
download these slides from: github.com/qtabs/compneuro/05_learning.{key, pdf}

# Questions from last session

# Q4.3 (Jun)

Predictive coding assumes the brain's internal model approximates the true generative process that creates sensory data. Find a few examples where the brain's model is systematically wrong. Why would the brain make these errors, if the strategy of using GMs is optimal?

# Q4.4 (Jingwen)

The McGurk effect demonstrates that visual information about mouth movements can change what we hear. Explain this phenomenon in terms of predictive coding. How does the brain treat different sensory modalities when forming its generative model?

# Q4.5 (Sebastian)

Many scientists claim the Free Energy Principle cannot be falsified. Research this controversy: what do critics mean by "unfalsifiable"? What is the difference between a framework that cannot be wrong and a framework that is not empirically testable? Give your own position on the Free Energy falsifiability debate.

# Q4.8 (Shuping)

Omission responses are neural responses to the unexpected absence of a stimulus. How does predictive coding explain these responses? In an omission, what is the $x$, what is $y$, and what is the prediction error? Do omission responses provide strong evidence for predictive coding or the Bayesian Brain hypothesis?

# Q4.10 (Itsaso)

Psychedelic drugs may work by disrupting the predictive apparatus supporting perception. Research this hypothesis. What are the therapeutic and ethical implications?

# Reinforcement Learning

## Computational Neuroscience - Lecture 5

Alejandro Tabas

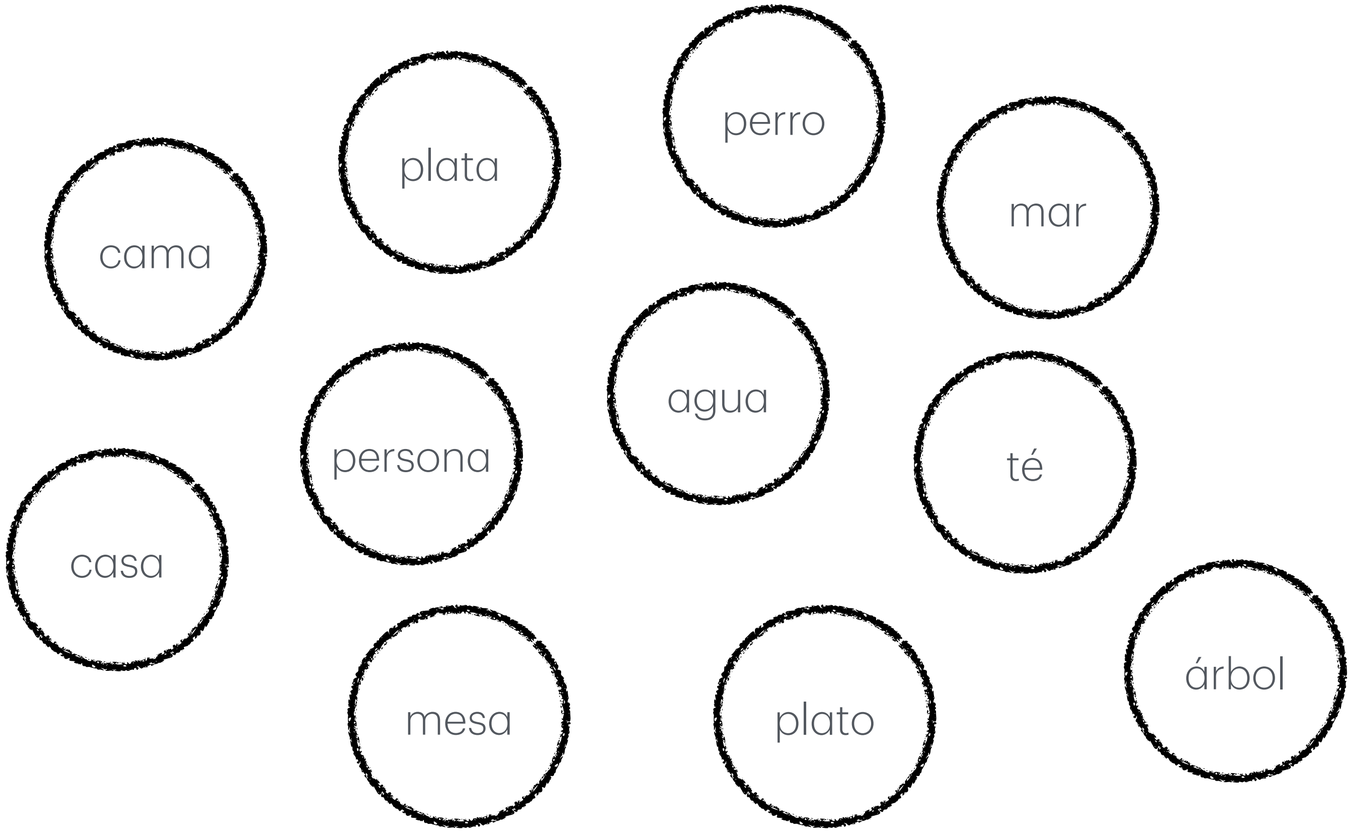
# Reinforcement Learning

1. Two types of learning
2. Reinforcement learning
3. The $k$-armed bandit problem
4. The $\varepsilon$-greedy algorithm
5. Markov Decision Problems
6. Conclusions

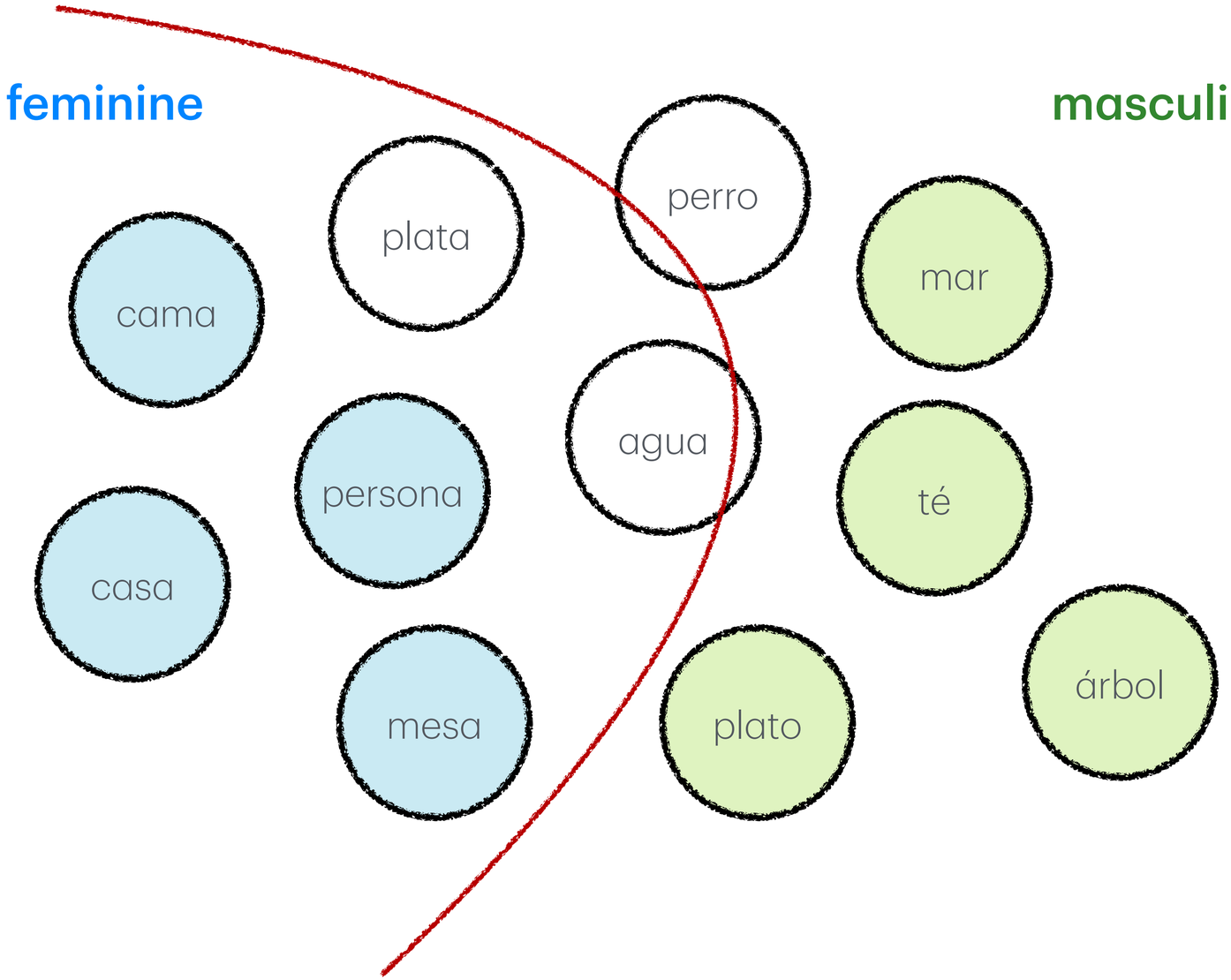
# Reinforcement Learning

1. **Two types of learning**
2. Reinforcement learning
3. The $k$-armed bandit problem
4. The $\varepsilon$-greedy algorithm
5. Markov Decision Problems
6. Conclusions

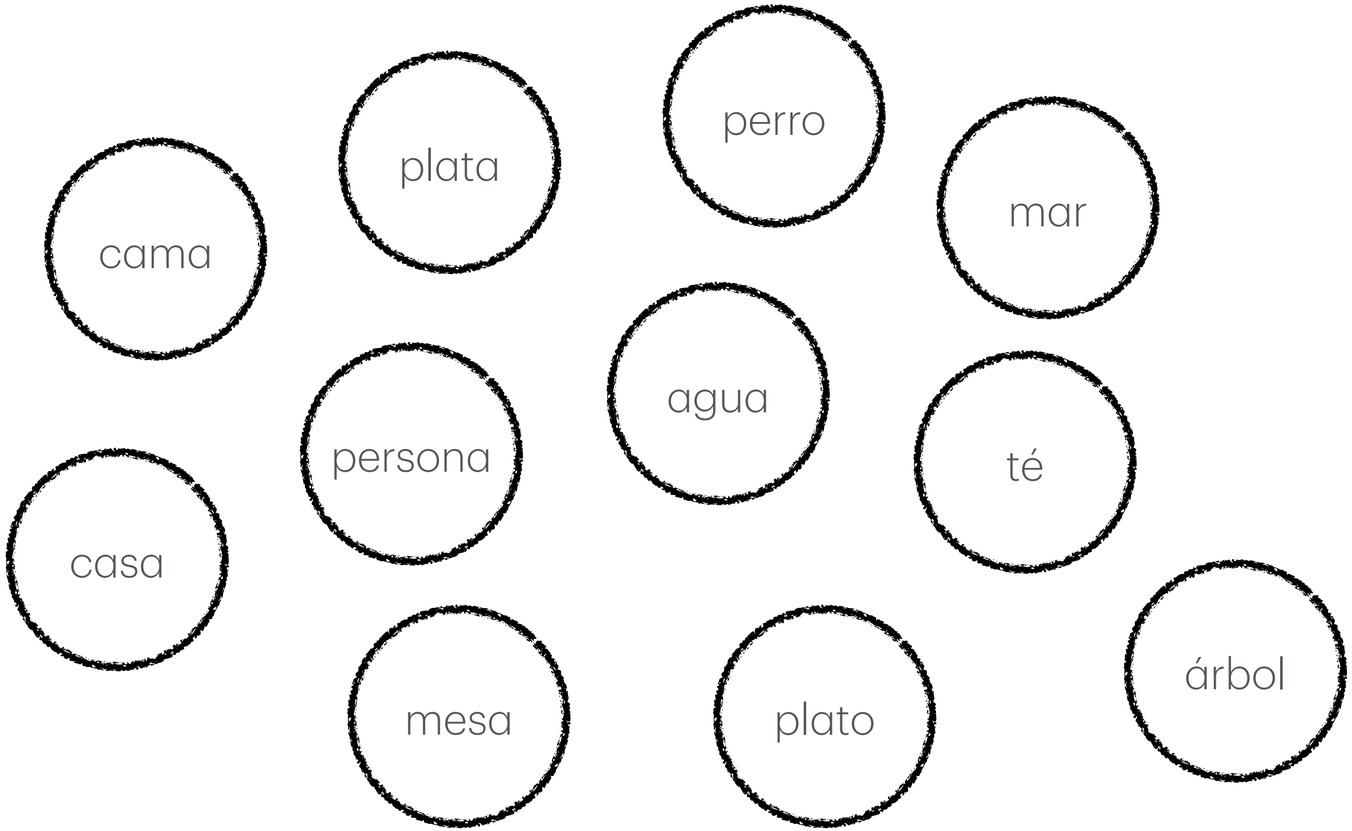
# A learning problem: word gender in Spanish



