Reinforcement Learning Fundamentals

Lecture 3: RL Framework

Dr Sandeep Manjanna
Assistant Professor, Plaksha University
sandeep.manjanna@plaksha.edu.in

Some material in this lecture is taken from

1. Prof. Ravindran's course: "Reinforcement Learning."
2. Dr Silver's course: "Reinforcement Learning."

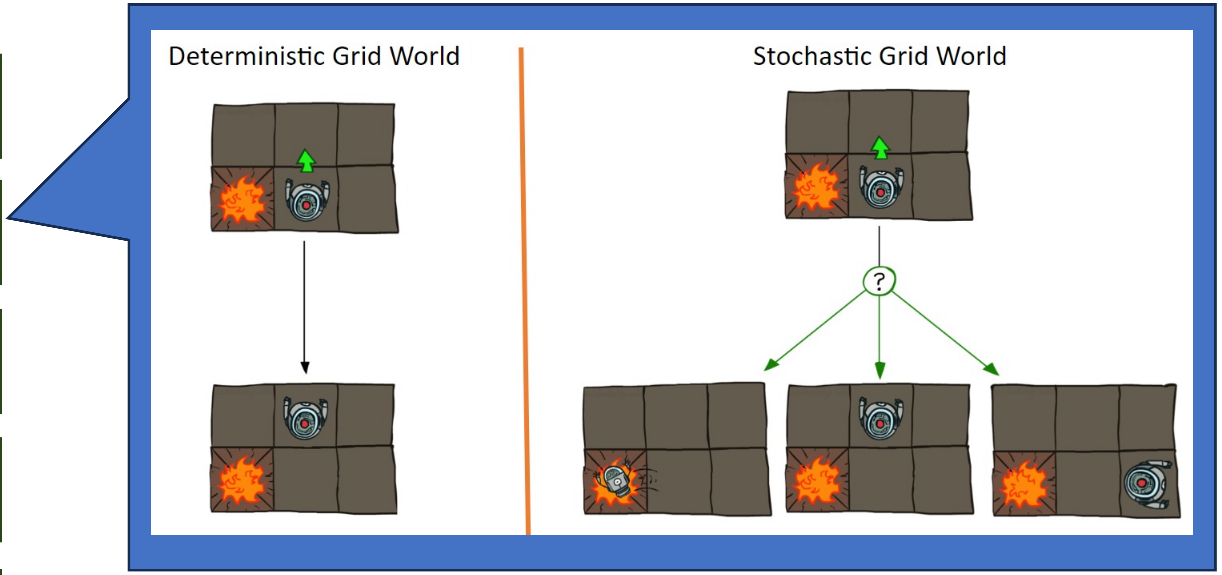
In today's class…

- RL Framework
- What is a State?
- Special cases: Fully and Partially Observable environments
- Temporal Difference
- Components of an RL agent
  - Value function
  - Policy
  - Model
- Categories of RL

Types of Environments

Environments in AI

- Fully Observable vs Partially Observable
- Deterministic vs Stochastic
- Competitive vs Collaborative
- Single-agent vs Multi-agent
- Static vs Dynamic
- Discrete vs Continuous
- Episodic vs Sequential
- Known vs Unknown



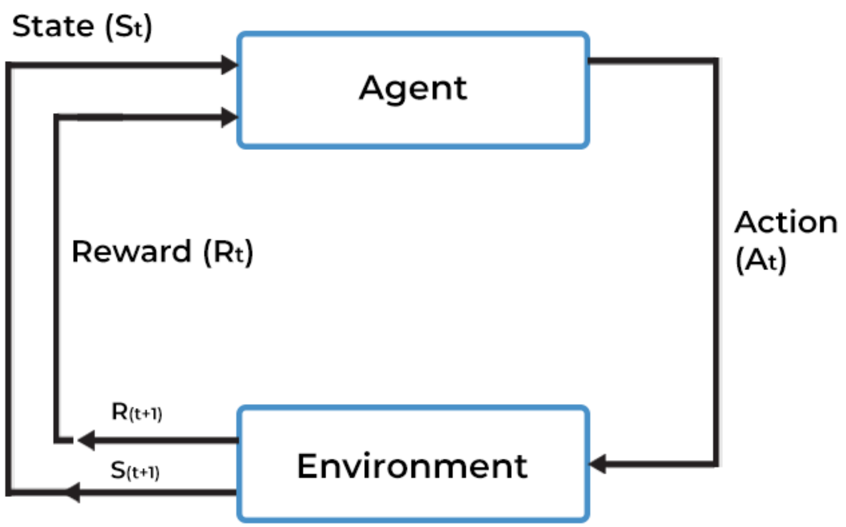
https://www.geeksforgeeks.org/types-of-environments-in-ai/

