Reinforcement learning

Lecture 0
Why RL, Why Now?

Simon Hirlaender

With many ideas from Emma Brunskills Lectures and some from Sergey Levine

Why RL? Why now?

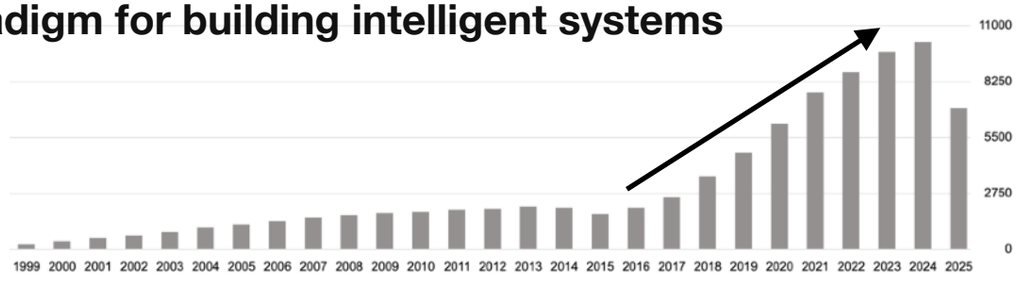
Who knows these guys?

2025 ACM A.M. Turing Award - “Nobel Prize of Computing”

➡Recognised for developing the **conceptual and algorithmic foundations of RL**

➡Pioneering since **1980s**: introduced core ideas, mathematical frameworks, and key algorithms

➡Their contributions shaped RL into a **central paradigm for building intelligent systems**



Why RL, and why now?

- We live in a world of increasing complexity, uncertainty, and interactivity
- RL provides a framework for learning to **solve complex problems from experience** → beyond static behaviour.
- **A growing paradigm not just in AI research** → but also in engineering, economics, healthcare, education …

But first things first…

- 1950s → Bellman & Dynamic Programming (Markov decision processes introduced, 1957)
- 1980s–90s → RL foundations (Sutton & Barto, Q-learning)
- 2010s → Deep RL (Atari, AlphaGo)
- 2020s → Real-world RL (RL from human feedback, Robotics, Industry)

The real power of RL

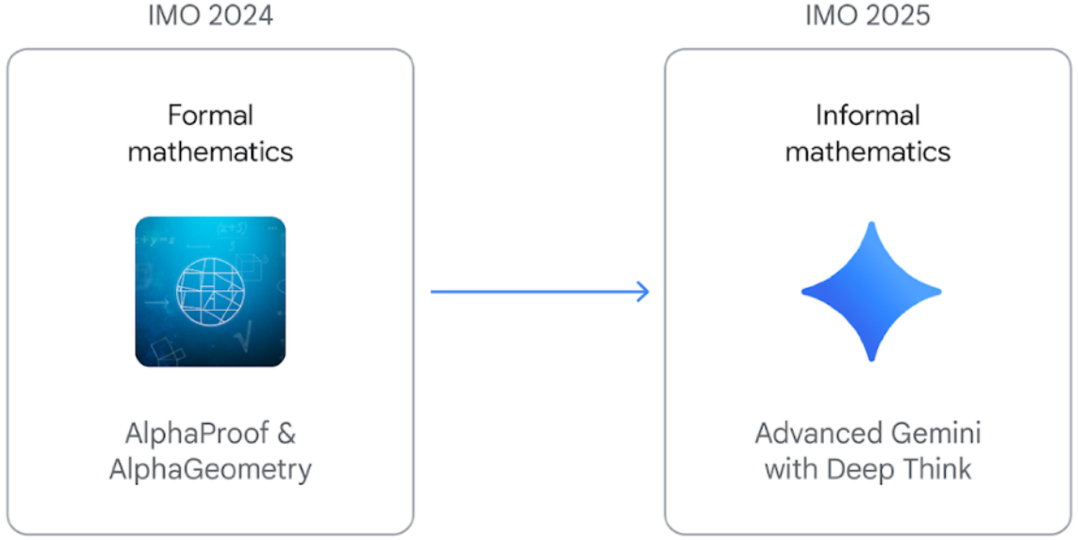
- **When we know the goal, but not the path**
- Effective where handcrafted or modelled rules are too rigid or unavailable
- Capable of learning from sparse and delayed rewards
- Fosters adaptive decision-making

- No examples of desired behaviour: e.g. because the goal is to go beyond human performance or there is no existing data for a task
- Enormous search or optimisation problem with delayed outcomes

Some really impressive examples

2025: AI scores Gold at the International Mathematical Olympiad

- **Gold-level score**, officially graded: 35/42 on IMO 2025, 5 of 6 problems within 4.5 hours.
- **Thinks in parallel**: Tries multiple solution paths before answering.
- **Learns with RL**: Trained with new reinforcement-learning techniques to strengthen multi-step reasoning.
- Checks its work: Produces clear, rigorous solutions in **natural language**.

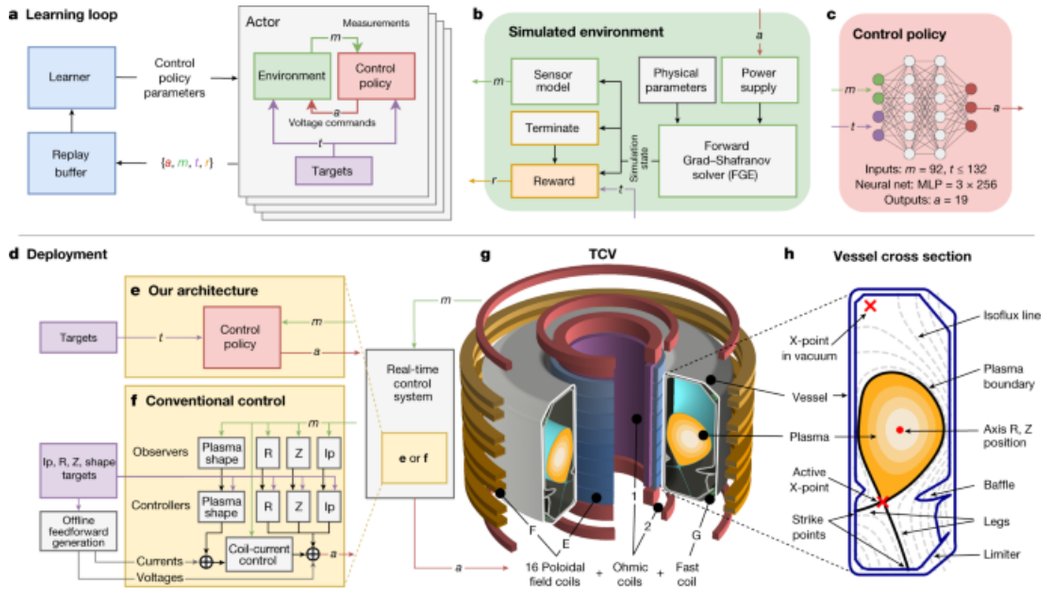


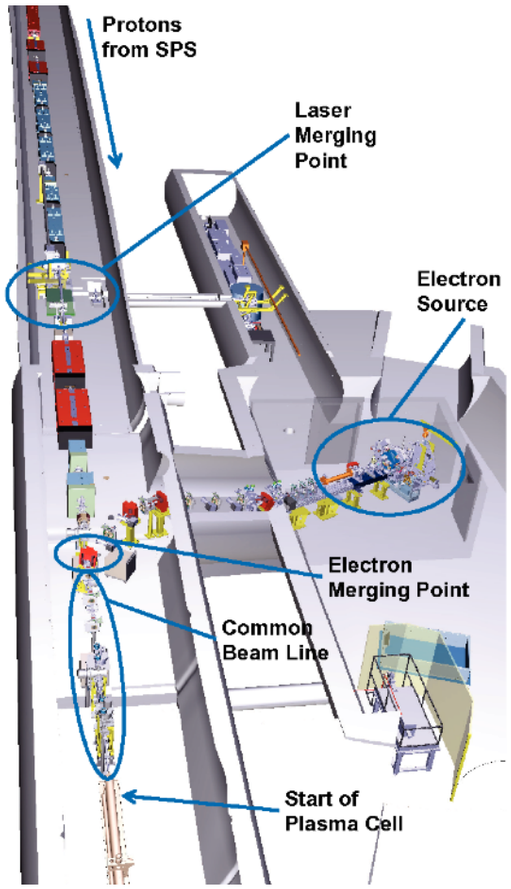
2024: Silver-level with AlphaProof (RL-based) + AlphaGeometry 2
2025: Natural-language “Deep Think” + RL, gold-level score.

https://deepmind.google/discover/blog/advanced-version-of-gemini-with-deep-think-officially-achieves-gold-medal-standard-at-the-international-mathematical-olympiad/

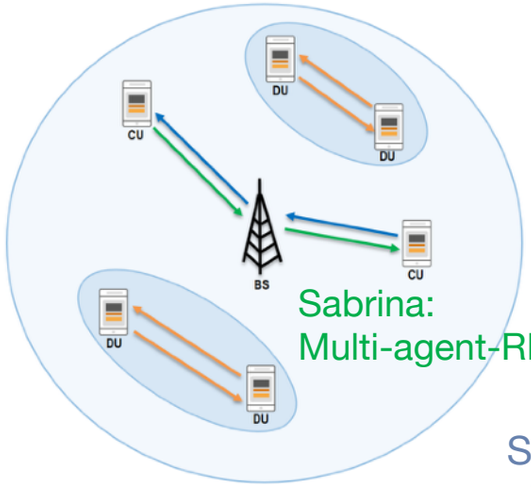
Embodied examples

Physics…

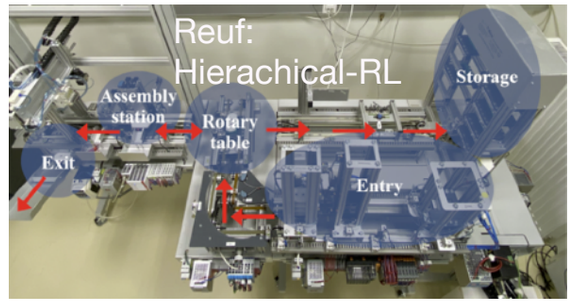

a Learning loop
Learner
Control policy parameters
Actor
Measurements
Environment
Control policy
Voltage commands
Replay buffer
Targets
b
Simulated environment
Sensor model
Physical parameters
Power supply
Terminate
Forward Grad–Shafranov solver (FGE)
Reward
Simulation state
c
Control policy
Inputs: m = 92, t ≤ 132
Neural net: MLP = 3 × 256
Outputs: a = 19
d Deployment
e Our architecture
Targets
Control policy
f Conventional control
Observers
Plasma shape
R
Z
Ip
Ip, R, Z, shape targets
Controllers
Offline feedforward generation
Currents
Voltages
Coil-current control
Real-time control system
e or f
g
TCV
16 Poloidal field coils
Ohmic coils
Fast coil
h
Vessel cross section
X-point in vacuum
Vessel
Plasma
Active X-point
Strike points
Isoflux line
Plasma boundary
Axis R, Z position
Baffle
Legs
Limiter


Protons from SPS
Laser Merging Point
Electron Source
Electron Merging Point
Common Beam Line
Start of Plasma Cell

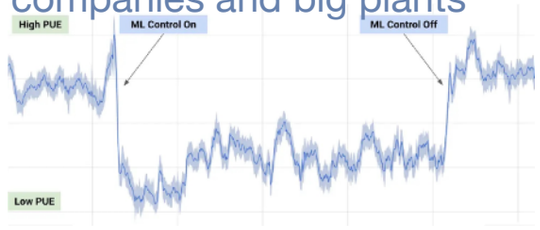
Wide range of applications…



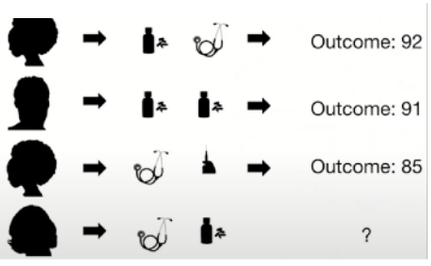
Sabrina:
Multi-agent-RL



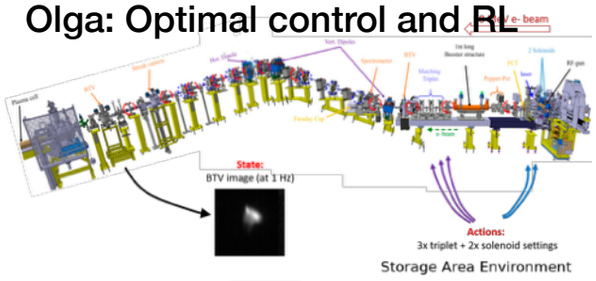
Sarah: reduce power consumption of companies and big plants



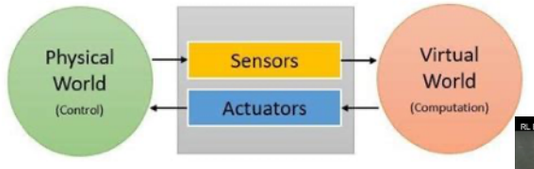
Sascha: Interpretable RL in medicine



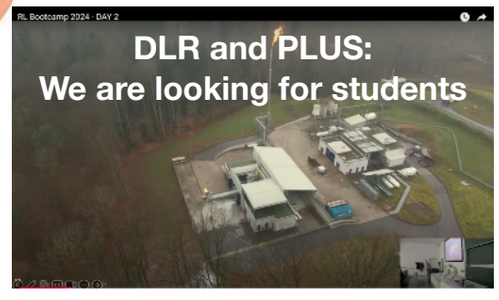
Olga: Optimal control and RL



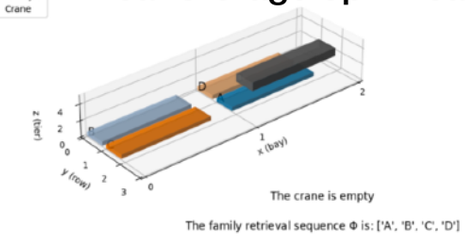
Georg: RL in cyber physical systems



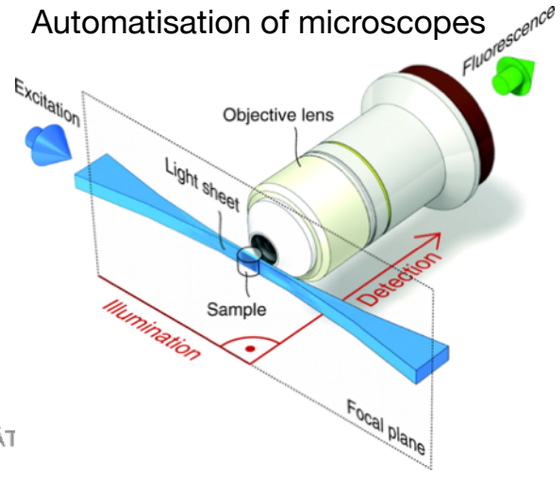
Control theory and RL:
steer a non-linear system



