What is Reinforcement Learning

About Me

Witthawin Sripheanpol (Ro)

B.Eng. Institute of Field Robotics (**FIBO**),
King Mongkut's University of Technology Thonburi

Project Manager & Lead Lecturer,
iGenius Robot Education

Data Specialist Consultant,
iDevfinite Solutions Co., Ltd.

Head Finetuning InstructGPT & RLHF Volunteer,
OpenThaiGPT

SuperAI Engineer Season4 : Silver Medal

Lead Senior AI Engineer,
Botnoi Group

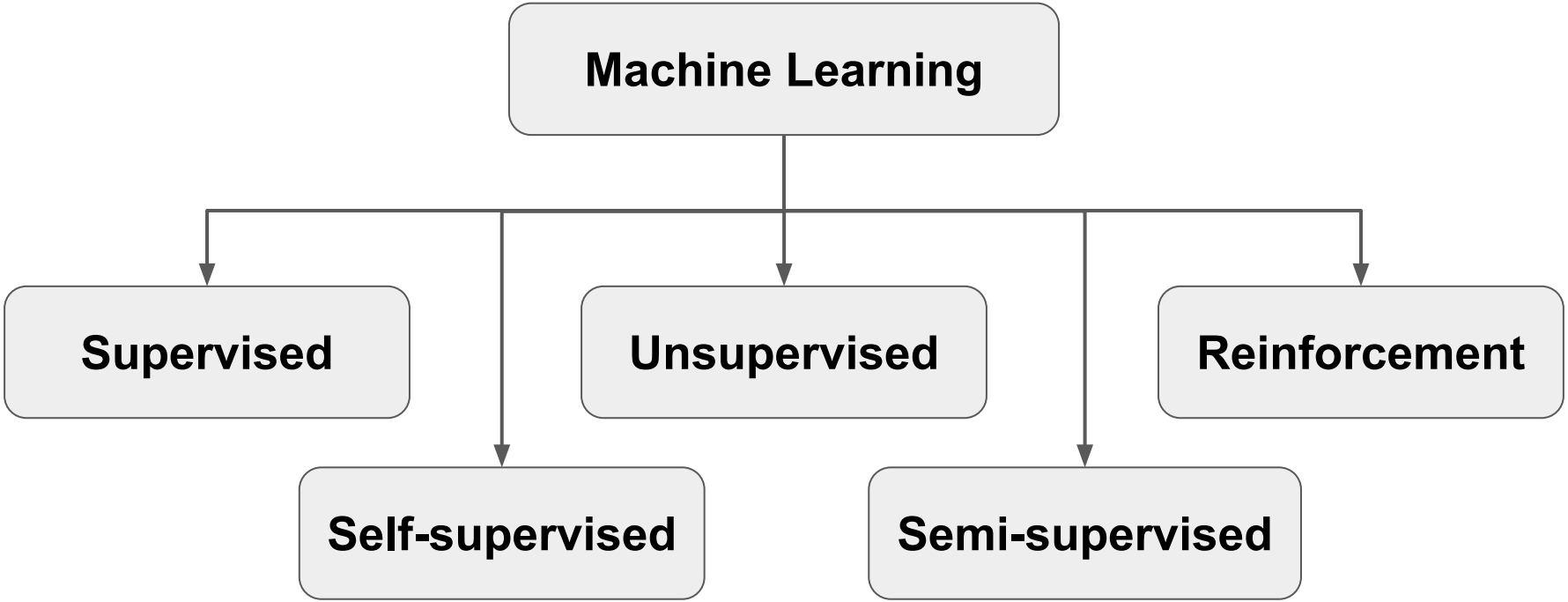
AI Researcher,
Chula-AI

AI & Data Solution Specialist,
Edvisory Co., Ltd.

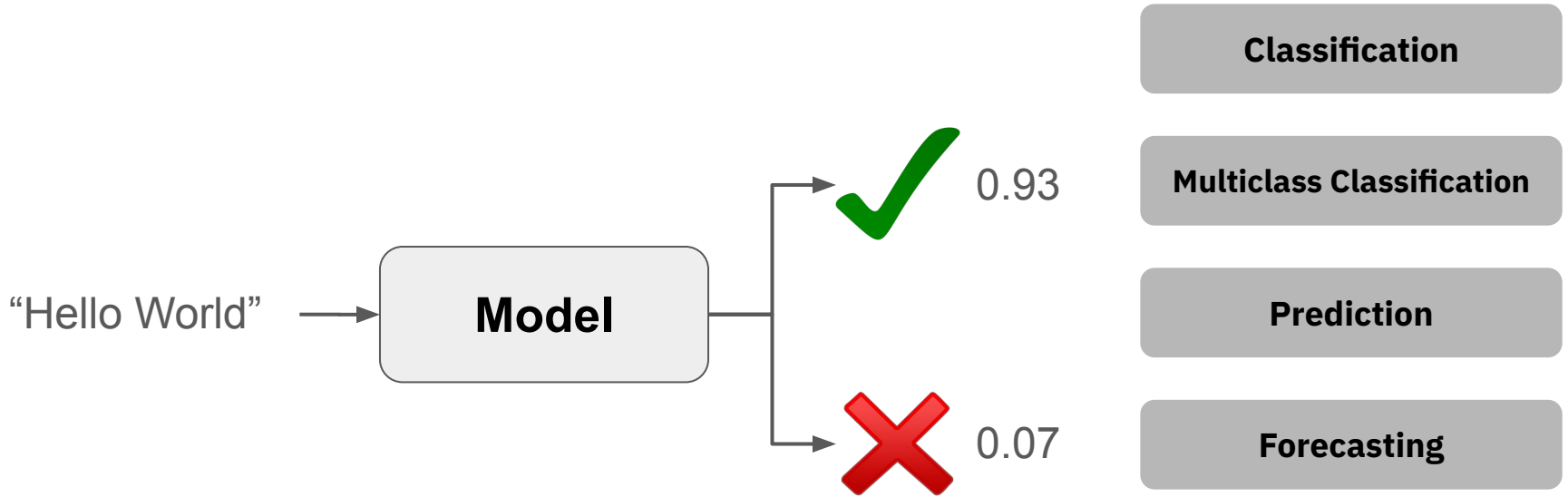
Agenda

- Recap Machine Learning
  - Supervised Learning
  - Unsupervised Learning
  - Reinforcement Learning
- Overview Agent - Environment
- How Agent take action ?
- Value Function & Q Learning
- Next step : Policy Optimization, Actor-Critic