Fully Observable Environments

… extent to which the agent has access to information about the current state of the environment.

- A fully observable environment is one in which the **agent has complete information about the current state of the environment.**
- The agent has direct access to all environmental features that are necessary for making decisions.
- Example?

Board games like chess or checkers.



Full observability: agent directly observes environment state

$$O_t = S_t^a = S_t^e$$

- Agent state = environment state = information state
- Formally, this is a Markov decision process (MDP)

Partially Observable Environments

- A partially observable environment is one in which the **agent does not have complete information about the current state of the environment.**

- The agent can only observe a subset of the environment, and some aspects of the environment may be hidden or uncertain.

- Examples?
  driving a car in traffic.
  A trading agent only observes current prices.

Now agent state $\neq$ environment state

Formally this is a partially observable Markov decision process (POMDP)

Agent must construct its own state representation S_t^a, e.g.

- Complete history: $S_t^a = H_t$
- Beliefs of environment state: $S_t^a = (\mathbb{P}[S_t^e = s^1], \ldots, \mathbb{P}[S_t^e = s^n])$
- Recurrent neural network: $S_t^a = \sigma(S_{t-1}^a W_s + O_t W_o)$

Inside an RL Agent

An RL agent may include one or more of these components:

- **Policy**: agent’s behavior function
- **Value function**: how good is each state and/or action
- **Model**: agent’s representation of the environment

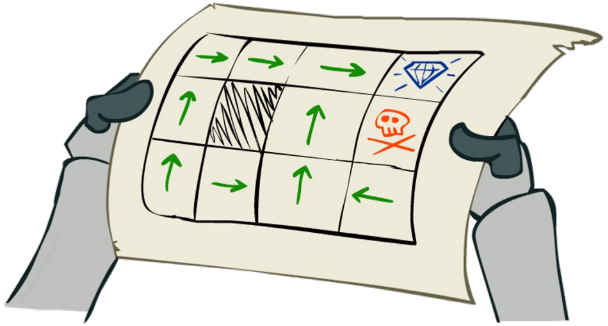
Inside an RL Agent

Policy

- A policy is the agent's behaviour
- It is a map from state to action, e.g.
- Deterministic policy: $a = \pi(s)$
- Stochastic policy: $\pi(a|s) = \mathbb{P}[A_t = a|S_t = s]$