Elisa: Storage optimisation



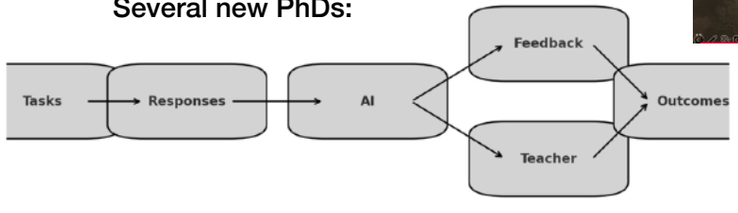
Gesine:
Automatisation of microscopes



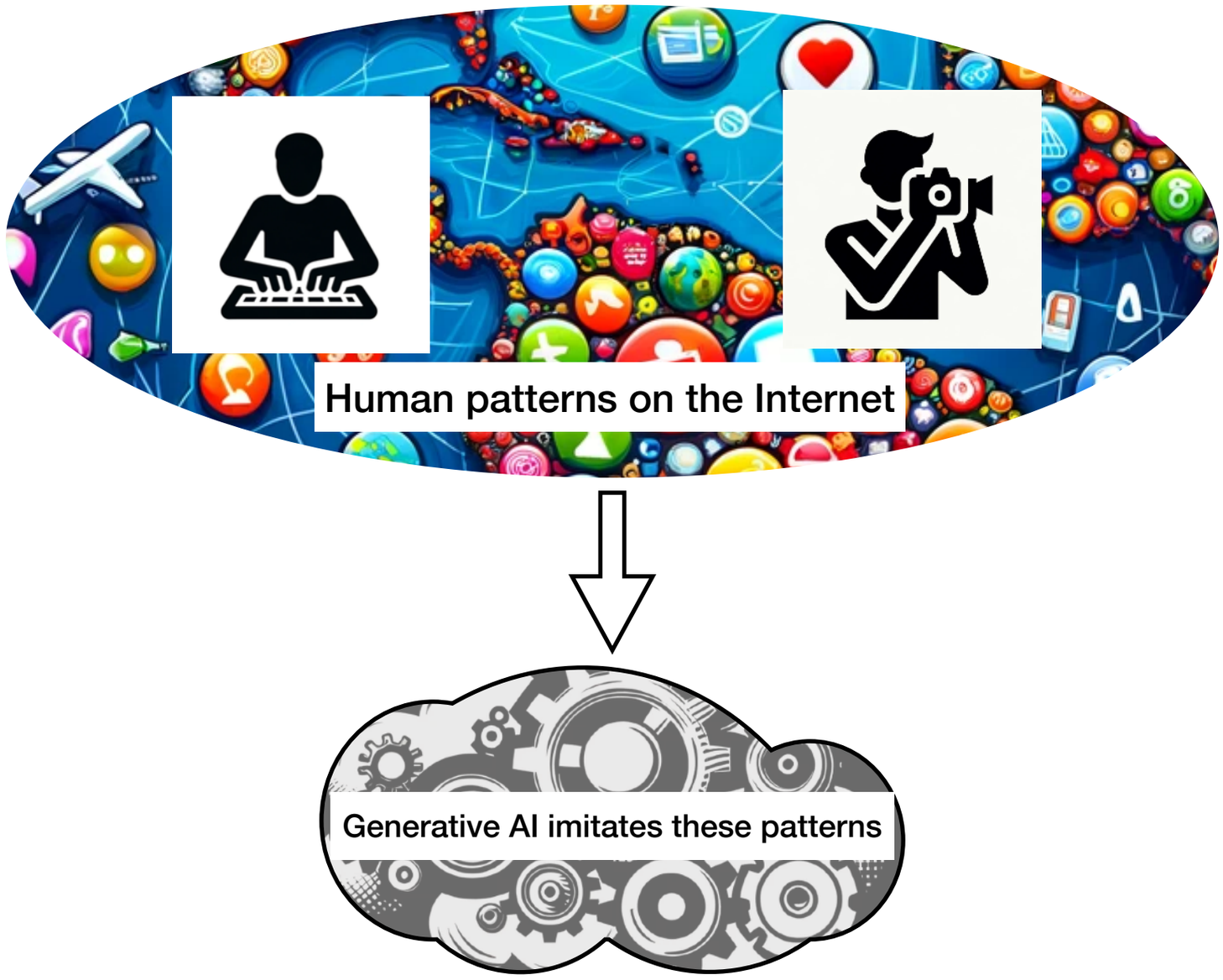
New master thesis

INSPIRE Diagnostic System (Minimalist Flow)

Several new PhDs:



How works popular generative AI?



RL: Beyond Imitation, Towards Discovery

Generative AI: Impressive because it imitates what humans can already create.

Vibrant portrait painting of Salvador Dalí with a robotic half face

A shiba inu wearing a beret and black turtleneck

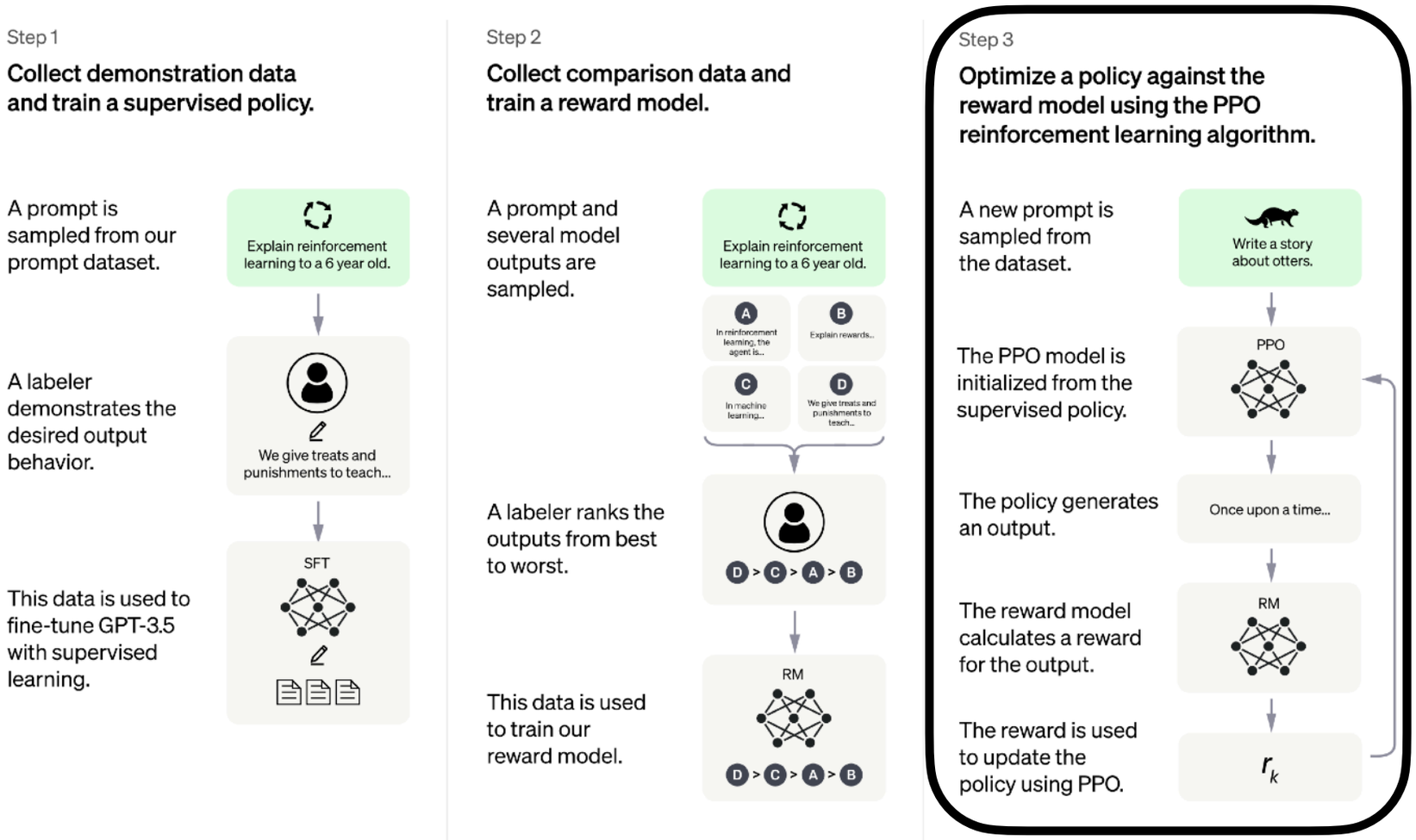
A close-up of a hand palm with leaves growing from it

Reinforcement Learning: Impressive because it discovers strategies no human imagined.

AlphaGo's 'Move 37': A breakthrough move no human expected — discovered through RL.

Adapted from Sergey Levine

But popular generative AI uses RL too



Chat GPT 3.5. “RLHF (Reinforcement Learning from Human Feedback) aligns model outputs with human preferences.”

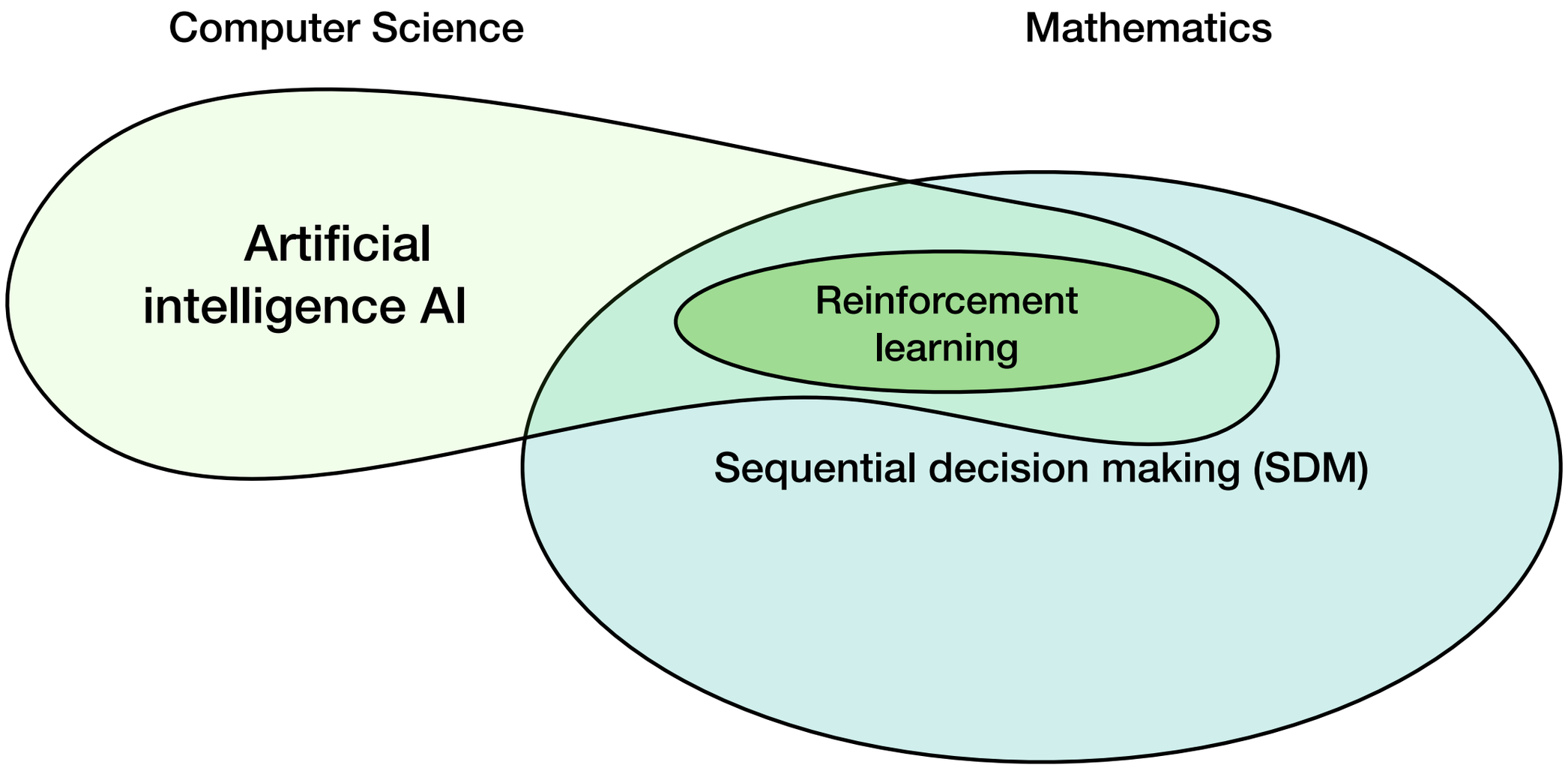
RL: From prediction to action

What is RL?

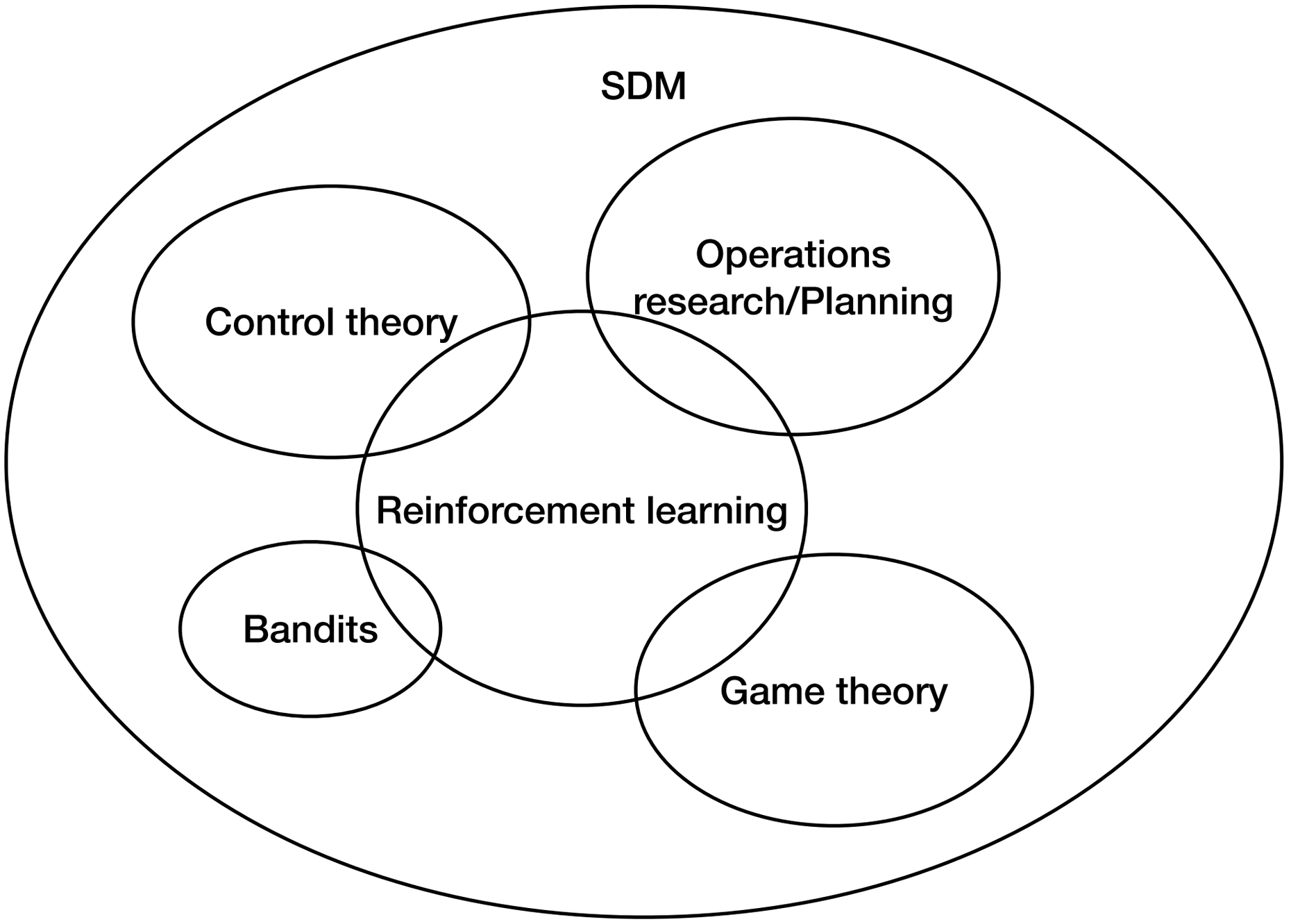
Fundamental challenge in (artificial) intelligence and machine learning:

Learning (from experience/data) how to make good decision sequences under uncertainty