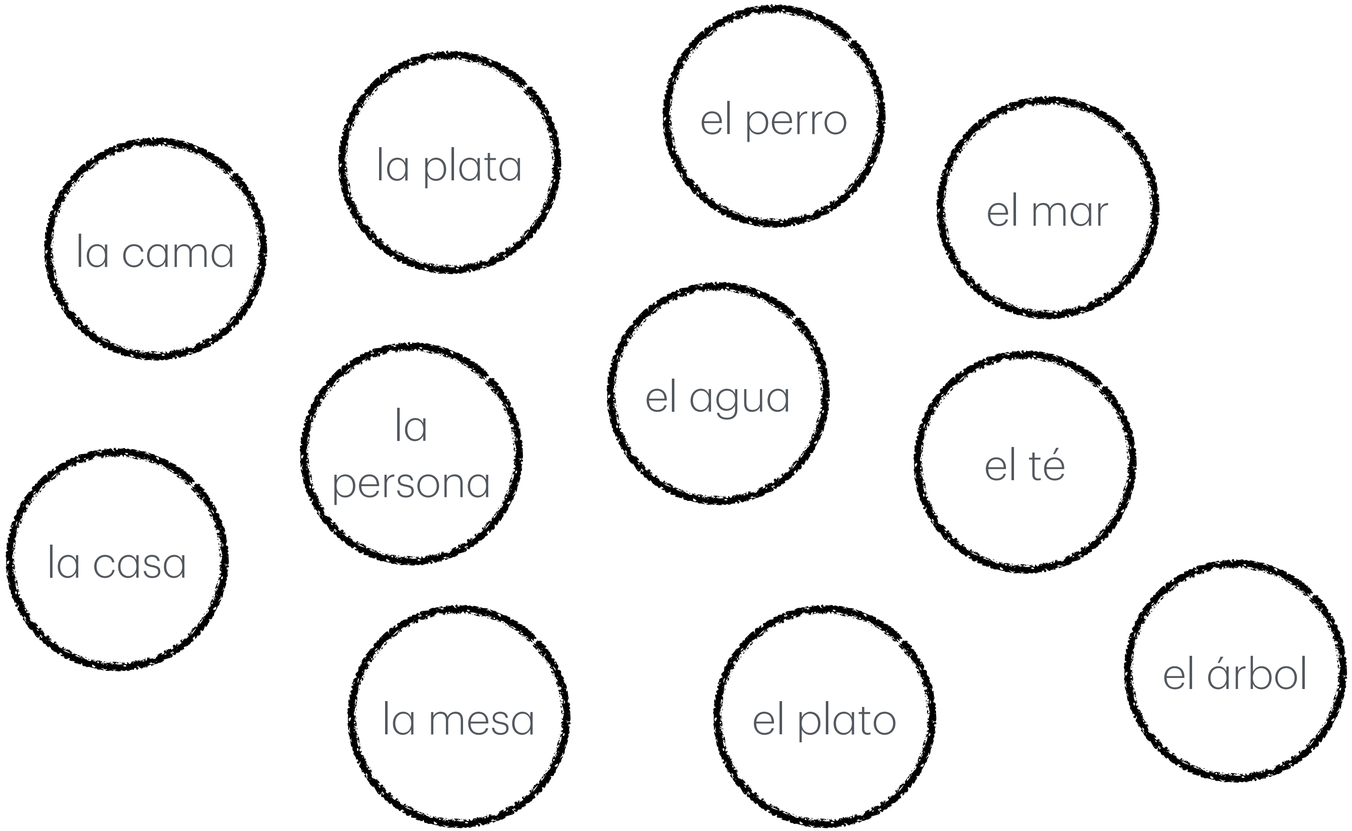
# Supervised learning



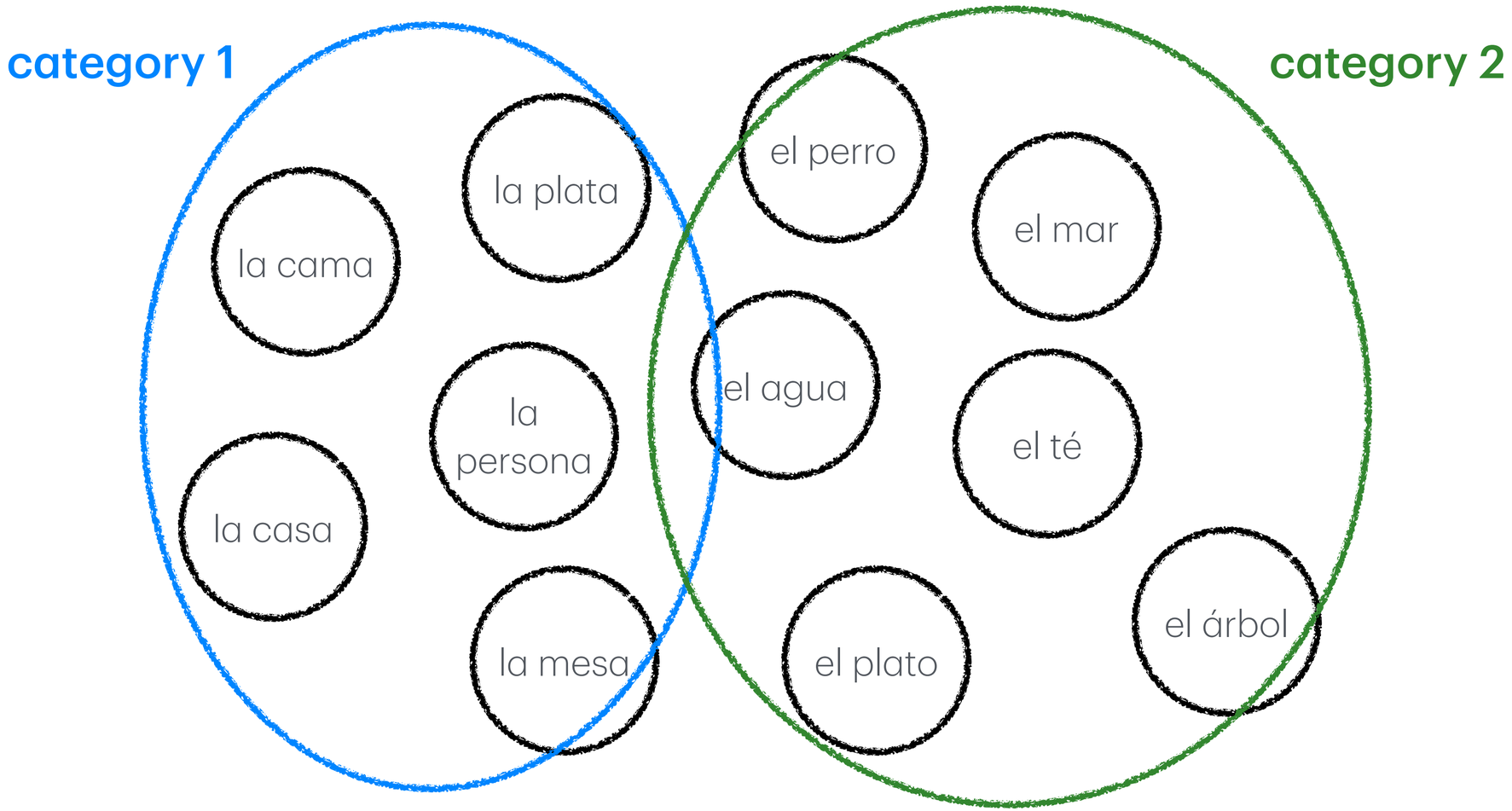
# Unsupervised learning



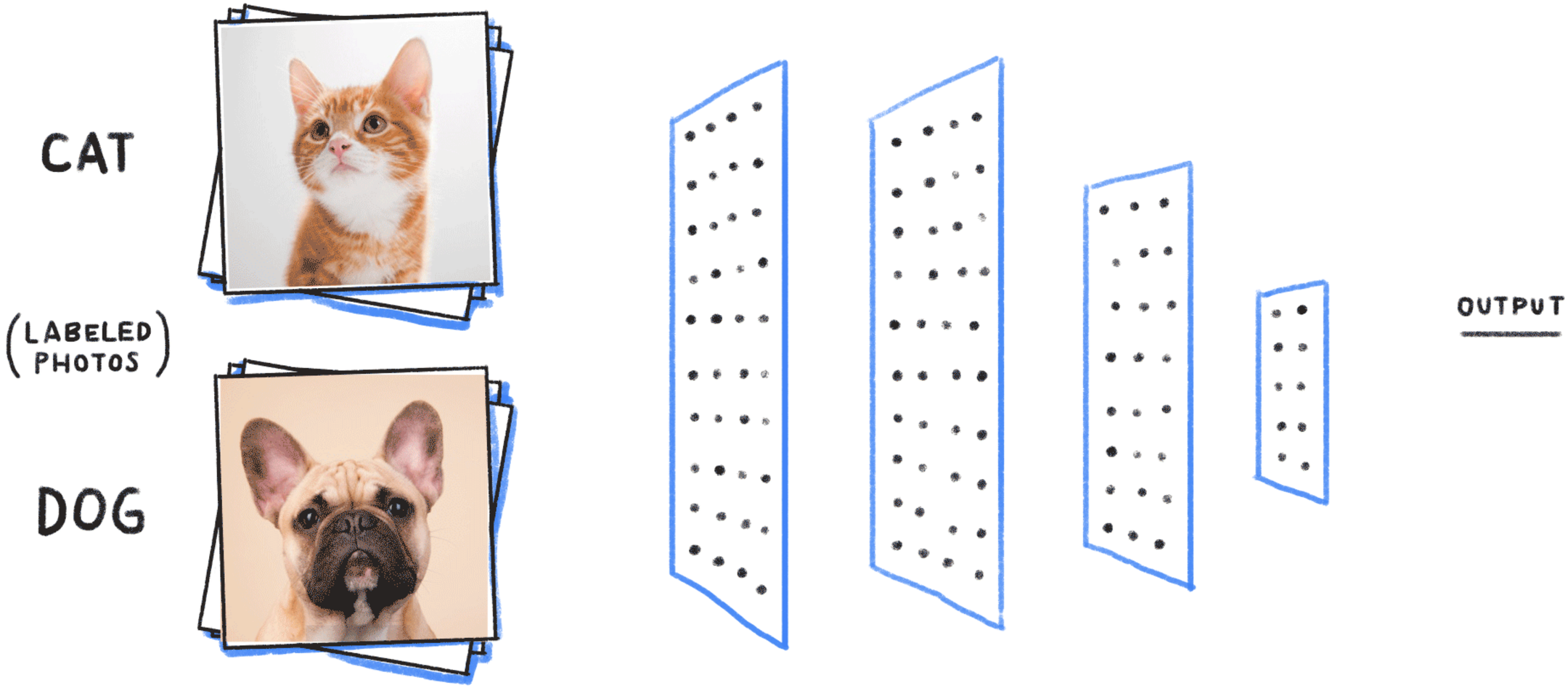
# Unsupervised learning



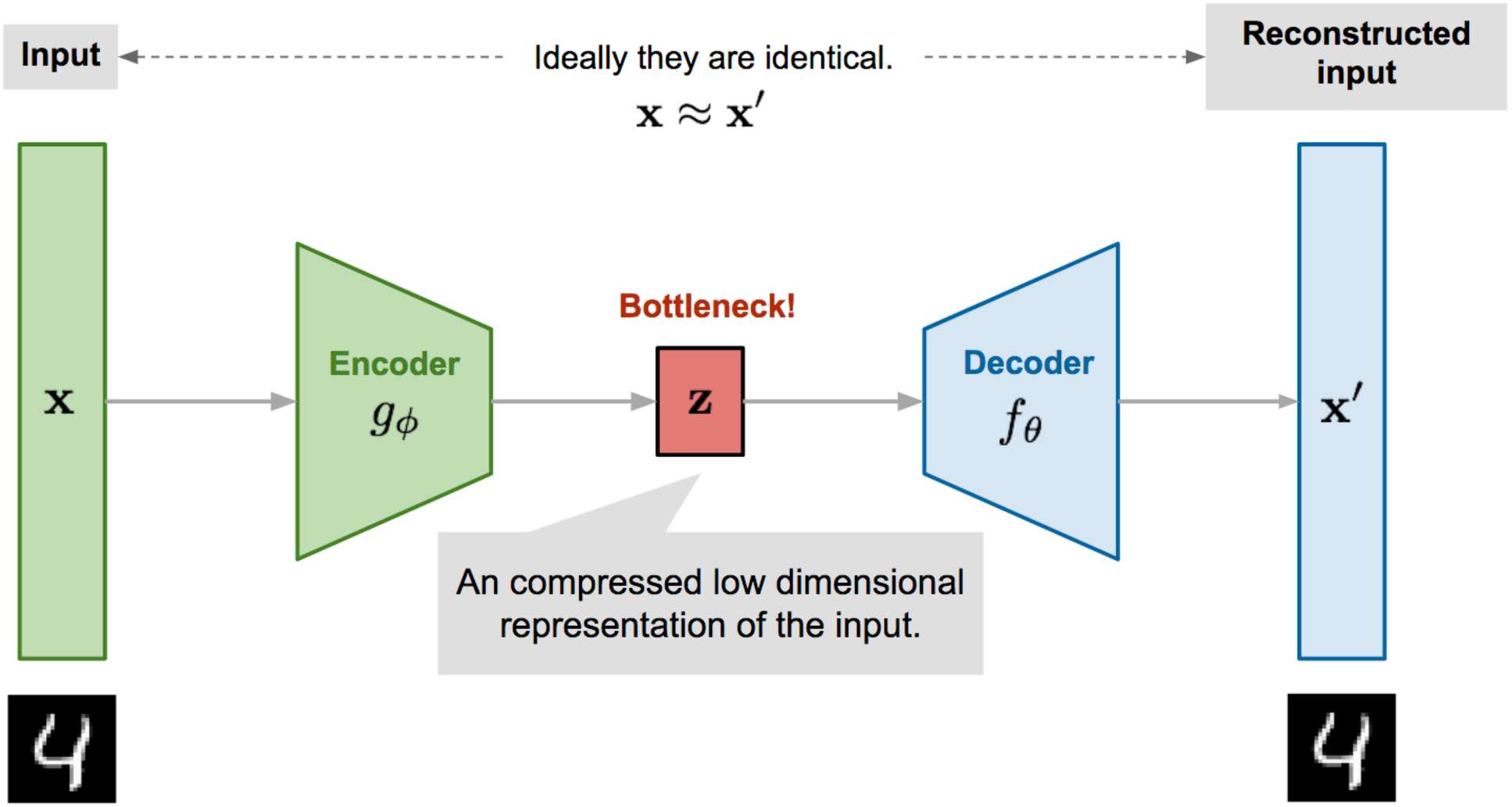
# Unsupervised learning



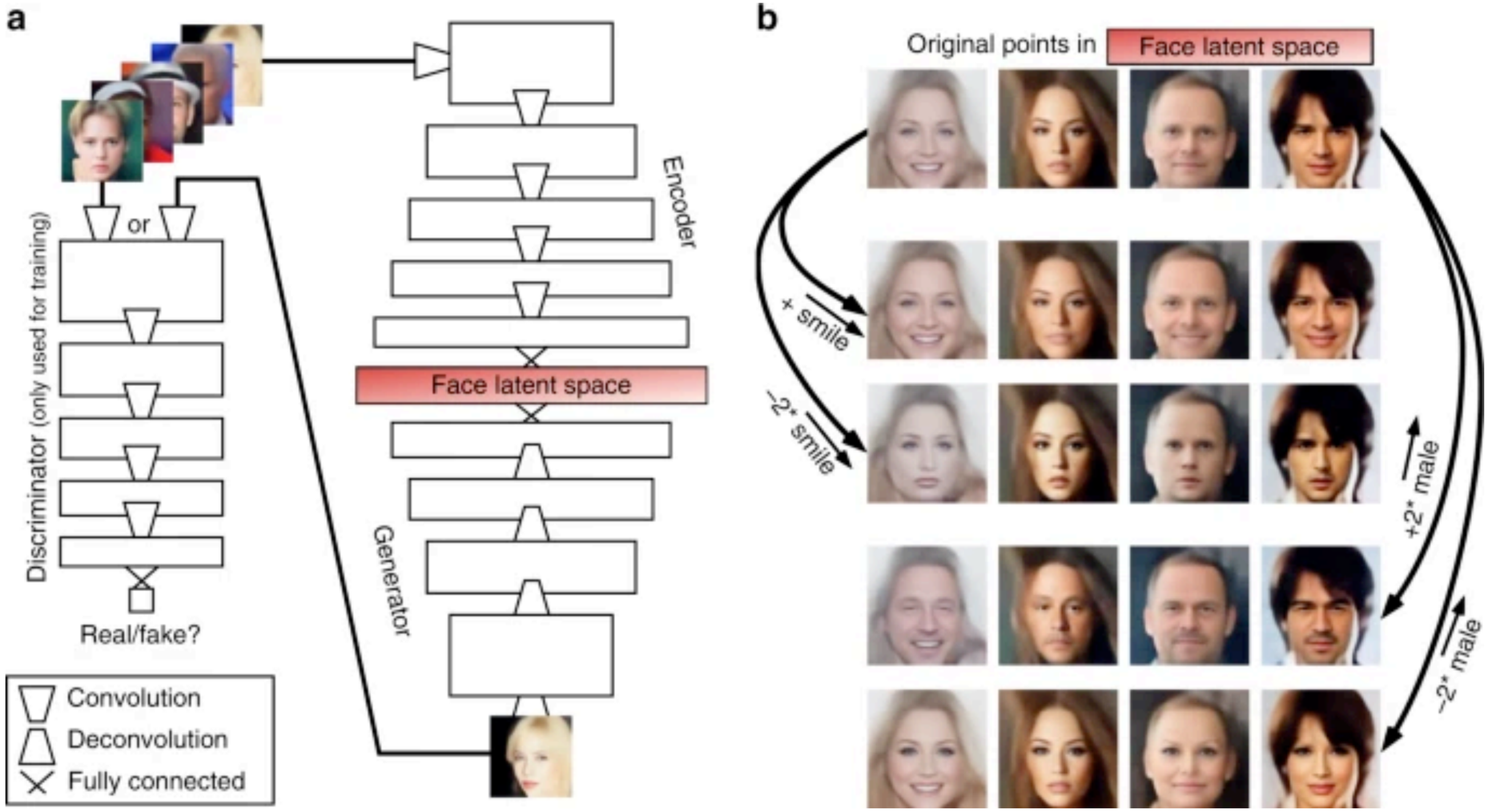