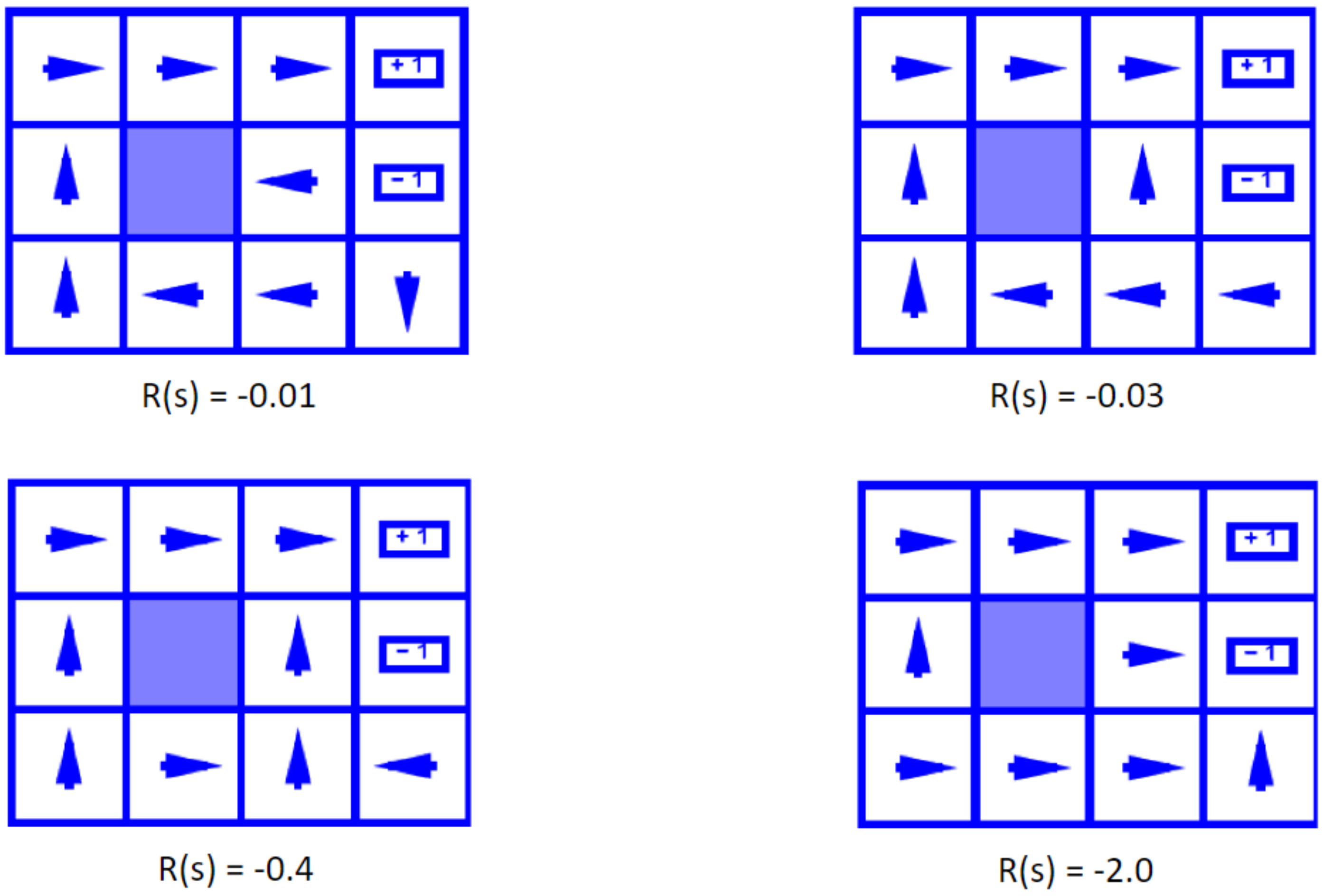
- In deterministic single-agent search problems, we wanted an optimal plan, or sequence of actions, from start to a goal

- For MDPs, we want an optimal policy $\pi^*: S \rightarrow A$
  - A policy π gives an action for each state
  - An optimal policy is one that maximizes expected utility / value if followed
  - An explicit policy defines a reflex agent



Inside an RL Agent
Policy

Optimal Policies



Dan Klein's and Pieter Abbeel's course: CS188 Intro to AI at UC Berkeley.

Inside an RL Agent

Value function

- Value function is a prediction of future reward
- Used to evaluate the goodness/badness of states
- And therefore to select between actions, e.g.

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

Why Discounting?

