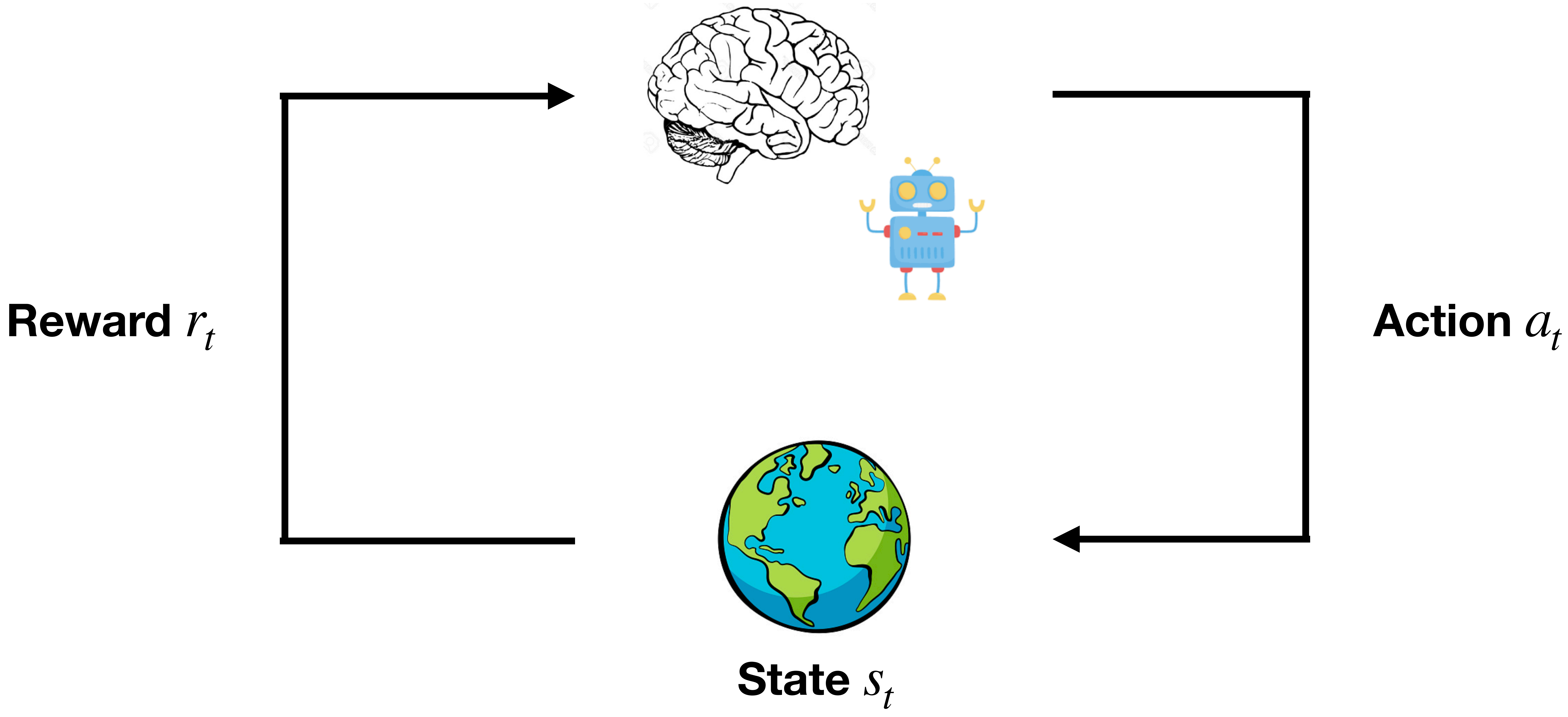
An introduction to Reinforcement Learning

26nd of April 2022

Philipp Schwartenbeck

AI Center, University of Tübingen

Recap: Basic setup of RL



Recap: What is reinforcement learning (RL)?

- RL is a **computational approach** to learning from **interactions** with the **environment**
  - Trial-and-error
  - Delayed reward
- Considers whole problem of **goal-directed** agent interacting with an **uncertain** environment
- Three main machine learning approaches
  - Supervised
  - Unsupervised
  - RL
- Very general account

Where are we?

Intro

Intro (cont)
Theories of Learning
- **Psychology, behaviour**
- **Rescorla-Wagner Learning**

Theories of Learning
- Neuroscience
- TD learning

Markov Decision Processes

Theories of control, action selection

Model-free and model-based RL
Exploitation vs. Exploration

Some coding
- Role of different parameters
- Model-fitting
- If possible: parameter recovery, model comparison

- 'Advanced' topics and current applications
  - Planning, Dyna, replay
  - Clever ways of planning, tree-search etc
  - Deep RL
  - Other current fancy developments

Recap This seminar: components

- Most of this is first time material - tell me if something doesn’t work, open for suggestions
  - Especially for second half of the seminar
- Structure
  - Theory (key reference: Sutton & Barto, 1998)
  - Research (key papers)
  - Coding (Python)
- Missing May date (24th of May): coding
- Grading: essay
  - Tell me if you would like to have additional grading during the term (coding exercise/s, presentation)