# Supervised learning example: ANN classifiers



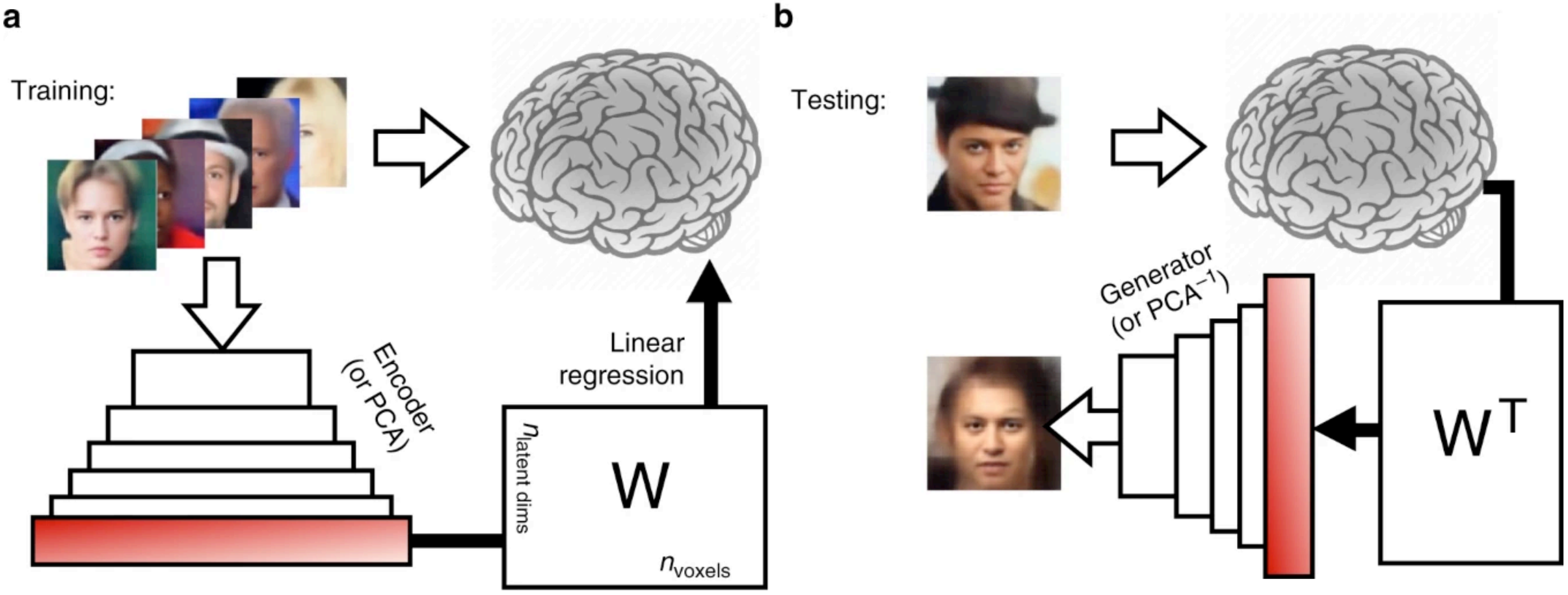
# Unsupervised learning example: autoencoders



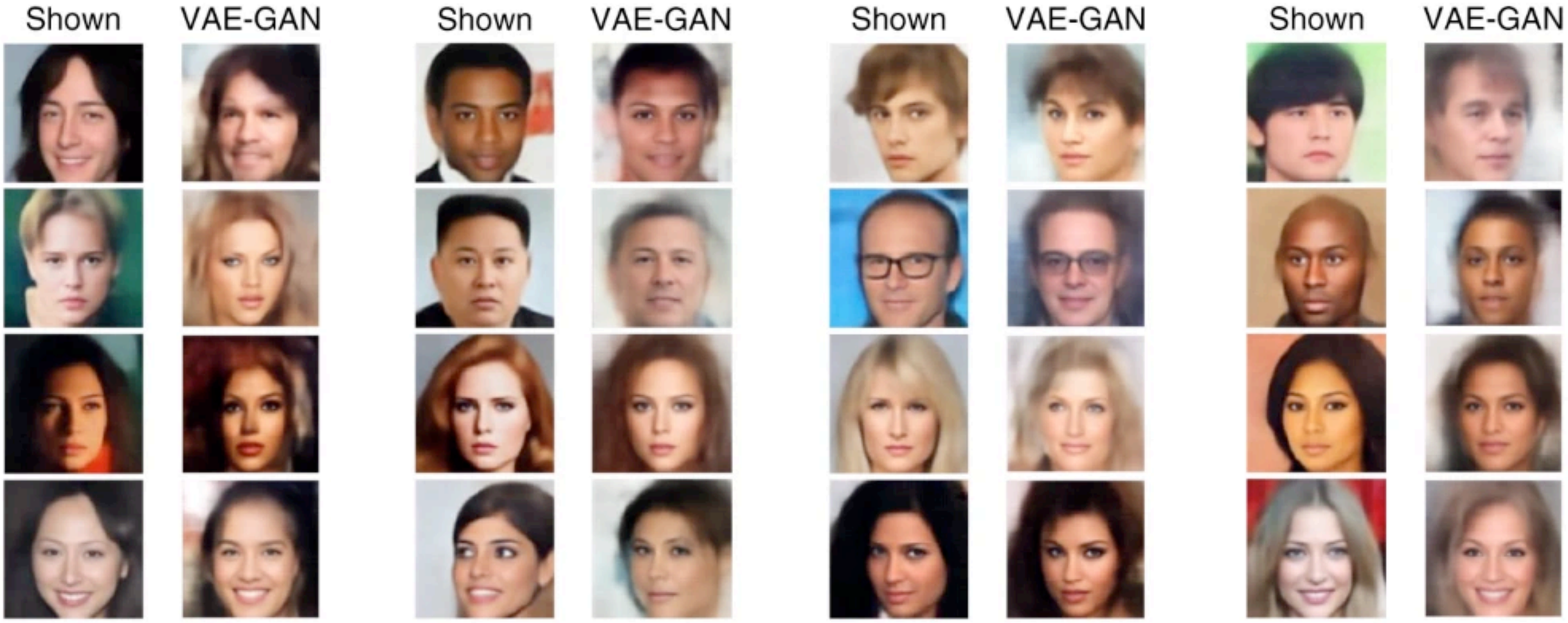
# Unsupervised learning example: GANs



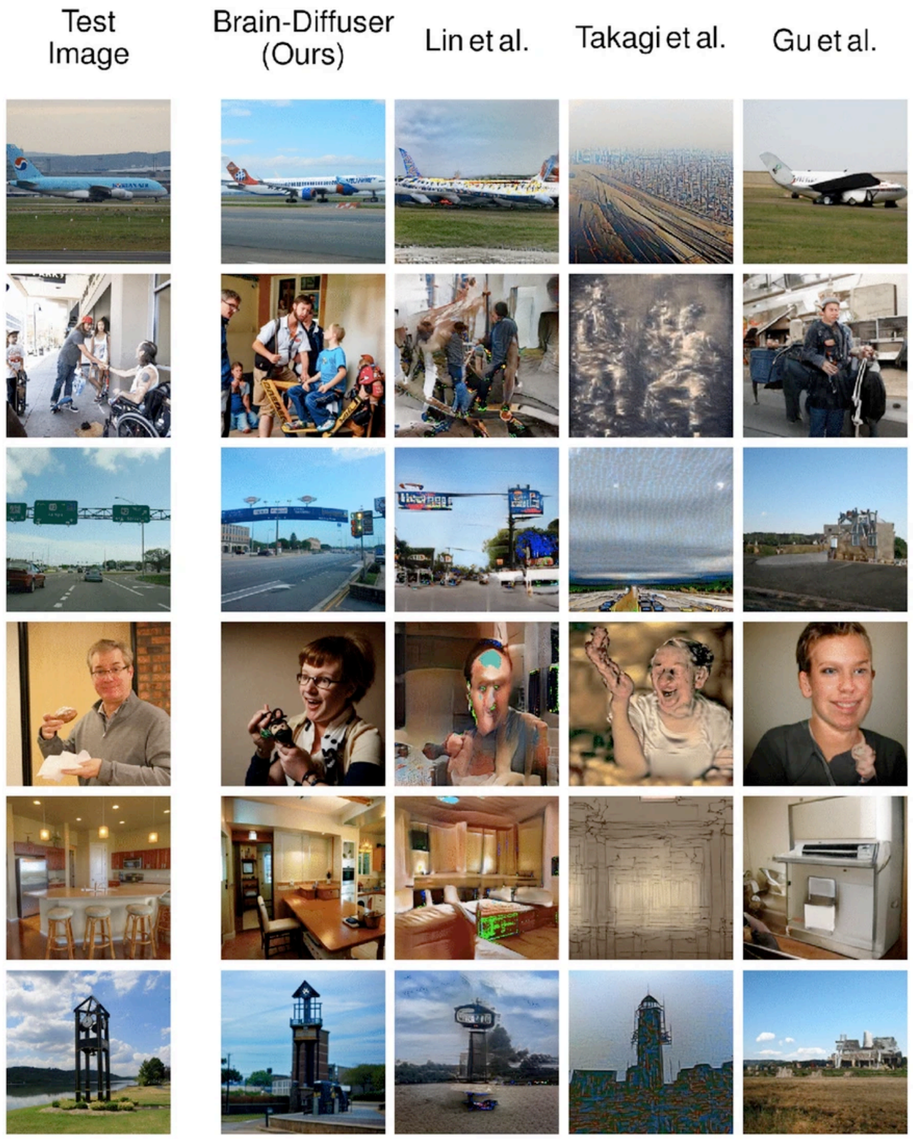
# Mind reading via latent-space decoding