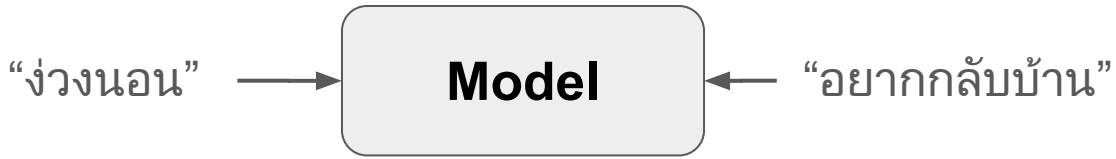
Recap Machine Learning



Supervised Learning

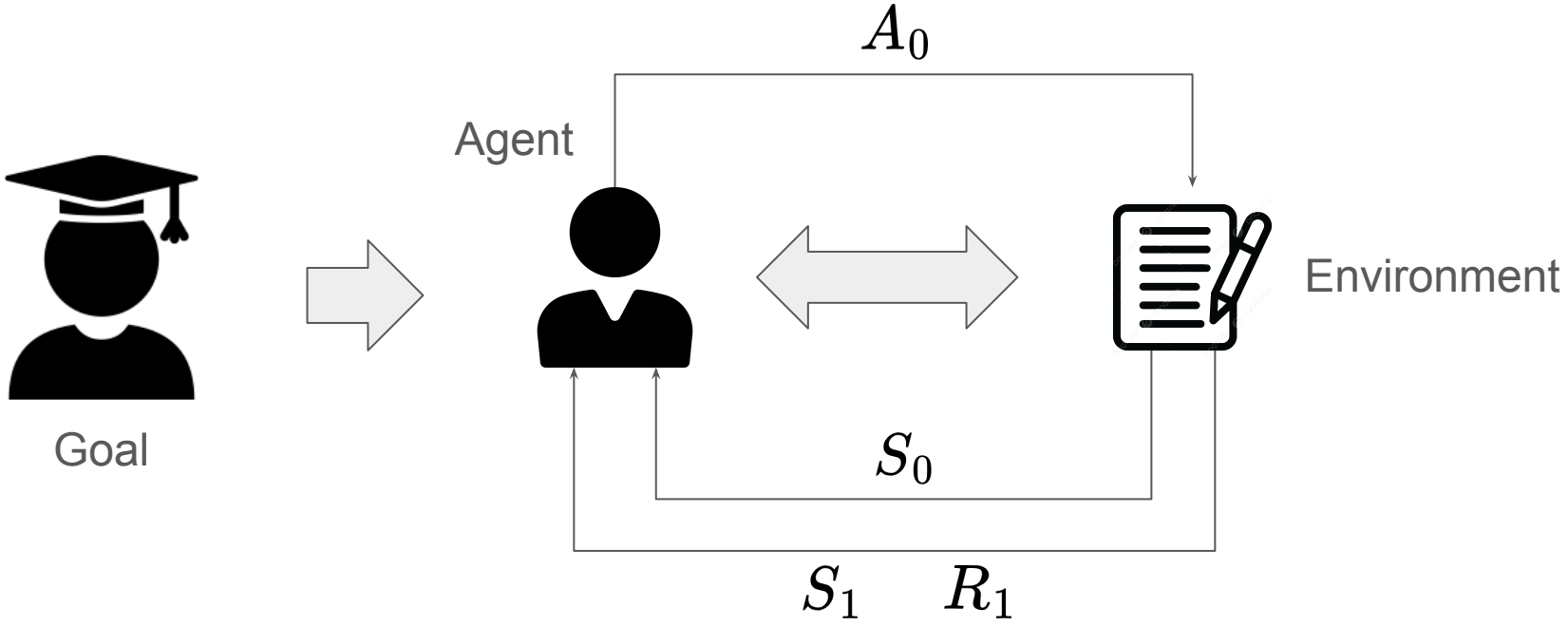


Unsupervised Learning



- **Association Rule**
- **Clustering**
- **Recommendation**
- **Dimension Reduction**

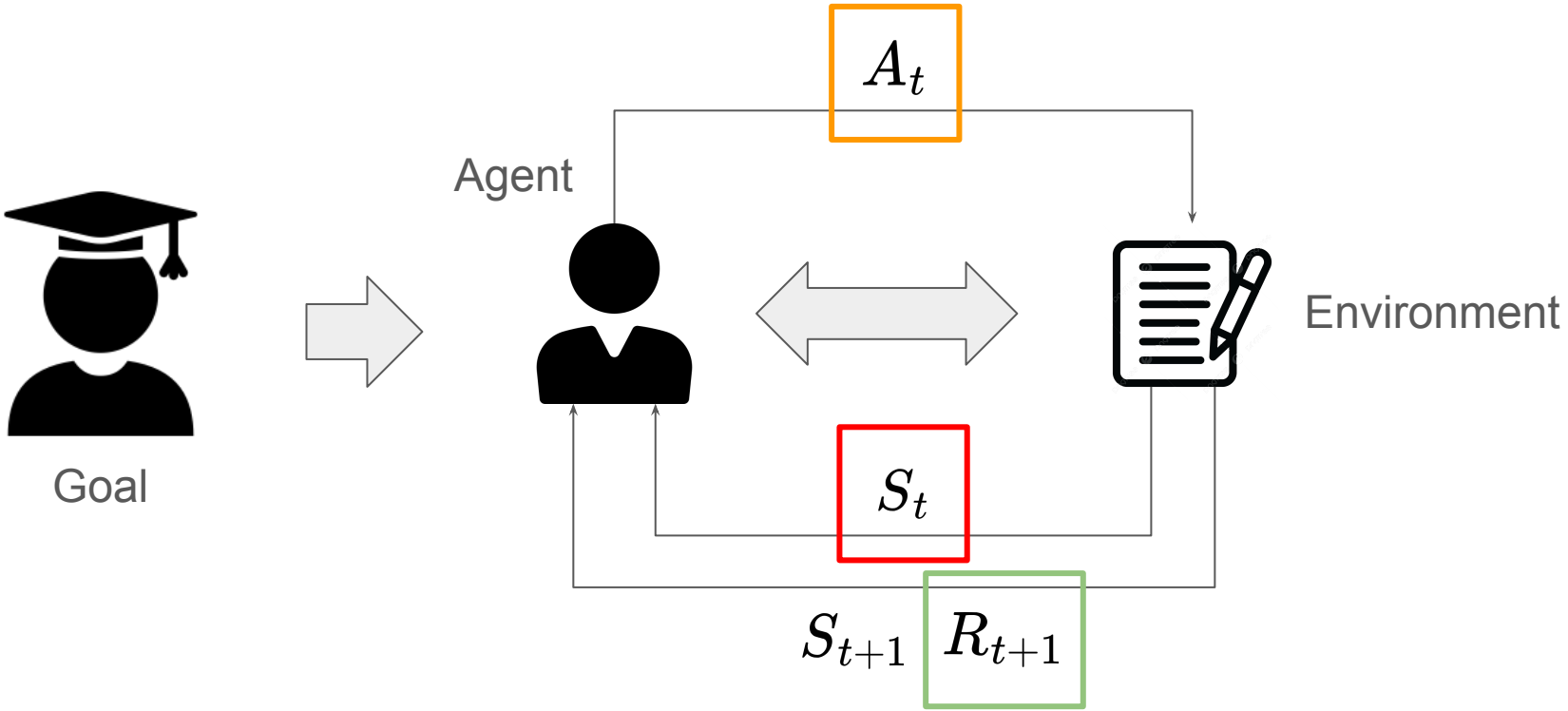
Reinforcement Learning



Objective

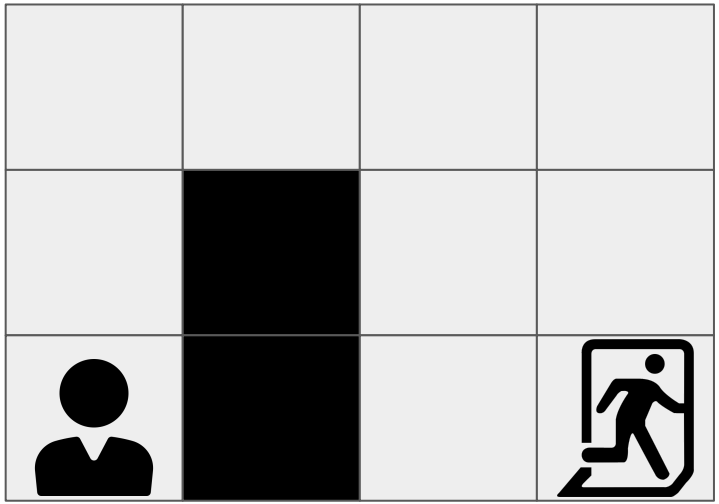
- **Maximize Cumulative Reward** – The primary goal in RL is to learn a policy that maximizes the expected cumulative reward over time.
- **Optimize Long-Term Value** – The agent must balance short-term and long-term rewards to achieve an optimal strategy.
- **Learn an Optimal Policy** – The goal is to find the best mapping from states to actions that results in the highest expected reward.
- **Explore and Exploit Efficiently** – The agent needs to balance exploration (trying new actions) and exploitation (choosing known good actions).
- **Handle Uncertainty and Partial Observability** – In many environments, the agent must learn under incomplete information (POMDP settings).