- It's reasonable to maximize the sum of rewards
- It's also reasonable to prefer rewards now to rewards later
- One solution: values of rewards decay exponentially

1

Worth Now

γ

Worth Next Step

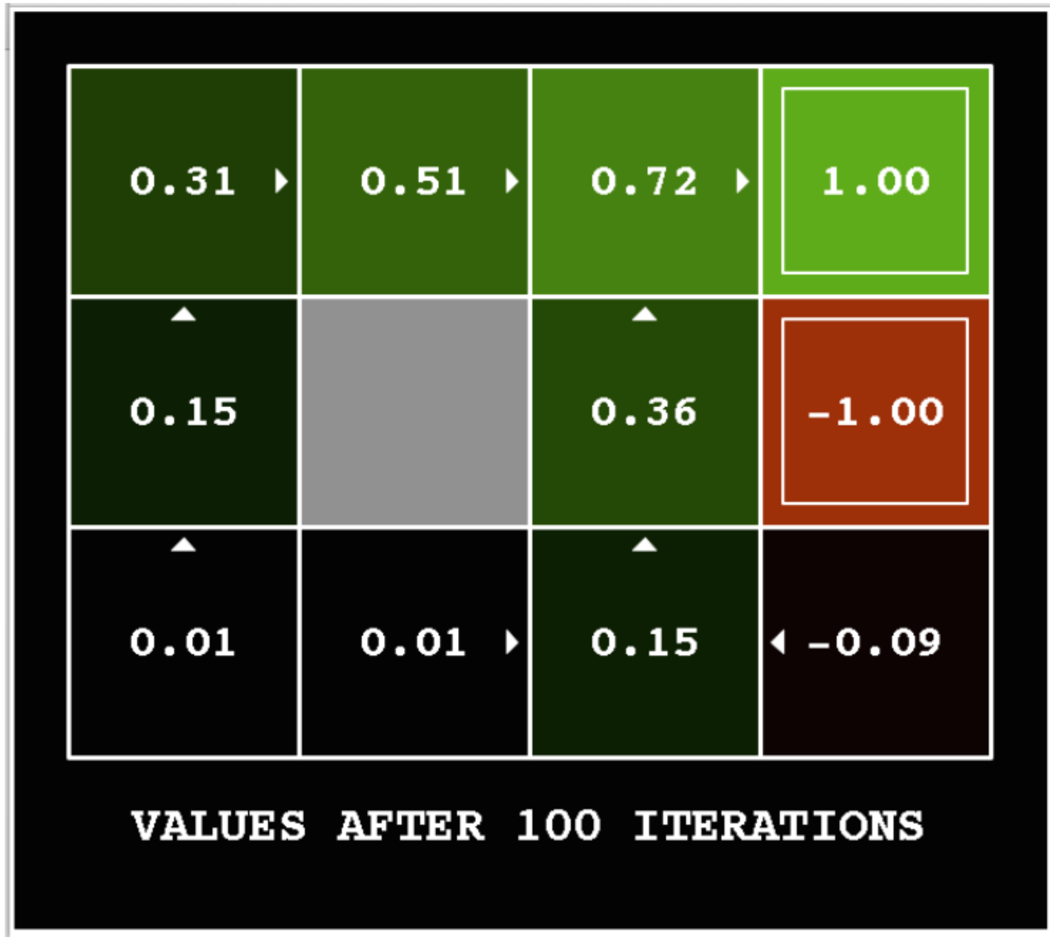
γ^2

Worth In Two Steps

Inside an RL Agent
Value function

Living reward = 0

Living reward = -0.1



Dan Klein's and Pieter Abbeel's course: CS188 Intro to AI at UC Berkeley.