Basics of (Reinforcement) Learning

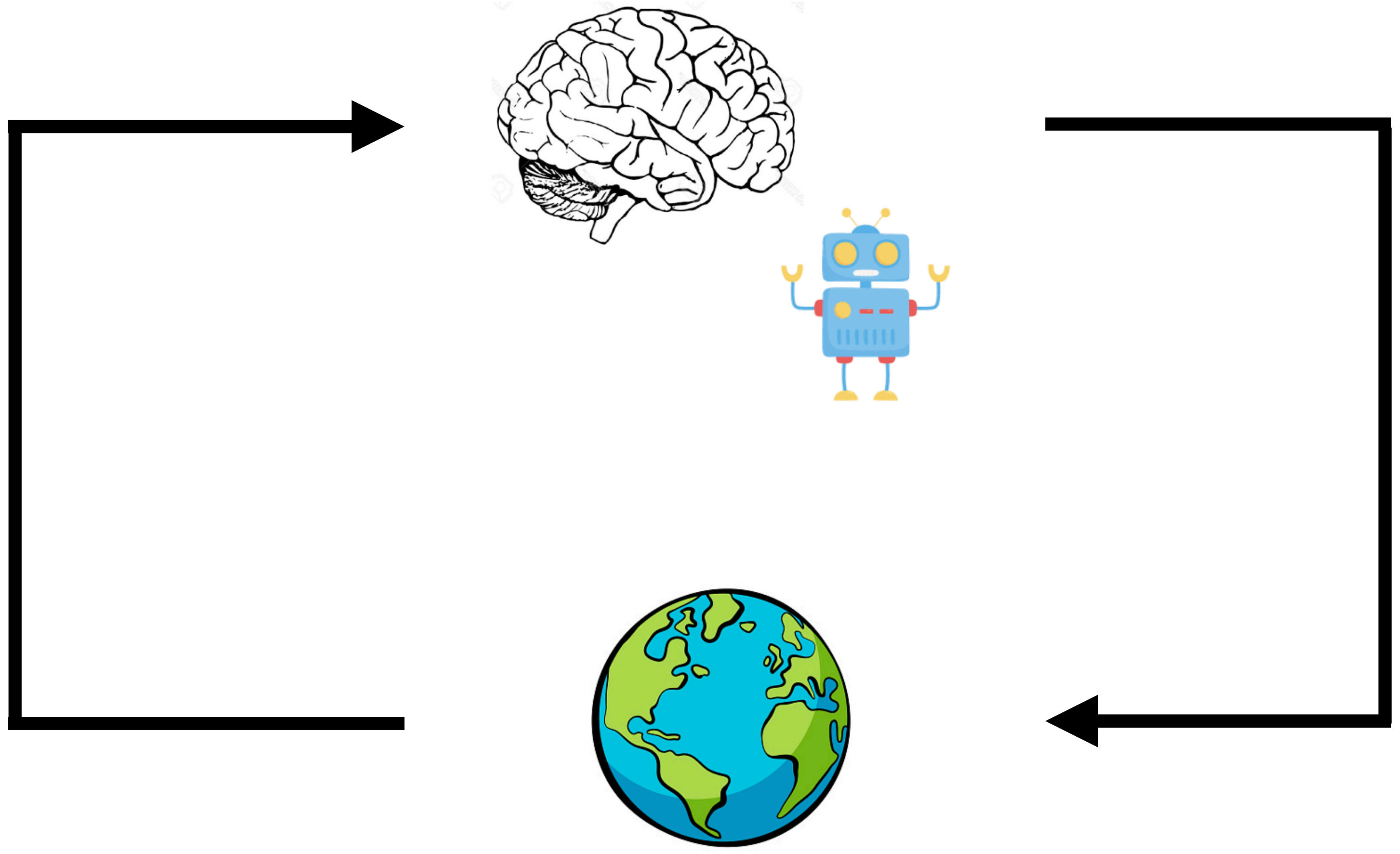
Basic setup: how to agents learn to act?

Based on a reward signal, agents learn **values of actions/states:**

$$V_\pi(s) = \mathbb{E}_\pi[R \mid s_0 = s]$$

Agents can learn a **model of the environment** to make smarter decisions, e.g.:

$$P(s_{t+1} = s \mid s_t = s, a_t = a)$$



Values can be **learnt** (simplified!!):

$$V(s) \leftarrow V(s) + \alpha \cdot (r - V(s))$$

Learning rate

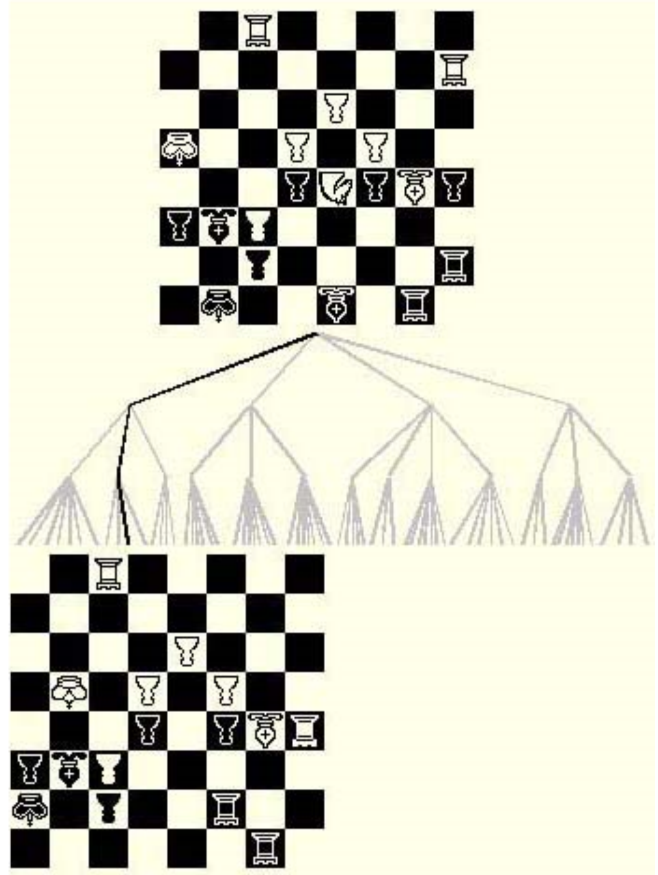
Prediction error

Action is governed by a **policy**:

$$\pi(a, s) = P(a_t = a \mid s_t = s)$$

(More) Examples

- **Chess**: what is…
  - The state?
  - An action?
  - A reward?
- How can values be learned over time?
- How could a model of the environment be useful?



Other relevant components:

- tree search
- position evaluation
- situation memory

Taken from Peter Dayan

(More) Examples

- **Learn how to walk**: what is…
  - The state?
  - An action?
  - A reward?
- How can values be learned over time?
- How could a model of the environment be useful?

