General Principles of Human and Machine Learning

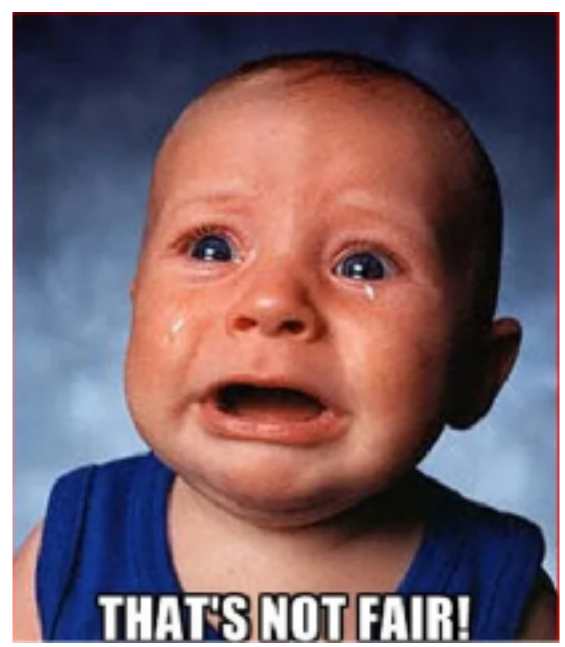
Lecture 3: Introduction to Reinforcement Learning

Dr Charline Tessereau

https://hmc-lab.com/GPHML.html

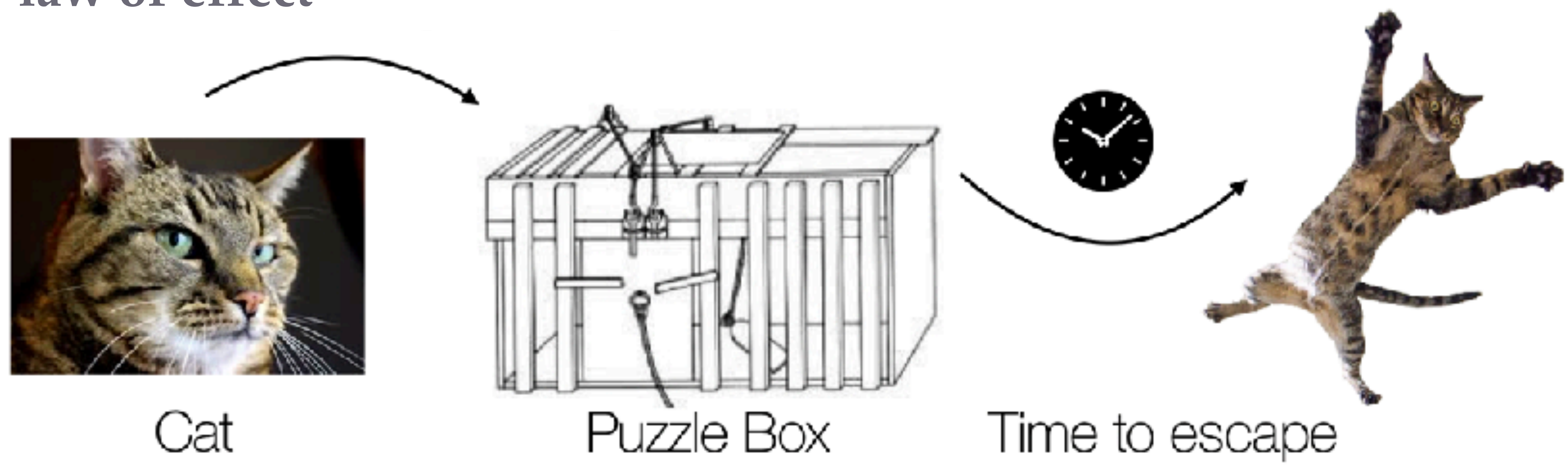
ANNOUNCEMENT

We will be taking the best 2 out of the 4 quizzes for fairness due to the bus strike last week.



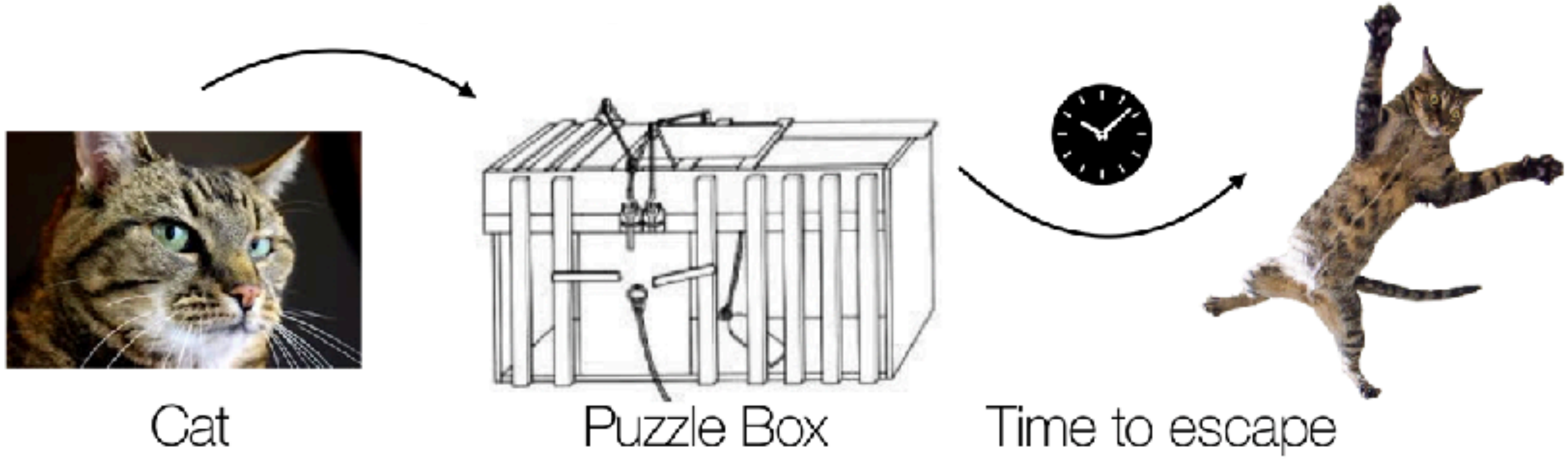
SUMMARY SO FAR

- **Thorndike's 'law of effect'**

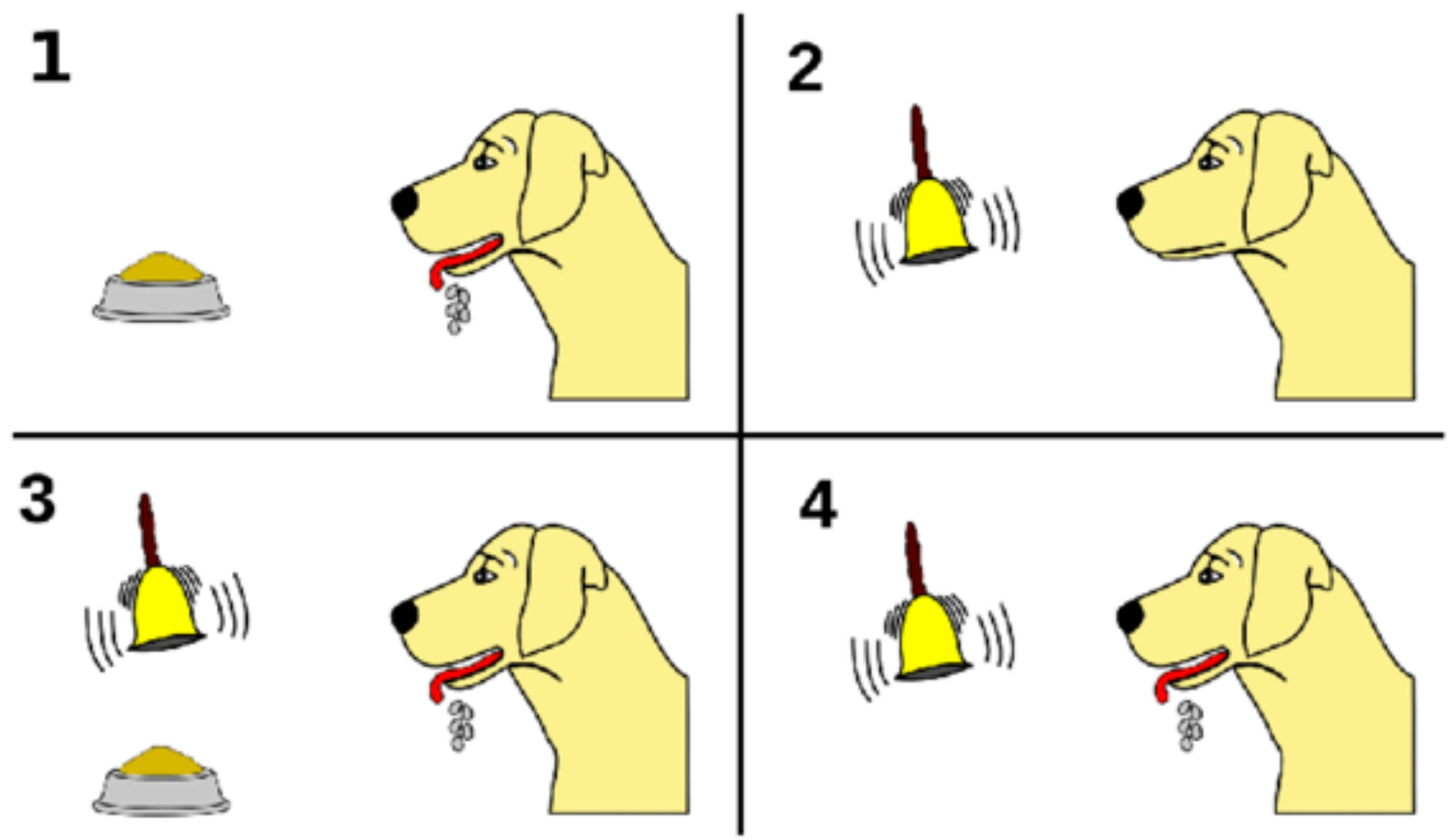


SUMMARY SO FAR

Thorndike's 'law of effect':

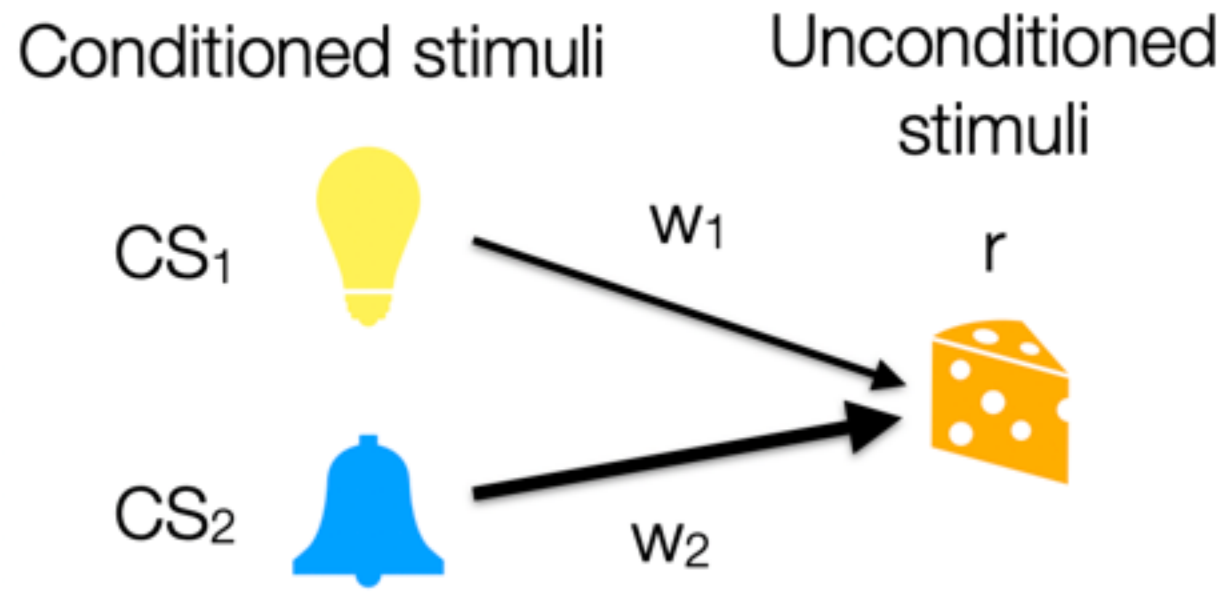


Pavlov:



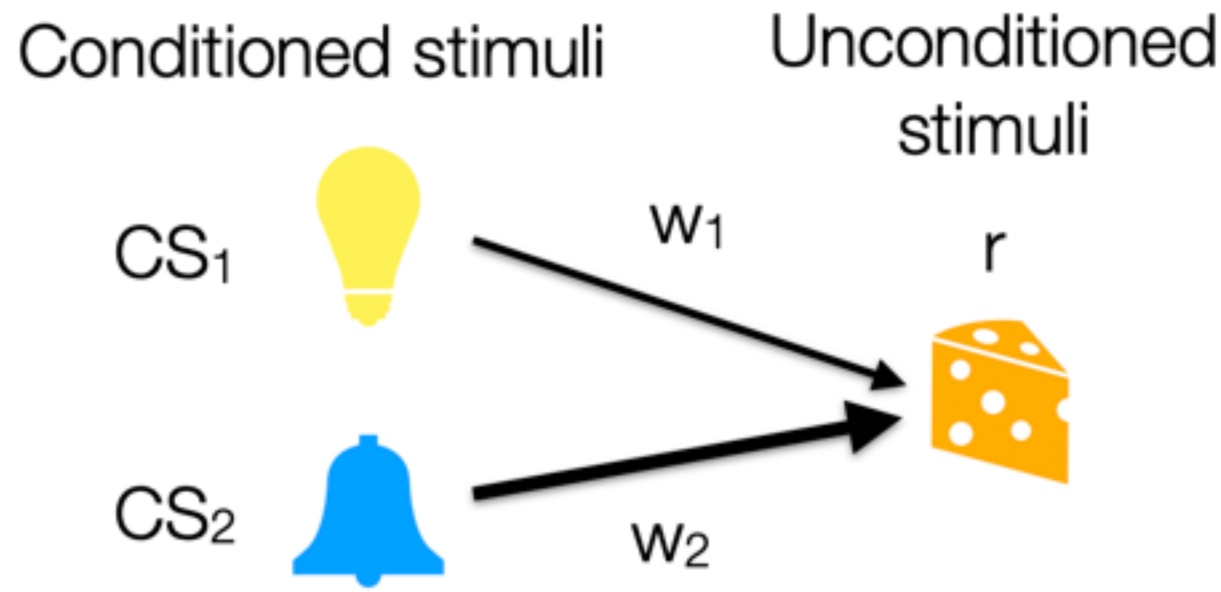
SUMMARY SO FAR

Rescorla-Wagner:



SUMMARY SO FAR

Rescorla-Wagner:

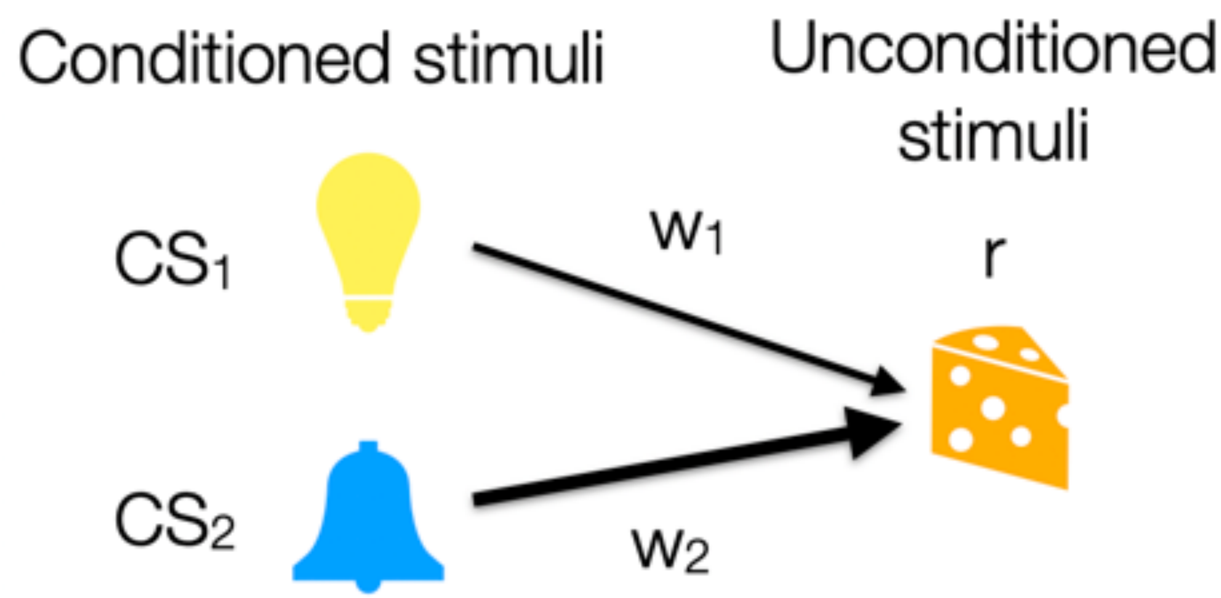


Rescorla-Wagner is primarily an association from stimulus to value:

- It formalizes the concept of 'classical conditioning'

SUMMARY SO FAR

Rescorla-Wagner:



Rescorla-Wagner is primarily an association from stimulus to value:

- It formalizes the concept of 'classical conditioning'

Watkins Q learning: extension to action selection

- Formalises the concept of 'operant conditioning'

FROM GOAL-DIRECTED LEARNING TO HABITS

Different forms of decisions in the ‘automatic’ spectrum:

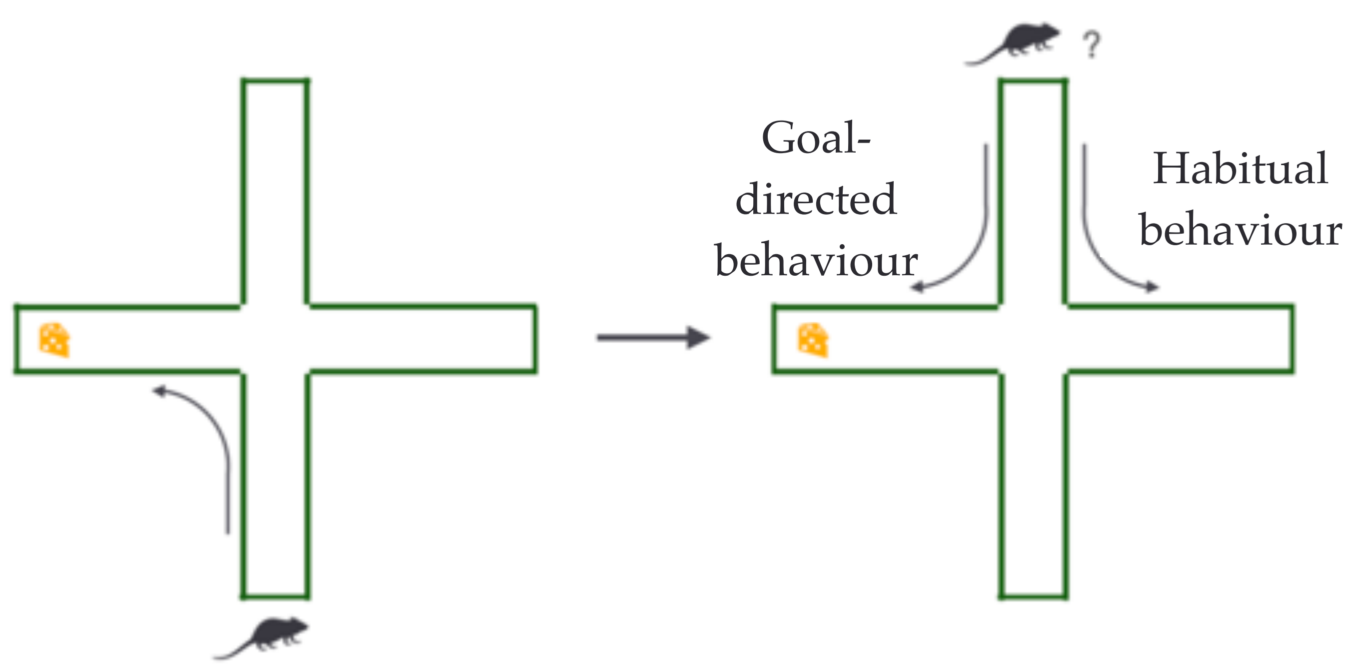
- Goal directed: one has an objective in mind
  - Involves planning
- Habitual:
  - Actions that are automatic

FROM GOAL-DIRECTED LEARNING TO HABITS

Learning = repeating past successful actions?

With repetitions:

- Goal-directed behaviour becomes automatic

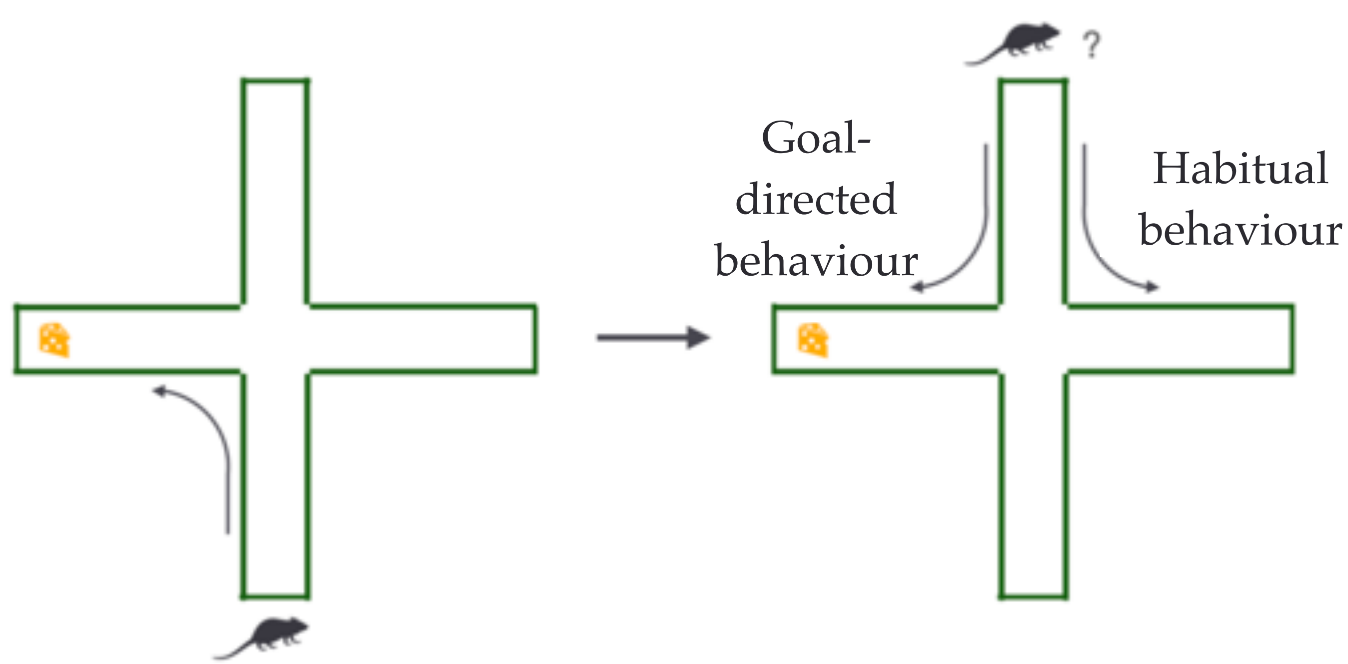


FROM GOAL-DIRECTED LEARNING TO HABITS

Learning = repeating past successful actions?

With repetitions:

- Goal-directed behaviour becomes automatic
- Experimentally, one can test this by looking at adaptation to changes



FROM GOAL-DIRECTED LEARNING TO HABITS

Learning = repeating past successful actions?

With repetitions:
- Goal-directed behaviour becomes automatic
- Experimentally, one can test this by looking at adaptation to changes

Efficient:
- Habits use less computational resources

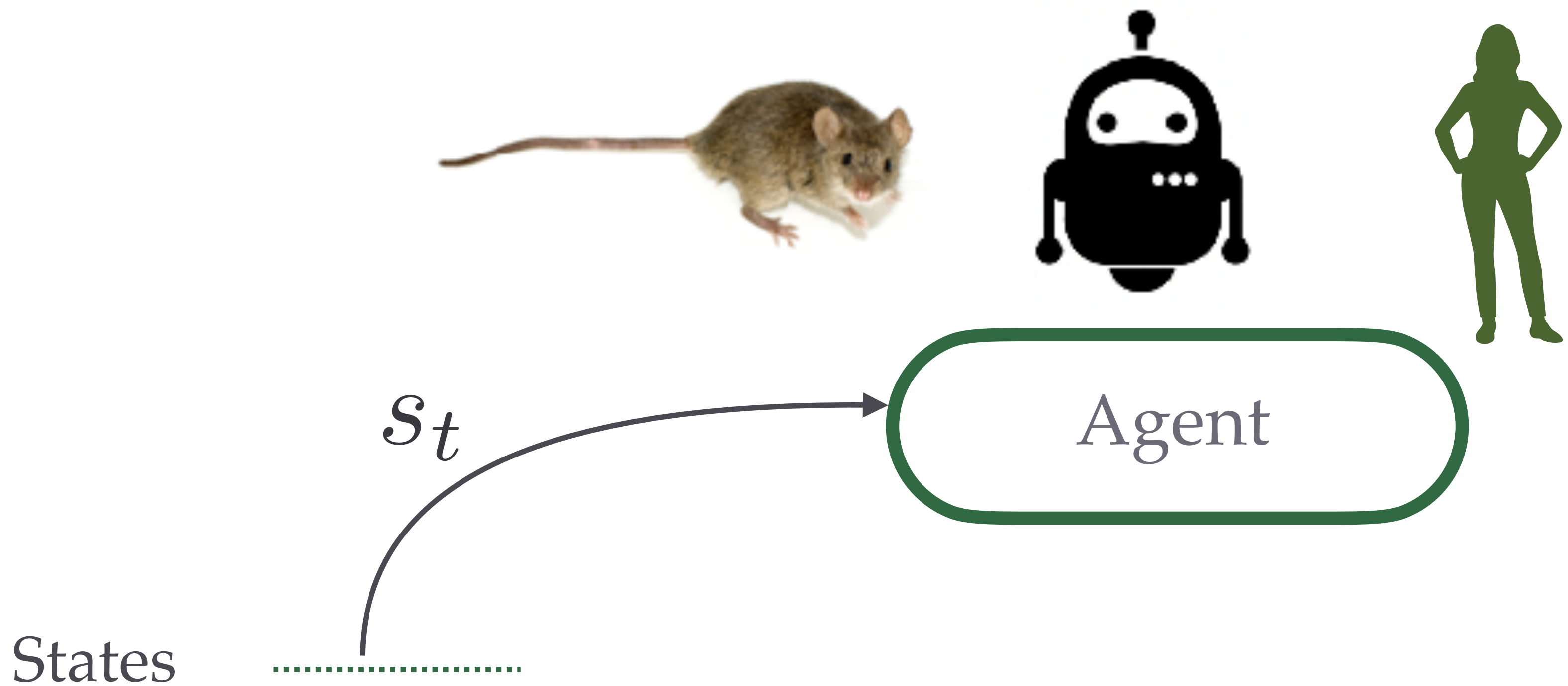
But… Not flexible:
- Habits can be blind of the outcome
- Example: addictions…

THEORY OF DECISION MAKING: REINFORCEMENT LEARNING

‘RL’ formalizes both operant and classical conditioning.

It formalizes decision making, both goal-directed and habitual.