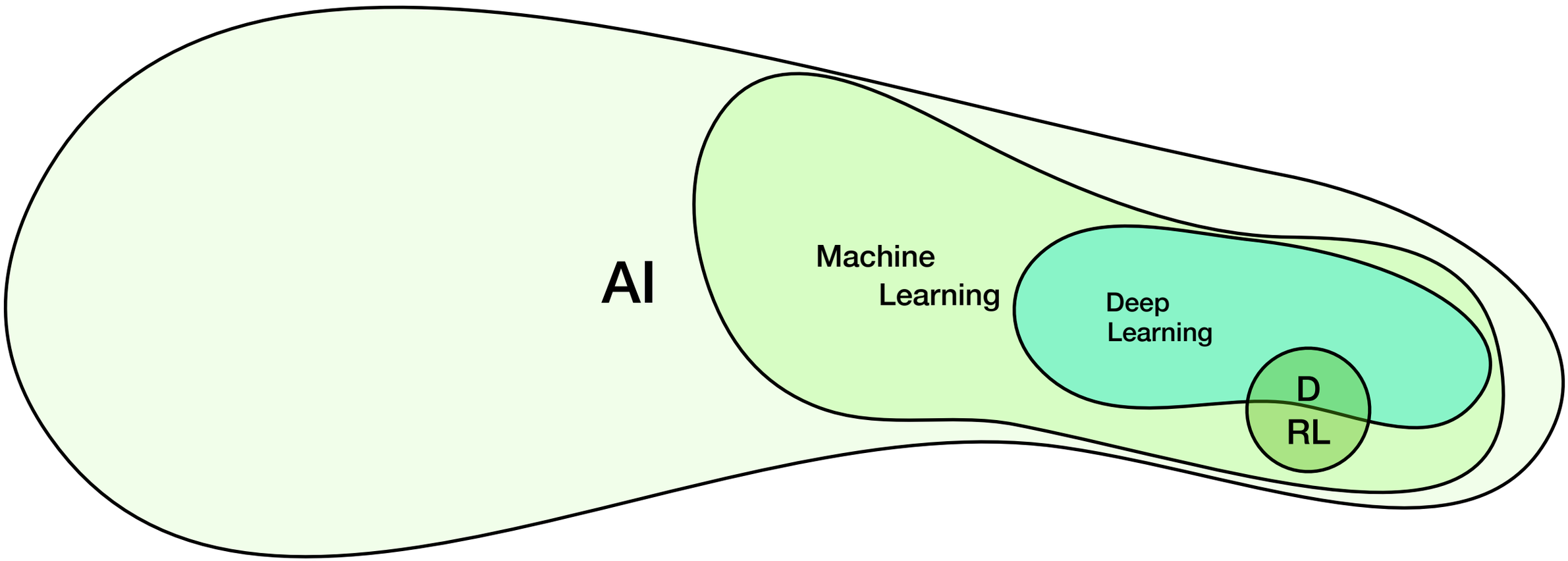
RL as AI's Approach to Sequential Decision Making



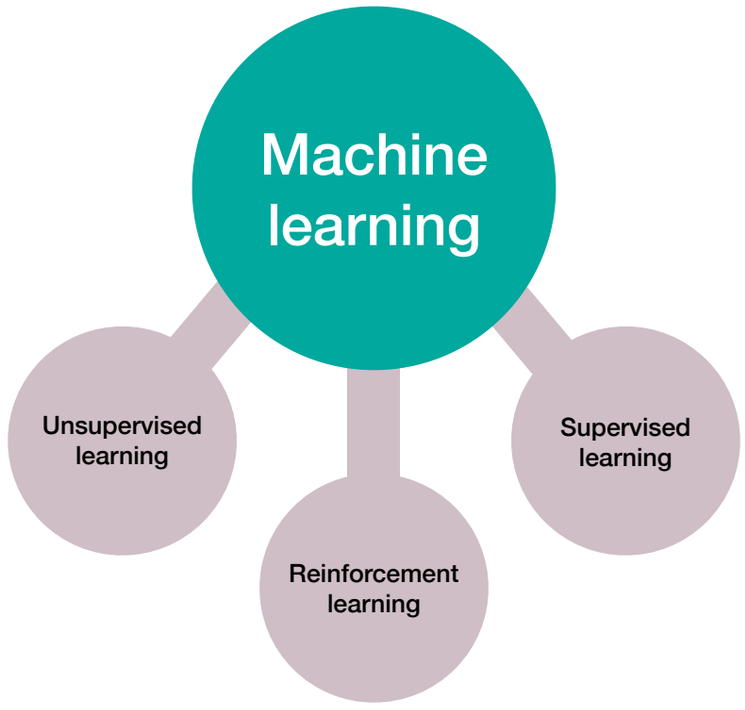
Where does RL stand in the field of sequential decision making (SDM)

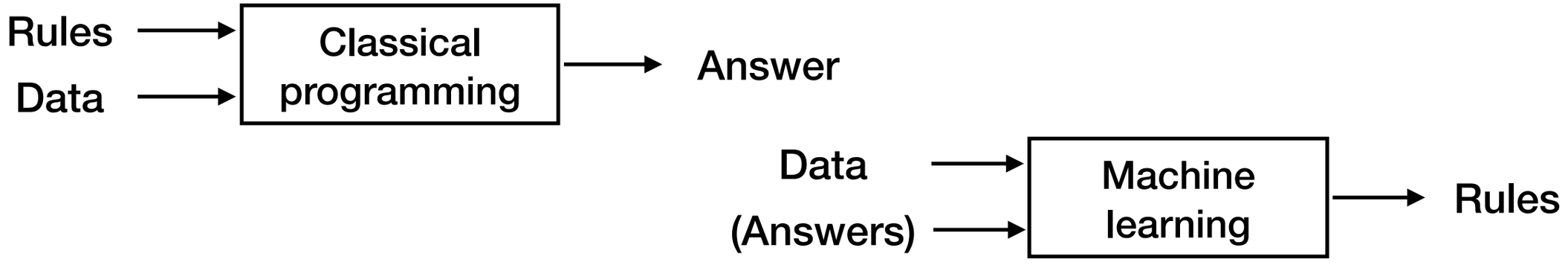


Where does RL stand in the world of artificial intelligence?



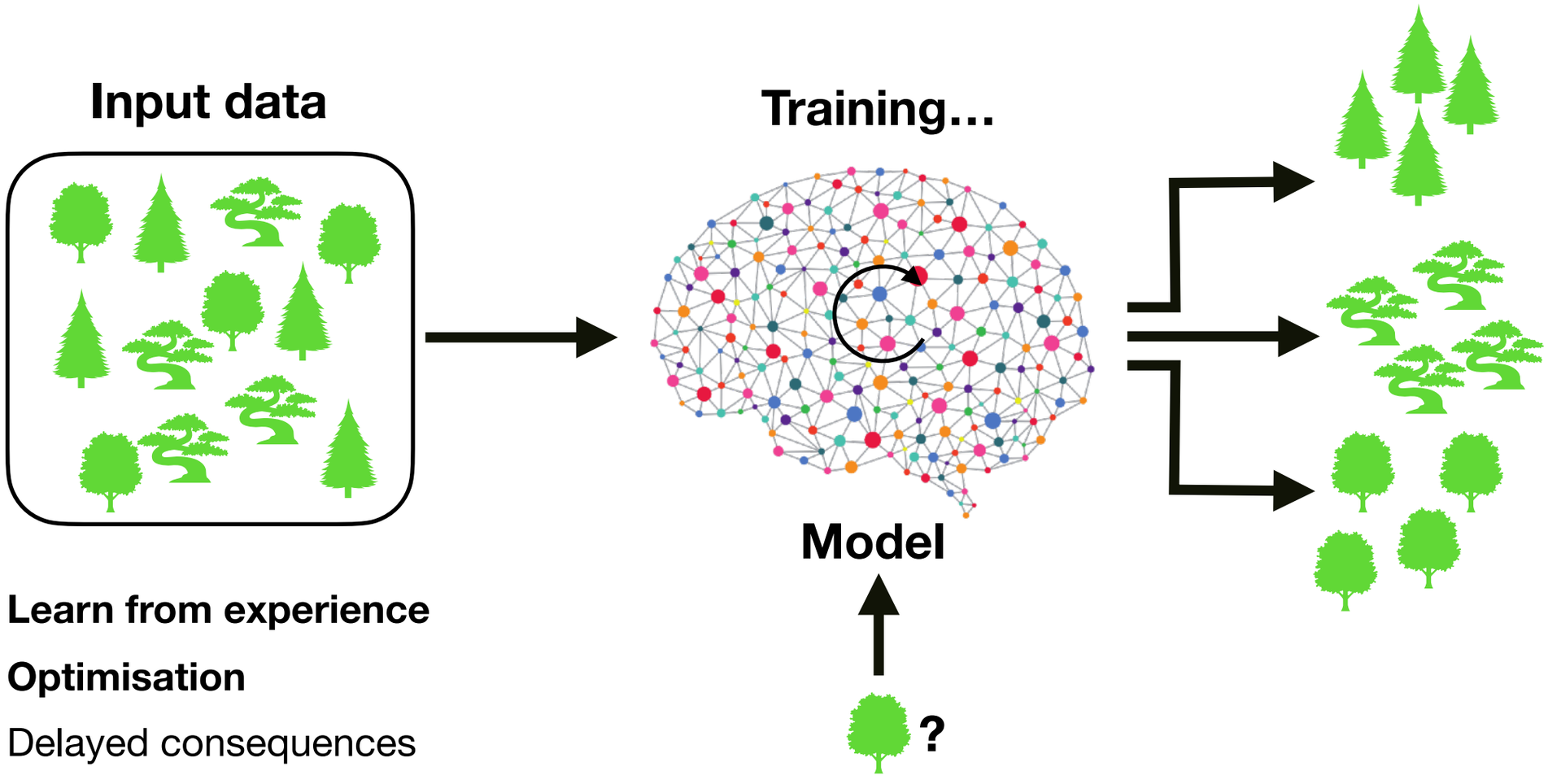
How do algorithms learn?





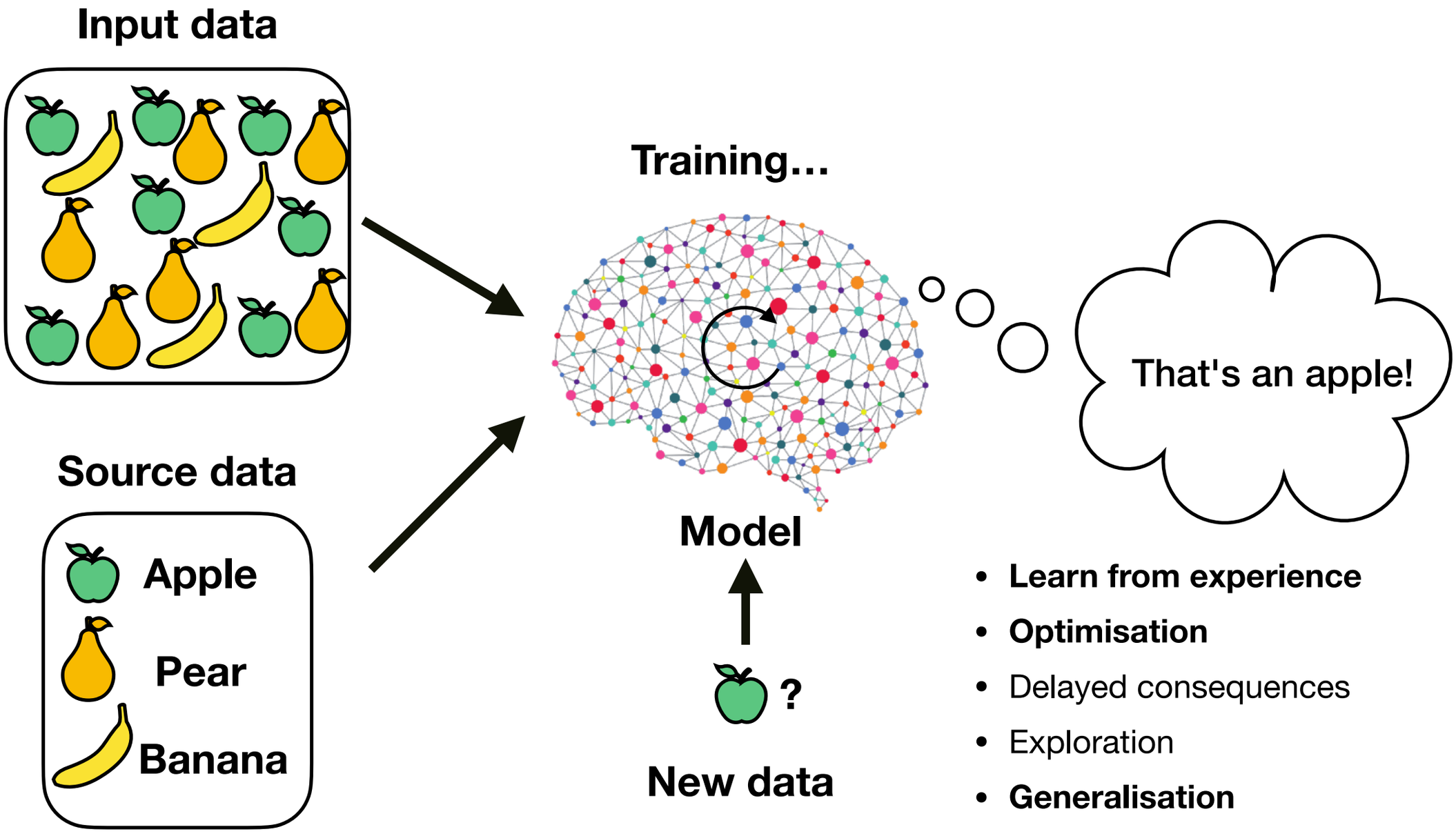
Algorithms learn from experience

Unsupervised Learning



- **Learn from experience**
- **Optimisation**
- Delayed consequences
- Exploration
- **Generalisation**
- **No labels**

Supervised Learning

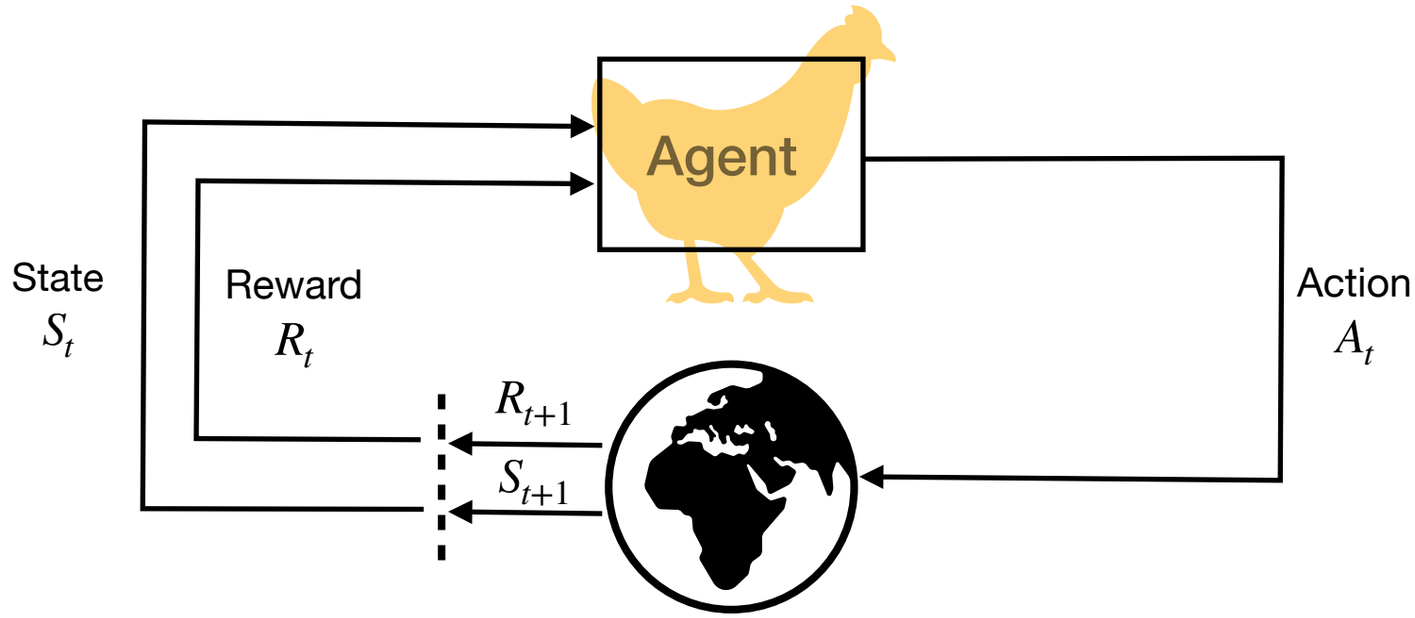


- **Learn from experience**
- **Optimisation**
- Delayed consequences
- Exploration
- **Generalisation**
- **Labels given**

How does RL work?

Information → Decision → Information → Decision → Information → ….

https://www.youtube.com/watch?v=spfpBrBjntg



Learns through interaction with the environment. Goal: maximise expected future cumulative reward.

We are looking for a policy → a function that tells us what action to take in every situation to achieve our goal.

