Reinforcement Learning Basics

Shusen Wang

Stevens Institute of Technology

A little bit probability theory…

Random Variable

- Random variable: unknown; its values depend on outcomes of random events.

Random Variable | *Possible Values* | *Random Events* | *Probabilities*

$$X = \begin{cases} 0 \\ 1 \end{cases}$$

0 ←  $\mathbb{P}(X = 0) = 0.5$

1 ←  $\mathbb{P}(X = 1) = 0.5$

Random Variable

- Random variable: unknown; its values depend on outcomes of random events.
- Uppercase letter X for a random variable.
- Lowercase letter x for an observed value.
- For example, I flipped a coin 4 times and observed:
  - $x_1 = 1,$
  - $x_2 = 1,$
  - $x_3 = 0,$
  - $x_4 = 1.$

Probability Density Function (PDF)

- PDF provides a relative likelihood that the value of the random variable would equal that sample.

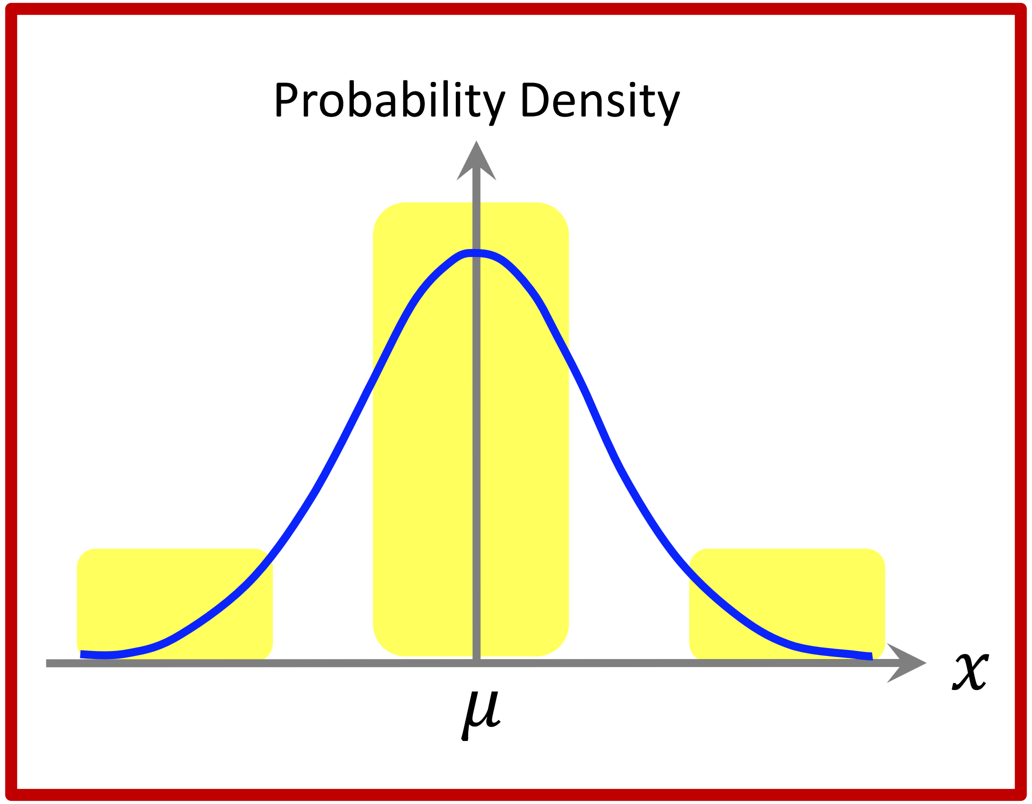
Probability Density Function (PDF)

- PDF provides a relative likelihood that the value of the random variable would equal that sample.

Example: Gaussian distribution

- It is a continuous distribution.
- PDF:

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\,\sigma^2}\right).$$



Probability Mass Function (PMF)

- PMF is a function that gives the probability that a discrete random variable is exactly equal to some value.

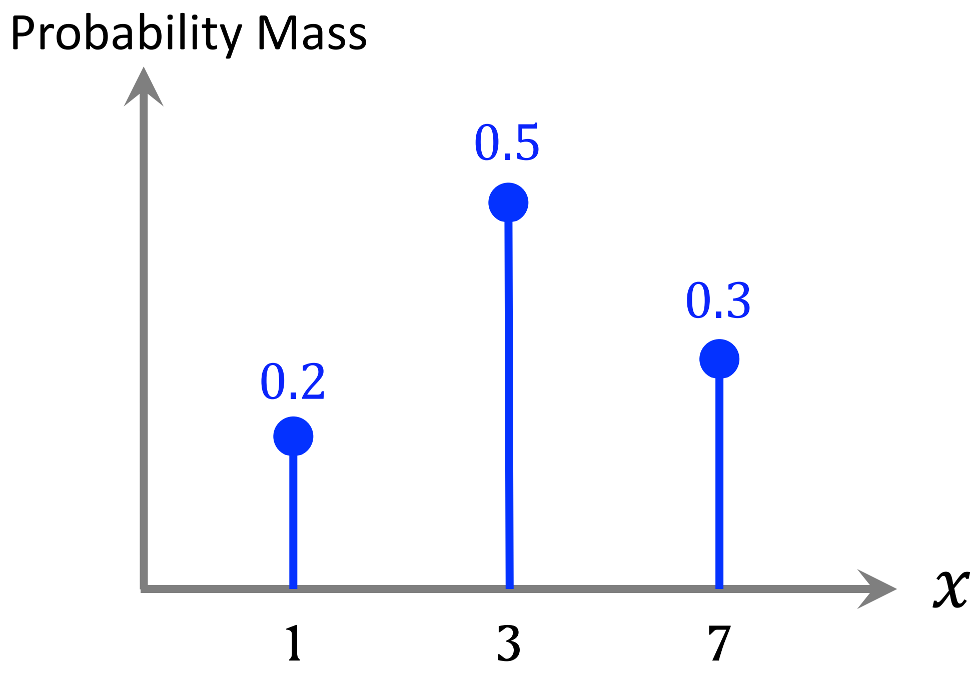
Probability Mass Function (PMF)

- PMF is a function that gives the probability that a discrete random variable is exactly equal to some value.

Example

- Discrete random variable: $\boxed{X \in \{1, 3, 7\}.}$
- PDF:

$$\boxed{\begin{aligned} p(1) &= 0.2, \\ p(3) &= 0.5, \\ p(7) &= 0.3. \end{aligned}}$$



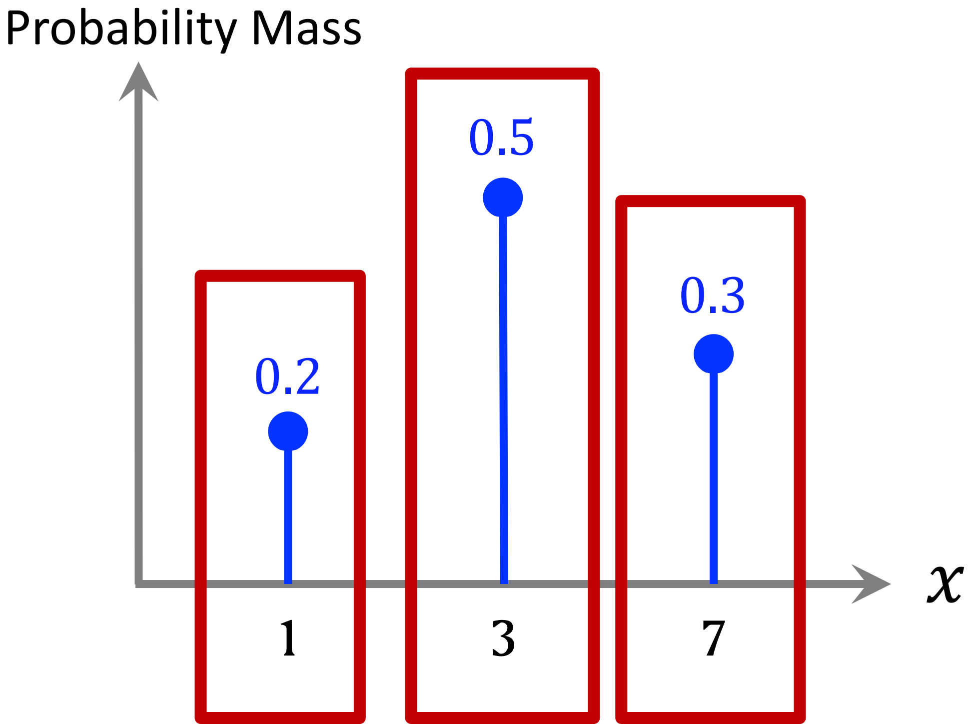
Probability Mass Function (PMF)

- PMF is a function that gives the probability that a discrete random variable is exactly equal to some value.

Example

- Discrete random variable: $X \in \{1, 3, 7\}$.
- PDF:

$$p(1) = 0.2,$$
$$p(3) = 0.5,$$
$$p(7) = 0.3.$$



Properties of PDF/PMF

- Random variable X is in the domain $\mathcal{X}$.
- For continuous distributions,

$$\int_{\mathcal{X}} p(x)\, dx = 1.$$

- For discrete distributions,

$$\sum_{x \in \mathcal{X}} p(x) = 1.$$

Expectation

- Random variable X is in the domain $\mathcal{X}$.
- For continuous distributions, the expectation of $f(X)$ is:

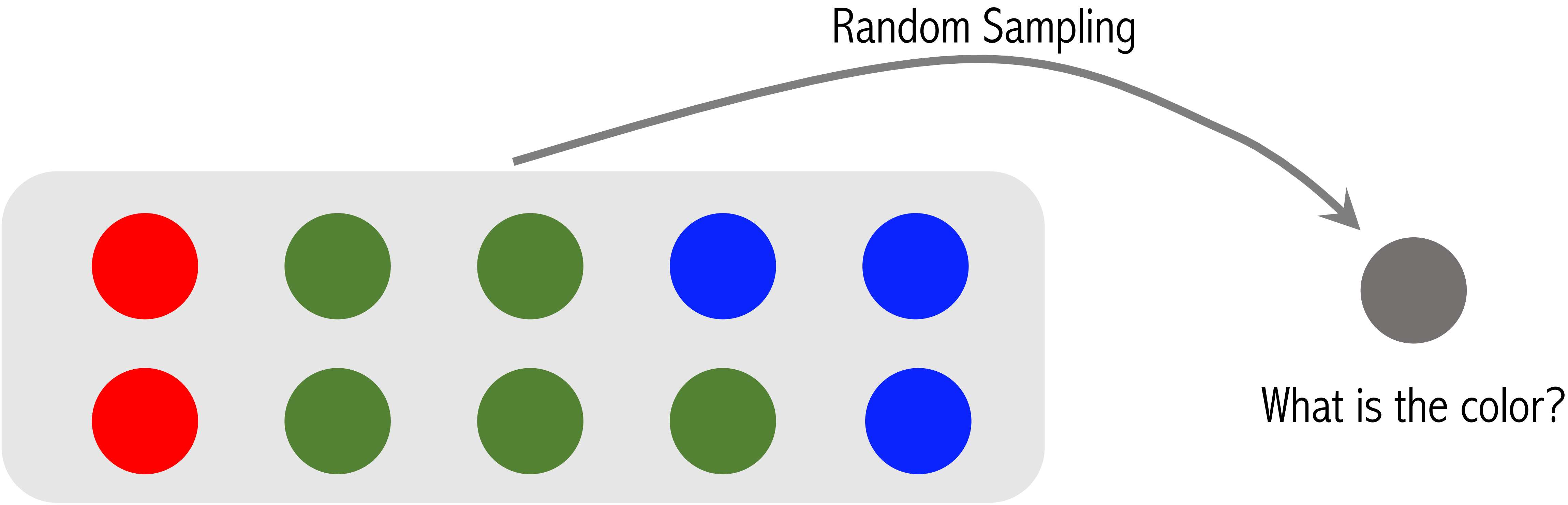
$$\underline{\mathbb{E}[f(X)]} = \underline{\int_{\mathcal{X}} p(x) \cdot f(x)\, dx}.$$

- For discrete distributions, the expectation of $f(X)$ is:

$$\mathbb{E}[f(X)] = \underline{\sum_{x \in \mathcal{X}} p(x) \cdot f(x)}.$$

Random Sampling

- There are 10 balls in the bin: 2 are red, 5 are green, and 3 are blue.



Random Sampling

- Sample red ball w.p. 0.2, green ball w.p. 0.5, and blue ball w.p. 0.3.