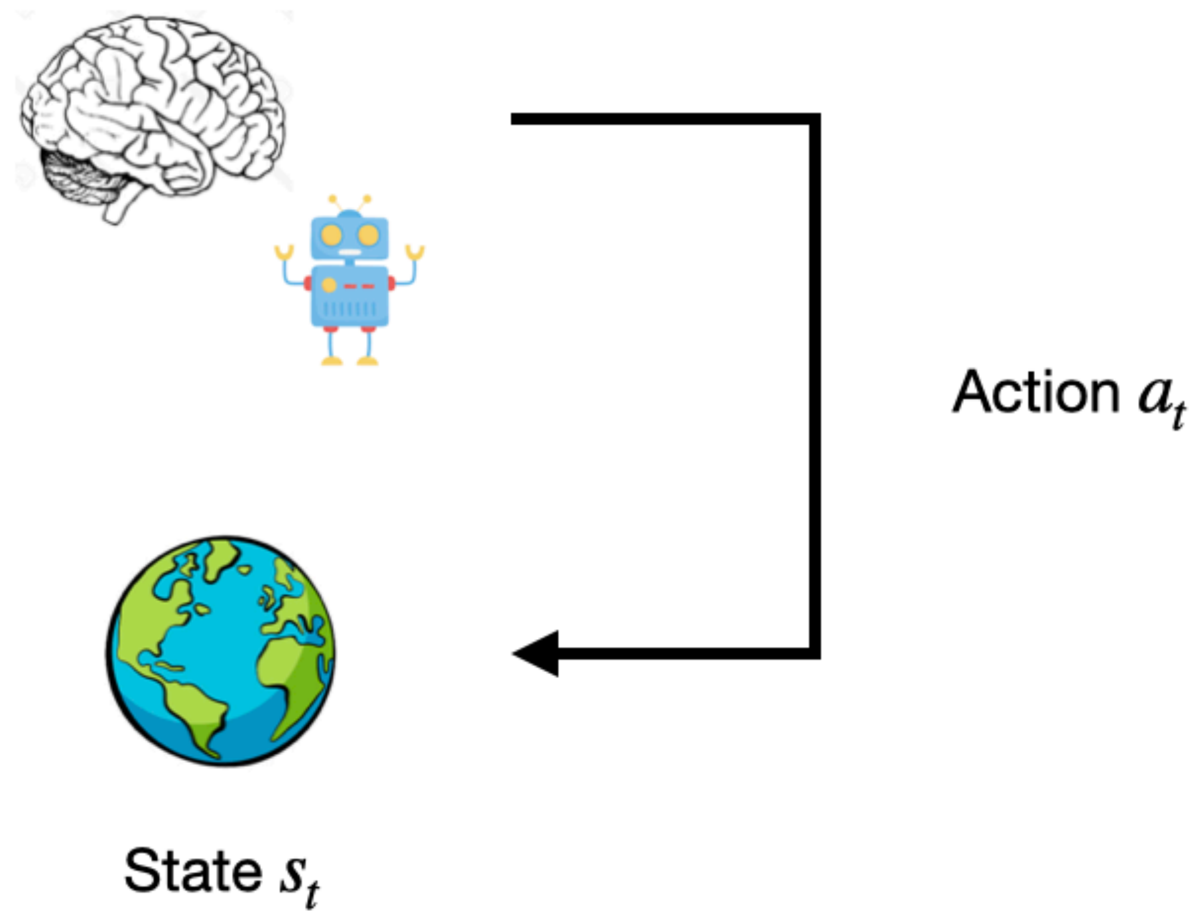
Examples extended..

- Other examples (see Sutton & Barto, pp 4-5):
- **Adaptive controller** adjusts parameters of a petroleum refinery's operation in real time
  - Optimise yield/cost/quality trade-off
  - Objective: specified marginal costs
  - Without sticking strictly to pre-defined set points
- **Mobile robot** decides to search for trash to collect or find its way back to battery recharging station
  - Decision based on current charge level of battery and how quickly recharger has been found in the past.
- **Prepare breakfast**
  - Subgoals, hierarchies
  - Conditional behaviour
  - Sense/access bodily states

Key features of all these examples

- (Danger of repeating myself): **Interaction** between active decision-making agent and its environment
  - Agent seeks to achieve a **goal**
  - **Uncertainty** about its environment
- Take into account **indirect, delayed consequences** of actions
  - Requires foresight or planning
- Need to **monitor** environment frequently
- **Judge progress** toward goal based on what can be sensed directly
- Use experience to **improve performance** over time (online vs. offline learning)
  - Basis for adjusting behaviour to exploit specific features of the task

Key Elements: Policy



- Defines agent's **way of behaving** at a given time
  - Sufficiently determines behaviour
  - Often stochastic (determine probabilities for each action)

- Mapping from (perceived) states of environment to actions
  - Cf., stimulus–response rules or associations

- Can be simple or difficult
  - Lookup table vs. extensive search

Action is governed by a **policy**:

$$\pi(a, s) = P(a_t = a \,|\, s_t = s)$$

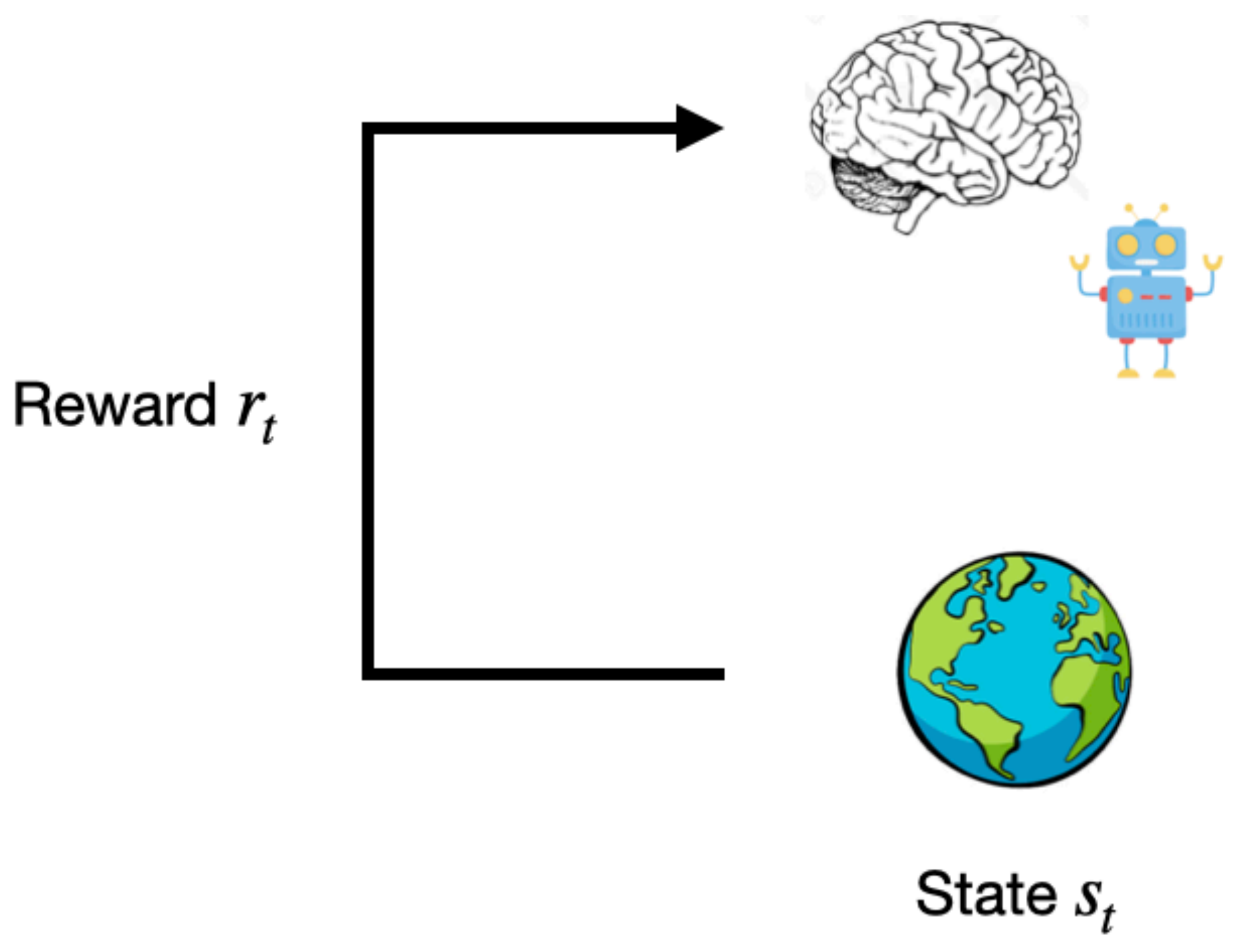
Key Elements: Reward signal

- = **goal** of a reinforcement learning problem
  - Single number from environment on each time-step

- SOLE objective is to maximise the total reward over the long run

- Primary basis for altering the policy
  - If action is followed by low reward, then the policy may be changed

- Often: stochastic function of state of environment and actions taken



Key Elements: Value function

Based on a reward signal, agents learn **values of actions/states:**

$$V_\pi(s) = \mathbb{E}_\pi[R \mid s_0 = s]$$

- = what is good in the **long run**
- Total amount of reward an agent can expect to accumulate over the future, starting from a given state
  - Long-term desirability of states
  - Taking into account states that are likely to follow and rewards available in those states
- A state might yield a **low immediate reward** but can still have a **high value**
  - Because regularly followed by other states that yield high rewards

Reward vs. Value

- Rewards = primary, values = secondary
  - Rewards are the basis for estimating value

Reward r_t

$$V_\pi(s) = \mathbb{E}_\pi[R \mid s_0 = s]$$

- BUT action selection is based on value, not immediate reward - why?

- It is more difficult to determine value than reward
  - Methods for value estimation are a central problem in RL

Key Elements: Model of Environment

Agents can learn a **model of the environment** to make smarter decisions, e.g.:

$$P(s_{t+1} = s \mid s_t = s, a_t = a)$$

- Mimics **behaviour of environment**
    - Allows to predict how the environment will behave
- E.g.: given a state and action, model can predict the resultant next state and next reward
- Models are used for planning
    - Considering possible future situations before they are actually experienced
- Methods for solving reinforcement learning problems that use models and planning are called model-based methods
    - Simpler model-free methods are explicitly trial-and-error learners (opposite of planning)

What are the limits of RL?

- How do we define a state? Are all states perceivable?
- What about problems that cannot be solved via learning (e.g. inference)?
- Is reward enough to explain behaviour/cognition/brains?

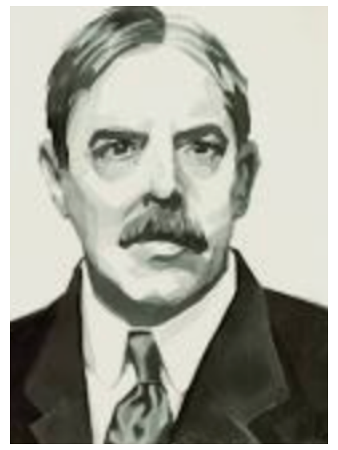
History and Theories of (Reinforcement) Learning

“Three” historical branches of RL

- Association learning, prediction (early 1900s)
- Optimal control (1960 onward)
- Learning *and* control (1980 onward)

Association learning, prediction

- Key problem: **trial and error** learning
  - Origins in animal learning
- Goes back to "**Law of effect**" (Edward Thorndike, 1911)



  - "The greater the satisfaction or discomfort, the greater the strengthening or weakening of the bond [between an animal's response and a situation]"
  - Also: Ivan Pavlov, Clark Hull, B.F. Skinner, Marvin Minsky, …
- See examples soon

History: Optimal Control

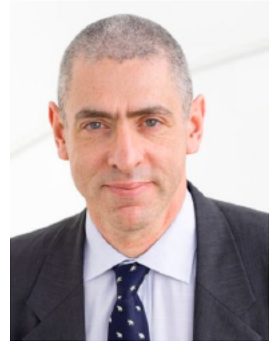
- Key problem: design **controller** to minimise/maximise a quantity
  - Based on value functions (e.g. dynamic programming)
- Not really *learning*
  - But incremental and iterative
- 1950s, Richard Bellman (**Bellman equation**)

- MDPs
  - Extension to ‘partial observability’: Leslie Kaelbling (1990s)

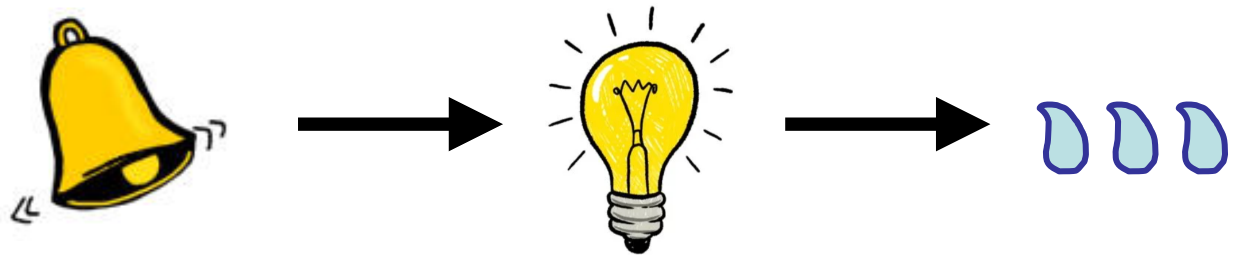
- More of this in a few weeks..