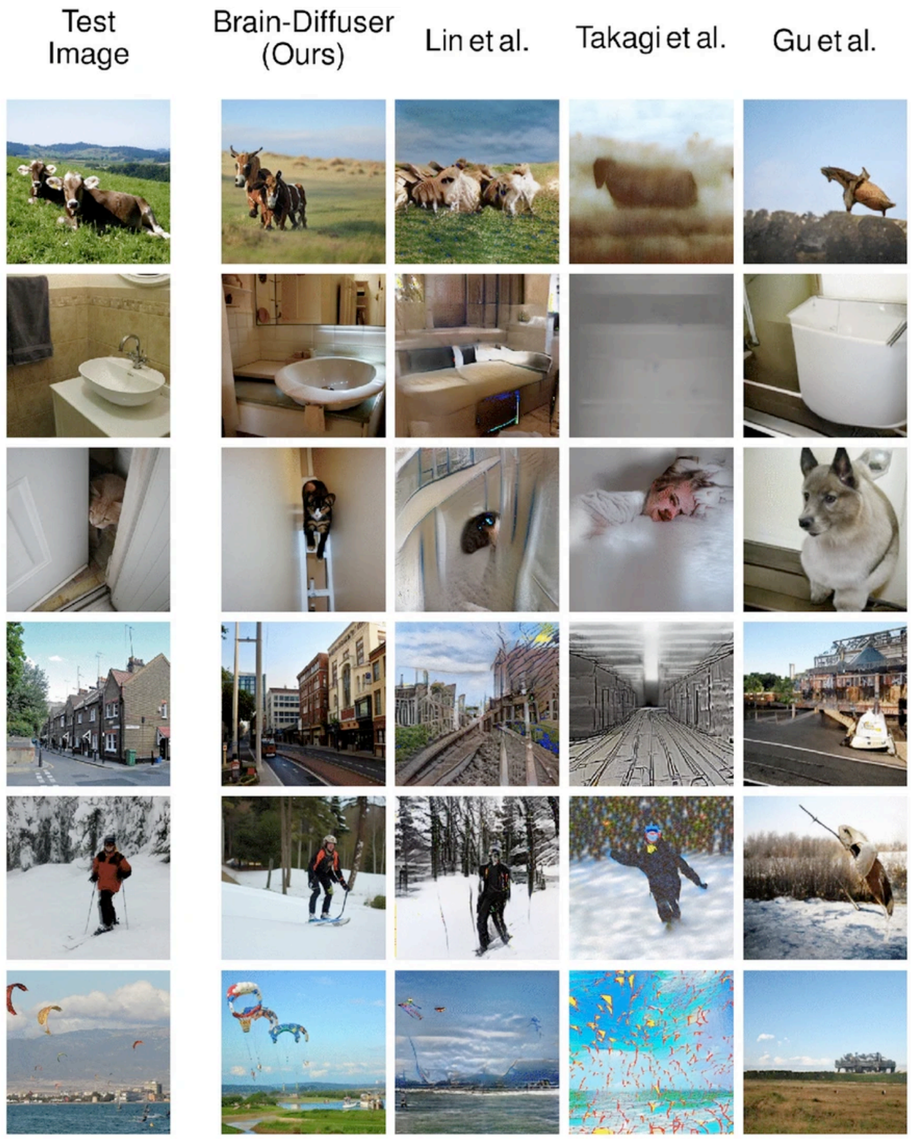


# Mind reading via latent-space decoding (2009)



# Mind reading via latent-space decoding (2009)

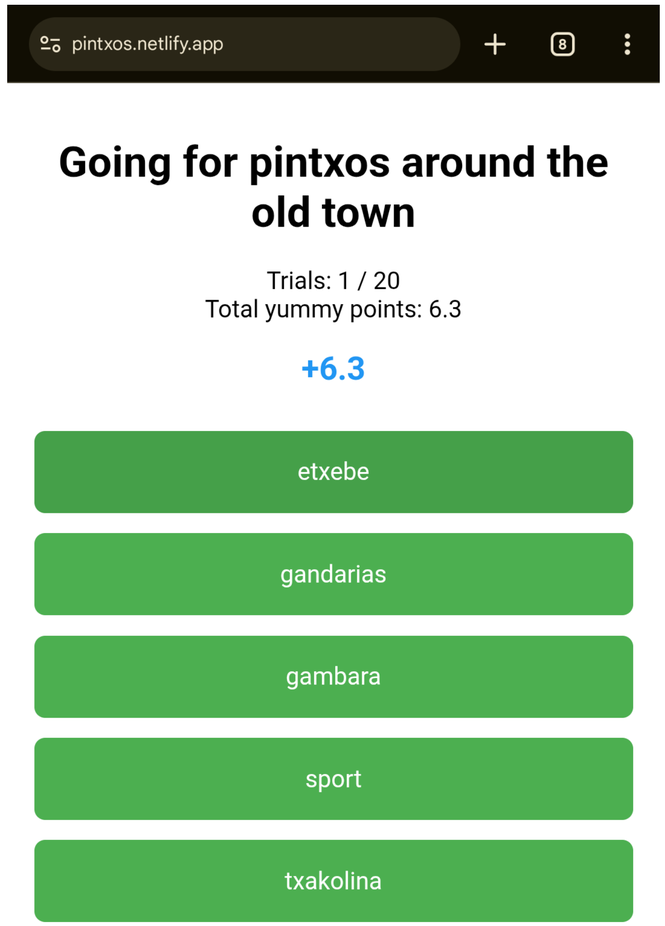


# Reinforcement Learning

1. Two types of learning
2. **Reinforcement learning**
3. The $k$-armed bandit problem
4. The $\varepsilon$-greedy algorithm
5. Markov Decision Problems
6. Conclusions

# pintxos.netlify.app



# How do we learn IRL?

- What strategy have you used to maximise your reward in the game?
- How do you learn:
  - what are your favourite bars and restaurants in a new city?
  - what type of fruits or beers or international cuisines you like?
  - who do you like the most within your social circles?
- Are these strategies supervised, or unsupervised?

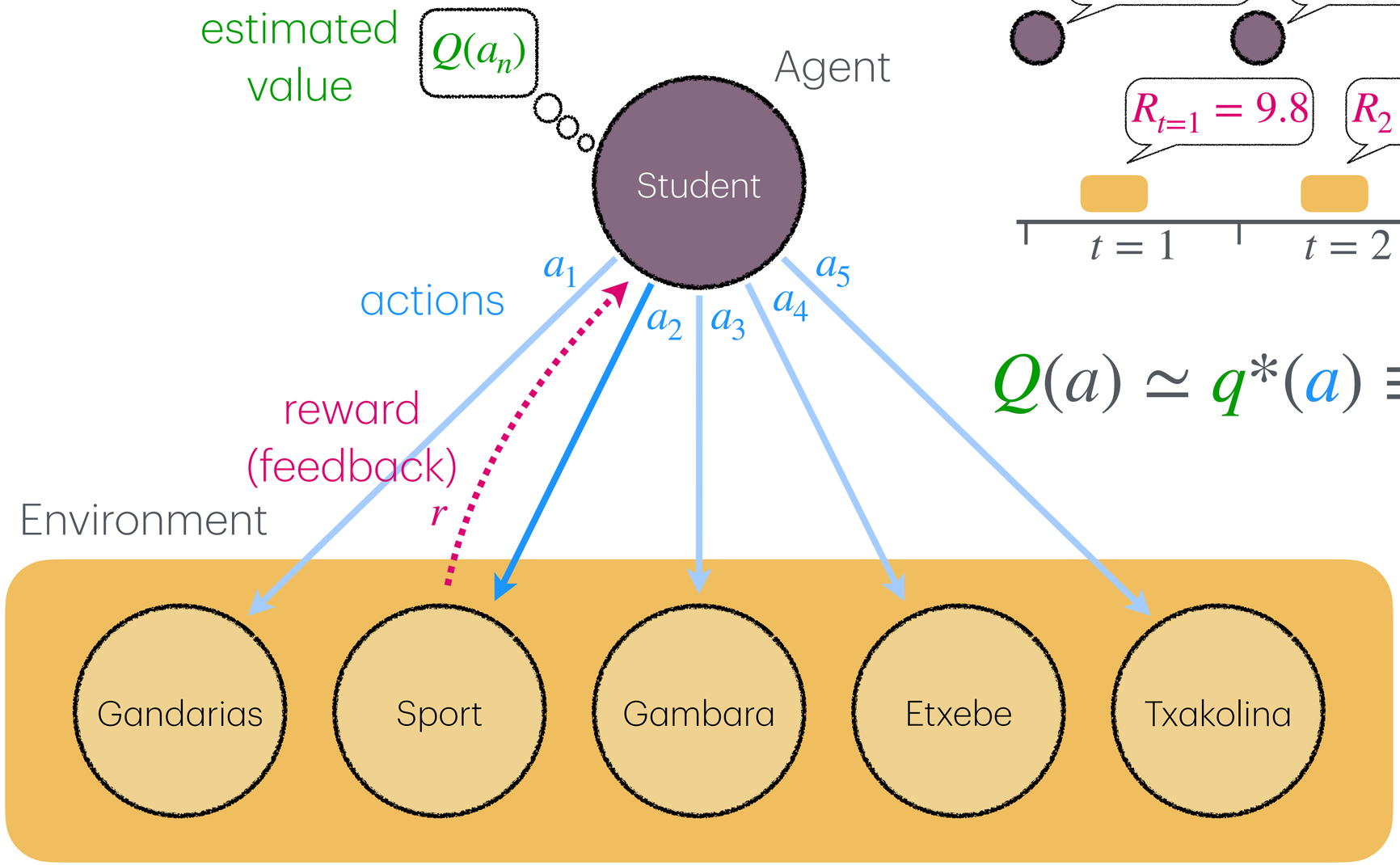
# Three types of learning

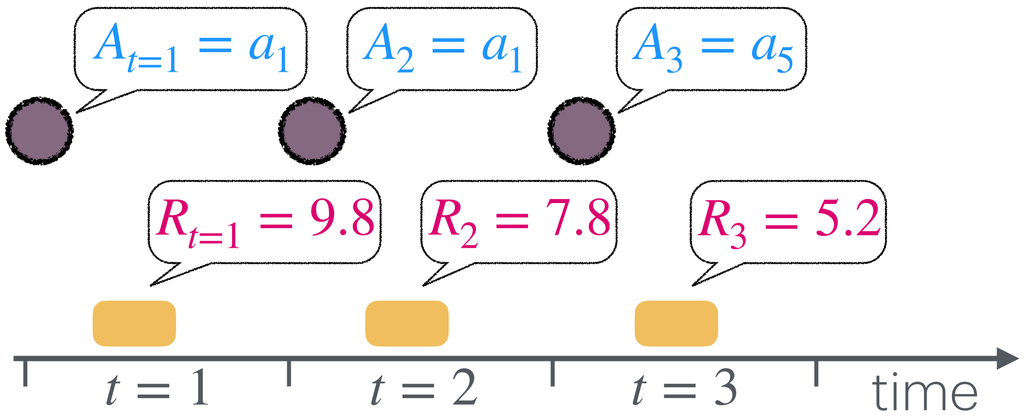

- Supervised learning (ground truth provided)
  - Learning words in a new language
  - Learning the multiplication tables
- Reinforcement learning (feedback provided, ground truth inaccessible)
  - Learning how to walk
  - Learning your favourite icecream flavour
- Unsupervised learning (nothing is provided)
  - Learning internal representations of the world
  - Learning the grammatical rules of L1 during development

# Reinforcement Learning

1. Two types of learning
2. Reinforcement learning
3. **The $k$-armed bandit problem**
4. The $\varepsilon$-greedy algorithm
5. Markov Decision Problems
6. Conclusions

# The $k$-armed bandit




$$Q(a) \simeq q^*(a) \equiv E[\, R_t \,|\, A_t = a \,]$$

# Estimating values

$$Q(a) \simeq q^*(a) \equiv E[\, R_t \,|\, A_t = a \,]$$

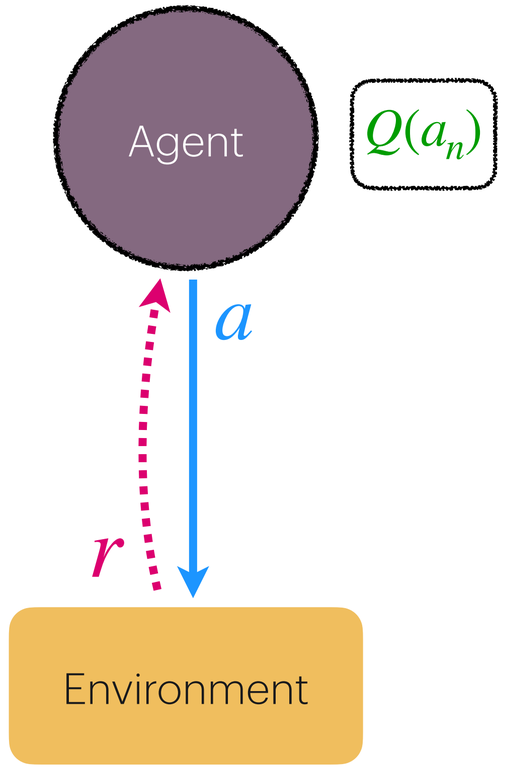
What would be the best way to estimate $Q$?