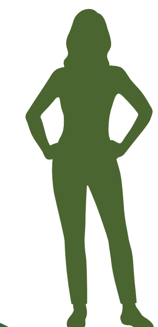
REINFORCEMENT LEARNING



Sutton and Barto. *Reinforcement learning: An introduction*. MIT press. (2018)

REINFORCEMENT LEARNING



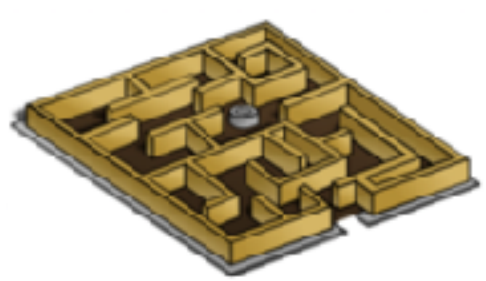
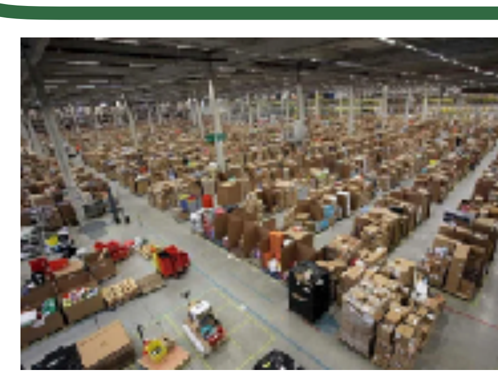


s_t

Agent

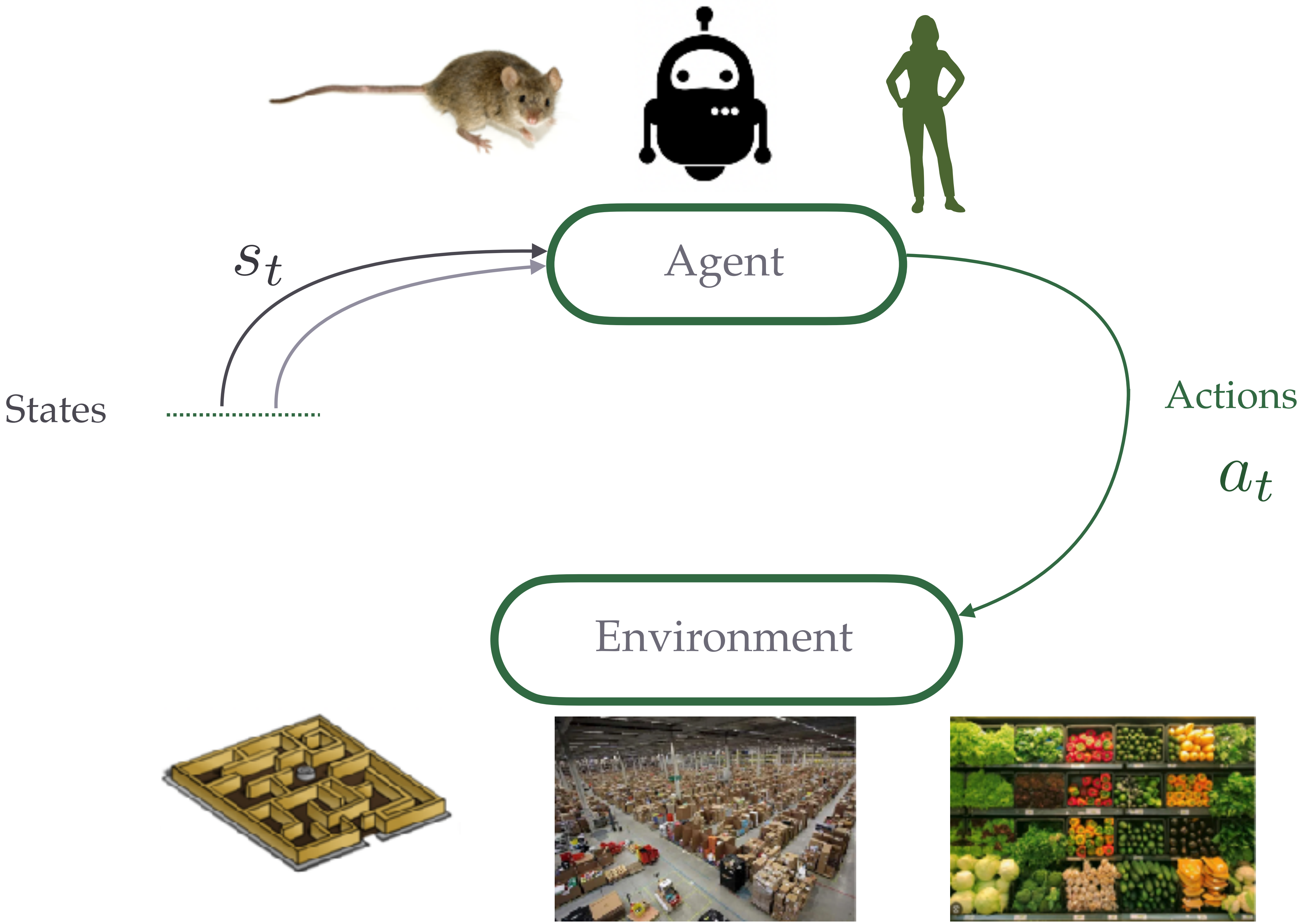
States

Environment

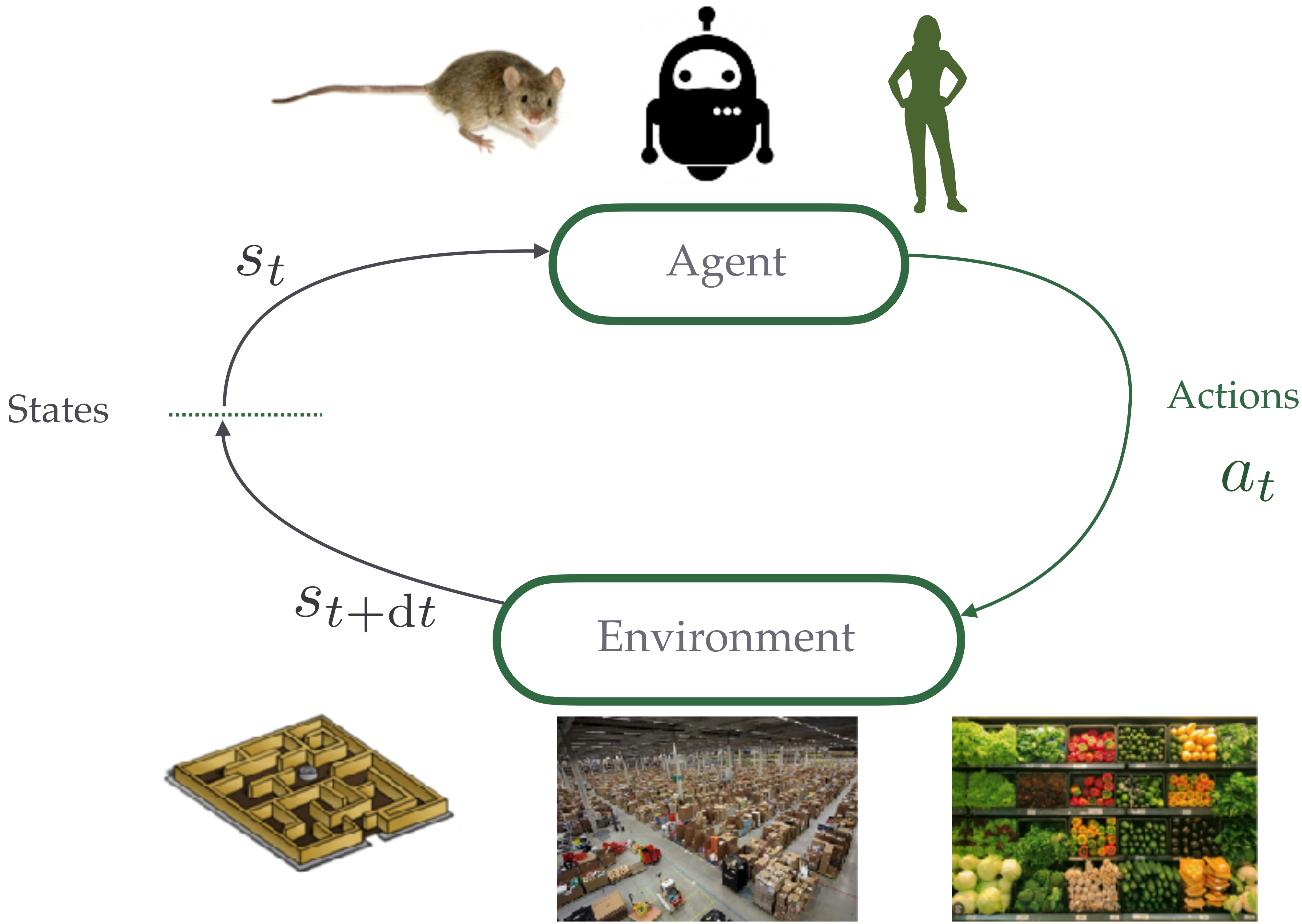

Sutton and Barto. *Reinforcement learning: An introduction*. MIT press. (2018)

REINFORCEMENT LEARNING



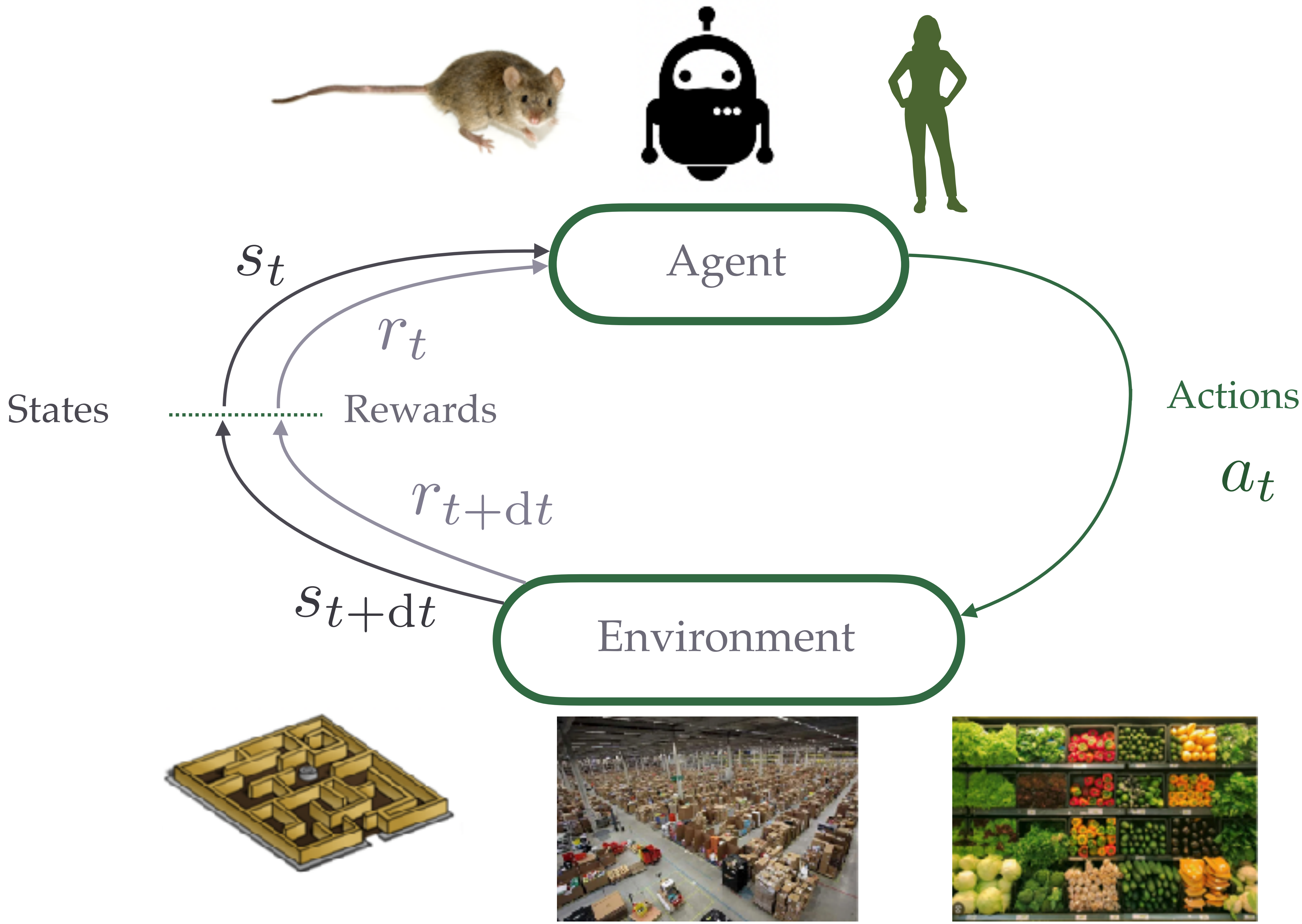
Sutton and Barto. *Reinforcement learning: An introduction*. MIT press. (2018)

REINFORCEMENT LEARNING



Sutton and Barto. *Reinforcement learning: An introduction*. MIT press. (2018)

REINFORCEMENT LEARNING

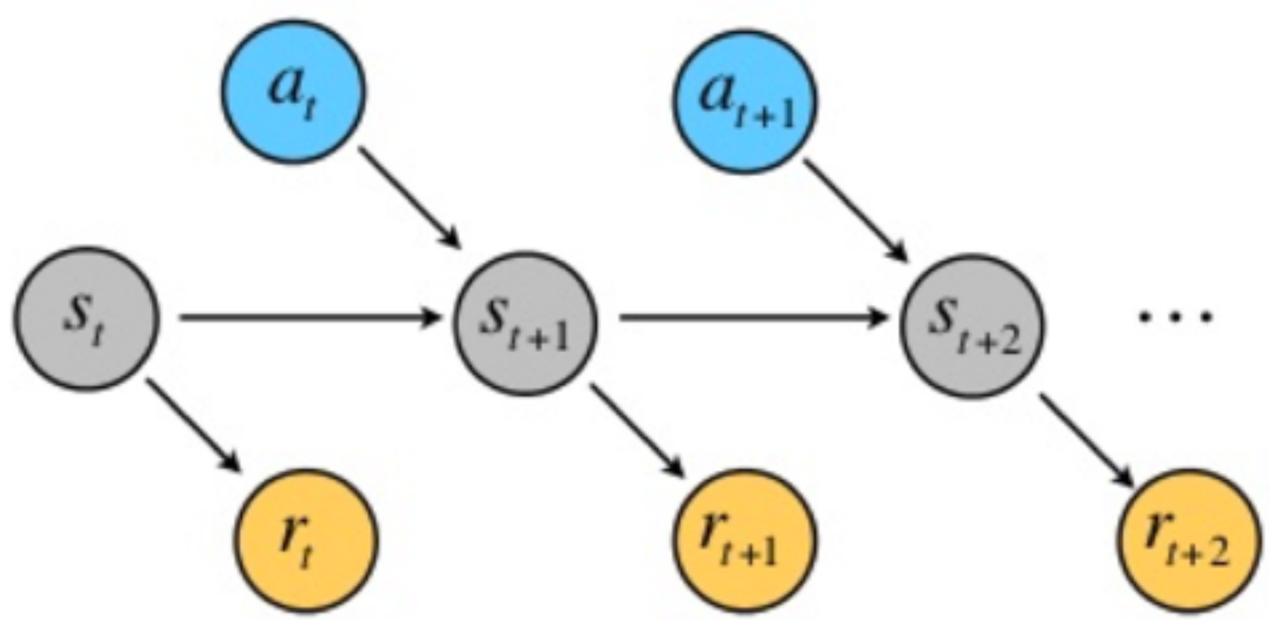


Sutton and Barto. *Reinforcement learning: An introduction*. MIT press. (2018)

REINFORCEMENT LEARNING

The environment can be described as a Markov Decision Process (MPD):