Inside an RL Agent

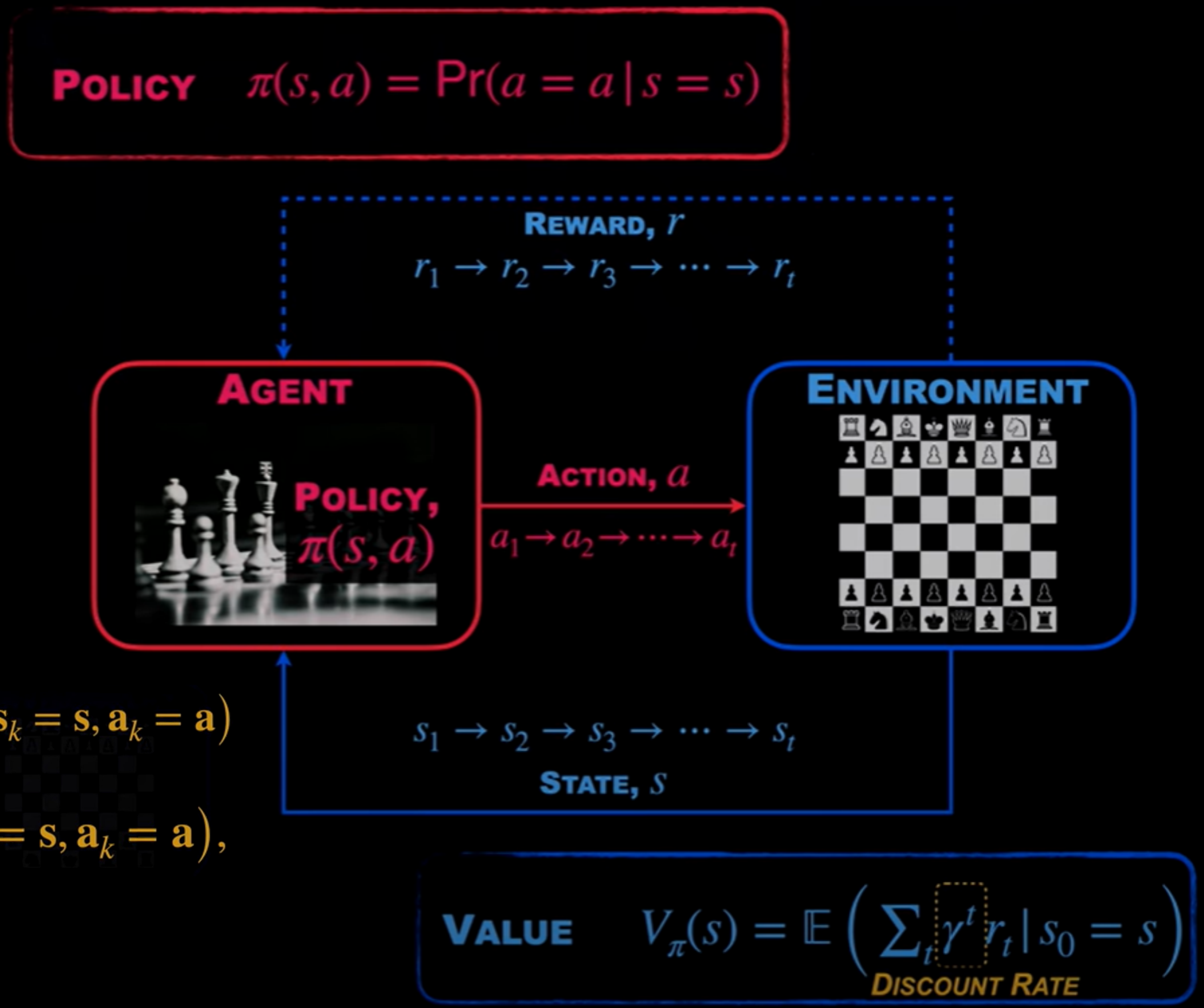
Model

- A model predicts what the environment will do next
- $\mathcal{P}$ predicts the next state
- $\mathcal{R}$ predicts the next (immediate) reward, e.g.

Transition Model → $\mathcal{P}_{ss'}^{a} = \mathbb{P}[S_{t+1} = s' \mid S_t = s, A_t = a]$

Reward Model → $\mathcal{R}_{s}^{a} = \mathbb{E}\left[R_{t+1} \mid S_t = s, A_t = a\right]$