- Averaging: $Q_t(a) = \frac{1}{N_a} \sum_{t\,:\,A_t=a} R_t$

- Updating: $Q_t(a) = Q_{t-1}(a) + \frac{1}{N_a}\left(R_t - Q_{t-1}(a)\right)$

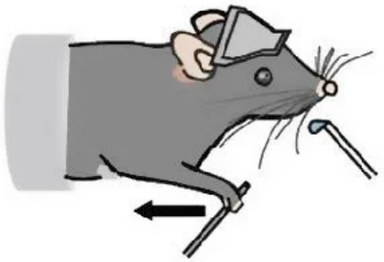
error $e$

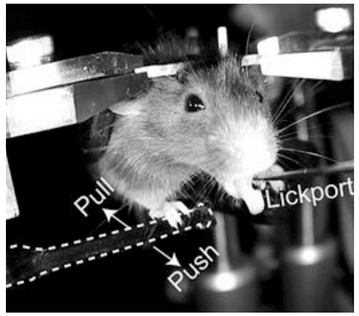
$$Q(a) \rightarrow Q(a) + \eta\, e$$

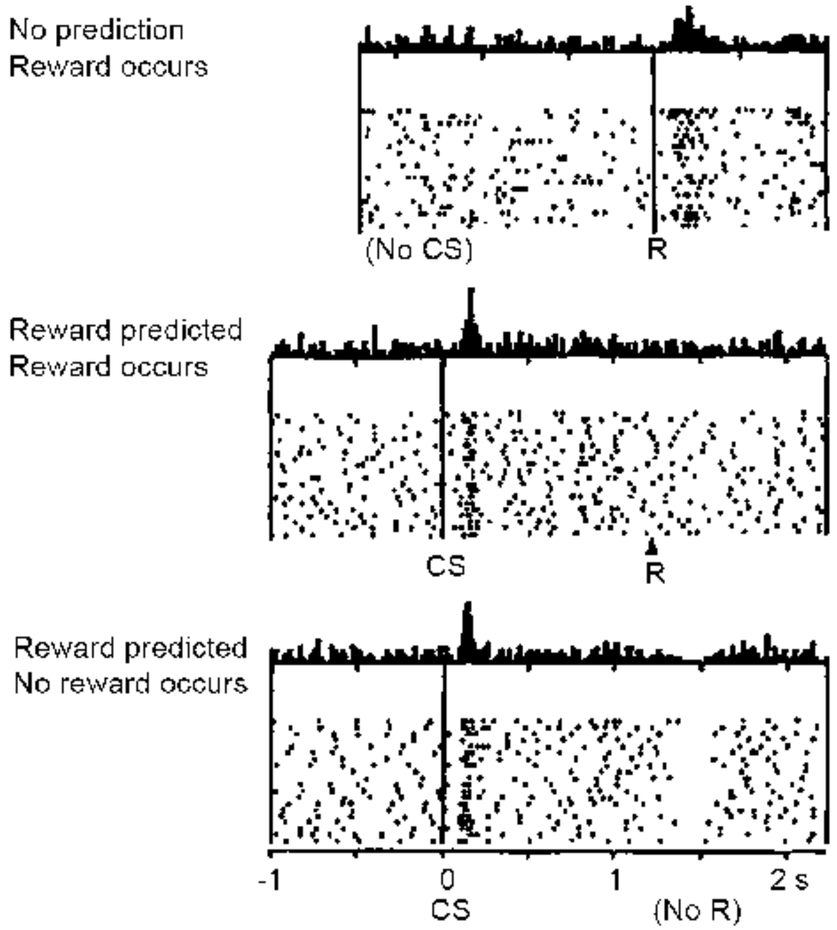


# Dopamine release and reward prediction error

$$Q(a) \rightarrow Q(a) + \eta\, e$$



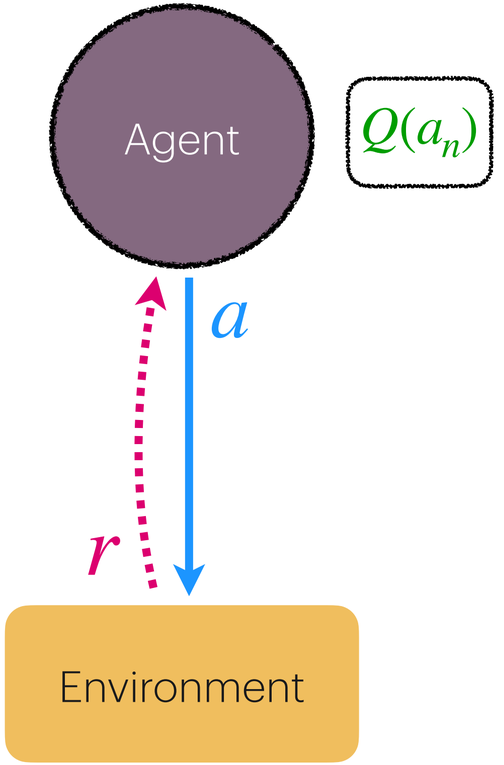



$$Q_t(a) > Q_{t-1}(a)$$
$$e > 0$$

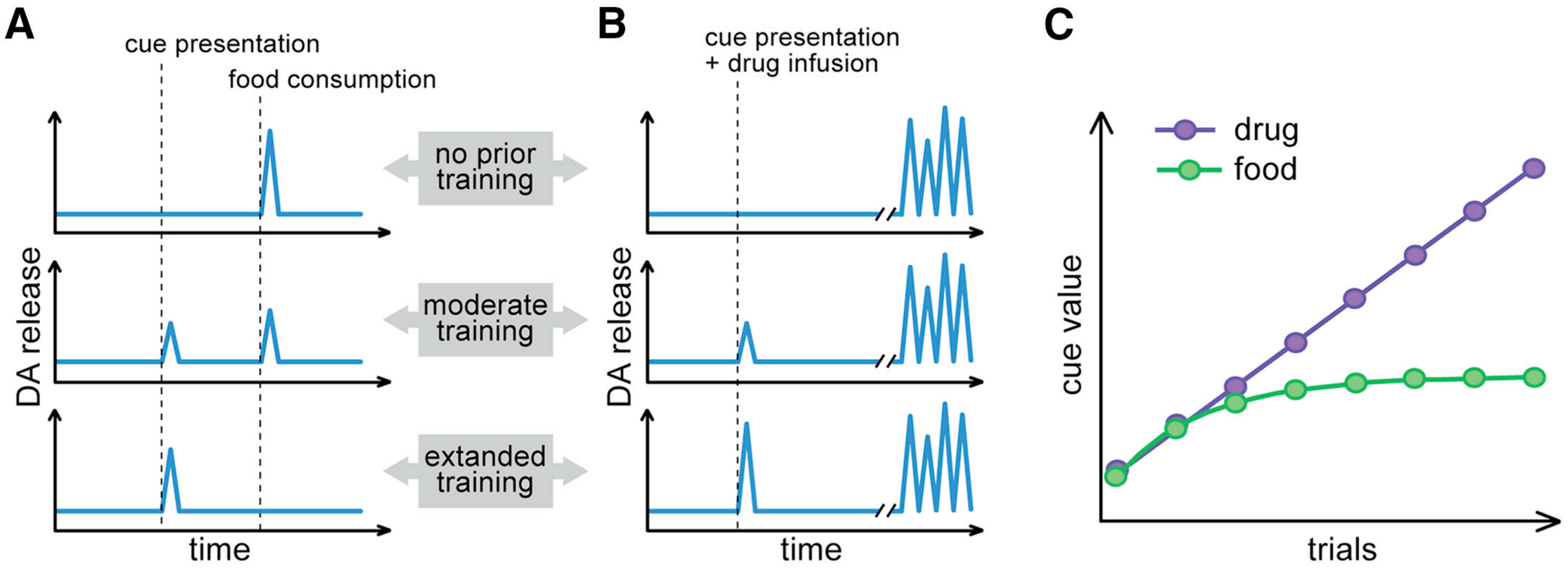
$$Q_t(a) = Q_{t-1}(a)$$
$$e = 0$$

$$Q_t(a) < Q_{t-1}(a)$$
$$e < 0$$



# Cocaine addiction and reinforcement learning

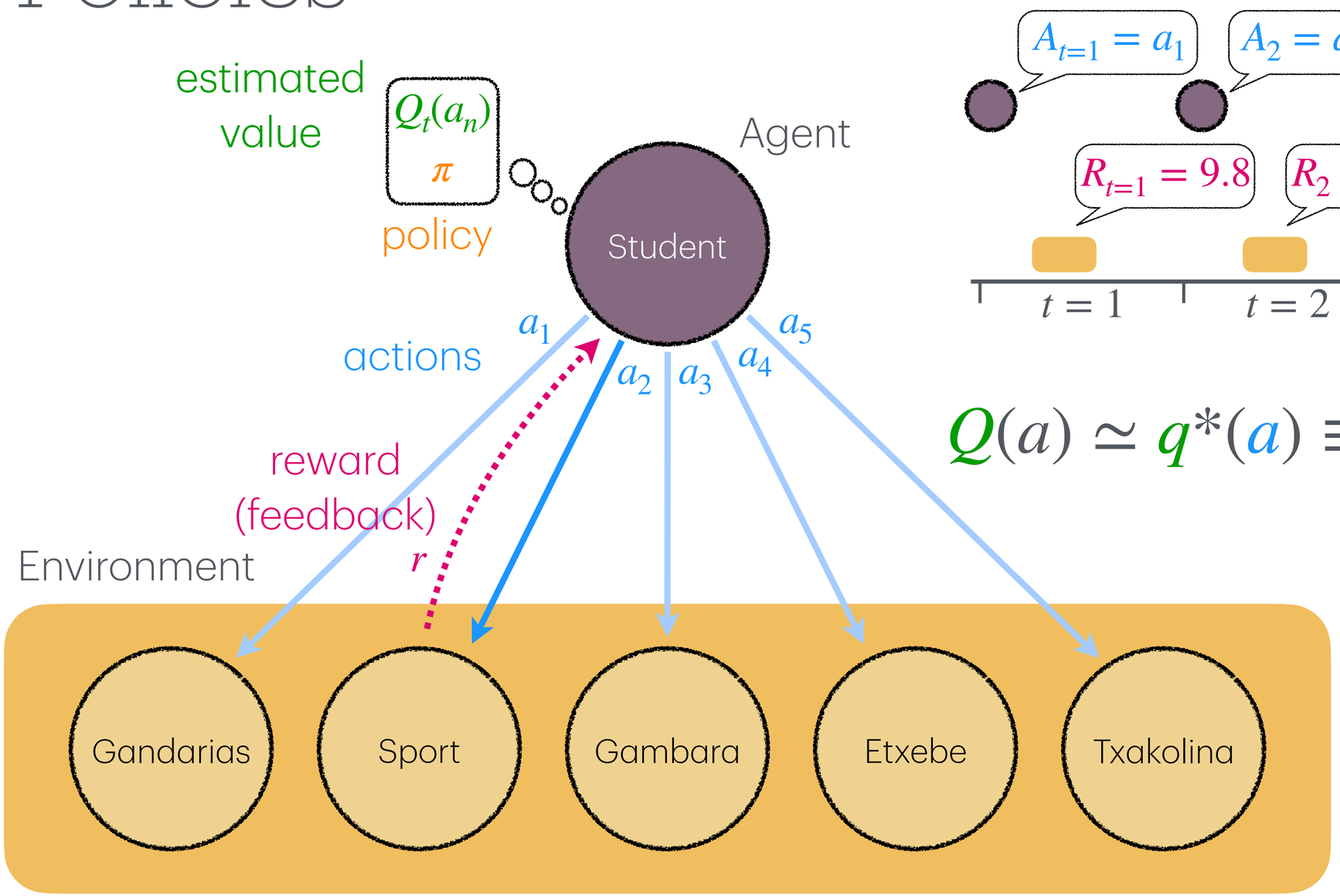
$$Q(a) \rightarrow Q(a) + \eta\, e$$

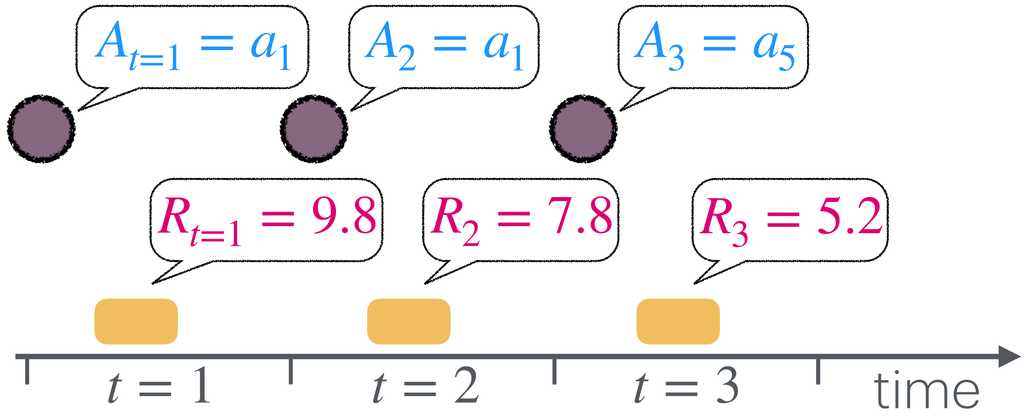


# Reinforcement Learning

1. Two types of learning
2. Reinforcement learning
3. The $k$-armed bandit problem
4. **The $\varepsilon$-greedy algorithm**
5. Markov Decision Problems
6. Conclusions

