Model: predict what env will do next

Modeled: dynamic programing
find smallest $\pi \to \pi^*$

$\pi \underset{\pi \to greedy(V)}{\overset{V \to V_\pi}{\rightleftarrows}} V$

Modeless: Monte Carlo
Temporal-difference } on-policy
Q-learning off-policy

Agent discovers good policy, from exp of env, wo lose much reward

Value func: $V_\pi(s) = E_\pi[R_{t+1} + \gamma R_{t+2} \cdots | S_t = s]$
prediction of future reward

Reward(R): $R_s = E[R_{t+1} | S_t = s]$

Return(G): $\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$

State trans prob(P) $= P(s'|s,a)$
• state trans matrix: [# of states X # of states]

Markov reward process <S, P, R, γ>

Bellman eq = immediate reward + successor reward

$$V(s) = E[R_{t+1} + \gamma V(S_{t+1}) | S_t]$$
$$= R + \gamma P V_{t+1} = (I - \gamma P)^{-1} R$$

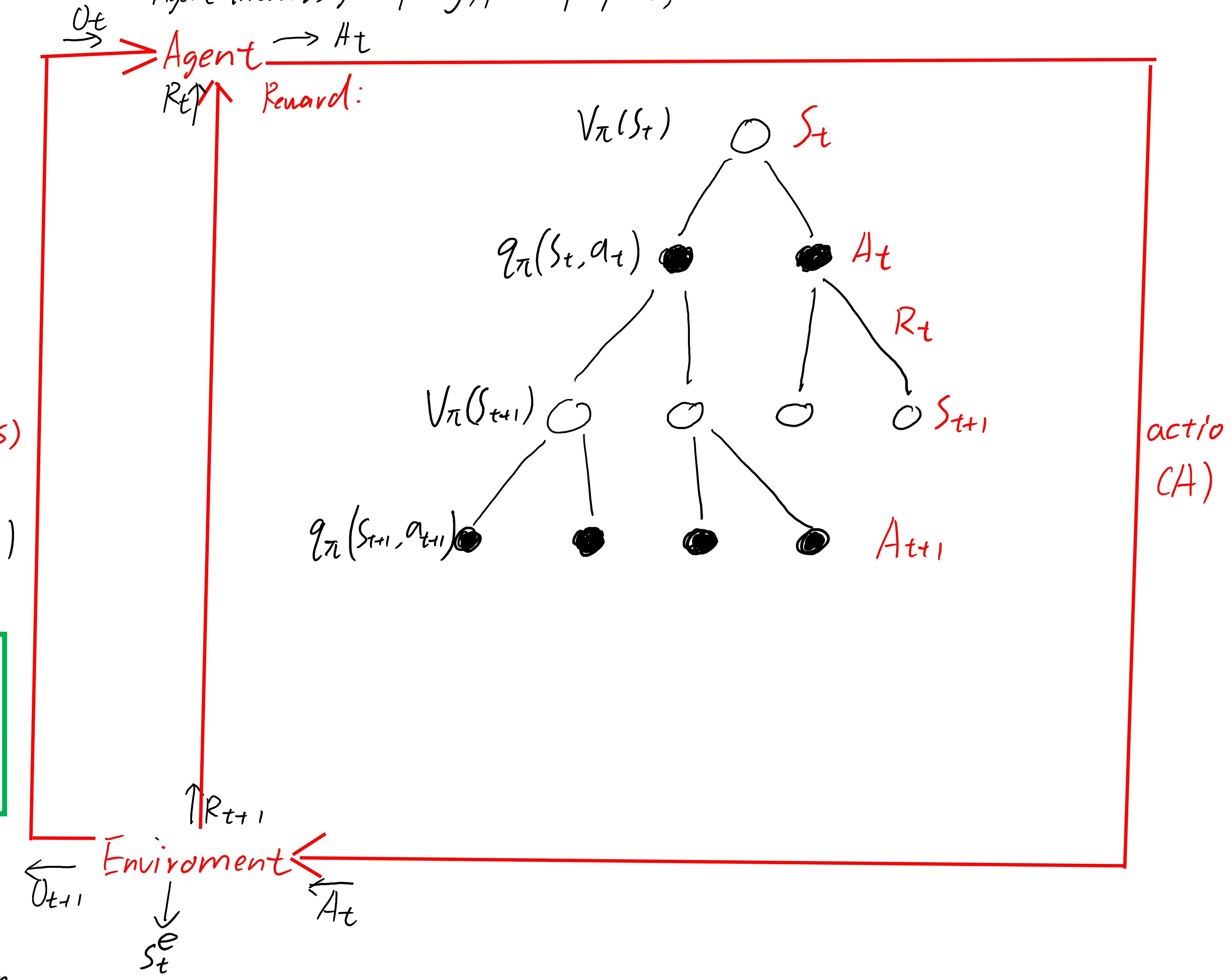
$O(\text{\# of states}^3)$ DP

=> finite states => MC
TD

infinite states => policy gradient

Policy(π): map from state to action; deterministic: $a = \pi(s)$
stochastic: $\pi(a|s) = P(A_t | S_t)$
strict sense stationary (time independent)

Markov state(s)
$P(S_{t+1} | S)$
$= P(S_{t+1} | S_1, \cdots S_t)$



Observation
Full: $O_t = S_t^a = S_t^e \Rightarrow MDP$
Partial: $S_t^a \neq S_t^e$ conversion

Markov decision process <S, P, R, γ, A>

Only depends on current state

$$P_{ss'}^{\pi} = \sum_{a \in A} \pi(a|s) P_{ss'}^{a}, \quad R_s^{\pi} = \sum_{a \in A} \pi(a|s) R_s^a$$

State value func:

expected return following policy π

$$V_\pi(s) = E_\pi[G_t | S_t] = E_\pi[R_{t+1} + \gamma V_\pi(S_{t+1}) | S_t]$$
$$= \sum_{a \in A} \pi(a|s) \left(R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a V_\pi(s') \right)$$

action value func:

expected return following action then policy

$$q_\pi(s,a) = E_\pi[G_t | S_t, A_t] = E_\pi[R_{t+1} + \gamma q(S_{t+1}, a_{t+1}) | S_t, A_t]$$
$$= R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a \sum_{a' \in A} \pi(a'|s') q_\pi(s', a')$$