RL involves

- Learn from experience
- Optimisation
- Delayed consequences
- Exploration
- Generalisation

Optimisation

- Goal is to find an optimal way to make decisions
  - ➡ Yielding best outcomes or at least very good outcomes
- Explicit notion of decision utility
- Example: finding minimum distance route between two cities given network of roads

Delayed Consequences

- Decisions now can impact things much later...
  - ➡ Saving for retirement
- Introduces two challenges
  - ➡ When planning: decisions involve reasoning about not just immediate benefit of a decision but also its longer term ramifications
  - ➡ When learning: temporal credit assignment is hard (what caused later high or low rewards?)

Exploration

- Learning about the world by making decisions
  - ➡ Agent as scientist
  - ➡ Learn to ride a bike by trying (and failing)
- Decisions impact what we learn about
  - ➡ Only get a reward for decision made
  - ➡ Don’t know what would have happened for other decision
  - ➡ If we choose to go to Salzburg instead of Vienna, we will have different later experiences...

Generalisation

- The policy is a mapping of accumulated experience to actions
- Why can't you just pre-program a policy?

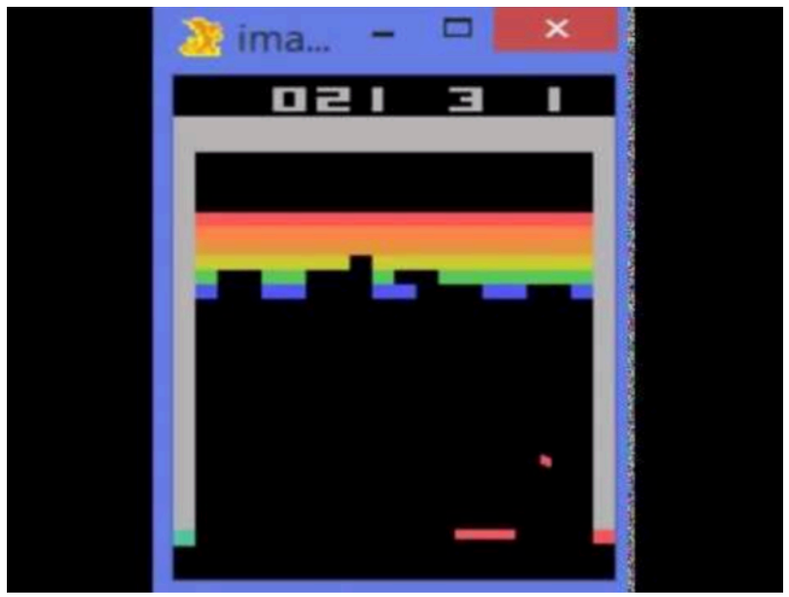
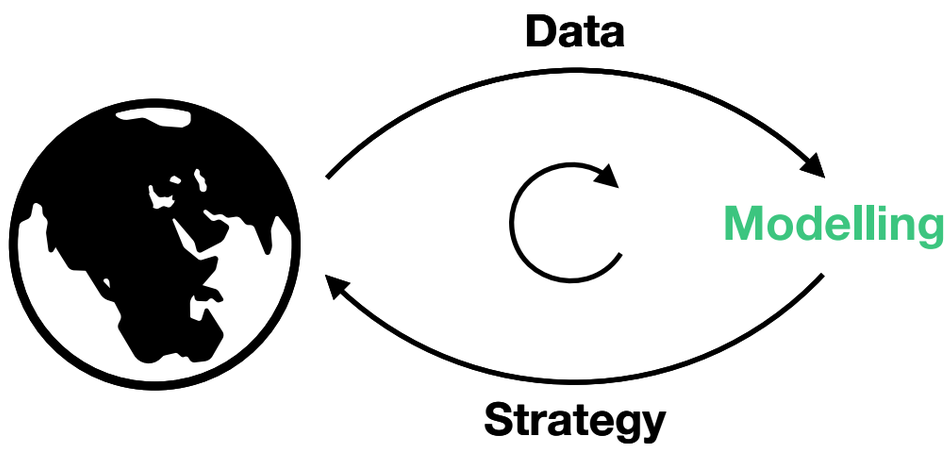


Figure: DeepMind Nature, 2015

- How many possible images are there?
  - $(256^{100\times200})^3$

RL goes beyond prediction

Information → decision → information → decision → information → …

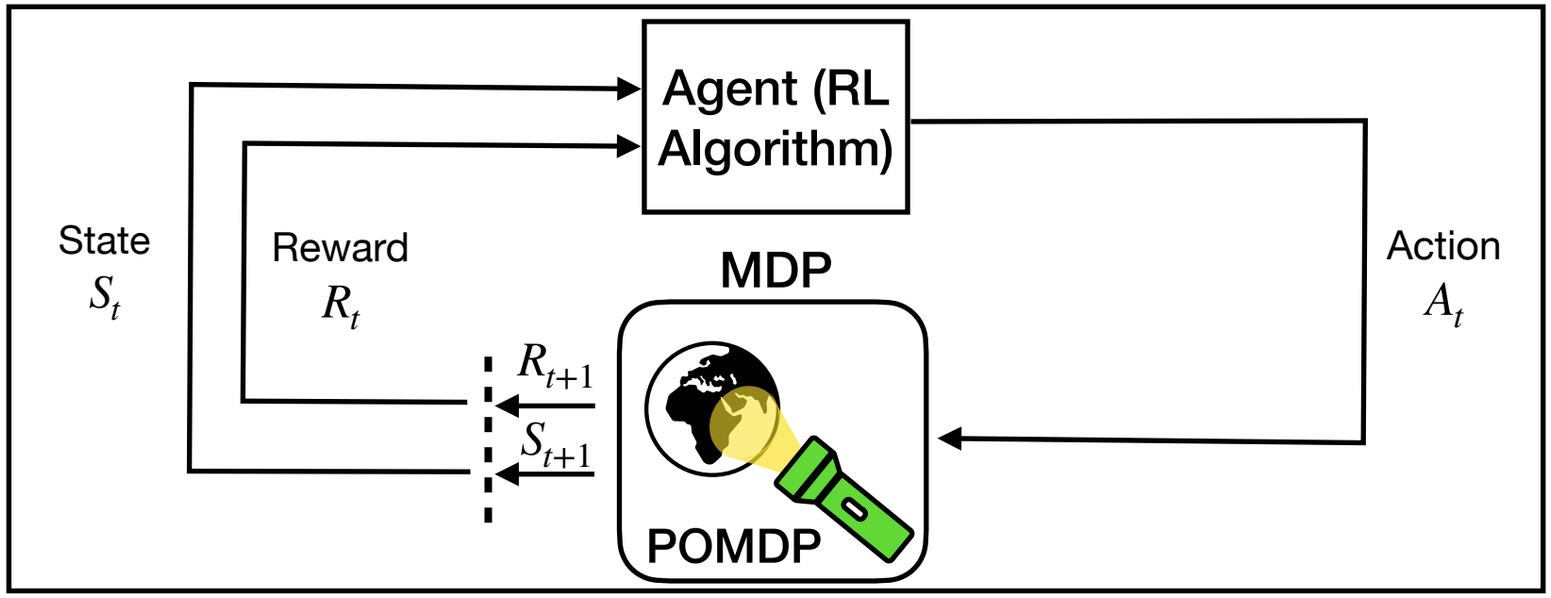


Supervised learning ≈ snapshot; RL ≈ chess game

RL decisions are made with an awareness of their consequences in the world

Basic Terminology

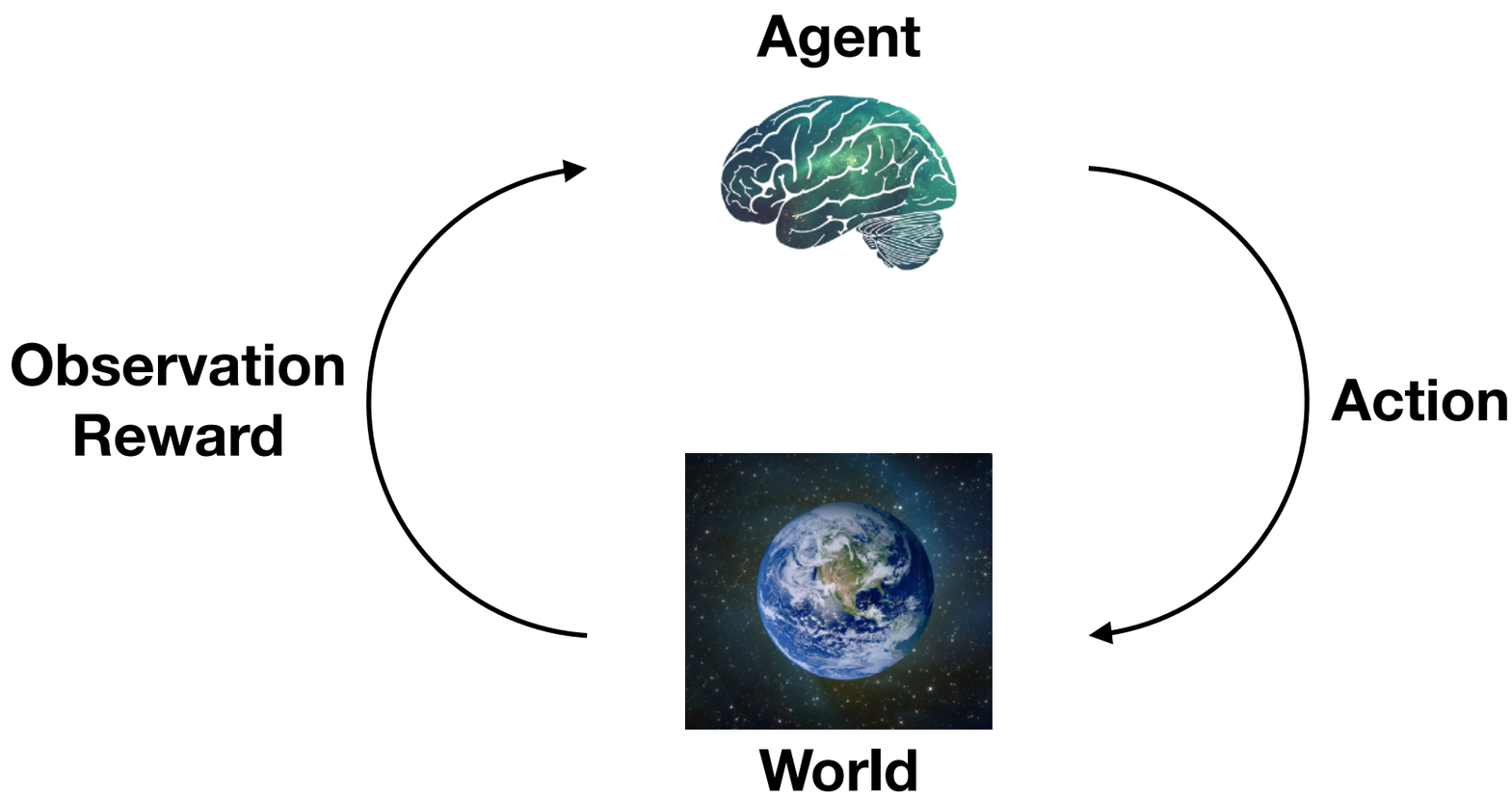
The entire problem



MDP Markov decision process
POMDP Partially observable Markov decision process

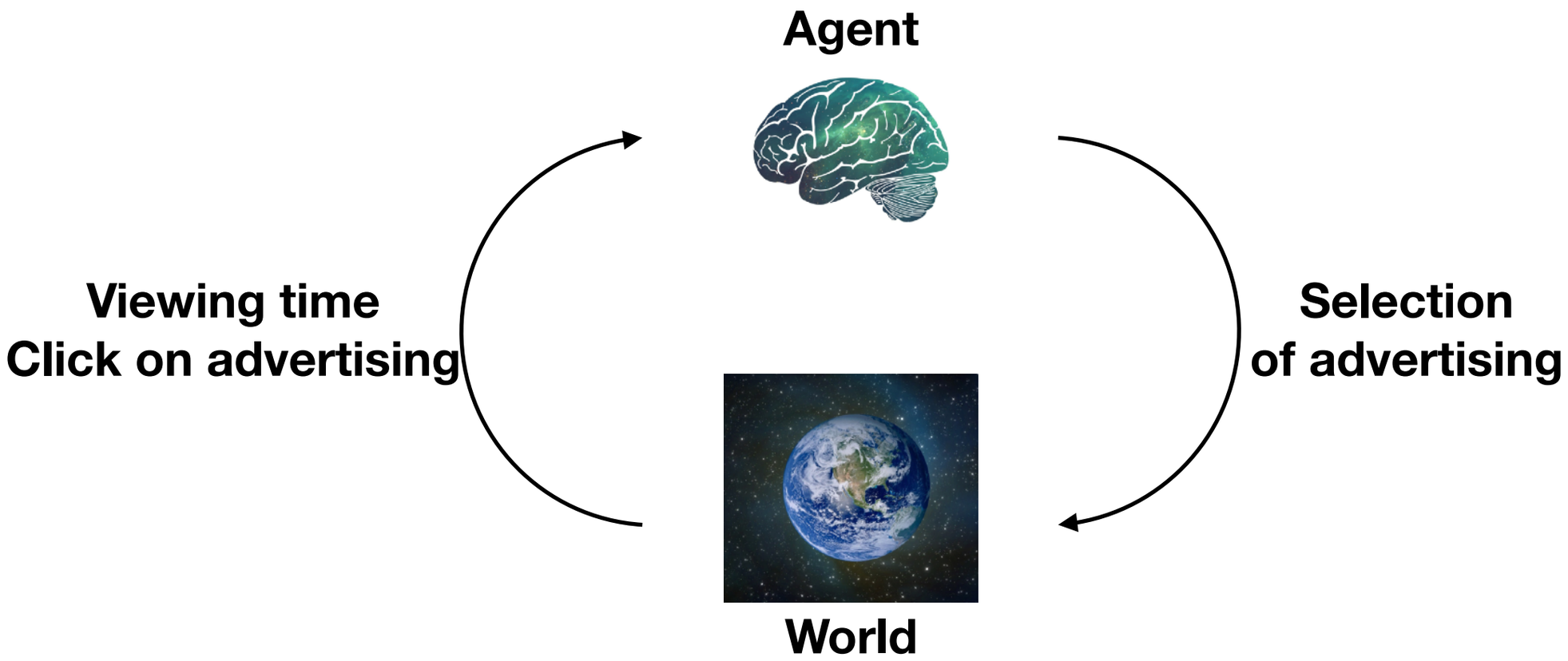
The world of an agent

Sequential decision making under uncertainty



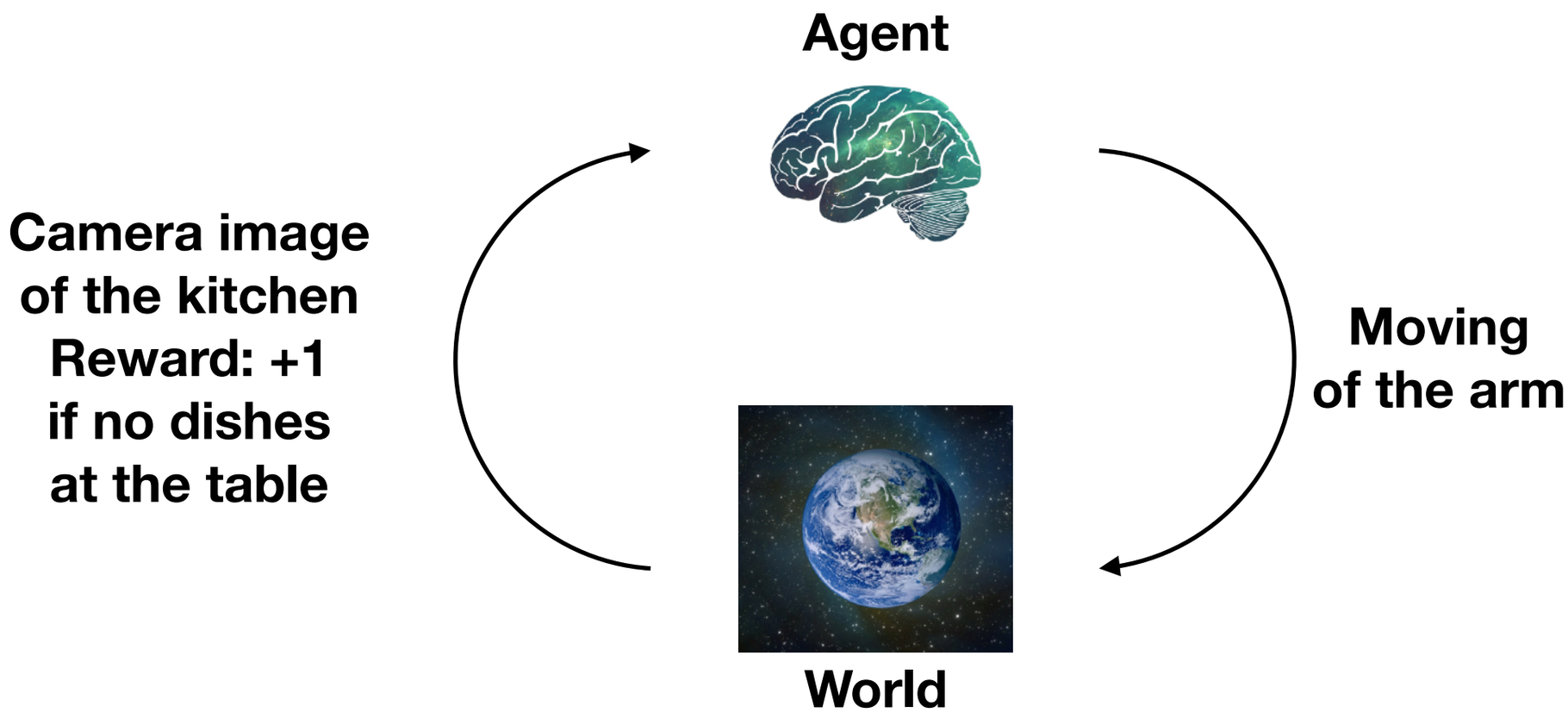
- **Goal**: Find actions to maximise a total expected future reward
- Possible balance of immediate and long-term reward
- Strategy may be needed to achieve high rewards

