TRPO & PPO

TRPO -> PPO

- Trust Region Policy Optimization: https://arxiv.org/pdf/1502.05477.pdf
- Proximal Policy Optimization: https://arxiv.org/pdf/1707.06347.pdf

TRPO:

$$\eta(\pi) = \mathbb{E}_{s_0, a_0, \ldots}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t)\right], \text{ where}$$

$$s_0 \sim \rho_0(s_0),\ a_t \sim \pi(a_t|s_t),\ s_{t+1} \sim P(s_{t+1}|s_t, a_t).$$

$$\textit{Policy Gradient} : \nabla_\theta \eta(\pi)$$

TRPO:

$$\eta(\pi) = \mathbb{E}_{s_0, a_0, \ldots} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t) \right], \text{ where}$$

$$s_0 \sim \rho_0(s_0),\ a_t \sim \pi(a_t|s_t),\ s_{t+1} \sim P(s_{t+1}|s_t, a_t).$$

$$\textit{Policy Gradient} : \nabla_\theta \eta(\pi)$$

TRPO:

$$\eta(\pi) = \mathbb{E}_{s_0, a_0, \ldots}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t)\right], \text{ where}$$

$$s_0 \sim \rho_0(s_0),\ a_t \sim \pi(a_t|s_t),\ s_{t+1} \sim P(s_{t+1}|s_t, a_t).$$

$$Q_\pi(s_t, a_t) = \mathbb{E}_{s_{t+1}, a_{t+1}, \ldots}\left[\sum_{l=0}^{\infty} \gamma^l r(s_{t+l})\right],$$

$$V_\pi(s_t) = \mathbb{E}_{a_t, s_{t+1}, \ldots}\left[\sum_{l=0}^{\infty} \gamma^l r(s_{t+l})\right],$$

$$A_\pi(s, a) = Q_\pi(s, a) - V_\pi(s), \text{ where}$$

$$a_t \sim \pi(a_t|s_t), s_{t+1} \sim P(s_{t+1}|s_t, a_t) \text{ for } t \geq 0.$$

TRPO:

$$\eta(\pi) = \mathbb{E}_{s_0, a_0, \ldots}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t)\right], \text{ where}$$

$$s_0 \sim \rho_0(s_0),\ a_t \sim \pi(a_t|s_t),\ s_{t+1} \sim P(s_{t+1}|s_t, a_t).$$

The following useful identity expresses the expected return of another policy $\tilde{\pi}$ in terms of the advantage over π, accumulated over timesteps (see Kakade & Langford (2002) or Appendix A for proof):

$$\eta(\tilde{\pi}) = \eta(\pi) + \mathbb{E}_{s_0, a_0, \cdots \sim \tilde{\pi}}\left[\sum_{t=0}^{\infty} \gamma^t A_\pi(s_t, a_t)\right] \qquad (1)$$

TRPO:

$$\eta(\pi) = \mathbb{E}_{s_0, a_0, \ldots}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t)\right], \text{ where}$$

$$s_0 \sim \rho_0(s_0),\; a_t \sim \pi(a_t|s_t),\; s_{t+1} \sim P(s_{t+1}|s_t, a_t).$$

$$\eta(\tilde{\pi}) = \eta(\pi) + \mathbb{E}_{s_0, a_0, \cdots \sim \tilde{\pi}}\left[\sum_{t=0}^{\infty} \gamma^t A_\pi(s_t, a_t)\right] \quad (1)$$

$$= \eta(\pi) + \sum_s \rho_{\tilde{\pi}}(s) \sum_a \tilde{\pi}(a|s) A_\pi(s, a)$$

$$L_\pi(\tilde{\pi}) = \eta(\pi) + \sum_s \rho_\pi(s) \sum_a \tilde{\pi}(a|s) A_\pi(s, a). \quad (3)$$

discounted visitation frequencies

$$\rho_\pi(s) = P(s_0 = s) + \gamma P(s_1 = s) + \gamma^2 P(s_2 = s) + \ldots$$

TRPO:

$$\eta(\pi) = \mathbb{E}_{s_0, a_0, \ldots} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t) \right], \text{ where}$$

$$s_0 \sim \rho_0(s_0),\ a_t \sim \pi(a_t | s_t),\ s_{t+1} \sim P(s_{t+1} | s_t, a_t).$$

$$\eta(\tilde{\pi}) = \eta(\pi) + \mathbb{E}_{s_0, a_0, \cdots \sim \tilde{\pi}} \left[\sum_{t=0}^{\infty} \gamma^t A_\pi(s_t, a_t) \right] \quad (1)$$

$$= \eta(\pi) + \sum_s \rho_{\tilde{\pi}}(s) \sum_a \tilde{\pi}(a|s) A_\pi(s, a)$$

$$L_\pi(\tilde{\pi}) = \eta(\pi) + \sum_s \rho_\pi(s) \sum_a \tilde{\pi}(a|s) A_\pi(s, a). \quad (3)$$

This is an approximation!

TRPO:

$$\eta(\pi) = \mathbb{E}_{s_0, a_0, \ldots} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t) \right], \text{ where}$$

$$s_0 \sim \rho_0(s_0),\ a_t \sim \pi(a_t | s_t),\ s_{t+1} \sim P(s_{t+1} | s_t, a_t).$$

$$\eta(\tilde{\pi}) = \eta(\pi) + \mathbb{E}_{s_0, a_0, \cdots \sim \tilde{\pi}} \left[\sum_{t=0}^{\infty} \gamma^t A_\pi(s_t, a_t) \right] \quad (1)$$

$$= \eta(\pi) + \sum_s \rho_{\tilde{\pi}}(s) \sum_a \tilde{\pi}(a|s) A_\pi(s, a)$$

$$L_\pi(\tilde{\pi}) = \eta(\pi) + \sum_s \rho_\pi(s) \sum_a \tilde{\pi}(a|s) A_\pi(s, a). \quad (3)$$

This is an approximation!
Should be $\rho_{\tilde{\pi}}$ instead of ρ_π

TRPO:

$$\eta(\pi) = \mathbb{E}_{s_0, a_0, \ldots}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t)\right], \text{ where}$$

$$s_0 \sim \rho_0(s_0),\ a_t \sim \pi(a_t|s_t),\ s_{t+1} \sim P(s_{t+1}|s_t, a_t).$$

This is an approximation!

Should be $\rho_{\tilde{\pi}}$ instead of ρ_π

$$L_\pi(\tilde{\pi}) = \eta(\pi) + \sum_s \rho_\pi(s) \sum_a \tilde{\pi}(a|s) A_\pi(s, a). \quad (3)$$

TRPO says: Approximately valid only if $\tilde{\pi} \sim \pi$

TRPO:

TRPO says: Approximately valid only if $\tilde{\pi} \sim \pi$

$$\sum_a \pi_\theta(a|s_n) A_{\theta_{\text{old}}}(s_n, a) = \mathbb{E}_{a\sim q}\left[\frac{\pi_\theta(a|s_n)}{q(a|s_n)} A_{\theta_{\text{old}}}(s_n, a)\right]$$

Our optimization problem in Equation (13) is exactly equivalent to the following one, written in terms of expectations:

$$\underset{\theta}{\text{maximize}}\, \mathbb{E}_{s\sim\rho_{\theta_{\text{old}}}, a\sim q}\left[\frac{\pi_\theta(a|s)}{q(a|s)} Q_{\theta_{\text{old}}}(s, a)\right] \tag{14}$$

$$\text{subject to } \mathbb{E}_{s\sim\rho_{\theta_{\text{old}}}}\left[D_{\text{KL}}(\pi_{\theta_{\text{old}}}(\cdot|s) \,\|\, \pi_\theta(\cdot|s))\right] \leq \delta.$$

TRPO:

Algorithm 1 Policy iteration algorithm guaranteeing non-decreasing expected return η

Initialize π_0.

for $i = 0, 1, 2, \dots$ until convergence **do**

Compute all advantage values $A_{\pi_i}(s, a)$.

Solve the constrained optimization problem

$$\pi_{i+1} = \arg\max_{\pi} \left[L_{\pi_i}(\pi) - C D_{\mathrm{KL}}^{\max}(\pi_i, \pi) \right]$$

where $C = 4\epsilon\gamma/(1-\gamma)^2$

and $L_{\pi_i}(\pi) = \eta(\pi_i) + \sum_s \rho_{\pi_i}(s) \sum_a \pi(a|s) A_{\pi_i}(s, a)$

end for

PPO:

$$L^{CPI}(\theta) = \hat{\mathbb{E}}_t\left[\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}\hat{A}_t\right] = \hat{\mathbb{E}}_t\left[r_t(\theta)\hat{A}_t\right].$$

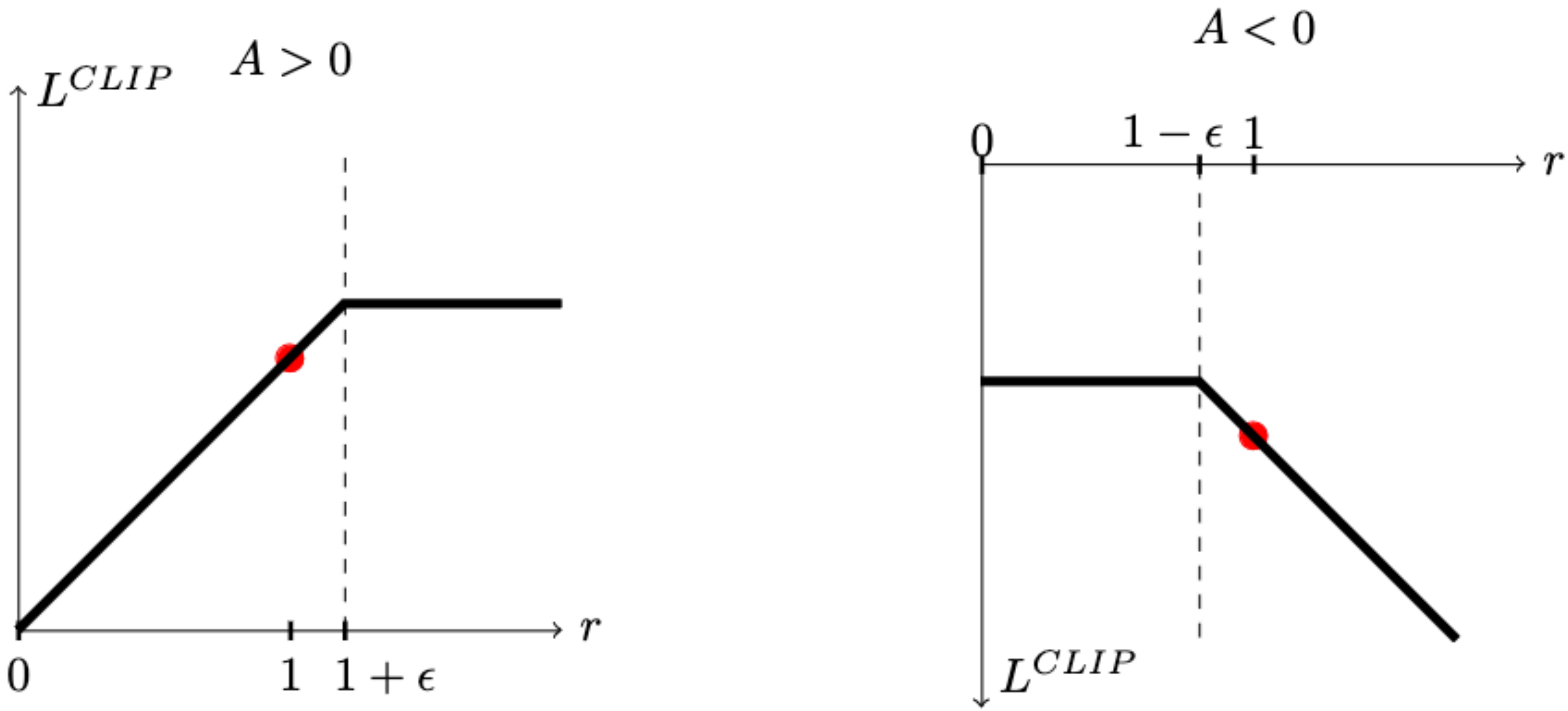
PPO:

$$L^{CPI}(\theta) = \hat{\mathbb{E}}_t\left[\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}\hat{A}_t\right] = \hat{\mathbb{E}}_t\left[r_t(\theta)\hat{A}_t\right].$$

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t\left[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)\right]$$

PPO:

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t\left[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)\right]$$



PPO:

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t\left[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)\right]$$

Algorithm 1 PPO-Clip

1: Input: initial policy parameters θ_0, initial value function parameters ϕ_0
2: **for** $k = 0, 1, 2, ...$ **do**
3: Collect set of trajectories $\mathcal{D}_k = \{\tau_i\}$ by running policy $\pi_k = \pi(\theta_k)$ in the environment.
4: Compute rewards-to-go $\hat{R}_t$.
5: Compute advantage estimates, $\hat{A}_t$ (using any method of advantage estimation) based on the current value function V_{ϕ_k}.
6: Update the policy by maximizing the PPO-Clip objective:

$$\theta_{k+1} = \arg\max_{\theta} \frac{1}{|\mathcal{D}_k|T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \min\left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_k}(a_t|s_t)} A^{\pi_{\theta_k}}(s_t, a_t),\;\; g(\epsilon, A^{\pi_{\theta_k}}(s_t, a_t))\right),$$

typically via stochastic gradient ascent with Adam.
7: Fit value function by regression on mean-squared error:

$$\phi_{k+1} = \arg\min_{\phi} \frac{1}{|\mathcal{D}_k|T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \left(V_\phi(s_t) - \hat{R}_t\right)^2,$$

typically via some gradient descent algorithm.
8: **end for**

Source: https://spinningup.openai.com/en/latest/algorithms/ppo.html

PPO:

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t\left[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)\right]$$

algorithm	avg. normalized score
No clipping or penalty	-0.39
Clipping, $\epsilon = 0.1$	0.76
Clipping, $\epsilon = 0.2$	**0.82**
Clipping, $\epsilon = 0.3$	0.70
Adaptive KL $d_{\text{targ}} = 0.003$	0.68
Adaptive KL $d_{\text{targ}} = 0.01$	0.74
Adaptive KL $d_{\text{targ}} = 0.03$	0.71
Fixed KL, $\beta = 0.3$	0.62
Fixed KL, $\beta = 1.$	0.71
Fixed KL, $\beta = 3.$	0.72
Fixed KL, $\beta = 10.$	0.69

PPO:

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t \Big[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t) \Big]$$

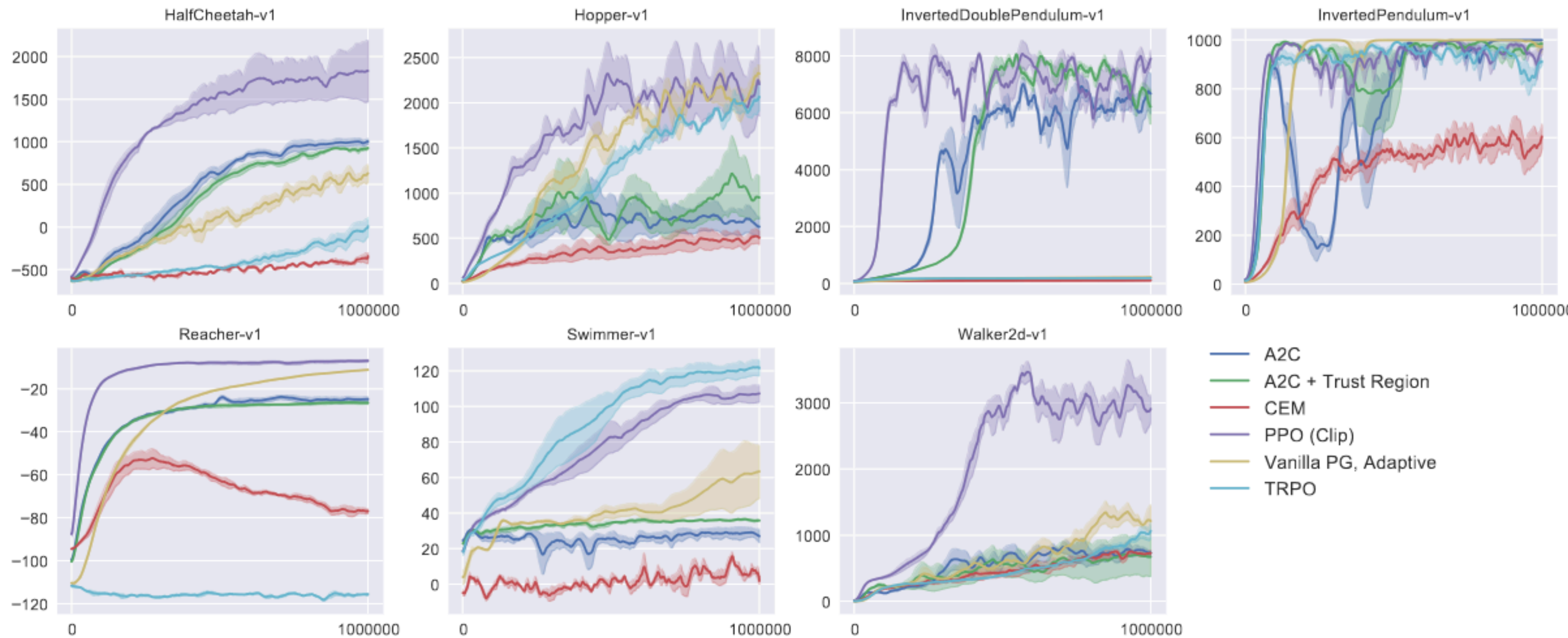


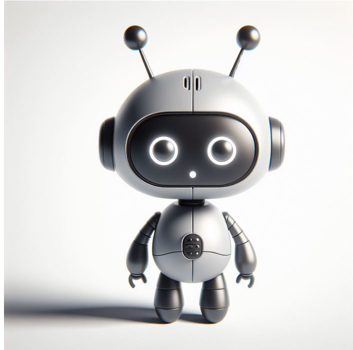
Figure 3: Comparison of several algorithms on several MuJoCo environments, training for one million timesteps.

RL: Review

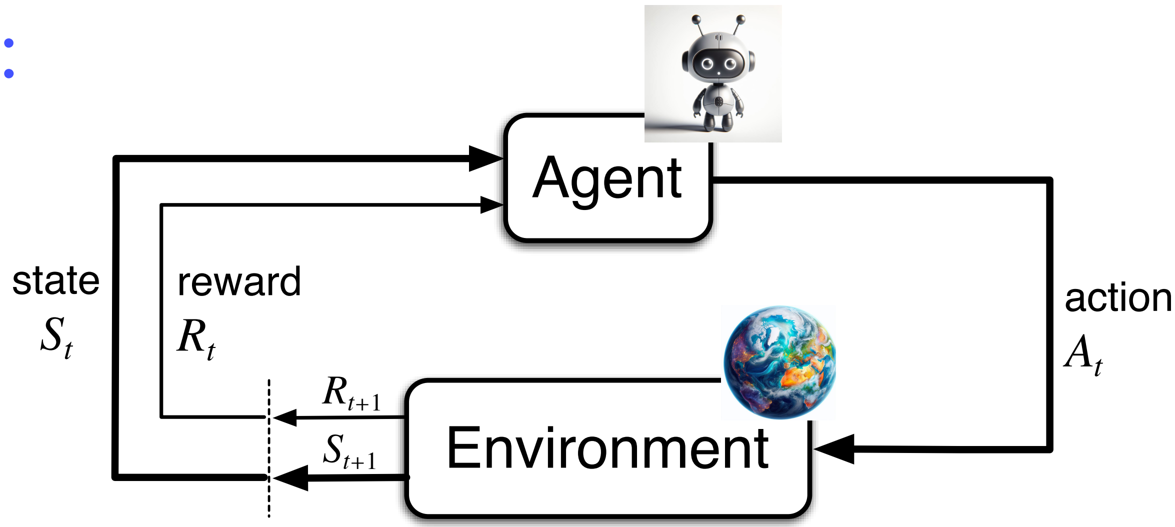
RL Setting:

Environment:

Agent:



RL Setting:



Agent and environment interact at discrete time steps: $t = 0, 1, 2, 3, \ldots$

Agent observes state at step t: $S_t \in \mathcal{S}$

produces action at step t : $A_t \in \mathcal{A}(S_t)$

gets resulting reward: $R_{t+1} \in \mathcal{R} \subset \mathbb{R}$

and resulting next state: $S_{t+1} \in \mathcal{S}^+$

Property of the Environment:

Property of the Environment:

Environment is Markov Decision Process (MDP)

$$p(S_{t+1}, R_{t+1} \mid A_t, S_t, A_{t-1}, S_{t-1}, \ldots, S_0)$$

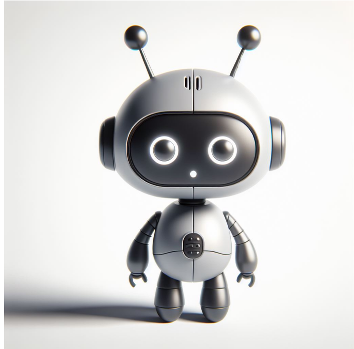
Property of the Environment:

Environment is Markov Decision Process (MDP)

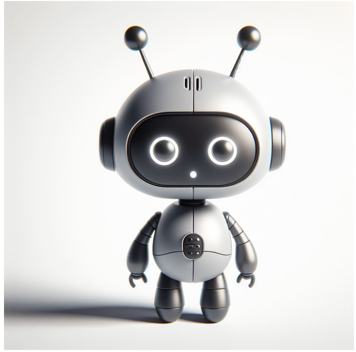
$$p(S_{t+1}, R_{t+1} \mid A_t, S_t, A_{t-1}, S_{t-1}, \ldots, S_0)$$

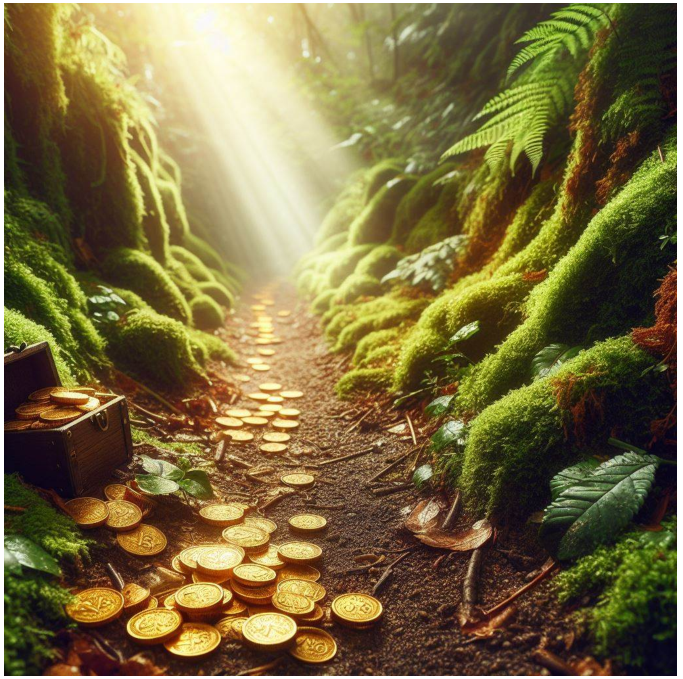
$$= p(S_{t+1}, R_{t+1} \mid A_t, S_t)$$

Agent's objective:



Agent's objective:



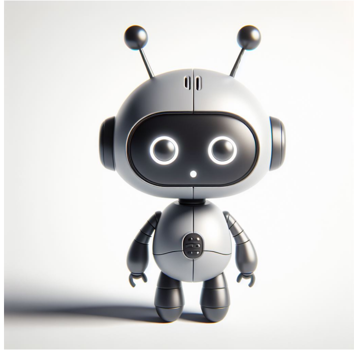


Maximize:

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1},$$

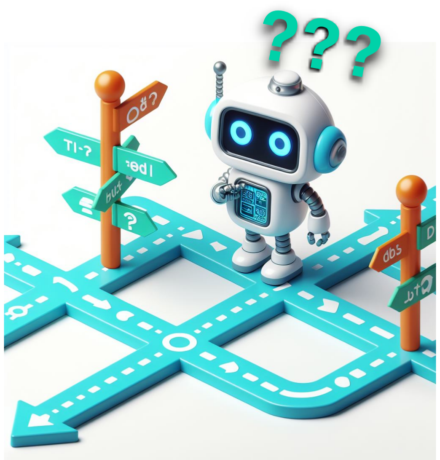
where γ, $0 \leq \gamma \leq 1$, is the **discount rate**.

Agent does 2 things:

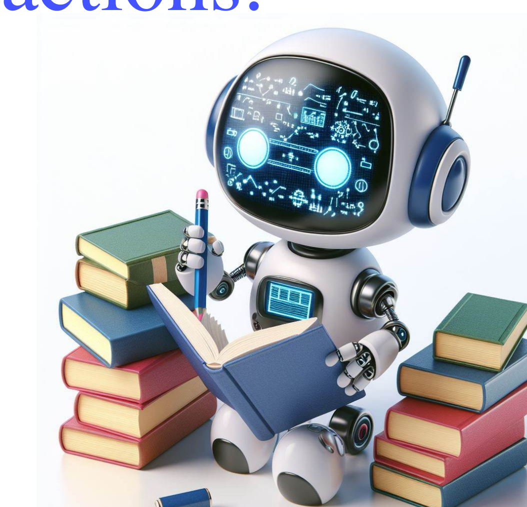


Agent does 2 things:

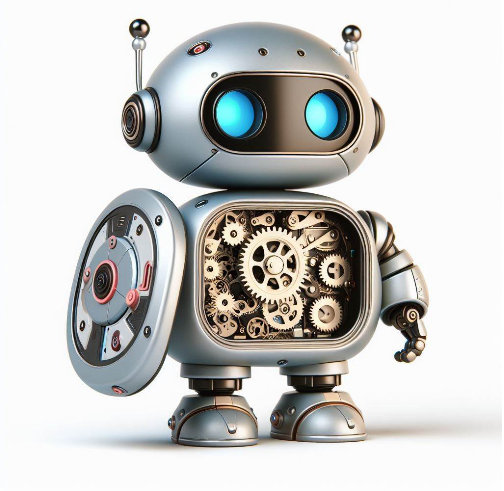
Choose actions:



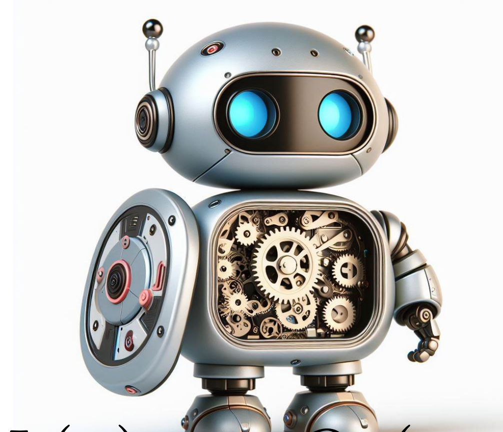
Learn how to choose better actions:



Different Parts of an Agent:



Different Parts of an Agent:



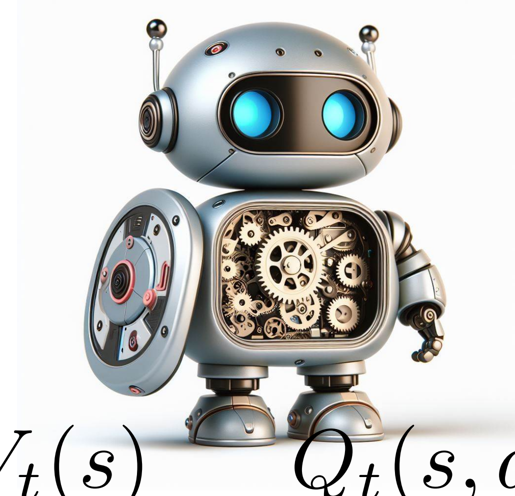
- Value Functions :  $V_t(s)$ $Q_t(s, a)$

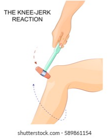
- World Model:  $S_{t+1}, R_{t+1} = M(S_t, A_t, \theta)$

- Policy: 

$A_t = \pi(S_t, \theta)$

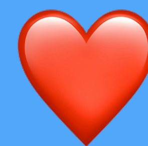
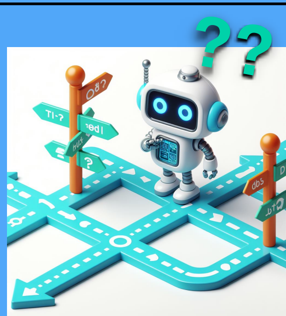
Different Parts of an Agent:



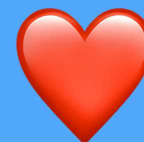
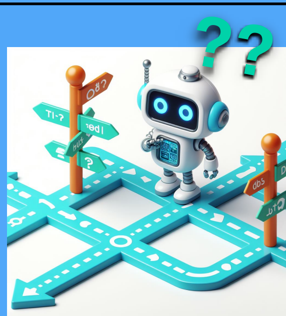
- Value Functions :  $V_t(s)$ $Q_t(s, a)$
- World Model:  $S_{t+1}, R_{t+1} = M(S_t, A_t, \theta)$
- Policy: 

$A_t = \pi(S_t, \theta)$
- (Replay Buffer of past experience)

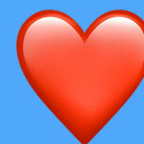
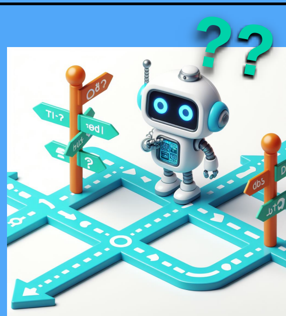
Grid of RL:

			
	Learn $\pi(S_t, \theta)$	Learn $V_t(s)$ $Q_t(s,a)$	Learn $S_{t+1}, R_{t+1} = M(S_t, A_t, \theta)$
	Use $\pi(S_t, \theta)$ to choose action.	Use $V_t(s)$ $Q_t(s,a)$ to choose action.	Use $S_{t+1}, R_{t+1} = M(S_t, A_t, \theta)$ to choose action.

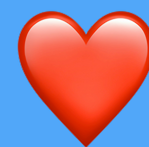
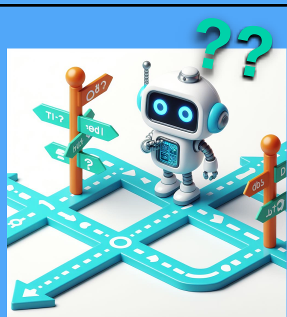
Grid of RL:

			
	Learn $\pi(S_t, \theta)$	Learn $V_t(s)$ $Q_t(s, a)$	Learn $S_{t+1}, R_{t+1} = M(S_t, A_t, \theta)$
	Use $\pi(S_t, \theta)$ to choose action.	Use $V_t(s)$ $Q_t(s, a)$ to choose action.	Use $S_{t+1}, R_{t+1} = M(S_t, A_t, \theta)$ to choose action.