Inside an RL Agent

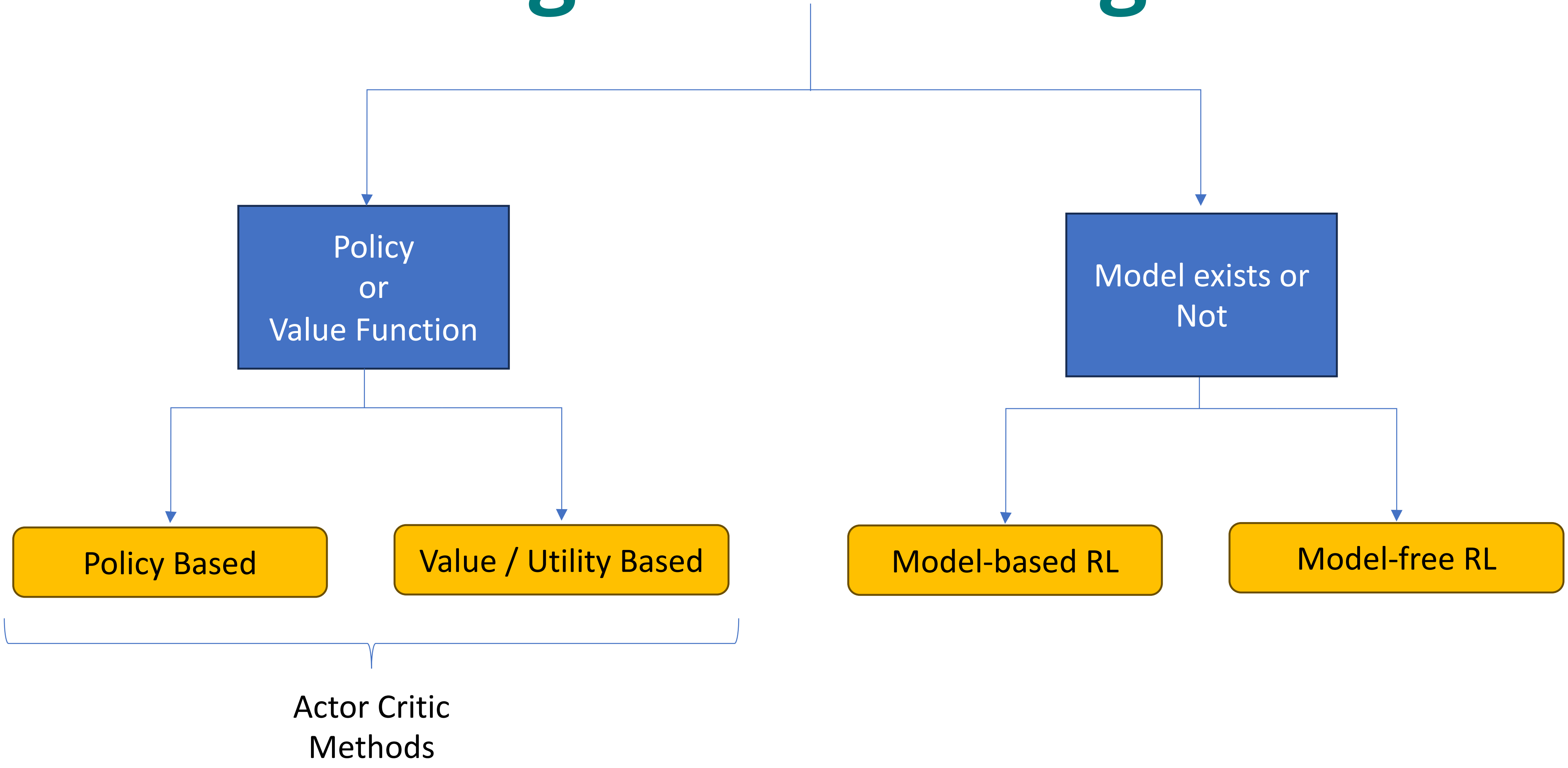


$$R(\mathbf{s}', \mathbf{s}, \mathbf{a}) = \mathbf{Pr}\left(\mathbf{r}_{k+1} \mid \mathbf{s}_{k+1} = \mathbf{s}', \mathbf{s}_k = \mathbf{s}, \mathbf{a}_k = \mathbf{a}\right)$$