Reinforcement Learning

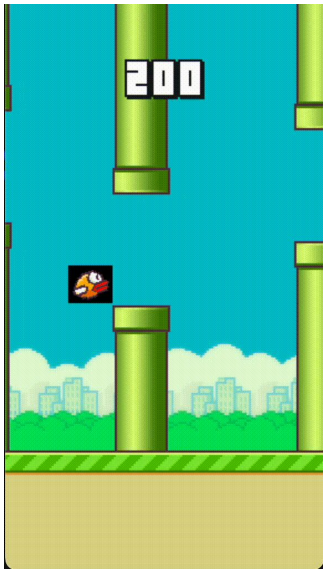


States/Observations Space S_t

States

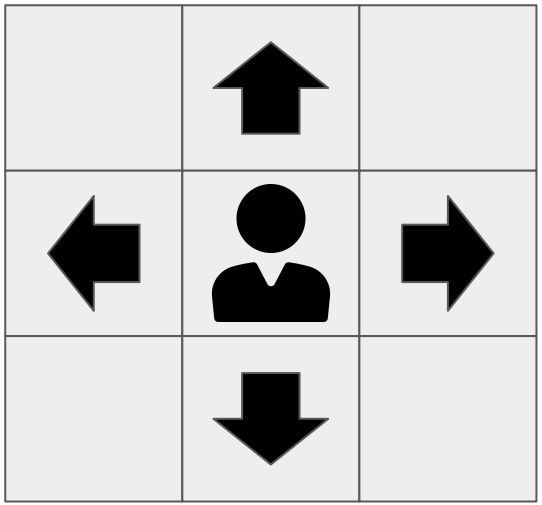


Observation



Action Space A_t

Discrete

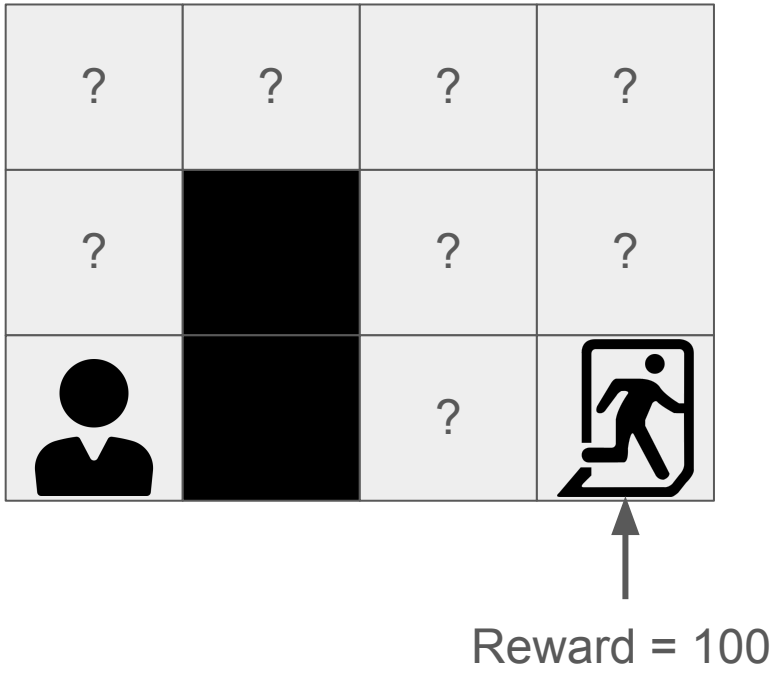


Up / Down / Left / Right

Continuous

Degree
(Number)

Rewards R_{t+1}



Rewards R_{t+1}

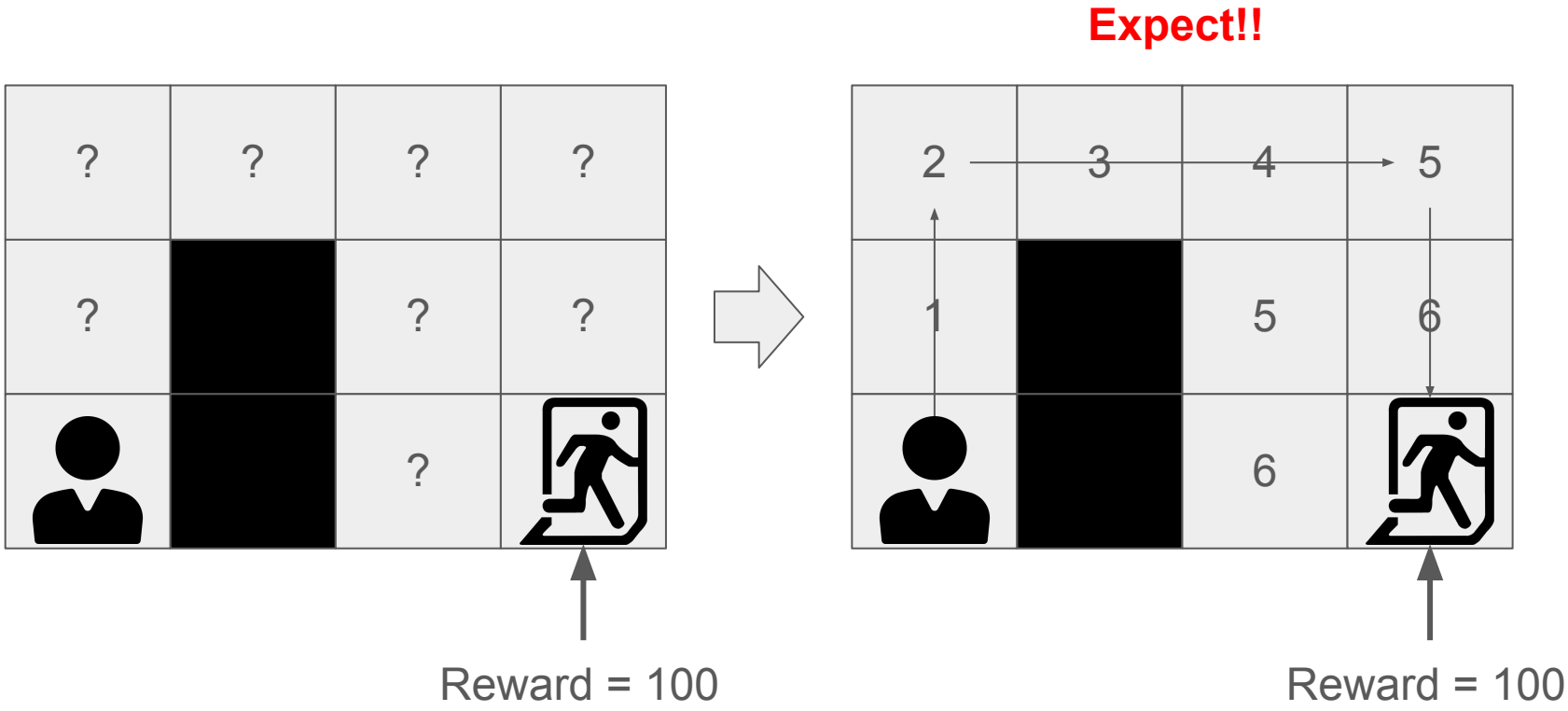
$$R(\tau) = r_{t+1} + r_{t+2} + r_{t+3} + r_{t+4} + \dots$$

Return: cumulative reward

Trajectory (read Tau)
Sequence of states and actions

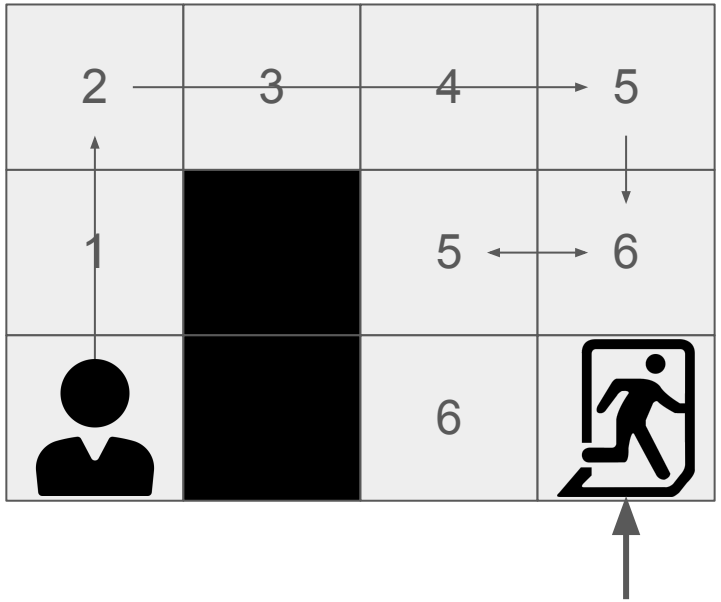
$$R(\tau) = \sum_{k=0}^{\infty} r_{t+k+1}$$

Rewards R_{t+1}



Rewards R_{t+1}

Actuality!!



$$R(\tau) = \sum_{k=0}^{\infty} r_{t+k+1}$$

T=1 -> Reward = 1

T=2 -> Reward = 1+2

T=3 -> Reward = 1+2+3

. . .