# Policies




$$Q(a) \simeq q^*(a) \equiv E[\, R_t \,|\, A_t = a \,]$$

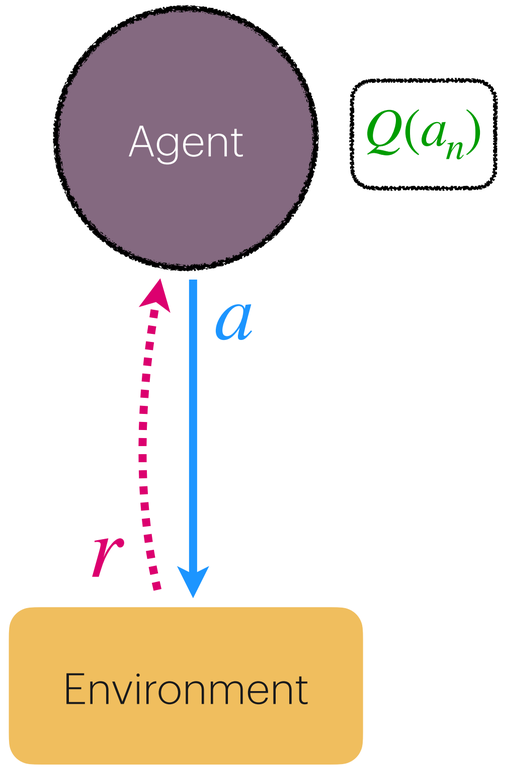
$$\pi : Q_t(a_n) \rightarrow A_{t+1}$$

$$A_{t+1} = \pi\left(Q_t(a_n)\right)$$

# The greedy policy

What would be the best policy to maximise reward?

$$A_{t+1} = \pi\left(Q_t(a_n)\right) = \arg\max_n Q_t(a_n)$$




Is there any problem with such policy?

- This is called the **greedy** policy

# Exploration and exploitation

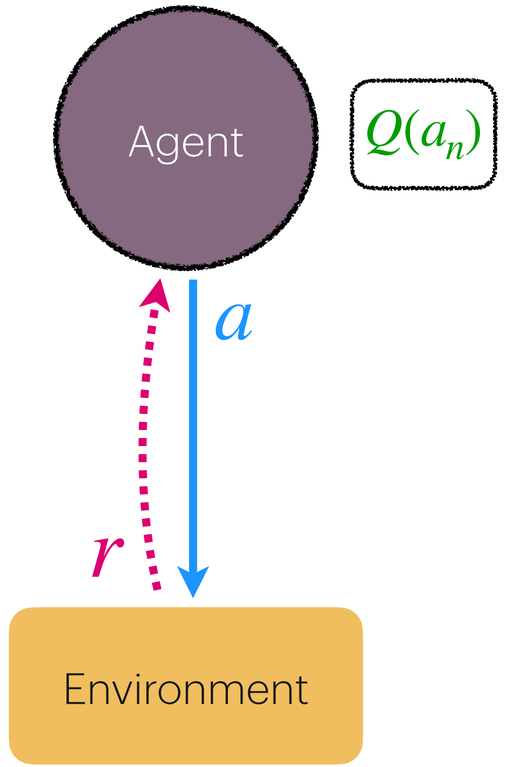
In RL settings agents can do two things to improve their reward:

**Exploit their knowledge (exploitation):**

- Select the action with the highest expected reward
- Directly accumulate reward

**Improve their knowledge (exploration):**

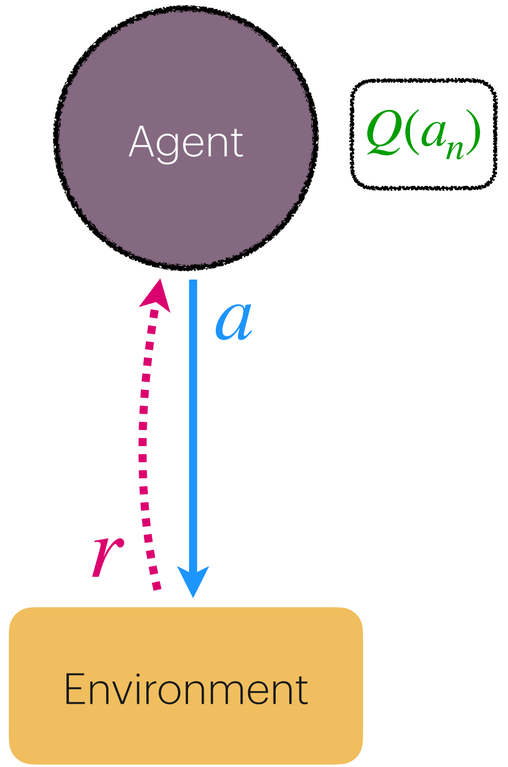
- Sample the environment to collect more knowledge
- Improve the accuracy of future exploitations



# $\epsilon$-greedy policy

What's the simplest policy that exploits knowledge without stop exploring?

$$A_{t+1} = \begin{cases} \arg\max_n Q_t(a_n), & \text{with probability } 1-\epsilon \\ a_n, \text{ with } n \sim \text{Uniform}(N), & \text{with probability } \epsilon \end{cases}$$




- This is called the $\epsilon$**- greedy** policy

Is this the optimal way to explore?

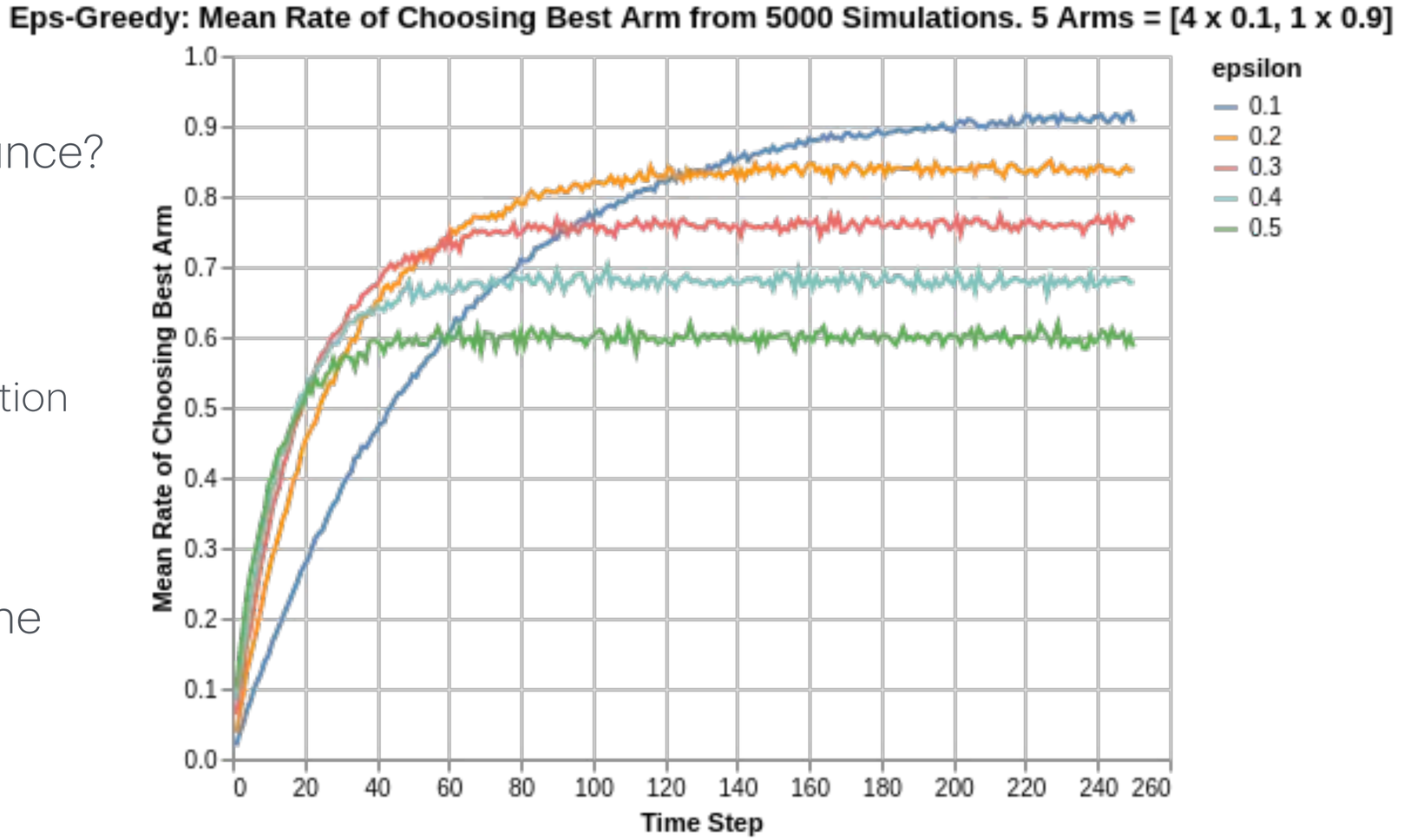
What would be the best way to select $\epsilon$?

# The benefits and costs of exploration

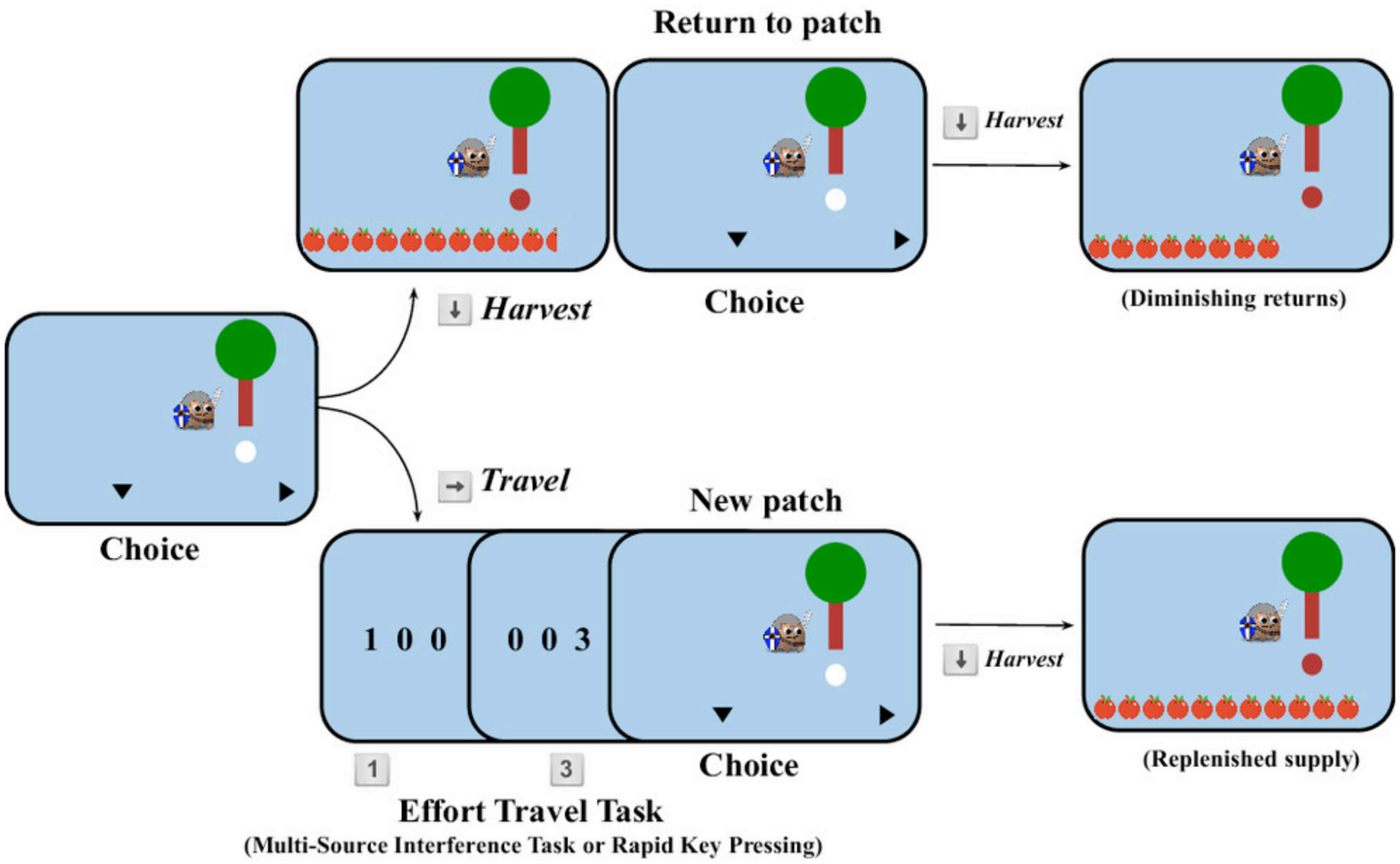
What's the effect of $\epsilon$ on performance?

