimizing $J_{PPO}(\theta)$

$$J_{PPO}^{\theta^k}(\theta) = J^{\theta^k}(\theta) - \beta KL(\theta, \theta^k)$$

Update parameters
several times

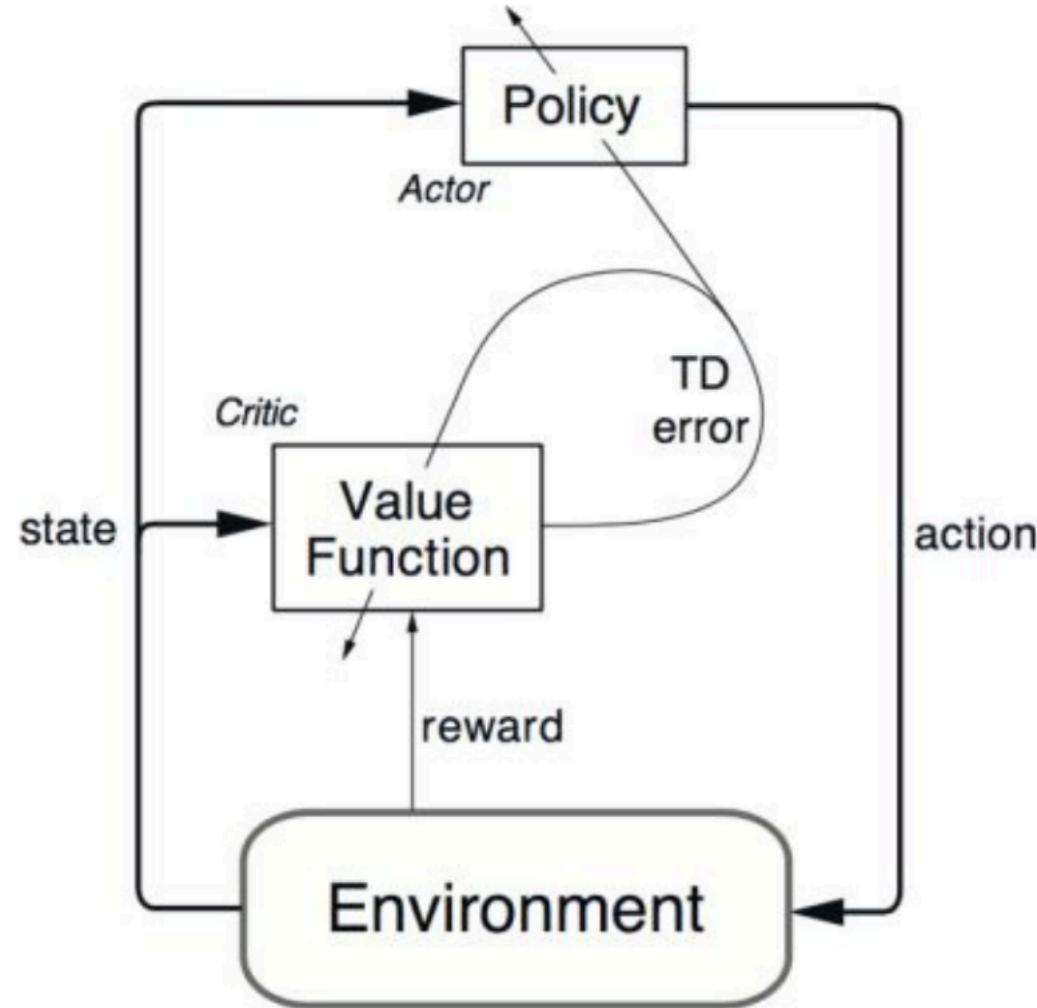
- If $KL(\theta, \theta^k) > KL_{max}$, increase β
- If $KL(\theta, \theta^k) < KL_{min}$, decrease β



DDPG:

Continuous control with deep reinforcement learning

Actor-Critic Framework



Algorithm 1 DDPG algorithm

Randomly initialize critic network $Q(s, a | \theta^Q)$ and actor $\mu(s | \theta^\mu)$ with weights θ^Q and θ^μ .
Initialize target network Q' and μ' with weights $\theta^{Q'} \leftarrow \theta^Q, \theta^{\mu'} \leftarrow \theta^\mu$
Initialize replay buffer R

for episode = 1, M **do**

- Initialize a random process \mathcal{N} for action exploration
- Receive initial observation state s_1
- for** t = 1, T **do**

 - Select action $a_t = \mu(s_t | \theta^\mu) + \mathcal{N}_t$ according to the current policy and exploration noise
 - Execute action a_t and observe reward r_t and observe new state s_{t+1}
 - Store transition (s_t, a_t, r_t, s_{t+1}) in R
 - Sample a random minibatch of N transitions (s_i, a_i, r_i, s_{i+1}) from R
 - Set $y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1} | \theta^{\mu'}) | \theta^{Q'})$
 - Update critic by minimizing the loss: $L = \frac{1}{N} \sum_i (y_i - Q(s_i, a_i | \theta^Q))^2$ update Critic Network
 - Update the actor policy using the sampled gradient:
Critic target Network $\nabla_{\theta^\mu} \mu|_{s_i} \approx \frac{1}{N} \sum_i \nabla_a Q(s, a | \theta^Q)|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu} \mu(s | \theta^\mu)|_{s_i}$ update Actor Network
 - Actor target Network

 Update the target networks:

 - $\theta^{Q'} \leftarrow \tau \theta^Q + (1 - \tau) \theta^{Q'}$
 - $\theta^{\mu'} \leftarrow \tau \theta^\mu + (1 - \tau) \theta^{\mu'}$

end for

end for

- REINFORCE:

Sutton, Richard S., David A. McAllester, Satinder P. Singh, and Yishay Mansour. "Policy gradient methods for reinforcement learning with function approximation." In *Advances in neural information processing systems*, pp. 1057-1063. 2000.

- PPO:

Schulman, John, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. "Proximal policy optimization algorithms." arXiv preprint arXiv:1707.06347 (2017).

- DDPG:

Lillicrap, Timothy P., Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. "Continuous control with deep reinforcement learning." arXiv preprint arXiv:1509.02971 (2015).

- David Silver <https://zhuanlan.zhihu.com/reinforce>
- CS294 <http://rail.eecs.berkeley.edu/deeprlcourse-fa17/index.html>