Random Sampling

- Sample red ball w.p. 0.2, green ball w.p. 0.5, and blue ball w.p. 0.3.

```
from numpy.random import choice

samples = choice(['R', 'G', 'B'], size=100, p=[0.2, 0.5, 0.3])
print(samples)
```

```
['R' 'G' 'R' 'R' 'R' 'R' 'B' 'B' 'B' 'G' 'G' 'B' 'G' 'B' 'B' 'G' 'B' 'G'
 'B' 'B' 'G' 'B' 'G' 'B' 'B' 'G' 'B' 'B' 'G' 'B' 'G' 'G' 'G' 'G' 'G' 'B'
 'B' 'B' 'B' 'B' 'B' 'G' 'G' 'B' 'R' 'R' 'B' 'R' 'B' 'G' 'R' 'G' 'R' 'G'
 'R' 'R' 'B' 'G' 'G' 'G' 'B' 'R' 'G' 'B' 'G' 'R' 'G' 'G' 'G' 'B' 'B' 'R'
 'G' 'G' 'B' 'B' 'R' 'B' 'B' 'B' 'R' 'B' 'G' 'B' 'R' 'B' 'R' 'G' 'B' 'R'
 'B' 'B' 'G' 'G' 'G' 'R' 'R' 'B' 'R' 'G']
```

Terminologies

Terminology: state and action

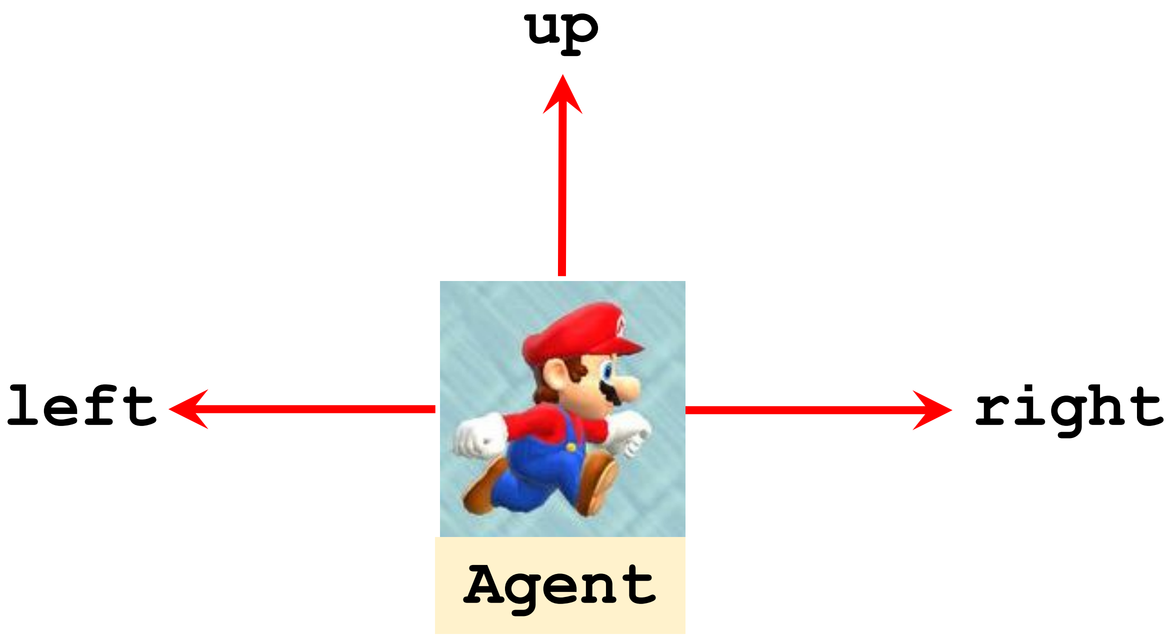
state s (this frame)

Action $a \in \{\text{left}, \text{right}, \text{up}\}$

Terminology: state and action

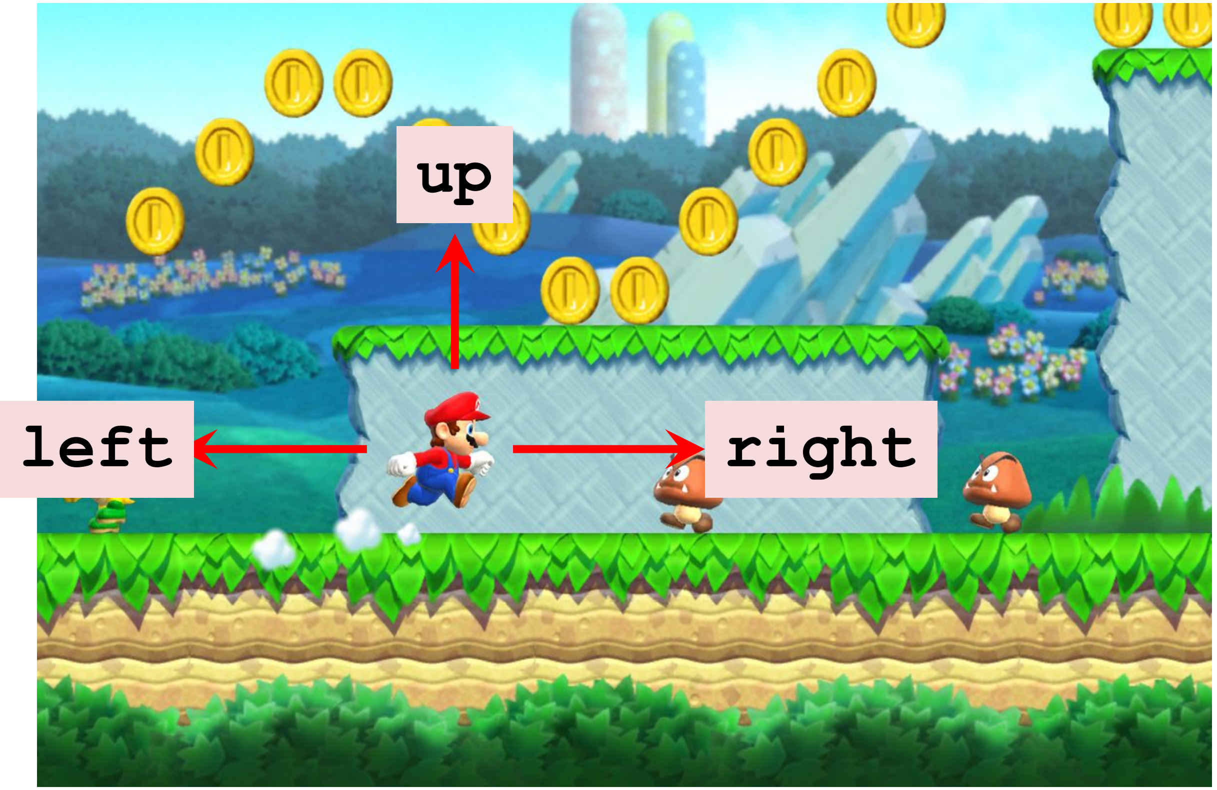
state s (this frame)

Action $a \in$ {left, right, up}



Terminology: policy

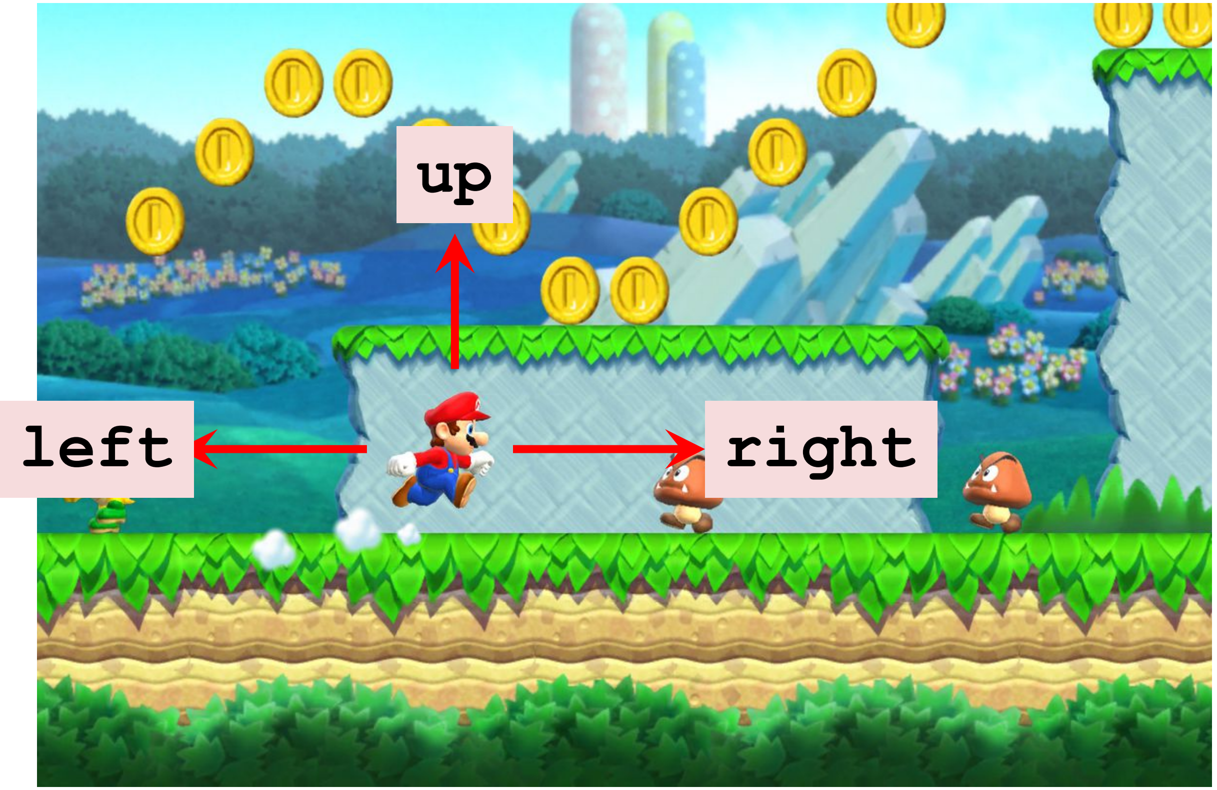
policy π



- Policy function $\pi: (s, a) \mapsto [0,1]$:

$$\pi(a \mid s) = \mathbb{P}(A = a \mid S = s).$$

Terminology: policy



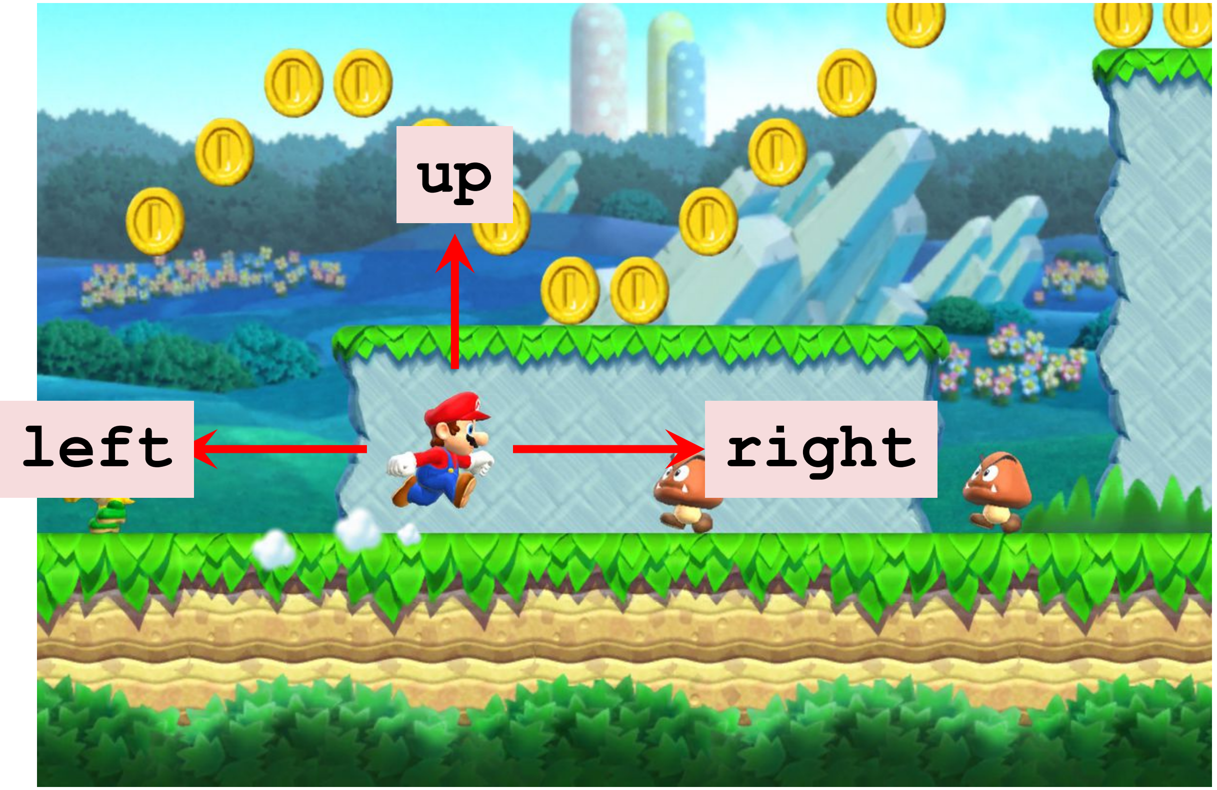
policy π

- Policy function $\pi: (s, a) \mapsto [0,1]$:

$$\pi(a \mid s) = \mathbb{P}(A = a \mid S = s).$$

Terminology: policy

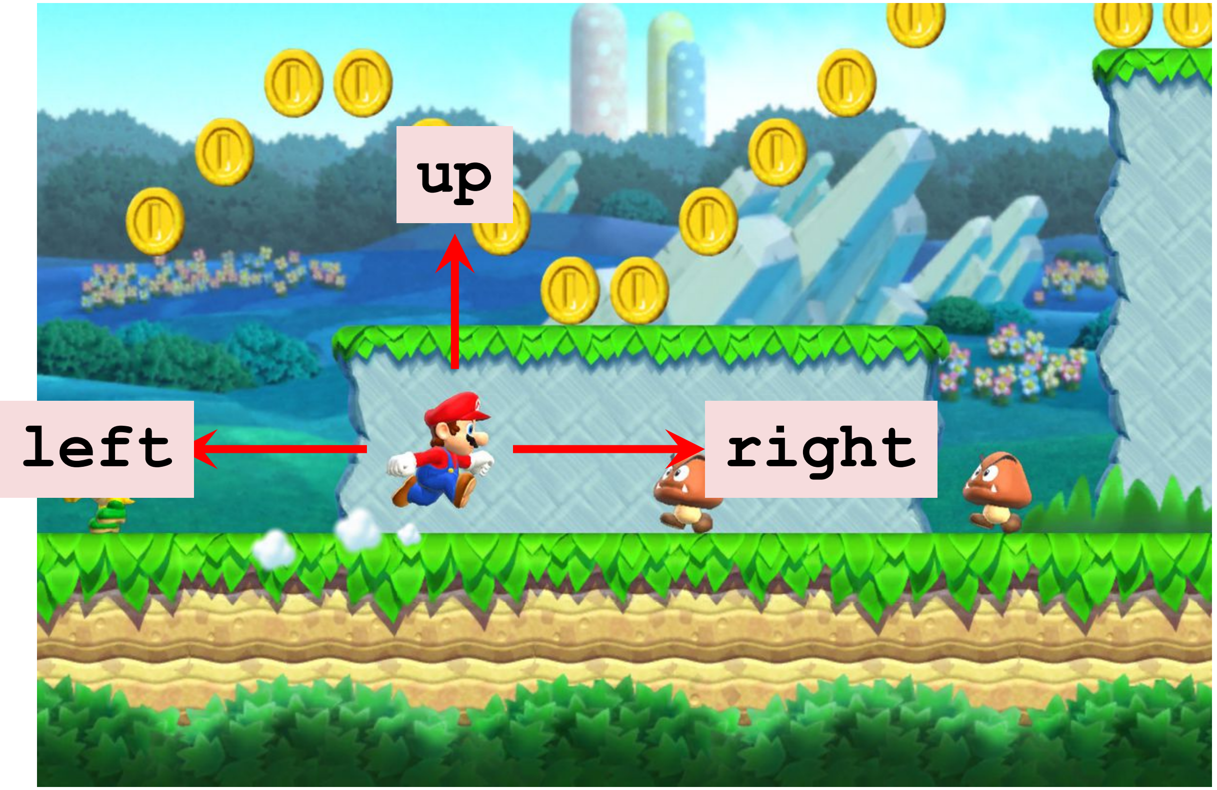
policy π



- Policy function $\pi: (s, a) \mapsto [0,1]$:

$$\pi(a \mid s) \; = \; \mathbb{P}(A = a \mid S = s).$$

Terminology: policy

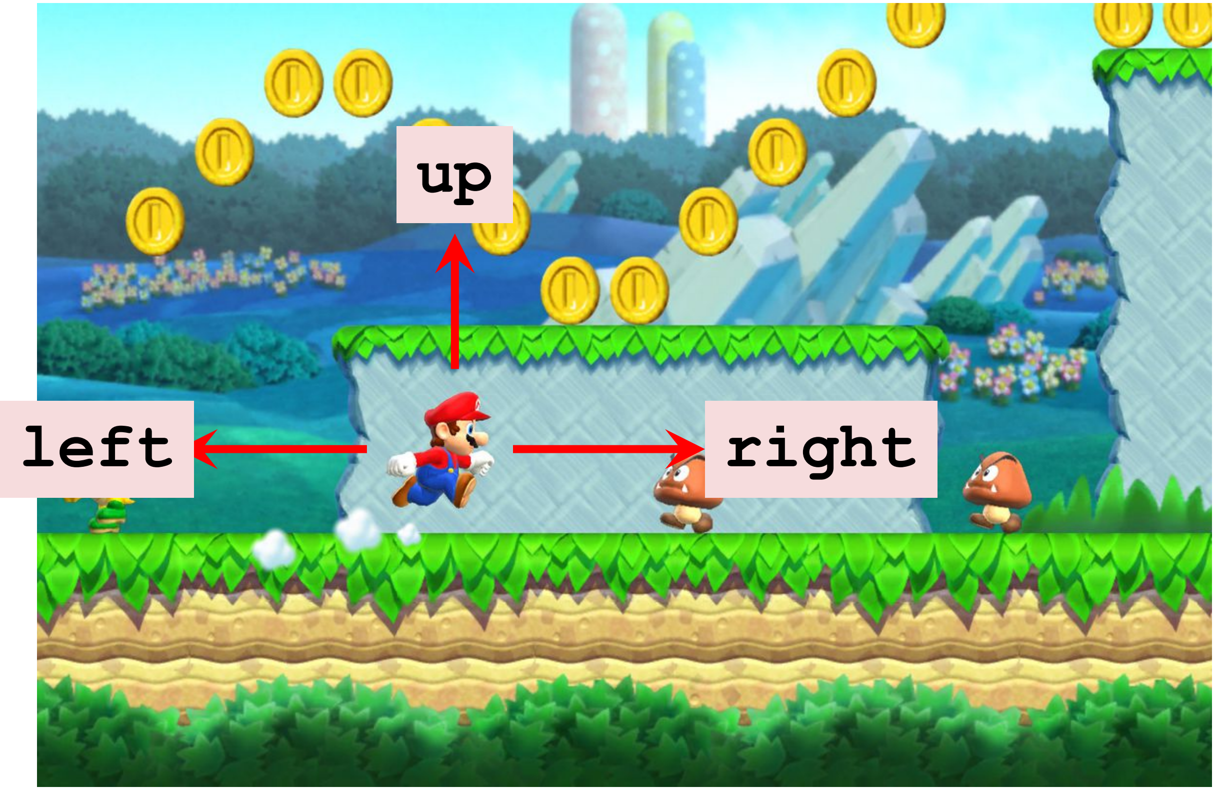


policy π

- $\pi(a \mid s)$ is the probability of taking action $A = a$ given state s , e.g.,
  - $\pi(\text{left} \mid s) = 0.2,$
  - $\pi(\text{right} \mid s) = 0.1,$
  - $\pi(\text{up} \mid s) = 0.7.$

Terminology: policy

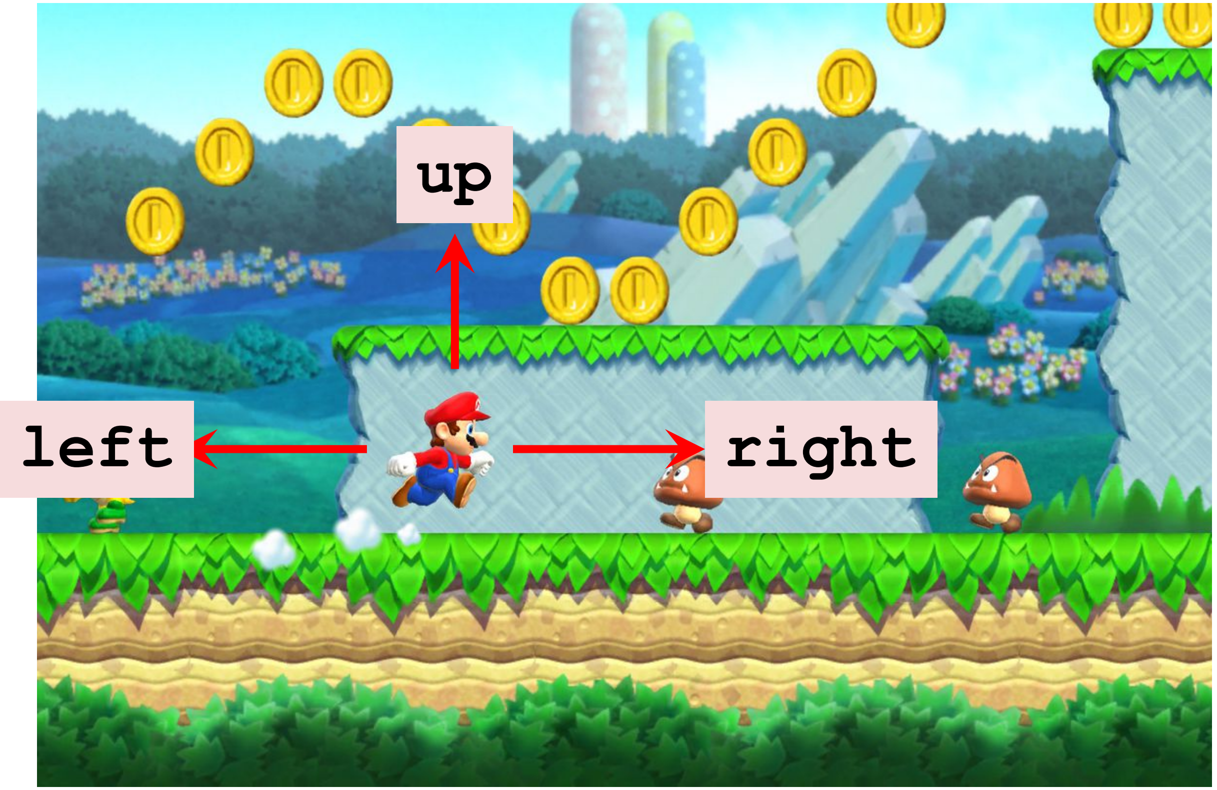
policy π



- $\pi(a \mid s)$ is the probability of taking action $A = a$ given state s , e.g.,
  - $\pi(\text{left} \mid s) = 0.2$,
  - $\pi(\text{right} \mid s) = 0.1$,
  - $\pi(\text{up} \mid s) = 0.7$.
- Upon observing state $S = s$, the agent's action A can be random.

Terminology: policy

Random or deterministic policy?



Terminology: reward

reward R

- Collect a coin: $R = +1$

Terminology: reward

reward R

- Collect a coin: $R = +1$
- Win the game: $R = +10000$

Terminology: reward

reward R

- Collect a coin: $R = +1$
- Win the game: $R = +10000$
- Touch a Goomba: $R = -10000$ (game over).

Terminology: reward

reward R

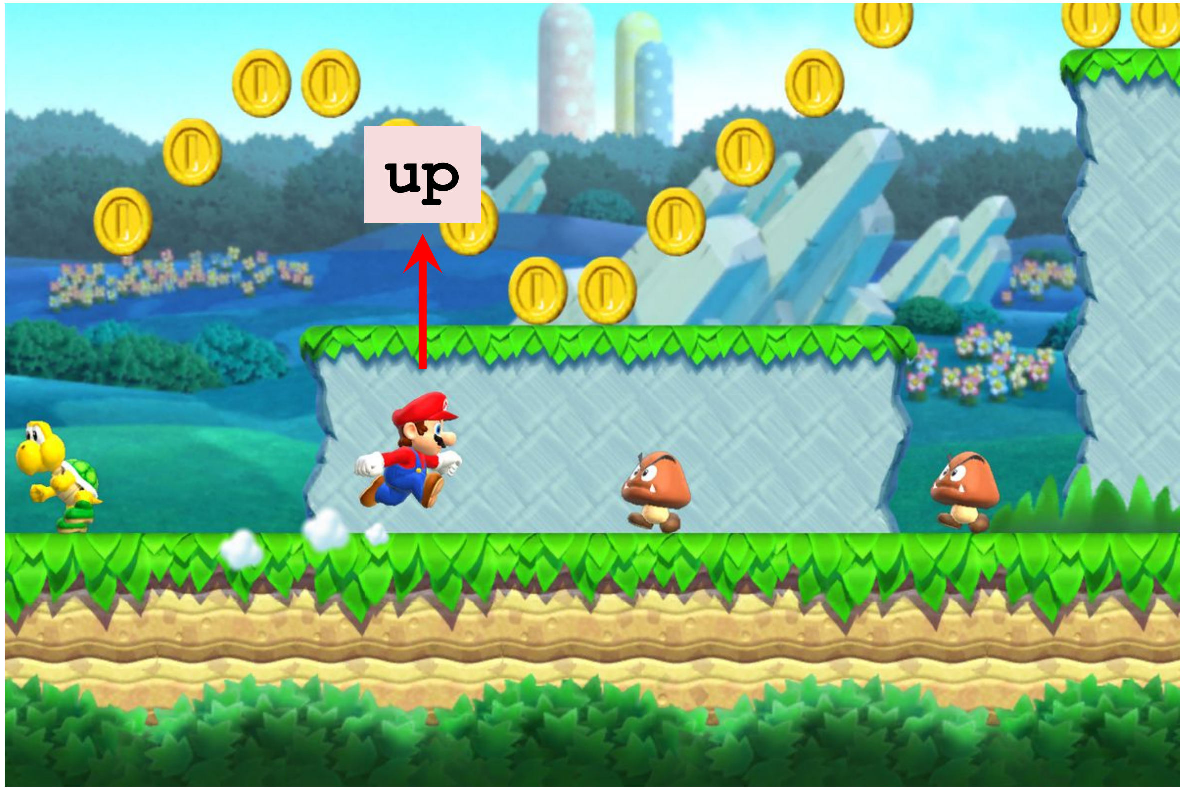
- Collect a coin: $R = +1$
- Win the game: $R = +10000$
- Touch a Goomba: $R = -10000$ (game over).
- Nothing happens: $R = 0$