History: Learning *and* Control

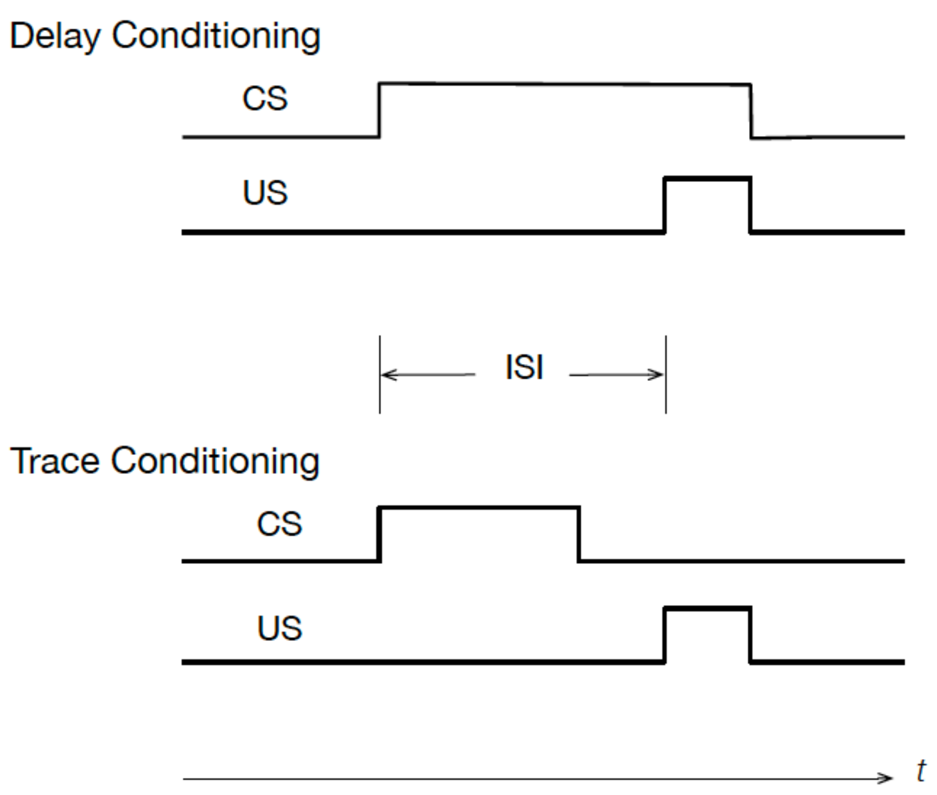
- **TD learning**, Actor-critic architecture (Sutton & Barto, 1981, 1982)
- Chris Watkins 1989 (+ Peter Dayan 1992): integrate dynamic programming with online learning

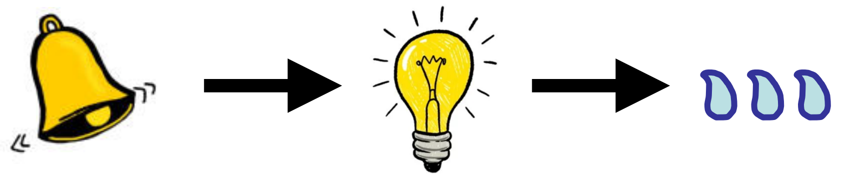


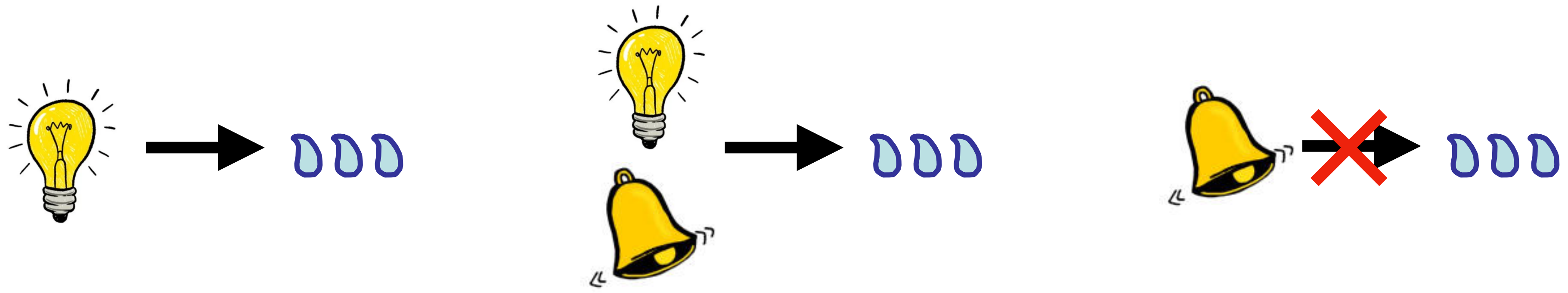
  - **Q learning**
- Key idea: use *experience* and *own value estimates*!
  - One example: secondary reinforcement



History: Psychology



- **Classical** (Pavlovian) **conditioning** (roughly) in domain of algorithms for **prediction**
  - Algorithms for **control**: **instrumental** (operant) **conditioning**
  - You probably know all this..
- At least two interesting phenomena in classical conditioning from algorithmic perspective:
  - **Higher-order conditioning**  Temporal Difference (TD) Learning
  - **Blocking**