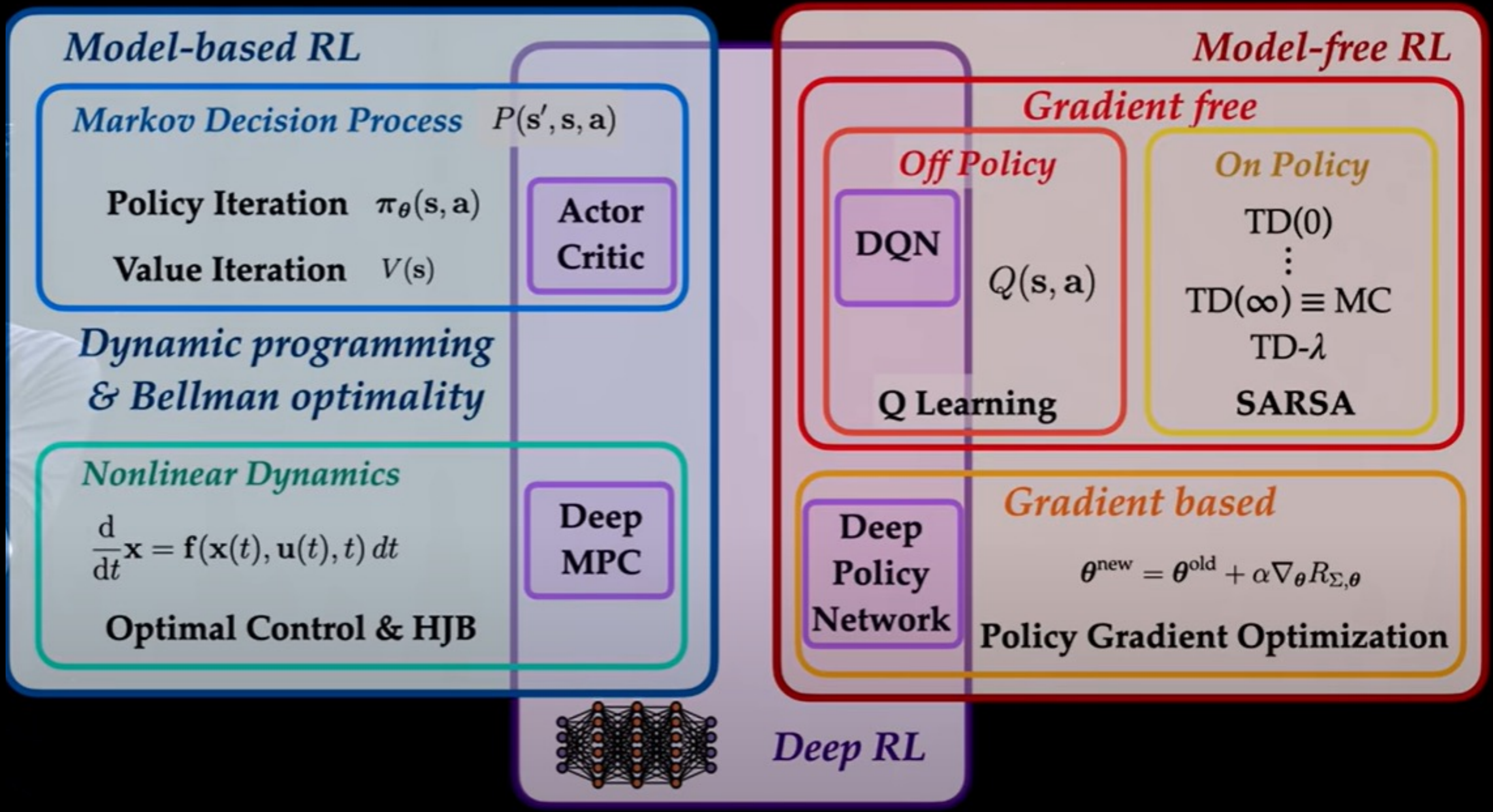
$$P(\mathbf{s}', \mathbf{s}, \mathbf{a}) = \mathbf{Pr}\left(\mathbf{s}_{k+1} = \mathbf{s}' \mid \mathbf{s}_k = \mathbf{s}, \mathbf{a}_k = \mathbf{a}\right),$$

Dr Steve Brunton’s course : Reinforcement Learning.

Categories of RL Agent



REINFORCEMENT LEARNING



Dr Steve Brunton’s course : Reinforcement Learning.