Deep Reinforcement Learning

Lecture 2 — Markov Decision Process

國立清華大學
NATIONAL TSING HUA UNIVERSITY

National Tsing Hua University
Department of Computer Science

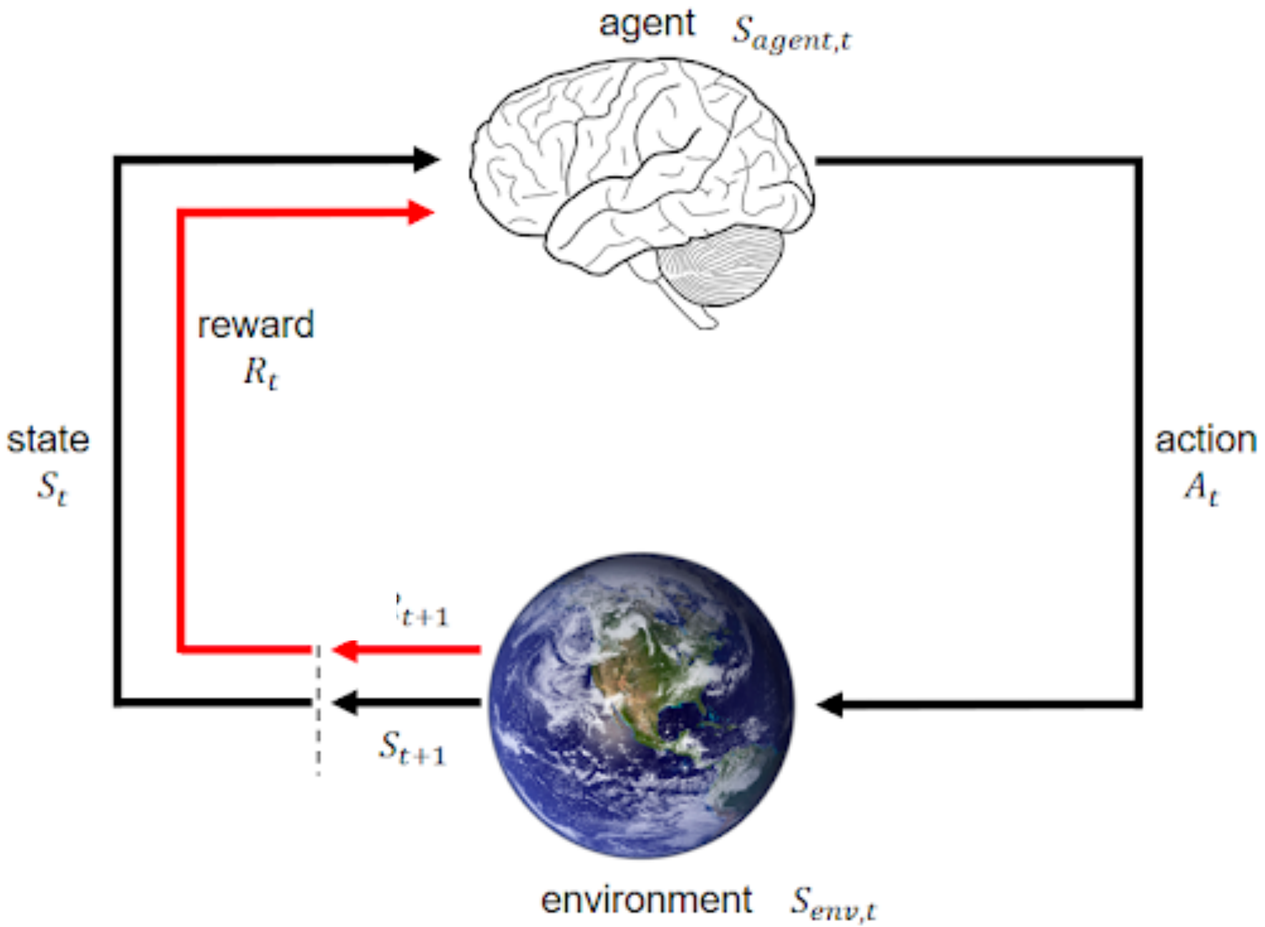
Prof. Chun-Yi Lee

Outline

- Background
- Markov Process Family
- Bellman Equation
- Value Function, Q-function, and Policy

Reinforcement Learning

Agent and Environment



The agent

- Receives a state $S_t \in S$
- Outputs an action $A_t \in A$
- Receives a reward $R_t \in R$

The environment

- Receives an action A_t
- Outputs the next state S_{t+1}
- Outputs a reward $R_t \in R$

Reinforcement Learning

The Goal of RL

Maximize the accumulated expected return G_t at each timestep *t*

$$G_t = R_t + R_{t+1} + R_{t+2} + \ldots + R_T$$

- The agent-environment interaction will end at time step *T* and reset to a starting state
  - This is called an **episode**
  - This kind of tasks are called **episodic tasks.**

Reinforcement Learning

The Goal of RL

Maximize the accumulated expected return G_t at each timestep *t*

$$G_t = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \ldots = \sum_{i=t}^{\infty} \gamma^{i-t} R_i$$

- When *T* is infinite, the agent-environment interaction does not end
  - Therefore, G_t could itself easily become infinite without a **discount factor** $\gamma \in [0,1]$
  - This kind of tasks are called **continuing tasks.**

Reinforcement Learning

Discount Factor

- γ close to 0 leads to “myopic” evaluation
- γ close to 1 leads to “far-sighted” evaluation