T=10 -> Reward = 1+2+3…+5+6+5+6

Rewards and the discounting

$$R(\tau) = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \gamma^3 r_{t+4} + \dots$$

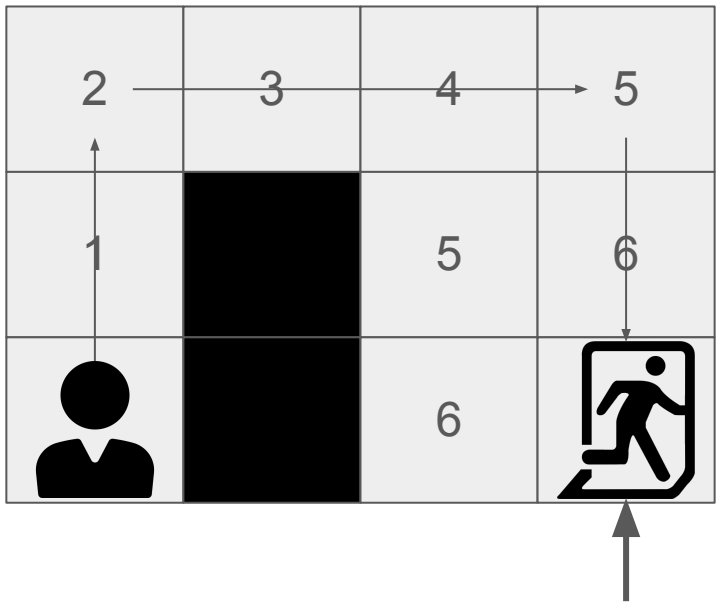
Return: cumulative reward

Gamma: discount rate

Trajectory (read Tau)
Sequence of states and actions

$$R(\tau) = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$$

Rewards and the discounting



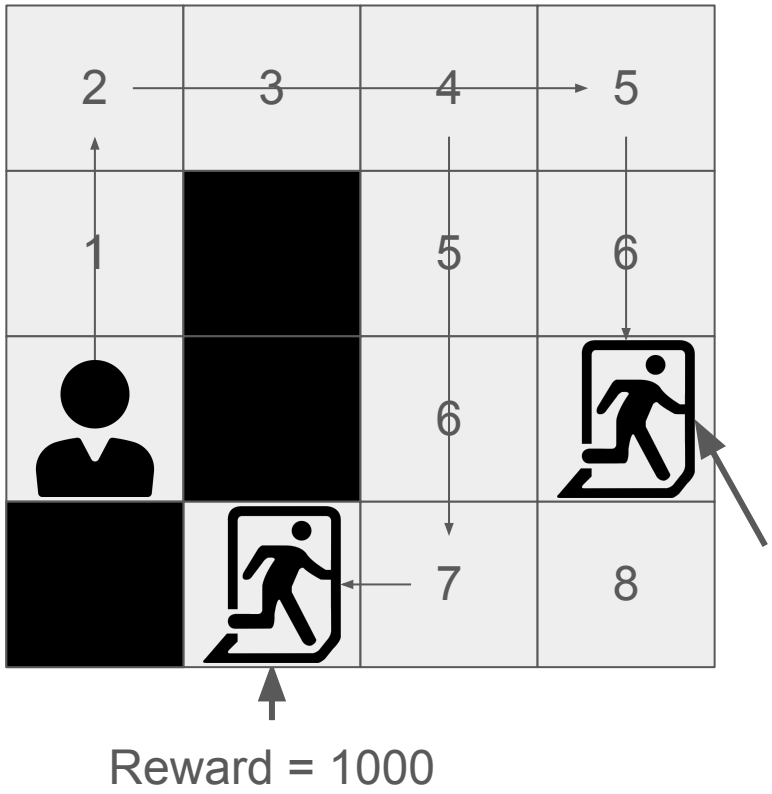
$$R(\tau) = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$$

T=1 -> Reward = 1*0.9

T=2 -> Reward = (1*0.9)+(2*0.9^2)

T=3 -> Reward = (1*0.9)+(2*0.9^2)+(3*0.9^3)