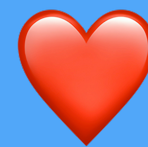
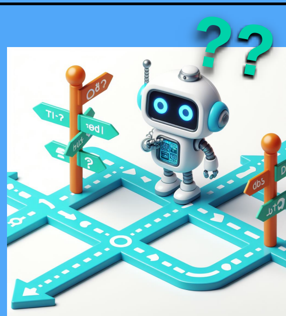
Grid of RL:

			
	Learn $\pi(S_t, \theta)$	Learn $V_t(s)$ $Q_t(s,a)$	Learn $S_{t+1}, R_{t+1} = M(S_t, A_t, \theta)$
	Use $\pi(S_t, \theta)$ to choose action.	Use $V_t(s)$ $Q_t(s,a)$ to choose action.	Use $S_{t+1}, R_{t+1} = M(S_t, A_t, \theta)$ to choose action.

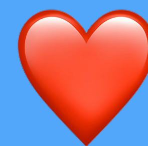
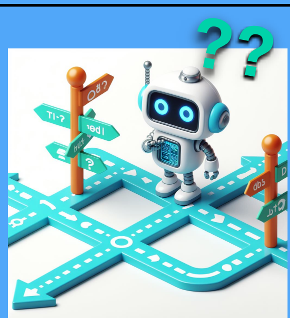
Grid of RL:

			
	Learn $\pi(S_t, \theta)$	Learn $V_t(s)$ $Q_t(s,a)$	Learn $S_{t+1}, R_{t+1} = M(S_t, A_t, \theta)$
	Use $\pi(S_t, \theta)$ to choose action.	Use $V_t(s)$ $Q_t(s,a)$ to choose action.	Use $S_{t+1}, R_{t+1} = M(S_t, A_t, \theta)$ to choose action.

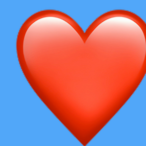
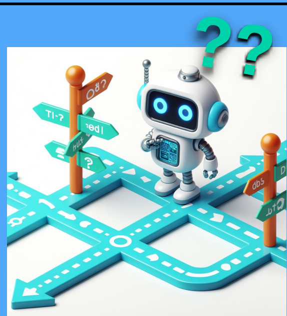
Grid of RL:

			
	Learn $\pi(S_t, \theta)$	Learn $V_t(s)$ $Q_t(s, a)$	Learn $S_{t+1}, R_{t+1} = M(S_t, A_t, \theta)$
	Use $\pi(S_t, \theta)$ to choose action.	Use $V_t(s)$ $Q_t(s, a)$ to choose action.	Use $S_{t+1}, R_{t+1} = M(S_t, A_t, \theta)$ to choose action.

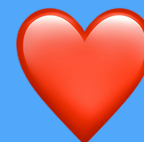
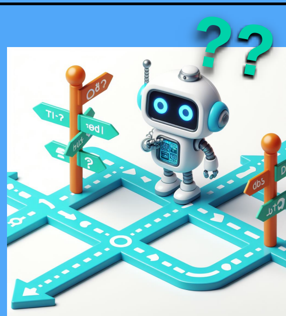
Grid of RL:

			
	Learn $\pi(S_t, \theta)$	Learn $V_t(s)$ $Q_t(s,a)$	Learn $S_{t+1}, R_{t+1} = M(S_t, A_t, \theta)$
	Use $\pi(S_t, \theta)$ to choose action.	Use $V_t(s)$ $Q_t(s,a)$ to choose action.	Use $S_{t+1}, R_{t+1} = M(S_t, A_t, \theta)$ to choose action.

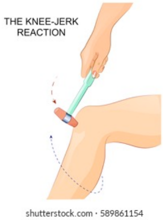
Grid of RL:

			
	Learn $\pi(S_t, \theta)$	Learn $V_t(s)$ $Q_t(s,a)$	Learn $S_{t+1}, R_{t+1} = M(S_t, A_t, \theta)$
	Use $\pi(S_t, \theta)$ to choose action.	Use $V_t(s)$ $Q_t(s,a)$ to choose action.	Use $S_{t+1}, R_{t+1} = M(S_t, A_t, \theta)$ to choose action.

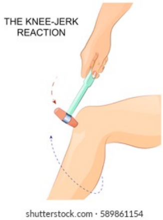
Grid of RL:

			
	Learn $\pi(S_t, \theta)$	Learn $V_t(s)$ $Q_t(s,a)$	Learn $S_{t+1}, R_{t+1} = M(S_t, A_t, \theta)$
	Use $\pi(S_t, \theta)$ to choose action.	Use $V_t(s)$ $Q_t(s,a)$ to choose action.	Use $S_{t+1}, R_{t+1} = M(S_t, A_t, \theta)$ to choose action.

TWO TYPES OF POLICY $\pi(S_t, \theta)$:

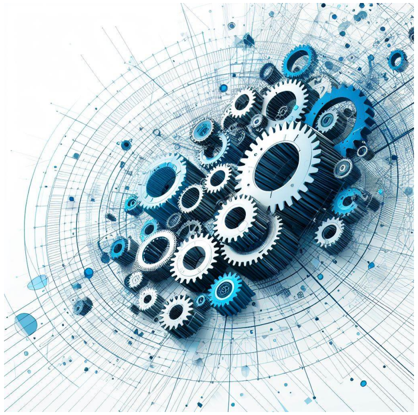


TWO TYPES OF POLICY $\pi(S_t, \theta)$:



Stochastic:

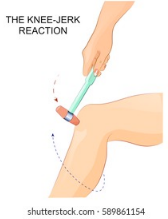
Deterministic:



$\pi(A_t, S_t, \theta)$ is probability.

$A_t = \pi(S_t, \theta)$

Stochastic: $\pi(A_t, S_t, \theta)$ is probability.



Act by sampling from the distribution:

Discrete Actions:

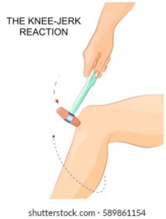
$$\pi(A_i \mid S) = \frac{\exp(\phi(A_i, S))}{\sum_j \exp(\phi(A_j, S))}$$

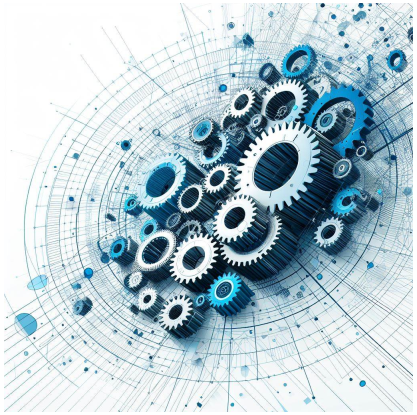
Continuous Actions:

$$\pi(A \mid S) = \mathcal{N}\left(\mu(S), \sigma(S)\right)$$

$$A = \mu(S) + \sigma(S)\epsilon, \qquad \epsilon \sim \mathcal{N}(0,1)$$

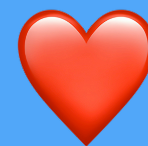
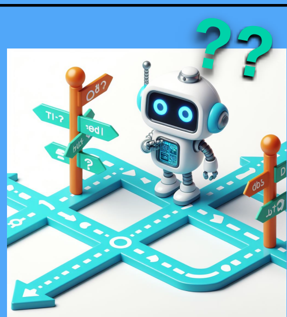
Deterministic: $A_t = \pi(S_t, \theta)$





Act by applying π to state: $A_t = \pi(S_t, \theta)$

Grid of RL:

<table>
<tr><td></td><td>
</td><td></td><td></td></tr>
<tr><td></td><td>Learn
$\pi(S_t, \theta)$</td><td>Learn
$V_t(s)$ $Q_t(s,a)$</td><td>Learn
$S_{t+1}, R_{t+1} = M(S_t, A_t, \theta)$</td></tr>
<tr><td></td><td>Use $\pi(S_t, \theta)$
to choose
action.</td><td>Use
$V_t(s)$ $Q_t(s,a)$
to choose
action.</td><td>Use
$S_{t+1}, R_{t+1} = M(S_t, A_t, \theta)$
to choose
action.</td></tr>
</table>

Value Functions

- The **value of a state** is the expected return starting from that state; depends on the agent's policy:

State - value function for policy π :

$$v_\pi(s) = E_\pi\left\{G_t \mid S_t = s\right\} = E_\pi\left\{\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s\right\}$$

- The **value of an action (in a state)** is the expected return starting after taking that action from that state; depends on the agent's policy:

Action - value function for policy π :

$$q_\pi(s,a) = E_\pi\left\{G_t \mid S_t = s, A_t = a\right\} = E_\pi\left\{\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s, A_t = a\right\}$$

Use $V_t(s)$ $Q_t(s,a)$ to choose action:

Action - value function for policy π :

$$q_\pi(s,a) = E_\pi\left\{G_t \mid S_t = s, A_t = a\right\} = E_\pi\left\{\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s, A_t = a\right\}$$

Use $V_t(s)$ $Q_t(s,a)$ to choose action:

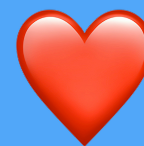
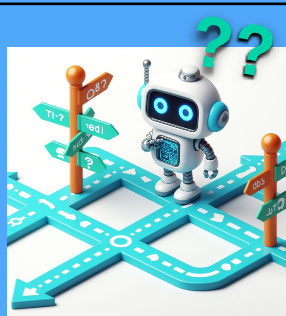
Action - value function for policy π :

$$q_\pi(s,a) = E_\pi\left\{G_t \mid S_t = s, A_t = a\right\} = E_\pi\left\{\sum_{k=0}^{\infty}\gamma^k R_{t+k+1} \mid S_t = s, A_t = a\right\}$$

Act by taking the action which maximizes the expected return according to the estimate Q:

$$A_t = \text{argmax}_a Q(a, S_t)$$

Grid of RL:

			
	Learn $\pi(S_t, \theta)$	Learn $V_t(s)$ $Q_t(s,a)$	Learn $S_{t+1}, R_{t+1} = M(S_t, A_t, \theta)$
	Use $\pi(S_t, \theta)$ to choose action.	Use $V_t(s)$ $Q_t(s,a)$ to choose action.	Use $S_{t+1}, R_{t+1} = M(S_t, A_t, \theta)$ to choose action.

