

# Markov Decision Processes

Chen-Yu Wei

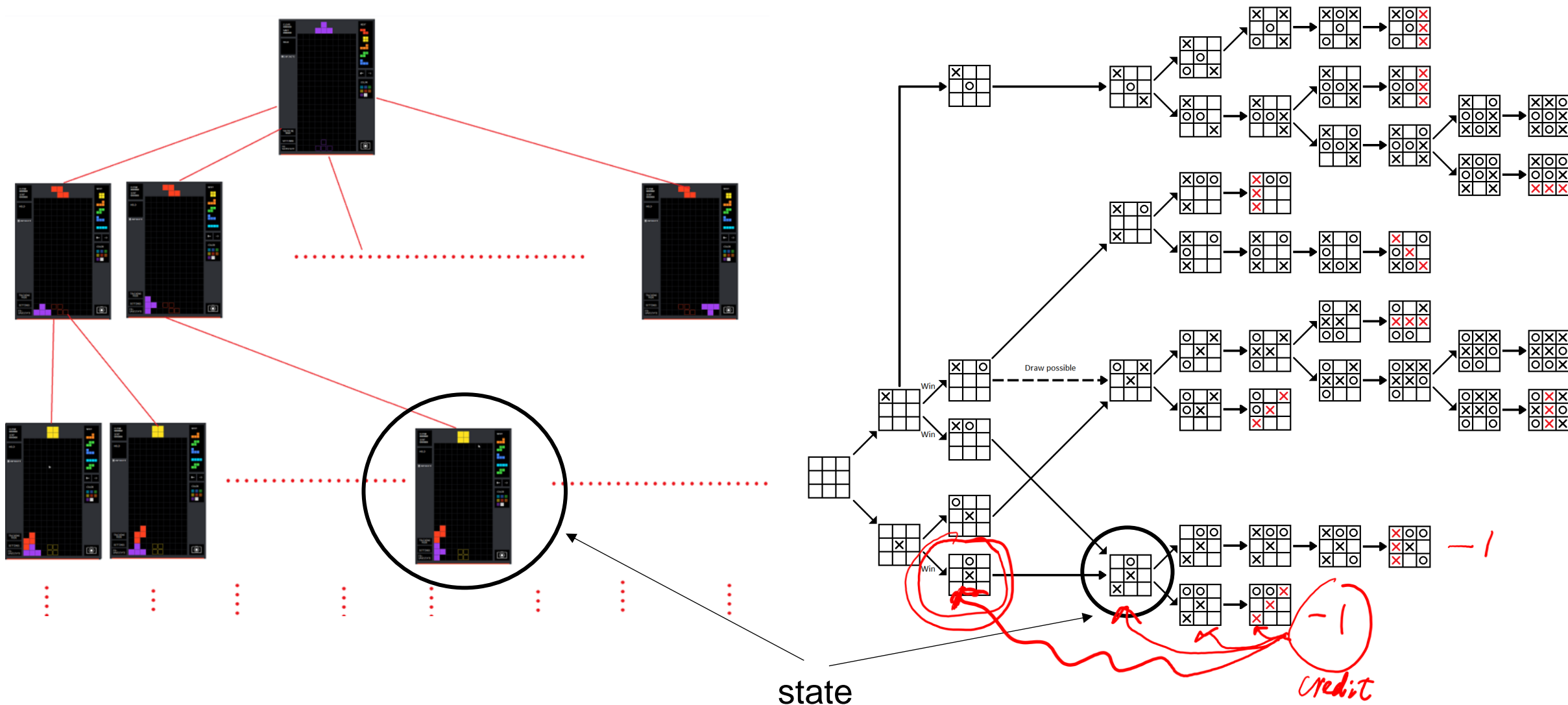
# Sequence of Actions



To win the game, the learner has to take a sequence of actions  $a_1 \rightarrow a_2 \rightarrow \dots \rightarrow a_H$ .  
The effect of a particular action may not be revealed instantaneously.

- Some effect may be revealed instantaneously
- Some may be revealed later

# Sequence of Actions



(a summary of the current status in a multi-stage game)

# Interaction Protocol (Episodic Setting)



For **episode**  $t = 1, 2, \dots, T$ :

$h \leftarrow 1$

Environment generates initial state  $s_{t,1}$

While episode  $t$  has not ended:

Learner chooses an action  $a_{t,h}$

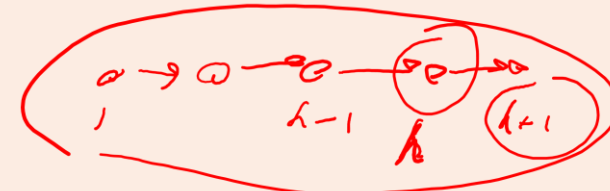
Learner observes instantaneous reward  $r_{t,h}$  with  $\mathbb{E}[r_{t,h}] = R(s_{t,h}, a_{t,h})$

Environment generates next state  $s_{t,h+1} \sim P(\cdot | s_{t,h}, a_{t,h})$

$h \leftarrow h + 1$

**Markov assumption:**

$r_{t,h}$  and  $s_{t,h+1}$  are conditionally independent of  $(s_{t,1}, a_{t,1}, \dots, s_{t,h-1}, a_{t,h-1})$  given  $s_{t,h}$

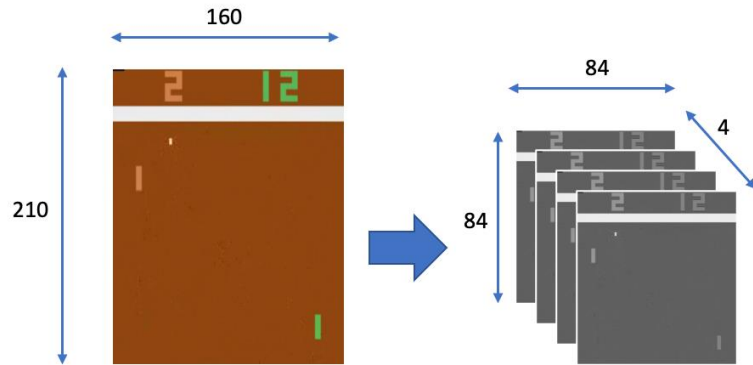


Goal: maximize

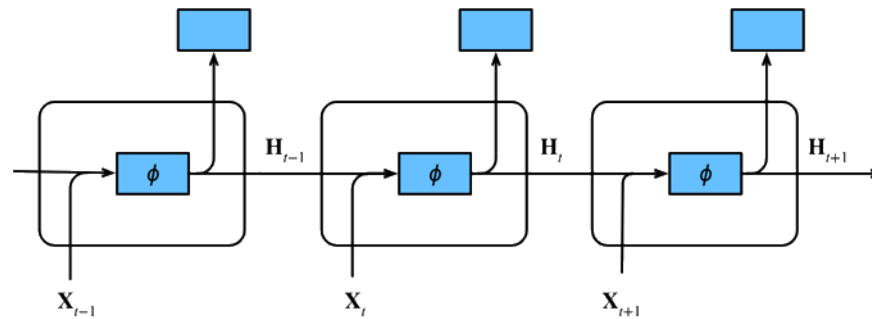
$$\sum_{t=1}^T \sum_{h=1}^{\tau_t} R(s_{t,h}, a_{t,h})$$

$\tau_t$ : length of episode  $t$

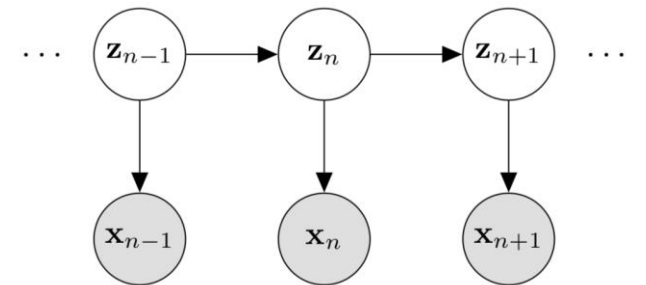
# From Observations to States



Stacking recent observations



Recurrent neural network



Hidden Markov model

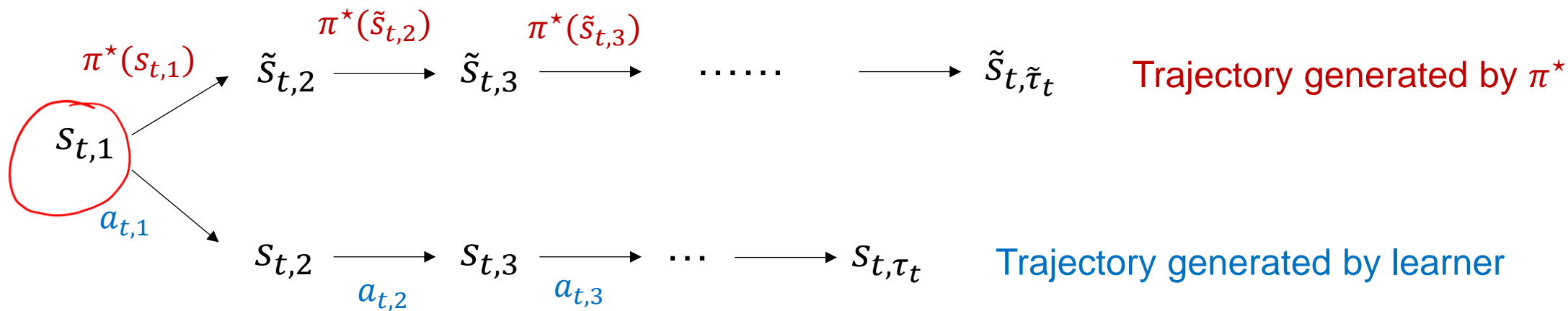
# Regret (Episodic Setting)

$$\pi^*: S \rightarrow A$$

$$\text{Regret} = \underbrace{\max_{\pi^*} \mathbb{E}^{\pi^*} \left[ \sum_{t=1}^T \sum_{h=1}^{\tilde{\tau}_t} R(\tilde{s}_{t,h}, \pi^*(\tilde{s}_{t,h})) \right]}_{\text{Benchmark}} - \sum_{t=1}^T \sum_{h=1}^{\tau_t} R(s_{t,h}, a_{t,h})$$

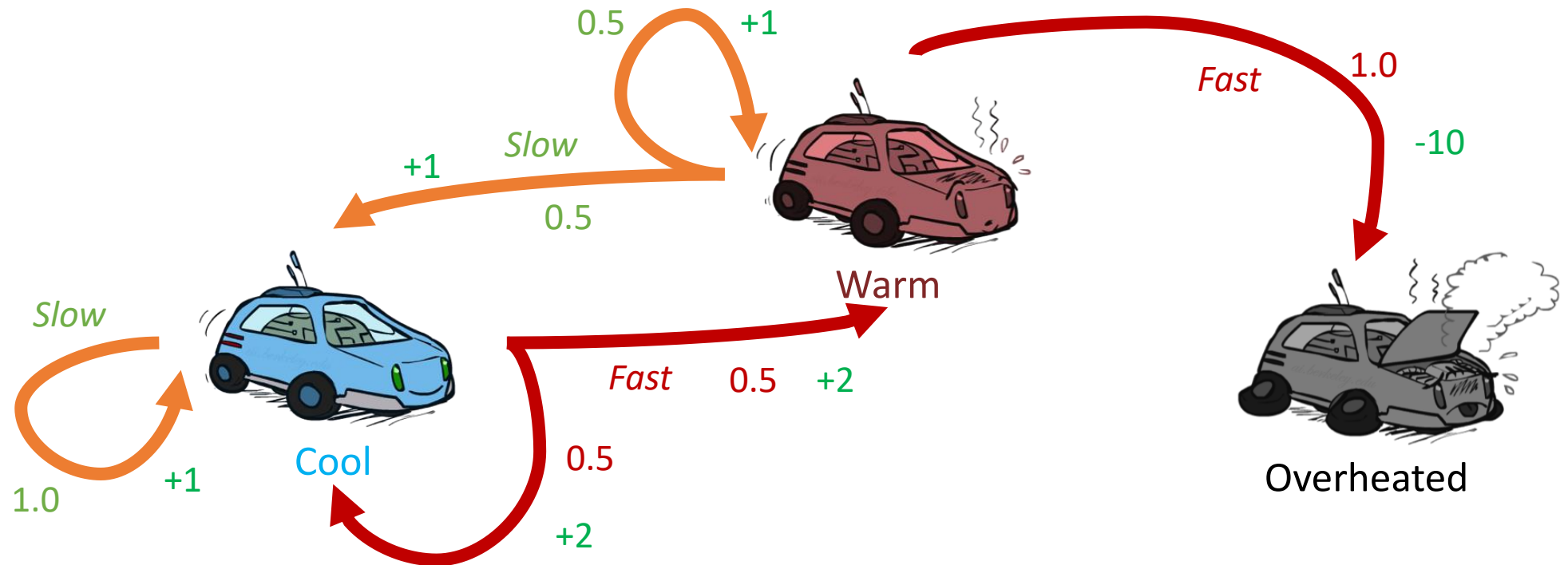
CB

$$\max_{\lambda^*} \sum_{t=1}^T R(x_t, \lambda^*(x_t)) - \sum_{t=1}^T R(x_t, a_t)$$

















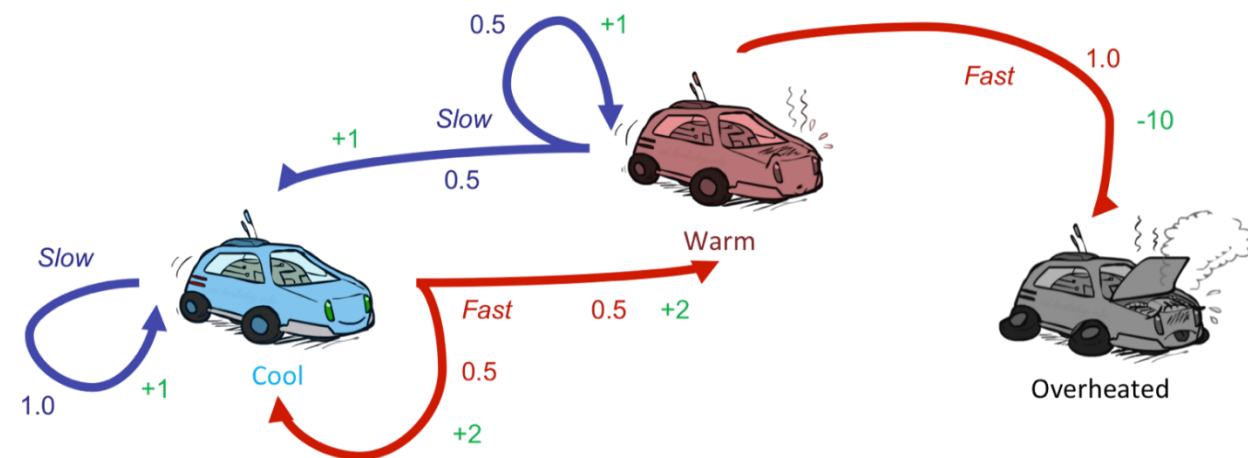
# Example: Racing

- A robot car wants to travel far, quickly
- Three states: **Cool**, **Warm**, Overheated
- Two actions: *Slow*, *Fast*
- Going faster gets double reward



# Example: Racing

$s$	$a$	$s'$	$P(s' s, a)$	$R(s, a)$
	Slow		1.0	+1
	Fast		0.5	+2
	Fast		0.5	+2
	Slow		0.5	+1
	Slow		0.5	+1
	Fast		1.0	-10
	(end)		1.0	0





# Formulations

- Interaction Protocol
  - Fixed-Horizon
  - Variable-Horizon (Goal-Oriented)
  - Infinite-Horizon
- Performance Metric
  - Total Reward
  - Average Reward
  - Discounted Reward
- Policy
  - Markov policy
  - Stationary policy

Horizon = Length of an episode

# Interaction Protocols (1/3): Fixed-Horizon

Horizon length is a fixed number  $H$

$h \leftarrow 1$

Observe initial state  $s_1 \sim \rho$

**While  $h \leq H$ :**

Choose action  $a_h$

Observe reward  $r_h$  with  $\mathbb{E}[r_h] = R(s_h, a_h)$

Observe next state  $s_{h+1} \sim P(\cdot | s_h, a_h)$

**Examples:** games with a fixed number of time

# Interaction Protocols (2/3): Goal-Oriented

The learner interacts with the environment until reaching **terminal states**  $\mathcal{T} \subset \mathcal{S}$

$h \leftarrow 1$

Observe initial state  $s_1 \sim \rho$

**While**  $s_h \notin \mathcal{T}$ :

    Choose action  $a_h$

    Observe reward  $r_h$  with  $\mathbb{E}[r_h] = R(s_h, a_h)$

    Observe next state  $s_{h+1} \sim P(\cdot | s_h, a_h)$

$h \leftarrow h + 1$

**Examples:** video games, robotics tasks, personalized recommendations, etc.

# Interaction Protocols (3/3): Infinite-Horizon

The learner continuously interacts with the environment

~~$h \leftarrow 1$~~

~~Observe initial state  $s_1 \sim \rho$~~

**Loop forever:**

Choose action  $a_h$

Observe reward  $r_h$  with  $\mathbb{E}[r_h] = R(s_h, a_h)$

Observe next state  $s_{h+1} \sim P(\cdot | s_h, a_h)$

$h \leftarrow h + 1$

**Examples:** network management, inventory management

# Formulations

- Interaction Protocol
  - Fixed-Horizon
  - Variable-Horizon (Goal-Oriented)
  - Infinite-Horizon
- Performance Metric
  - Total Reward
  - Average Reward
  - Discounted Reward
- Policy
  - Markov policy
  - Stationary policy

# Performance Metric

**Total Reward** (for episodic setting):

$$\sum_{h=1}^{\tau} r_h$$

( $\tau$ : the step where the episode ends)

**Average Reward** (for infinite-horizon setting):

$$\lim_{H \rightarrow \infty} \frac{1}{H} \sum_{h=1}^H r_h$$

**Discounted Total Reward** (for episodic or infinite-horizon):

$$\sum_{h=1}^{\tau} \gamma^{h-1} r_h$$

$\tau$ : the step where the episode ends, or  $\infty$  in the infinite-horizon case

$\gamma \in [0,1)$ : discount factor

$$\gamma = 0.99$$

# Interaction Protocols vs. Performance Metrics

Fixed-Horizon	“natural” objective ----->	Total Reward	
Goal-Oriented	----->	Total Reward	Could be unbounded
Infinite-horizon	----->	Average Reward	Could have constant change for an infinitesimal change in policy

## Discounted Total Reward?

Focusing more on the **recent** reward

There is a potential mismatch between our ultimate goal and what we optimized.

# Formulations

- Interaction Protocol
  - Fixed-Horizon
  - Variable-Horizon (Goal-Oriented)
  - Infinite-Horizon
- Performance Metric
  - Total Reward
  - Average Reward
  - Discounted Reward
- Policy
  - Markov policy
  - Stationary policy



# Policy for MDPs

$$\pi = (\pi_1, \pi_2, \dots, \pi_H, \dots)$$

$\uparrow$

## Markov Policy

$h$  : step index

$$a_h \sim \pi_h(\cdot | s_h) \in \Delta_A$$
$$a_h = \pi_h(s_h) \in A$$

(space of dist)

For **fixed-horizon** setting, there exists an optimal policy in this class

✓

## Stationary Policy $\subseteq$ Markov Policy

$$a_h \sim \pi(\cdot | s_h)$$
$$a_h = \pi(s_h)$$

For **infinite-horizon/goal-oriented** settings, there exists an optimal policy in this class

✓

△ Fixed-horizon (Markov Policy) (total reward)

✓ Goal-oriented (Stationary Policy) (Discounted reward)

A **stationary policy** specifies

$$\pi(\text{Slow} \mid \text{Cool})$$

$$\pi(\text{Fast} \mid \text{Cool})$$

$$\pi(\text{Slow} \mid \text{Warm})$$

$$\pi(\text{Fast} \mid \text{Warm})$$

A **Markov policy** specifies

$$\pi_h(\text{Slow} \mid \text{Cool})$$

$$\pi_h(\text{Fast} \mid \text{Cool})$$

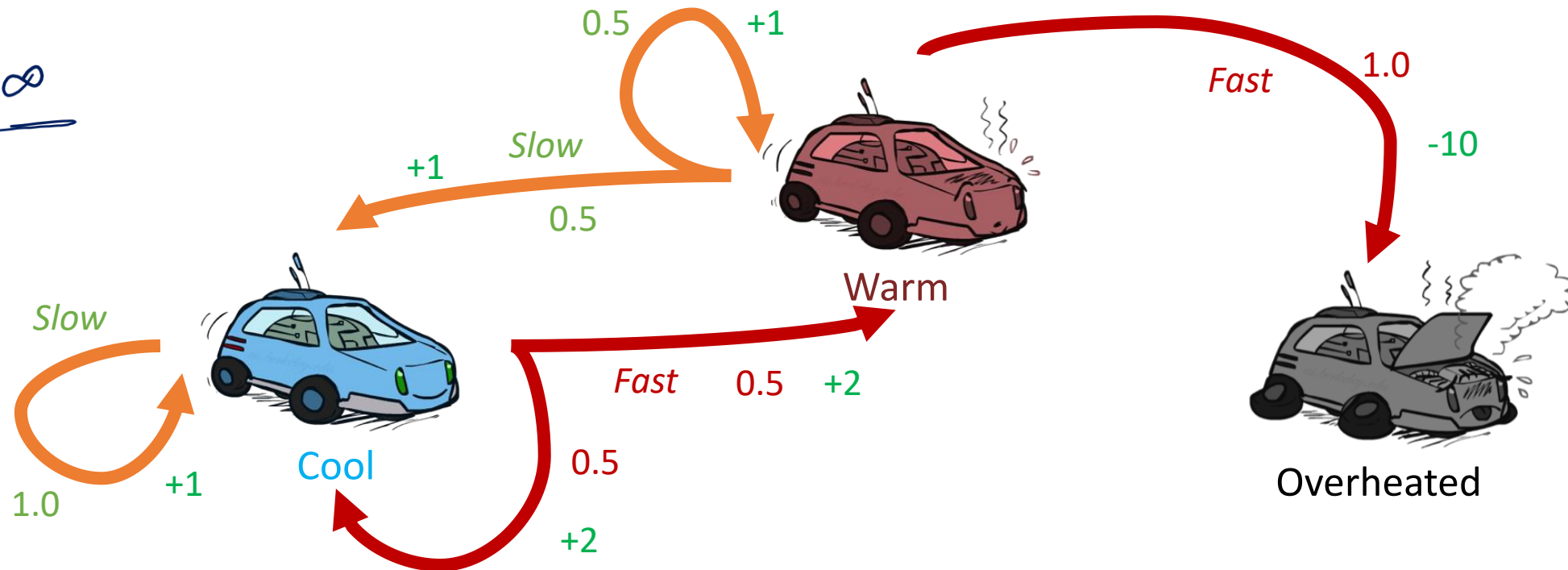
$$\pi_h(\text{Slow} \mid \text{Warm})$$

$$\pi_h(\text{Fast} \mid \text{Warm})$$

$$\forall h$$

$H = 5$

$H = \infty$



# **Value Iteration**

(Fixed-Horizon)

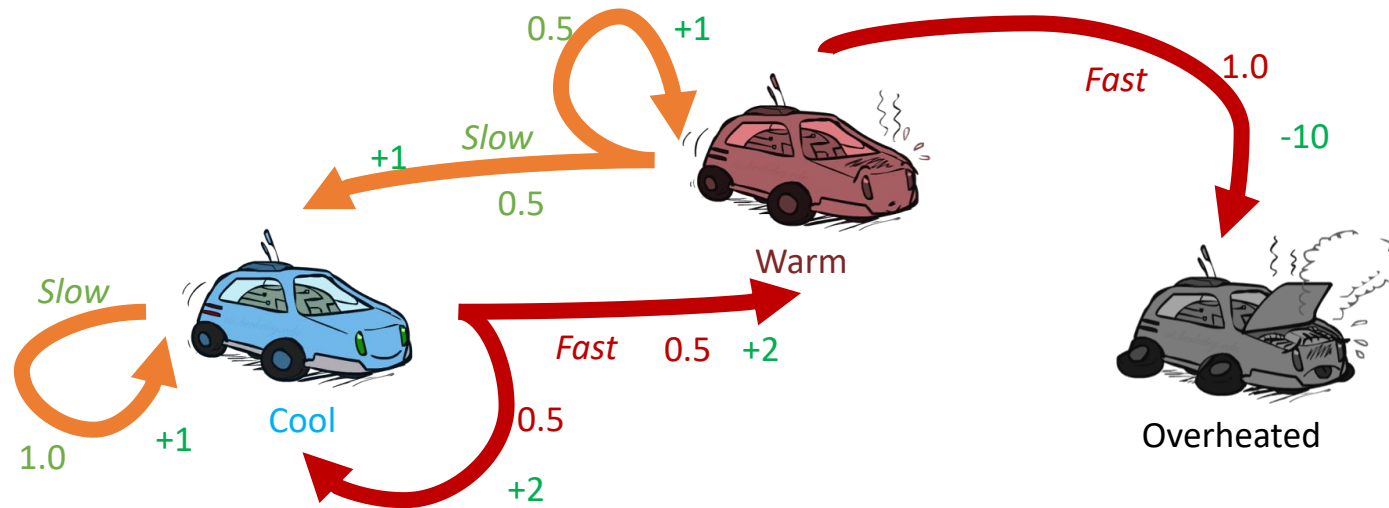
# Two Tasks

**Policy Evaluation:** Calculate the expected total reward of a given policy

What is the expected total reward for the policy  $\pi(\text{cool}) = \text{fast}$ ,  $\pi(\text{warm}) = \text{slow}$ ?

**Policy Optimization:** Find the best policy

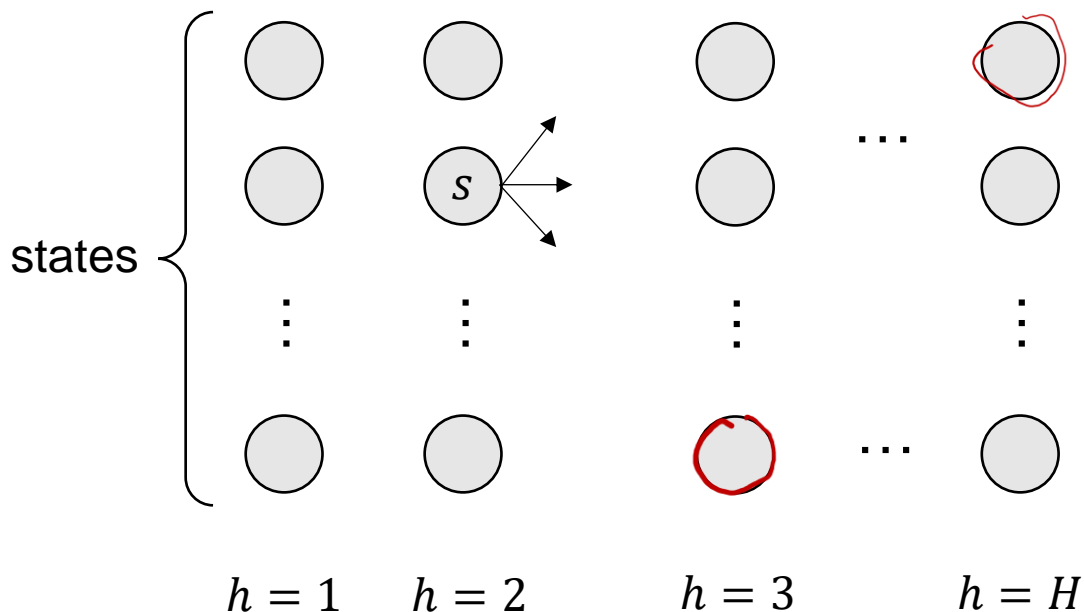
What is the policy that achieves the highest expected total reward?



# Value Iteration for Policy Evaluation

$$\pi = (\pi_1, \dots, \pi_H)$$

$$\mathbb{E}^{\pi} \left[ \sum_{h=1}^H R(s_h, a_h) \right]$$



State transition:  $P(s'|s, a)$

Reward:  $R(s, a)$

$$V_i^{\pi}(s)$$

expected total

$$= \sum_s P(s) V_i^{\pi}(s)$$

$$Q_h^{\pi}(s, a) = \mathbb{E}^{\pi} \left[ \sum_{k=h}^H R(s_k, a_k) \mid (s_h, a_h) = (s, a) \right]$$

$$V_h^{\pi}(s) = \mathbb{E}^{\pi} \left[ \sum_{k=h}^H R(s_k, a_k) \mid s_h = s \right] \quad R(s, a)$$

**Backward induction:**

$$Q_H^{\pi}(s, a) = R(s, a)$$

$$V_{H+1}^{\pi}(s) = 0 \quad \forall s$$

For  $h = H, \dots, 1$ : for all  $s, a$

$$Q_h^{\pi}(s, a) = R(s, a) + \underbrace{\sum_{s'} P(s'|s, a) V_{h+1}^{\pi}(s')}_{\text{Expected total reward of } \pi \text{ from step } h+1}$$

$$V_h^{\pi}(s) = \sum_a \pi_h(a|s) Q_h^{\pi}(s, a)$$

# Bellman Equation

$Q_h^\pi$  is called “the state-action value functions of policy  $\pi$ ”

$V_h^\pi$  is called “the state value function of policy  $\pi$ ”

Both can be just called “**value functions**”

$$Q_h^\pi(s, a) = R(s, a) + \sum_{s'} P(s'|s, a) V_{h+1}^\pi(s')$$

$$V_h^\pi(s) = \sum_a \pi_h(a|s) Q_h^\pi(s, a)$$

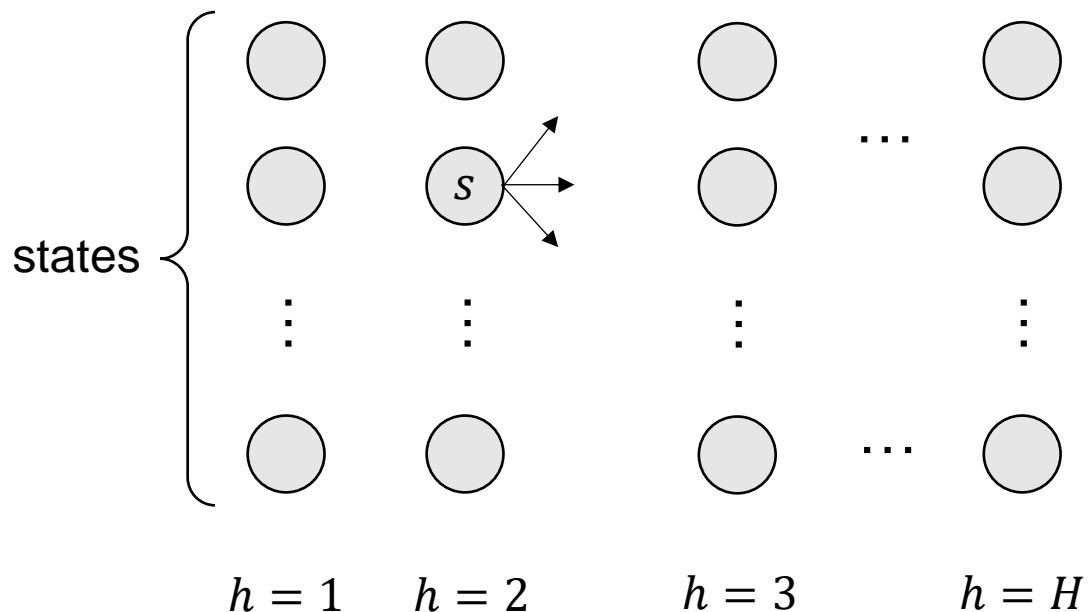
or

$$Q_h^\pi(s, a) = R(s, a) + \sum_{s', a'} P(s'|s, a) \pi_{h+1}(a'|s') Q_{h+1}^\pi(s', a')$$

or

$$V_h^\pi(s) = \sum_a \pi_h(a|s) \left( R(s, a) + \sum_{s'} P(s'|s, a) V_{h+1}^\pi(s') \right)$$

# Value Iteration for Policy Optimization



State transition:  $P(s'|s, a)$

Reward:  $R(s, a)$

$$Q_h^*(s, a) = \max_{\pi \in \Pi_M} \mathbb{E}^\pi \left[ \sum_{k=h}^H R(s_k, a_k) \mid (s_h, a_h) = (s, a) \right]$$

$$V_h^*(s) = \max_{\pi \in \Pi_M} \mathbb{E}^\pi \left[ \sum_{k=h}^H R(s_k, a_k) \mid s_h = s \right]$$

**Backward induction:**

$$V_{H+1}^*(s) = 0 \quad \forall s$$















For  $h = H, \dots, 1$ : for all  $s, a$

$$Q_h^*(s, a) = R(s, a) + \underbrace{\sum_{s'} P(s'|s, a) V_{h+1}^*(s')}_{\text{Expected optimal total reward from step } h+1}$$

Expected optimal total  
reward from step  $h+1$

$$V_h^*(s) = \max_a Q_h^*(s, a) \quad \pi_h^*(s) = \operatorname{argmax}_a Q_h^*(s, a)$$

# Exercise

$s$	$a$	$s'$	$P(s' s, a)$	$R(s, a)$
	Slow		1.0	+1
	Fast		0.5	+2
	Fast		0.5	+2
	Slow		0.5	+1
	Slow		0.5	+1
	Fast		1.0	-10
	(end)		1.0	0

Assume  $H = 3$

$$Q_3^*(s, a) = R(s, a)$$

$$Q_3^*(\text{cool}, \text{slow}) = 1$$

$$Q_3^*(\text{cool}, \text{fast}) = 2$$

$$Q_3^*(\text{warm}, \text{slow}) = 1$$

$$Q_3^*(\text{warm}, \text{fast}) = -10$$

$$V_3^*(s)$$

$$V_3^*(\text{cool}) = 2$$

$$V_3^*(\text{warm}) = 1$$

$$Q_2^*(s, a) = R(s, a) + \sum_{s'} P(s'|s, a) V_3^*(s')$$

$$Q_2^*(\text{cool}, \text{slow}) = 1 + V_3^*(\text{cool}) = 3$$

$$Q_2^*(\text{cool}, \text{fast}) = 2 + 0.5 V_3^*(\text{cool}) + 0.5 V_3^*(\text{warm}) = 3.5$$

$$Q_2^*(\text{warm}, \text{slow}) = 1 + 0.5 V_3^*(\text{cool}) + 0.5 V_3^*(\text{warm}) = 2.5$$

$$Q_2^*(\text{warm}, \text{fast}) = -10$$

$$V_2^*(s)$$

$$V_2^*(\text{cool}) = 3.5$$

$$\pi_2^*(\text{cool}) = \text{fast}$$

$$V_2^*(\text{warm}) = 2.5$$

$$\pi_2^*(\text{warm}) = \text{slow}$$



# Bellman Optimality Equation

$Q_h^*$  : optimal state-action value functions

$V_h^*$  : optimal state value functions  
or “**optimal value functions**”

$$Q_h^*(s, a) = R(s, a) + \sum_{s'} P(s'|s, a) V_{h+1}^*(s')$$

$$V_h^*(s) = \max_a Q_h^*(s, a)$$

or

$$Q_h^*(s, a) = R(s, a) + \sum_{s'} P(s'|s, a) \left( \max_{a'} Q_{h+1}^*(s', a') \right)$$

or

$$V_h^*(s) = \max_a \left( R(s, a) + \sum_{s'} P(s'|s, a) V_{h+1}^*(s') \right)$$

$$\pi_h^*(s) = \operatorname{argmax}_a Q_h^*(s, a)$$

# Recall: Regret

$$\text{Regret} = \max_{\pi^*} \mathbb{E}^{\pi^*} \left[ \sum_{t=1}^T \sum_{h=1}^{\tilde{\tau}_t} R(\tilde{s}_{t,h}, \pi^*(\tilde{s}_{t,h})) \right] - \sum_{t=1}^T \sum_{h=1}^{\tau_t} R(s_{t,h}, a_{t,h})$$

$$\mathbb{E}[\text{Regret}] = \mathbb{E} \left[ \sum_{t=1}^T \left( \underline{V_1^*(s_{t,1})} - \underline{V_1^{\pi_t}(s_{t,1})} \right) \right]$$

$$= \mathbb{E} \left[ \sum_{t=1}^T \left( \underline{V_1^*(\rho)} - \underline{V_1^{\pi_t}(\rho)} \right) \right]$$

$$V_1^\pi(\rho) \triangleq \mathbb{E}_{s \sim \rho} [V_1^\pi(s)]$$

$$\underline{s_{t,1} \sim \rho}$$

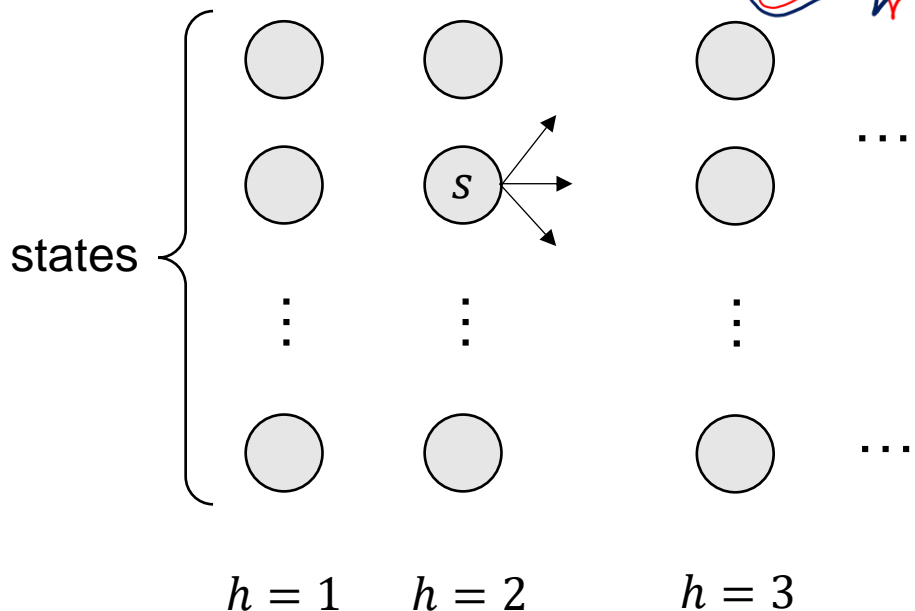
# Value Iteration

(Infinite-Horizon) (or variable-horizon)

# Value Iteration for Policy Evaluation

$$Q_i^z(s,a) = R(s,a) + \mathbb{E}^{\pi} \left[ \gamma \sum_{h=2}^i \gamma^{h-2} R(s_h, a_h) \mid s_2 \sim p(\cdot | s, a) \right]$$

$$\mathbb{E} \left[ \sum_{h=1}^{\infty} \gamma^{h-1} R(s_h, a_h) \right] = \mathbb{E} \left[ \sum_{h=1}^{\infty} \gamma^{h-1} R(s_h, a_h) \right]$$



weight   1    $\gamma$     $\gamma^2$

State transition:  $P(s' | s, a)$

Reward:  $R(s, a)$

$$Q_i^{\pi}(s, a) = \mathbb{E}^{\pi} \left[ \sum_{h=1}^i \gamma^{h-1} R(s_h, a_h) \mid (s_0, a_0) = (s, a) \right]$$

$$V_i^{\pi}(s) = \mathbb{E}^{\pi} \left[ \sum_{h=1}^i \gamma^{h-1} R(s_h, a_h) \mid s_1 = s \right]$$

For fixed horizon  
 $i = (H+1) - h$

$$Q^{\pi}(s, a) = Q_{\infty}^{\pi}(s, a) \quad V^{\pi}(s) = V_{\infty}^{\pi}(s)$$

$$V_0^{\pi}(s) = 0 \quad \forall s$$

$$V_{H+1}^z(s) = 0 \quad \text{fixed horizon}$$

For  $i = 1, 2, 3, \dots$  for all  $s, a$

$$Q_i^{\pi}(s, a) = R(s, a) + \gamma \sum_{s'} P(s' | s, a) V_{i-1}^{\pi}(s')$$

$$V_i^{\pi}(s) = \sum_a \pi(a | s) Q_i^{\pi}(s, a)$$

$$Q_i^{\pi}(s, a) \approx Q_i^z(s, a)$$

If  $|Q_i^{\pi}(s, a) - Q_{i-1}^{\pi}(s, a)| \leq \epsilon$  for all  $s, a$ : **terminate**

$$\left\{ \begin{array}{l} Q^z(s,a) = \mathbb{E}^z \left[ \sum_{h=1}^{\infty} \gamma^{h-1} R(s_h, a_h) \mid \langle s_1, a_1 \rangle = (s, a) \right] \\ V^z(s) = \mathbb{E}^z \left[ \sum_{h=1}^{\infty} \gamma^{h-1} R(s_h, a_h) \mid \underline{s_1 = s} \right] = \sum_a \pi(a|s) \mathbb{E}^z \left[ \sum_{h=1}^{\infty} \gamma^{h-1} R(s_h, a_h) \mid s_1 = s, a_1 = a \right] \end{array} \right.$$

$$Q^z(s,a) = R(s,a) + \mathbb{E}^z \left[ \sum_{h=2}^{\infty} \gamma^{h-1} R(s_h, a_h) \mid s_2 \sim p(\cdot | s, a) \right] \quad \parallel \quad \sum_a \pi(a|s) Q^z(s,a)$$

$$= R(s,a) + \gamma \sum_{s'} p(s'|s,a) \mathbb{E}^z \left[ \sum_{h=2}^{\infty} \gamma^{h-2} R(s_h, a_h) \mid s_2 = s' \right]$$

$$= R(s,a) + \gamma \sum_{s'} p(s'|s,a) \mathbb{E}^z \left[ \sum_{h=1}^{\infty} \gamma^{h-1} R(s_h, a_h) \mid s_1 = s' \right]$$

$V^z(s')$

$$= R(s,a) + \gamma \sum_{s'} p(s'|s,a) V^z(s')$$

# Bellman Equation

$$Q^{\pi}(s,a) = Q_{\infty}^{\pi}(s,a)$$

$$\mathbb{E}_{s \sim p} (V^{\pi}(s))$$

$$\underline{Q}^{\pi}(s,a) = R(s,a) + \gamma \sum_{s'} P(s'|s,a) \underline{V}^{\pi}(s')$$

$$V^{\pi}(s) = \sum_a \pi(a|s) Q^{\pi}(s,a)$$

or

$$Q^{\pi}(s,a) = R(s,a) + \gamma \sum_{s',a'} P(s'|s,a) \pi(a'|s') Q^{\pi}(s',a')$$

or

$$V^{\pi}(s) = \sum_a \pi(a|s) \left( R(s,a) + \gamma \sum_{s'} P(s'|s,a) V^{\pi}(s') \right)$$

# Convergence

$$\star \quad |Q_i^\pi(s, a) - Q_{i-1}^\pi(s, a)| \leq \epsilon \quad \forall s, a \quad (*)$$

1. Value Iteration for policy evaluation will terminate.
2. When it terminates, it holds that

$$|Q_i^\pi(s, a) - Q^\pi(s, a)| \leq \frac{\epsilon}{1 - \gamma} \quad \forall s, a$$

$$\begin{aligned} \underline{Q_i^\pi(s, a)} &= R(s, a) + \gamma \sum_{s', a'} P(s' | s, a) \pi(a' | s') Q_{i-1}^\pi(s', a') \\ &= R(s, a) + \gamma \sum_{s', a'} P(s' | s, a) \pi(a' | s') \underline{Q_i^\pi(s', a')} + \gamma \sum_{s', a'} P(s' | s, a) \pi(a' | s') \left( \overbrace{Q_{i-1}^\pi(s', a') - Q_i^\pi(s', a')}^{\in [-\epsilon, \epsilon]} \right) \end{aligned}$$

If (\*) holds, then the last term can be upper bounded by  $\gamma \cdot \epsilon \leq \epsilon$

$$\Rightarrow |Q_i^\pi(s, a) - (R(s, a) + \gamma \sum_{s', a'} P(s' | s, a) \pi(a' | s') \underline{Q_i^\pi(s', a')})| \leq \epsilon$$

# Convergence (A More General Statement of 2.)

**Value error  $\leq (1 - \gamma)^{-1}$  Bellman Error**

Let  $f: \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$  be **any** function (not necessarily generated by Value Iteration)

If

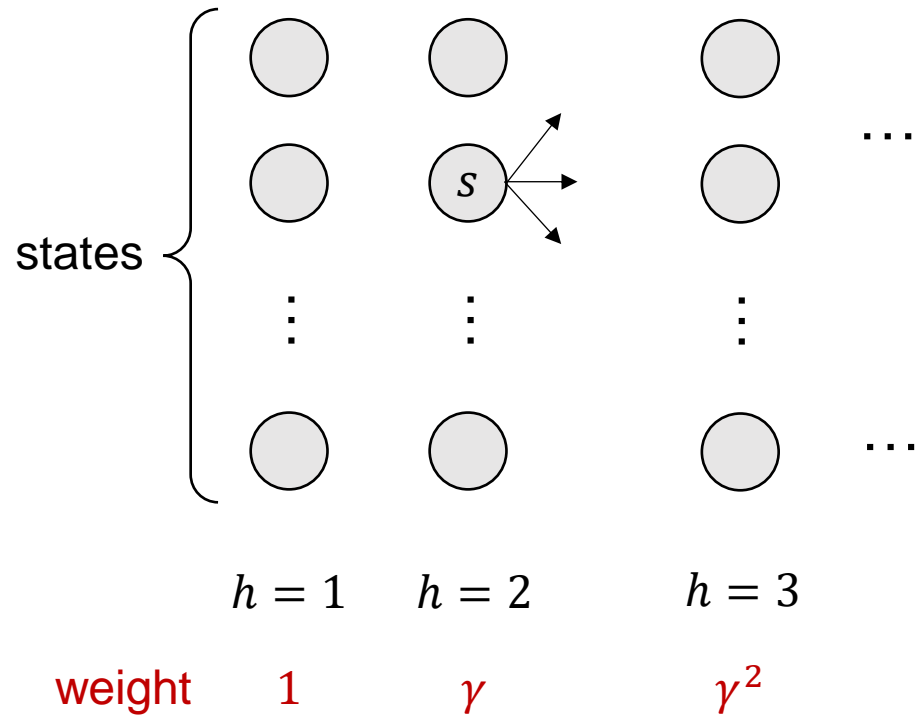
$$\left| f(s, a) - \left( R(s, a) + \gamma \sum_{s', a'} P(s' | s, a) \pi(a' | s') f(s', a') \right) \right| \leq \epsilon$$

for all  $s, a$ , then

$$\boxed{|f(s, a) - Q^\pi(s, a)|} \leq \frac{\epsilon}{1 - \gamma} \quad \leftarrow \quad \underline{f = Q_i^\pi}$$



# Value Iteration for Policy Optimization



State transition:  $P(s'|s, a)$

Reward:  $R(s, a)$

$$Q_i^*(s, a) = \max_{\pi} \mathbb{E}^{\pi} \left[ \sum_{h=1}^i \gamma^{h-1} R(s_h, a_h) \mid (s_0, a_0) = (s, a) \right]$$

$$V_i^*(s) = \max_{\pi} \mathbb{E}^{\pi} \left[ \sum_{h=1}^i \gamma^{h-1} R(s_h, a_h) \mid s_0 = s \right]$$

$$Q^*(s, a) = Q_{\infty}^*(s, a) \quad V^*(s) = V_{\infty}^*(s)$$

$$V_0^*(s) = 0 \quad \forall s$$

For  $i = 1, 2, 3, \dots$ : for all  $s, a$

$$Q_i^*(s, a) = R(s, a) + \gamma \sum_{s'} P(s'|s, a) V_{i-1}^*(s')$$

$$V_i^*(s) = \max_a Q_i^*(s, a)$$

If  $|Q_i^*(s, a) - Q_{i-1}^*(s, a)| \leq \epsilon$  for all  $s, a$ : **terminate**

# Bellman Optimality Equation

$$\pi^*(s) = \operatorname{argmax}_a Q^*(s, a)$$

$$Q^*(s, a) = R(s, a) + \gamma \sum_{s'} P(s'|s, a) V^*(s')$$

$$V^*(s) = \max_a Q^*(s, a)$$

or

$$Q^*(s, a) = R(s, a) + \gamma \sum_{s'} P(s'|s, a) \max_{a'} Q^*(s', a')$$

or

$$V^*(s) = \max_a \left( R(s, a) + \gamma \sum_{s'} P(s'|s, a) V^*(s') \right)$$

# Convergence

1. Value Iteration for policy optimization will terminate.

2. When it terminates, it holds that

$$|Q_i^*(s, a) - Q^*(s, a)| \leq \frac{\epsilon}{1 - \gamma} \quad \forall s, a$$

3. When it terminates, it holds that

$$V^*(s) - V^{\hat{\pi}}(s) \leq \frac{2\epsilon}{(1 - \gamma)^2} \quad \forall s$$

where  $\hat{\pi}(s) = \operatorname{argmax}_a Q_i^*(s, a)$

# Convergence (A More General Statement of 2.)

**Value error  $\leq (1 - \gamma)^{-1}$  Bellman Error**

Let  $f: \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$  be **any** function (not necessarily generated by Value Iteration)

If

$$\left| f(s, a) - \left( R(s, a) + \gamma \sum_{s'} P(s'|s, a) \max_{a'} f(s', a') \right) \right| \leq \epsilon \quad \forall s, a$$

then

$$|f(s, a) - Q^*(s, a)| \leq \frac{\epsilon}{1 - \gamma} \quad \forall s, a$$

# Convergence (A More General Statement of 3.)

**Suboptimality  $\leq (1 - \gamma)^{-1}$  Value Error**

Let  $f: \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$  be **any** function (not necessarily generated by Value Iteration)

If

$$|f(s, a) - Q^*(s, a)| \leq \epsilon \quad \forall s, a$$

then

$$V^*(s) - V^{\pi_f}(s) \leq \frac{2\epsilon}{1 - \gamma} \quad \forall s$$

where  $\pi_f(s) = \operatorname{argmax}_a f(s, a)$

# **Policy Iteration**

# Policy Iteration

## Policy Iteration

For  $i = 1, 2, \dots$

$$\forall s, \quad \pi_i(s) \leftarrow \operatorname{argmax}_a Q^{\pi_i}(s, a)$$

**Theorem (monotonic improvement).** Policy Iteration ensures

$$\forall s, a, \quad Q^{\pi_{i+1}}(s, a) \geq Q^{\pi_i}(s, a)$$

(We will prove this later.)

# Generalized Policy Iteration

$N = \infty \Rightarrow$  Policy Iteration

$N = 1 \Rightarrow$  Value Iteration for policy optimization

For  $i = 1, 2, \dots$

$$\pi_i(s) = \max_a Q_i(s, a) \quad \leftarrow \text{Policy update}$$

$$Q \leftarrow Q_i$$

Repeat for  $N$  times:

$$Q(s, a) \leftarrow R(s, a) + \gamma \sum_{s', a'} P(s' | s, a) \pi_i(a' | s') Q(s', a')$$

$\leftarrow$  Value update

$$Q_{i+1} \leftarrow Q$$

**Notice:** in value iteration for PO, there may not exist a policy  $\pi$  such that  $Q_i = Q^\pi$

In contrast, in policy iteration we have  $Q_i = Q^{\pi_{i-1}}$

VI for PO can be viewed as PI **with incomplete policy evaluation**



# Summary

- Value Iteration for Policy Optimization (VI for PO)
  - Is essentially a **dynamic programming** algorithm
  - Finds the value functions of the optimal policy
- Value Iteration for Policy Evaluation (VI for PE)
  - Also a **dynamic programming** algorithm
  - Finds the value functions of the given policy
- Policy Iteration (PI)
  - An iterative policy improvement algorithm
  - Each iteration involves a policy evaluation subtask
- VI for PO and PI can be viewed as special cases of Generalized PI

# **Performance Difference Lemma**

# Unanswered Questions

- For an estimation  $\hat{Q}(s, a) \approx Q^*(s, a)$  with error, how can we bound

$$V^*(\rho) - V^{\hat{\pi}}(\rho) \quad \text{where } \hat{\pi}(s) = \max_a \hat{Q}(s, a)?$$

- How to show that Policy Iteration leads to monotonic policy improvement?
- Also, how are these methods related to the third challenge of online RL: credit assignment?

# Performance Difference Lemma

For any two stationary policies  $\pi'$  and  $\pi$  in the discounted setting,

$$\begin{aligned}\mathbb{E}_{s \sim \rho} [V^{\pi'}(s)] - \mathbb{E}_{s \sim \rho} [V^{\pi}(s)] &= \sum_{s,a} d_{\rho}^{\pi'}(s) (\pi'(a|s) - \pi(a|s)) Q^{\pi}(s, a) \\ &= \sum_s d_{\rho}^{\pi'}(s, a) (Q^{\pi}(s, a) - V^{\pi}(s))\end{aligned}$$

$$d_{\rho}^{\pi}(s) \triangleq \mathbb{E}^{\pi} \left[ \sum_{h=1}^{\infty} \gamma^{h-1} \mathbb{I}\{s_h = s\} \mid s_1 \sim \rho \right] \quad \text{Discounted frequency of visitation to state } s$$

$$d_{\rho}^{\pi}(s, a) \triangleq \mathbb{E}^{\pi} \left[ \sum_{h=1}^{\infty} \gamma^{h-1} \mathbb{I}\{(s_h, a_h) = (s, a)\} \mid s_1 \sim \rho \right]$$

# Performance Difference Lemma (Fixed-Horizon)

For any two Markov policies  $\pi'$  and  $\pi$  in the fixed-horizon setting,

$$\begin{aligned}\mathbb{E}_{s_1 \sim \rho} [V_1^{\pi'}(s_1)] - \mathbb{E}_{s_1 \sim \rho} [V_1^{\pi}(s_1)] &= \sum_{h=1}^H \sum_{s,a} d_{\rho,h}^{\pi'}(s) (\pi'_h(a|s) - \pi_h(a|s)) Q_h^{\pi}(s,a) \\ &= \sum_{h=1}^H \sum_{s,a} d_{\rho,h}^{\pi'}(s,a) (Q_h^{\pi}(s,a) - V_h^{\pi}(s))\end{aligned}$$

$$d_{\rho,h}^{\pi}(s) \triangleq \mathbb{E}^{\pi}[\mathbb{I}\{s_h = s\} \mid s_1 \sim \rho] = \mathbb{P}^{\pi}(s_h = s \mid s_1 \sim \rho)$$

$$d_{\rho,h}^{\pi}(s,a) \triangleq \mathbb{E}^{\pi}[\mathbb{I}\{(s_h, a_h) = (s,a)\} \mid s_1 \sim \rho] = \mathbb{P}^{\pi}((s_h, a_h) = (s,a) \mid s_1 \sim \rho)$$