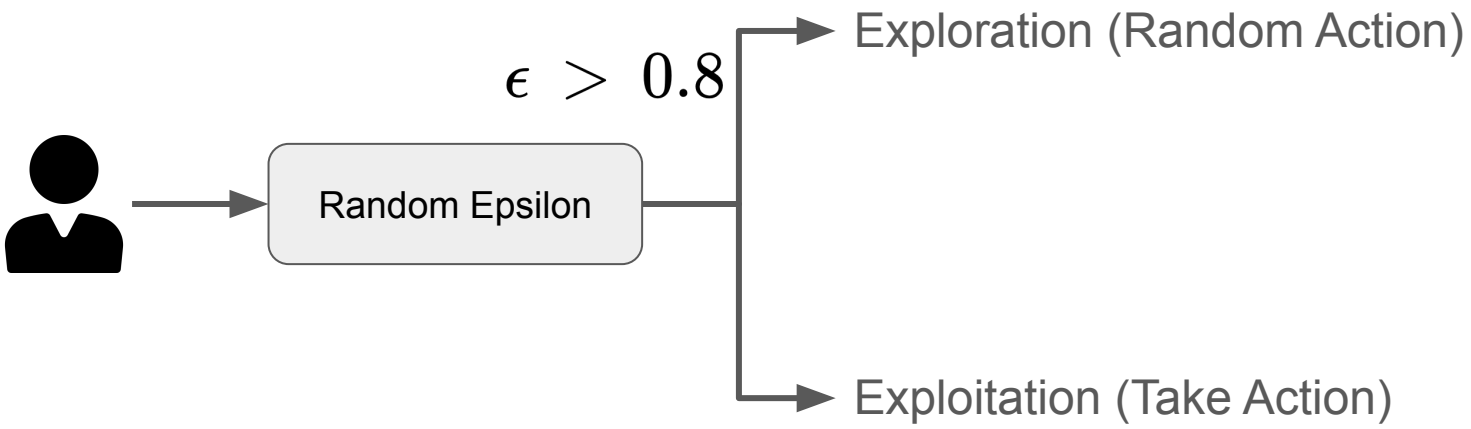
Exploration / Exploitation

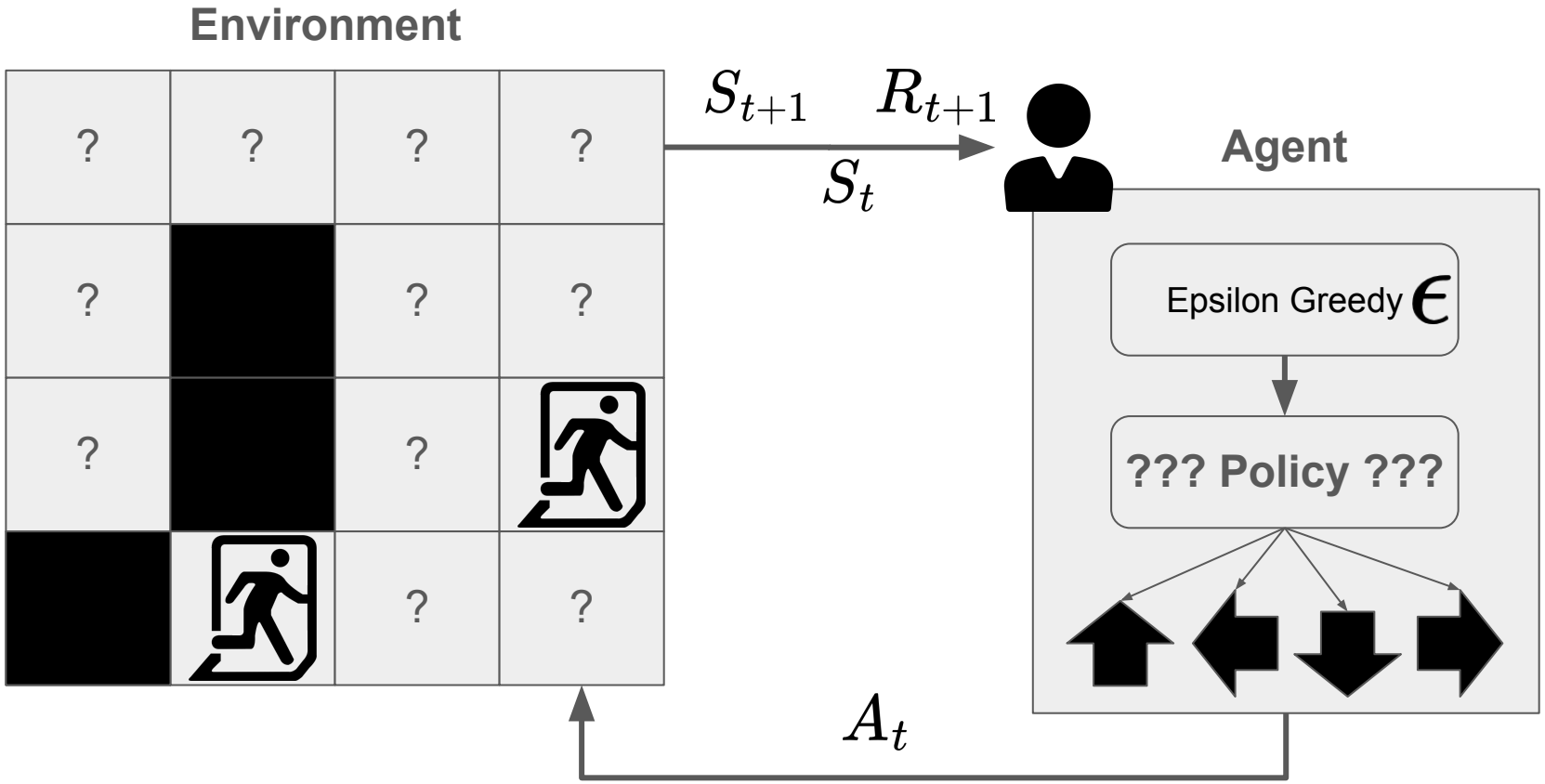


Exploration (Random Action)
Exploitation (Take Action)

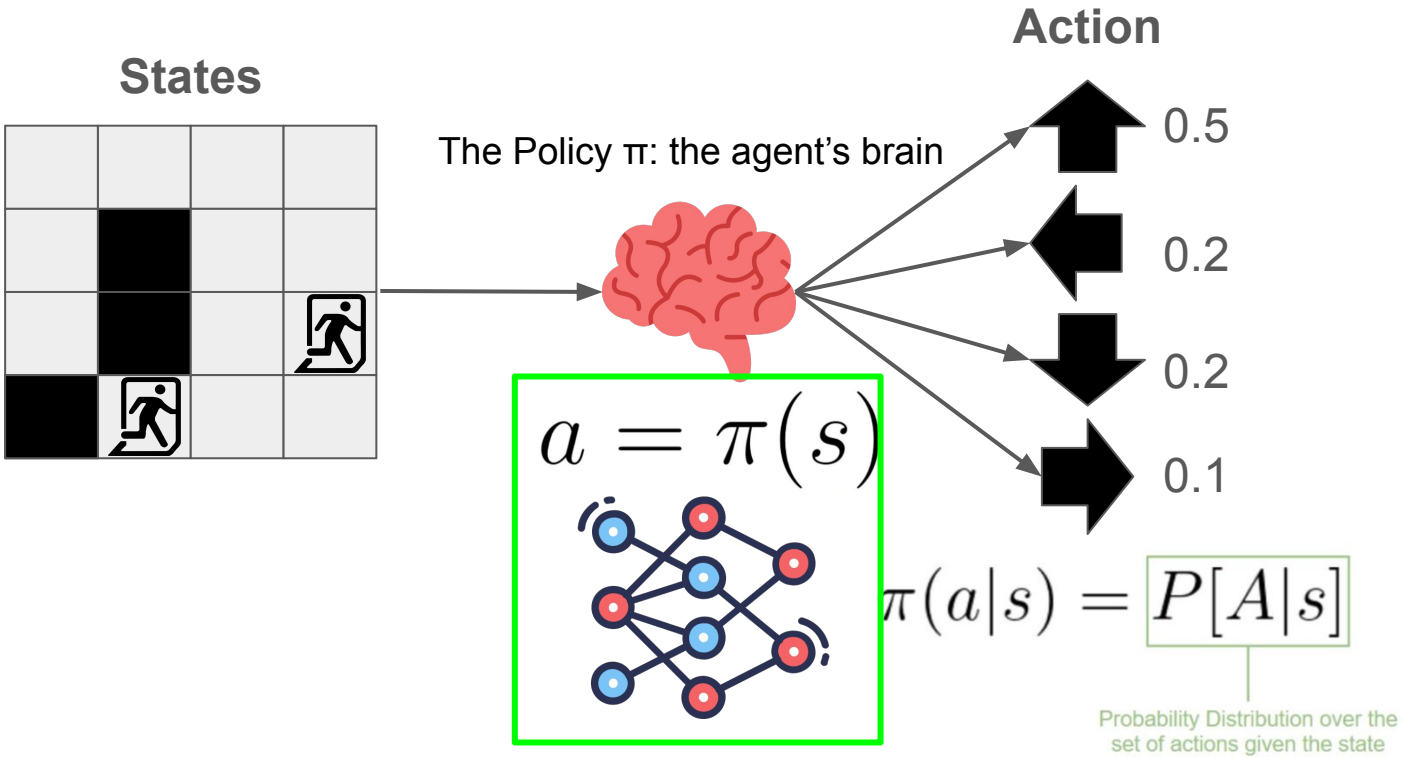
“Epsilon Greedy”