Use $S_{t+1}, R_{t+1} = M(S_t, A_t, \theta)$ to choose action:

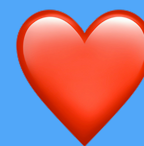
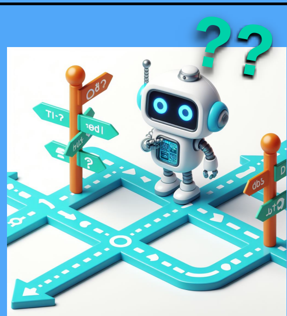
Use $S_{t+1}, R_{t+1} = M(S_t, A_t, \theta)$ to choose action:

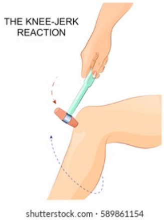
Use $S_{t+1}, R_{t+1} = M(S_t, A_t, \theta)$ in conjunction with planning algorithms (reactive planning):

Examples:

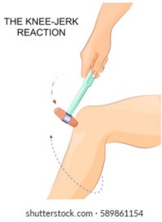
- Sampling future trajectories and taking the best one (as seen in PlaNet)
- Monte-Carlo Tree Search (as seen in MuZero)
- Cross-Entropy Method (with particles) (as seen in Dreamer Paper)

Grid of RL:

			
	Learn $\pi(S_t, \theta)$	Learn $V_t(s)$ $Q_t(s,a)$	Learn $S_{t+1}, R_{t+1} = M(S_t, A_t, \theta)$
	Use $\pi(S_t, \theta)$ to choose action.	Use $V_t(s)$ $Q_t(s,a)$ to choose action.	Use $S_{t+1}, R_{t+1} = M(S_t, A_t, \theta)$ to choose action.



Learning $\pi(S_t, \theta)$:



Learning $\pi(S_t, \theta)$:

Action - value function for policy π :

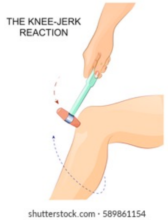
$$q_\pi(s,a) = E_\pi\left\{G_t \mid S_t = s, A_t = a\right\} = E_\pi\left\{\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s, A_t = a\right\}$$

Learning $\pi(S_t, \theta)$:

Action - value function for policy π :

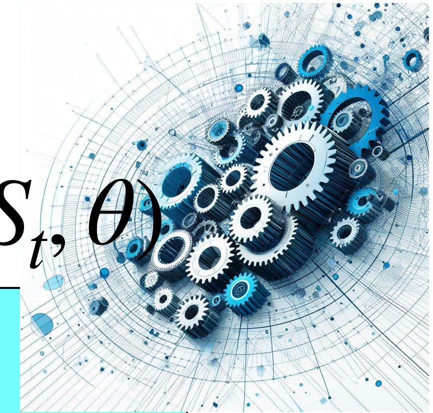
$$q_\pi(s,a) = E_\pi\left\{G_t \mid S_t = s, A_t = a\right\} = E_\pi\left\{\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s, A_t = a\right\}$$

$$\theta_{t+1} = \theta_t + \alpha \underbrace{\nabla_\theta q_\pi}_{\nabla_\theta J}$$



Learning $\pi(S_t, \theta)$:

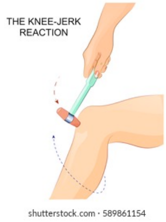
Deterministic and Continuous: $A_t = \pi(S_t, \theta)$



Action - value function for policy π :

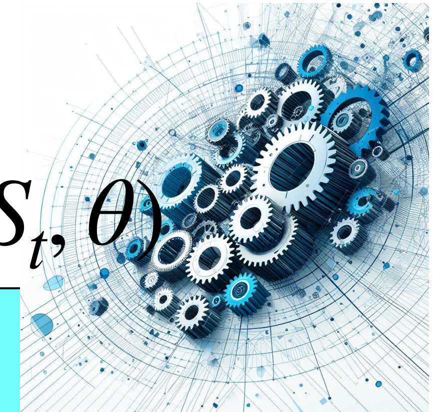
$$q_\pi(s,a) = E_\pi\left\{G_t \mid S_t = s, A_t = a\right\} = E_\pi\left\{\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s, A_t = a\right\}$$

$$J_\theta(\pi \mid S_0 = S) = q_\pi(\pi(S), S) \approx Q_\pi(\pi(S, \theta), S)$$



Learning $\pi(S_t, \theta)$:

Deterministic and Continuous: $A_t = \pi(S_t, \theta)$



Action - value function for policy π :

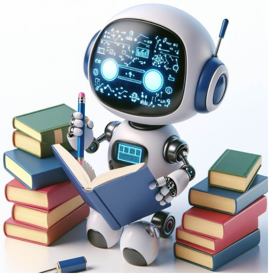
$$q_\pi(s,a) = E_\pi\left\{G_t \mid S_t = s, A_t = a\right\} = E_\pi\left\{\sum_{k=0}^{\infty}\gamma^k R_{t+k+1} \mid S_t = s, A_t = a\right\}$$

$$J_\theta(\pi \mid S_0 = S) = q_\pi(\pi(S), S) \approx Q_\pi(\pi(S, \theta), S)$$

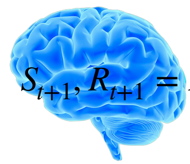
$$\nabla_\theta J_\theta(\pi \mid S_0 = S) \approx \nabla_\theta Q_\pi(\pi(S, \theta), S)$$

$$= \sum_i^m \frac{\partial Q_\pi(A = \pi(S, \theta), S)}{\partial a_i} \nabla_\theta \pi_i(S, \theta)$$

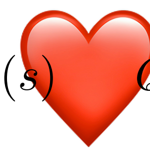
$$= \nabla_A Q_\pi\left(A = \pi(S, \theta), S\right) \nabla_\theta \pi(S, \theta)$$



RL Learning Map:



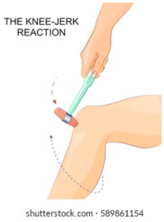
$S_{t+1}, R_{t+1} = M(S_t, A_t, \theta)$



$V_t(s)$ $Q_t(s, a)$

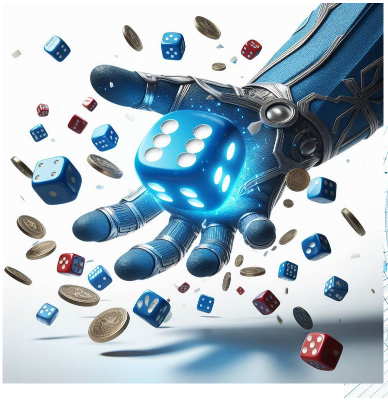
Deterministic Policy Gradient, DDPG

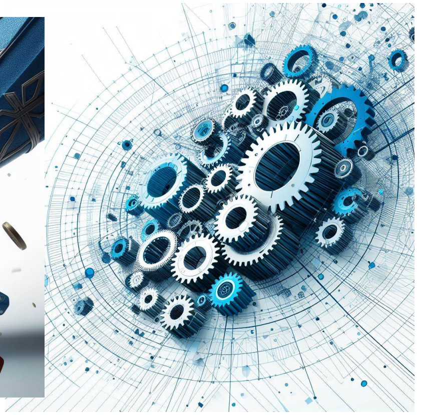
$\pi(S_t, \theta)$



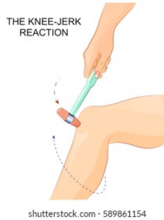
Learning $\pi(S_t, \theta)$:

Deterministic or Stochastic:



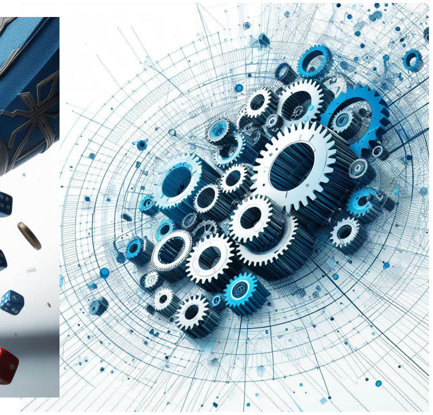


If we can plan with a World-Model M,
and planning gives us a next action A:

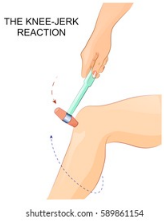


Learning $\pi(S_t, \theta)$:

Deterministic or Stochastic:

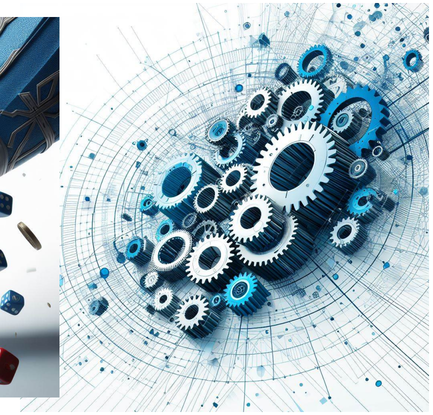


If we can plan with a World-Model M, and planning gives us a next action A: We can can use A as a target for supervised learning of π.



Learning $\pi(S_t, \theta)$:

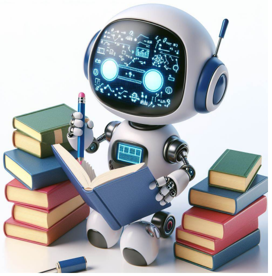
Deterministic or Stochastic:



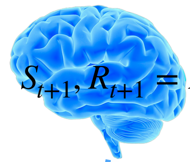
If we can plan with a World-Model M, and planning gives us a next action A: We can can use A as a target for supervised learning of π.

Continuous A: regression problem: MSE loss

Discrete A, stochastic π: classification problem: Cross-Entropy loss

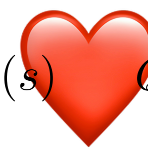


RL Learning Map:



$S_{t+1}, R_{t+1} = M(S_t, A_t, \theta)$

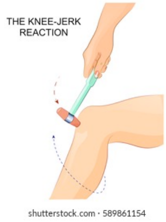
Supervised-Learning of Plan



$V_t(s)$ $Q_t(s, a)$

Deterministic Policy Gradient, DDPG

$\pi(S_t, \theta)$

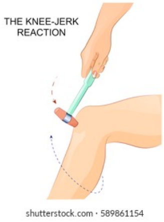


Learning $\pi(S_t, \theta)$:

Stochastic: $\pi(A_t, S_t, \theta)$ is probability.