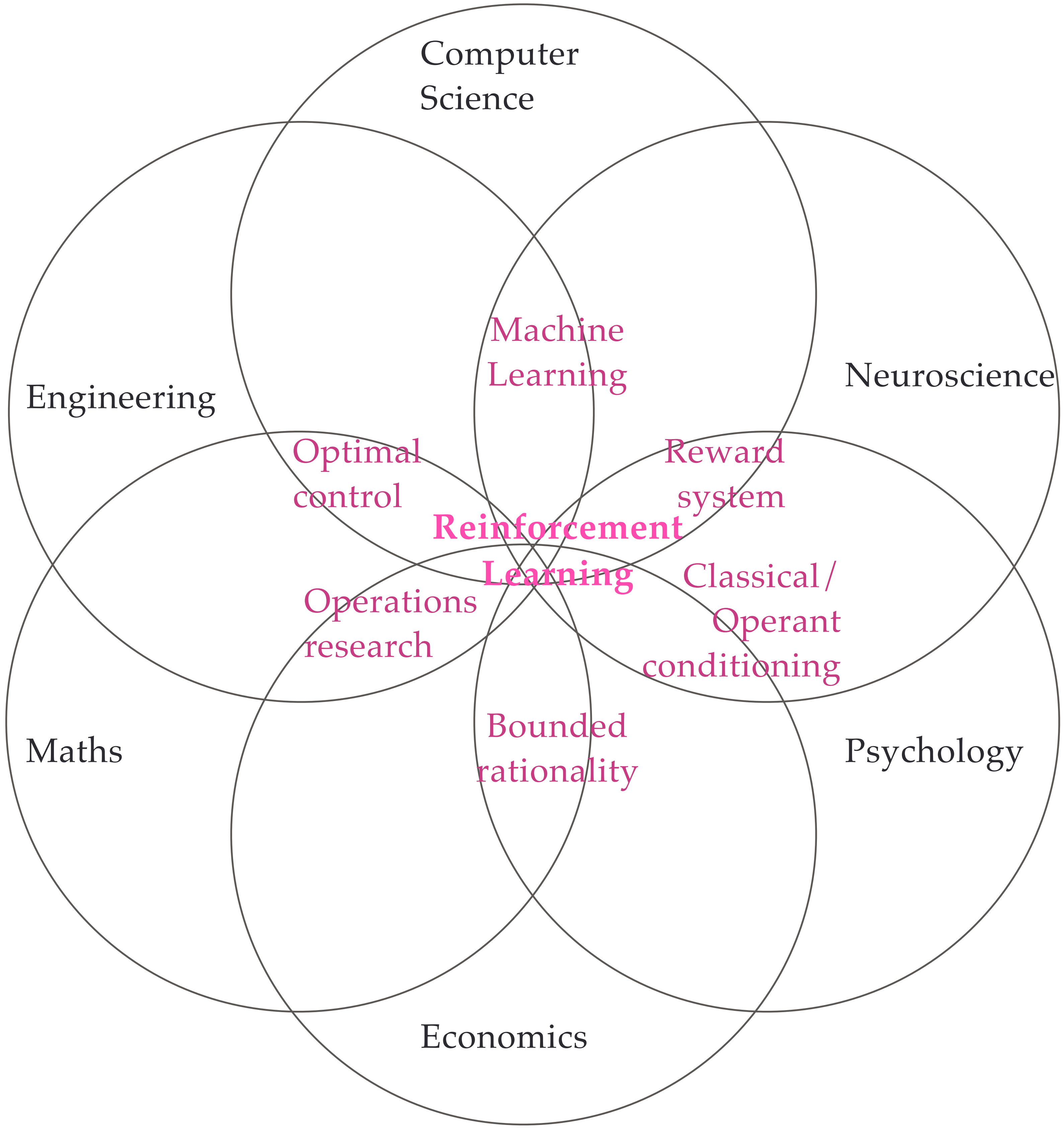
- The assumption is that the environment is fully described at the current state: Markov Properties $P(s_{t+1} \mid s_t, a_t)$
- In general, non-Markovian processes are much more complex



Sutton and Barto. *Reinforcement learning: An introduction*. MIT press. (2018)

REINFORCEMENT LEARNING

Science of learning to make decisions from interaction with the environment

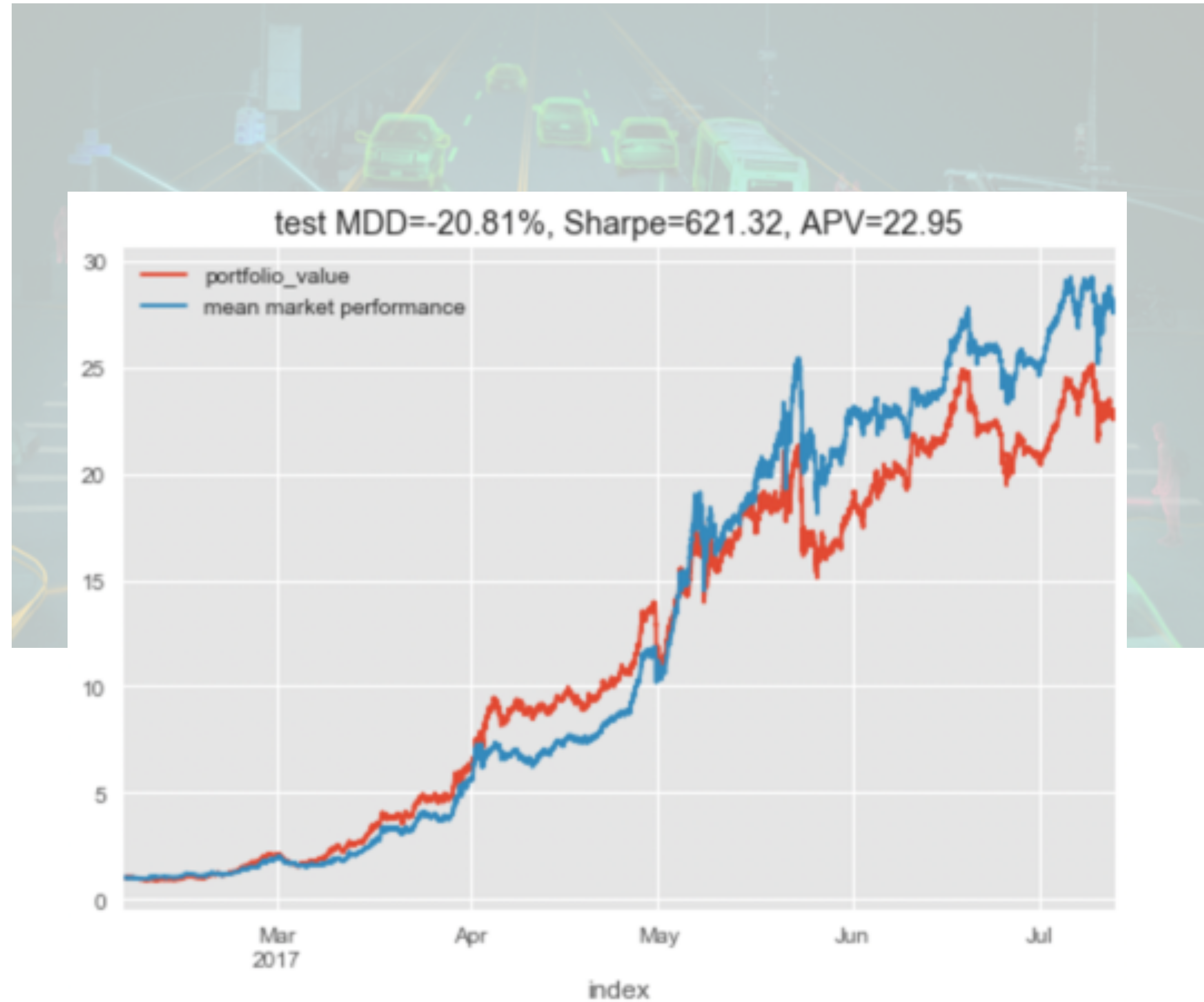


REINFORCEMENT LEARNING: EXAMPLES

- **Robots**
- Investment portfolio
- Play videos or board games
- Brain Machine Interface
- Animals

REINFORCEMENT LEARNING: EXAMPLES

- Robots
- **Investment portfolio**
- Play videos or board games
- Brain Machine Interface
- Animals



REINFORCEMENT LEARNING: EXAMPLES

- Robots
- Investment portfolio
- **Play videos or board games**
- Brain Machine Interface
- Animals

Nair, Preashant. *PNAS*.(2013)

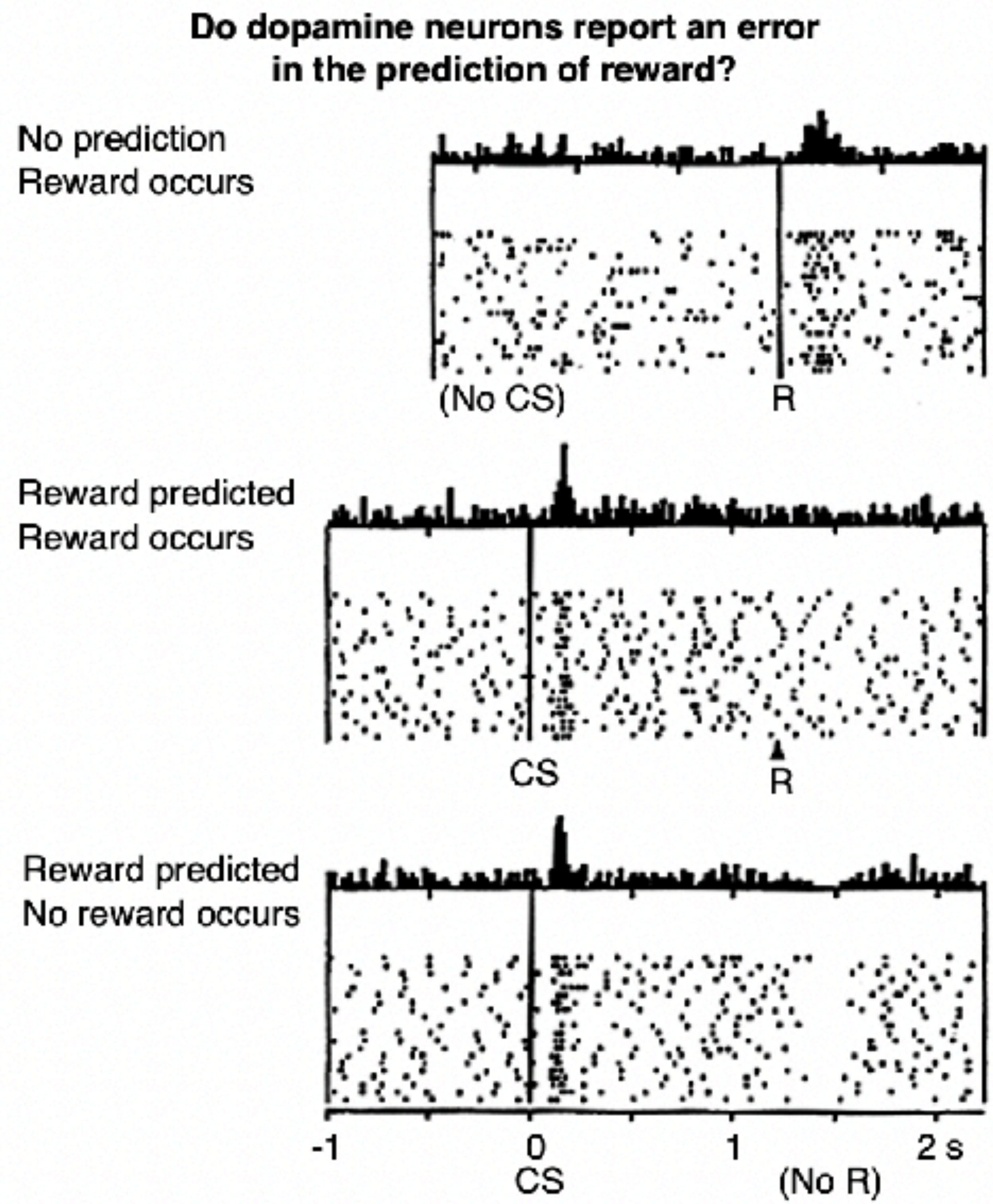
REINFORCEMENT LEARNING: EXAMPLES

- Robots
- Investment portfolio
- Play videos or board games
- **Brain Machine Interface**
- Animals

Nair, Preashant. *PNAS*.(2013)

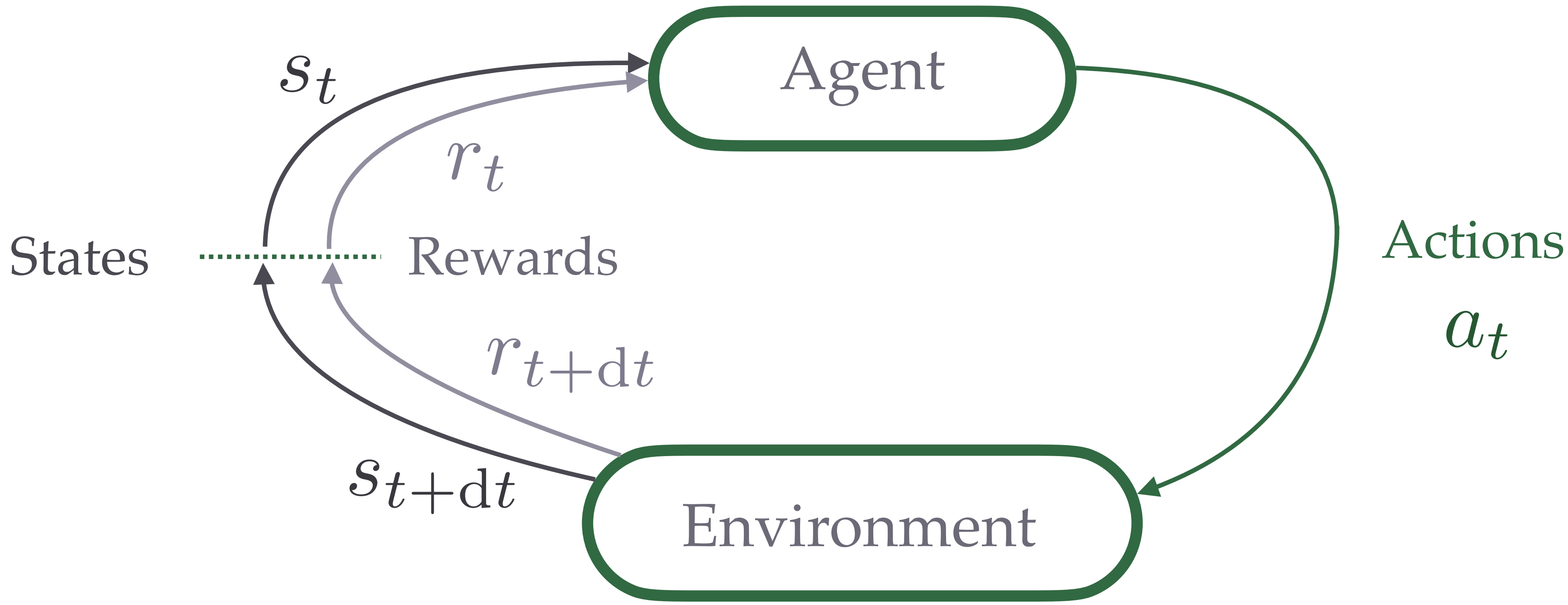
REINFORCEMENT LEARNING: EXAMPLES

- Robots
- Investment portfolio
- Play videos or board games
- Brain Machine Interface
- **Animals**