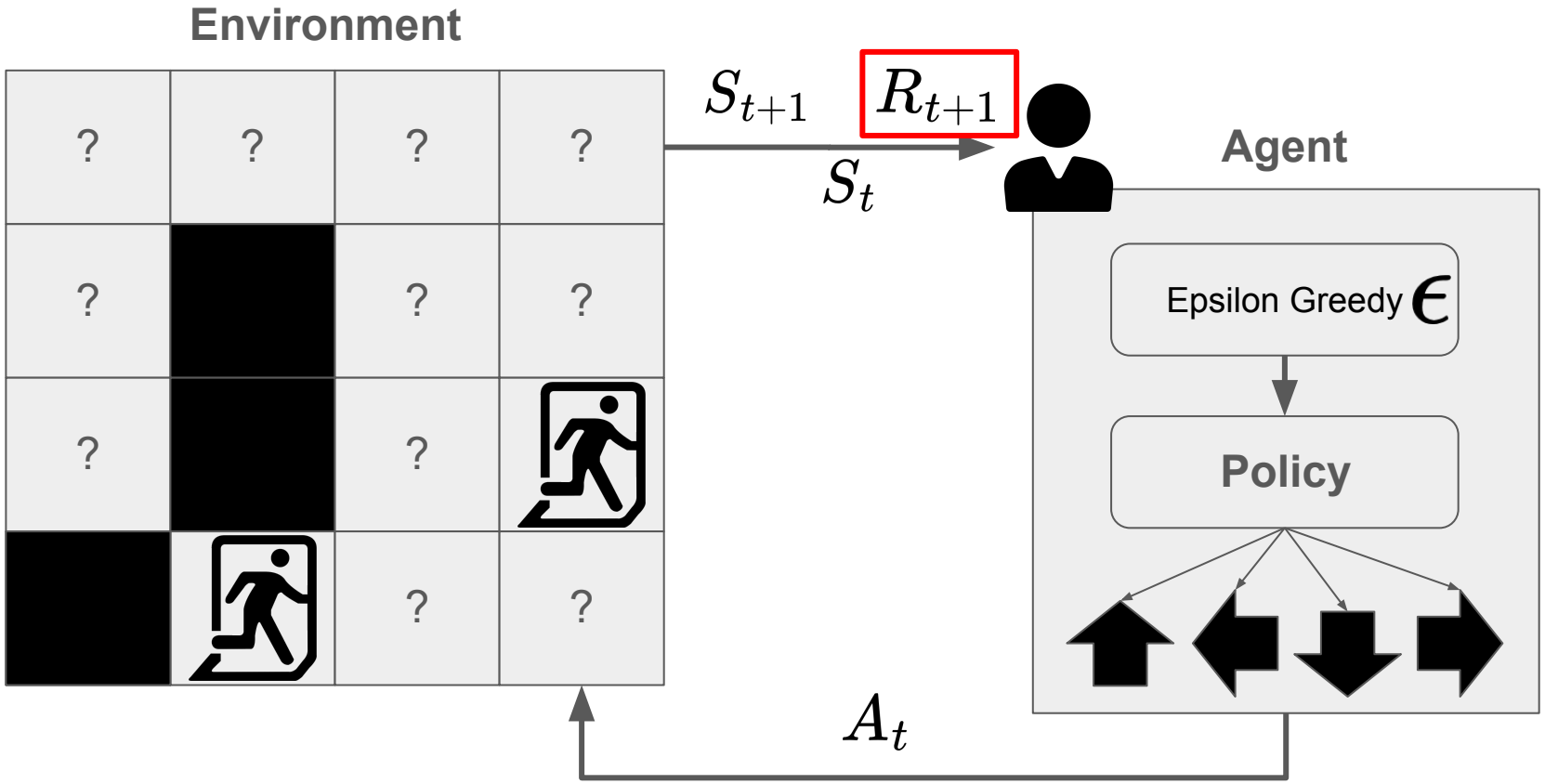
Epsilon Greedy ϵ



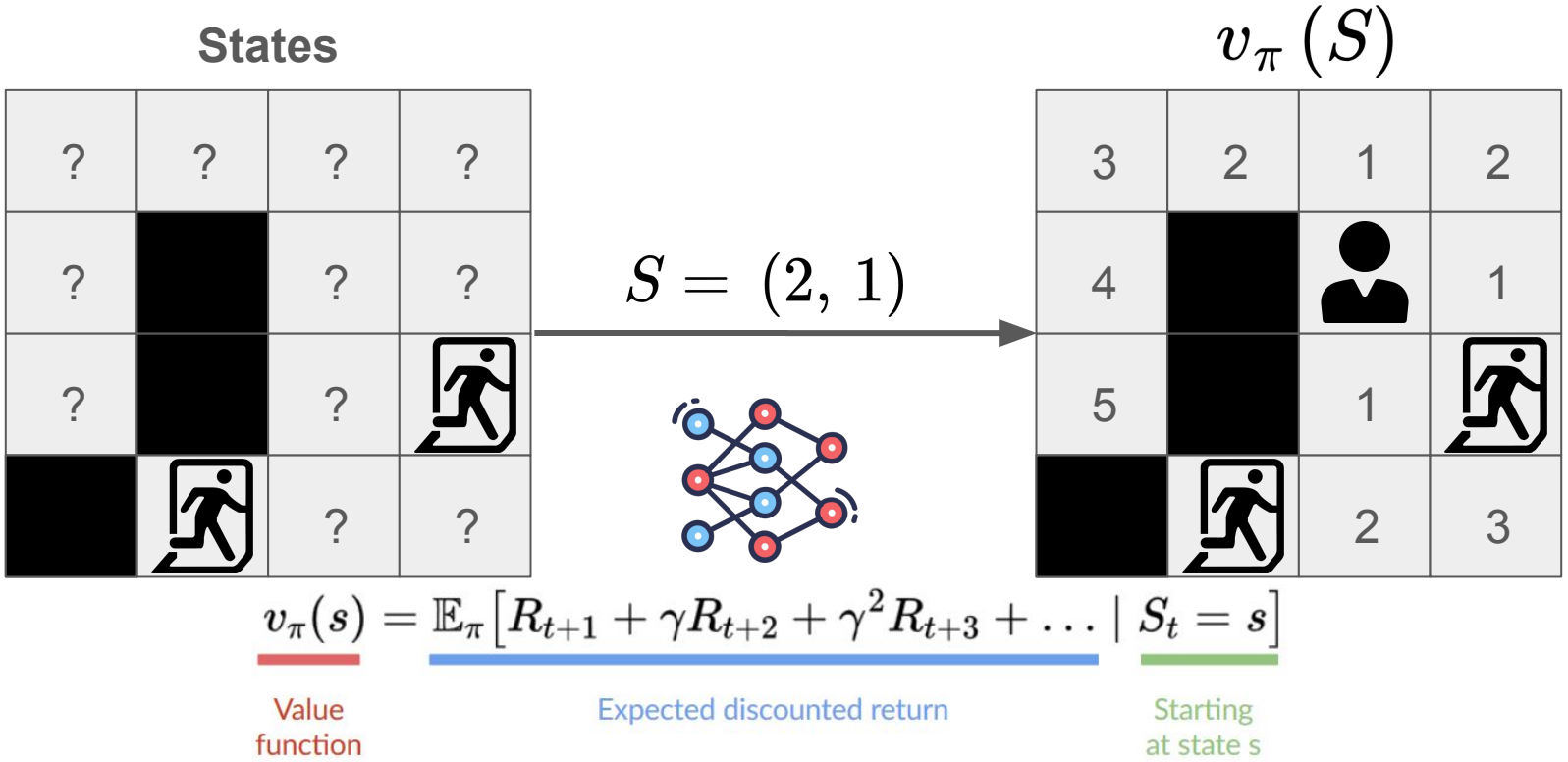

Environment
? ? ? ?
? ? ? ?
? ? ? ?
? ? ?
S_{t+1}
R_{t+1}
S_t
Agent
Epsilon Greedy ϵ
??? Policy ???
A_t

Policy $a = \pi(s)$


States
Action
The Policy π: the agent's brain
0.5
0.2
0.2
0.1
$a = \pi(s)$
$\pi(a|s) = P[A|s]$
Probability Distribution over the set of actions given the state


Environment
?
?
?
?
?
?
?
?
?
?
?
?
?
?
S_{t+1}
R_{t+1}
S_t
Agent
Epsilon Greedy ϵ
Policy
A_t

Value Function $v_\pi(S)$



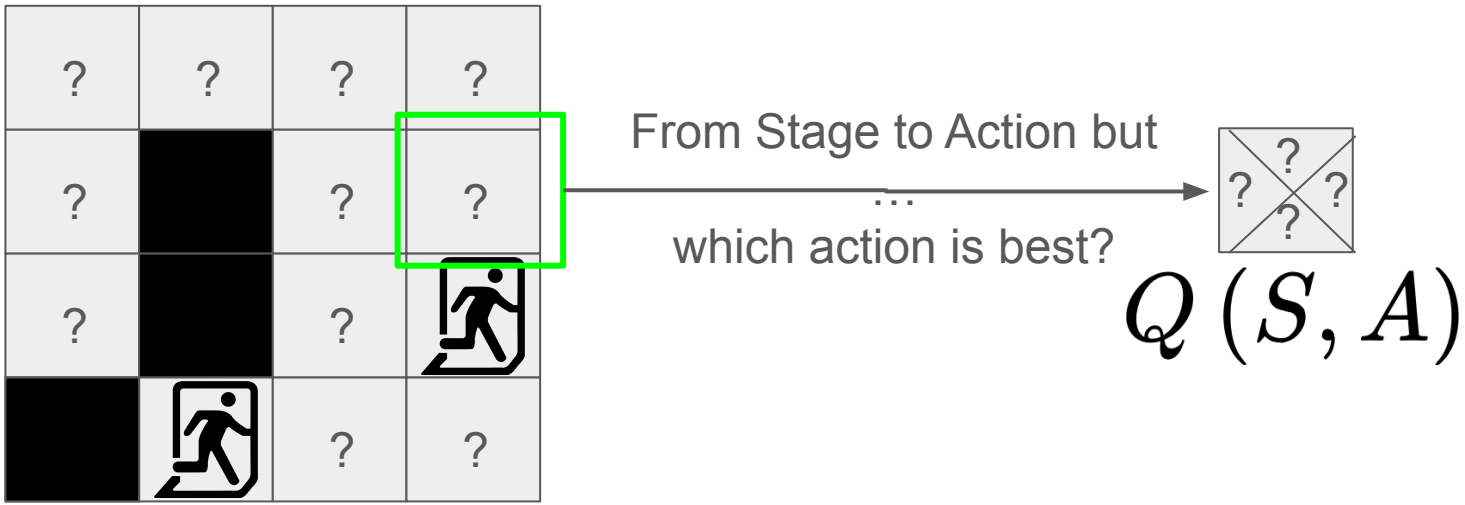
$$v_\pi(s) = \mathbb{E}_\pi\left[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots \mid S_t = s\right]$$

Value function

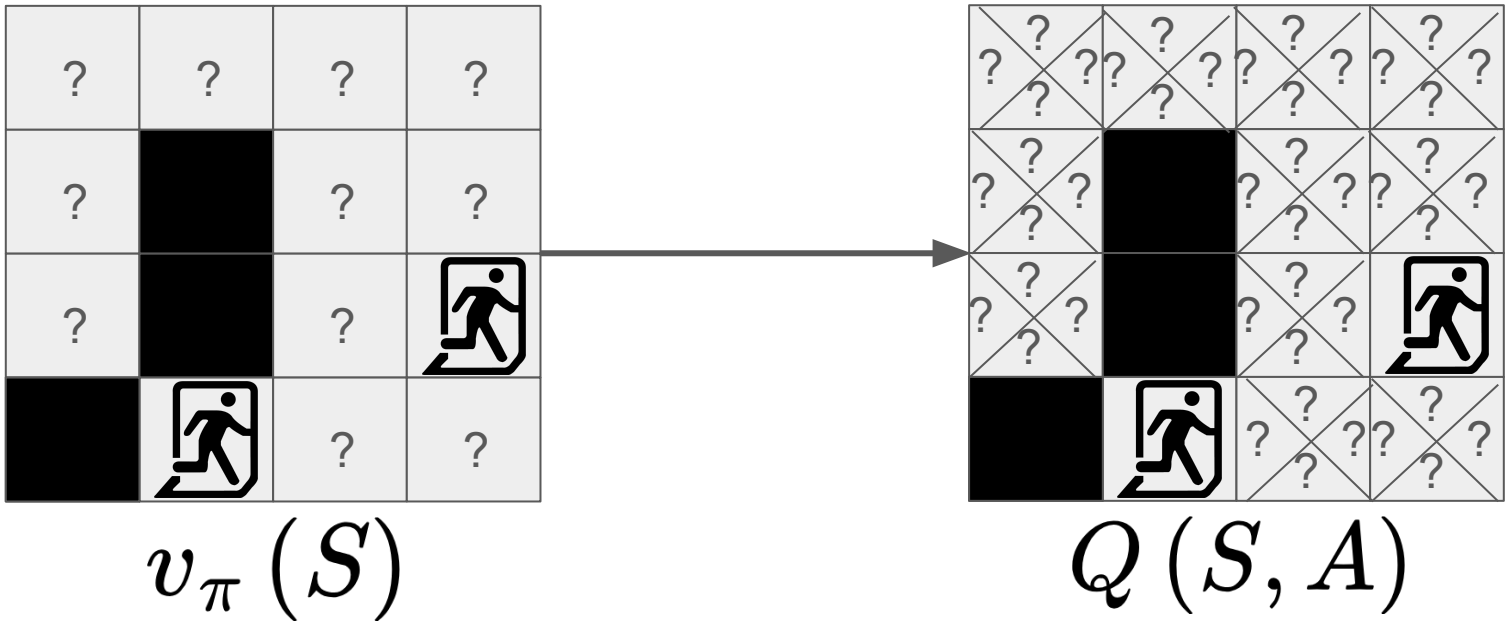
Expected discounted return

Starting at state s

Value Function $v_\pi(S) \longrightarrow Q(S, A)$



Q Function $Q(S, A)$

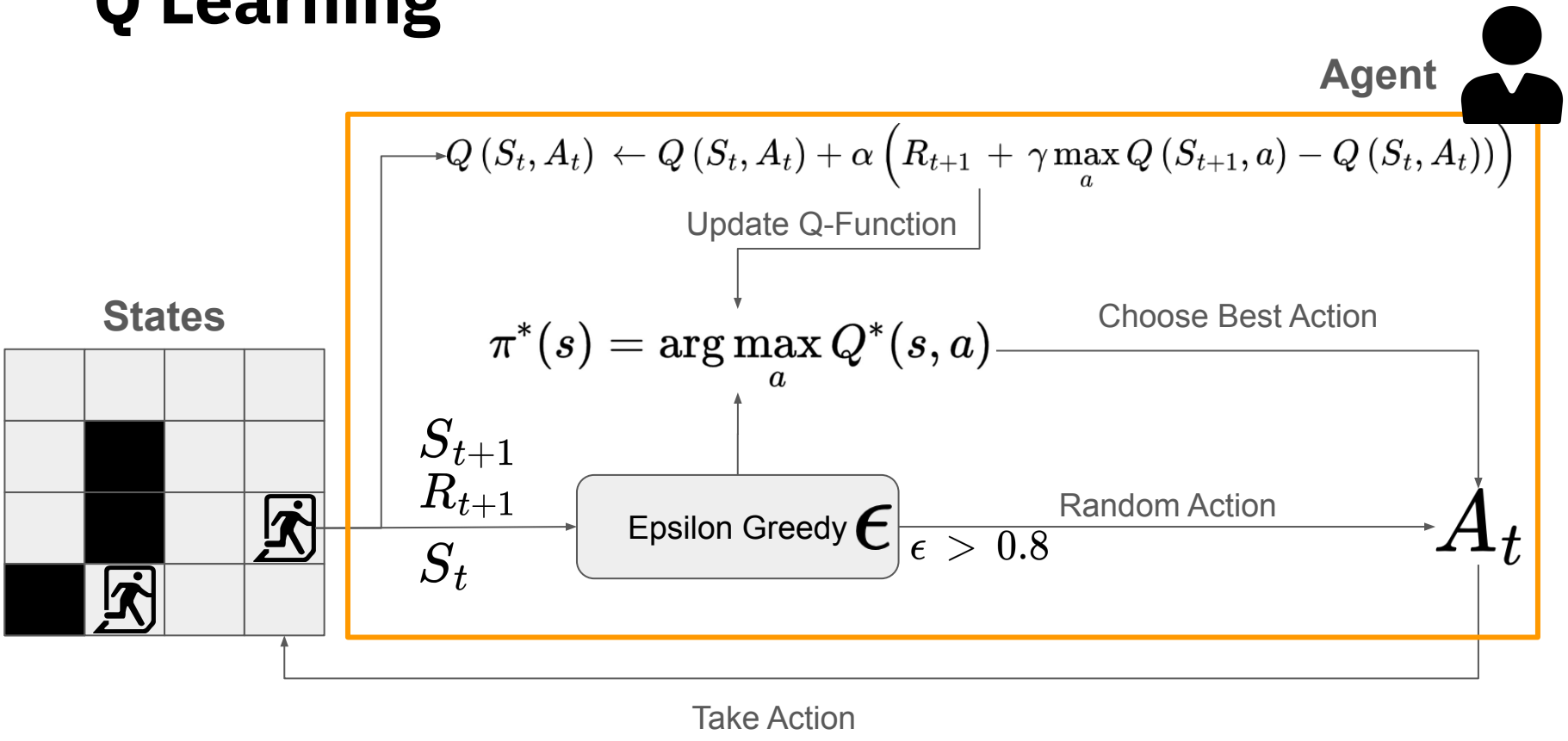


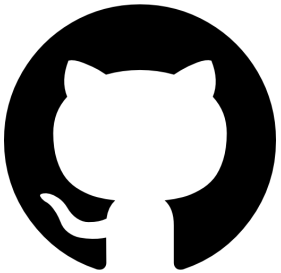
Q Learning

Algorithm 14: Sarsamax (Q-Learning)