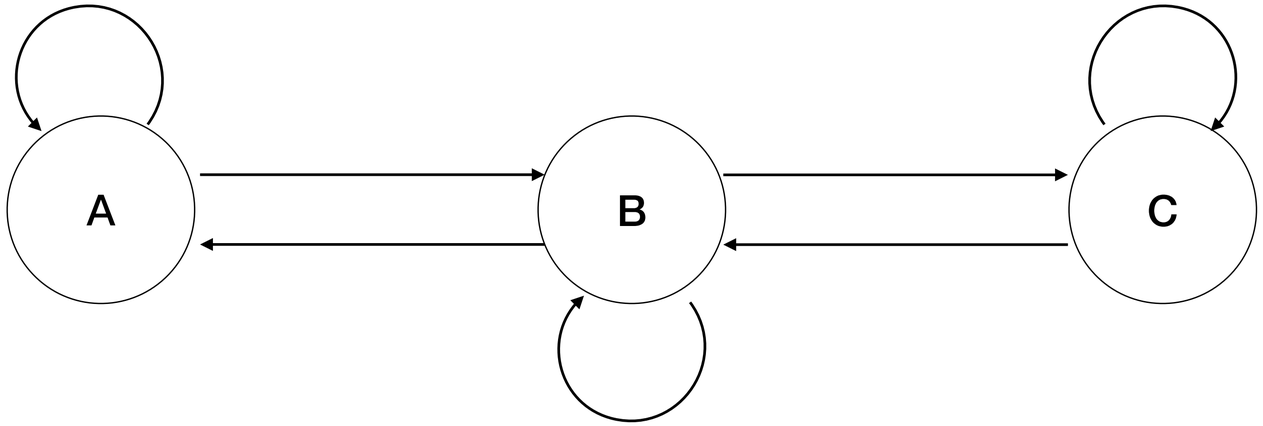
$$G_t = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \ldots = \sum_{i=t}^{\infty} \gamma^{i-t} R_i$$

Outline

- Background
- Markov Process Family
- Bellman Equation
- Value Function, Q-function, and Policy

Markov Process

- A Markov Process is a random process that satisfies the ***Markov property***



Andrey Markov

Markov Process

Markov Property

- If the state signal has ***Markov Property***, the environment's response at $t+1$ depends only on the state and action representations at t

$$P(S_{t+1} \mid S_t, A_t) = P(S_{t+1} \mid S_0, A_0, \ldots S_{t-1}, A_{t-1}, S_t, A_t)$$

Markov Process

State Transition

- The state transition probability is defined as follows:

$$P_{ss'} = P(S_{t+1} = s' \mid S_t = s)$$

- The state transition matrix P is defined as follows:

$$P = \begin{pmatrix} P_{11} & \cdots & P_{1n} \\ \vdots & & \vdots \\ P_{n1} & \cdots & P_{nn} \end{pmatrix}$$

Markov Process

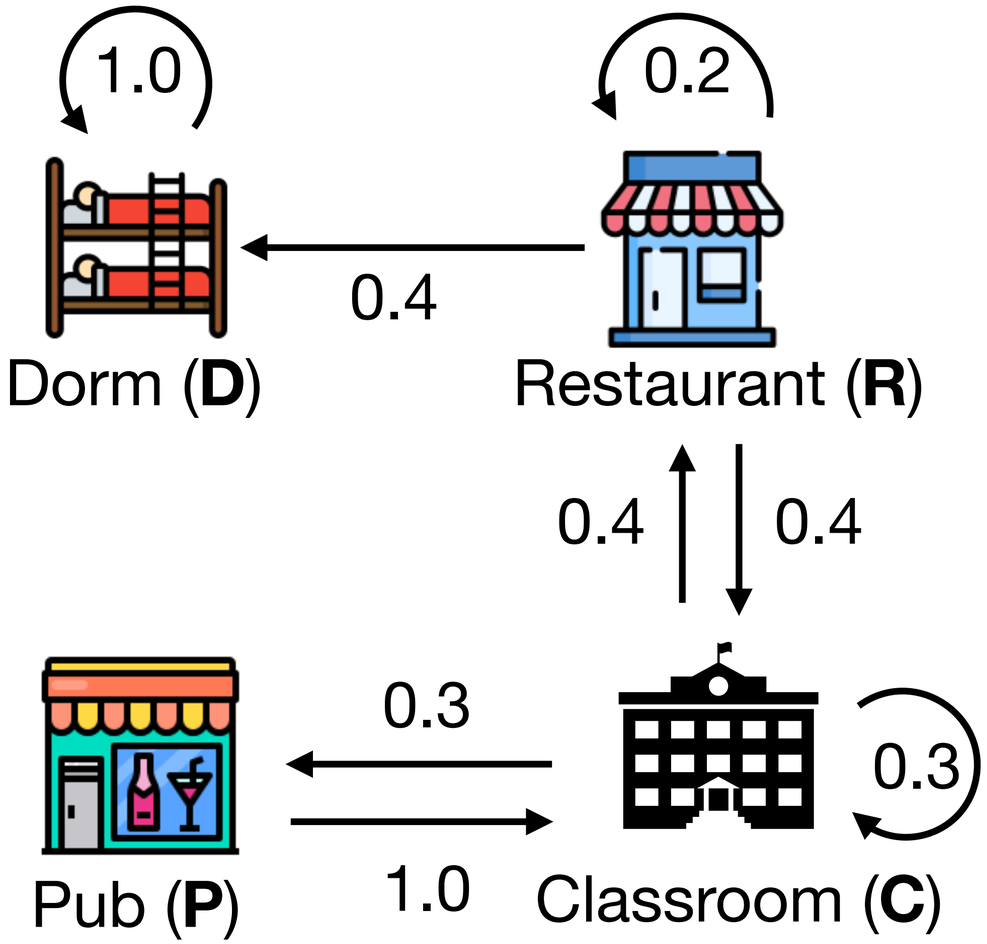
Definition

- S is a set of states
- P is a state transition probability matrix, $\forall s, s' \in S$,

  $$P_{ss'} = P(S_{t+1} = s' \mid S_t = s)$$
- A Markov can be represented as a tuple = $\{S, P\}$