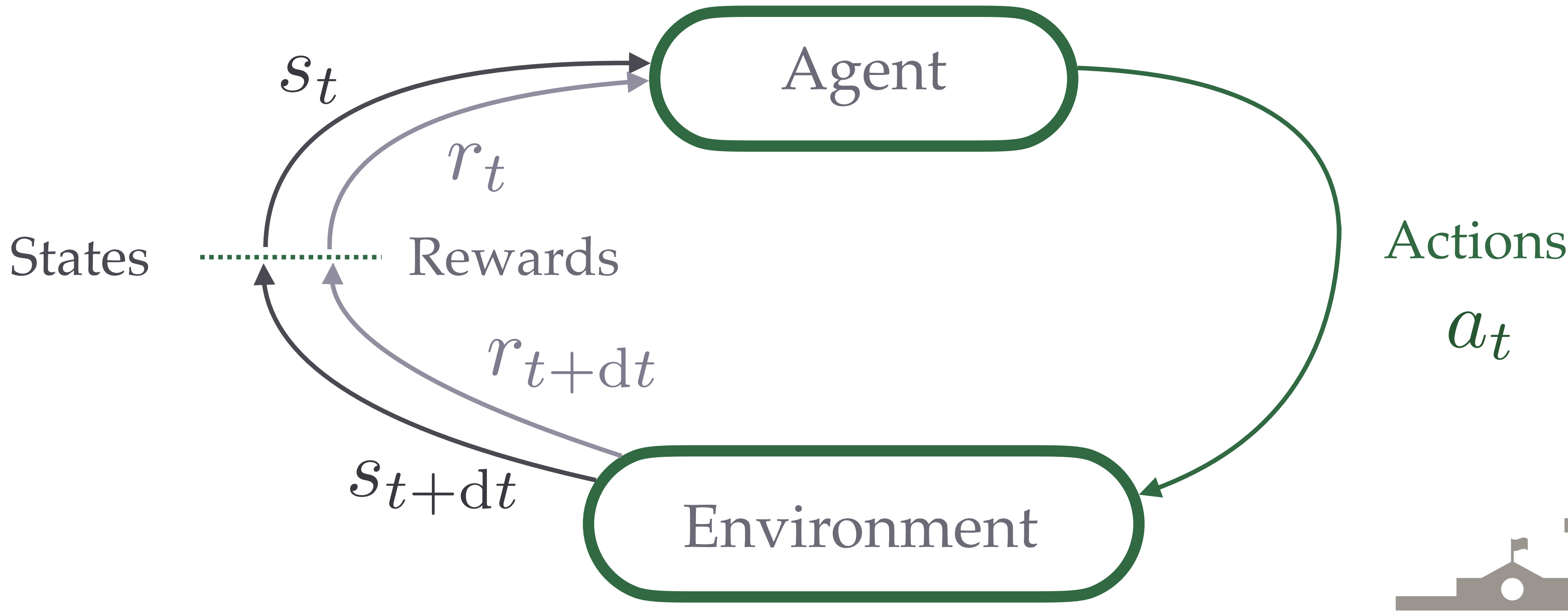
Schulz, Dayan, Montague. *Science*.(1997)

Nair, Preashant. *PNAS*.(2013)



Policy: $\pi_t(a|s) = P(a_t = a|s_t = s)$

?



Policy: $\pi_t(a|s) = P(a_t = a|s_t = s)$

Value: $V^{\pi}(s_t) = E_{\pi}(R_t|\, s_t = s)$

Value function: expected cumulated future discounted rewards

Value Function:

$$V^{\pi}(s_t) = E_{\pi}(R_t \mid s_t = s)$$

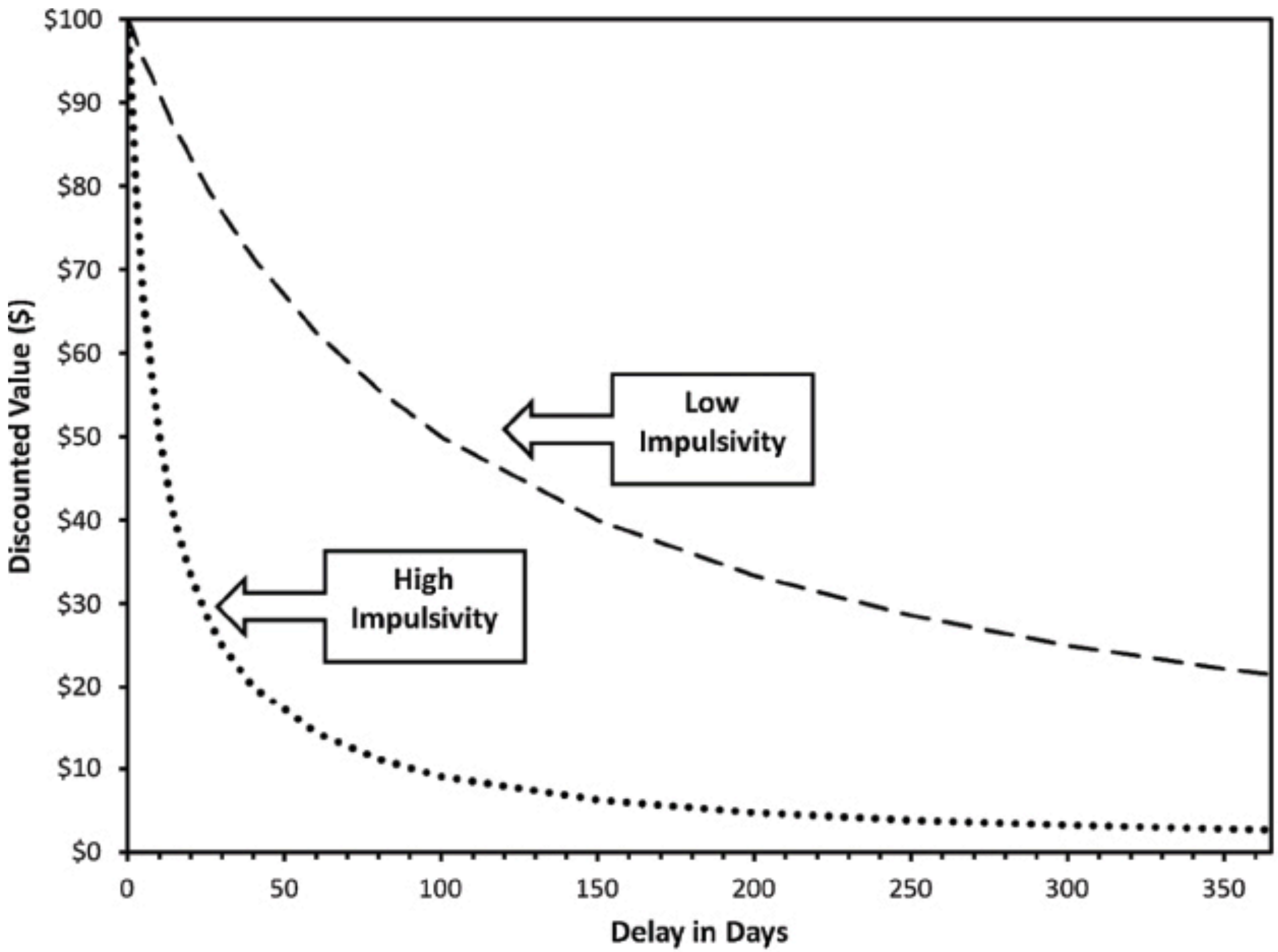
$$= E_{\pi}\left(\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s\right)$$

Sum of all future rewards

discounted

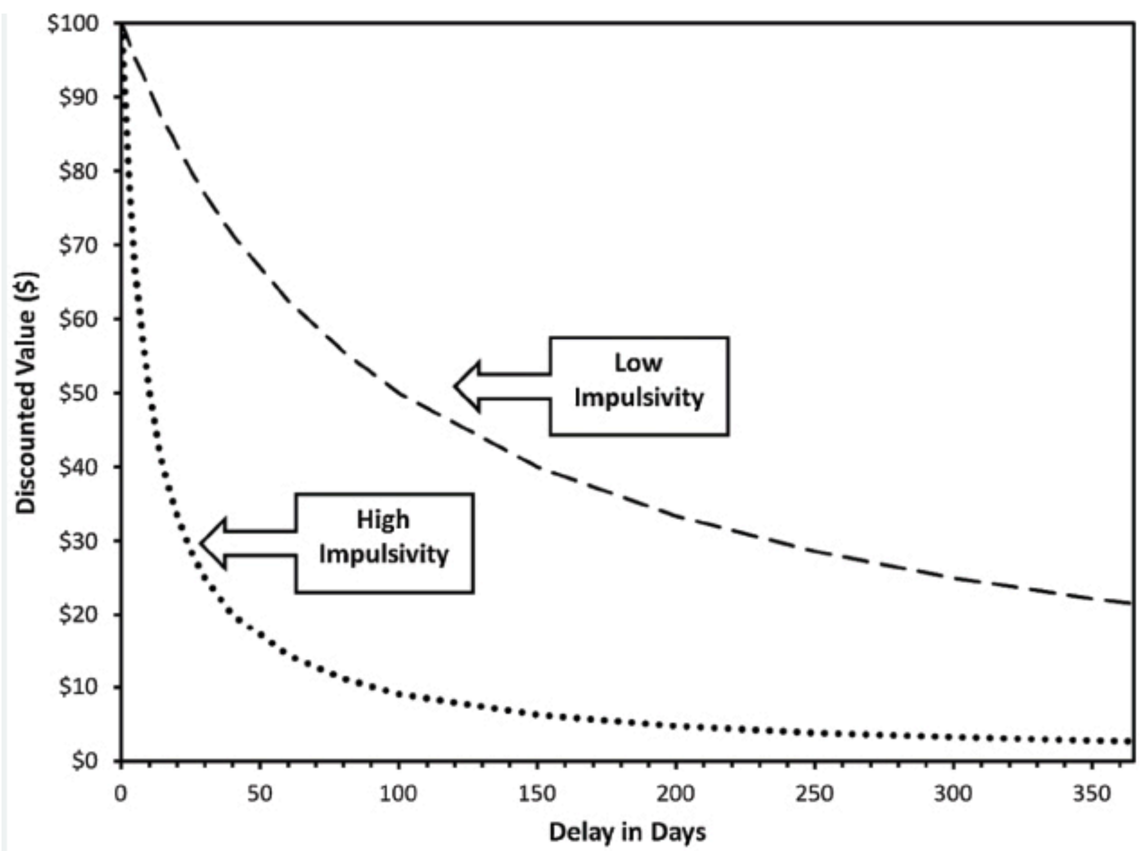
Discount factor:

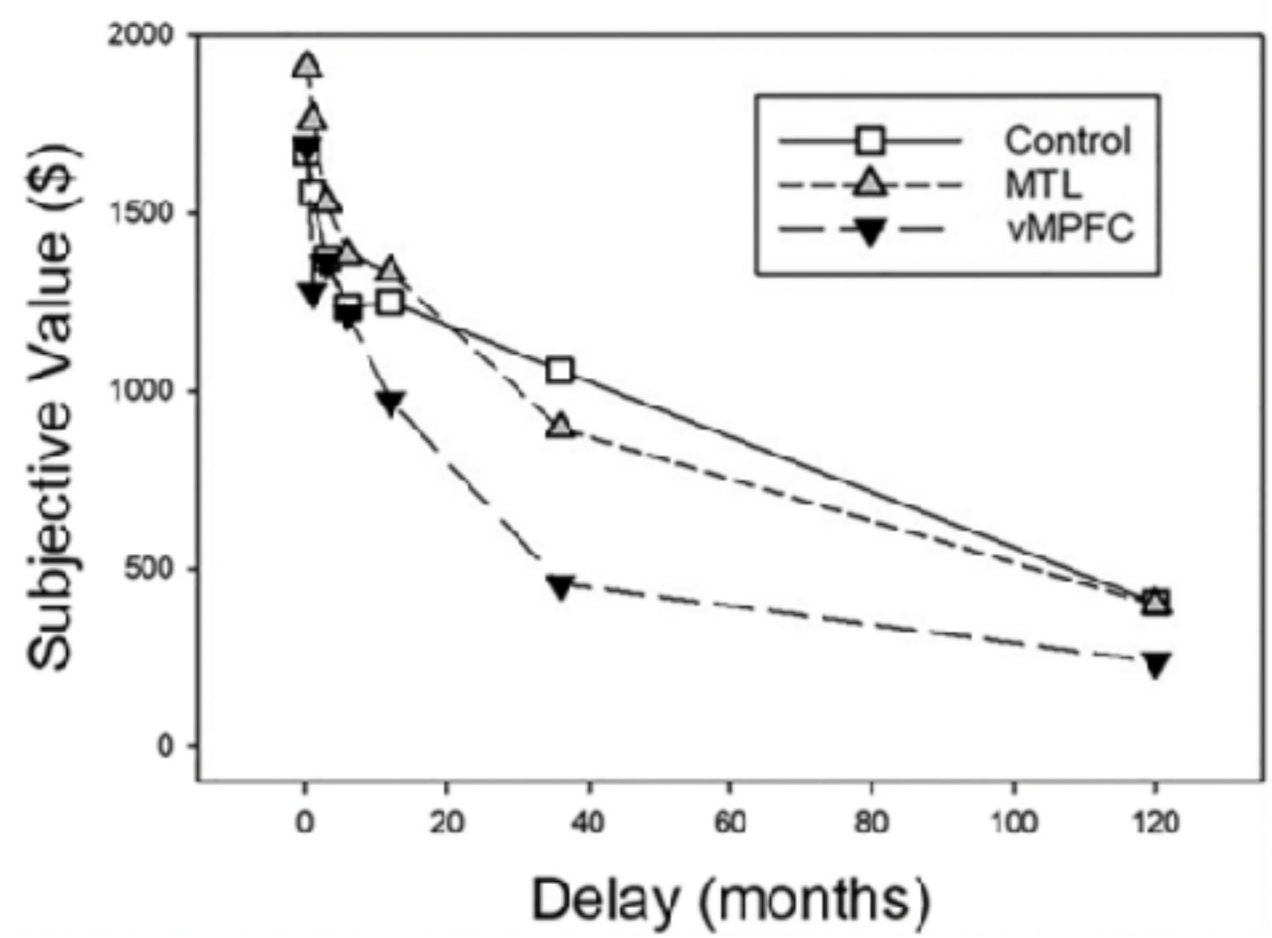
- emerges from uncertainty
- can characterize degrees of impulsivity