Example: Web Advertising



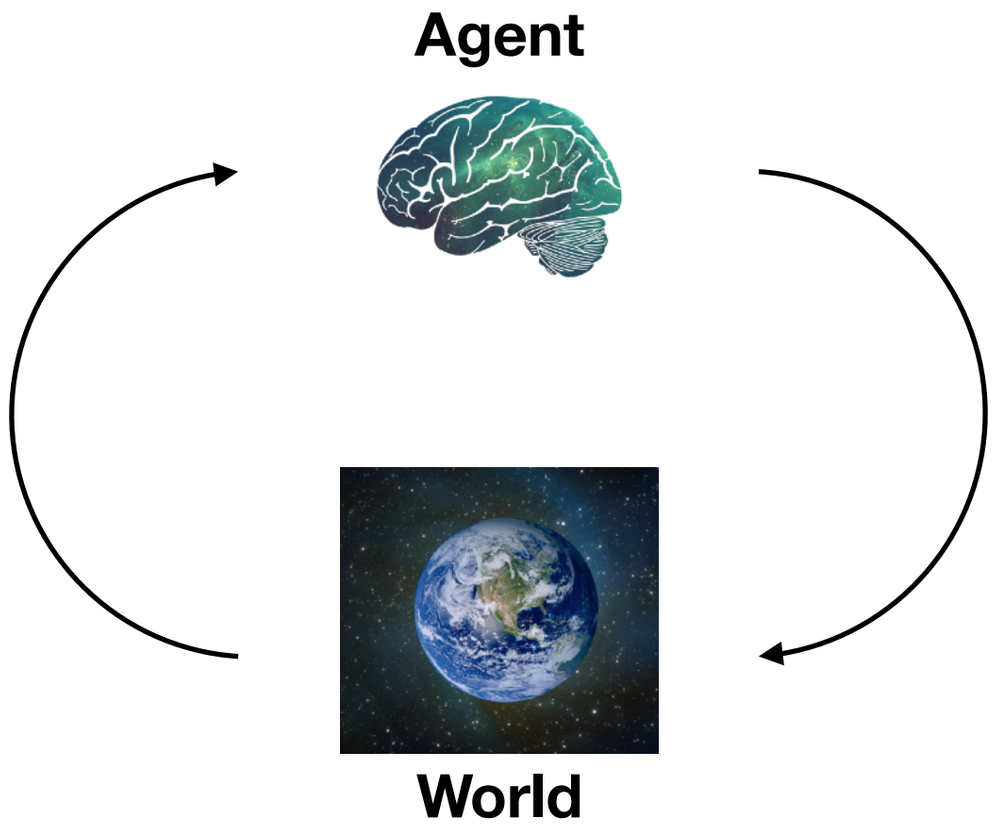
- Goal: Find actions to maximise a total expected future reward
- Possible balance of immediate and long-term reward
- Strategy may be needed to achieve high reward

Example: Household robots



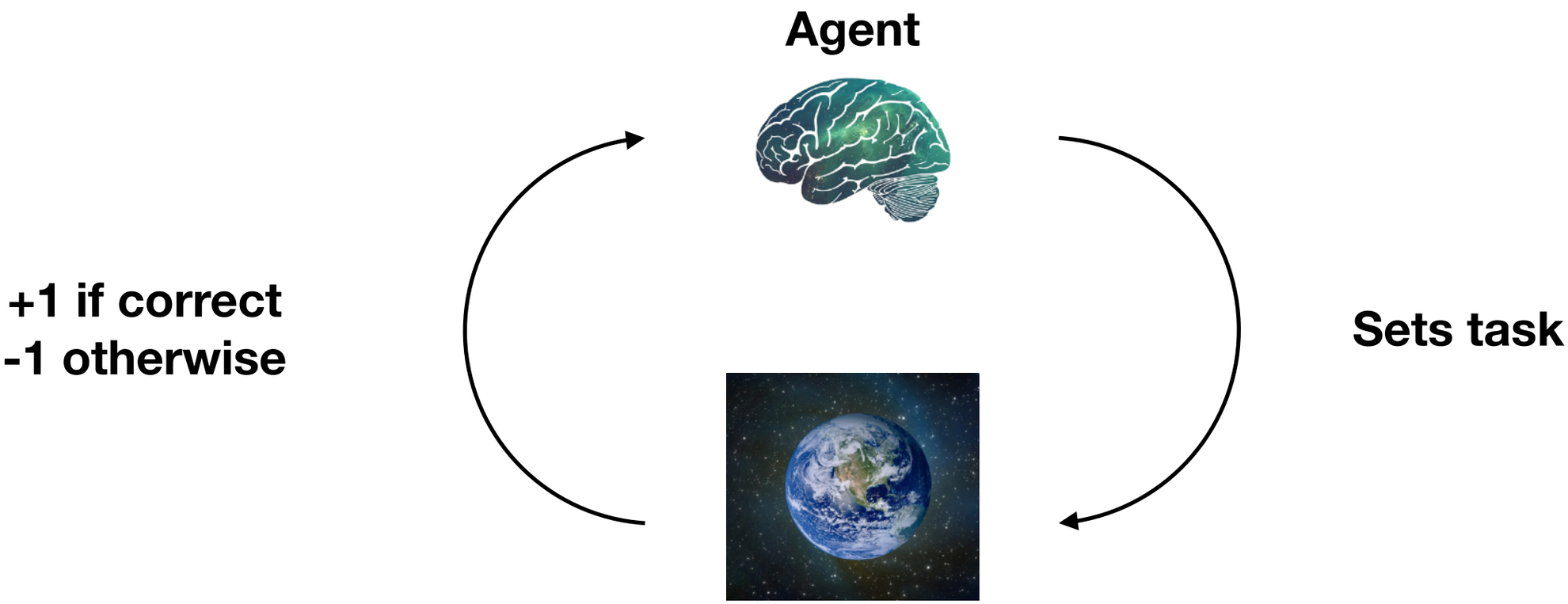
- Goal: Find actions to maximise a total expected future reward
- Possible balance of immediate and long-term rewards
- **Strategy may be needed to achieve high rewards**

Example: Blood pressure regulation



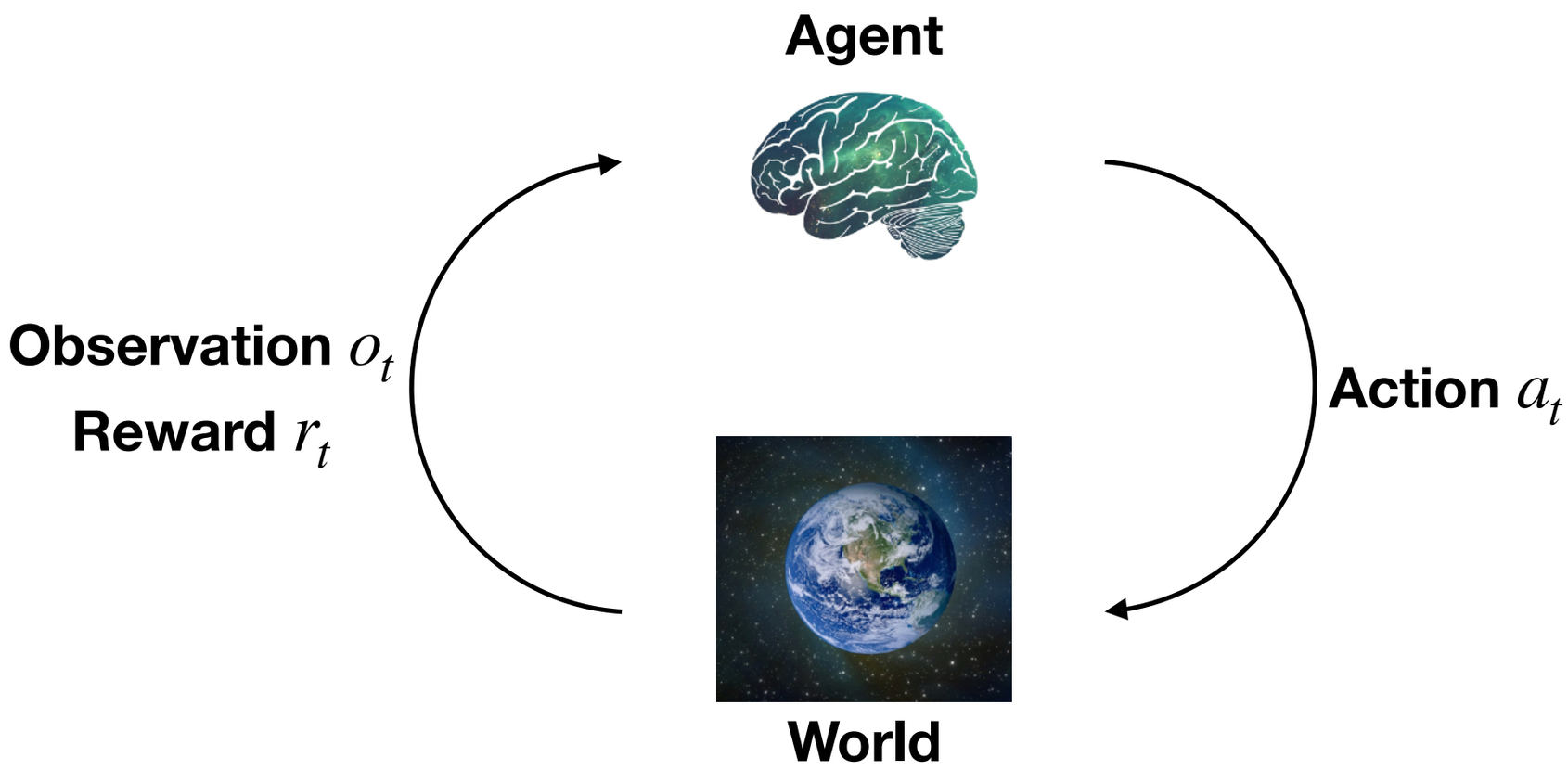
- Goal: Find actions to maximise a total expected future reward
- **Possible balance of immediate and long-term rewards**
- **Strategy may be needed to achieve high rewards**

Tutor application



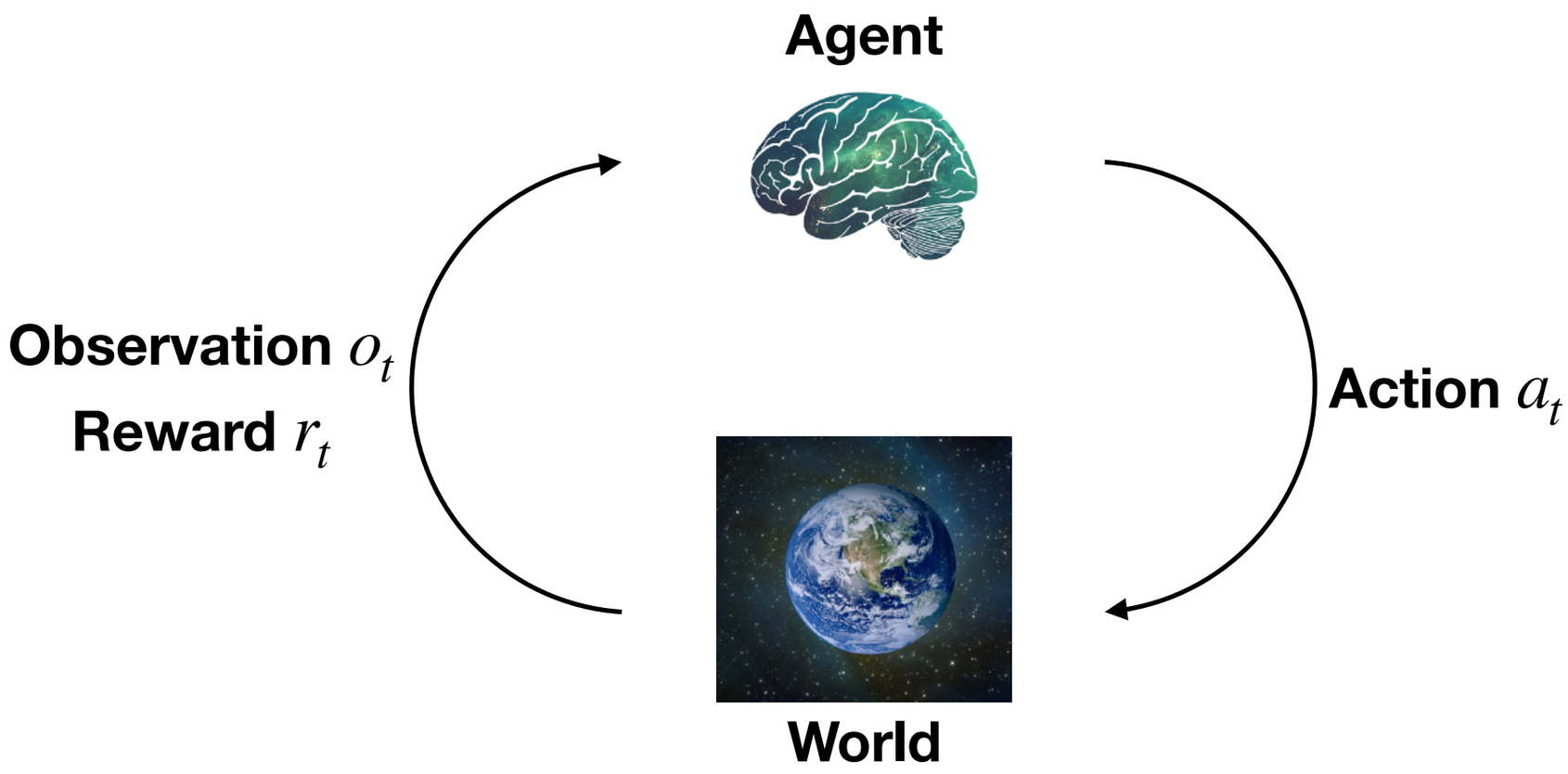
- Pupil should learn addition (easier) and subtraction (harder)
- Teacher agent can give tasks for addition or subtraction
- The agent receives a reward +1 if the student solves the task correctly, -1 otherwise

Sequential decision making: the agent and the world (discrete time)