Markov Process

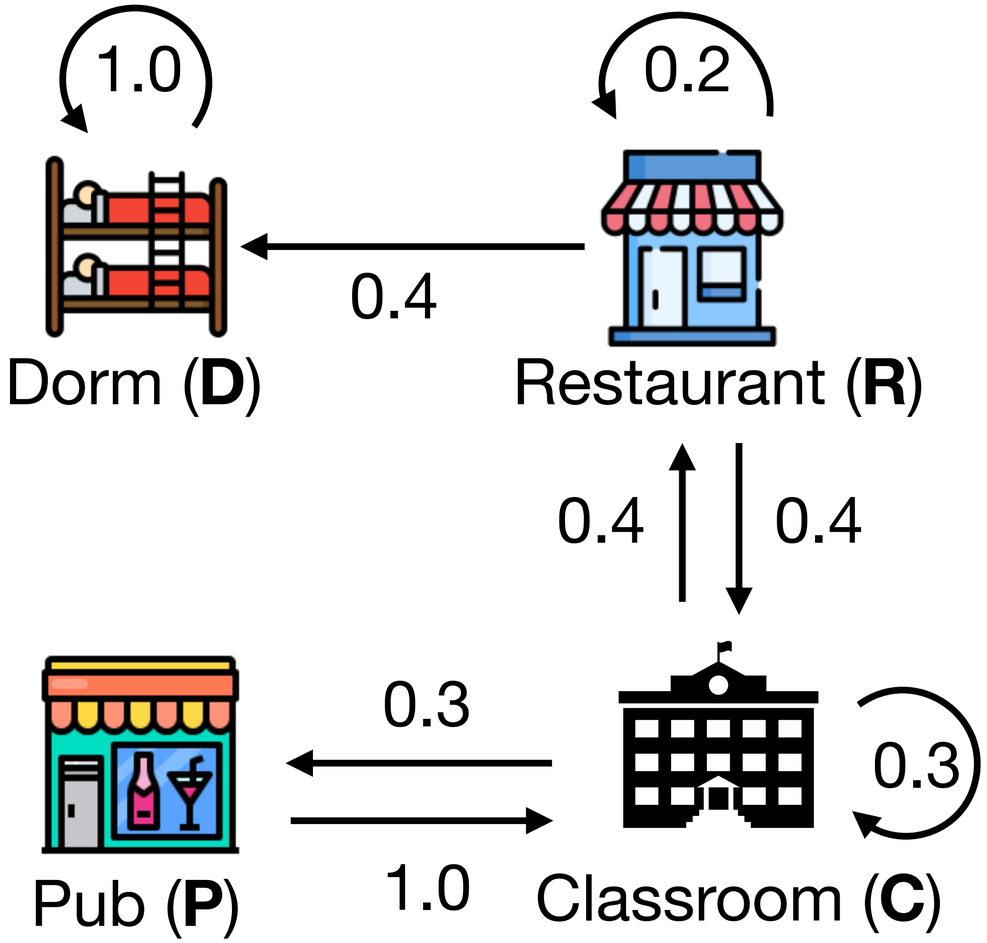
Example



- Markov Chain starting from **R**
  - Sample state transitions:
    - **R C C R D**
    - **R D D**
    - **R C C P C C R D**