Terminology: state transition

state transition

old state $\xrightarrow{\text{action}}$ new state

Terminology: state transition

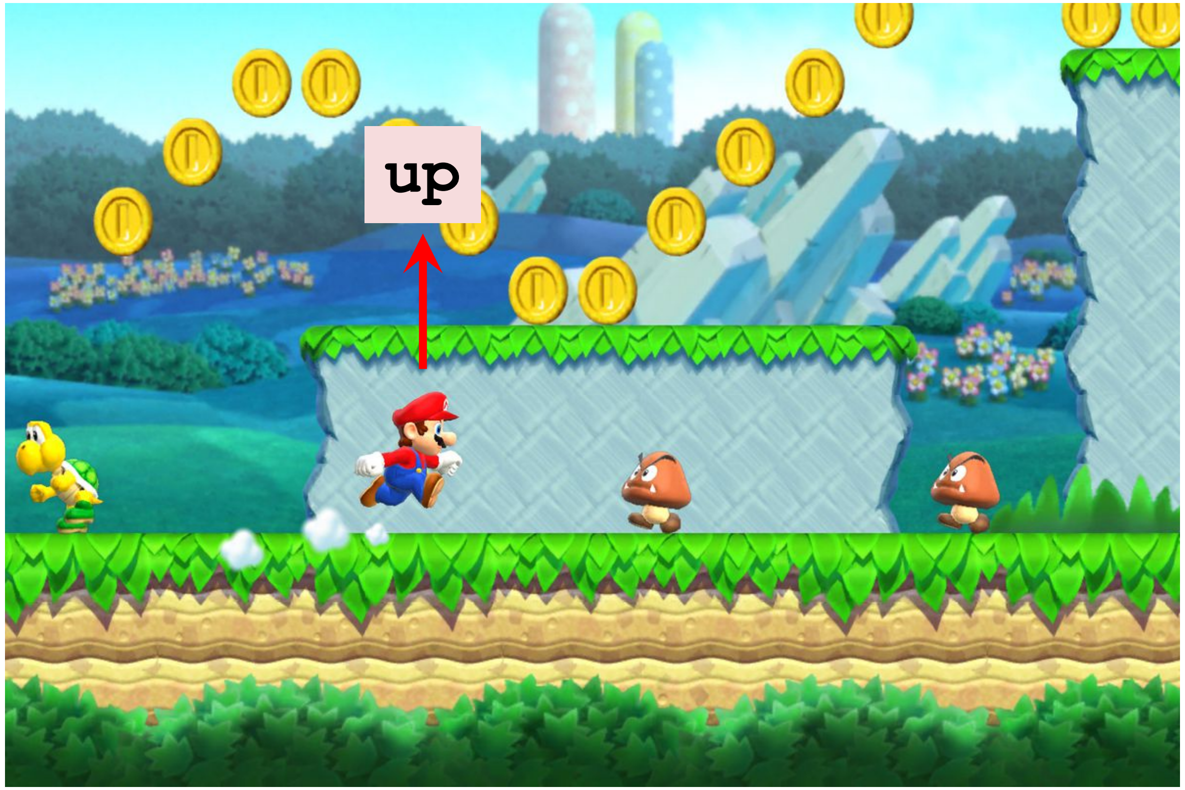


state transition

old state $\xrightarrow{\text{action}}$ new state

- E.g., "up" action leads to a new state.

Terminology: state transition



state transition

old state $\xrightarrow{\text{action}}$ new state

- E.g., "up" action leads to a new state.

- State transition can be random.
- Randomness is from the environment.

Terminology: state transition

state transition

old state $\xrightarrow{\text{action}}$ new state

- E.g., "up" action leads to a new state.

- State transition can be random.
- Randomness is from the environment.

Terminology: state transition

state transition

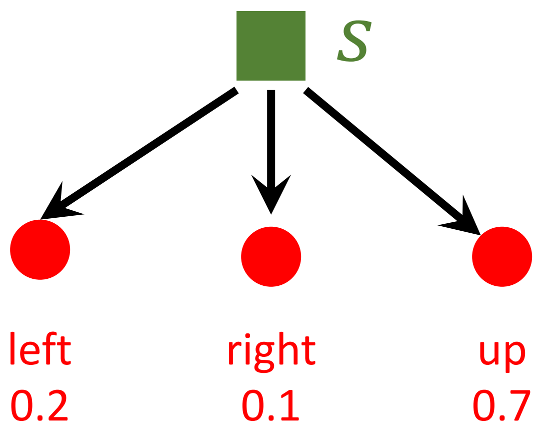
old state $\xrightarrow{\text{action}}$ new state

- E.g., "up" action leads to a new state.

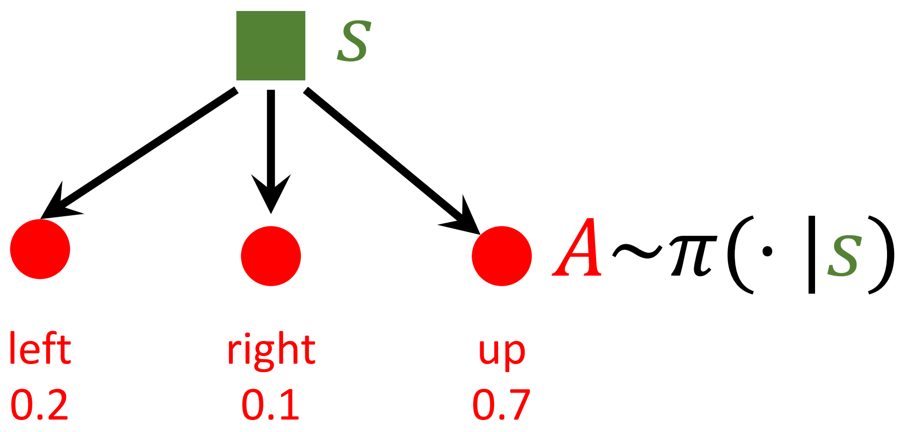
- State transition can be random.
- Randomness is from the environment.
- $p(s'|s,a) = \mathbb{P}(S' = s'|S = s, A = a)$.

Two Sources of Randomness

Randomness in Actions



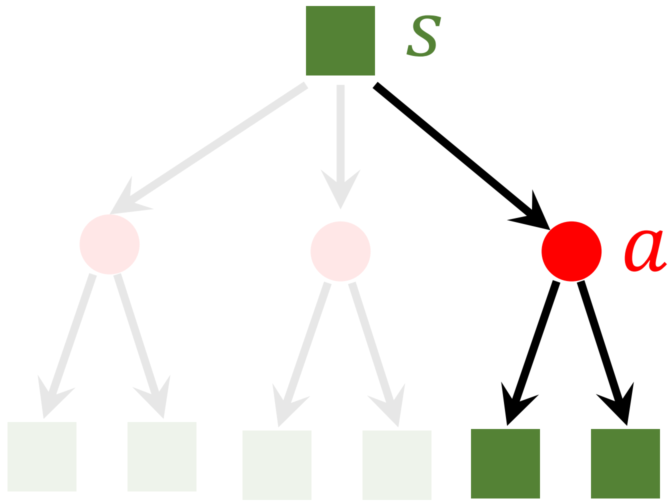
Randomness in Actions



Given state s, the action can be random, e.g., .

- $\pi(\text{“left”} \mid s) = 0.2$,
- $\pi(\text{“right”} \mid s) = 0.1$,
- $\pi(\text{“up”} \mid s) = 0.7$.

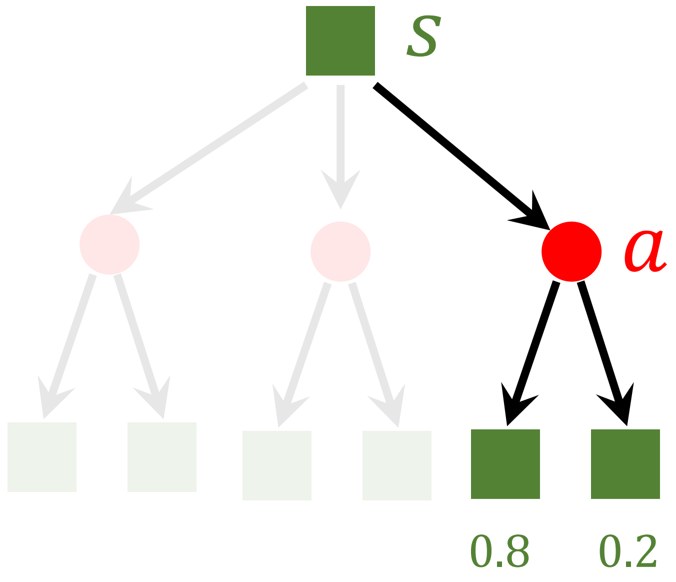
Randomness in States



- State transition can be random.
- The environment generates the new state S' by

$$S' \sim p(\cdot \mid s, a) \,.$$

Randomness in States



- State transition can be random.
- The environment generates the new state S' by

$$S' \sim p(\cdot \mid s, a)\ .$$

Two Sources of Randomness

- The randomness in action is from the policy function:

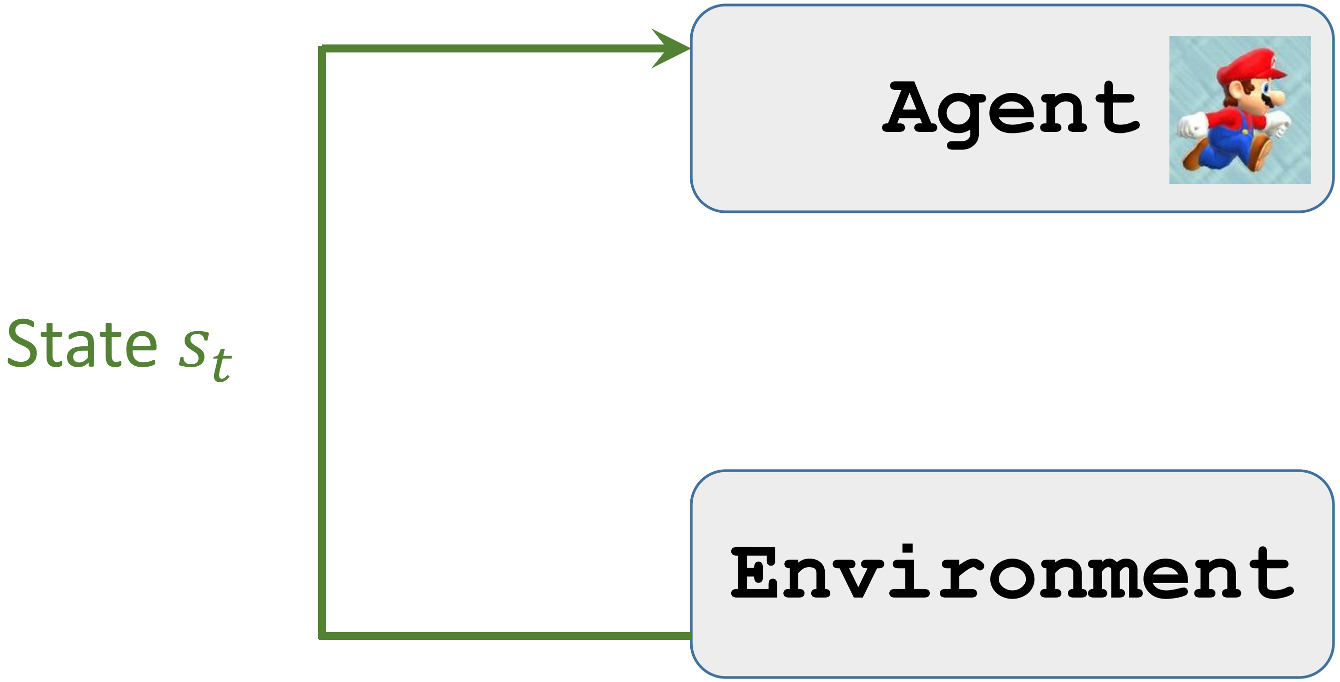
$$A \sim \pi(\cdot \mid s)\,.$$

- The randomness in state is from the state-transition function:

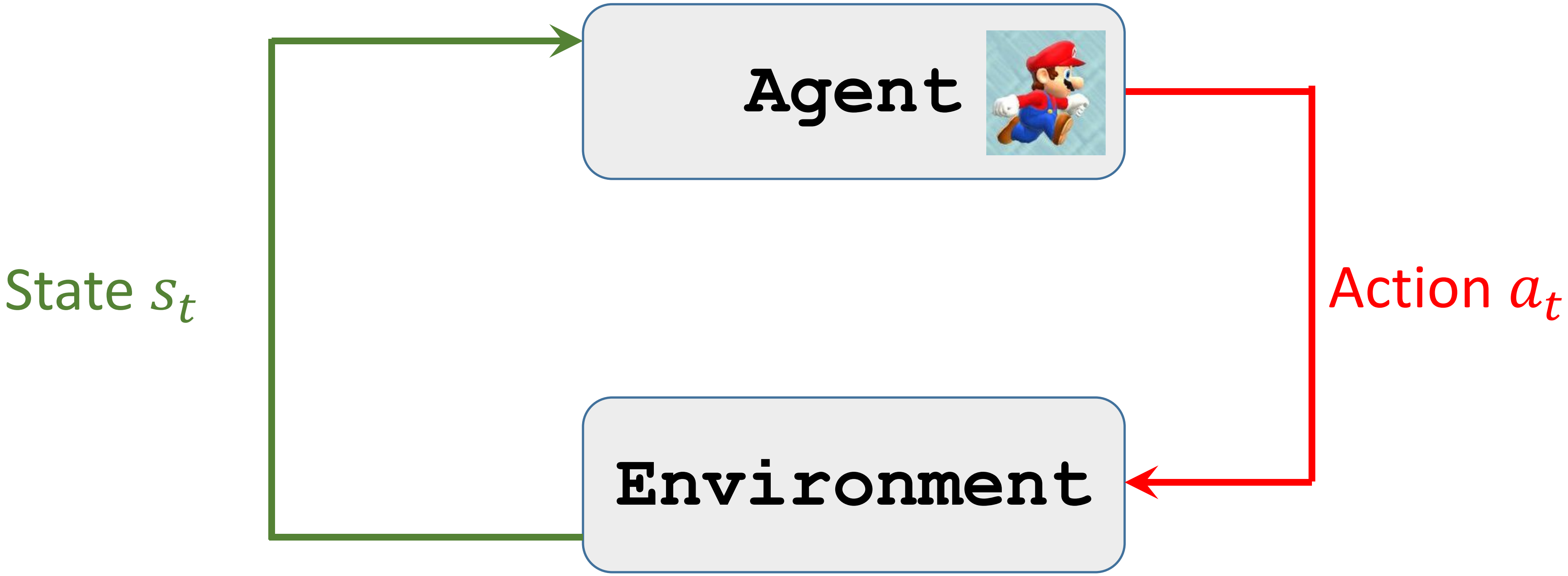
$$S' \sim p(\cdot \mid s, a)\,.$$

Agent-Environment Interaction

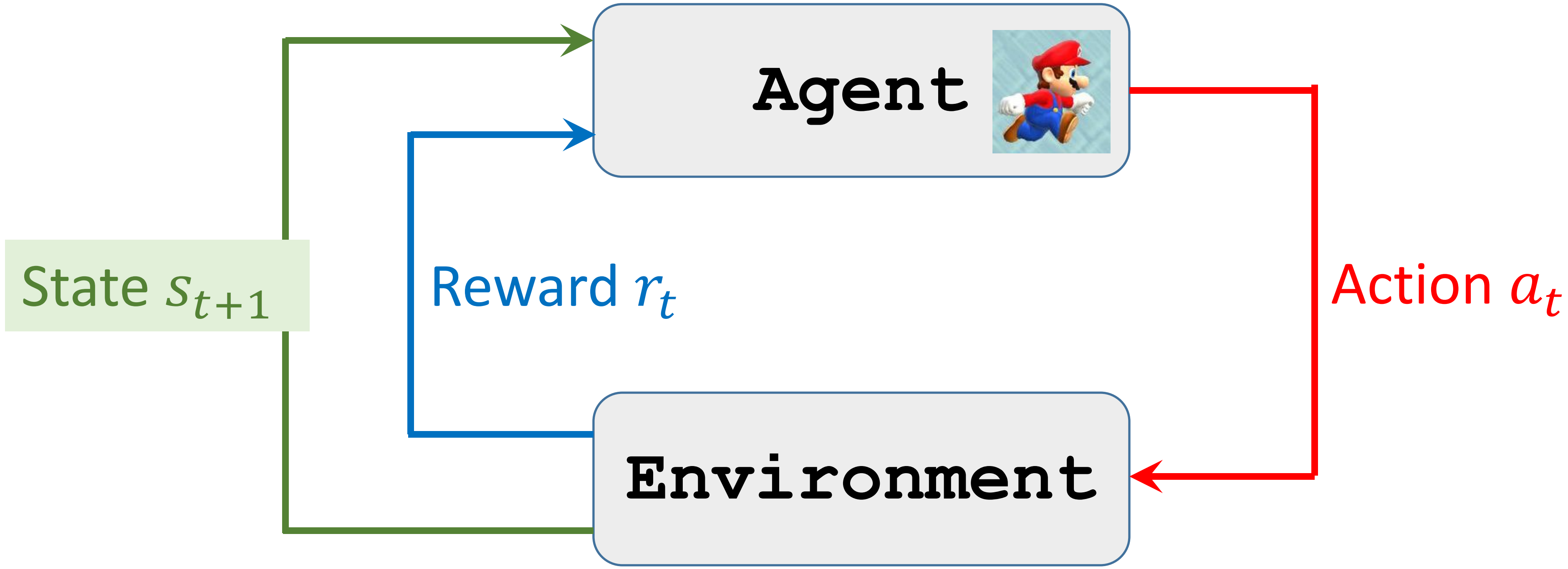
Agent-Environment Interaction



Agent-Environment Interaction



Agent-Environment Interaction



Play game using AI

- Observe state s_t, select action $a_t \sim \pi(\cdot \mid s_t)$, and execute a_t.
- The environment gives new state s_{t+1} and reward r_t.

Play game using AI

- Observe state s_t, select action $a_t \sim \pi(\cdot \mid s_t)$, and execute a_t.
- The environment gives new state s_{t+1} and reward r_t.

$$
\begin{tikzcd}
s_1 \arrow[r] & a_1 \arrow[r] \arrow[dr] & s_2 \\
& & r_1
\end{tikzcd}
$$

Play game using AI

- Observe state s_t, select action $a_t \sim \pi(\cdot \mid s_t)$, and execute a_t.
- The environment gives new state s_{t+1} and reward r_t.

$$\begin{tikzcd}
s_1 \arrow[r] & a_1 \arrow[r] \arrow[dr] & s_2 \arrow[r] & a_2 \arrow[r] \arrow[dr] & s_3 \arrow[r] & a_3 \arrow[r] \arrow[dr] & s_4 \arrow[r] & \cdots \\
& & r_1 & & r_2 & & r_3 &
\end{tikzcd}$$

Play game using AI

- (state, action, reward) trajectory:

$$\underline{s_1, a_1, r_1, \quad s_2, a_2, r_2, \quad \cdots \quad , \quad s_n, a_n, r_n.}$$

- One episode is from the the beginning to the end (Mario wins or dies).

$$\begin{tikzcd}
s_1 \arrow[r] & a_1 \arrow[r] \arrow[dr] & s_2 \arrow[r] & a_2 \arrow[r] \arrow[dr] & s_3 \arrow[r] & a_3 \arrow[r] \arrow[dr] & s_4 \arrow[r] & \cdots \\
& & r_1 & & r_2 & & r_3 &
\end{tikzcd}$$

Play game using AI

- (state, action, reward) trajectory:

$$s_1, a_1, r_1, \quad s_2, a_2, r_2, \quad \cdots \quad , \quad s_n, a_n, r_n.$$

- One episode is from the the beginning to the end (Mario wins or dies).

- What is a good policy?
- A good policy leads to big cumulative reward: $\sum_{t=1}^{n} \gamma^{t-1} \cdot r_t$.
- Use the rewards to guide the learning of policy.

Rewards and Returns

Return

Definition: Return (aka cumulative future reward).

- $U_t = R_t + R_{t+1} + R_{t+2} + R_{t+3} + \cdots$

Return

Definition: Return (aka cumulative future reward).

- $U_t = R_t + R_{t+1} + R_{t+2} + R_{t+3} + \cdots$

Question: At time t, are R_t and R_{t+1} equally important?

- Which of the followings do you prefer?
  - I give you \$100 right now.
  - I will give you \$100 one year later.

Return

Definition: Return (aka cumulative future reward).

- $U_t = R_t + R_{t+1} + R_{t+2} + R_{t+3} + \cdots$

Question: At time t, are R_t and R_{t+1} equally important?

- Which of the followings do you prefer?
  - I give you \$80 right now.
  - I will give you \$100 one year later.

Return

Definition: Return (aka cumulative future reward).

- $U_t = R_t + R_{t+1} + R_{t+2} + R_{t+3} + \cdots$

Question: At time t, are R_t and R_{t+1} equally important?

- Which of the followings do you prefer?
  - I give you \$80 right now.
  - I will give you \$100 one year later.

- Future reward is less valuable than present reward.
- R_{t+1} should be given less weight than R_t.

Discounted Returns

Definition: Return (aka cumulative future reward).

- $U_t = R_t + R_{t+1} + R_{t+2} + R_{t+3} + \cdots$

Definition: Discounted return (aka cumulative discounted future reward).

- γ: discount factor (tuning hyper-parameter).
- $U_t = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \gamma^3 R_{t+3} + \cdots$

Discounted Returns

Definition: Discounted return (at time t).

- $U_t = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \cdots + \gamma^{n-t} R_n.$

Randomness in Returns

Definition: Discounted return (at time t).

- $U_t = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \cdots + \gamma^{n-t} R_n.$

At the end of the game, we observe u_t.

- We observe all the rewards, $r_t, \; r_{t+1}, \; r_{t+2}, \; \cdots, \; r_n.$
- We thereby know the discounted return:

$$u_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots + \gamma^{n-t} r_n.$$

Randomness in Returns

Definition: Discounted return (at time t).

- $U_t = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \cdots + \gamma^{n-t} R_n.$

At time t, the rewards, $R_t, \cdots, R_n$, are random, so the return U_t is random.

Randomness in Returns

Definition: Discounted return (at time t).

- $U_t = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \cdots + \gamma^{n-t} R_n.$

At time t, the rewards, $R_t, \cdots, R_n$, are random, so the return U_t is random.

- Reward R_i depends on S_i and A_i.
- States can be random: $S_i \sim p(\cdot \mid s_{i-1}, a_{i-1})$.
- Actions can be random: $A_i \sim \pi(\cdot \mid s_i)$.
- If either S_i or A_i is random, then R_i is random.

Randomness in Returns

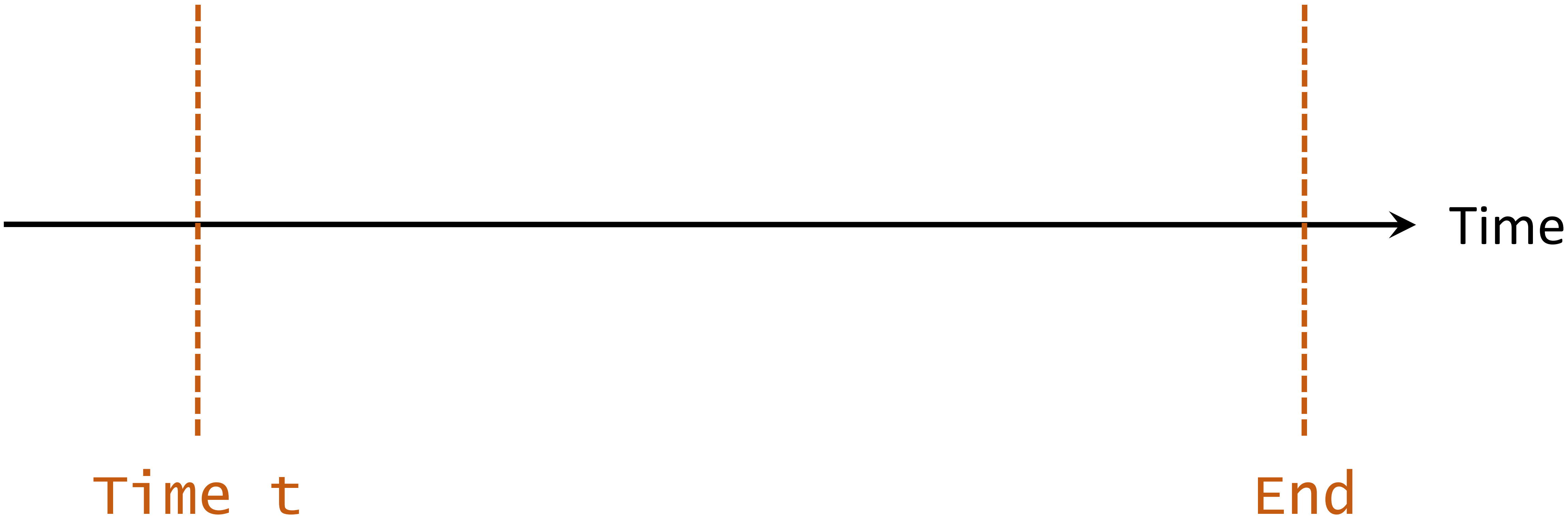
Definition: Discounted return (at time t).