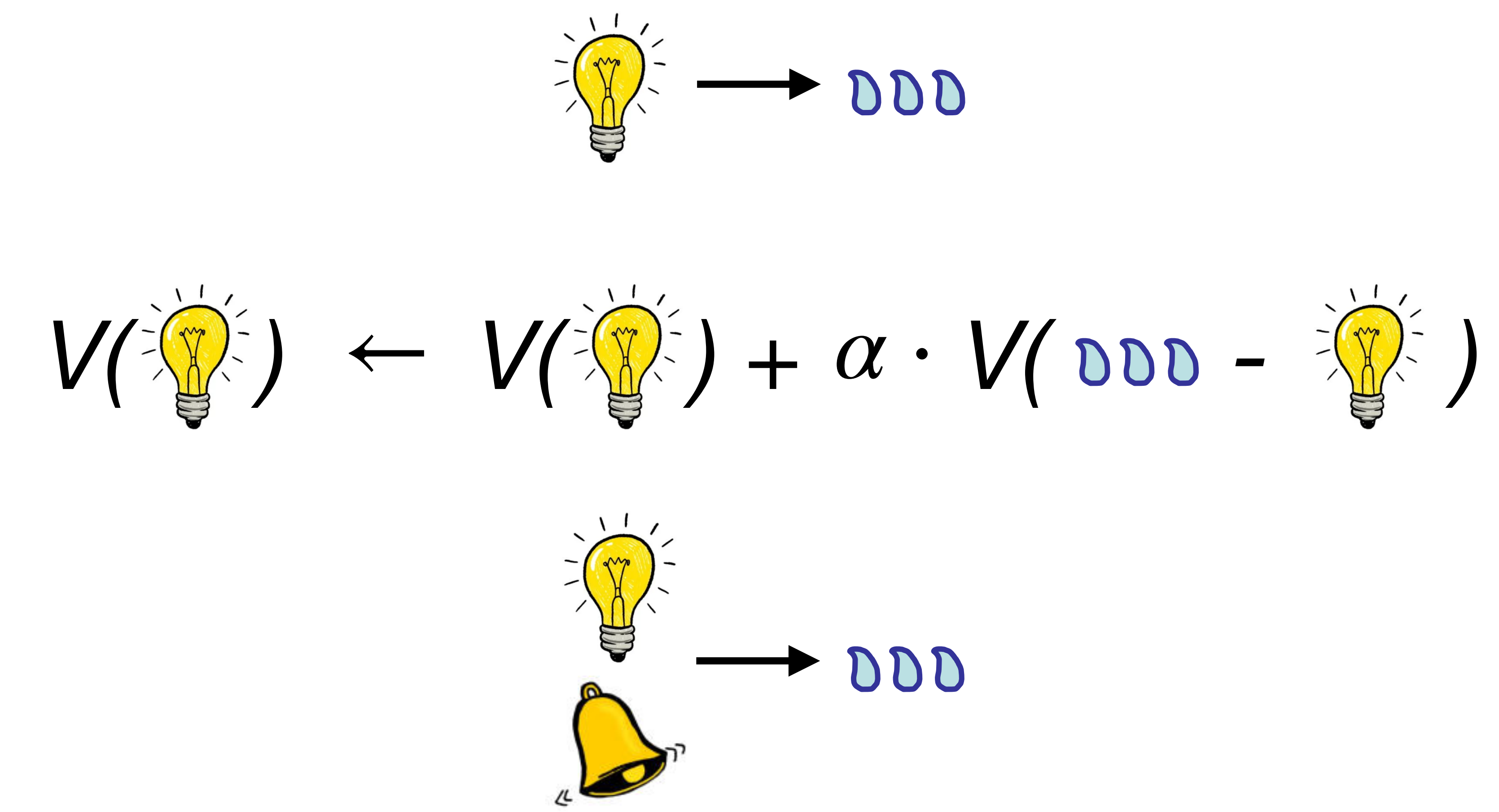


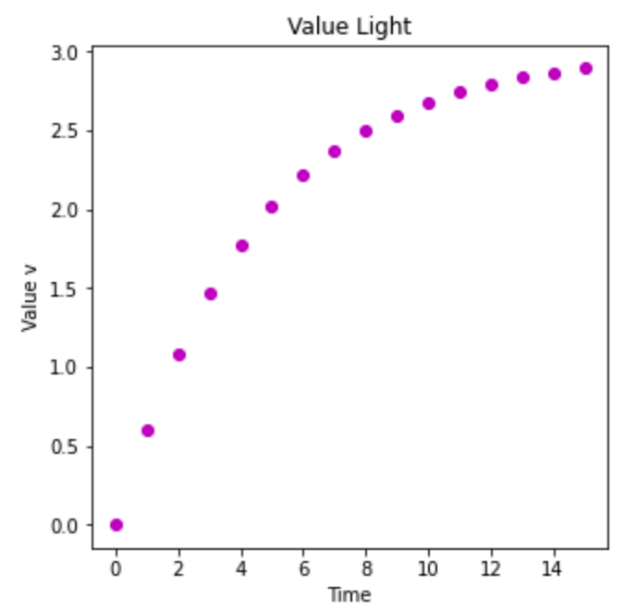
Rescorla-Wagner Learning

Basics of Learning: Blocking and Rescorla-Wagner Learning

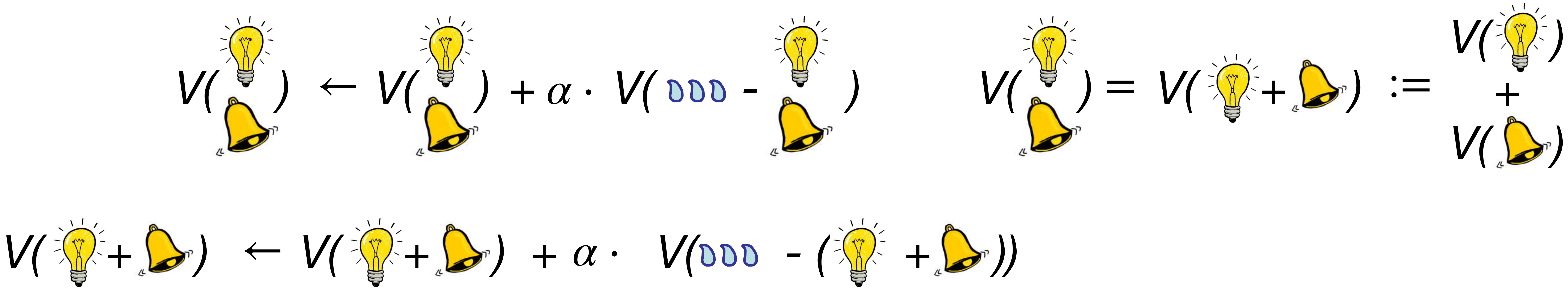
Learn associative strength between a CS and US

$$V(s) \leftarrow V(s) + \alpha \cdot (r - V(s))$$





Introduce a second CS:



What does the value of the sound CS look like at different stages of learning?

Coding: Python, Google Collab

https://github.com/schwartenbeckph/RL-Course/tree/main/2022_04_26

Basics of Learning: Higher order conditioning and TD learning (next time)

- Extends Rescorla–Wagner model
  - Address how within-trial *and* between-trial timing relationships among stimuli influence learning
  - How can higher-order conditioning arise?
- Real-time
  - t labels time steps within *or* between trials

$$V(s_t) \leftarrow V(s_t) + \alpha \cdot (r + \gamma \cdot V(s_{t+1}) - V(s_t))$$

  - Think of time between t and t+1 as a small time interval, say .01 second
- Solves:
  - Higher order conditioning
  - NO blocking if CS_2 is moved before previously learnt CS_1
  - A lot of other things..