Discount factor:

- emerges from uncertainty
- can characterize degrees of impulsivity





Mok, Jenkin NY, et al. *Journal of Cognitive Neuroscience*. (2021)

Value function: expected cumulated future discounted rewards

Value Function:

$$V^{\pi}(s_t) = E_{\pi}(R_t | s_t = s)$$

$$= E_{\pi}\left(\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s\right)$$

$$\left[R_{ss'}^{a} + \gamma E_{\pi}\left(\sum_{k=0}^{\infty} \gamma^k r_{t+k+2} | s_{t+1} = s'\right)\right]$$

Current reward + sum of discounted rewards to be expected in the future states

Value function: expected cumulated future discounted rewards

Value Function:

$$V^{\pi}(s_t) = E_{\pi}(R_t |\, s_t = s)$$

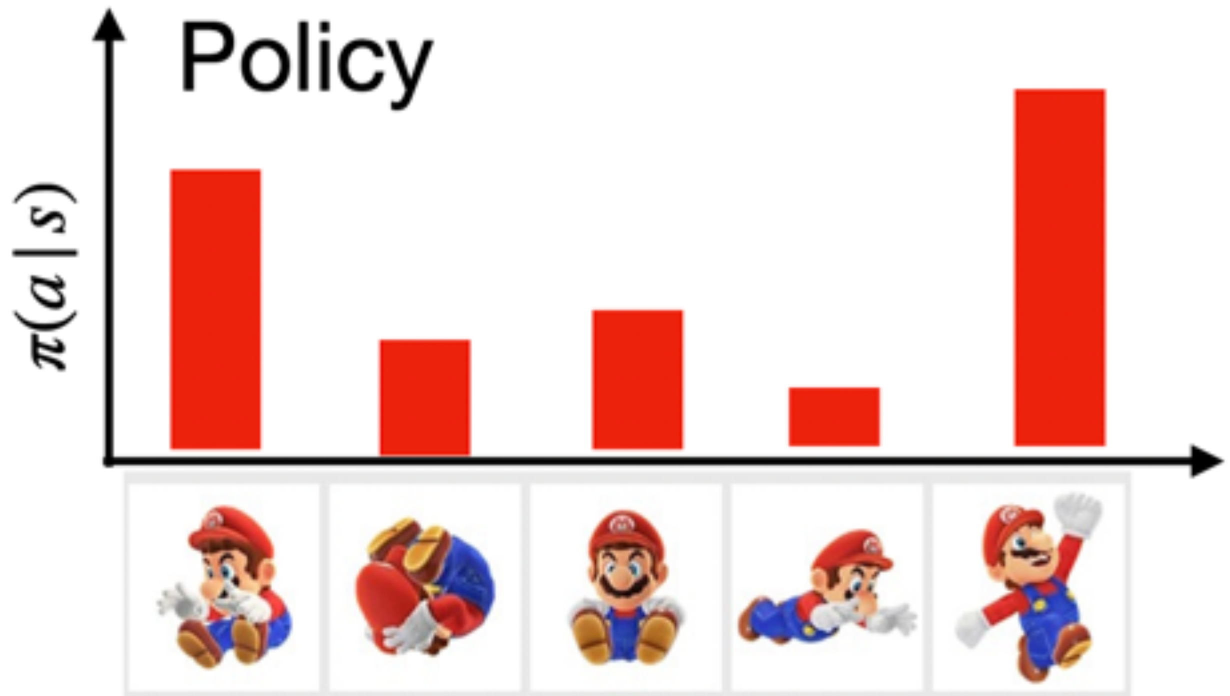
$$= E_{\pi}\left(\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s\right)$$

Sum over all possible actions weighted by their probabilities π Policy

$$= \sum_a \pi(s,a) \left[R^a_{ss'} + \gamma E_{\pi}\left(\sum_{k=0}^{\infty} \gamma^k r_{t+k+2} | s_{t+1} = s'\right)\right]$$

Current reward + sum of discounted rewards to be expected in the future states

Policy: probability function over the action space, conditioned on the state



Has to be learned…

Value function: expected cumulated future discounted rewards

Value Function:

$$V^{\pi}(s_t) = E_{\pi}(R_t \mid s_t = s)$$

$$= E_{\pi}\left(\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s\right)$$

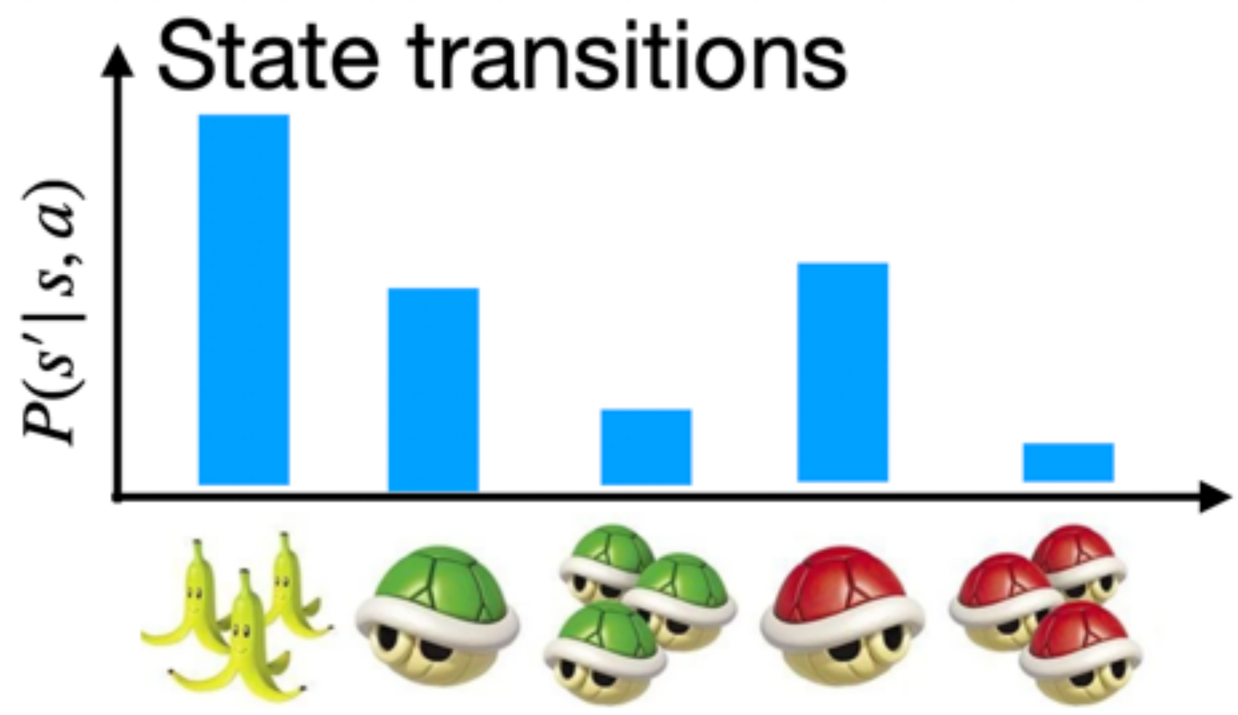
$$= \sum_{a} \pi(s,a) \sum_{s'} P^{a}_{ss'}\left[R^{a}_{ss'} + \gamma E_{\pi}\left(\sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \mid s_{t+1} = s'\right)\right]$$

Sum over all possible actions weighted by their probabilities π Policy

Current reward + sum of discounted rewards to be expected in the future states

Transition probabilities from the environment

Transition: probability to reach a state s′ from state s



- Intrinsic of the environment
- Can change with time (routes close, new routes form…)

Value function: expected cumulated future discounted rewards

Current reward + sum of discounted rewards to be expected in the future states

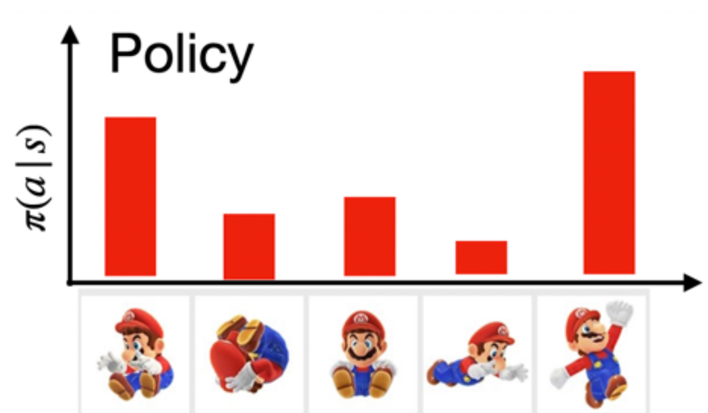
Value Function: $V^{\pi}(s_t) = E_{\pi}(R_t |\, s_t = s)$

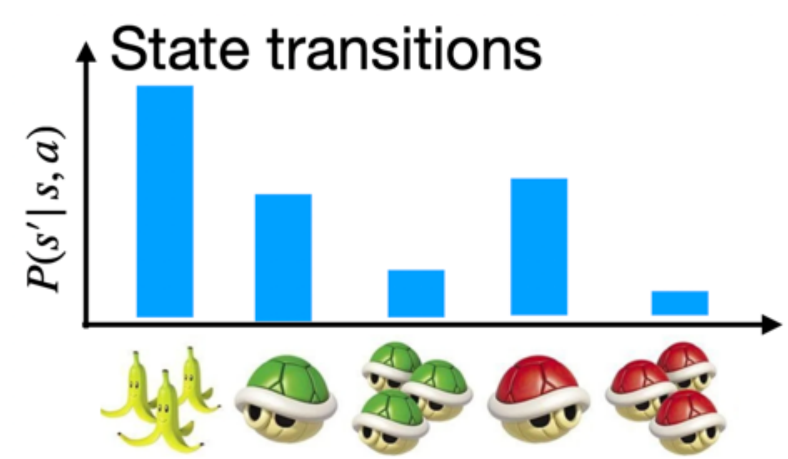
$$= \sum_a \pi(s,a) \sum_{s'} P^a_{ss'} \left[R^a_{ss'} + \gamma E_{\pi} \left(\sum_{k=0}^{\infty} \gamma^k r_{t+k+2} | s_{t+1} = s' \right) \right]$$

Sum over all possible actions weighted by their probabilities

Transition probabilities from the environment

Policy π





Model-based:

The transitions between states and the reward vector over the states are known.

Model-based:

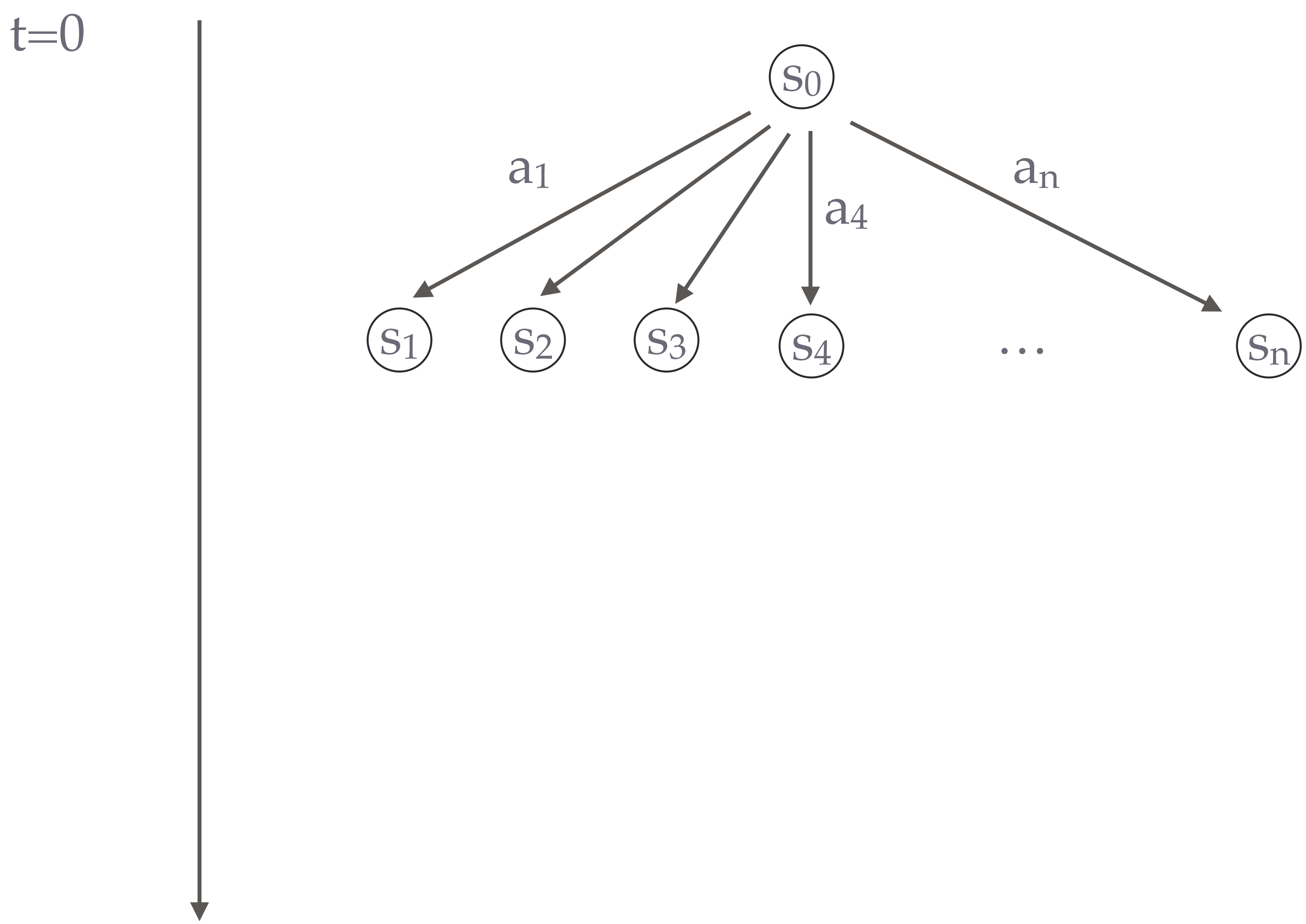
The transitions between states and the reward vector over the states are known.

Learning the optimal policy using simulated experience
=graph search

Model-based:

The transitions between states and the reward vector over the states are known.