- $U_t = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \cdots + \gamma^{n-t} R_n.$

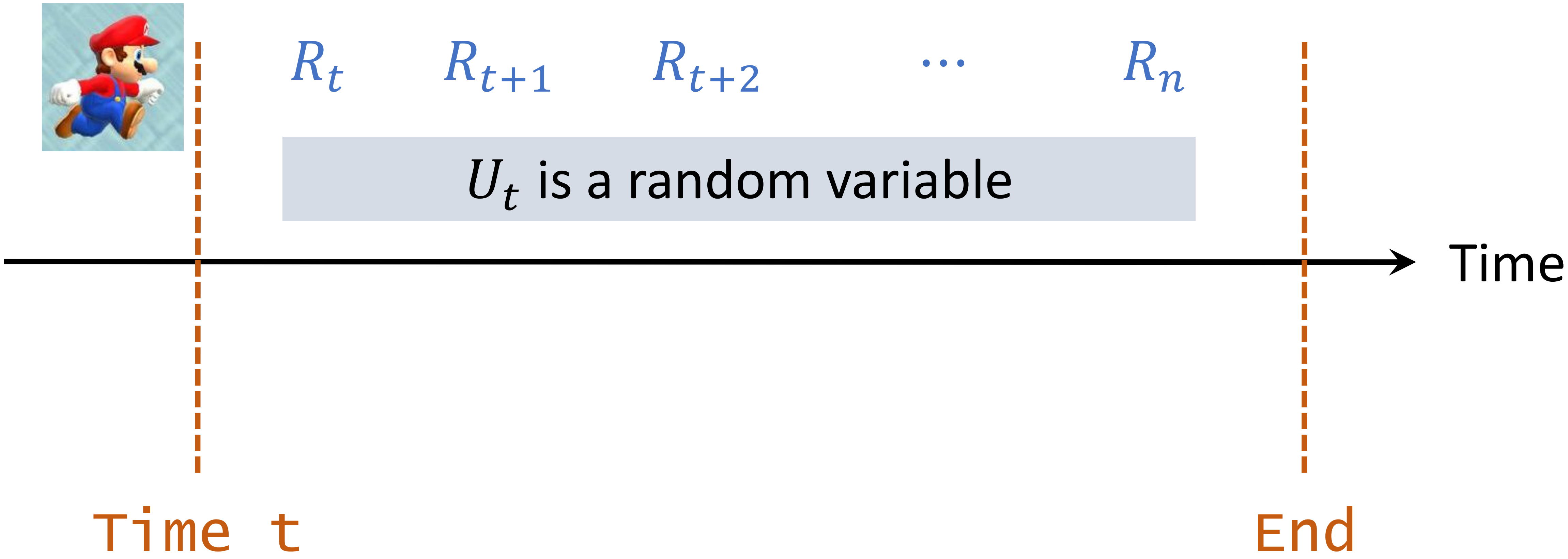
At time t, the rewards, $R_t, \cdots, R_n$, are random, so the return U_t is random.

- Reward R_i depends on S_i and A_i.
- U_t depends on $\underline{R_t, R_{t+1}, \cdots, R_n}$.
- ➔ U_t depends on $\underline{S_t, A_t, S_{t+1}, A_{t+1}, \cdots, S_n, A_n}$.

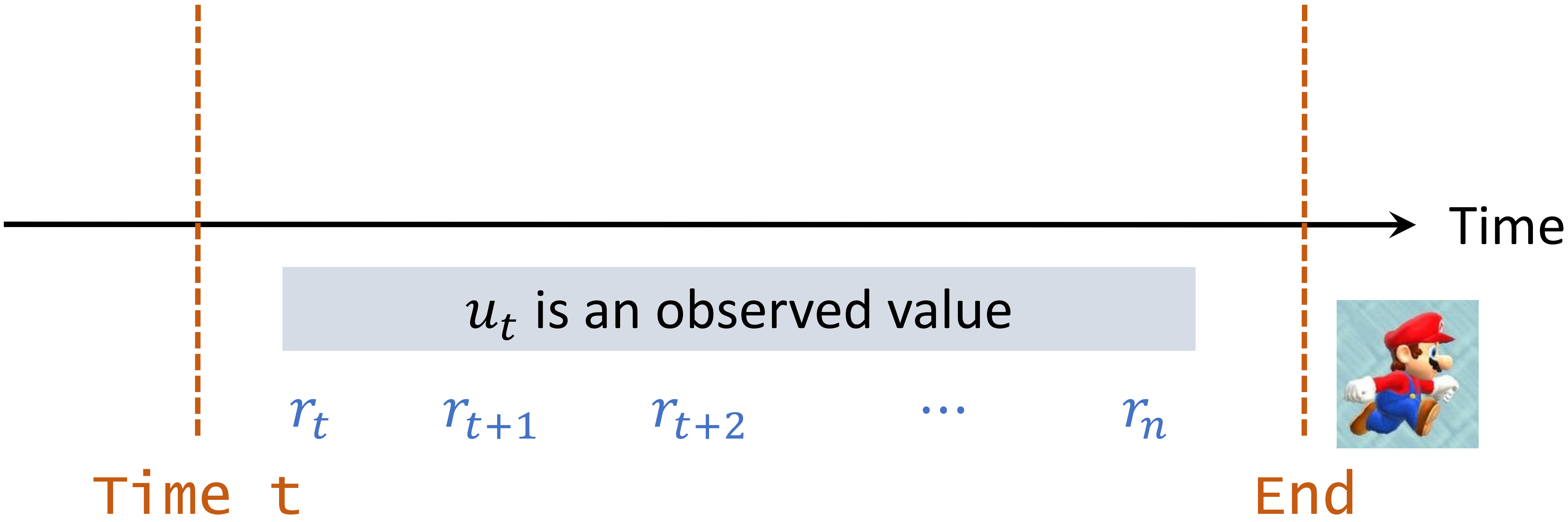
Randomness in Returns



Randomness in Returns



Randomness in Returns



Value Functions

Action-Value Function $Q_\pi(s, a)$

Definition: Discounted return.

- $U_t = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \cdots + \gamma^{n-t} R_n.$

Action-Value Function $Q_\pi(s, a)$

Definition: Discounted return.

- $U_t = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \cdots + \gamma^{n-t} R_n.$

Definition: Action-value function.

- $\boxed{Q_\pi(s_t, a_t)} = \boxed{\mathbb{E}\left[U_t \mid S_t = s_t, A_t = a_t \right].}$

Action-Value Function $Q_\pi(s, a)$

Definition: Discounted return.

- $U_t = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \cdots + \gamma^{n-t} R_n.$

Definition: Action-value function.

- $Q_\pi(s_t, a_t) = \mathbb{E}[U_t \mid S_t = s_t, A_t = a_t].$

U_t depends on states $\underline{S_t, S_{t+1}, \cdots, S_n}$ and actions $\underline{A_t, A_{t+1}, \cdots, A_n}$.

Action-Value Function $Q_\pi(s, a)$

Definition: Discounted return.

- $U_t = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \cdots + \gamma^{n-t} R_n.$

Definition: Action-value function.

- $Q_\pi(s_t, a_t) = \mathbb{E}\left[U_t \mid \boxed{S_t = s_t, A_t = a_t} \right].$

Regard s_t and a_t as observed values.

Regard $\underline{S_{t+1}, \cdots, S_n}$ and $\underline{A_{t+1}, \cdots, A_n}$ as random variables.

Action-Value Function $Q_\pi(s, a)$

Definition: Discounted return.

- $U_t = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \cdots + \gamma^{n-t} R_n.$

Definition: Action-value function.

- $Q_\pi(s_t, a_t) = \boxed{\mathbb{E}}[\, U_t \mid S_t = s_t, A_t = a_t \,].$

Regard $\underline{S_{t+1}, \cdots, S_n}$ and $\underline{A_{t+1}, \cdots, A_n}$ as random variables.

Action-Value Function $Q_\pi(s, a)$

Definition: Discounted return.

- $U_t = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \cdots + \gamma^{n-t} R_n$.

Definition: Action-value function.

- $Q_\pi(s_t, a_t) = \boxed{\mathbb{E}}[\, U_t \mid S_t = s_t, A_t = a_t \,]$.

- $S_{t+1} \sim p(\cdot \mid s_t, a_t)$,

$\vdots$

- $S_n \sim p(\cdot \mid s_{n-1}, a_{n-1})$.

- $A_{t+1} \sim \pi(\cdot \mid s_{t+1})$,

$\vdots$

- $A_n \sim \pi(\cdot \mid s_n)$.

Action-Value Function $Q_\pi(s, a)$

Definition: Discounted return.

- $U_t = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \cdots + \gamma^{n-t} R_n.$

Definition: Action-value function.

- $Q_\pi(s_t, a_t) = \mathbb{E}[U_t \mid S_t = s_t, A_t = a_t].$

Action-Value Function $Q_\pi(s, a)$

Definition: Discounted return.

- $U_t = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \cdots + \gamma^{n-t} R_n.$

Definition: Action-value function.

- $Q_\pi(s_t, a_t) = \mathbb{E}[U_t \mid S_t = s_t, A_t = a_t].$

Action-Value Function $Q_\pi(s, a)$

Definition: Discounted return.

- $U_t = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \cdots + \gamma^{n-t} R_n.$

Definition: Action-value function.

- $Q_\pi(s_t, a_t) = \mathbb{E}[U_t \mid S_t = s_t, A_t = a_t].$

- $Q_\pi(s_t, a_t)$ depends on s_t, a_t, π, and p.
- $Q_\pi(s_t, a_t)$ is dependent of $\underline{S_{t+1}, \cdots, S_n}$ and $\underline{A_{t+1}, \cdots, A_n}$.

State-Value Function $V_\pi(s)$

Definition: Discounted return.

- $U_t = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \cdots + \gamma^{n-t} R_n.$

Definition: Action-value function.

- $Q_\pi(s_t, a_t) = \mathbb{E}[\, U_t \mid S_t = s_t, A_t = a_t \,].$

Definition: State-value function.

- $V_\pi(s_t) = \mathbb{E}_A\, [Q_\pi(s_t, A)]$

State-Value Function $V_\pi(s)$

Definition: Discounted return.

- $U_t = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \cdots + \gamma^{n-t} R_n.$

Definition: Action-value function.

- $Q_\pi(s_t, a_t) = \mathbb{E}[U_t \mid S_t = s_t, A_t = a_t].$

Definition: State-value function.

- $V_\pi(s_t) = \mathbb{E}_A[Q_\pi(s_t, A)]$

Taken w.r.t. the action $A \sim \pi(\cdot \mid s_t)$.

State-Value Function $V_\pi(s)$

Definition: Discounted return.

- $U_t = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \cdots + \gamma^{n-t} R_n.$

Definition: Action-value function.

- $Q_\pi(s_t, a_t) = \mathbb{E}[\, U_t \mid S_t = s_t, A_t = a_t \,].$

Definition: State-value function.

- $V_\pi(s_t) = \mathbb{E}_A[Q_\pi(s_t, A)] = \underline{\sum_a \pi(a|s_t) \cdot Q_\pi(s_t, a)}.$ (Actions are discrete.)

Taken w.r.t. the action $A \sim \pi(\cdot \mid s_t)$.

State-Value Function $V_\pi(s)$

Definition: Discounted return.

- $U_t = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \cdots + \gamma^{n-t} R_n.$

Definition: Action-value function.

- $Q_\pi(s_t, a_t) = \mathbb{E}[U_t \mid S_t = s_t, A_t = a_t].$

Definition: State-value function.

- $V_\pi(s_t) = \mathbb{E}_A[Q_\pi(s_t, A)] = \sum_a \pi(a|s_t) \cdot Q_\pi(s_t, a).$ (Actions are discrete.)
- $V_\pi(s_t) = \mathbb{E}_A[Q_\pi(s_t, A)] = \underline{\int \pi(a|s_t) \cdot Q_\pi(s_t, a)\, da}.$ (Actions are continuous.)

Understanding the Value Functions

- Action-value function: $\underline{Q_\pi(s, a)} = \underline{\mathbb{E}\left[U_t | S_t = s, A_t = a\right]}.$
- Given policy π, $Q_\pi(s, a)$ evaluates how good it is for an agent to pick action a while being in state s.