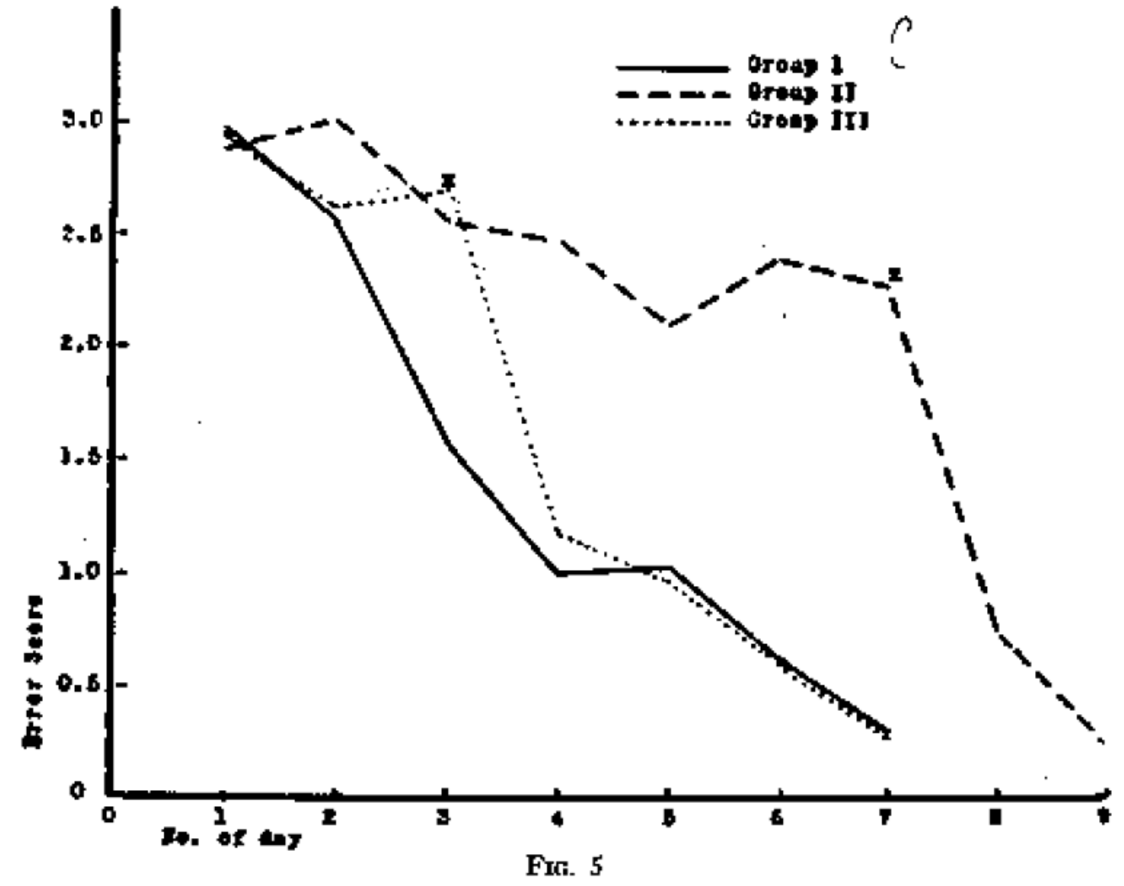
Basics of Learning: Instrumental conditioning

- Learning depends on the **consequences of behaviour**
  - Delivery of reinforcing stimulus contingent on what animal does
- Is both associative *and* selective
  - Unlike supervised learning (only associative)
  - Important aspect of **exploration**
- Problem: rewards often **sparse** - how to find the ‘correct’ actions?
  - Solution in lab experiments: **shaping** (*generalisability* of actions)
- Dealing with delayed reinforcement (cf. *trace conditioning*)
  - TD learning
  - Eligibility traces

Basics of Learning: Cognitive Maps

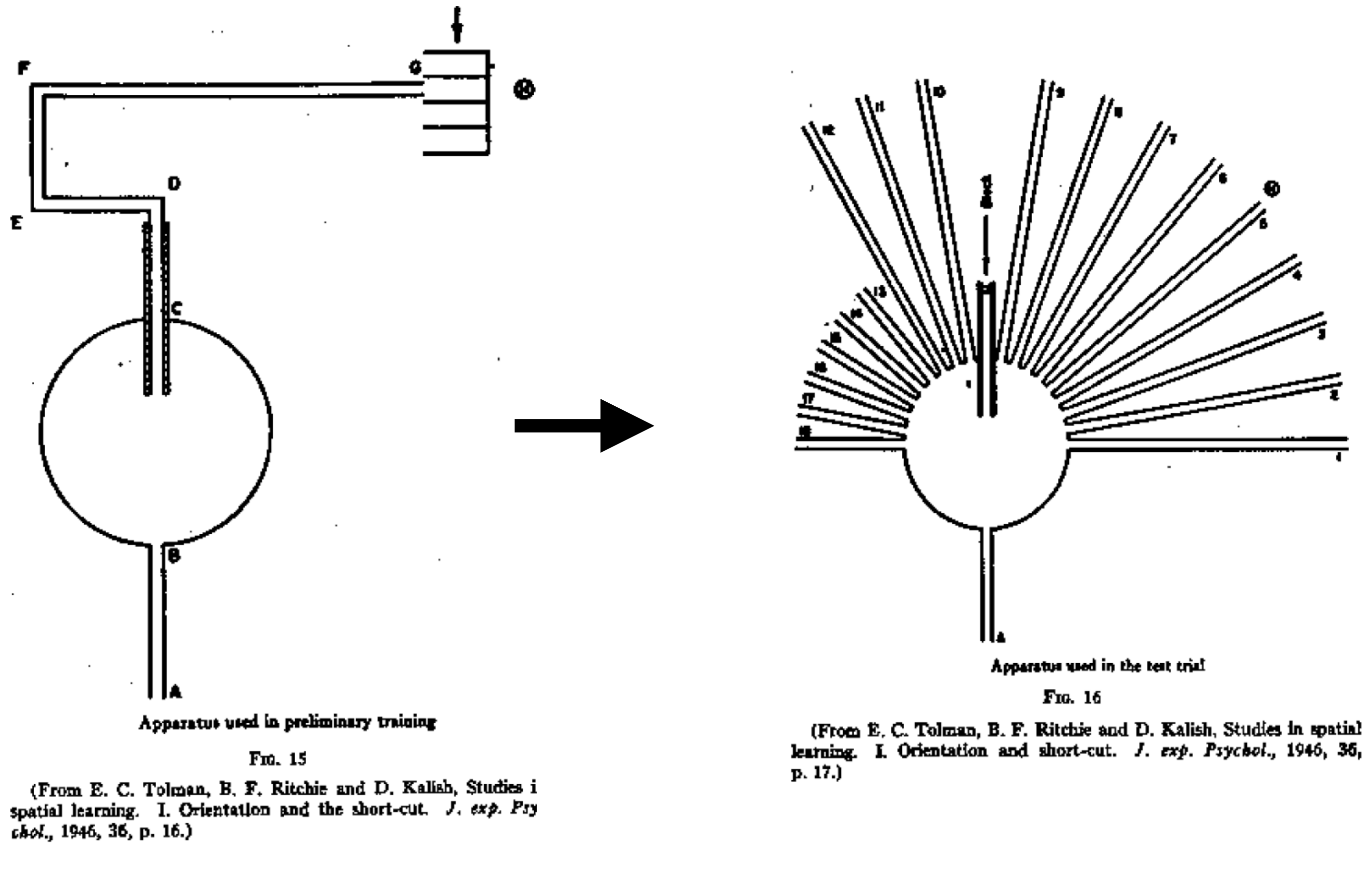
- Highlights the role of **environment model** - two parts:
  - **State-transition** part encodes effect of actions on state changes
  - **Reward model** part encodes reward signals expected for state/state-action pair