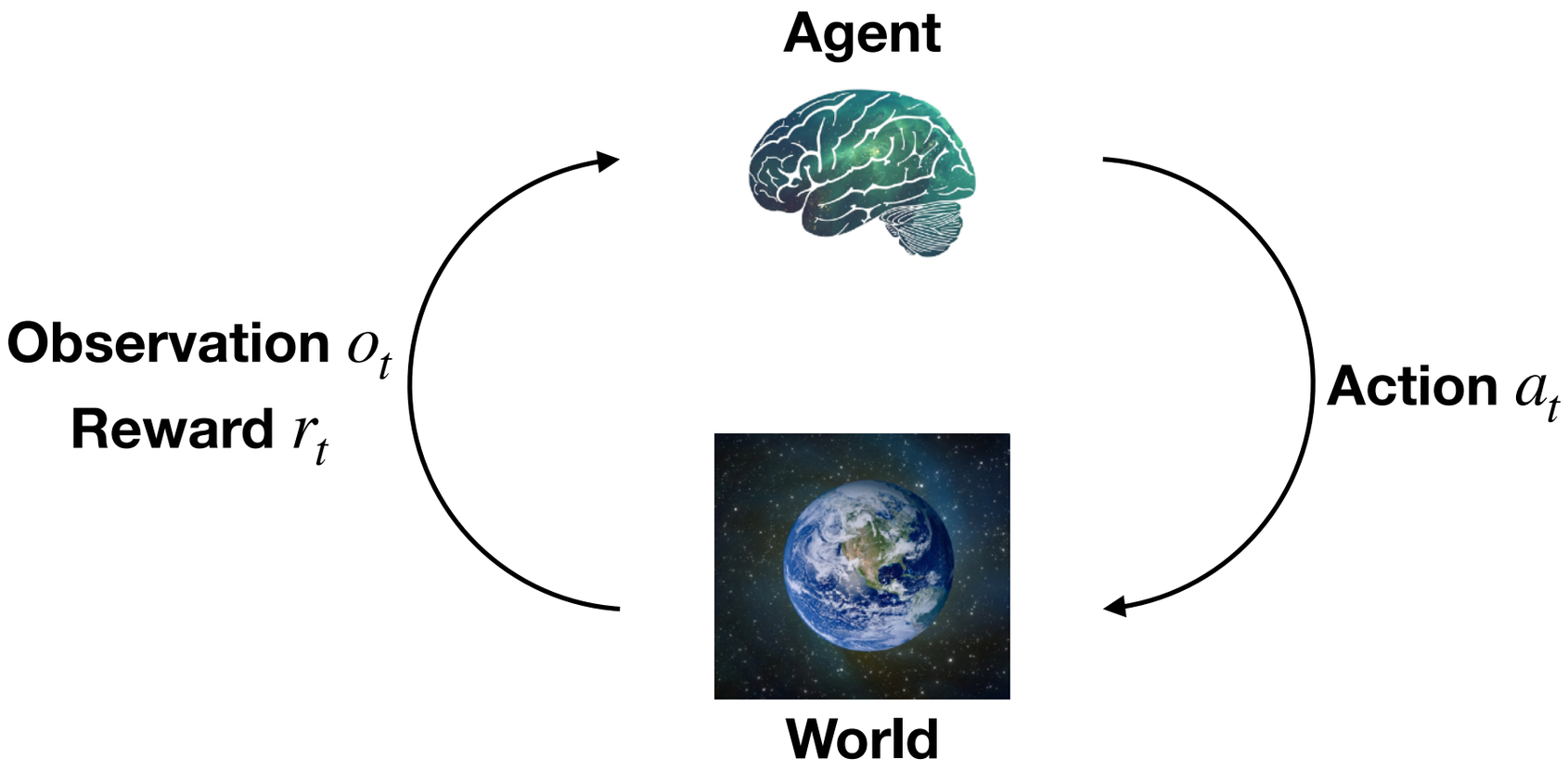
- Each time step t:
  - ➡ The agent performs an action a_t
  - ➡ The environment/world is updated and emits an observation o_t and a reward r_t
  - ➡ The agent receives the observation o_t and the reward r_t

History: Sequence of past observations, actions and rewards



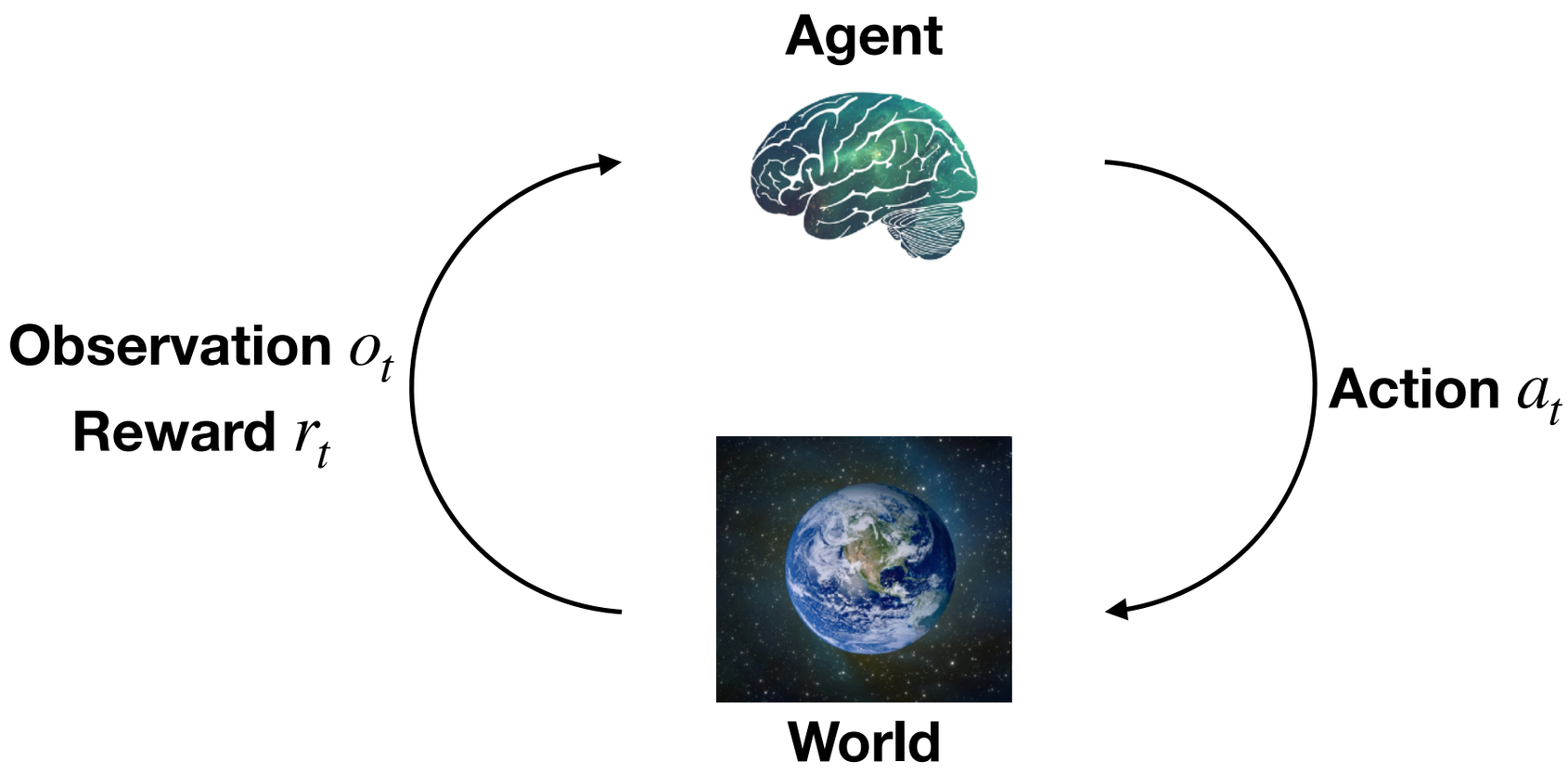
- The history $h_t = (o_0, a_1, o_1, r_1, \ldots a_t, o_t, r_t)$
- Agent selects an action based on h_t
- State s_t information is sufficient to determine what happens next: $s_t = f(h_t)$

World State



- The true state of the environment/world which determines how the environment generates the next observation
- **Mostly hidden from or unknown to the agent**
- Even if known, maybe not all needed by the agent

Internal representation of the agent (agent state)



- What the agent/algorithm uses to decide how to set actions
- In general, this is a function of history: $s_t = f(h_t)$
- Can provide meta-information such as the status of the algorithm (number of calculations performed, etc.) or the decision-making process (how many steps are left until the end of the episode)

Markov assumption

- Information state: sufficient statistics of the history
- State s_t is Markov if and on $p(s_{t+1} \mid s_t, a_t) = p(s_{t+1} \mid h_t, a_t)$
- The future is independent of the past given the present

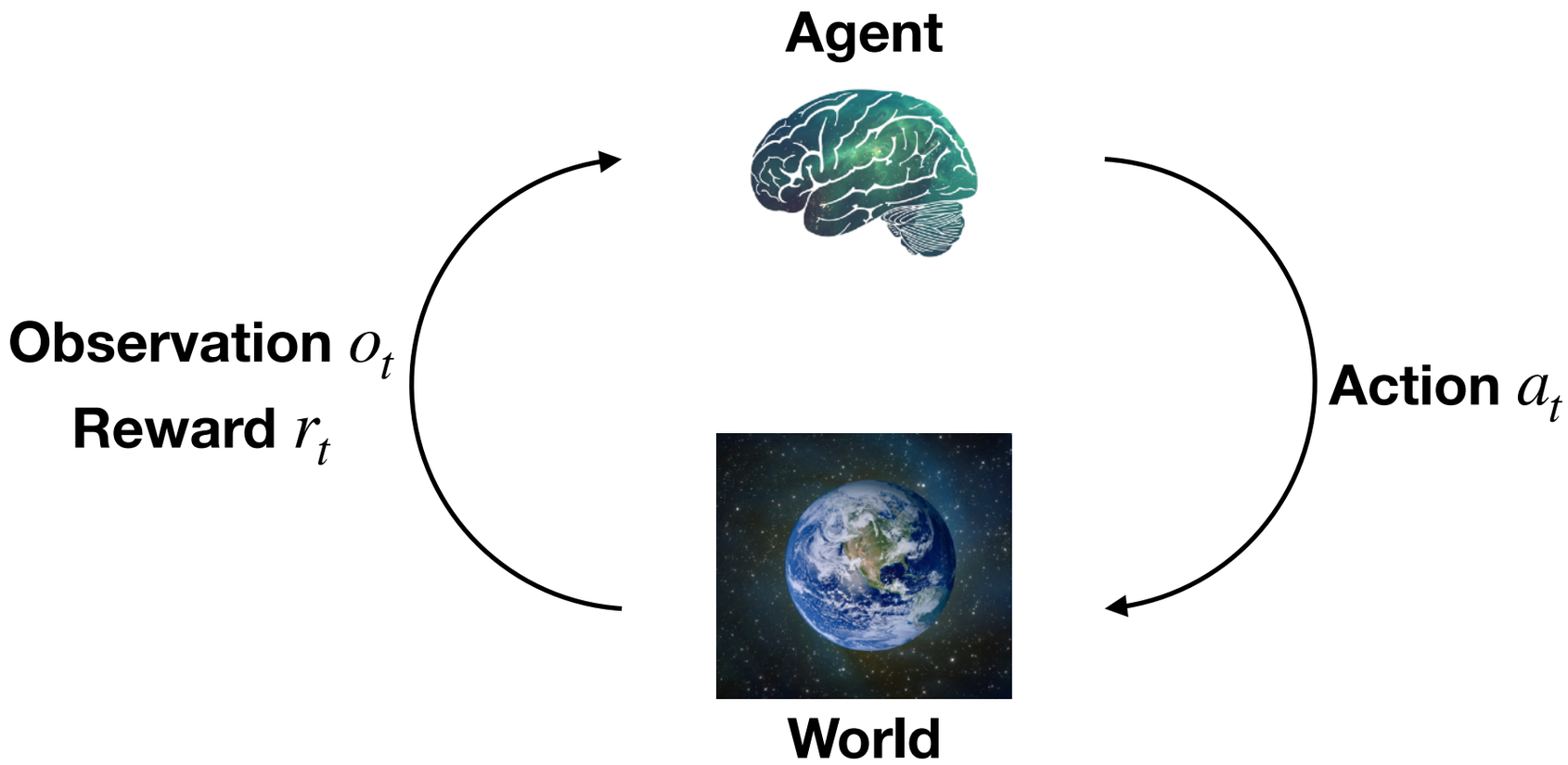
Markov assumption

- Information state: **sufficient statistics of the history**
- State s_t is Markov if and on $p(s_{t+1} \mid s_t, a_t) = p(s_{t+1} \mid h_t, a_t)$
- The future is independent of the past given the present
- High blood pressure control: s_t - current blood pressure, a_t - Medication or not. Markov system?
- Website shopping: s_t currently viewed product, a_t which other product do you recommend? Markov system?

Popularity of the Markov assumption

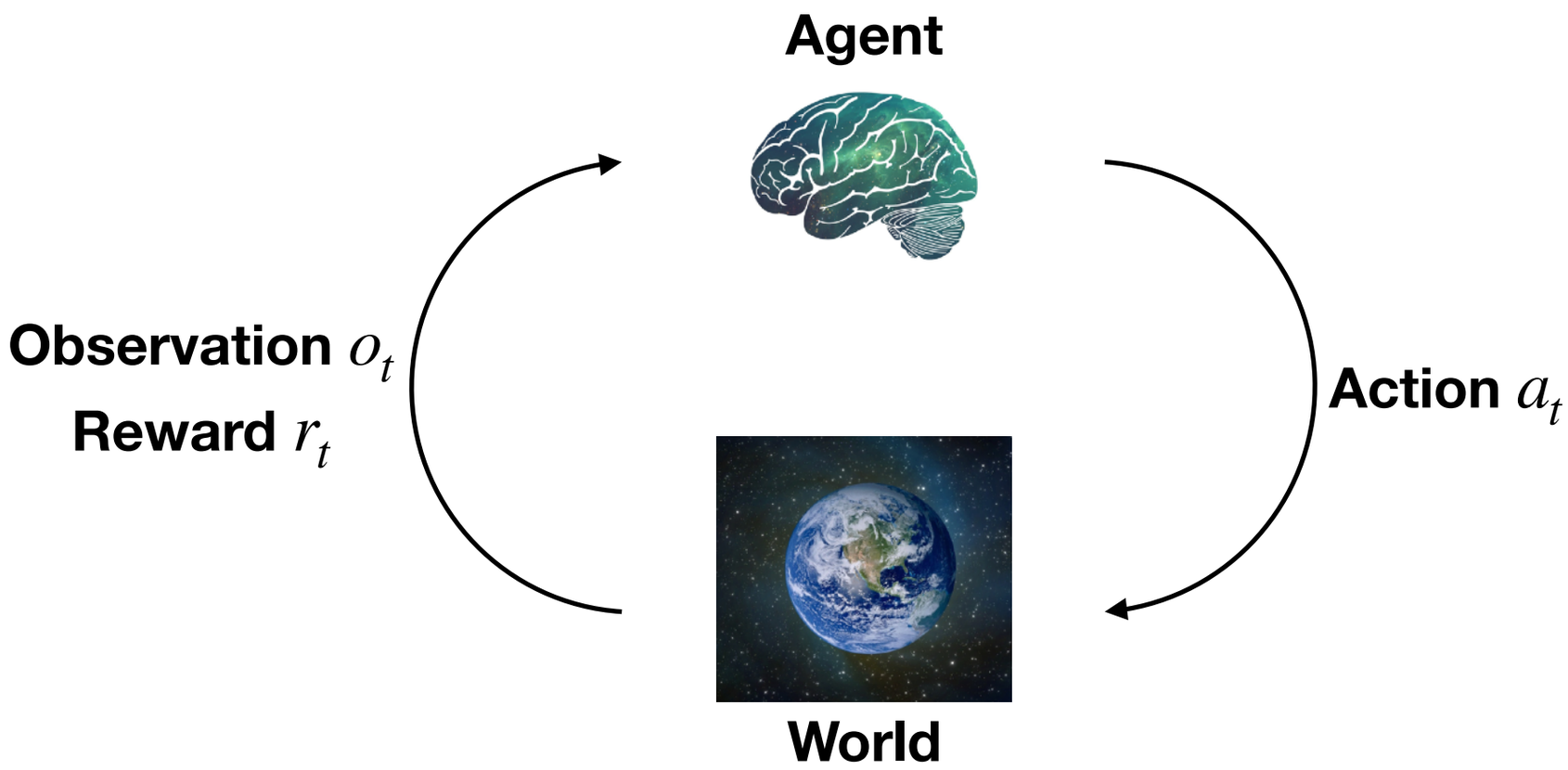
- **Can always be fulfilled**
  - ➡ Take history as a state: $s_t = h_t$
- In practice, it is often assumed that the last observation provides sufficient historical statistics: $s_t = o_t$
- State representation has a big influence on:
  - ➡ Computational complexity
  - ➡ Required data
  - ➡ Resulting performance