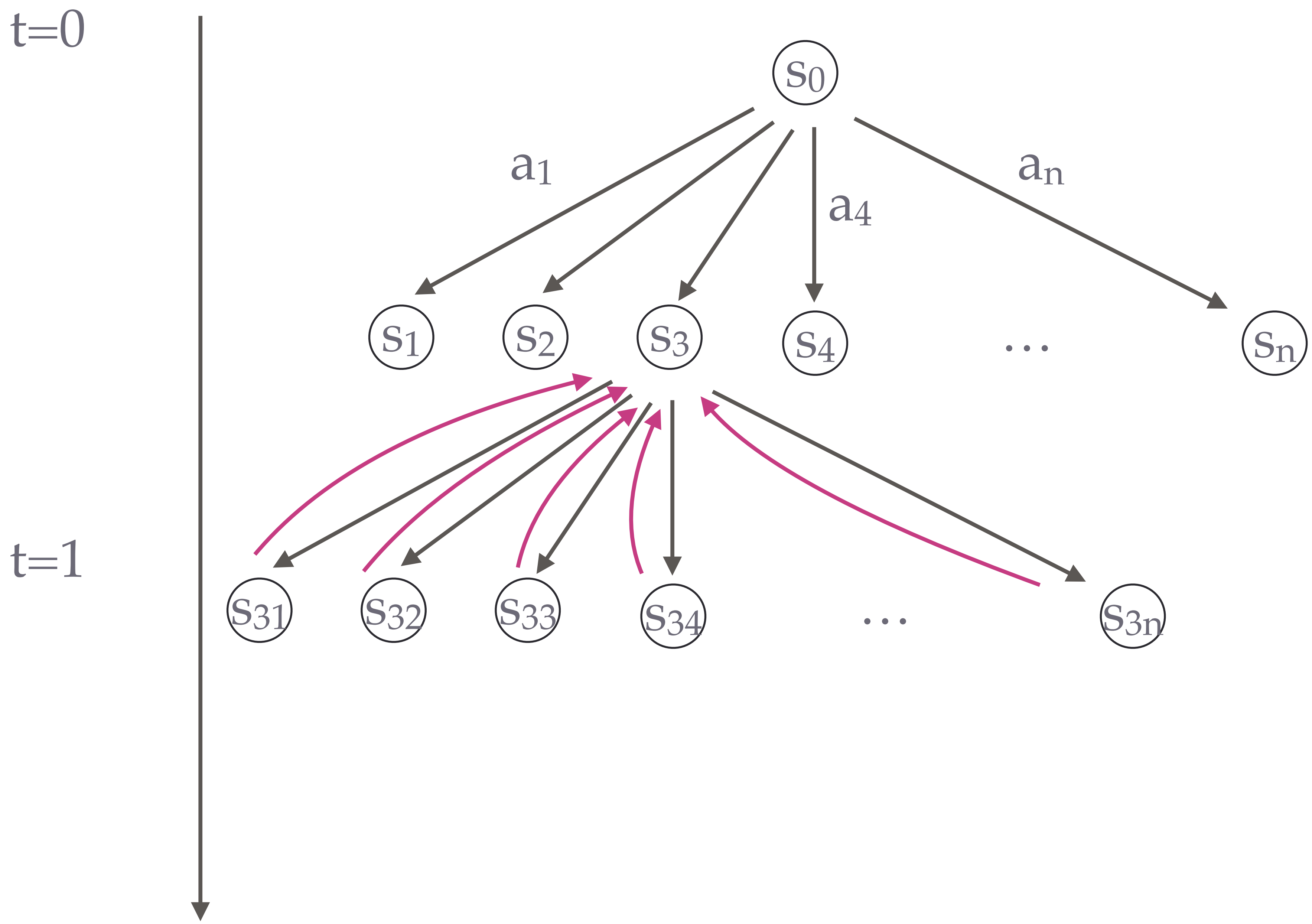
Learning the optimal policy using simulated experience
=graph search



Model-based:

The transitions between states and the reward vector over the states are known.

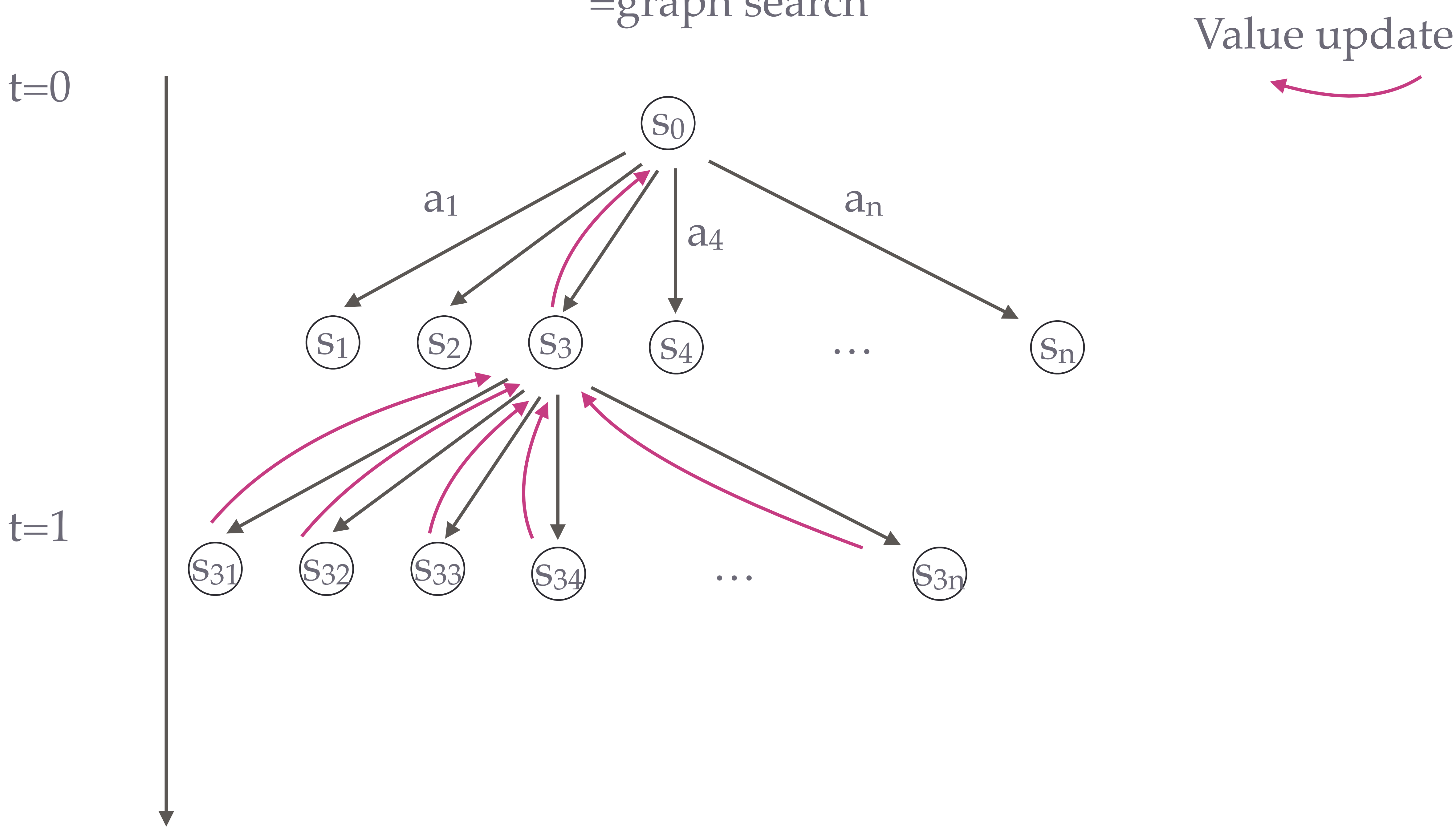
Learning the optimal policy using simulated experience
=graph search



Model-based:

The transitions between states and the reward vector over the states are known.

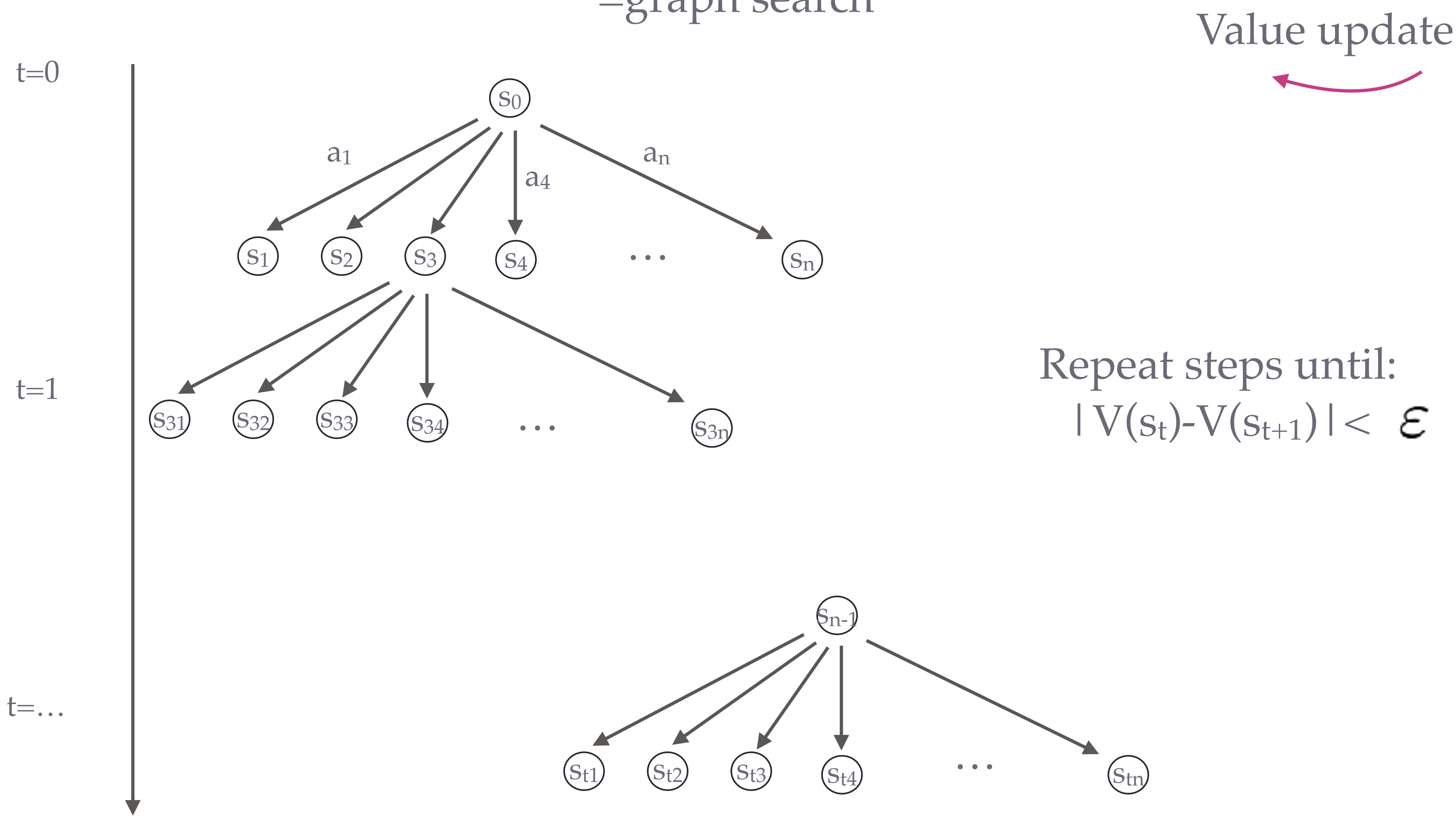
Learning the optimal policy using simulated experience
=graph search



Model-based:

The transitions between states and the reward vector over the states are known.

Learning the optimal policy using simulated experience
=graph search



Repeat steps until:
$|V(s_t)-V(s_{t+1})| < \varepsilon$

Model-based:

Benefits:

- Sample efficiency:
  - No need to experience before knowing what to do
- Exploration:
  - The agent has information over the whole space
- Flexibility:
  - As the transitions and reward vector are given, the agent can adapt to any change by planning ahead

Model-based:

Benefits:

- Sample efficiency:
  - No need to experience before knowing what to do
- Exploration:
  - The agent has information over the whole space
- Flexibility:
  - As the transitions and reward vector are given, the agent can adapt to any change by planning ahead

Drawbacks:

- Model bias/inaccuracy:
  - The quality of the representation influences estimates and decisions
- Complexity:
  - In practice, this approach is not scalable to big state spaces
- Unrealism:
  - In practice, it is very hard to have exhaustive information on the environment

Model-free:

When the agent does not have information about the transition and the rewards of the environment:

- It has to learn from experience
- It must store the value of explored states to inform decision making

The value function can be broken down into:
The reward right now + the future discounted reward

$$V^{\pi}(s_t) = E_{\pi}(R_t | s_t = s)$$
$$= E_{\pi}\left(\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s\right)$$

This leads to a recursive link between the current and future value function

$$
\begin{aligned}
V^{\pi}(s_t) &= E_{\pi}(R_t |\, s_t = s) \\
&= E_{\pi}\left(\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s \right)
\end{aligned}
$$

$$V^{\pi}(s_t) = r_t + \gamma V^{\pi}(s_{t+1})$$

This leads to a recursive link between the current and future value function

$$V^{\pi}(s_t) = E_{\pi}(R_t |\, s_t = s)$$
$$= E_{\pi}\left(\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s\right)$$

$$V^{\pi}(s_t) = r_t + \gamma V^{\pi}(s_{t+1})$$

Bellman equation:

- The optimization problem is broken down into sub-problems
- The best action that I can choose now is choosing the best action now and keep on choosing the best action afterwards

This leads to a recursive link between the current and future value function

$$
\begin{aligned}
V^{\pi}(s_t) &= E_{\pi}(R_t | \, s_t = s) \\
&= E_{\pi}\left(\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s \right)
\end{aligned}
$$

$$V^{\pi}(s_t) = r_t + \gamma V^{\pi}(s_{t+1})$$

Bellman equation:

- The optimization problem is broken down into sub-problems
- The best action that I can choose now is choosing the best action now and keep on choosing the best action afterwards