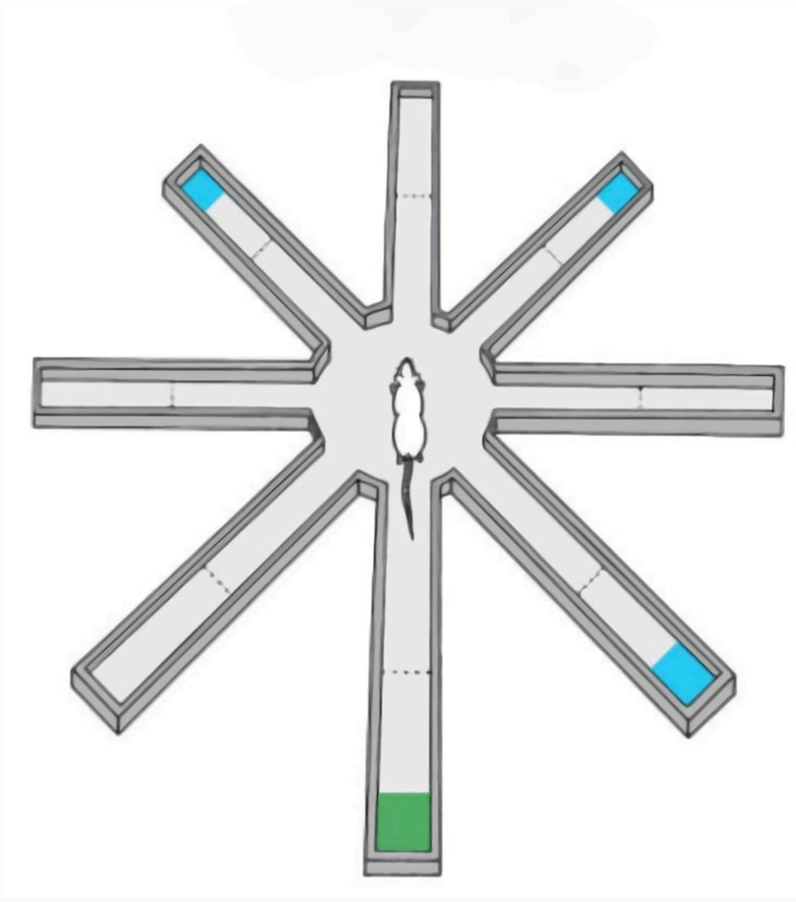
- $\epsilon \sim$ speed of learning
- $1 - \epsilon \sim$ maximum possible exploitation

Can you think of a strategy with the benefits of the two options?



# Measuring the exploration/exploitation balance



# Using exploration/exploitation to model behaviour

**Insights in every-day cognition:**

- Attention disperses during exploration and focuses during exploitation
- Individual differences in exploration vs. exploitation predict different creative domains
- Reduced spatial exploration in midlife correlates with poorer spatial memory

**Aging and development:**

- Children show broad exploration early in development, gradually shifting toward exploitation
- The elder shift to exploitation (accumulated knowledge, reduced cognitive control)

**Disorders:**

- Anxiety and mood disorders often enhance exploratory behaviours
- Depression impact decision stability (and sensitivity to reward)
- Schizophrenia, OCD, and ADHD characterised by excessive switching

(review: doi.org/10.1186/s12888-025-06837-w)

# Reinforcement Learning

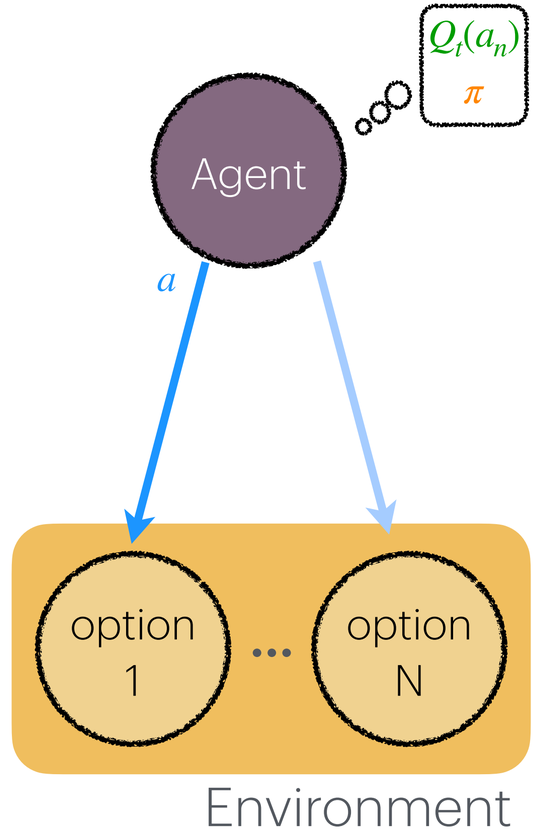
1. Two types of learning
2. Reinforcement learning
3. The $k$-armed bandit problem
4. The $\varepsilon$-greedy algorithm
5. **Markov Decision Problems**
6. Conclusions

# Limits of the $k$-armed bandit

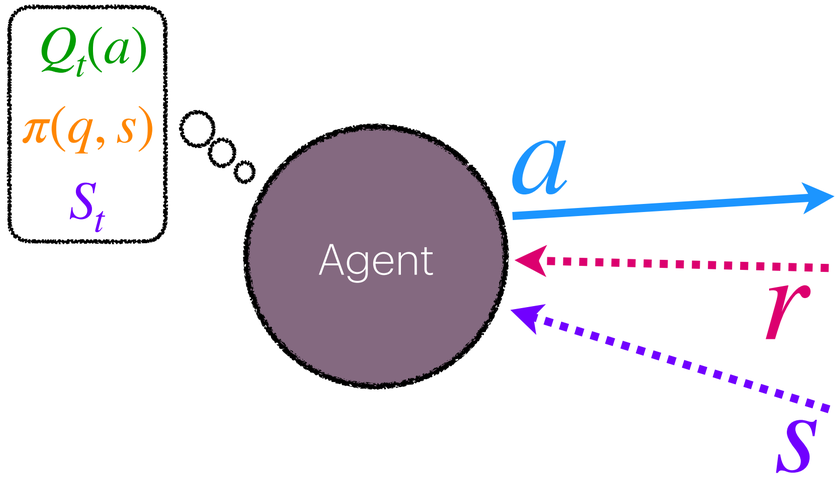
Is the $k$-armed problem a good model of the RL problem? What is missing?

- Reward distributions are fixed
  - IRL distributions usually change in time (e.g. a bar changes its cook)
  - Sofia will show us how to address this
- All actions are always accessible
  - IRL rewarding actions (eating) require previous actions (cooking)
- Possible actions are limited to the number of arms
  - IRL actions belong to a continuum (e.g. adding salt)
- Order in which actions are performed is irrelevant for reward
  - IRL reward depends on the chain of actions (e.g. sitting after a hike)
- Feedback is given immediately after performing the action
  - IRL reward may have some delay (e.g. getting food after ordering)

**$k$-armed bandit model**



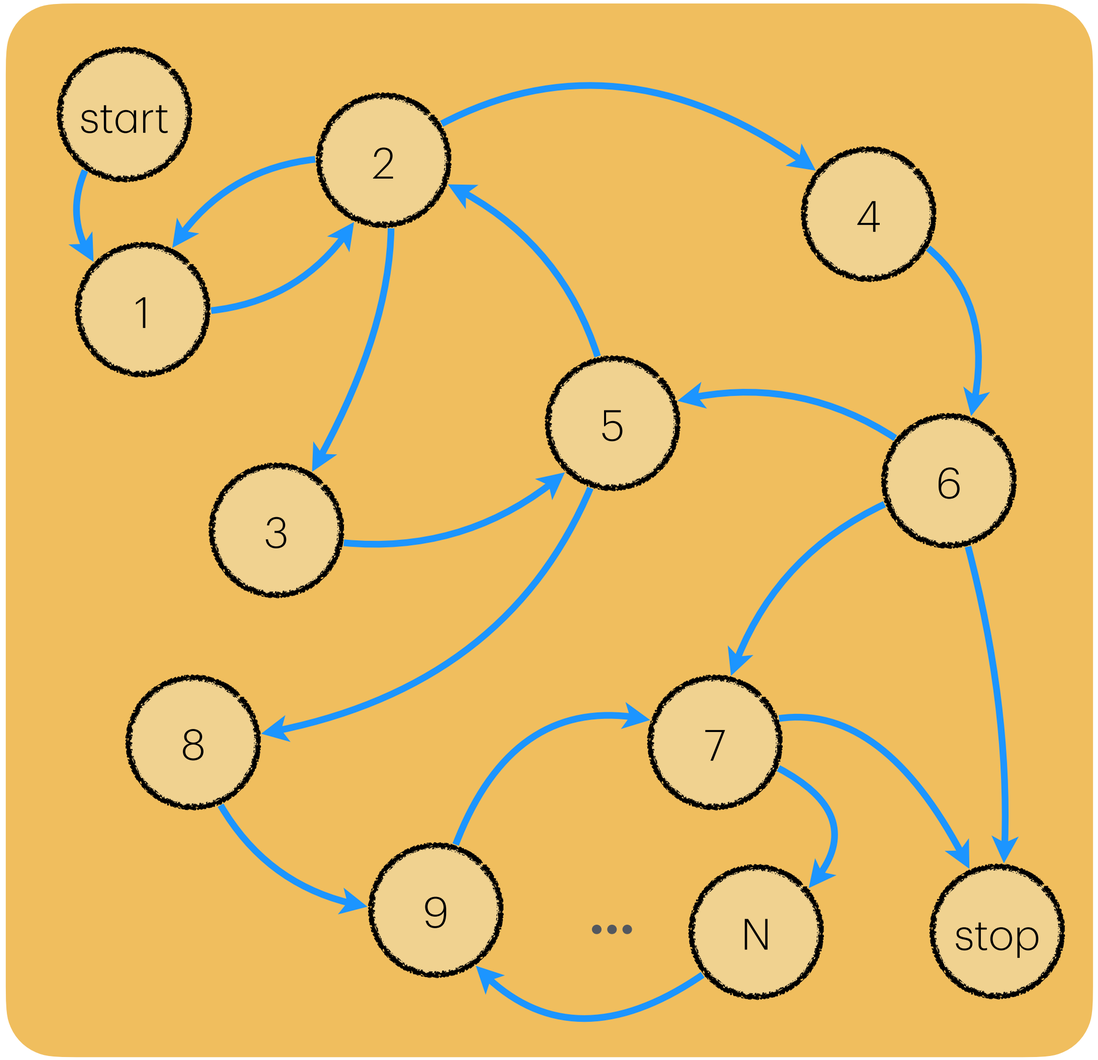
# Finite Markov Decision Processes: state dependency