Markov Process

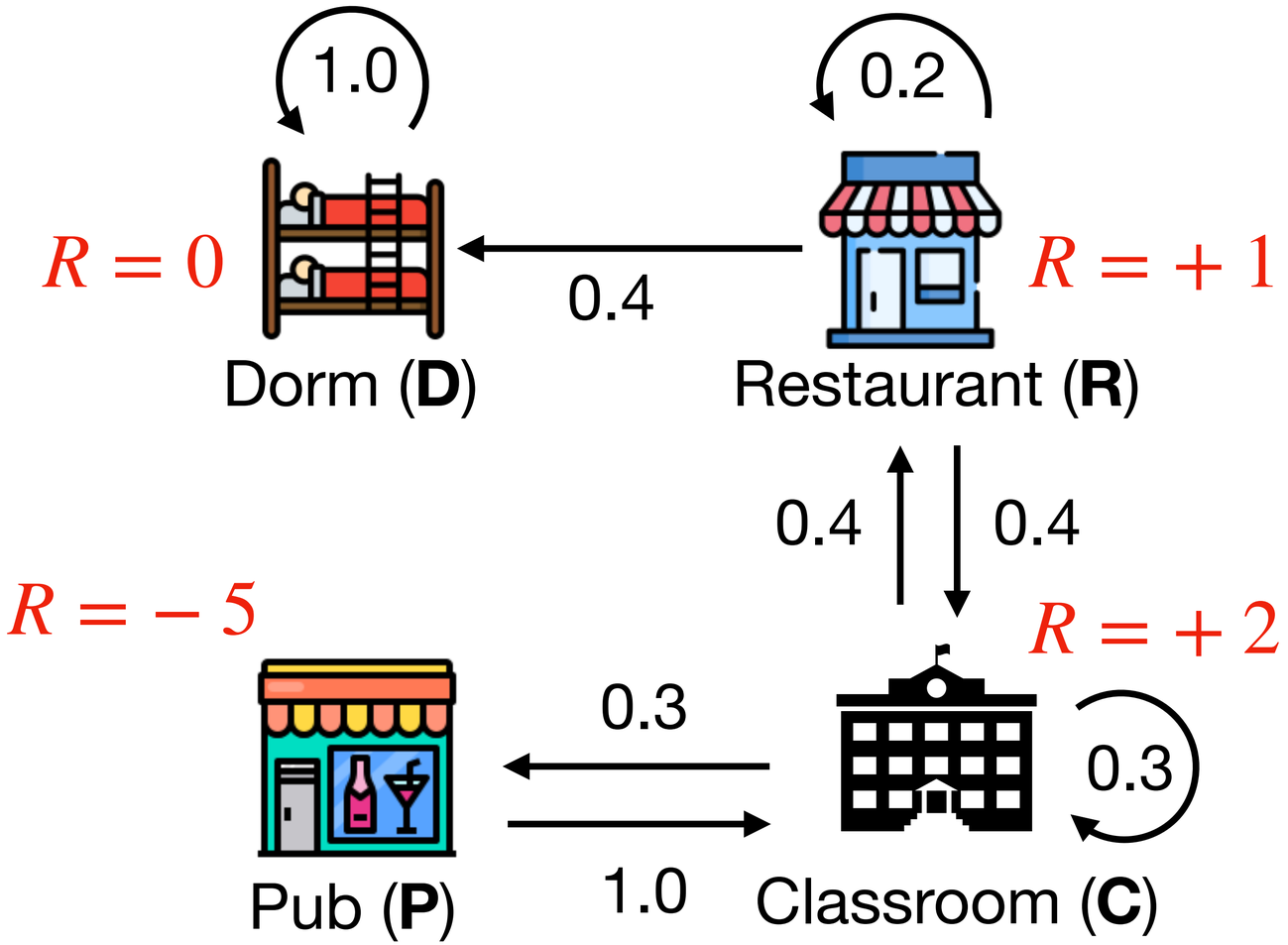
Example



$$P = \begin{array}{c} \begin{matrix} \mathbf{D} & \mathbf{R} & \mathbf{C} & \mathbf{P} \end{matrix} \\ \begin{pmatrix} 1.0 & 0 & 0 & 0 \\ 0.4 & 0.2 & 0.4 & 0 \\ 0 & 0.4 & 0.3 & 0.3 \\ 0 & 0 & 1.0 & 0 \end{pmatrix} \end{array} \begin{matrix} \mathbf{D} \\ \mathbf{R} \\ \mathbf{C} \\ \mathbf{P} \end{matrix}$$

Markov Reward Process

Example



Value Function

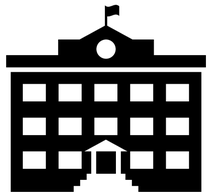
- State value function (in MRP) : the expected return starting from state s

$$V(s) = \mathbb{E}[G_t \,|\, s_t = s]$$

$$G_t = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \ldots = \sum_{i=t}^{\infty} \gamma^{i-t} R_i$$

Restaurant

$R = +1$



Classroom

$R = +2$

Outline

- Background
- Markov Process Family
- Bellman Equation
- Value Function, Q-function, and Policy

Markov Reward Process (MRP)

Definition

- A extension of Markov Processes by adding **a reward signal** to each state, and is called a Markov Reward Process (MRP)
- S is a set of states
- P is a state transition probability matrix, $\forall s, s' \in S$, $P_{ss'} = P(S_{t+1} = s' | S_t = s)$
- R denotes the reward function
- An MRP can be represented as a tuple = $\{S, P, R\}$

Decompose a Value Function (1/2)

$$
\begin{aligned}
V(s) &= \mathbb{E}[G_t \,|\, s_t = s] \\
&= \mathbb{E}[R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \dots \,|\, s_t = s] \\
&= \mathbb{E}[R_t + \gamma (R_{t+1} + \gamma^1 R_{t+2} + \dots) \,|\, s_t = s] \\
&= \mathbb{E}[R_t + \gamma (G_{t+1}) \,|\, s_t = s] \\
&= \mathbb{E}[R_t + \gamma V(s_{t+1}) \,|\, s_t = s]
\end{aligned}
$$

Decompose a Value Function (2/2)

$$\mathbb{E}[R_t + \gamma(G_{t+1}) \,|\, s_t = s]$$

$$= \mathbb{E}[R_t \,|\, s_t = s] + \mathbb{E}[\gamma(G_{t+1}) \,|\, s_t = s]$$

Law of Total Expectation

$E[X|Y]=E[E[X|Y,Z]|Y]$

$$= \mathbb{E}[R_t \,|\, s_t = s] + \gamma\mathbb{E}[\mathbb{E}[G_{t+1} \,|\, s_t = s, s_{t+1} = s'] \,|\, s_t = s]$$

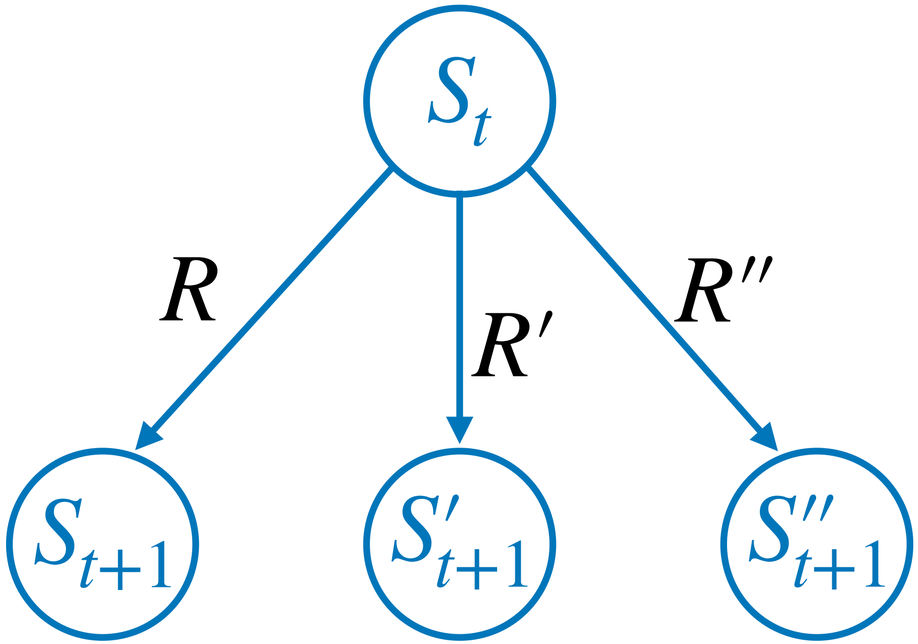
$$= \mathbb{E}[R_t \,|\, s_t = s] + \gamma\mathbb{E}[\mathbb{E}[G_{t+1} \,|\, s_{t+1} = s'] \,|\, s_t = s]$$