Policy Gradient Theorem:

$$\nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}}(\pi) = \mathbb{E}_\pi\left[\sum_{t=0}^{T} \left(\gamma^t G_t\right) \nabla_{\boldsymbol{\theta}} \log \pi(A_t | S_t)\right]$$

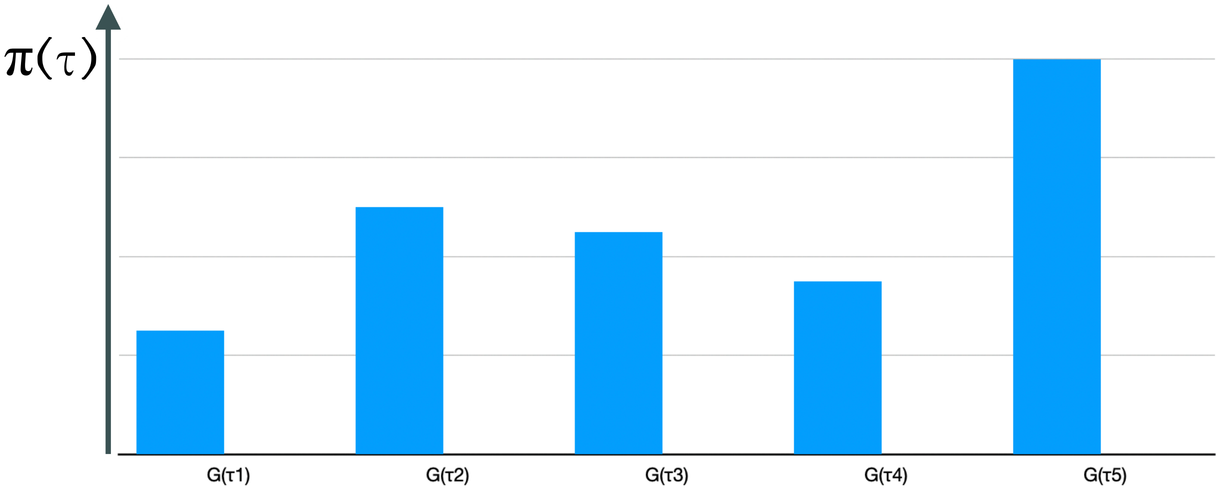


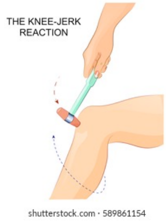
Learning $\pi(S_t, \theta)$:

Stochastic: $\pi(A_t, S_t, \theta)$ is probability.

Policy Gradient Theorem:

$$\nabla_{\theta} J_{\theta}(\pi) = \mathbb{E}_{\pi}[\sum_{t=0}^{T} (\gamma^t G_t) \nabla_{\theta} \log \pi(A_t | S_t)]$$





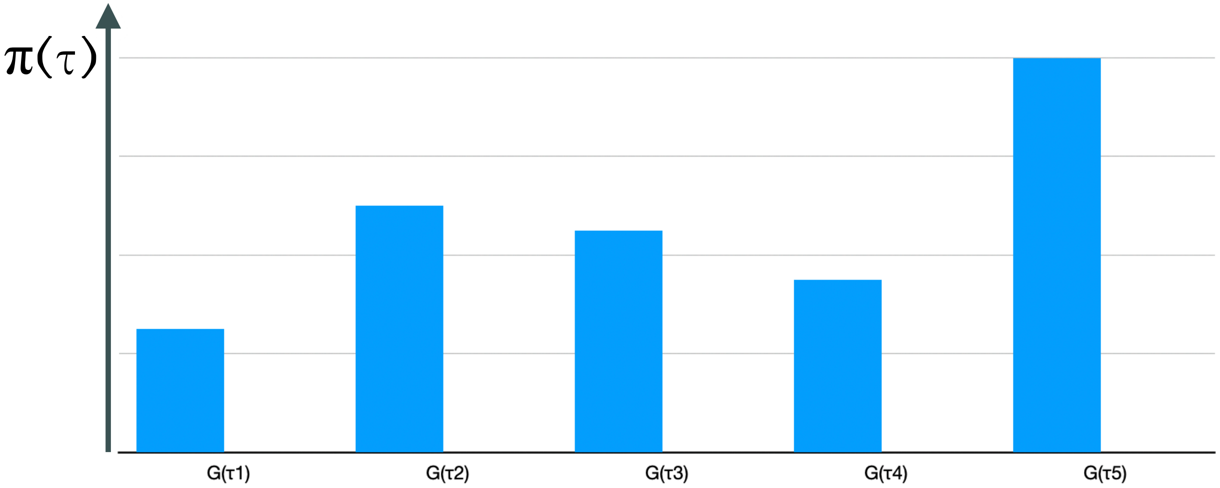
Learning $\pi(S_t, \theta)$:

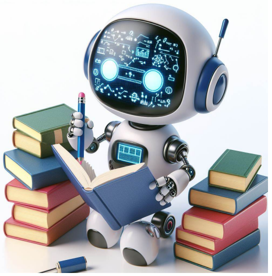
Stochastic: $\pi(A_t, S_t, \theta)$ is probability.

Policy Gradient Theorem:

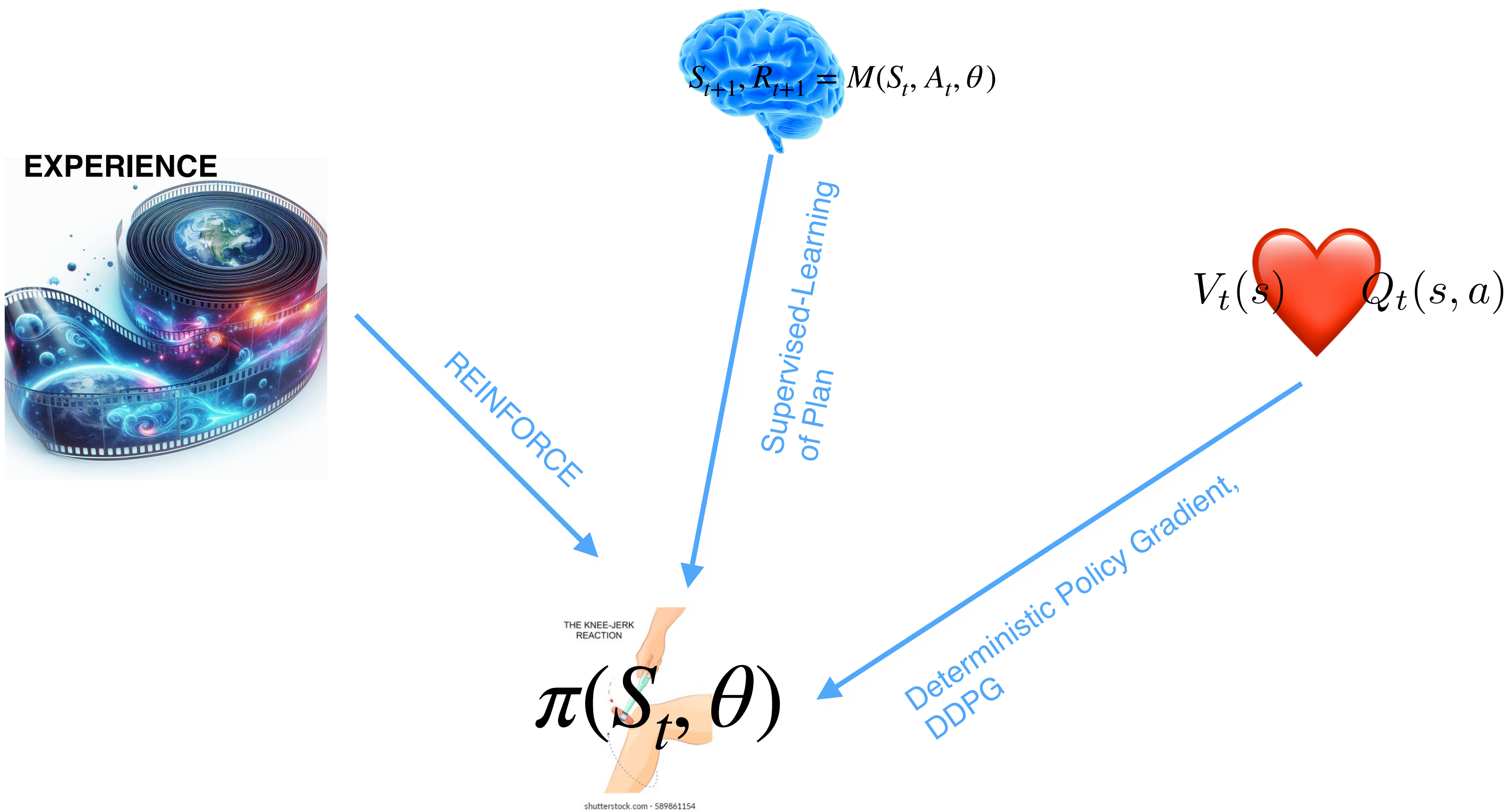
REINFORCE Estimates G with Monte-Carlo

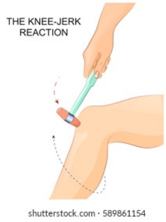
$$\nabla_{\theta} J_{\theta}(\pi) = \mathbb{E}_{\pi}[\sum_{t=0}^{T} (\gamma^t G_t)\, \nabla_{\theta} \log \pi(A_t | S_t)]$$





RL Learning Map:



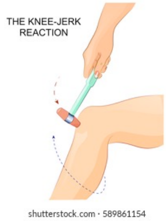


Learning $\pi(S_t, \theta)$:

Stochastic: $\pi(A_t, S_t, \theta)$ is probability.

Policy Gradient Theorem:

$$\nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}}(\pi) = \mathbb{E}_{\pi}[\sum_{t=0}^{T} \left(\gamma^t G_t\right) \nabla_{\boldsymbol{\theta}} \log \pi(A_t | S_t)]$$



Learning $\pi(S_t, \theta)$:

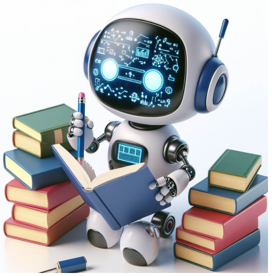
Stochastic: $\pi(A_t, S_t, \theta)$ is probability.