Full observation (fully observable) Markov decision process (MDP)



- Observed state is state of the world: $s_t = o_t$

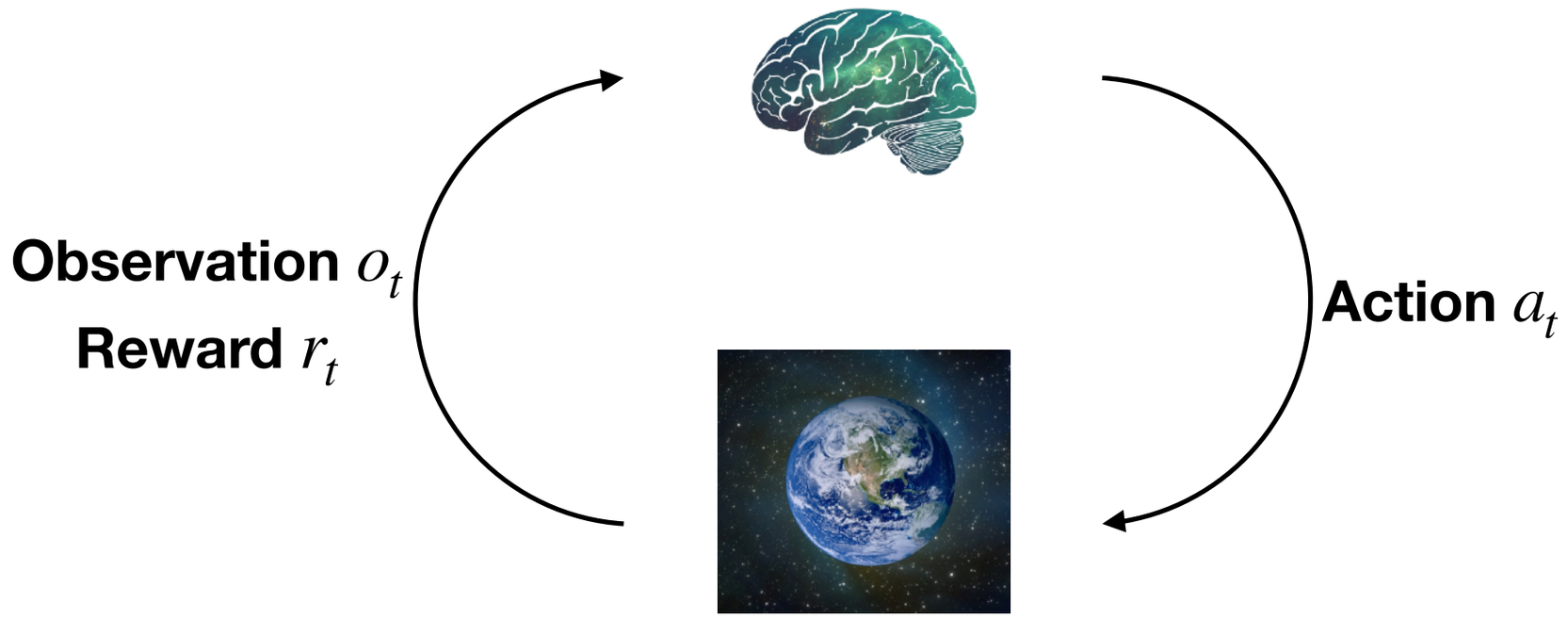
Partial observation (partially observable) POMDP



- Agent state is not world state
- Agent constructs own state e.g:
  - ➡ Using history $s_t = h_t$, or guessing the state of the world, or RNNs

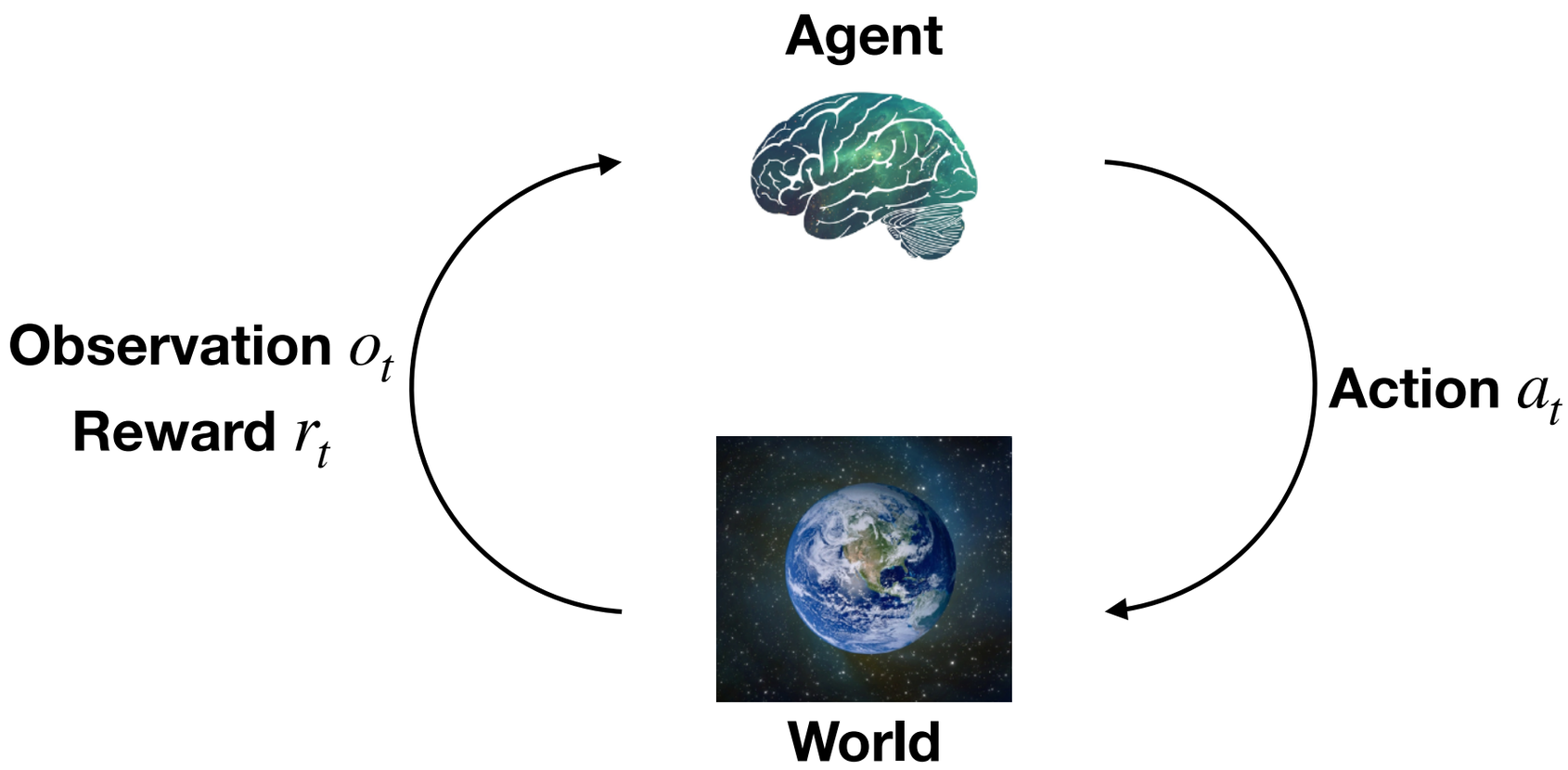
Example POMDP

- Poker player: only has his own hand
- Healthcare: not all psychological processes accessible



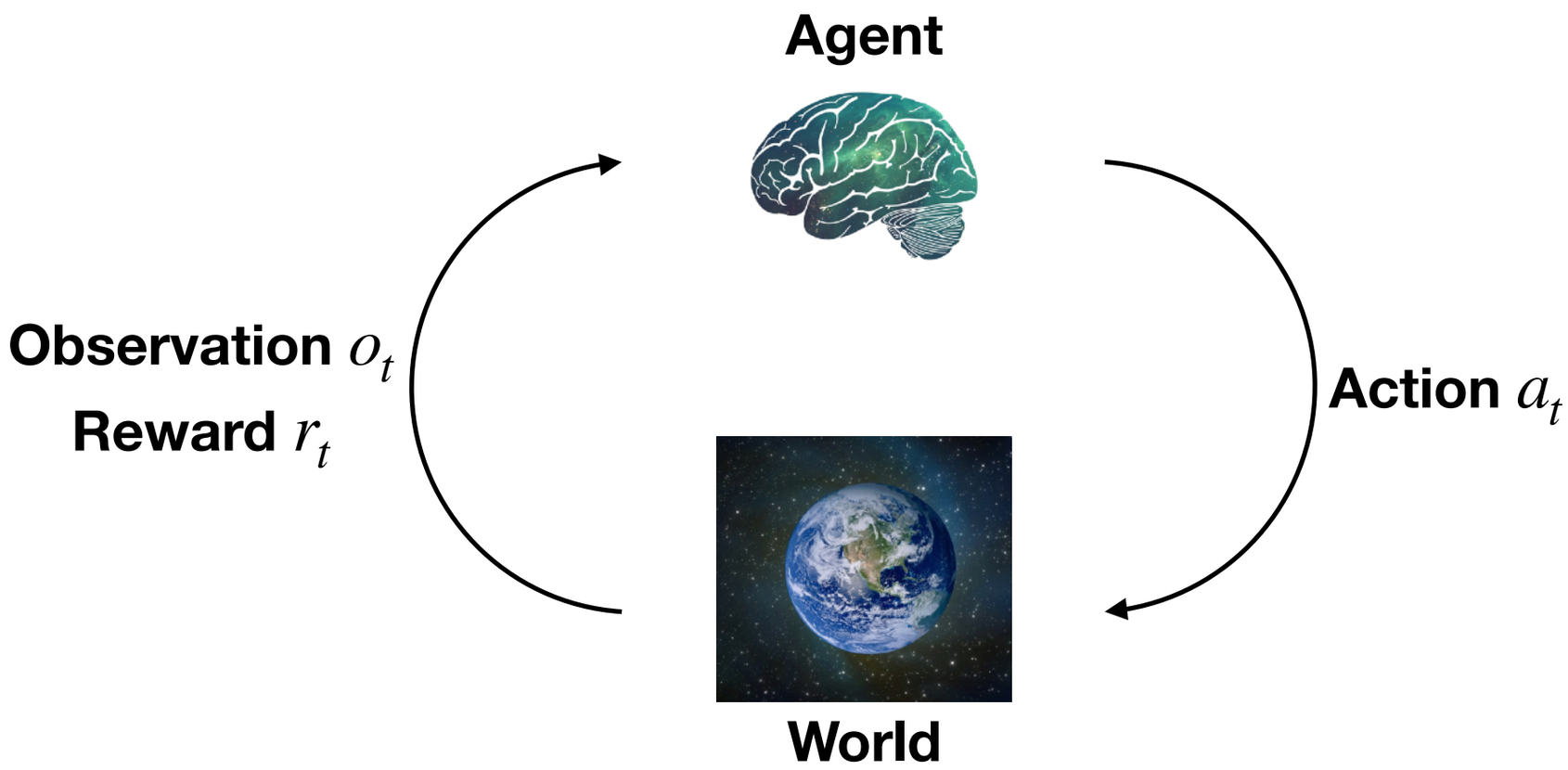
- Agent state is not world state
- Agent constructs own state e.g:
  - ➡ Using history $s_t = h_t$, or guessing the state of the world, or RNNs

Types of sequential decision-making processes: Bandits



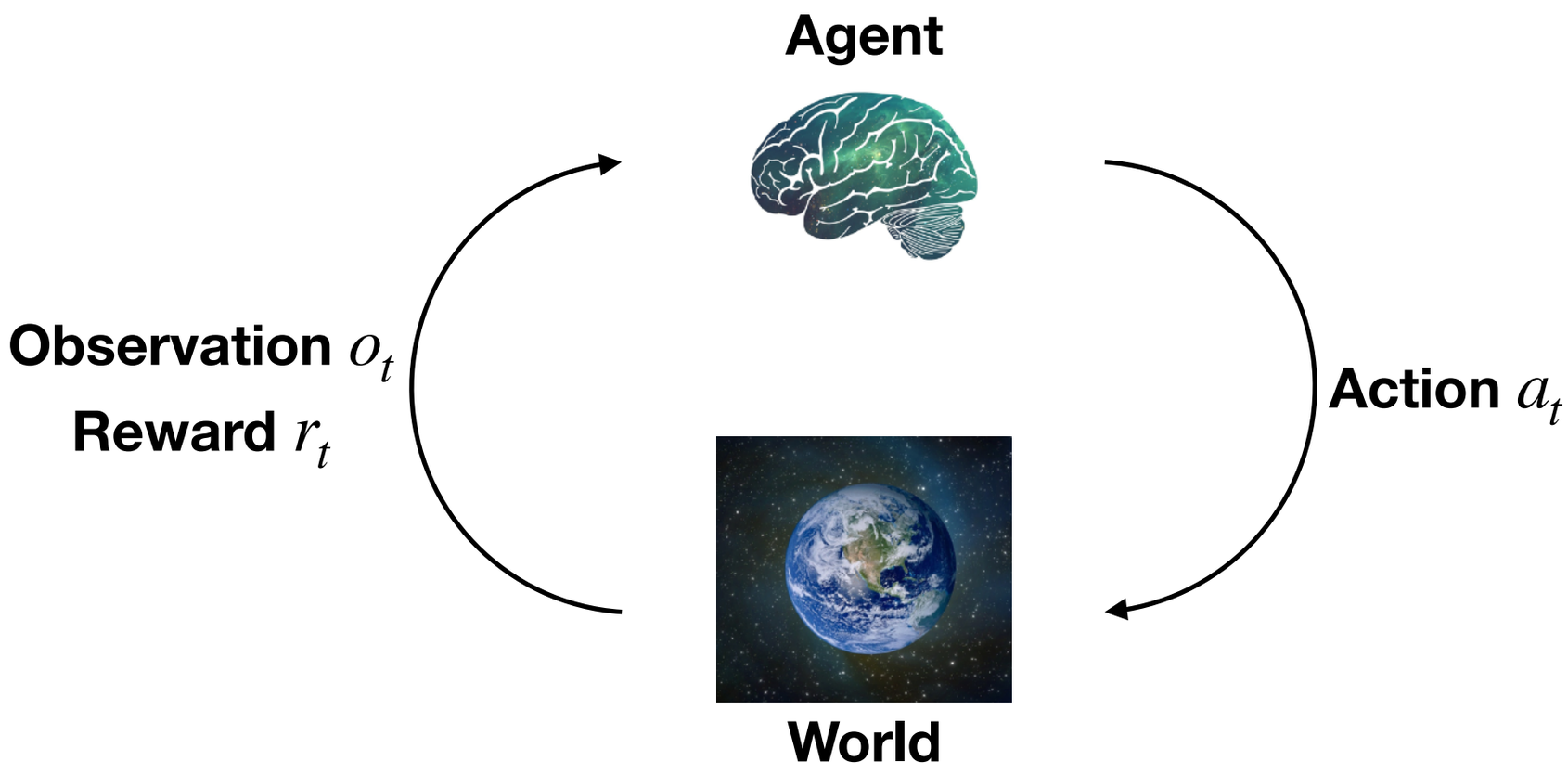
- Bandits: Actions have no influence on the next observation
- There are no delayed rewards

Types of Sequential Decision Processes: MDPs and POMDPs



- Actions influence the future
- Recognition of the respective contributions (credit assignment) and strategic actions may be required

Types of sequential decision-making processes: How the world is changing



- Deterministic: history and action is given, single observation and reward
  - ➡ General assumption in robotics and control problems
- Stochastic: history and action is given, many observations and rewards
  - ➡ General assumption for clients, patients, areas that are difficult to model

Example: Perseverance as a special MDP

https://mars.nasa.gov/mars2020/

s_1	s_2	s_3	s_4	s_5	s_6	s_7
						

- States: Location of the rover $(s_1, \ldots, s_7)$
- Actions: Movement to the left or right
- Rewards:
  - ➡ +1 in state s_1
  - ➡ +10 in state s_7
  - ➡ 0 otherwise