$$= \mathbb{E}[R_t \,|\, s_t = s] + \mathbb{E}[\gamma(V(s_{t+1})] \,|\, s_t = s]$$

$$= \mathbb{E}[R_t + \gamma V(s_{t+1}) \,|\, s_t = s]$$

Bellman Equation

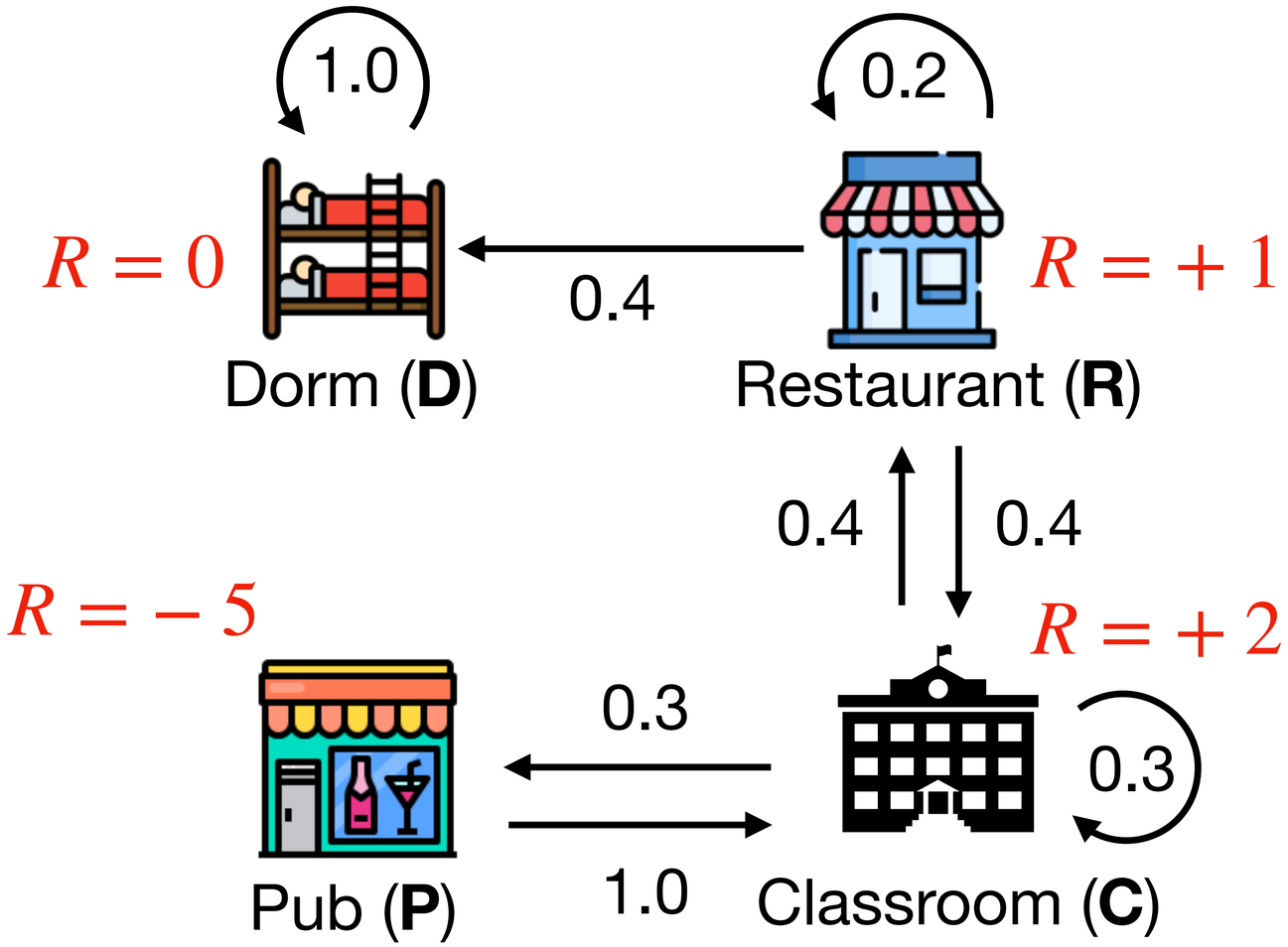
- Decompose the value function into two parts: the immediate reward and the discounted future value.

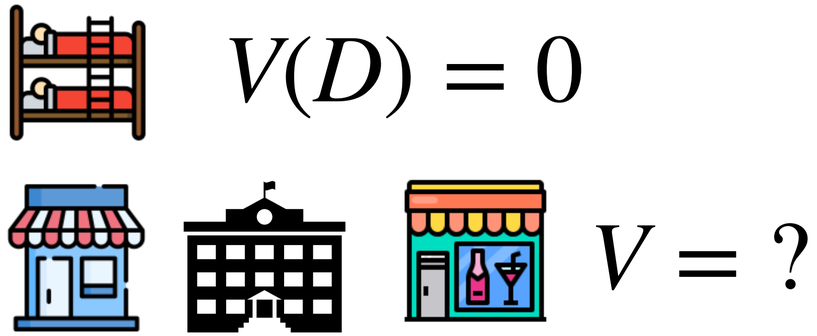


$$V(s) = \mathbb{E}[R_t + \gamma V(s_{t+1}) \,|\, s_t = s]$$

Value Function

Value Function of a State

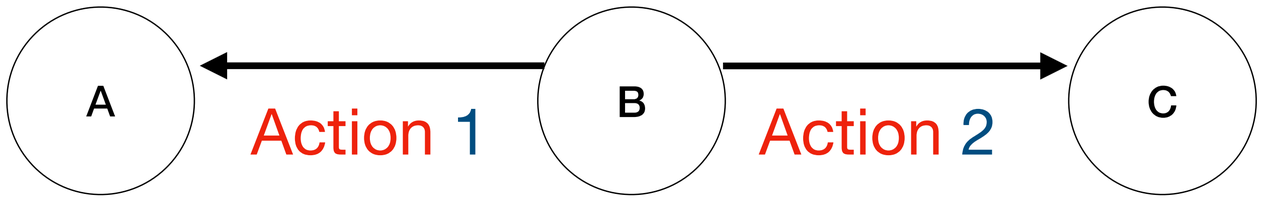




- Value function can be derived by:
  - Dynamic programming
  - Monte-Carlo evaluation
  - TD learning
- The above methods will be introduced in the later courses

Markov Decision Processes (MDP)

- An extension to the Markov Reward Processes with decisions (i.e., actions)



- RL problems are usually formulated as an MDP
- The state transition probability is defined as follows:

$$P_{ss'} = P(S_{t+1} = s' \,|\, S_t = s, A_t = a)$$

Markov Decision Processes (MDP)

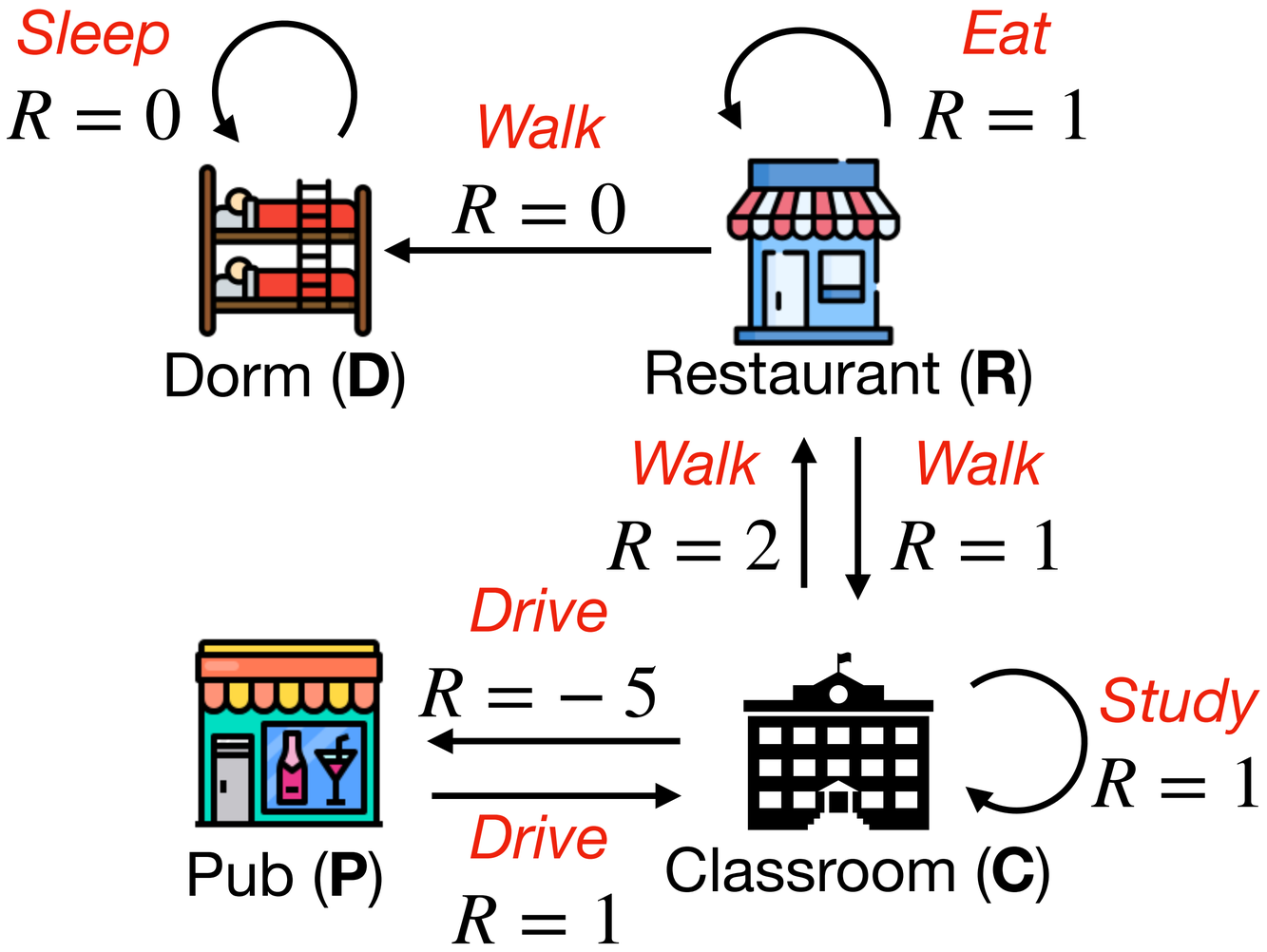
Definition

- S is a set of states
- P is a state transition probability matrix, $\forall s, s' \in S$,

  $$P_{ss'} = P(S_{t+1} = s' \mid S_t = s, A_t = a)$$
- R denotes the reward function
- A is a finite set of actions
- An MDP can be represented as a tuple = $\{S, P, R, A\}$

Markov Decision Process

Example



Outline

- Background
- Markov Process Family
- Bellman Equation
- Value Function, Q-function, and Policy

Policy

Definition

- Policy π : A probability distribution over actions given states, expressed as:

$$\pi(a \mid s) = P[A_t = a \mid S_t = s]$$

- It represents the behavior of an agent
- It only depends on the current state

State Value Function

Definition

- State value function (in MDP): the expected return starting from state s **following policy** π

$$V_\pi(s) = \mathbb{E}_\pi[G_t \,|\, s_t = s]$$

- It is also usually called the **Value function**

State-Action Value Function

Definition

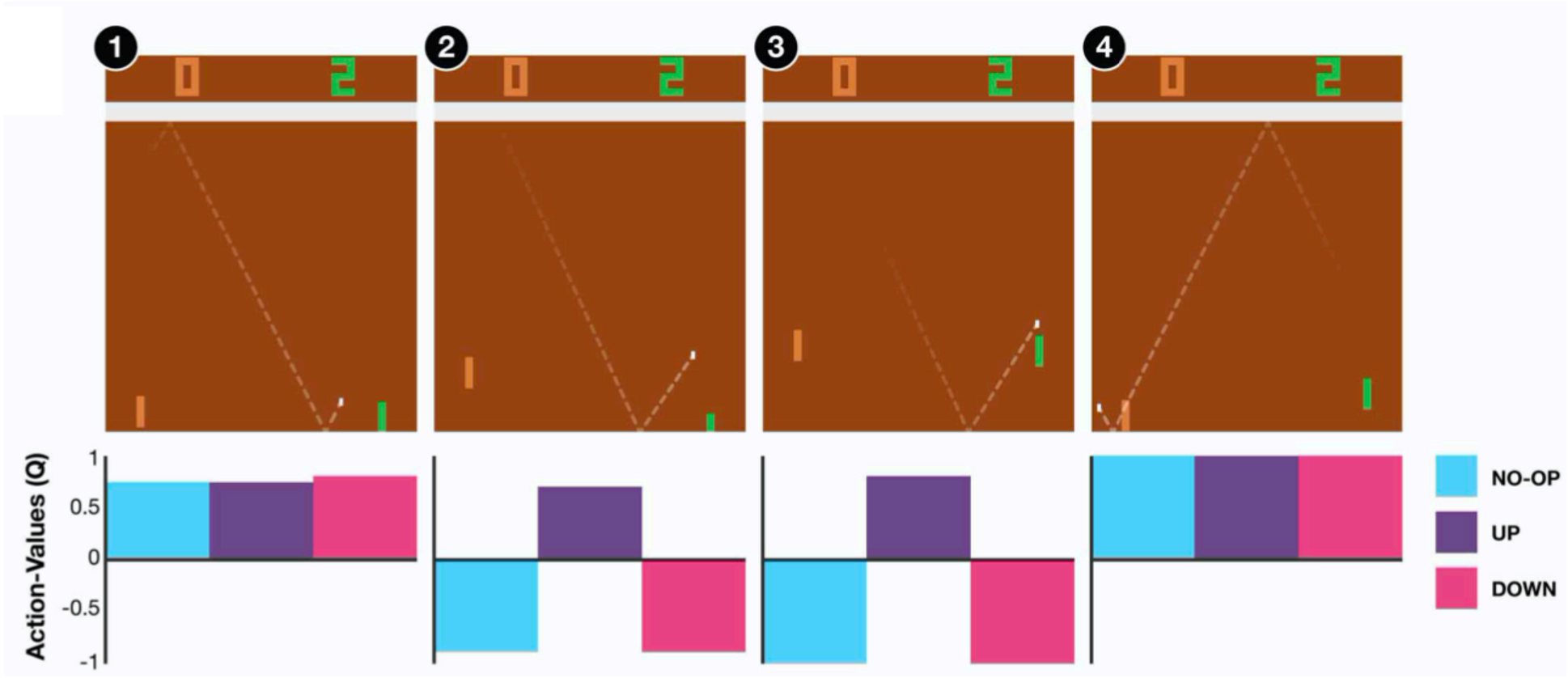
- State-Action value function: the expected return starting from state s, **taking action a,** and following policy π

$$Q_\pi(s, a) = \mathbb{E}_\pi[G_t \,|\, s_t = s, A_t = a]$$

- It is also usually called the **Q function**

State-Action Value Function

Sample Images



State Value Function

State Value Function

$$
\begin{aligned}
V_\pi(s) &= \mathbb{E}_\pi[G_t \,|\, s_t = s] \\
&= \mathbb{E}_\pi[R_t + \gamma(G_{t+1}) \,|\, s_t = s] \\
&= \sum_a \pi(a \,|\, s) \sum_{s'} p(s' \,|\, s, a)[R_t + \gamma \mathbb{E}[G_{t+1} \,|\, s_{t+1} = s']] \\
&= \sum_a \pi(a \,|\, s) \sum_{s'} p(s' \,|\, s, a)[R_t + \gamma V_\pi(s')]
\end{aligned}
$$

Optimal Value Function

Definition

- Optimal State Value Function $V_*(s)$: Maximum value function over all policies:

$$V_*(s) = \max_\pi V_\pi(s)$$

- Optimal Q Function $Q_*(s, a)$: Maximum Q function over all policies:

$$Q_*(s, a) = \max_\pi Q_\pi(s, a)$$

Optimal Policy

Definition

- Optimal Policy π_* : Its expected return is greater than or equal to other policies for all states

$$\pi_* \geq \pi' \quad if \quad V_{\pi_*}(s) \geq V_{\pi'}(s), \forall s \in S$$

Optimal Policy

Definition

- If you find the optimal policy, you solve the “MDP” problem
- There may exist more than one optimal policy
- Abbreviation:
  - $V_{\pi_*}(s) = V_*(s)$
  - $Q_{\pi_*}(s, a) = Q_*(s, a)$

Partially Observable MDP (POMDP)

Example — Fog of War

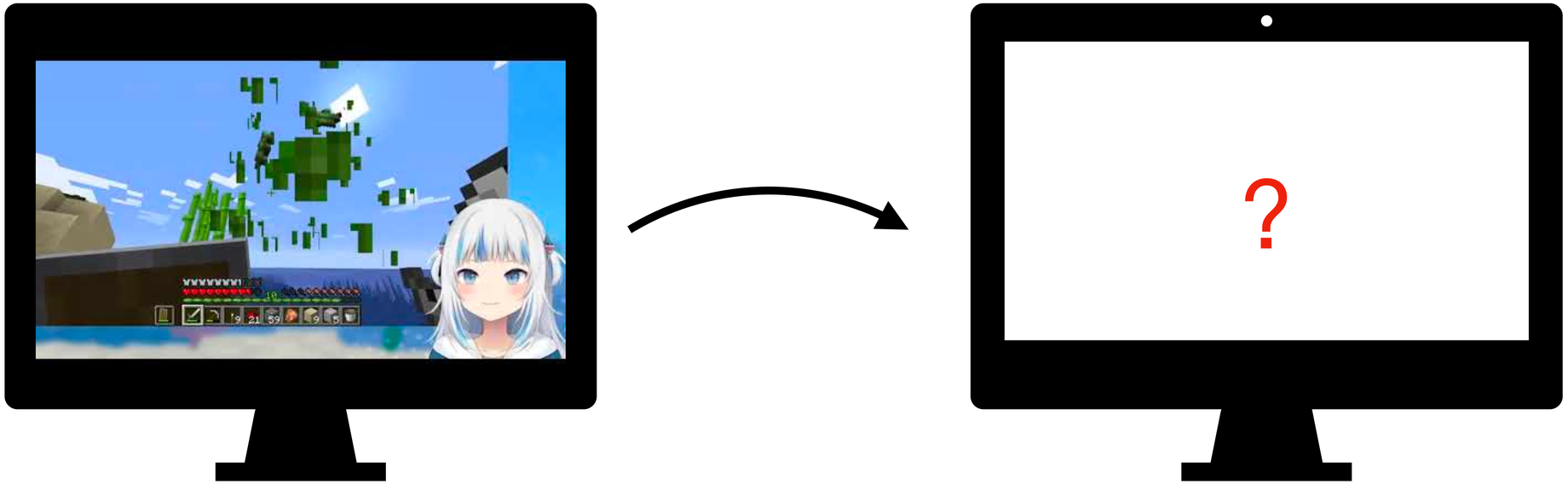
Partially Observable MDP (POMDP)

Definition

- S is a set of states
- O is a set of observations
- P is a state transition probability matrix, $\forall s, s' \in S$, $P_{ss'} = P(S_{t+1} = s' \mid S_t = s)$
- R denotes the reward function
- A is a finite set of actions
- Z is an observation function, $Z^a_{s'o} = P(O_{t+1} = o \mid S_{t+1} = s', A_t = a)$
- A POMDP can be represented as a tuple = $\{S, P, R, A, O, Z\}$

Uncertain Environment

Noise TV - Showing a random image



We cannot not know
the next state

Uncertain Environment

Noise TV - Showing a random image

Environment without any noise TV

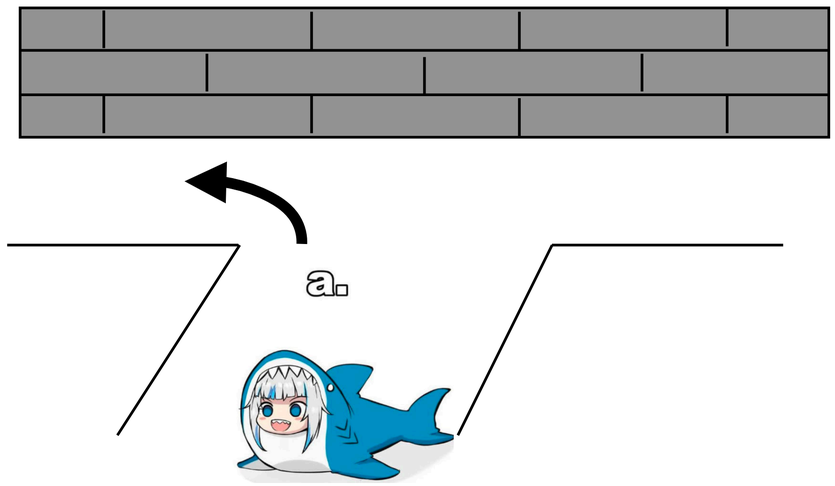
Environment with a noise TV

Deterministic Policy

Definition

- A policy outputs actions directly

$$a = \pi(s)$$

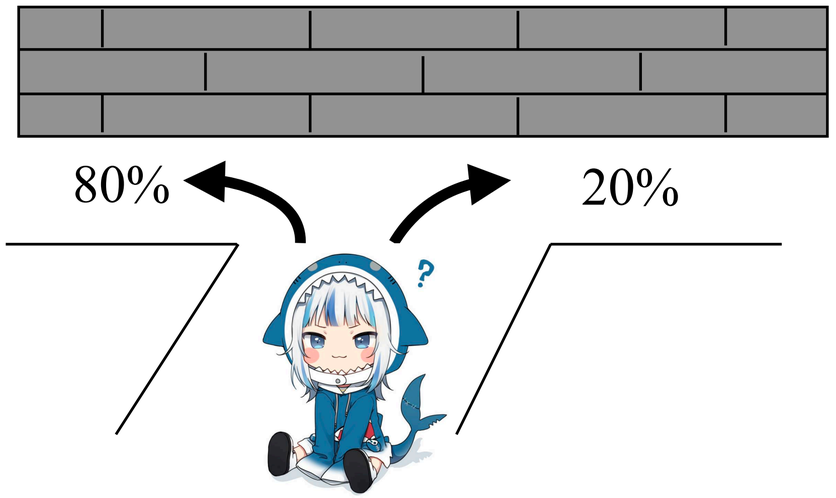


Stochastic Policy

Definition

- A policy outputs a probability distribution over actions

$$a \sim \pi(a \mid s) = \mathbb{P}[A_t = a \mid S_t = s]$$



Stochastic Policy

Why

- It is beneficial for an agent to explore the state space without always taking the same action
- Stochastic policies are usually used in uncertain environments
- They can obtain better performance than deterministic policies in certain conditions


THANK
YOU !