Understanding the Value Functions

- Action-value function: $Q_\pi(s, a) = \mathbb{E}\left[U_t | S_t = s, A_t = a\right]$.
- Given policy π, $Q_\pi(s, a)$ evaluates how good it is for an agent to pick action a while being in state s.

- State-value function: $V_\pi(s) = \underline{\mathbb{E}_A\left[Q_\pi(s, A)\right]}$
- For fixed policy π, $V_\pi(s)$ evaluates how good the situation is in state s.
- $\underline{\mathbb{E}_S[V_\pi(S)]}$ evaluates how good the policy π is.

Evaluating Reinforcement Learning

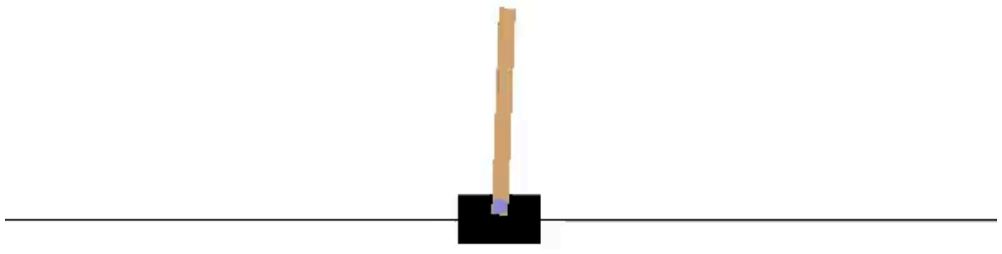
OpenAI Gym

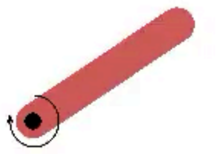
- Gym is a toolkit for developing and comparing reinforcement learning algorithms.
- https://gym.openai.com/

OpenAI Gym

- Gym is a toolkit for developing and comparing reinforcement learning algorithms.
- https://gym.openai.com/

Classical control problems



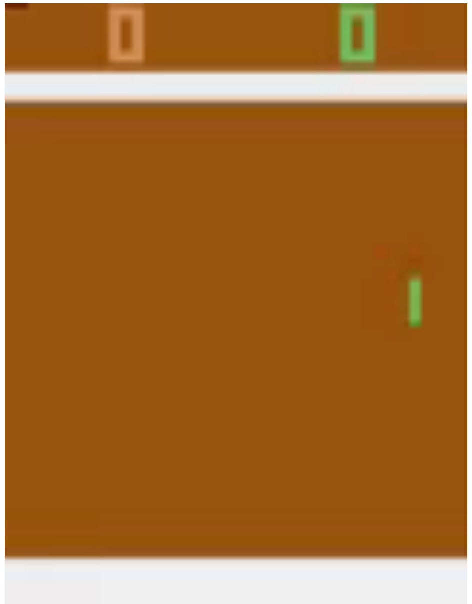
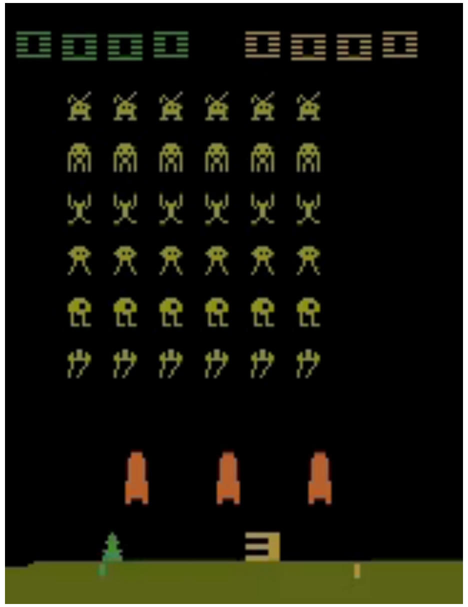


Cart Pole

Pendulum

OpenAI Gym

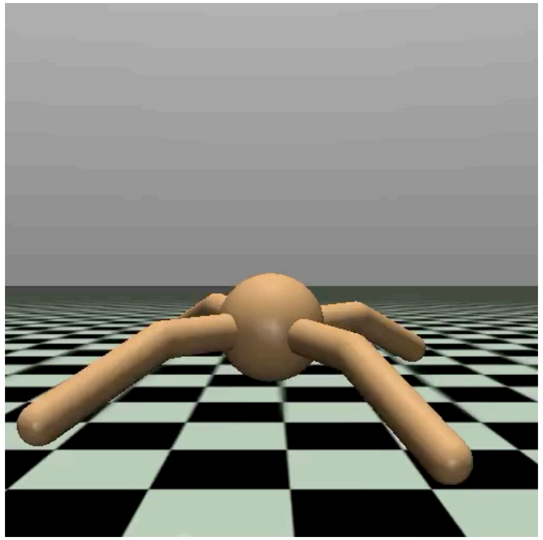
- Gym is a toolkit for developing and comparing reinforcement learning algorithms.
- https://gym.openai.com/

Atari Games		
		
Pong	Space Invader	Breakout

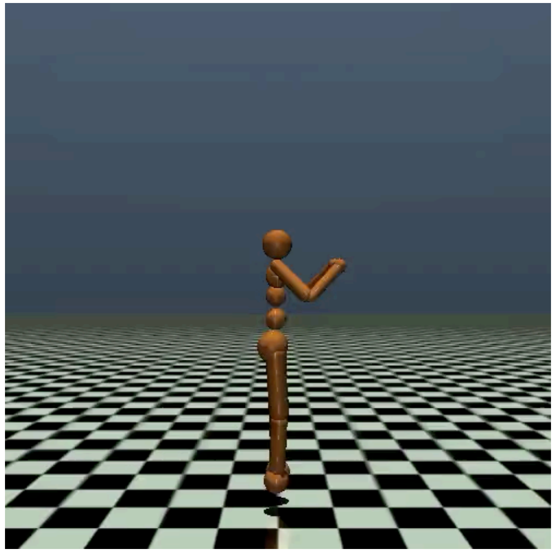
OpenAI Gym

- Gym is a toolkit for developing and comparing reinforcement learning algorithms.
- https://gym.openai.com/

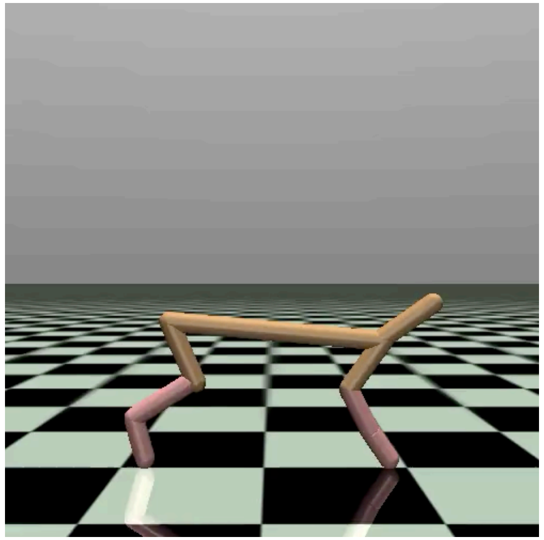
MuJoCo (Continuous control tasks.)



Ant

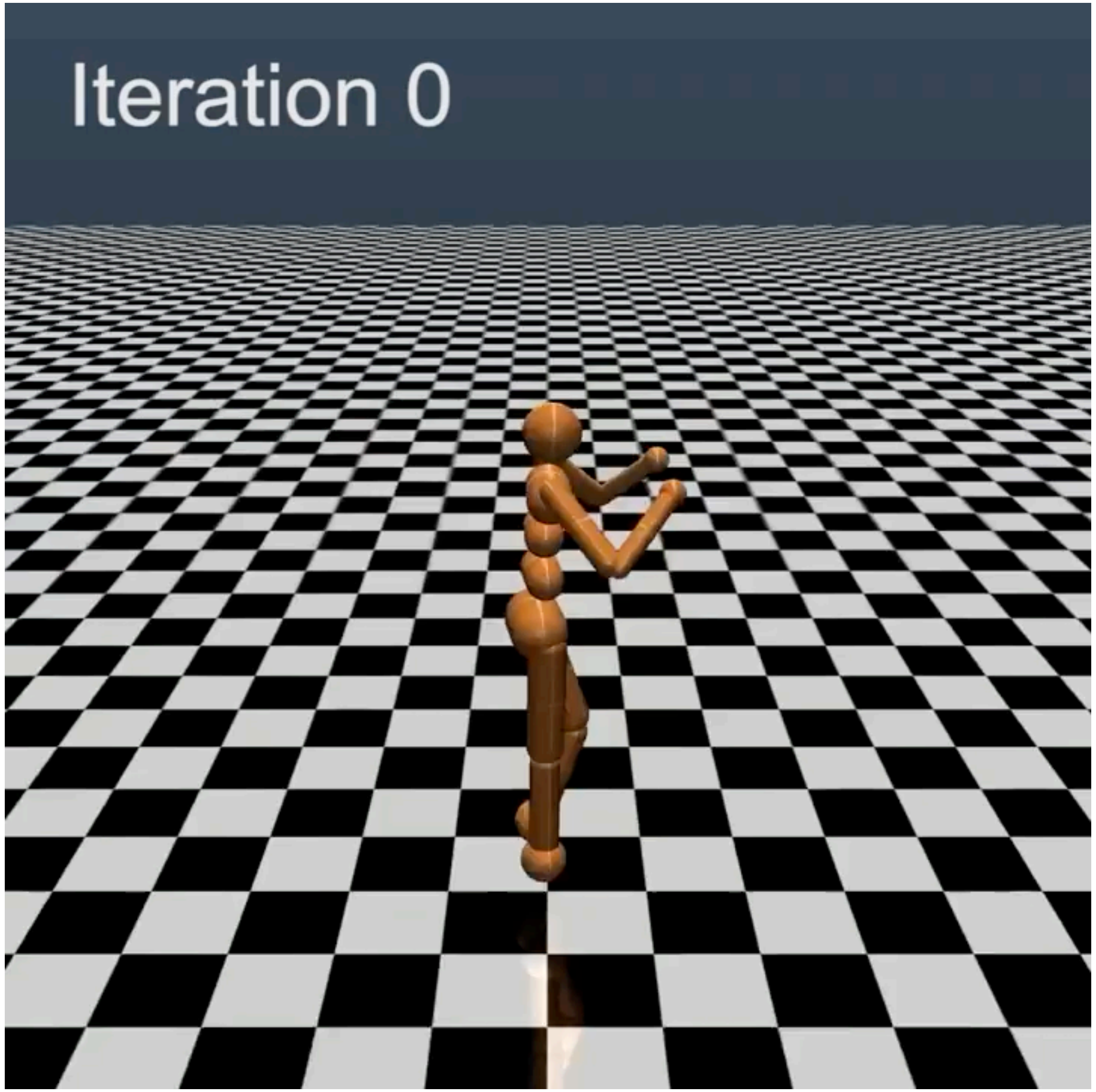


Humanoid



Half Cheetah

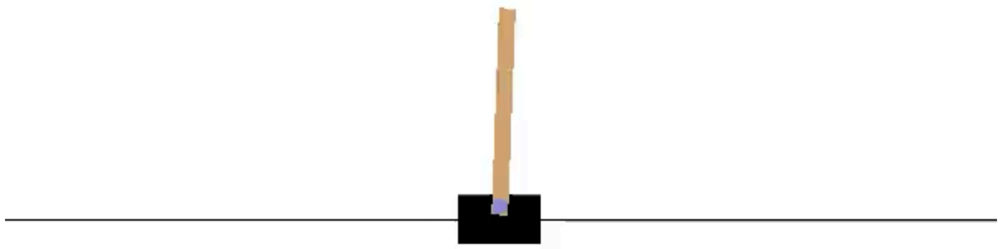
OpenAI Gym



Play CartPole Game

```
import gym
env = gym.make('CartPole-v0')
```

- Get the environment of CartPole from Gym.
- "env" provides states and reward.



Play CartPole Game

```
state = env.reset()

for t in range(100):
    env.render()
    print(state)

    action = env.action_space.sample()
    state, reward, done, info = env.step(action)

    if done:
        print('Finished')
        break

env.close()
```

A window pops up rendering CartPole.

A random action.

"done=1" means finished (win or lose the game)

Summary

Summary

Terminologies

- Agent 
- Environment
- State s
- Action a
- Reward r

Summary

Terminologies

- Agent 
- Environment
- State s
- Action a
- Reward r
- Policy $\pi(a|s)$
- State transition $p(s'|s, a)$

Summary

Terminologies

- Agent 
- Environment
- State s
- Action a
- Reward r
- Policy $\pi(a|s)$
- State transition $p(s'|s, a)$

Return and Value

- Return:

$$U_t = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \cdots$$

Summary

Terminologies

- Agent 
- Environment
- State s
- Action a
- Reward r
- Policy $\pi(a|s)$
- State transition $p(s'|s,a)$

Return and Value

- Return:

$$U_t = R_t + \gamma\, R_{t+1} + \gamma^2\, R_{t+2} + \cdots$$

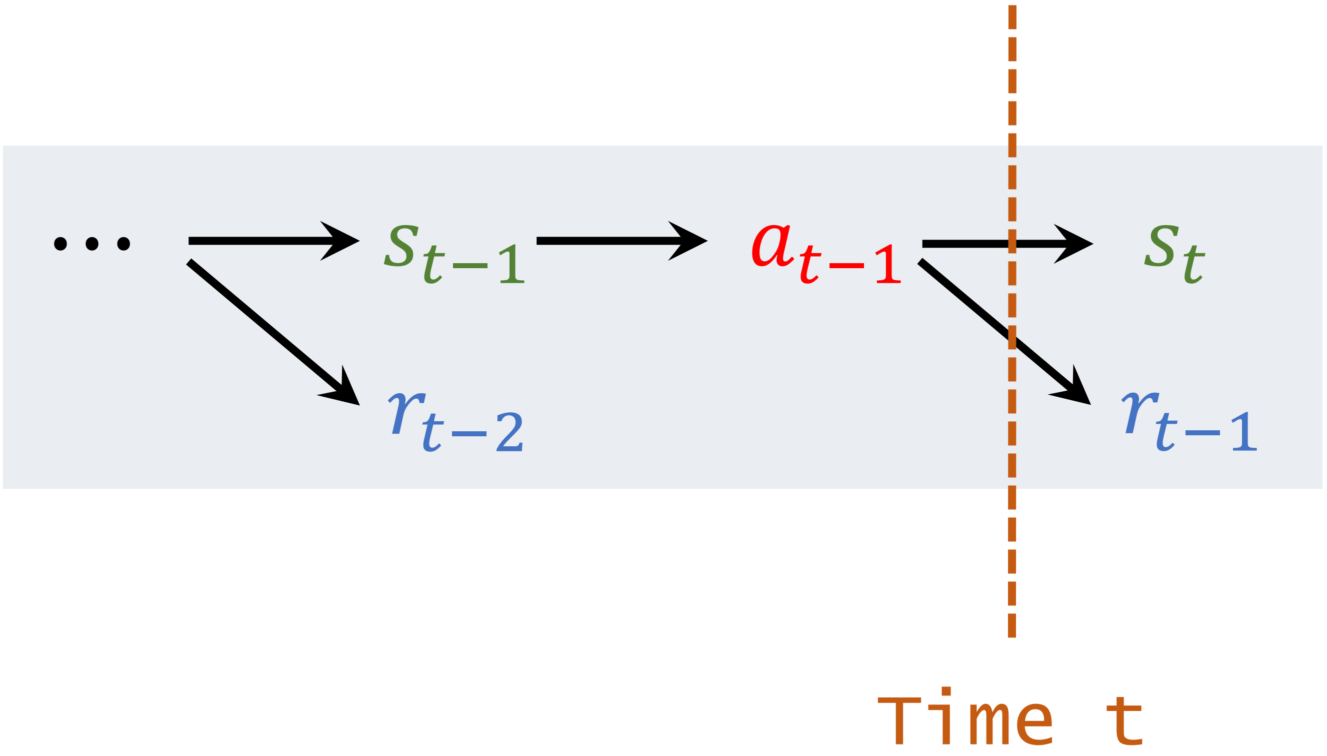
- Action-value function:

$$Q_\pi(s_t, a_t) = \mathbb{E}\,[U_t | s_t, a_t].$$

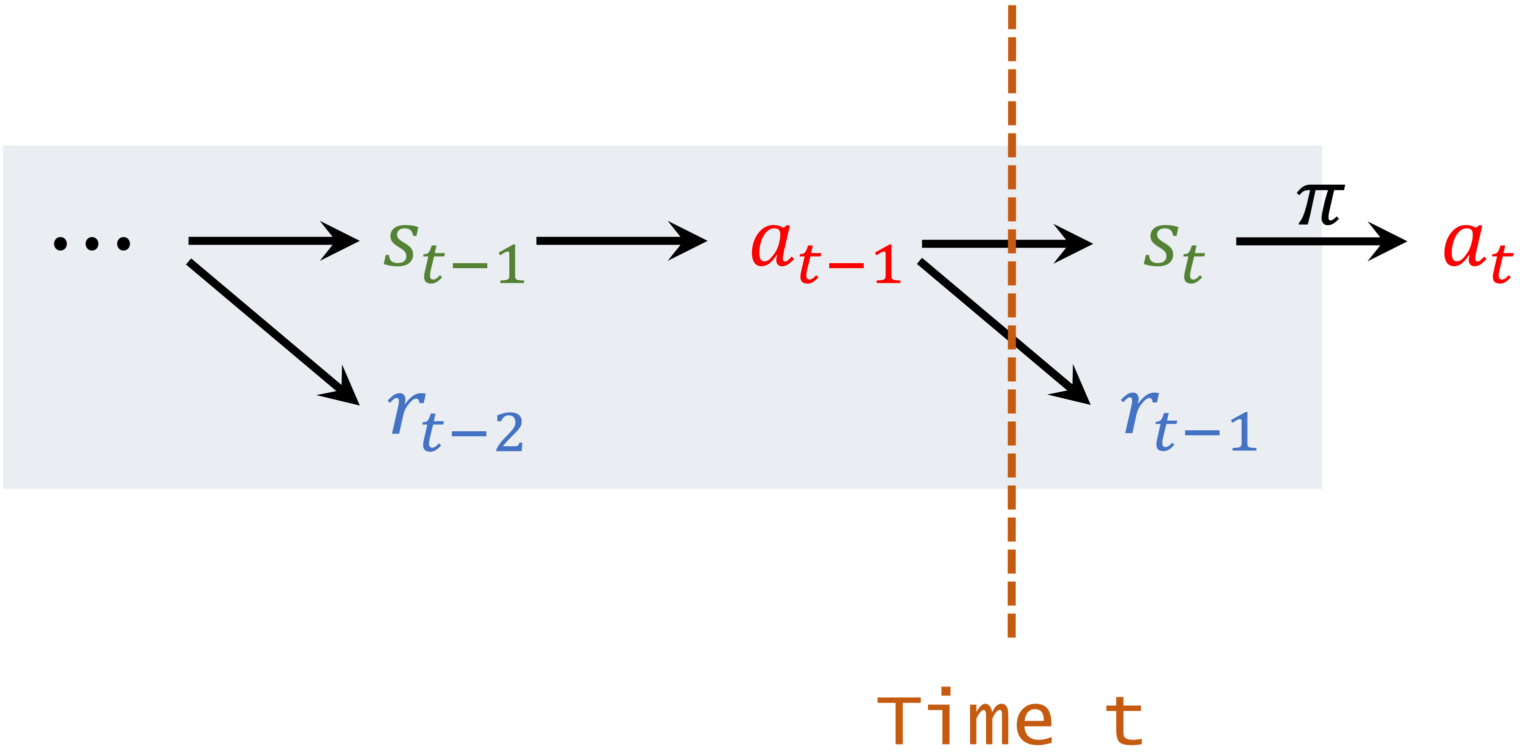
- State-value function:

$$V_\pi(s_t) = \mathbb{E}_{A\sim\pi}[Q_\pi(s_t, A)].$$

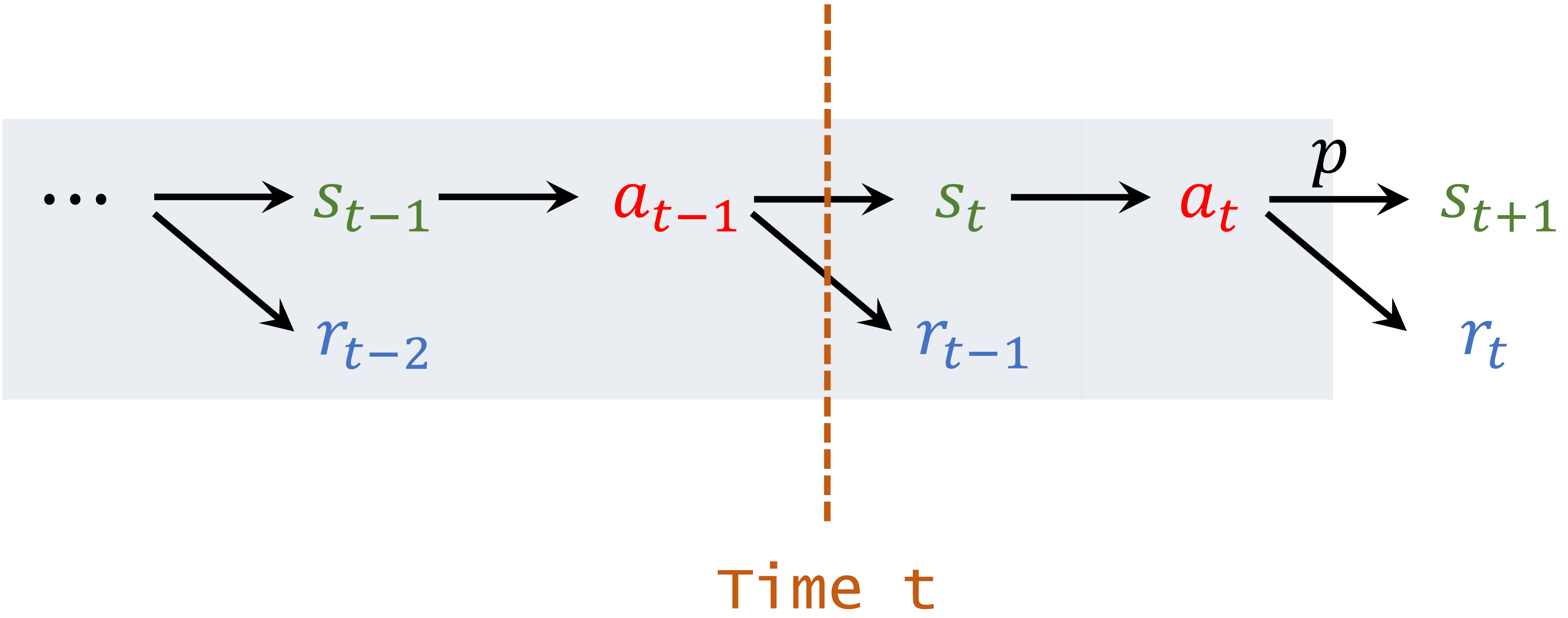
Play game using AI



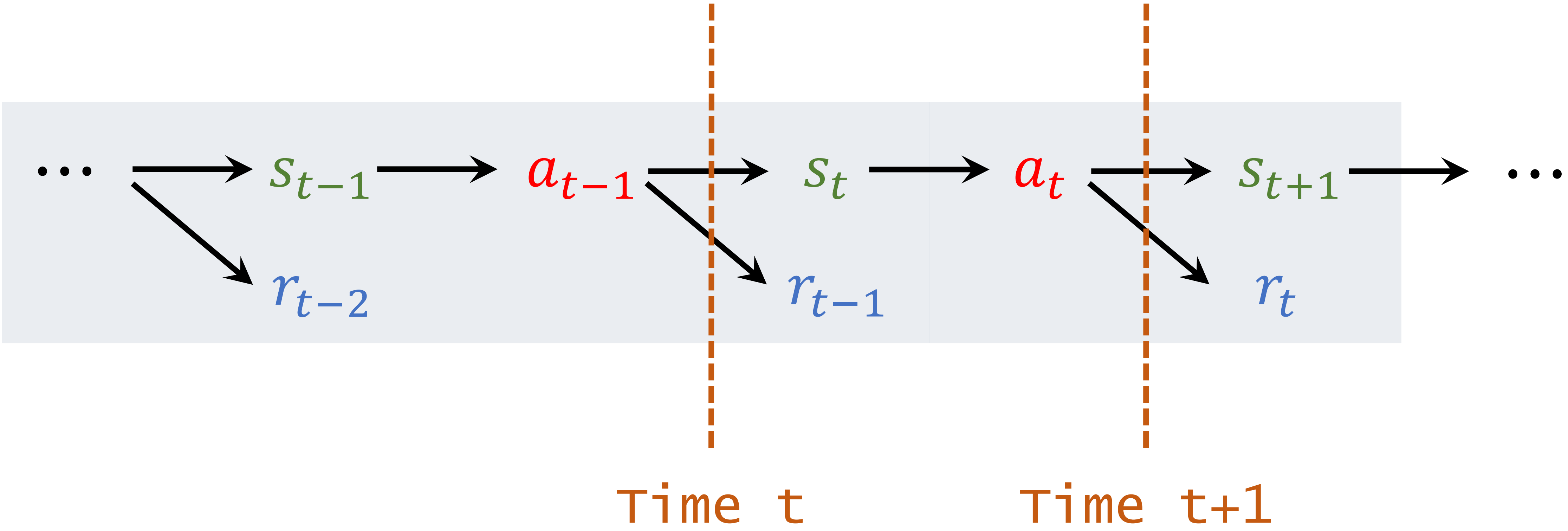
Play game using AI



Play game using AI



Play game using AI



Thank You!