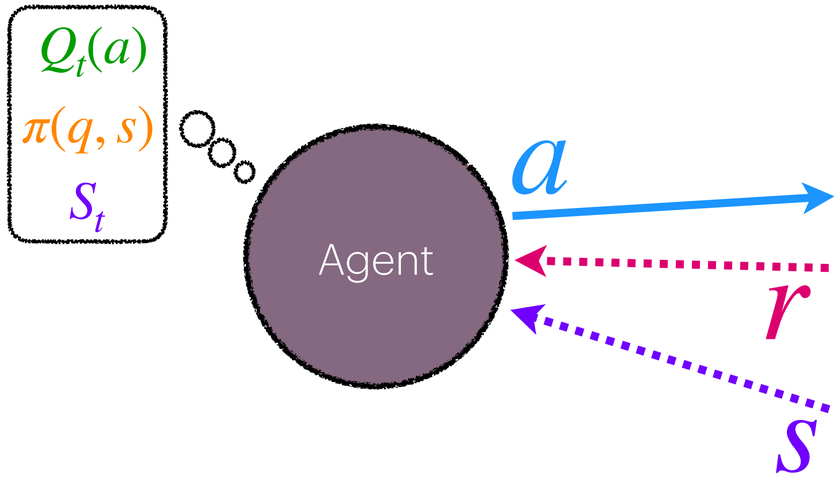

$$\pi : Q_t(a_n), S_t \rightarrow A_{t+1}$$

$$A_{t+1} = \pi\left(Q_t(a_n), S_t\right)$$

Environment



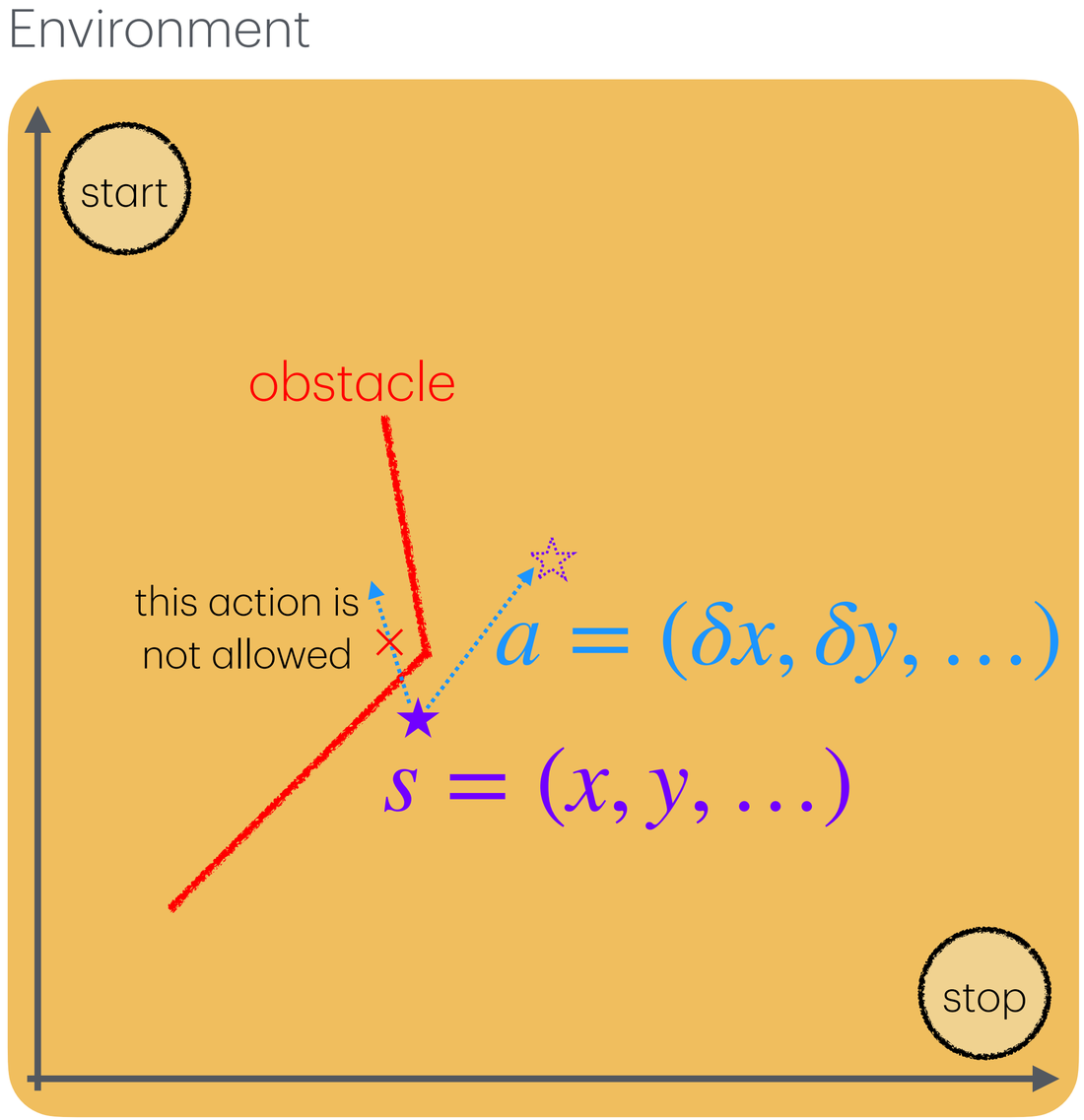
# Markov Decision Processes: continuous environment



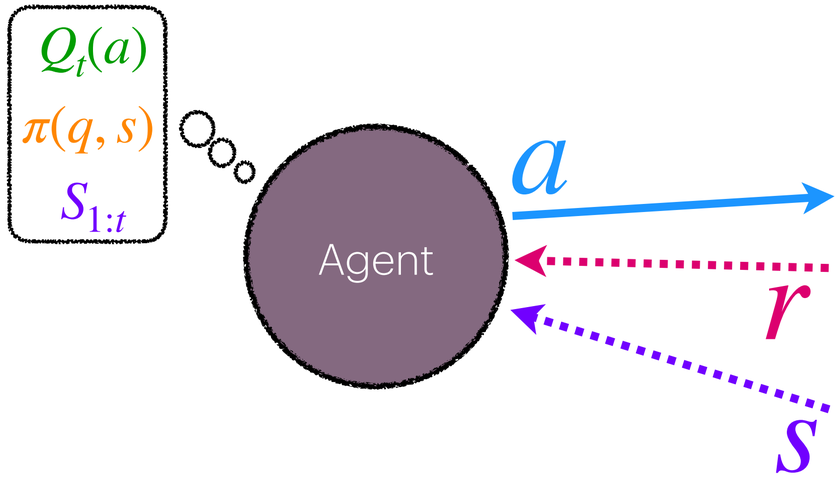

$$\pi : Q_t(a_n), S_t \rightarrow A_{t+1}$$

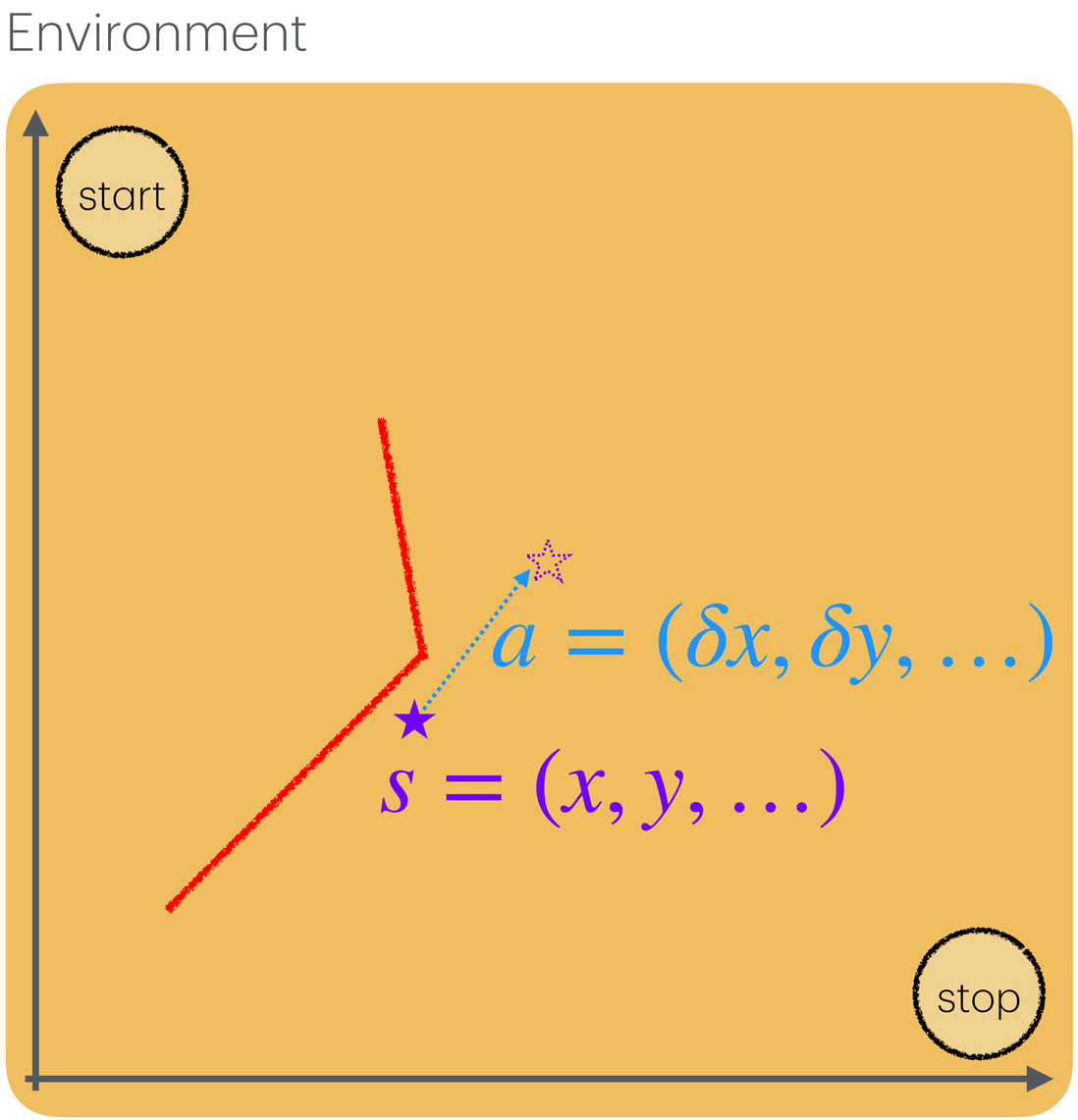
$$A_{t+1} = \pi\left(Q_t(a_n), S_t\right)$$

$$R_t = r\left(X_t, Y_t, \ldots\right) = r\left(S_t\right)$$



# Rewards depend on chain of actions ~ state trajectory




$$\pi : Q_t(a_n), S_{1:t} \rightarrow A_{t+1}$$

$$A_{t+1} = \pi\left(Q_t(a_n), S_{1:t}\right)$$

$$R_t = r\left(S_1, S_2, \ldots, S_t\right) = r\left(S_{1:t}\right)$$

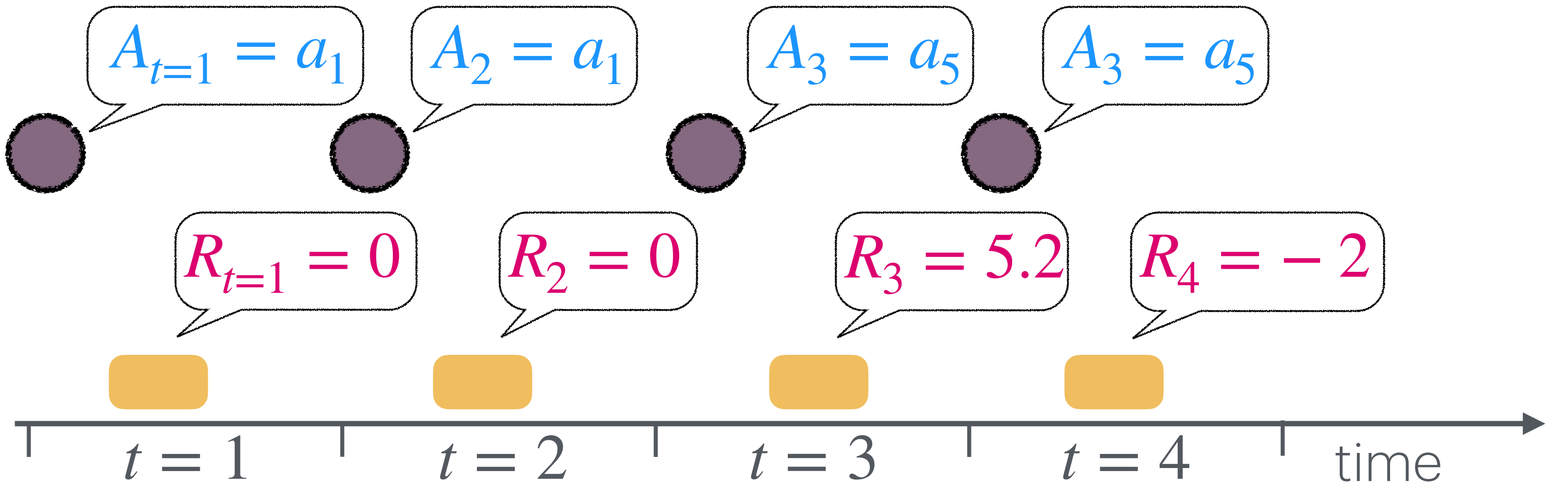
# Credit assignment and delayed rewards

**Credit assignment problem:**

- How does the agent know what action / chain of actions caused each reward?

Example:

- Food poisoning after a night full of pintxos



# Reinforcement Learning in Neuroscience

**Finite Markov Decision Processes revealed:**

- Navigation and goal-directed planning follow FMDP theoretical predictions
- FMDP explains humans thinking time for planning during complex navigation tasks

**$k$-armed bandit problem**

- A simplified model of the reinforcement learning problem
- Allows us to model reward prediction error, which is mediated by dopamine release
- Introduces the exploration/exploitation dilemma, with a myriad of applications for cognition

**The full reinforcement learning problem**

- Credit assignment and continuous states make modelling mathematically challenging
- Removing the assumptions of the $k$-armed bandit drastically expands our modelling horizons

# Reinforcement Learning

1. Two types of learning
2. Reinforcement learning
3. The $k$-armed bandit problem
4. The $\varepsilon$-greedy algorithm
5. Elements of reinforcement learning
6. **Conclusions**

# Take home messages

**Three normative modelling frameworks for learning in the brain:**

- Supervised, reinforcement, and unsupervised learning

**Continuous Markov Decision Processes useful to model:**

- Motor control and planning/execution of reaching movements
- Continuous learning trajectories in perceptual sensitivity

**Credit assignment problem:**

- Sequential navigation: evaluating intermediate states by expected future reward
- Social learning: assigning credit to others' actions during observation
- Language acquisition: determining which words/structures caused successful communication

# Questions for next session

# Questions for next session: Q5.1

In ANNs, supervised learning is implemented through backpropagation. Explain how this algorithm works. Why is this process considered biologically implausible? Can the brain implement supervised learning at all? If so, how?

# Questions for next session: Q5.2

Unsupervised learning discovers structure without guidance. What does 'discovering structure' actually mean? What time of criteria would you expect the brain to use to structure the world?

# Questions for next session: Q5.3

Dopamine is thought to convey reward prediction errors. Depression has been linked to blunted dopamine responses. Research this connection. How might the RL framework explain depressive symptoms? How or how not is this model useful?

# Questions for next session: Q5.4

RL algorithms learn to maximise reward. But humans and animals are curious: they often enjoy exploring even when it doesn't lead to immediate reward. Can the RL framework accommodate for curiosity? First think about it, and then research some computational models of curiosity. What are your conclusions?

# Questions for next session: Q5.5

Reflect on the academic and personal trajectory of yourself, someone you know well, or some fictional character. Identify specific decisions driven by exploration and others driven by exploitation. When was each of the two strategies useful? When did they backfire? Is RL a usefuk model of how these decisions were taken?

# Questions for next session: Q5.6

Consider social and political affairs in the light of the credit assignment problem. How do people typically assign credit in political ambiguous situations? Describe your conclusions and their implications.

# Questions for next session: Q5.7

The k-armed bandit is a foundational RL model under a series of assumptions that make the problem tractable. However, real brains operate in environments that do not fulfil those assumptions. If we build theories of brain function on simplified models, what might we systematically misunderstand about neural computation? Is there anything we can do about it?

# All course materials:

github.com/qtabs/compneuro4cogneuros