- In practice, it is very often implemented using **Dynamic Programming**

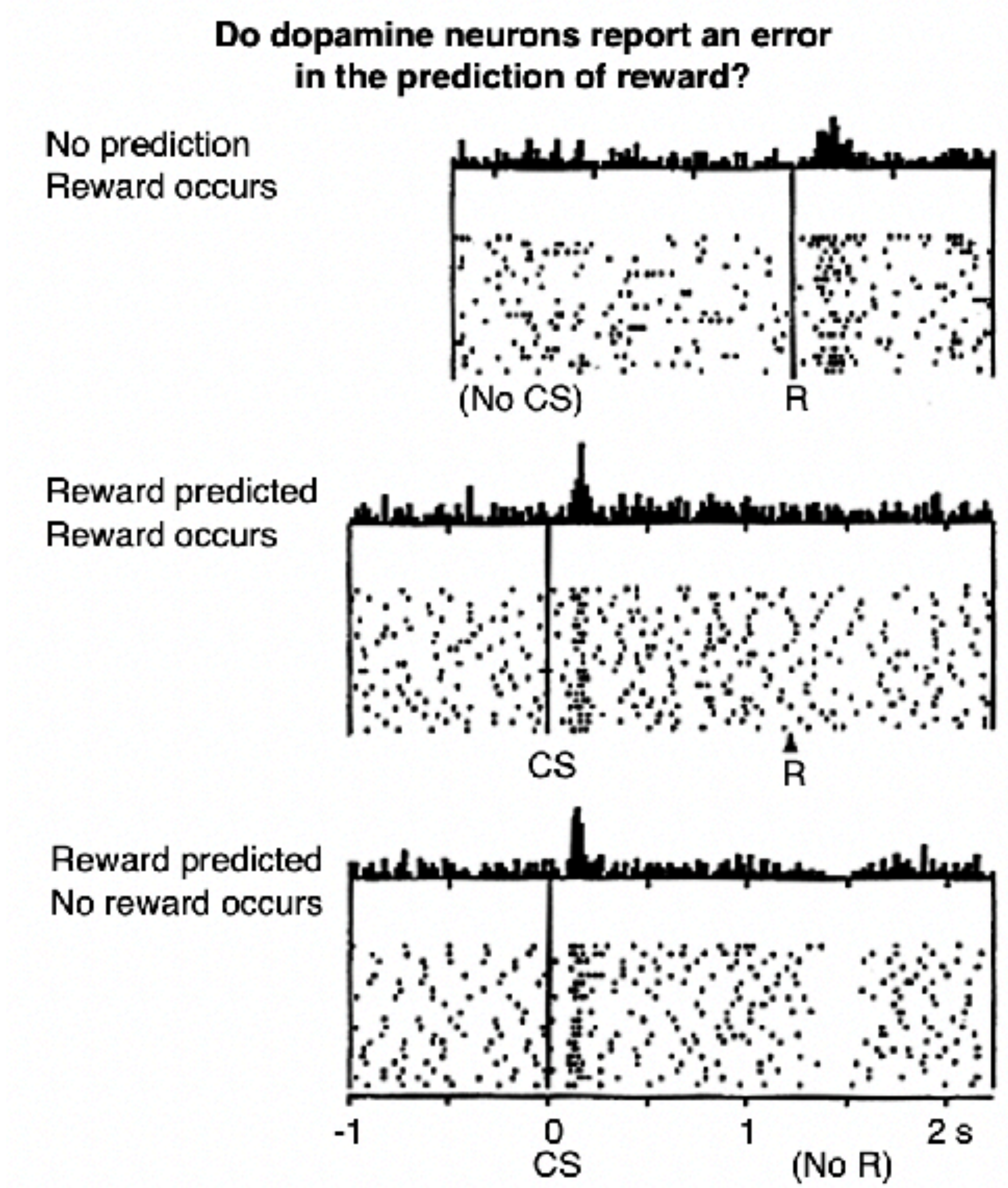
Temporal Difference Error:

$$V^{\pi}(s_t) = r_t + \gamma V^{\pi}(s_{t+1})$$

Bellman equation leads to the temporal difference error

$$\delta_t = r_t + \gamma V^{\pi}(s_{t+1}) - V^{\pi}(s_t)$$

Temporal Difference Error:



Sutton and Barto. *Reinforcement learning: An introduction*. MIT press. (2018)

Using bootstrapping to learn from experience:

$$V^{\pi}(s_t) = r_t + \gamma V^{\pi}(s_{t+1})$$

t=0 r=0 $\longleftrightarrow$

V( )= 0

Using bootstrapping to learn from experience:

$$V^{\pi}(s_t) = r_t + \gamma V^{\pi}(s_{t+1})$$

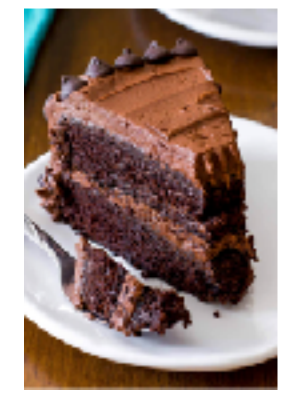
t=0 r=0 ⟷

V( )= 0

⟶

V( )= 1

⟷ r=1



Using bootstrapping to learn from experience:

$$V^{\pi}(s_t) = r_t + \gamma V^{\pi}(s_{t+1})$$

t=0

r=0 ⟷  ⟶  ⟷ r=1 

V( )= 0

V( )= 1

t=1

V( )= γ

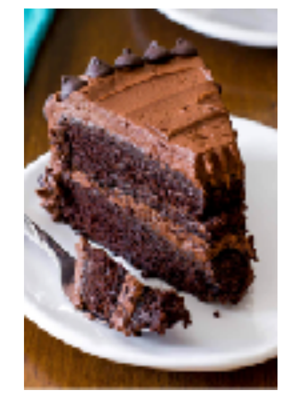
Using bootstrapping to learn from experience:

$$V^{\pi}(s_t) = r_t + \gamma V^{\pi}(s_{t+1})$$

t=0 r=0 $\longleftrightarrow$

$\longrightarrow$

$\longleftrightarrow$ r=1



V( )= 0 V( )= 1

t=1 V( )= γ

Also applies to Q-learning, in which the value of the state action pair is updated

Model-free:

Benefits:

- Fast/computationally simple:
  - The estimate can be used as soon as the first experience is made
  - Only one cached value
- Versatile:
  - Very easy to apply to learning many different problems
- Robustness:
  - Robust to model inaccuracies

Model-free:

Benefits:

- Fast/computationally simple:
  - The estimate can be used as soon as the first experience is made
  - Only one cached value
- Versatile:
  - Very easy to apply to learning many different problems
- Robustness:
  - Robust to model inaccuracies

Drawbacks:

- Sample inefficiency:
  - One need many experiences to have a correct estimate of the value function
  - Leads to inflexibility to changes
- Convergence:
  - Hardly converge to the optimal policy
- Exploration-exploitation trade-off:
  - As new knowledge depends on past knowledge

Model-based vs model-free
Summary

Model-based:

- Uses all the branches of the graph
- Agent has access to a model of the world, including transitions and rewards
- Computationally expensive, but flexible

Model-free:

- Uses a single cached estimate
- Agent is blind to the transitions and reward structures
- Computationally simple, but not flexible

Model-based vs model-free
Summary

Model-based:
- Uses all the branches of the graph
- Agent has access to a model of the world, including transitions and rewards
- Computationally expensive, but flexible

Model-free:
- Uses a single cached estimate
- Agent is blind to the transitions and reward structures
- Computationally simple, but not flexible

In reality:
Model-based leads to a form of goal-directed behaviour.
Model-free leads to a form of habitual behaviour.
In the examples presented, all assume full knowledge of the state space from the agent.

But wait… what does 'model' mean? Can humans really access it?

How to test it: change the transition structure

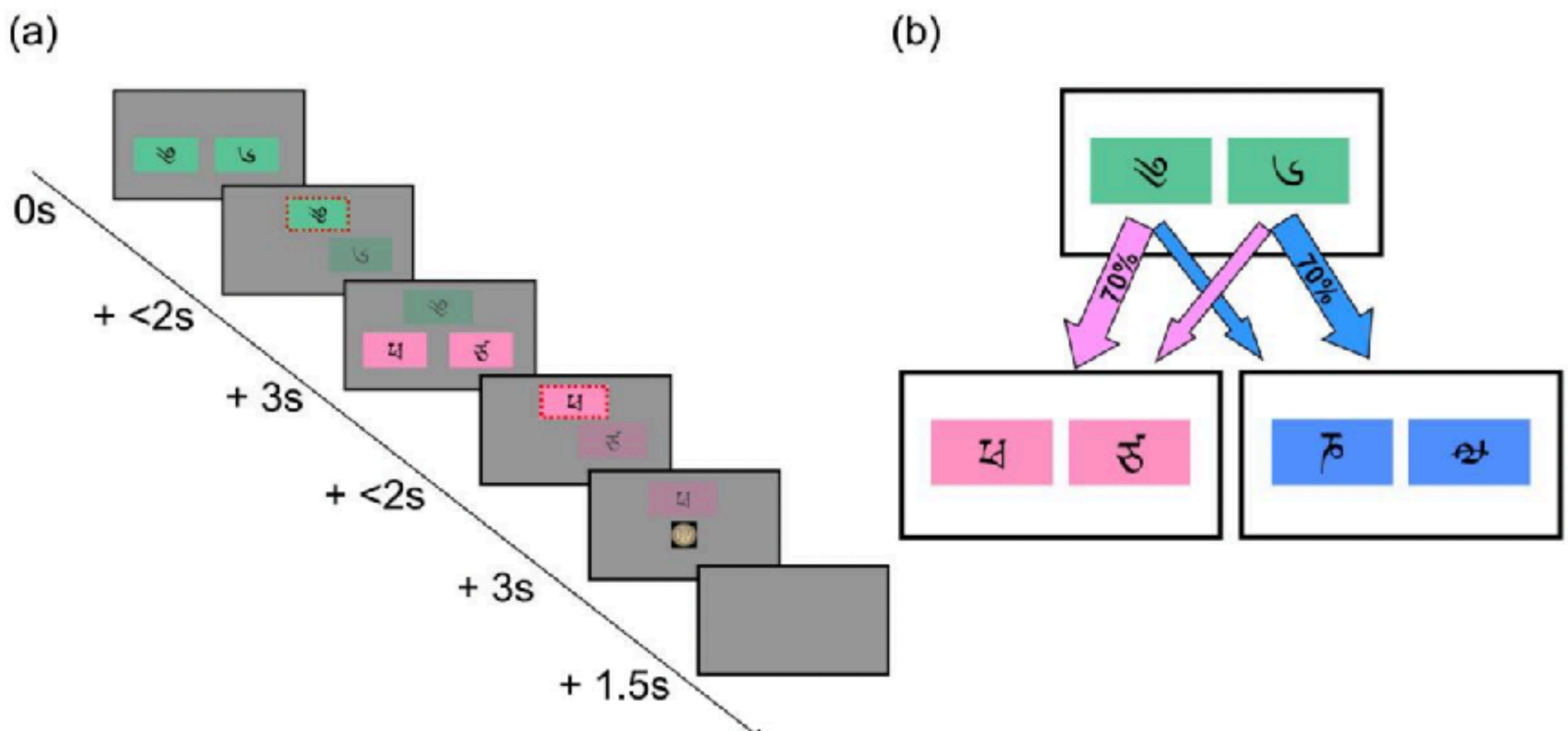
- A model-free agent will take very long, if ever, to adjust
- A model-free agent will automatically adapt

But wait… what does 'model' mean? Can humans really access it?

How to test it: change the transition structure

- A model-free agent will take very long, if ever, to adjust
- A model-free agent will automatically adapt

Key example: the 2-steps task (Daw, *et al.* Neuron, 2011):

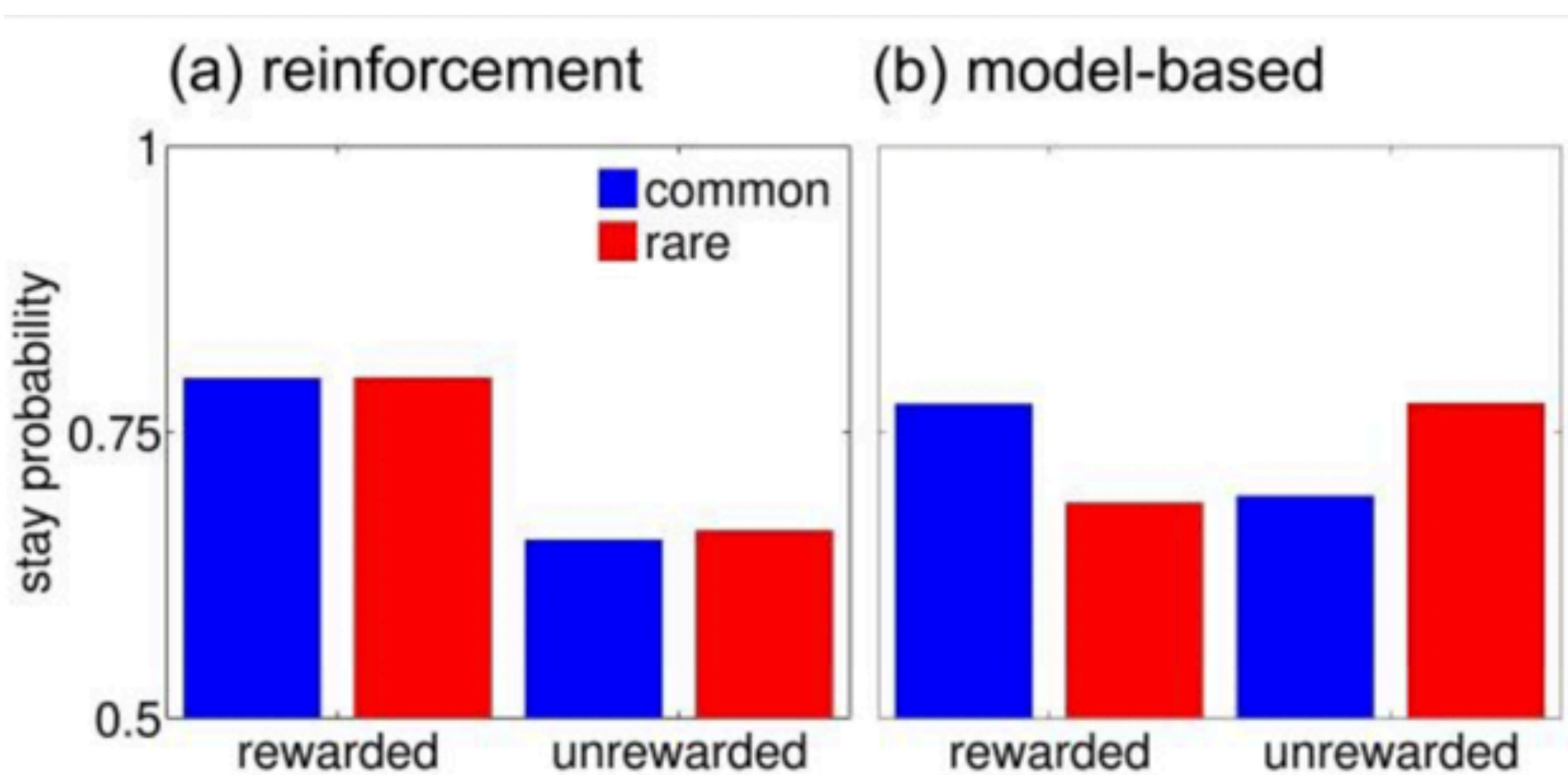


But wait… what does 'model' mean? Can humans really access it?

Key example: the 2-steps task (Daw, *et al.* Neuron, 2011):

Intuition:

- A model based agent knows the transition structure, therefore should be less sensible to negative reward prediction errors after a rare transition
- A model-free agent will adapt its behaviour to experience regardless of the frequency of the transition



But wait… what does 'model' mean? Can humans really access it?

Humans are in between… to be continued…