Latent learning



Tolman 1948: Cognitive Maps in Rats and Men

Structure learning



Tolman 1948: Cognitive Maps in Rats and Men

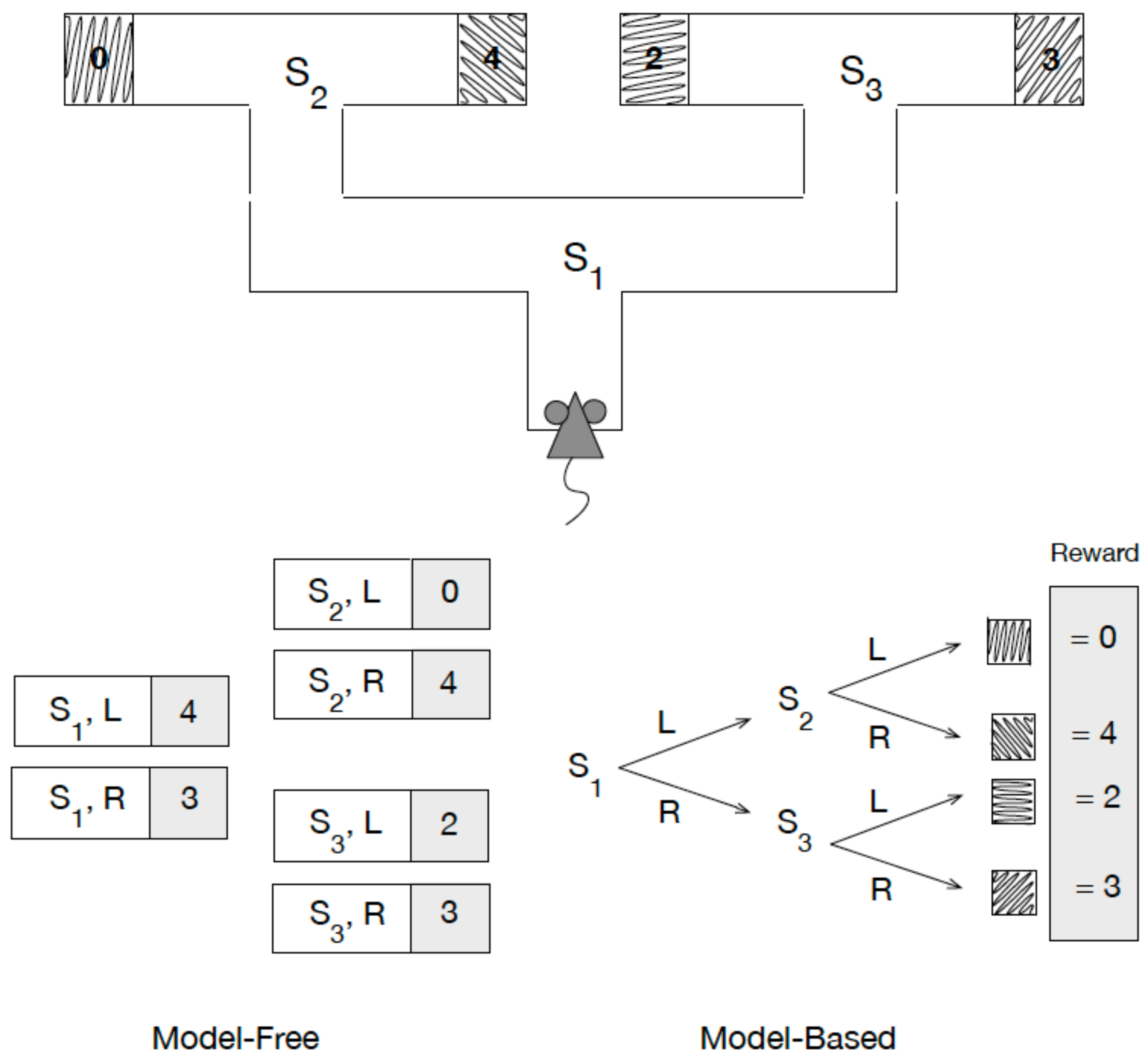
‘Multiple Systems’ in RL

- **Model-based RL**
  - Build a forward model of the task and outcomes
  - Search in the forward model
    - Optimal use of information
    - Computationally ruinous
- **Model-free RL**
  - learn values, which summarise future worth
    - computationally trivial
    - bootstrap-based; so statistically inefficient
- Learn both – select according to uncertainty

Slide from Peter Dayan

‘Multiple Systems’ in RL

Action values estimates of highest return rat can expect
- For each action
- Taken from each (nonterminal) state



Environment model consisting of
- State-transition model
- Reward model (decision tree)

Model-free agent:
Change policy or action value for a state =
move to state -> act from it (many times) -> experience consequences of actions

Model-based agent:
Change in reward model automatically leads to change in policy (through planning)

Key test to distinguish those two: **outcome-devaluation** (Also: two-step task - some other time)