RL Algorithm components

RL Algorithm components

- Often includes one or more of:
  - ➡ **Model**: Representation of how the world changes after setting an action.
  - ➡ **Policy**: function from the agent state to the action
  - ➡ **Value function**: Future rewards from a state or state/action when following a certain policy

Model

- **Internal representation** of the agent how the world changes with actions
- **Transitions/dynamics model** predicts next state

$$p(s_{t+1} = s' \mid s_t = s, a_t = a)$$

- **Reward model** predicts immediate reward

$$r(s_t = s, a_t = a) = \mathbb{E}[r_t \mid s_t = s, a_t = s]$$

Example Mars Rover

s_1	s_2	s_3	s_4	s_5	s_6	s_7
$\hat{r} = 0$	$\hat{r} = 0$	$\hat{r} = 0$	$\hat{r} = 0$	$\hat{r} = 0$	$\hat{r} = 0$	$\hat{r} = 0$

- The numbers reflect the agent's reward model
- Part of the agent's dynamic model:
  - ➡ $0.5 = P(s_1 \mid s_1, \text{right}) = P(s_2 \mid s_1, \text{right})$
  - ➡ $0.5 = P(s_2 \mid s_2, \text{right}) = P(s_3 \mid s_2, \text{right}) \ldots$
- Model can be wrong

Policy - Strategy

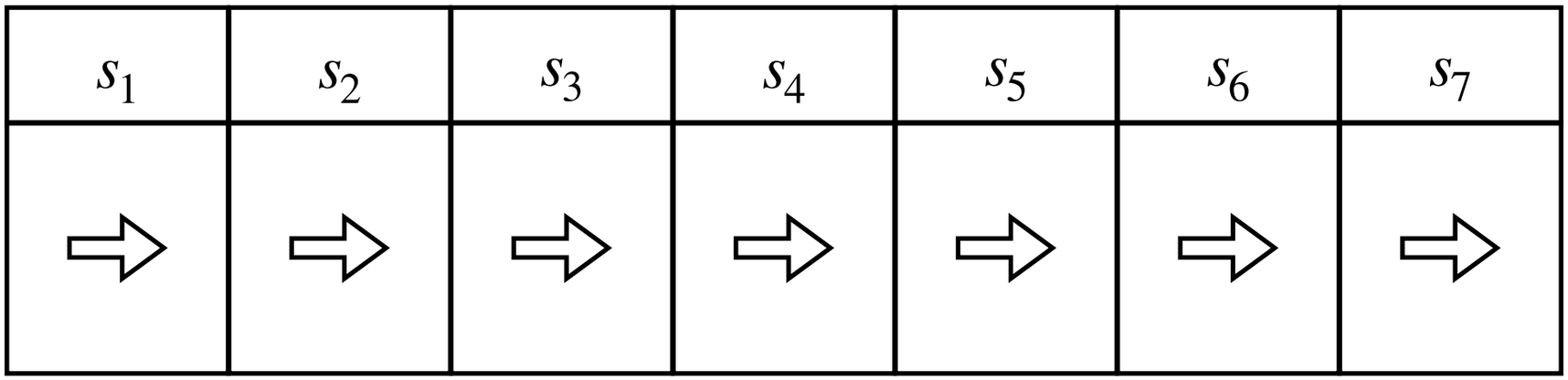
- Policy π determines how the agent sets actions
- $\pi : S \mapsto A$, maps from the states to the actions
  - ➡ Deterministic:

$$\pi(s) = a$$

  - ➡ Stochastic:

$$\pi(a \mid s) = Pr(a_t = a \mid s_t = a)$$

Example Mars Rover



- Policy is shown as an arrow
- $\pi(s_1) = \pi(s_2) = \ldots = \pi(s_7) =$ right
- Is this policy deterministic or not?

Value function

- Value function V^{π} is the **expected, discounted sum of future rewards** under a specific policy π

$$V^{\pi} = \mathbb{E}_{\pi}[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 r_{t+3} + \ldots | s_t = s]$$

- $\gamma \in [0,1]$ is the discount factor that controls the range from immediate to future rewards
- Can be used to quantify how good/bad a condition or action is
- For decision-making when policies are compared

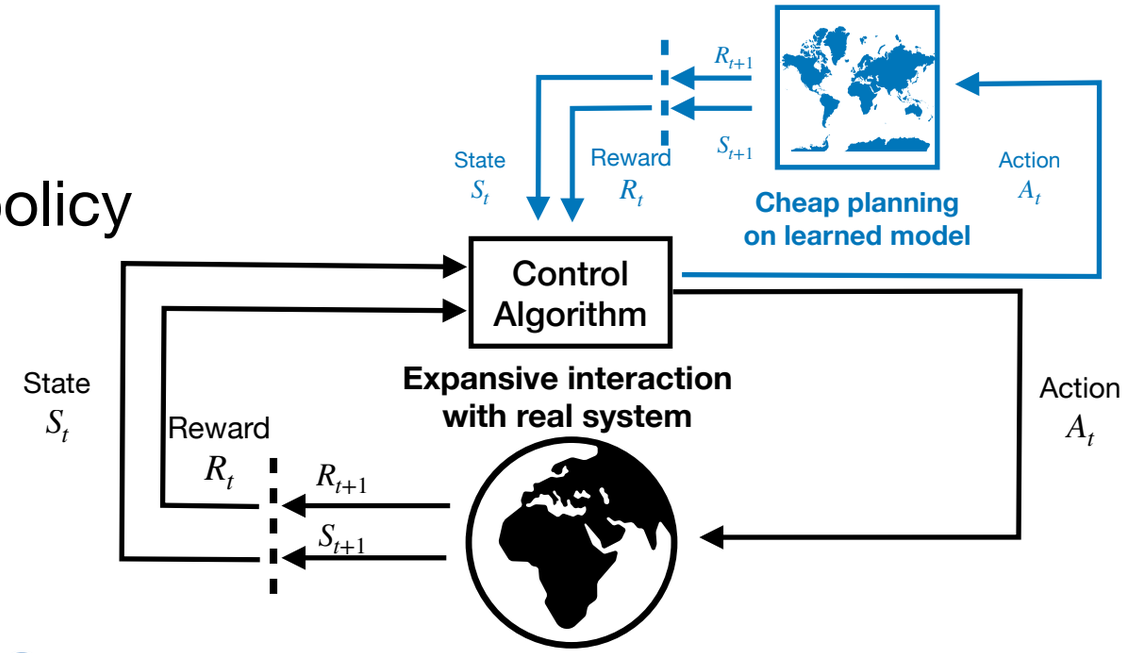
Example Mars Rover

s_1	s_2	s_3	s_4	s_5	s_6	s_7
$V^\pi(s_1) = 1$	$V^\pi(s_2) = 0$	$V^\pi(s_3) = 0$	$V^\pi(s_4) = 0$	$V^\pi(s_5) = 0$	$V^\pi(s_6) = 0$	$V^\pi(s_7) = 10$

- Discount factor $\gamma = 0$
- $\pi(s_1) = \pi(s_2) = \ldots = \pi(s_7) =$ right
- The numbers show the value of $V^\pi(s)$ for this policy π and this γ

Types of agents

- Model-free (model-free)
  - ➡ Explicit: Value function and/or policy
- Model-based
  - Explicit: Model
  - There may or may not be a policy and a value function



Other important aspects (optional)

Main difficulties in learning to make good decision sequences

- Planning (internal calculation of the agent)
  - ➡ Model of how the world works exists
    - Dynamics and reward model
  - ➡ Algorithms calculate how actions are set to maximise the expected reward
    - No Interaction with the real environment
- Reinforcement Learning
  - Agent does not know how the world works
  - Interacts with the world to implicitly/explicitly learn how it works
  - Agent improves its policy (may involve planning)

Planning example

- Solitaire: Single player card game
- All rules known/perfect model
- If you set an action a in the state s
  - ➡ Possibility to calculate the probability distribution of subsequent states
  - ➡ Possibility to calculate a potential score
- Agent can plan ahead to make an optimal decision
  - ➡ E.G.: Dynamic programming, tree search,...

Reinforcement learning example

- Solitaire without a rule book
- Direct learning by trying out actions and observing the subsequent states
- Try to find a policy over time (which leads to a high reward)

Exploration and exploitation

- Agent only learns what happens for the actions it performs
  - ➡ Mars rover trying to turn left learns the reward and next state for turning left but not for turning right
  - ➡ Obvious, but it leads to a dilemma

Exploration und exploitation

- **Agent only learns what happens for the actions it performs**
- How should the agent's actions divide up its actions?
  - ➡ Exploration: Trying out new things that may lead to better decisions in the future
  - ➡ Exploitation: Selection of actions from which a good reward is expected based on experience
- Usually a compromise between exploration and exploitation
  - ➡ Possible sacrifice of reward in order to learn a potentially better policy through exploration

Exploration and exploitation examples

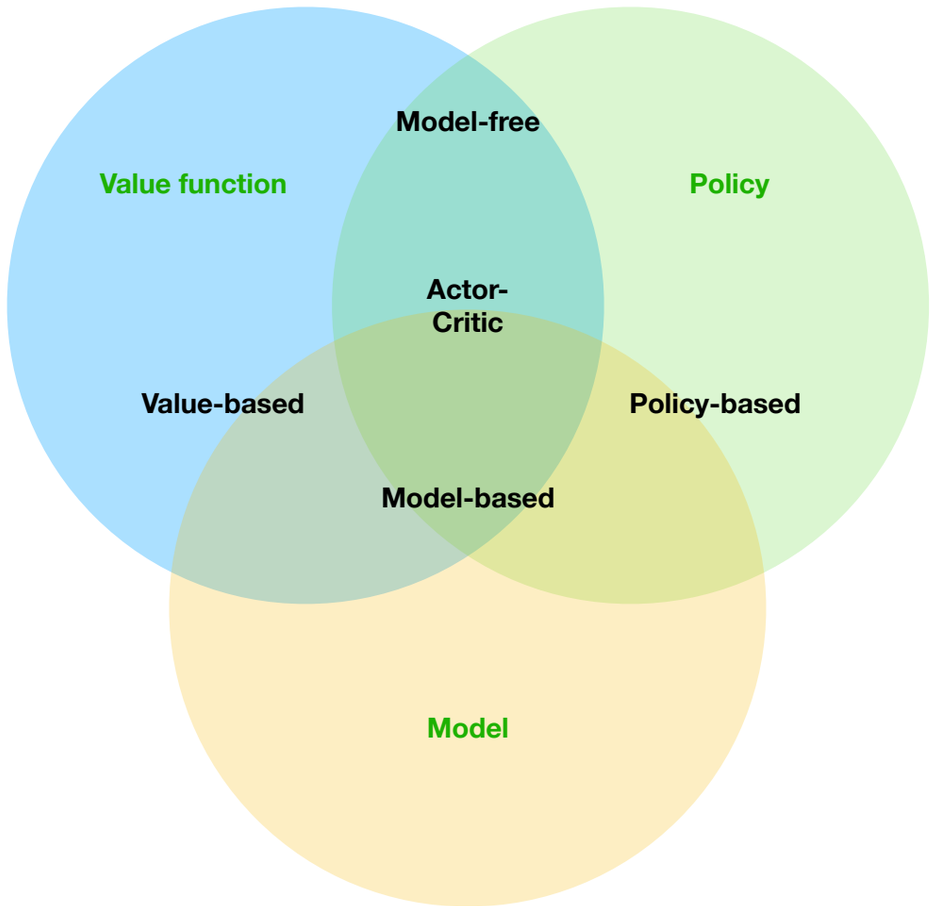
- Movies
  - ➡ Exploitation: Watching a movie you know and like
  - ➡ Exploration: Watching a new movie
- Advertising
  - ➡ Exploitation: Choose the commercial that works best
  - ➡ Exploration: Choose a different commercial
- Choosing a route when driving
  - ➡ Exploitation: Drive the fastest route based on previous experience
  - ➡ Exploration: Try another route

Evaluation and control

- Evaluation
  - ➡ Estimating/predicting the expected return under a given policy
- Control
  - ➡ Optimisation: finding the best policy

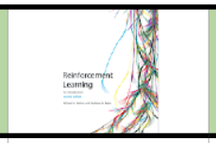
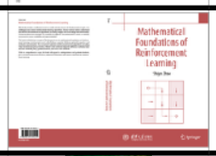
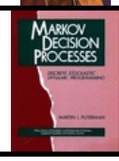
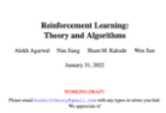
Types of RL agents: What the algorithm learns

- Model-based
  - ➡ Explicit: Model
  - ➡ Implicit: Policy and/or value function (model can be used for planning: calculate a policy and/or value function)
- Value-based
  - ➡ Explicit: Value function
  - ➡ Implicit: Policy (can be obtained from value function)
- Policy based
  - ➡ Explicit: Policy
  - ➡ No value function
- Actor-critical
  - ➡ Explicit: Policy
  - ➡ Explicit: Value function
- Algorithms can also have an explicit model, value function and policy



Resources

RL text books

Reinforcement Learning: An Introduction	http://incompleteideas.net/book/the-book-2nd.html	Sutton, Barto	2020		Reference	Free
Reinforcement Learning: An Overview	https://arxiv.org/pdf/2412.05265	Kevin P. Murphy	2025			Free
Mathematical Foundation of Reinforcement Learning	https://github.com/MathFoundationRL/Book-Mathematical-Foundation-of-Reinforcement-Learning	Shiyu Zhao	2025			Free
Deep Reinforcement Learning	Uni Bibliothek	Hao DongZihan Ding Shanghang Zhang	2020		State of the art	Free
Algorithms of Reinforcement Learning	https://sites.ualberta.ca/~szepesva/rlbook.html	Csaba Szepesvari	2019		More Mathematical	Free
Rollout, Policy Iteration, and Distributed Reinforcement Learning	http://www.athenasc.com/rolloutbook_athena.html	Dimitri P. Bertsekas	2020		For control people	
Markov Decision Processes: Discrete Stochastic Dynamic Programming		Martin L. Puterman	1996		Standart in Markov Decission Prozesses	
Reinforcement Learning: Theory and Algorithms	https://rltheorybook.github.io/rltheorybook_AJKS.pdf	Alekh Agarwal Nan Jiang Sham M. Kakade Wen Sun	2022		Theoretical	Free
ALGORITHMS FOR DECISION MAKING	https://algorithmsbook.com/#	MYKEL J. KOCHENDERFER, TIM A. WHEELER, AND KYLE H. WRAY	2022		State of the art	Free
Artificial Intelligence: A Modern Approach	http://aima.cs.berkeley.edu	Stuart Russell and Peter Norvig	2010		Standart in AI	Free

Some important libraries

Mainstream & Widely Used	Academic & Research-Focused
Stable Baselines3 (SB3) – Easy PyTorch implementations; great for prototyping & teaching	**Acme (DeepMind)** – Modular framework for reproducible, distributed RL research
Ray RLlib – Scalable, distributed RL for industry & production	**CleanRL** – Minimal, single-file implementations; ideal for learning & transparency
TF-Agents – Modular TensorFlow framework	**Garage** – Unified interface for many algorithms; benchmarking focus
Tianshou – Flexible PyTorch library, popular in applied & research work	**Sample Factory** – High-throughput training, multi-agent RL experiments

Library	Description	Examples / Scope
Gymnasium (successor of OpenAI Gym)	Standard API for RL tasks	Classic control (CartPole, MountainCar), toy text, Atari, Box2D
PettingZoo	Multi-agent extension of Gym API	Board games (chess, Go), card games, pursuit-evasion, cooperative/competitive multi-agent
Farama Foundation Ecosystem	Maintains and develops core RL environment libraries	Gymnasium, PettingZoo, Minigrid, Miniworld → *de facto standard for RL*

Summary and key takeaways

Summary and key takeaways

- Summary
  - ➡ RL = learning to act, not just to predict
  - ➡ Formalised by MDPs, made powerful with deep learning
  - ➡ RL enables strategies beyond human intuition (e.g. AlphaGo's Move 37)
  - ➡ Used today in games, robotics, industry, healthcare, generative AI

- Key Takeaways
  - ➡ Strengths: trial & error, autonomy, adaptability, discovery
  - ➡ Challenges: reward design, sample efficiency, stability, safety, transfer
  - ➡ RL is unique in AI: it turns data into decisions
  - ➡ This bootcamp will take you from basics → algorithms → applications