Policy Gradient Theorem:

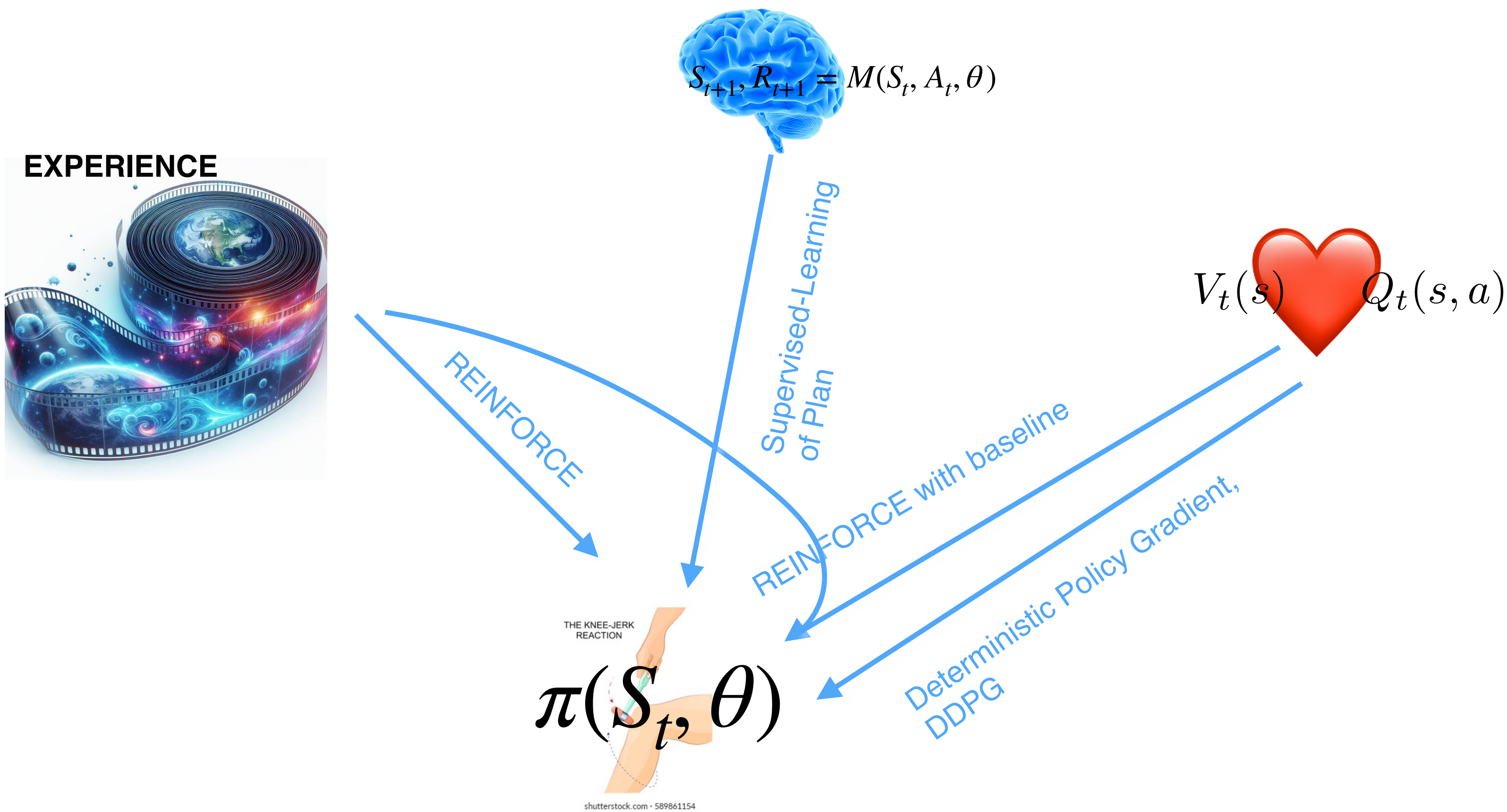
REINFORCE Estimates G with Monte-Carlo

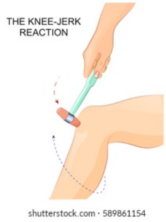
$$\nabla_\theta J_\theta(\pi) = \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t \left(q_\pi(S_t, A_t) - v_\pi(S_t)\right) \nabla_\theta \log(\pi)\right]$$

Advantage



RL Learning Map:





Learning $\pi(S_t, \theta)$:

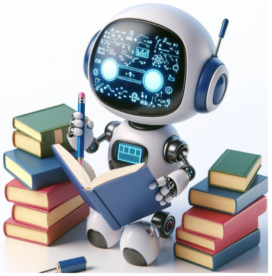
Stochastic: $\pi(A_t, S_t, \theta)$ is probability.

Policy Gradient Theorem:

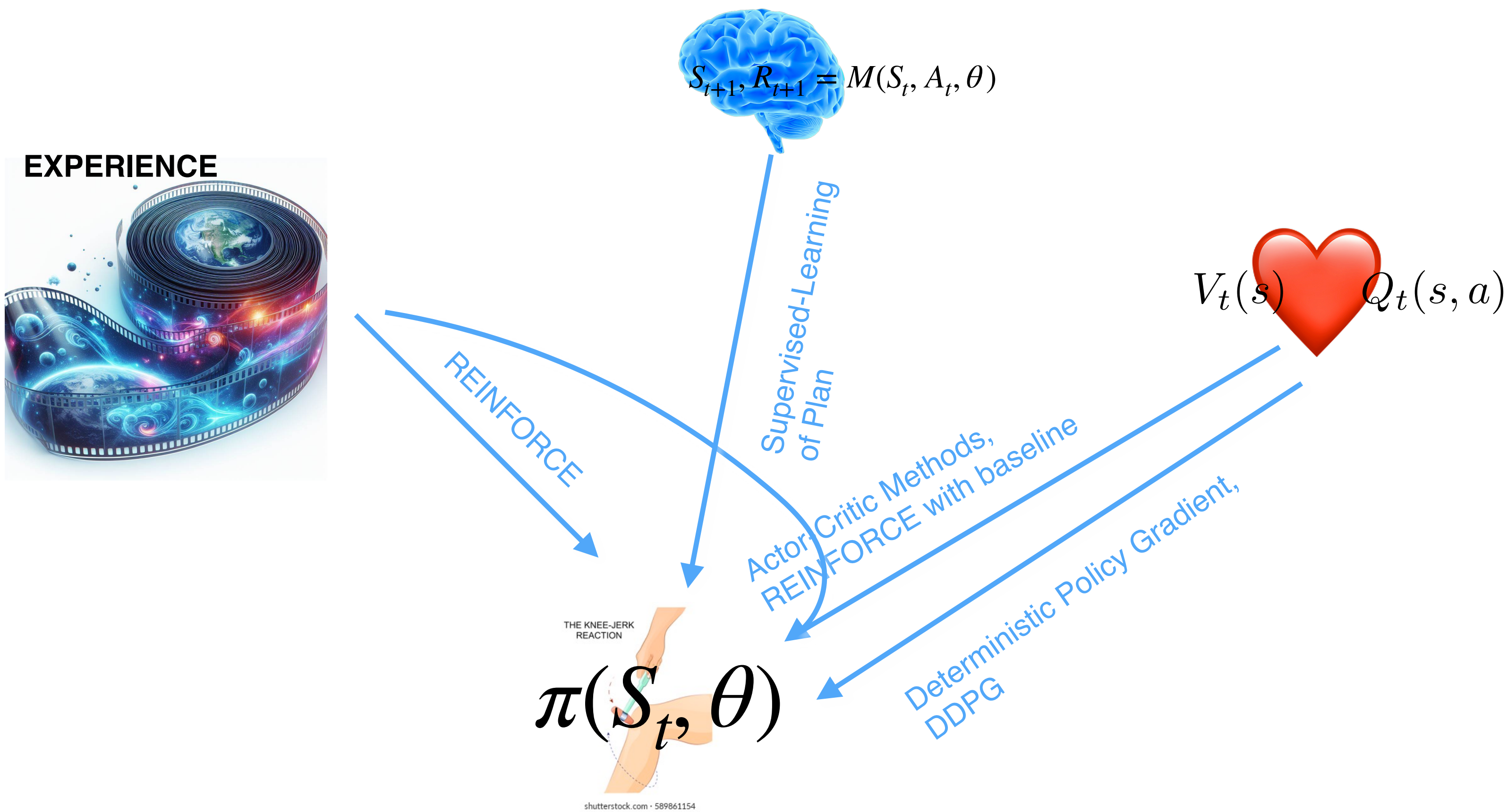
$$\nabla_\theta J_\theta(\pi) = \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t \left(q_\pi(S_t, A_t) - v_\pi(S_t)\right) \nabla_\theta \log(\pi)\right]$$

Advantage

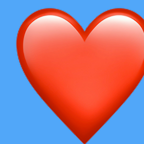
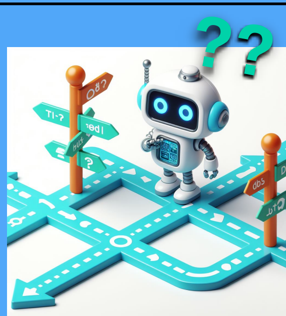
Actor-Critic: use V and/or Q to estimate G or Advantage , e.g. TD(λ)



RL Learning Map:



Grid of RL:

			
	Learn $\pi(S_t, \theta)$	Learn $V_t(s)$ $Q_t(s, a)$	Learn $S_{t+1}, R_{t+1} = M(S_t, A_t, \theta)$
	Use $\pi(S_t, \theta)$ to choose action.	Use $V_t(s)$ $Q_t(s, a)$ to choose action.	Use $S_{t+1}, R_{t+1} = M(S_t, A_t, \theta)$ to choose action.

Learn $V_t(s)$ $Q_t(s,a)$:

Action - value function for policy π :

$$q_\pi(s,a) = E_\pi\left\{G_t \mid S_t = s, A_t = a\right\} = E_\pi\left\{\sum_{k=0}^{\infty}\gamma^k R_{t+k+1} \mid S_t = s, A_t = a\right\}$$

State - value function for policy π :

$$v_\pi(s) = E_\pi\left\{G_t \mid S_t = s\right\} = E_\pi\left\{\sum_{k=0}^{\infty}\gamma^k R_{t+k+1} \mid S_t = s\right\}$$

Learn $V_t(s)$ $Q_t(s,a)$:

4 value functions

	state values	action values
prediction	v_π	q_π
control	v_*	q_*

- All theoretical objects, expected values
- Distinct from their estimates: $V_t(s)$ $Q_t(s,a)$

Learn $V_t(s)$ $Q_t(s,a)$:

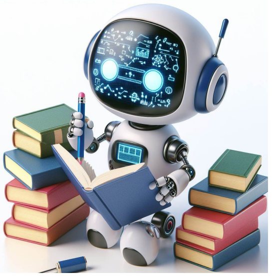
$$q_\pi(s,a) = \mathbb{E}\{G_t \mid S_t = s, A_t = a, A_{t+1:\infty} \sim \pi\} \qquad q_\pi : \mathcal{S} \times \mathcal{A} \to \Re$$

Monte-Carlo Estimate :

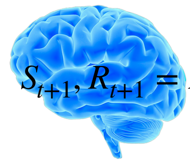
$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1},$$

where $\gamma, 0 \le \gamma \le 1$, is the **discount rate**.