Input: policy π, positive integer *num_episodes*, small positive fraction α, GLIE $\{\epsilon_i\}$
Output: value function Q ($\approx q_\pi$ if *num_episodes* is large enough)
Initialize Q arbitrarily (e.g., $Q(s,a) = 0$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}(s)$, and $Q(\textit{terminal-state}, \cdot) = 0$)
for $i \leftarrow 1$ ***to*** *num_episodes* **do** Step 1
  $\epsilon \leftarrow \epsilon_i$
  Observe S_0
  $t \leftarrow 0$
  repeat
    Choose action A_t using policy derived from Q (e.g., ϵ-greedy) Step 2
    Take action A_t and observe R_{t+1}, S_{t+1} Step 3
    $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha(R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t))$ Step 4
    $t \leftarrow t + 1$
  until S_t *is terminal*;
end
return Q

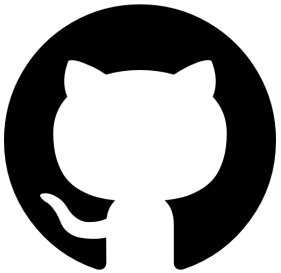
Q Learning


Agent
$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\left(R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)\right)$
Update Q-Function
States
$\pi^*(s) = \arg\max_a Q^*(s, a)$
Choose Best Action
S_{t+1}
R_{t+1}
S_t
Epsilon Greedy ϵ
$\epsilon > 0.8$
Random Action
A_t
Take Action



Workshop #1

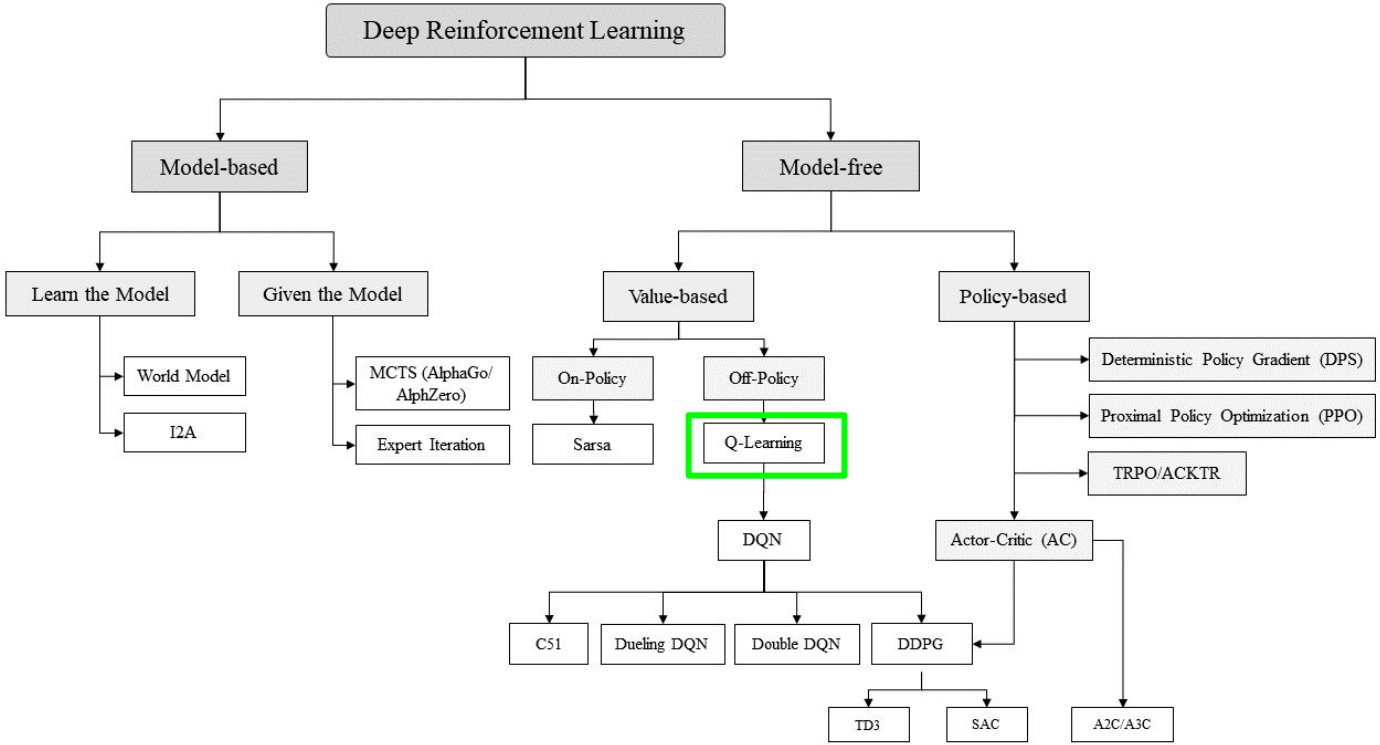
https://github.com/ro-witthawin/BU-DeepReinforcementLearning



Workshop #2

https://github.com/ro-witthawin/BU-DeepReinforcementLearning

Reinforcement Learning Organize Chart



Ref : https://smartmobilityalgorithms.github.io/book/content/LearnToSearch/ReinforcementLearning.html

Reinforcement Learning

Model-based RL

Markov Decision Process $P(\mathrm{s}', \mathrm{s}, \mathrm{a})$

Policy Iteration $\pi_\theta(\mathrm{s}, \mathrm{a})$

Value Iteration $V(\mathrm{s})$

Actor Critic

Dynamic programming & Bellman optimality

Nonlinear Dynamics

$$\frac{\mathrm{d}}{dt}\mathbf{x} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t)\, dt$$

Optimal Control & HJB

Deep MPC

Model-free RL

Gradient free

Off Policy

DQN

$Q(\mathrm{s}, \mathrm{a})$

Q Learning

On Policy

TD(0)

⋮

TD(∞) ≡ MC

TD-λ

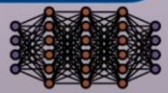
SARSA

Gradient based

Deep Policy Network

$$\theta^{\text{new}} = \theta^{\text{old}} + \alpha \nabla_\theta R_{\Sigma,\theta}$$

Policy Gradient Optimization



Deep RL

Journal of Artificial Intelligence Research 79 (2024) 1167-1236 Submitted 11/2023; published 04/2024

Structure in Deep Reinforcement Learning: A Survey and Open Problems

Aditya Mohan A.MOHAN@AI.UNI-HANNOVER.DE
Institute of Artificial Intelligence
Leibniz University Hannover

Amy Zhang AMY.ZHANG@AUSTIN.UTEXAS.EDU
University of Texas at Austin,
Meta AI

Marius Lindauer M.LINDAUER@AI.UNI-HANNOVER.DE
Institute of Artificial Intelligence,
L3S Research Center
Leibniz University Hannover

Ref : https://arxiv.org/pdf/2306.16021

University of Washington : https://www.youtube.com/watch?v=0MNVhXEX9to
Google Deepmind : https://www.youtube.com/watch?v=TCCjZe0y4Qc
MIT : https://www.youtube.com/watch?v=8JVRbHAVCws&t=1s
Hugging Face RL : https://huggingface.co/learn/deep-rl-course/en/unit0/introduction

การประชุมเทคโนโลยีโอเพนซอร์สชั้นนำของเอเชีย

งานประจำปีครั้งที่ 16 • วิทยากรนานาชาติกว่า 170 คน • 200+ หัวข้อเสวนา

Event supported by depa

FOSS asia summit 2025

13-15 March

Bangkok, Thailand

summit.fossasia.org

Open Source • AI • Cloud • Security • Database • Hardware • Data Science


<CoDev> × fossasia
fossasia
Summit
2025
FREE
FOSSASIA Summit 2025 for free!!!
สำหรับชาว CoDev และนักศึกษา (มูลค่า 1,000 - 3,000 บาท)
เปิดโอกาสเรียนรู้จากผู้นำเทคโนโลยีและสร้างเครือข่าย
เพียงเข้าร่วมดิสคอร์ด แนะนำตัว และรับไปเลย

See ya next time

Witthawin Sripheanpol (Ro)

Witthawin Sripheanpol

ro-witthawin/

I’m AI Engineer & Researcher