RL: Lecture 18

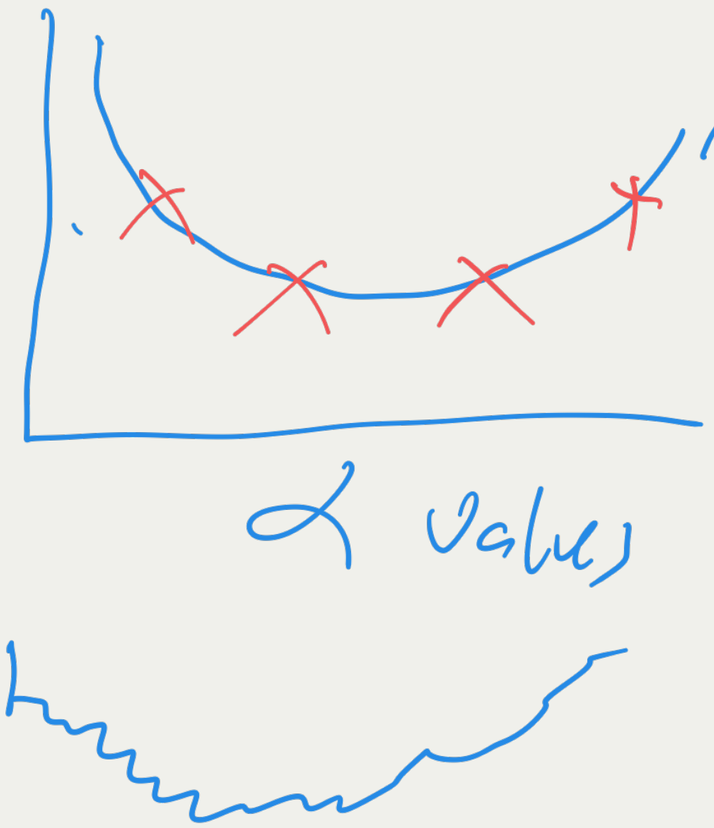
Harvey Mudd College

April 6, 2020

Neil Rhodes

PA 5

Quiz 8 available today from 1 – 11:59 PDT?

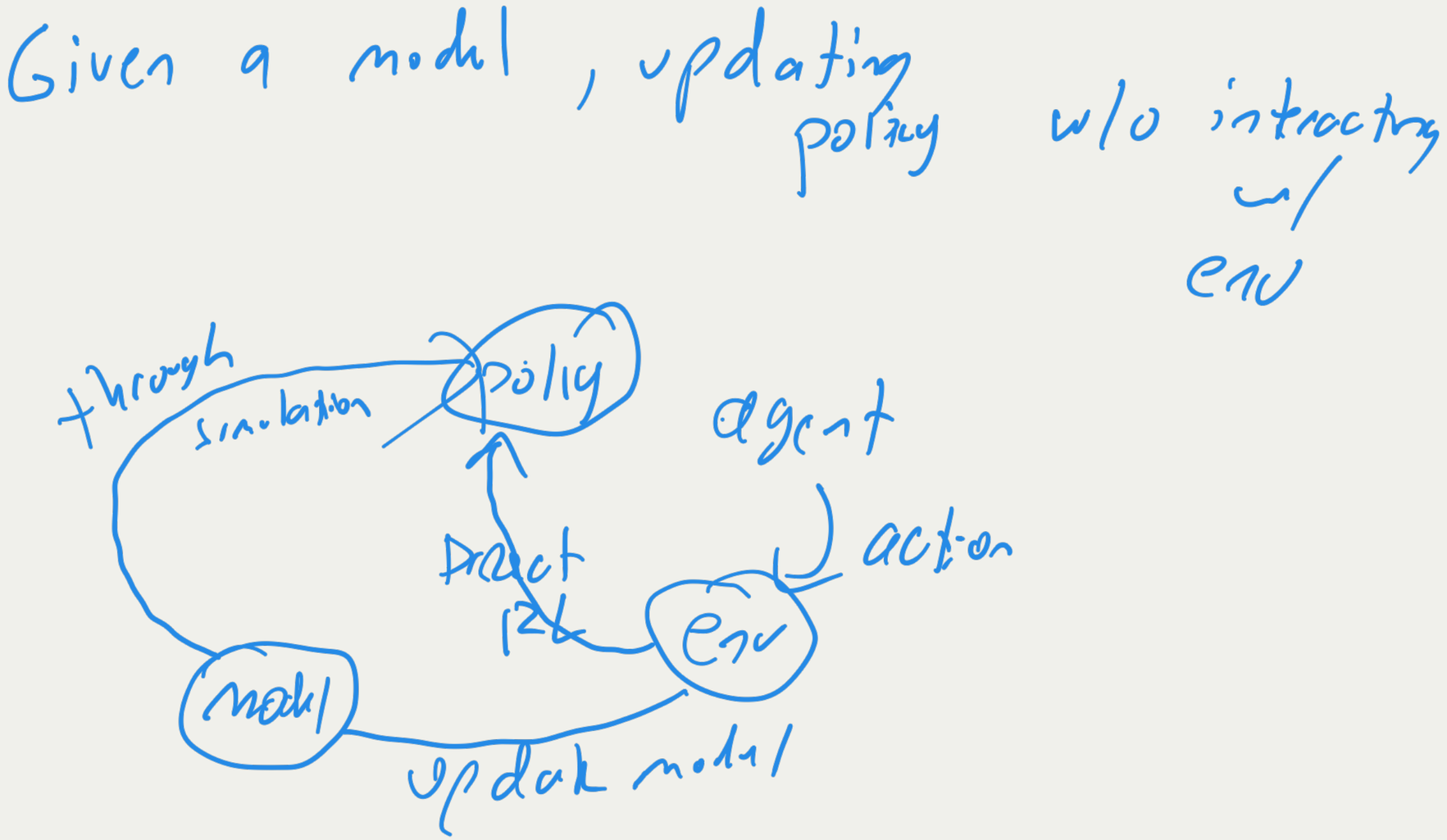


Same episode

PRNG

pseudo random number generator

What is planning?

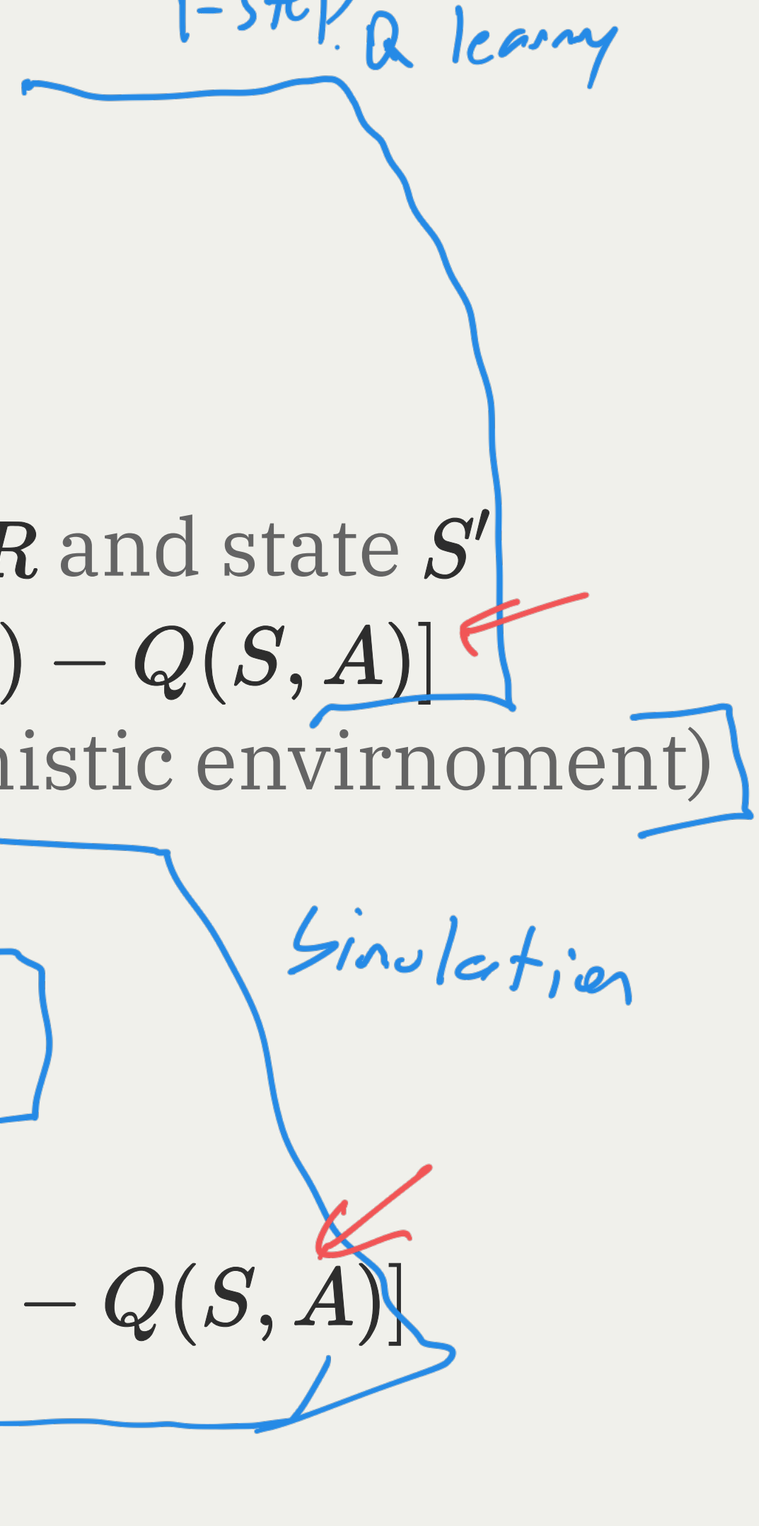


Tabular Dyna-Q Algorithm

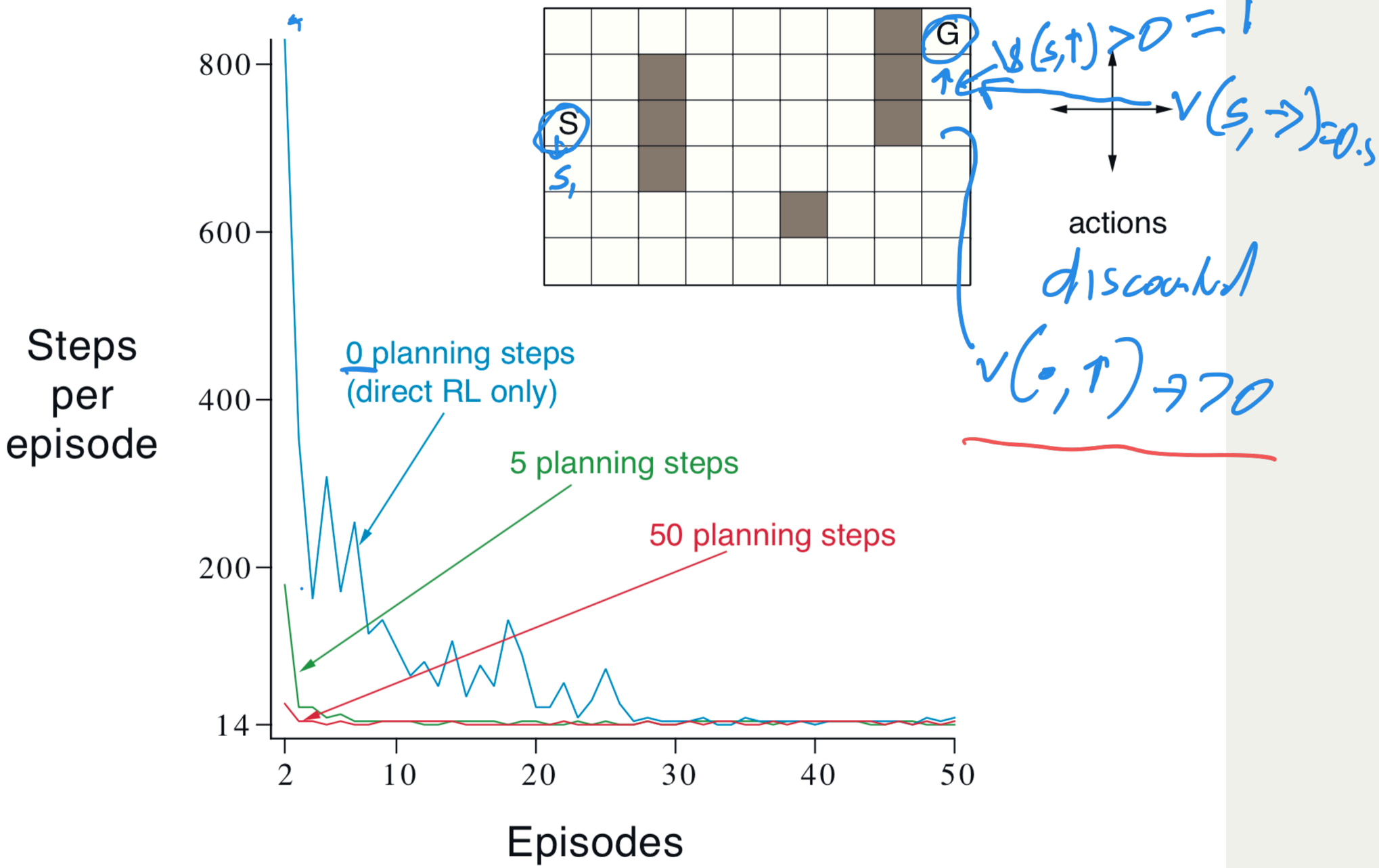
Initialize $Q(s,a)$ and $Model(s,a)$ for all a, s

Loop forever:

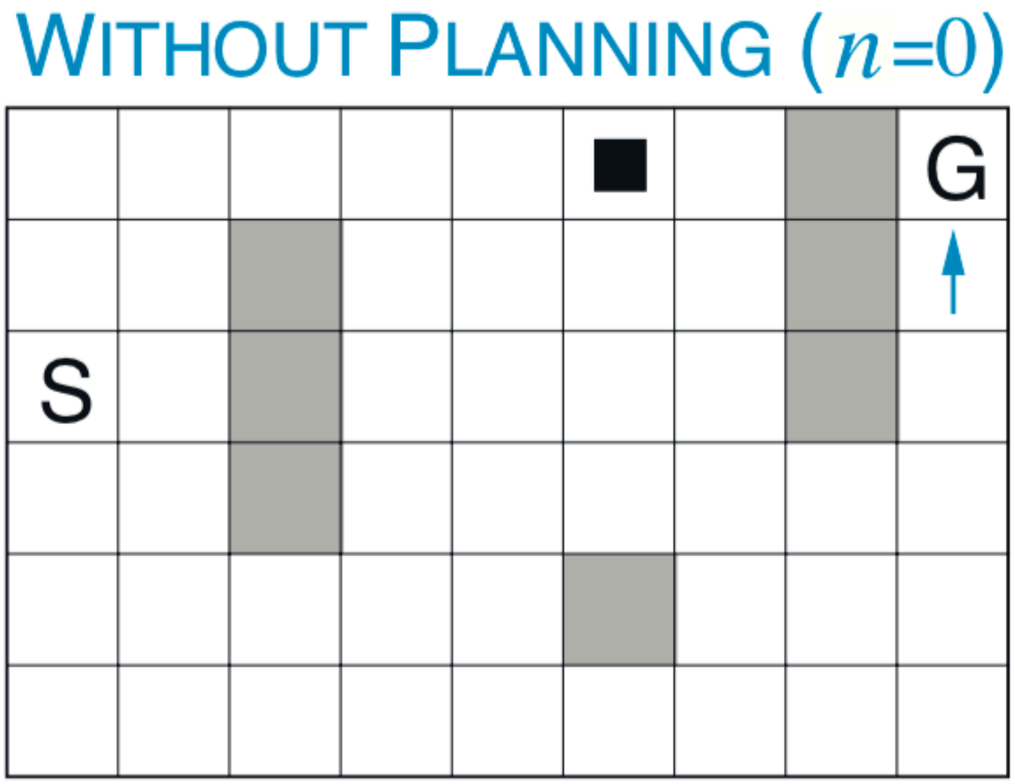
1. $S \leftarrow$ current (nonterminal) state
2. $A \leftarrow \epsilon - \mathbf{greedy}(S,Q)$
3. Take action A; observe resultant reward R and state S'
4. $Q(S,A) \leftarrow Q(S,A) + \alpha[R + \gamma \mathbf{max}_a Q(S',a) - Q(S,A)]$
5. $\mathbf{Model}(S,A) \leftarrow (R,S')$ (assumes deterministic envirnoment)
6. Loop repeat n times
   - $S \leftarrow$ random previously observed state
   - $A \leftarrow$ random action previously taken in S
   - $R, S' \leftarrow \mathbf{Model}(S,A)$
   - $Q(S,A) \leftarrow Q(S,A) + \alpha[R + \gamma \mathbf{max}_a Q(S',a) - Q(S,A)]$

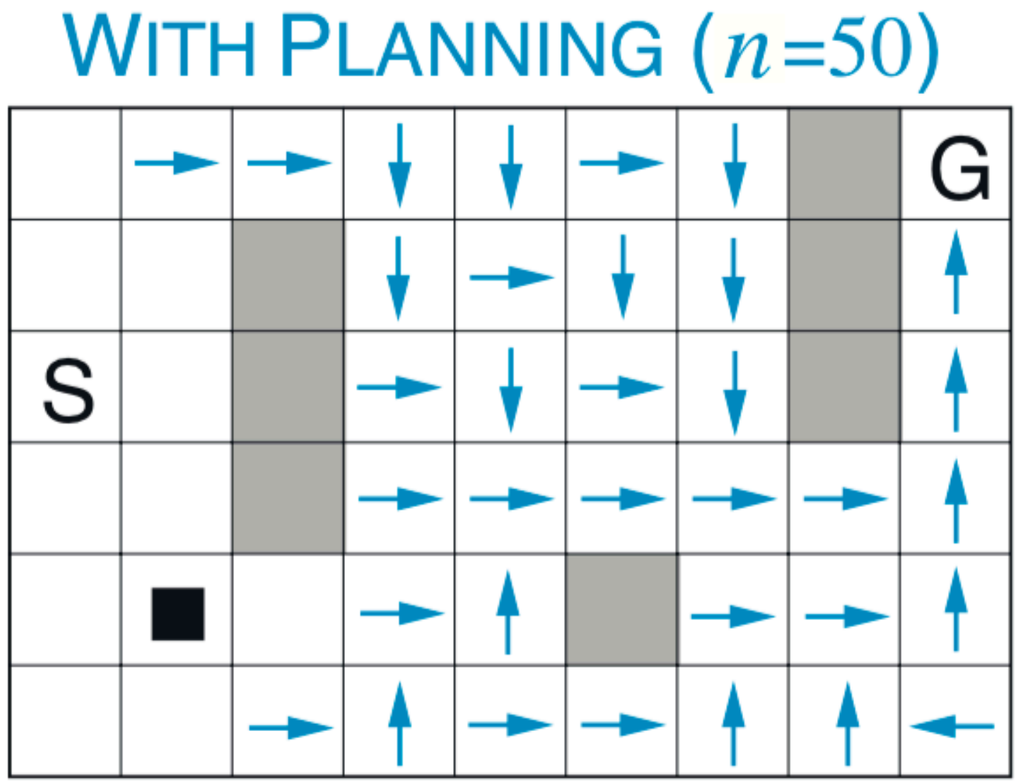


Dyna Maze (Figure 8.2)



Dyna Maze (Figure 8.3)



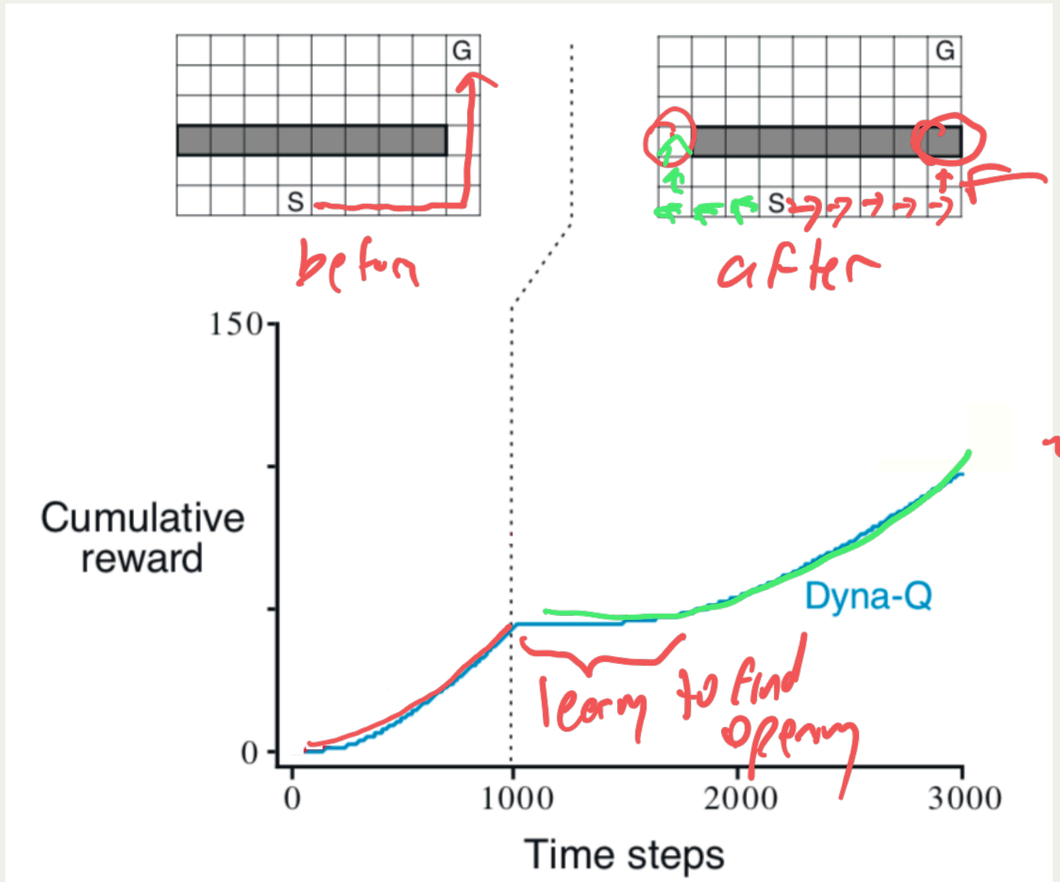


after ≈ 400 steps

~~about~~ after ≈ 114 steps

When the Model is Wrong: Optimistic Model

changing environment



$Q(\cdot,\tau)$ reduce this

$model(\cdot,\tau) = (\cdot,0)$

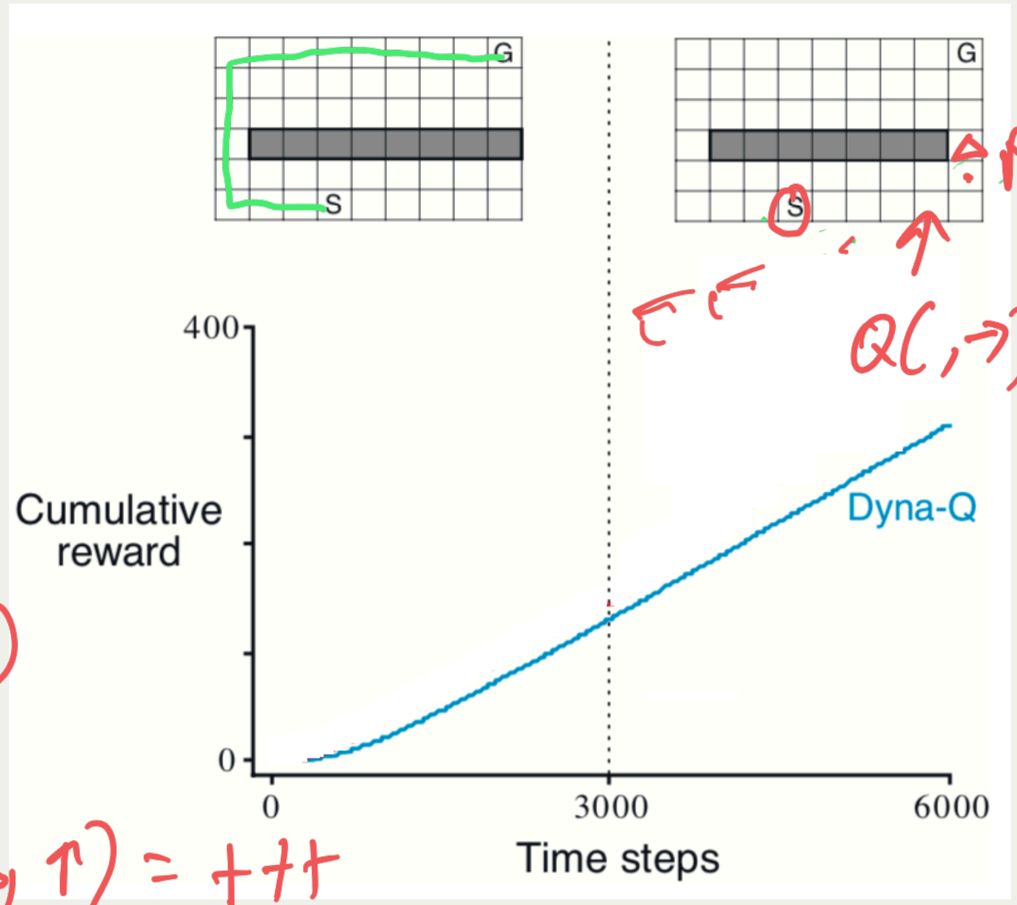
planning

$$Q(\cdot,\tau) \mathrel{+}= \alpha\left[0 + \gamma \max_a Q(\cdot,a) - Q(\cdot,\tau)\right]$$

$$\mathrel{+}= \alpha\left[0 + \gamma Q(\cdot,\theta) - Q(\cdot,a)\right]$$

$$= \alpha\left[(\gamma-1)\,Q(\cdot,\tau)\right]$$

When the Model is Wrong: Pessimistic Model



Q(S,→) → −
model remain (S,0)

Q(·,↑) →
model change to (△,0)

ε ?

(△,a)

Q(△,↑) = +?+

Q(·,↑) → +
Q(S,→) → +
has Q(·,→) → +
model Q(S,↓) → +

Q(·,→)

[model(·,↑) =
(·,0)

in order
to change
is to try
(·,↑) in real-world

In real world, only way
to do that is a bunch of exploratory steps

encourage explor stale parts of the model

Dyna-Q+: Dyna-Q + heuristics for encouraging model updates

- Provide an implicit reward to exploring stale transitions

$$Q(S,A) \leftarrow Q(S,A) + \alpha[R + \kappa\sqrt{\tau(S,A)} + \gamma \max_a Q(S',a) - Q(S,A)]$$

only in planning

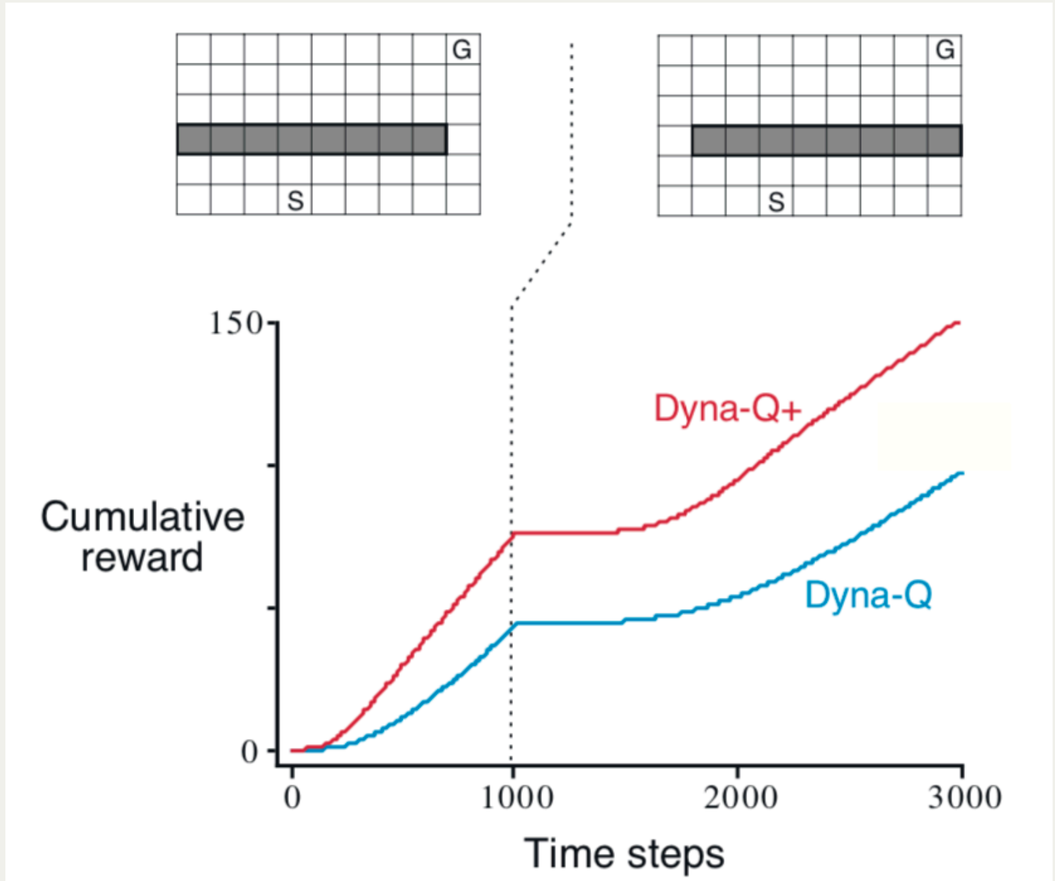
based on model

- Allows actions that have never been tried from a state to be considered in planning (initial model: such an action leads back to the same state with a reward of 0)

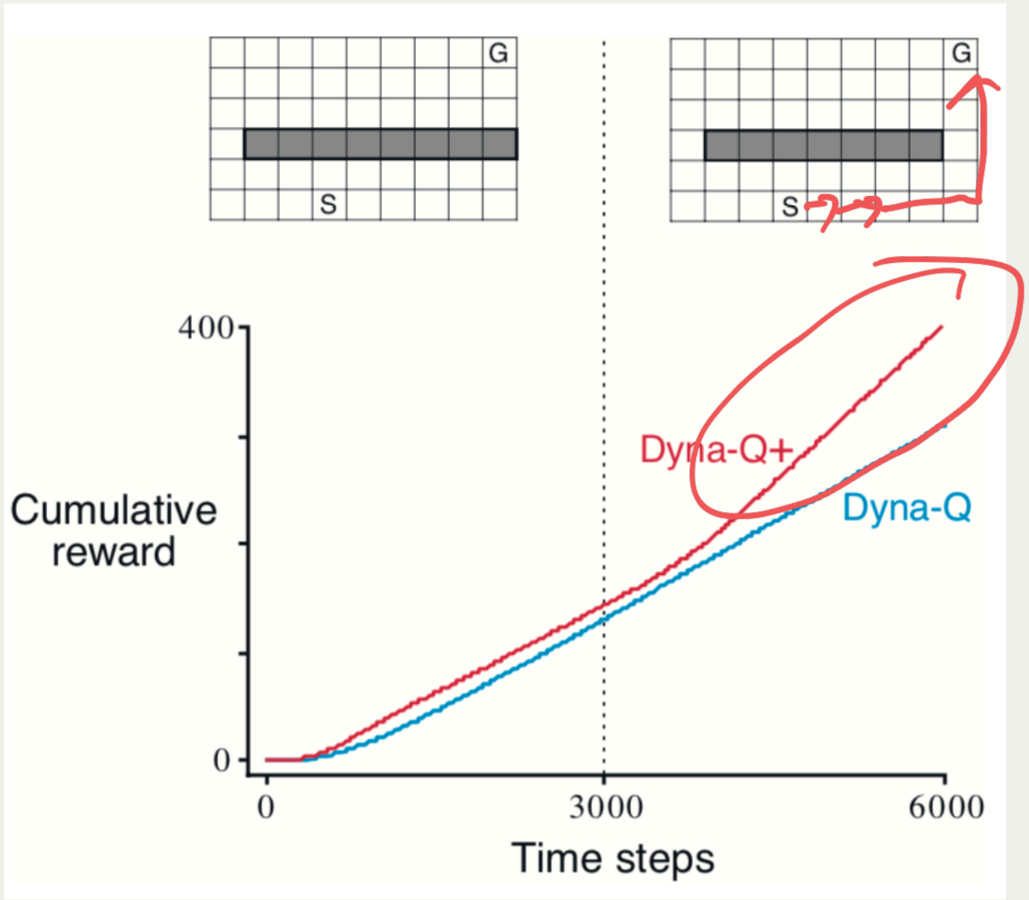
K = constant

$\tau(S,A)$ measures how long since we tried S,A in real world

When the Model is Wrong: Optimistic Model



When the Model is Wrong: Pessimistic Model

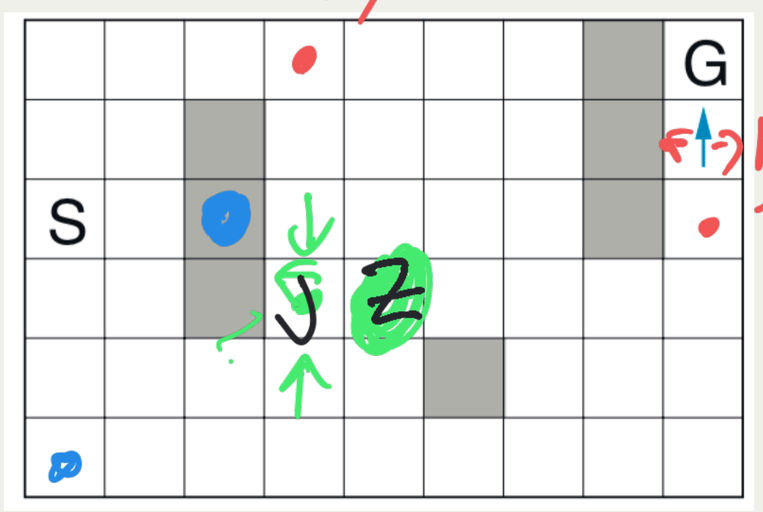


Dyna-Q tries all state-action pairs uniformly.

Is there a better way?

model(•, →) = (•, 0)

@ beginning of second episode



Q(S', A') changes from 0.5 to 11.5 large

Q(S, A) changes from 0.5623 to 0.5624

based on how changed small

how many non-zero Q(state, action)

One

in our model

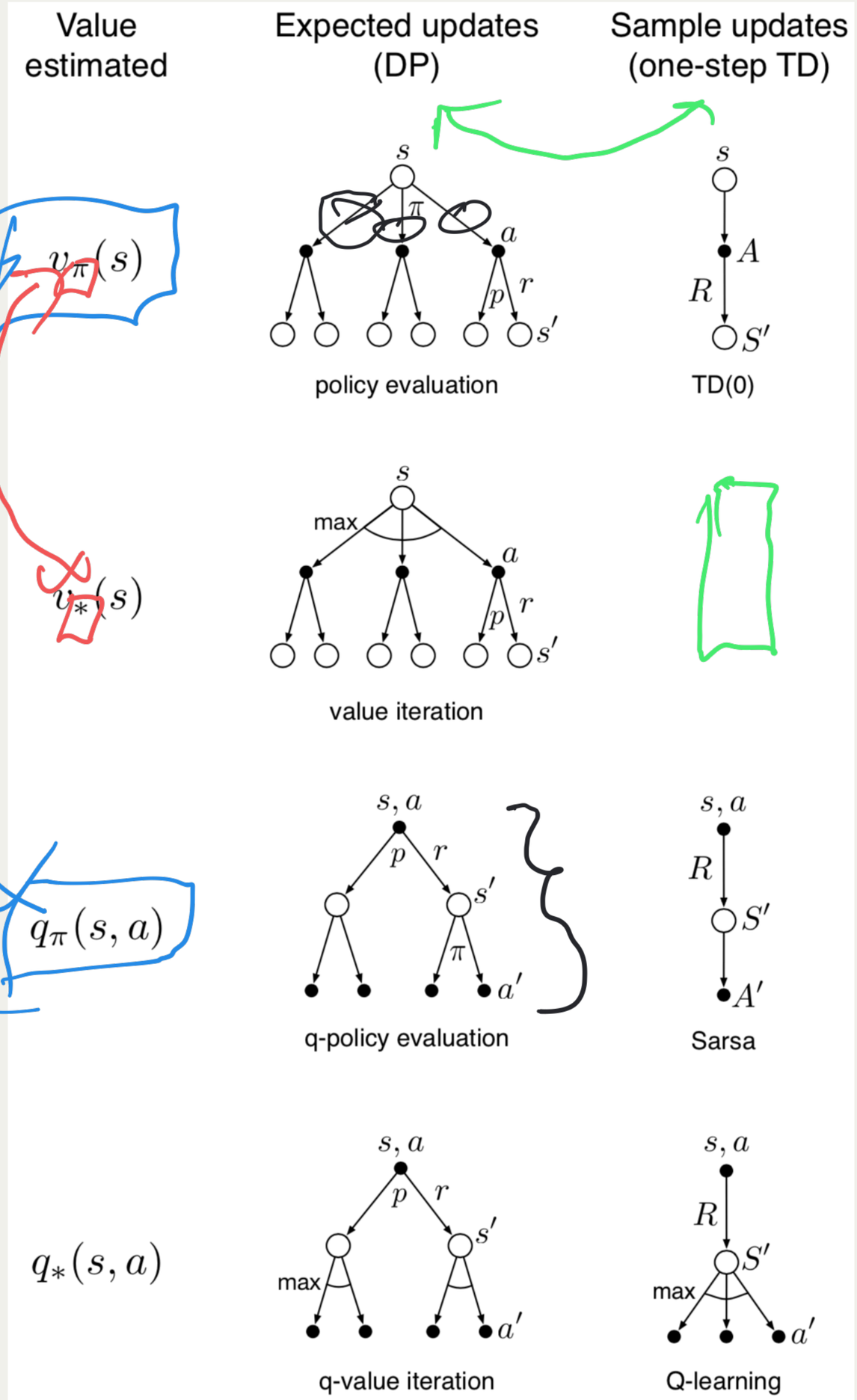
(Z, ←) → (S, ∅)

Prioritized Sweeping (Det. Env.)

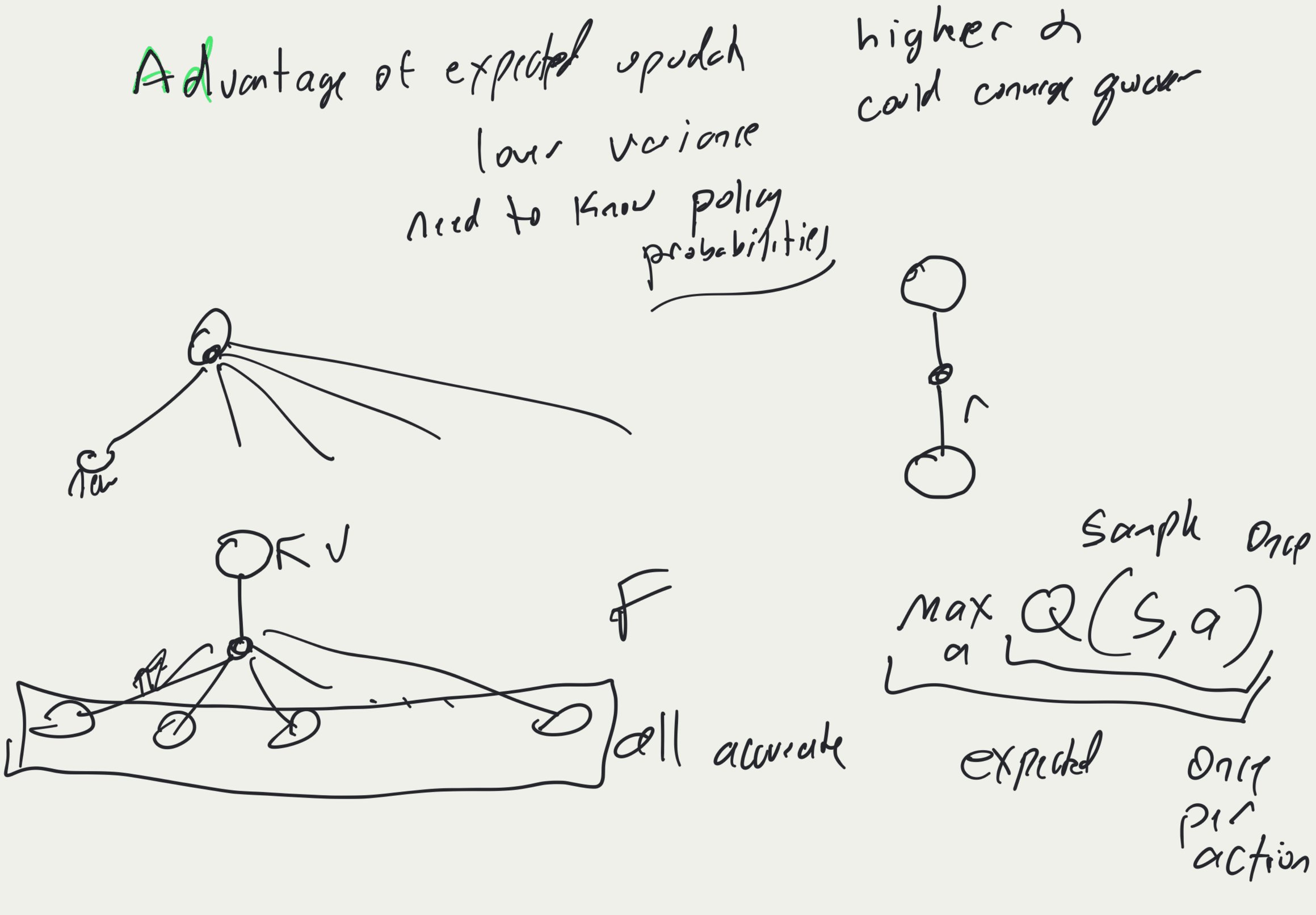
Initialize $Q(s,a)$ and $Model(s,a)$ for all a, s, $PQueue$ to empty
Loop forever:
1. $S \leftarrow$ current (nonterminal) state; 2. $A \leftarrow \text{policy}(S,Q)$
3. Take action A; observe resultant reward R and state S'
4. $\text{Model}(S,A) \leftarrow (R,S')$ (assumes deterministic envirnoment)
5. $P = \lvert R + \gamma \max_a Q(S',a) - Q(S,A) \rvert$ ← in charge of Q
6. If $P > \Theta$, insert (S,A) into $PQueue$ with priority P Θ = min visible change
7. Loop repeat n times while $PQueue$ is not empty
   a. $S,A = first(PQueue); R,S' \leftarrow \text{Model}(S,A)$
   b. $Q(S,A) \leftarrow Q(S,A) + \alpha[R + \gamma \max_a Q(S',a) - Q(S,A)]$
   c. Loop for all $\bar{S}, \bar{A}$ predicted to lead to S: From the model
      i. $\bar{R} =$ pred. reward for $\bar{S}, \bar{A}, S$
      ii. $P = \bar{R} + \gamma \max_a Q(S',a) - Q(\bar{S},\bar{A})$ ΔQ would be
      iii. If $P > \Theta$, insert $(\bar{S},\bar{A})$ into $PQueue$ with priority P

Dimensions

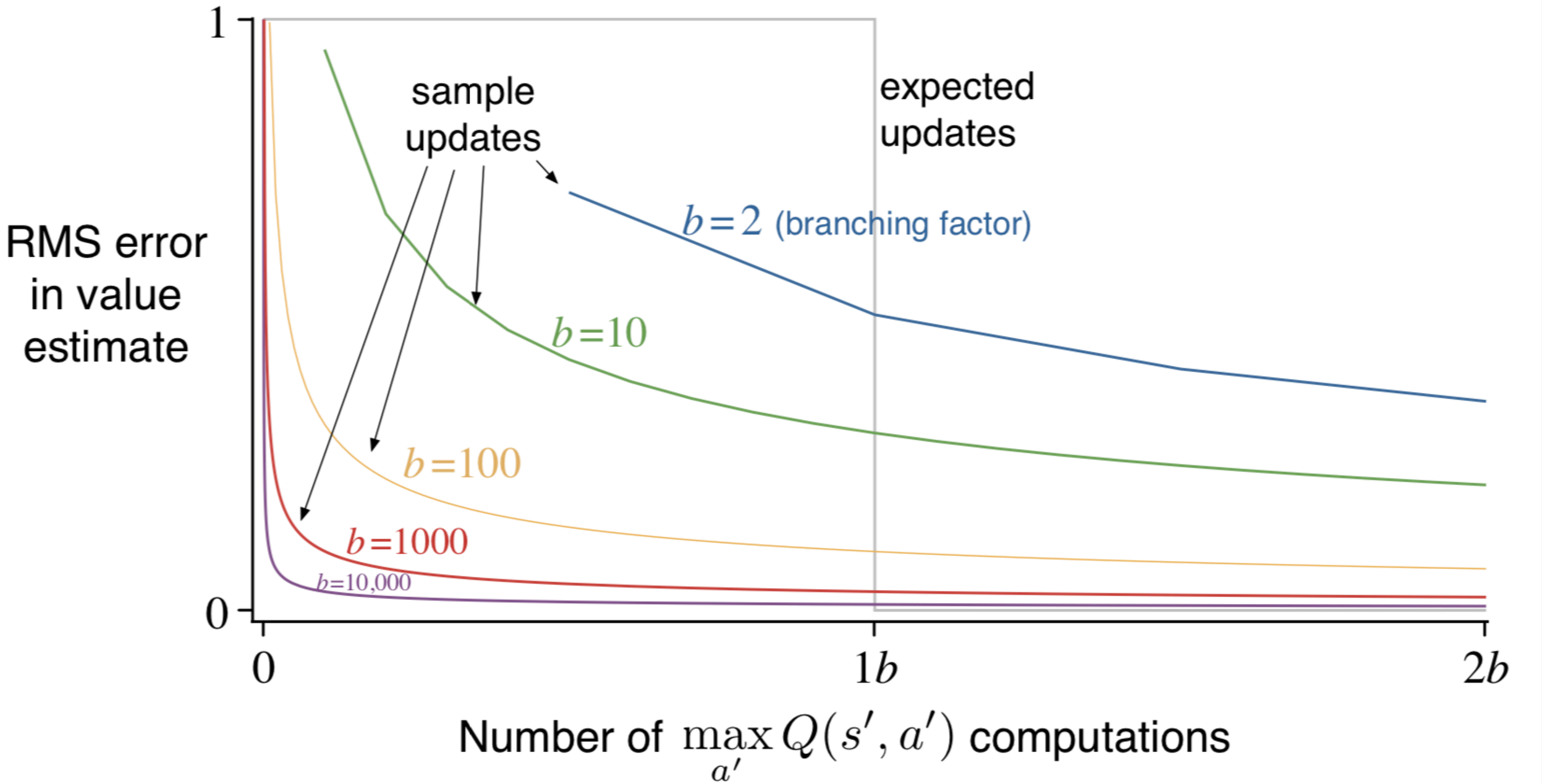
- Update state/action values
- Optimal vs. arbitrary policy
- Expected vs. sample updates



Expected updates > sample updates?



Expected updates > sample updates?



Trajectory Sampling

Sampling:

- Uniform
- According to on-policy distribution

Trajectory Sampling

Sampling:

- Uniform
- According to on-policy distribution

When to plan

- In the background
- At decision time

Assum s' & r are independent

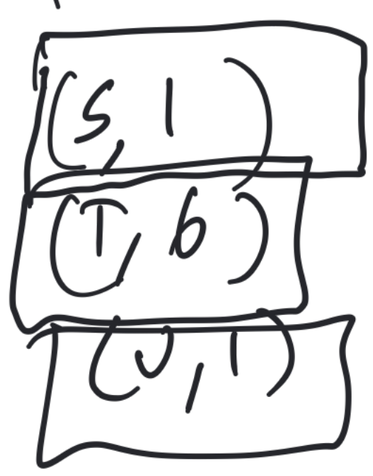
$(s, A) \rightarrow r\ 3$
$\rightarrow r\ 2$
$\rightarrow r\ 5$
$\rightarrow r\ 7$

$r\ -5$
$r\ -5$
$r\ -8$

model
(s, A)
reward ?
4/4
[3, 2, 5, 7]

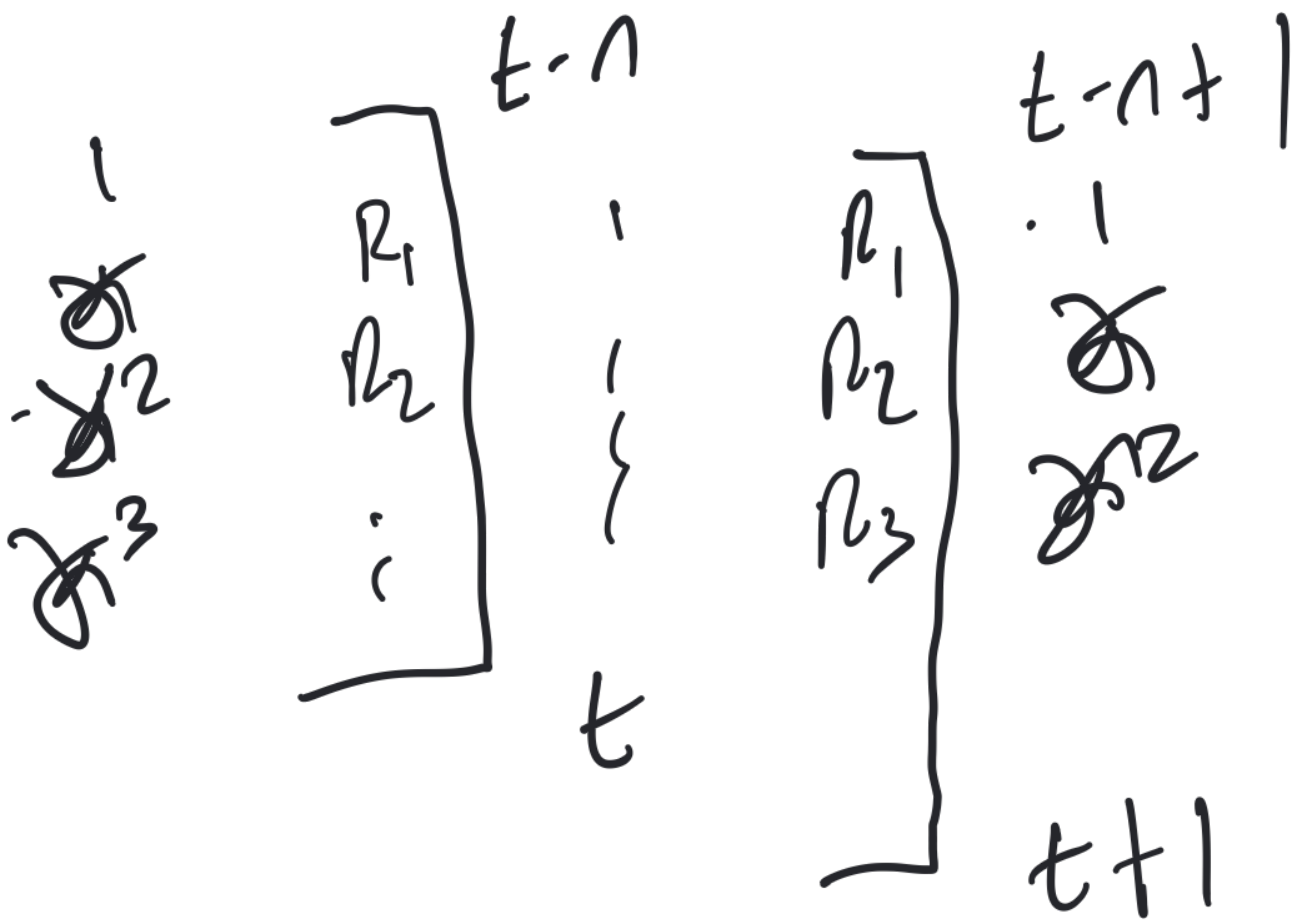
Keep avg reward

$(s, A) \rightarrow s$
T, T, T, T, T
T
U



$\rightarrow$ s' exponential moving avg of reward

(1, 6, 1)
↓ exponential weighted avg
(8, 1, 1)



t-n
t
t-n+1
t+1