- *Every-Visit MC:* average returns for *every* time *s* is visited in an episode
- *First-visit MC:* average returns only for *first* time *s* is visited in an episode
- Both converge asymptotically



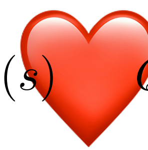
RL Learning Map:



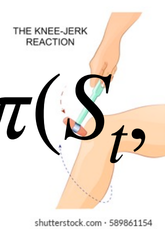
$S_{t+1}, R_{t+1} = M(S_t, A_t, \theta)$

EXPERIENCE

Monte-Carlo Policy Evaluation



$V_t(s)$ $Q_t(s,a)$



$\pi(S_t, \theta)$

Learn $V_t(s)$ $Q_t(s, a)$:

$$q_\pi(s,a) = \mathbb{E}\{G_t \mid S_t = s, A_t = a, A_{t+1:\infty} \sim \pi\} \qquad q_\pi : \mathcal{S} \times \mathcal{A} \to \Re$$

Bootstrapping :

- TD: $G_t^{(1)} \doteq R_{t+1} + \gamma V_t(S_{t+1})$
  - Use V_t to estimate remaining return

- *n*-step TD:
  - 2 step return: $G_t^{(2)} \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 V_t(S_{t+2})$
  - *n*-step return: $G_t^{(n)} \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n V_t(S_{t+n})$

    with $G_t^{(n)} \doteq G_t$ if $t + n \geq T$

Learn $V_t(s)$ $Q_t(s, a)$:

$$q_\pi(s, a) = \mathbb{E}\{G_t \mid S_t = s, A_t = a, A_{t+1:\infty} \sim \pi\} \qquad q_\pi : \mathcal{S} \times \mathcal{A} \to \Re$$

(Expected) SARSA (Bellman Eqn) :

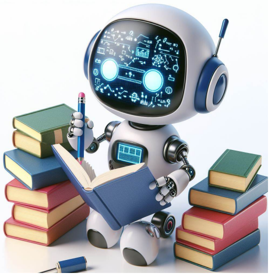
$$\begin{aligned} Q(S_t, A_t) &\leftarrow Q(S_t, A_t) + \alpha\Big[R_{t+1} + \gamma \mathbb{E}[Q(S_{t+1}, A_{t+1}) \mid S_{t+1}] - Q(S_t, A_t)\Big] \\ &\leftarrow Q(S_t, A_t) + \alpha\Big[R_{t+1} + \gamma \sum \pi(a|S_{t+1}) Q(S_{t+1}, a) - Q(S_t, A_t)\Big] \end{aligned}$$

Learn $V_t(s)$ $Q_t(s,a)$:

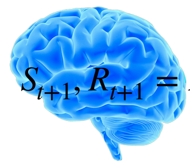
$$q_*(s,a) = \max_\pi q_\pi(s,a) \qquad q_* : \mathcal{S} \times \mathcal{A} \to \Re$$

Q-Learning (Bellman Optimality Eqn):

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \Big[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t) \Big]$$



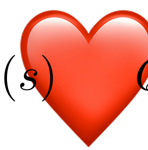
RL Learning Map:

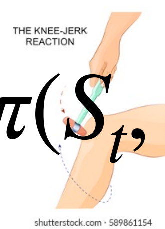


$S_{t+1}, R_{t+1} = M(S_t, A_t, \theta)$

Bootstrap methods:
Q-Learning, SARSA, TD(λ), n-step TD

Monte-Carlo Policy Evaluation

$V_t(s)$  $Q_t(s, a)$



$\pi(S_t, \theta)$

Learn $V_t(s)$ $Q_t(s,a)$ through pro-active planning:

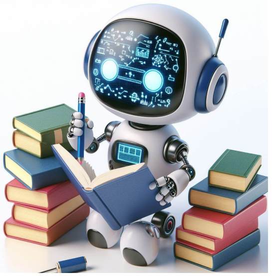
USE IMAGINED EXPERIENCE USING MODEL M:

(Expected) SARSA (Bellman Eqn) :

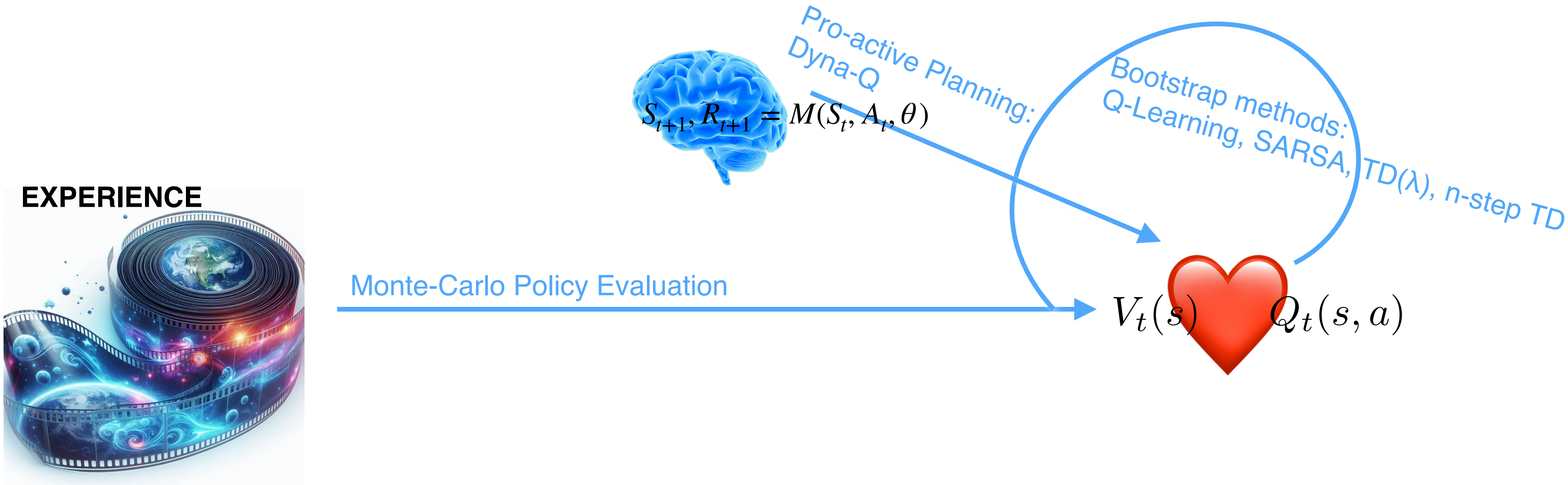
$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \Big[R_{t+1} + \gamma \mathbb{E}[Q(S_{t+1}, A_{t+1}) \mid S_{t+1}] - Q(S_t, A_t) \Big]$$
$$\leftarrow Q(S_t, A_t) + \alpha \Big[R_{t+1} + \gamma \sum \pi(a|S_{t+1}) Q(S_{t+1}, a) - Q(S_t, A_t) \Big]$$

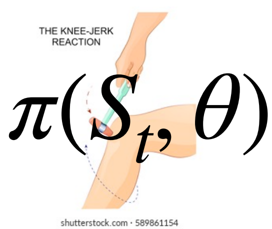
Q-Learning (Bellman Optimality Eqn):

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \Big[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t) \Big]$$

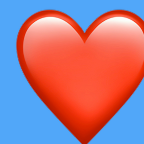
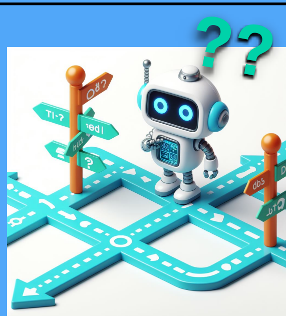


RL Learning Map:





Grid of RL:

			
	Learn $\pi(S_t, \theta)$	Learn $V_t(s)$ $Q_t(s,a)$	Learn $S_{t+1}, R_{t+1} = M(S_t, A_t, \theta)$
	Use $\pi(S_t, \theta)$ to choose action.	Use $V_t(s)$ $Q_t(s,a)$ to choose action.	Use $S_{t+1}, R_{t+1} = M(S_t, A_t, \theta)$ to choose action.

Learn $S_{t+1}, R_{t+1} = M(S_t, A_t, \theta)$:

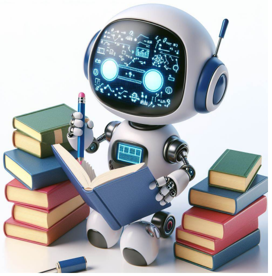
Use transition $S_t, A_t, R_{t+1}, S_{t+1}$:

Supervised learning

- Target for $\hat{S}_{t+1}, \hat{R}_{t+1} = M(S_t, A_t, \theta)$ is S_{t+1}, R_{t+1}
- Target for inverse model

$$\hat{S}_t, \hat{R}_{t+1} = M_{inv}(S_{t+1}, A_t, \psi)$$

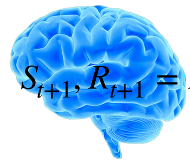
is S_t, R_{t+1}



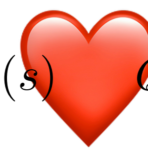
RL Learning Map:

EXPERIENCE

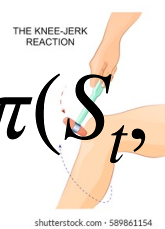
Supervised Model Learning



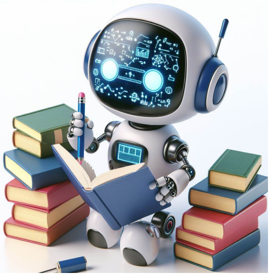
$S_{t+1}, R_{t+1} = M(S_t, A_t, \theta)$



$V_t(s)$ $Q_t(s, a)$



$\pi(S_t, \theta)$



RL Learning Map: