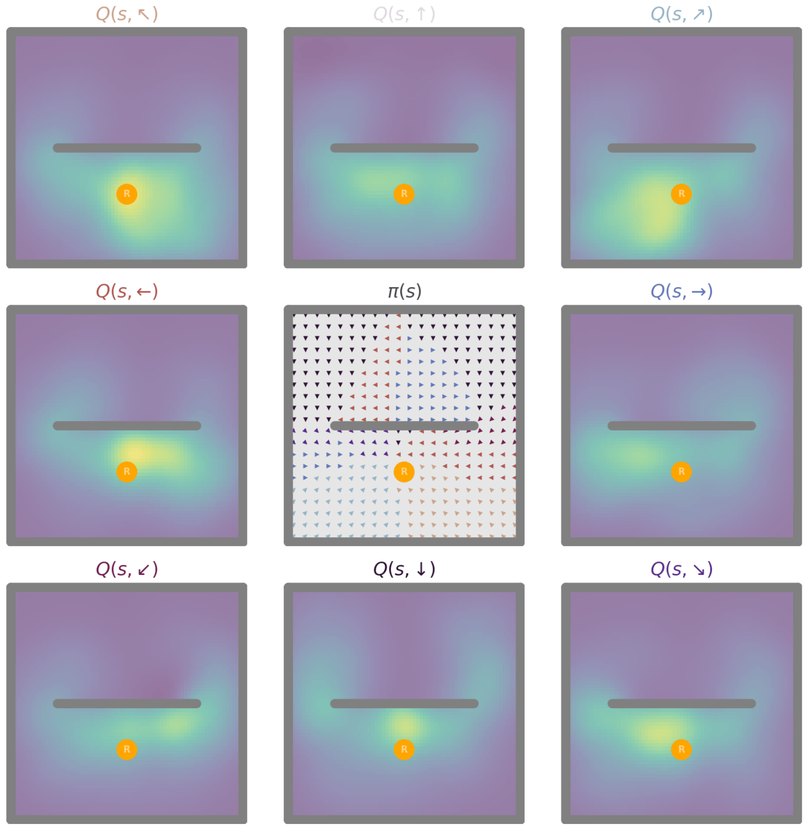
Reinforcement Learning in the brain

Jesse Geerts

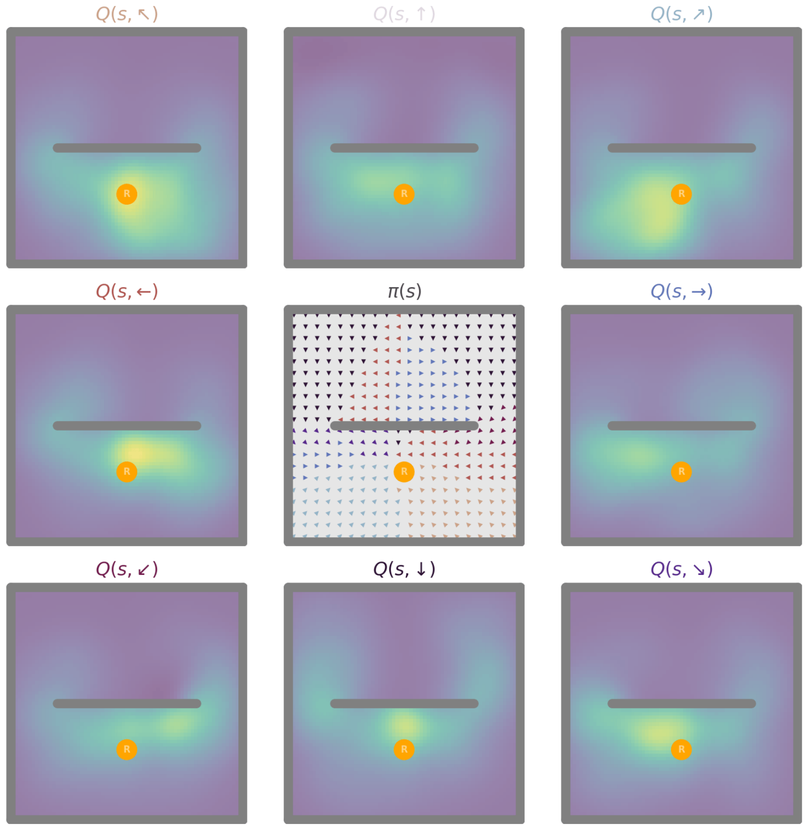
SWC Neuroinformatics 2025



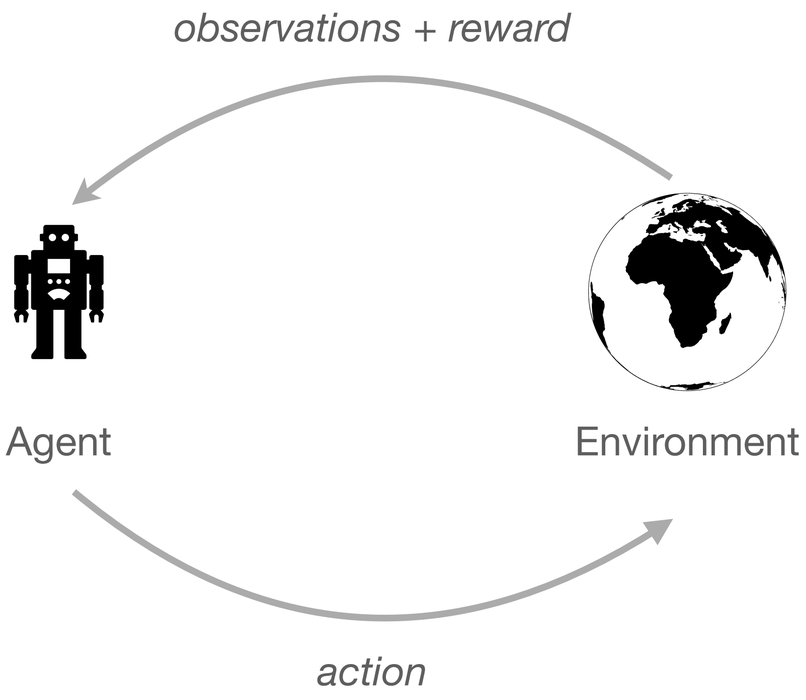
Reinforcement Learning in the brain

Jesse Geerts

SWC Neuroinformatics 2025

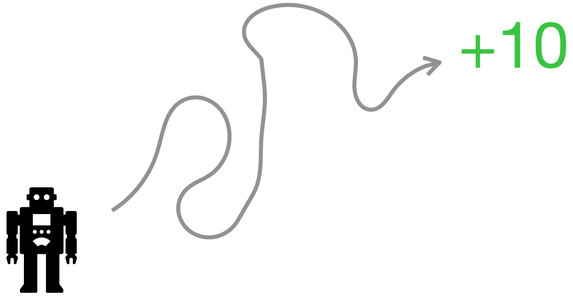


The RL problem: given a task, which actions maximize reward?



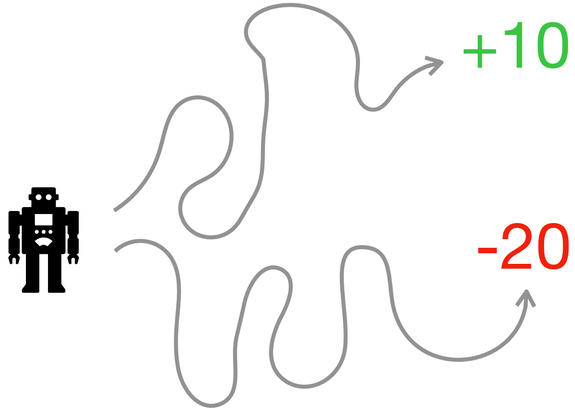
RL differs from supervised and unsupervised learning

Reinforcement learning



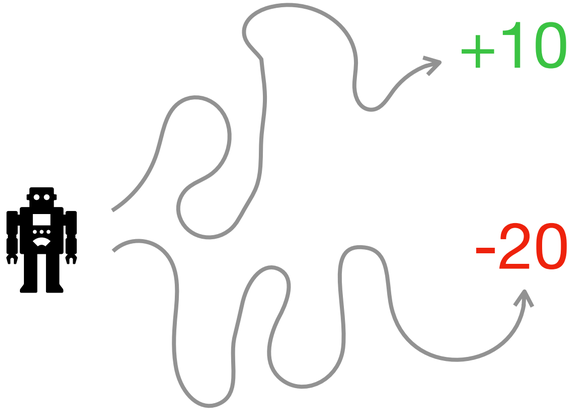
RL differs from supervised and unsupervised learning

Reinforcement learning

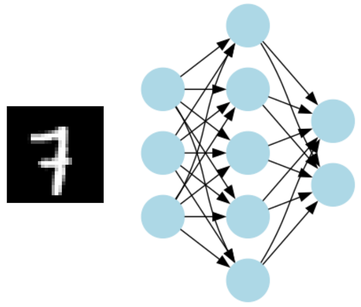


RL differs from supervised and unsupervised learning

Reinforcement learning



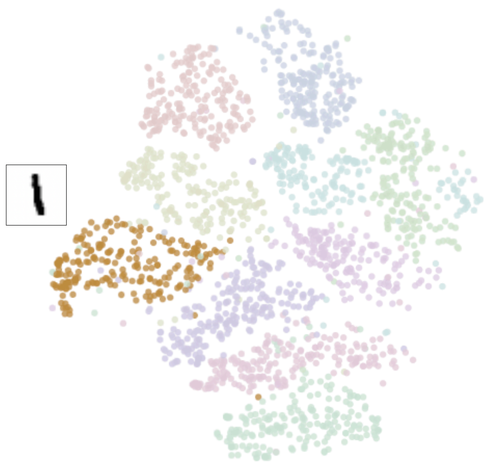
Supervised learning



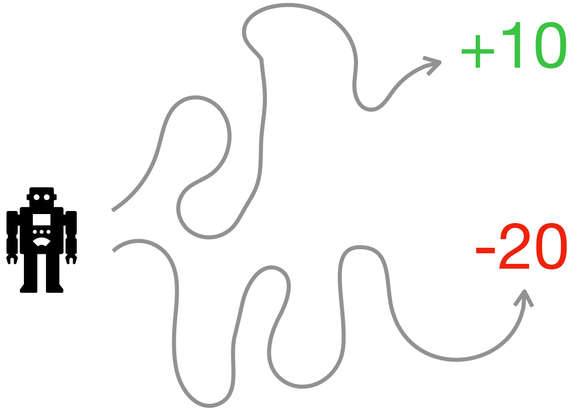
[6]

RL differs from supervised and unsupervised learning

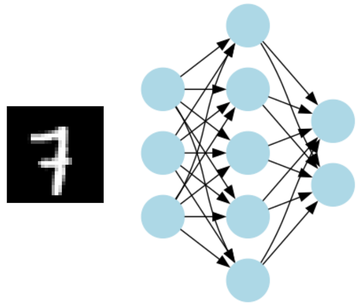
Unsupervised learning



Reinforcement learning



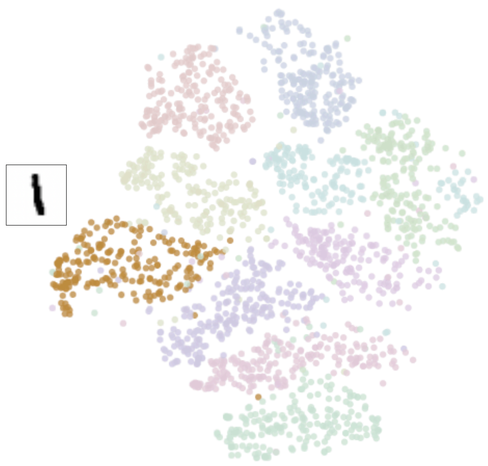
Supervised learning



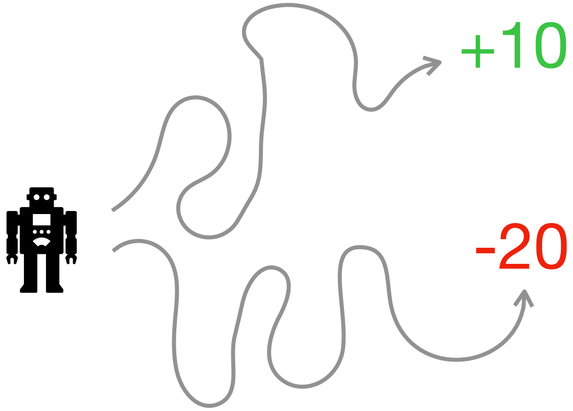
[6]

RL differs from supervised and unsupervised learning

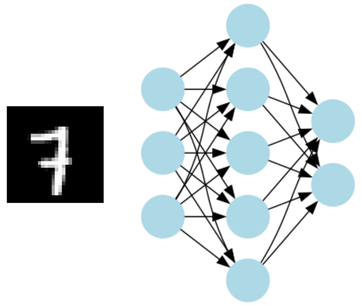
Unsupervised learning



Reinforcement learning



Supervised learning

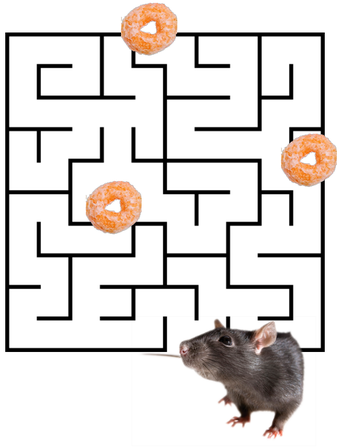


[6]

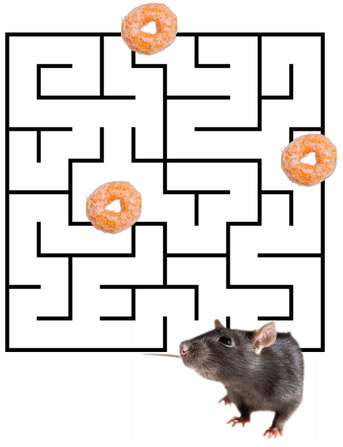
Model of autonomous learning

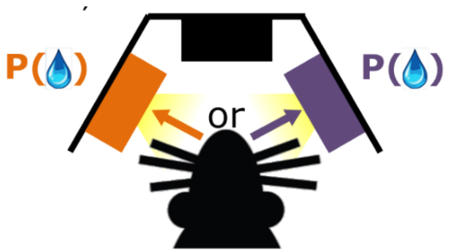
Many problems can be described as an RL problem

Many problems can be described as an RL problem

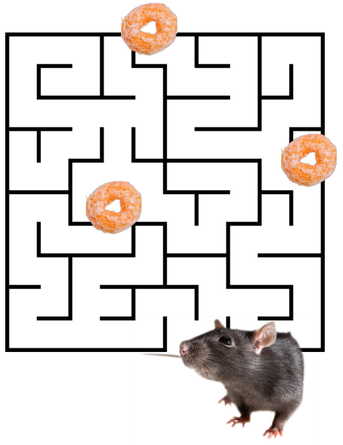


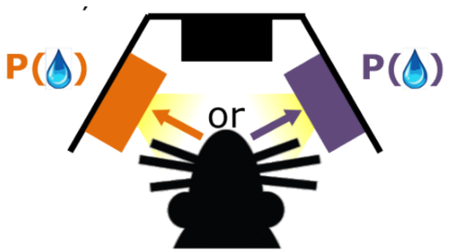
Many problems can be described as an RL problem

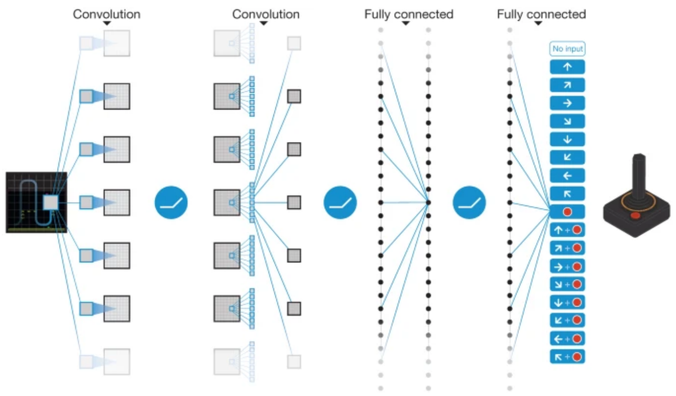




Many problems can be described as an RL problem

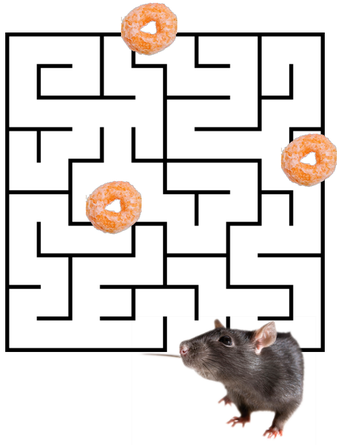


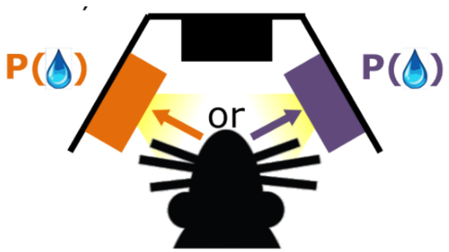


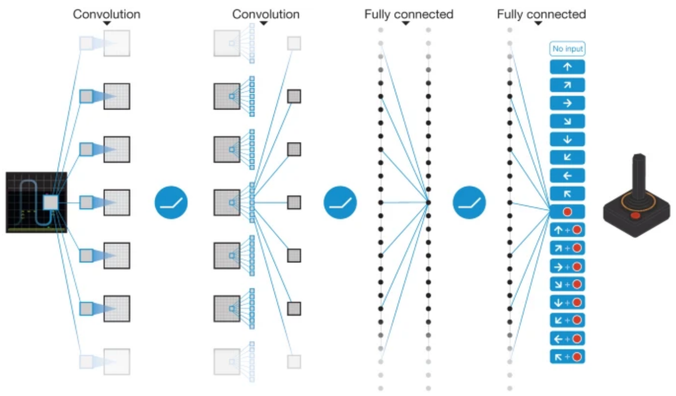


Mnih et al. (2014)

Many problems can be described as an RL problem



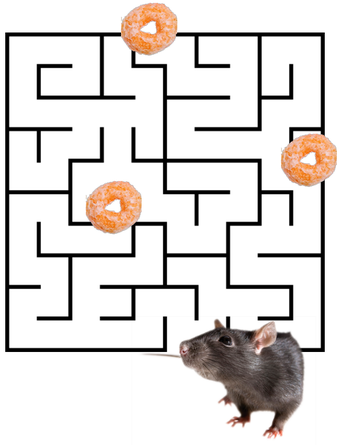


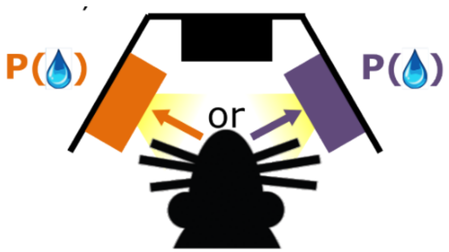


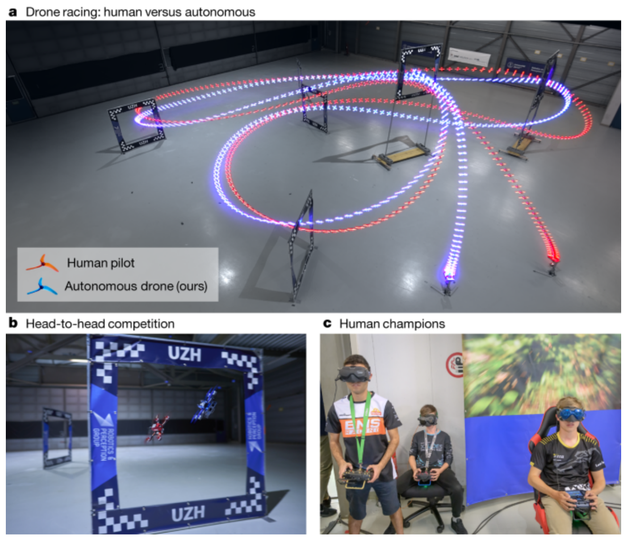
Mnih et al. (2014)

Silver et al. (2018)

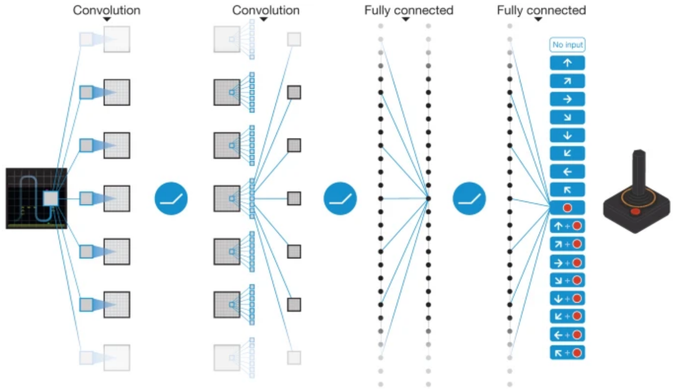
Many problems can be described as an RL problem







Kauffman et al. (2023)



Mnih et al. (2014)

Silver et al. (2018)

Plan for today

Plan for today

- **How do we predict reward?**
  - Classical conditioning and associative learning theory

Plan for today

- **How do we predict reward?**
  - Classical conditioning and associative learning theory
- **How do we predict *future* reward?**
  - Temporal difference learning and dopamine

Plan for today

- **How do we predict reward?**
  - Classical conditioning and associative learning theory
- **How do we predict *future* reward?**
  - Temporal difference learning and dopamine
- **How do we optimise actions to maximise future reward?**
  - Policy iteration, SARSA

Plan for today

- **How do we predict reward?**
  - Classical conditioning and associative learning theory
- **How do we predict *future* reward?**
  - Temporal difference learning and dopamine
- **How do we optimise actions to maximise future reward?**
  - Policy iteration, SARSA
- **What to do in large or continuous environments?**
  - Function approximation and representations of place

Plan for today

- **How do we predict reward?**
  - Classical conditioning and associative learning theory
- **How do we predict *future* reward?**
  - Temporal difference learning and dopamine
- **How do we optimise actions to maximise future reward?**
  - Policy iteration, SARSA
- **What to do in large or continuous environments?**
  - Function approximation and representations of place
- **Representation learning for RL**
  - Learning a predictive representation to support more flexible RL

Plan for today

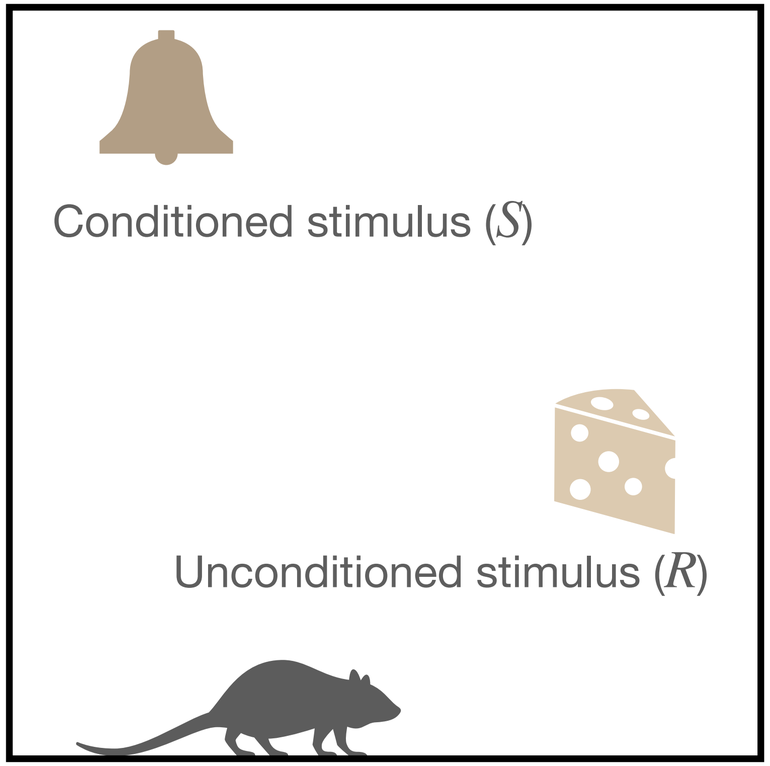
- **How do we predict reward?**
  - Classical conditioning and associative learning theory
- **How do we predict *future* reward?**
  - Temporal difference learning and dopamine
- **How do we optimise actions to maximise future reward?**
  - Policy iteration, SARSA
- **What to do in large or continuous environments?**
  - Function approximation and representations of place
- **Representation learning for RL**
  - Learning a predictive representation to support more flexible RL
- **The limits of model-free RL and the case for a cognitive map**

Plan for today

- **How do we predict reward?**
  - Classical conditioning and associative learning theory
- **How do we predict *future* reward?**
  - Temporal difference learning and dopamine
- **How do we optimise actions to maximise future reward?**
  - Policy iteration, SARSA
- **What to do in large or continuous environments?**
  - Function approximation and representations of place
- **Representation learning for RL**
  - Learning a predictive representation to support more flexible RL
- **The limits of model-free RL and the case for a cognitive map**
- Friday: practical with code

Classical conditioning and the Rescorla-Wagner model

Pavlovian conditioning



Learning to predict reward using prediction errors

Rescorla & Wagner (1972)

S🔔 $\rightarrow$ R🧀

Learning to predict reward using prediction errors

Rescorla & Wagner (1972)

S🔔 $\rightarrow$ R🧀

$$V = \mathbb{E}[R]$$

value = *expected* reward

Learning to predict reward using prediction errors

Rescorla & Wagner (1972)

S🔔 $\rightarrow$ R🧀

$$V = \mathbb{E}[R]$$

value = *expected* reward

$$\hat{V} \leftarrow \hat{V} + \alpha \cdot (R - \hat{V})$$

Learning to predict reward using prediction errors

Rescorla & Wagner (1972)

S🔔 $\rightarrow$ R🧀

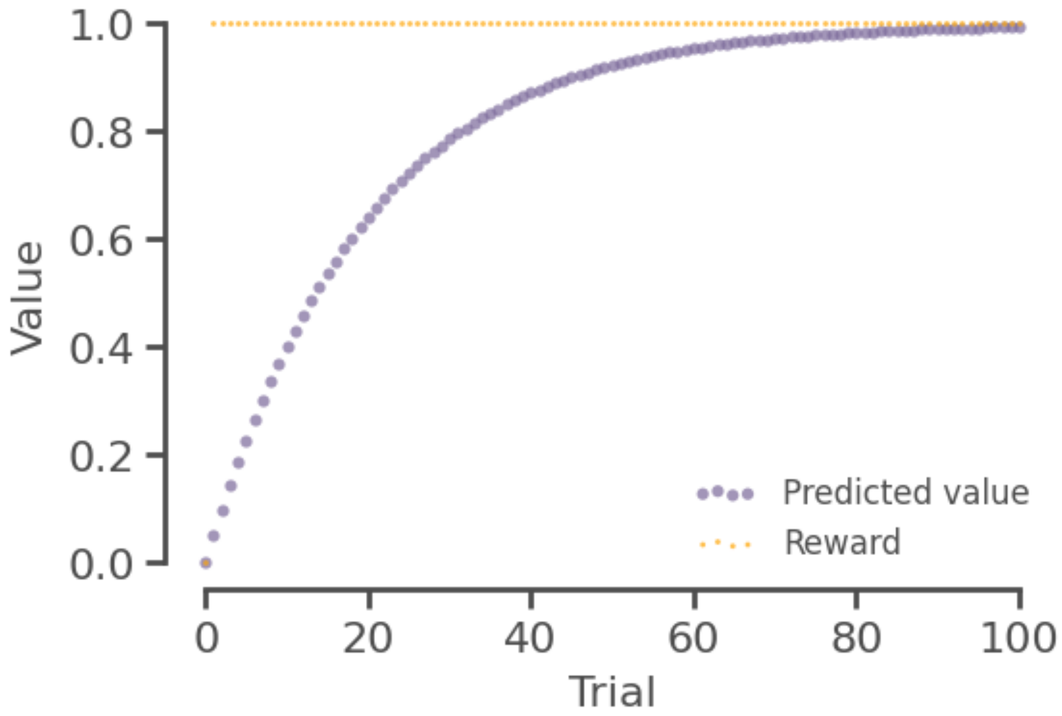
$$V = \mathbb{E}[R]$$

value = *expected* reward

$$\hat{V} \leftarrow \hat{V} + \alpha \cdot (R - \hat{V})$$

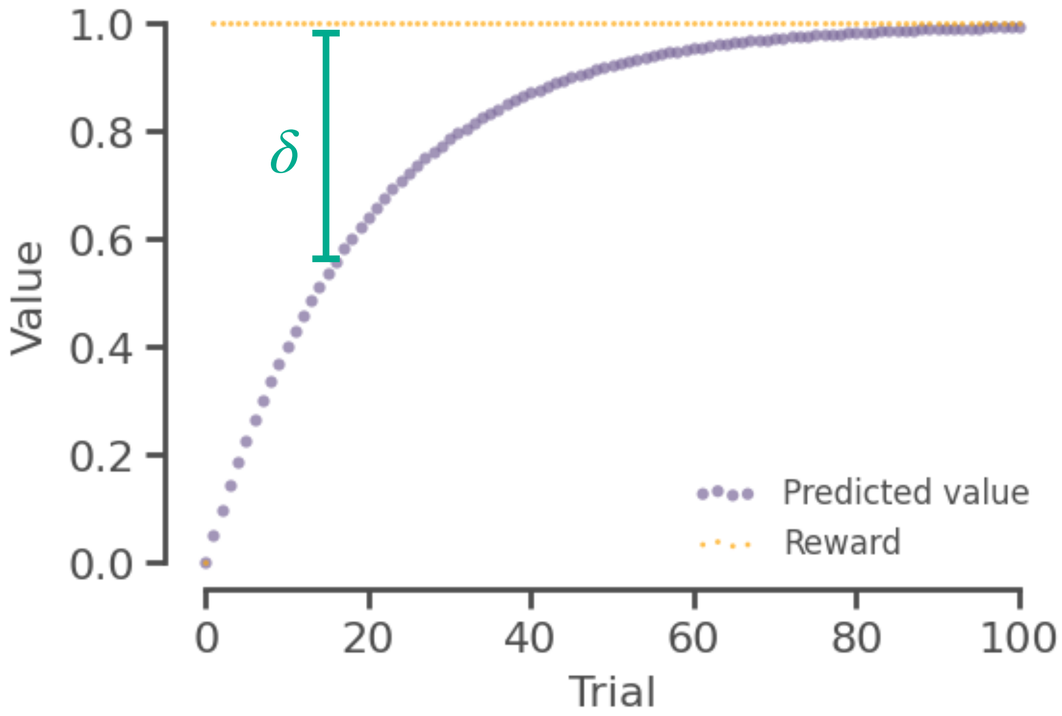
reward prediction error δ

Learning in the RW model



$$\hat{V} \leftarrow \hat{V} + \alpha \cdot (R - \hat{V})$$

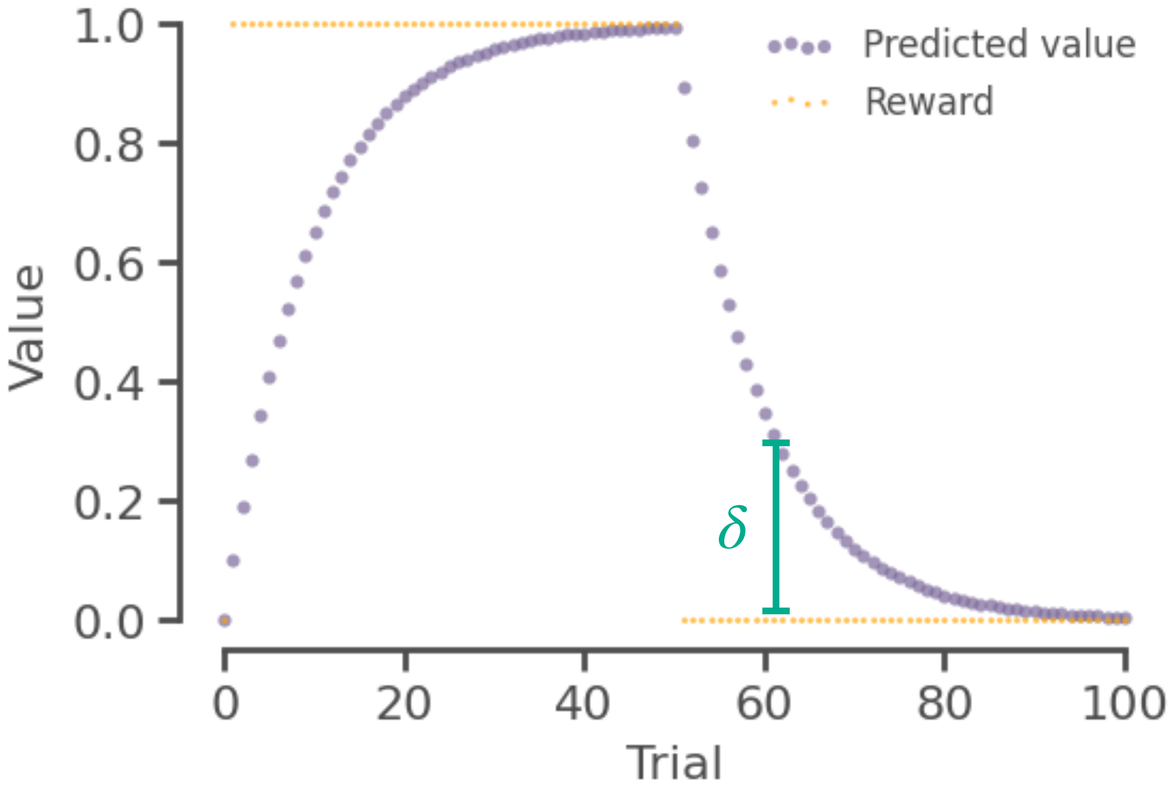
Learning in the RW model



$$\hat{V} \leftarrow \hat{V} + \alpha \cdot (R - \hat{V})$$

reward prediction error δ

Extinction in the RW model



Example “blocking” experiment with multiple stimuli

	Cues	Reward	Expectation
Phase 1	S_A	R $\times 1$	High

Kamin (1969)

Example “blocking” experiment with multiple stimuli

	Cues	Reward	Expectation
Phase 1	S_A🔔	R🧀 $\times 1$	High
Phase 2	S_A🔔 $+ S_B$💡	R🧀 $\times 1$	High

Kamin (1969)

Example “blocking” experiment with multiple stimuli

	Cues	Reward	Expectation
Phase 1	S_A🔔	R🧀 $\times 1$	High
Phase 2	S_A🔔 $+ S_B$💡	R🧀 $\times 1$	High
Test	S_B💡 ?		Low

Kamin (1969)

Rescorla-Wagner with multiple stimuli

$$\mathbf{S} = \left[S_A🔔, S_B💡\right]$$

$$\mathbf{w} = \left[w_A, w_B\right]$$

Rescorla & Wagner (1972)

Rescorla-Wagner with multiple stimuli

$$\mathbf{S} = \left[S_A🔔, S_B💡\right]$$

$$\mathbf{w} = \left[w_A, w_B\right]$$

$$\hat{V}(\mathbf{S}) = \sum_i S_i \cdot w_i$$

value estimate = weighted sum of stimuli

Rescorla & Wagner (1972)

Rescorla-Wagner with multiple stimuli

$$\mathbf{S} = \left[S_A🔔, S_B💡\right]$$

$$\mathbf{w} = \left[w_A, w_B\right]$$

$$\hat{V}(\mathbf{S}) = \sum_i S_i \cdot w_i$$

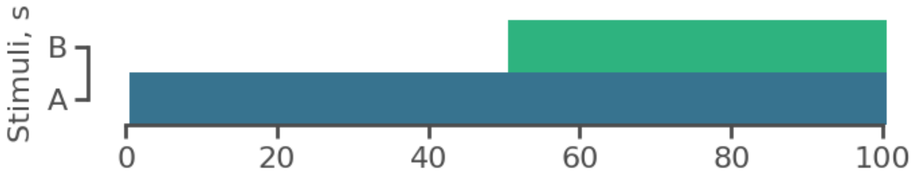
value estimate = weighted sum of stimuli

$$\mathbf{w} \leftarrow \mathbf{w} + \alpha \cdot \boxed{(R - \hat{V})} \cdot \mathbf{S}$$

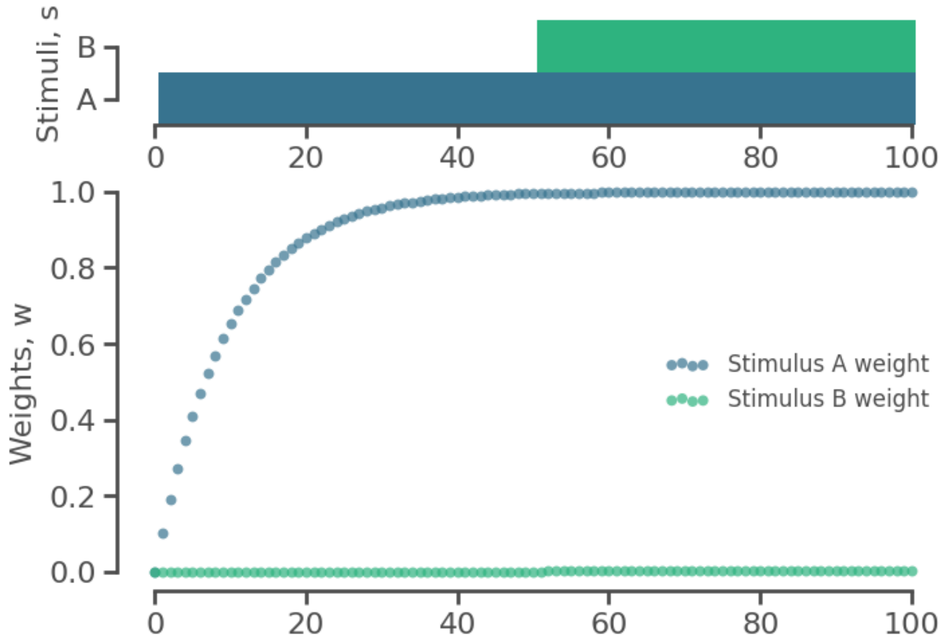
reward prediction error δ

Rescorla & Wagner (1972)

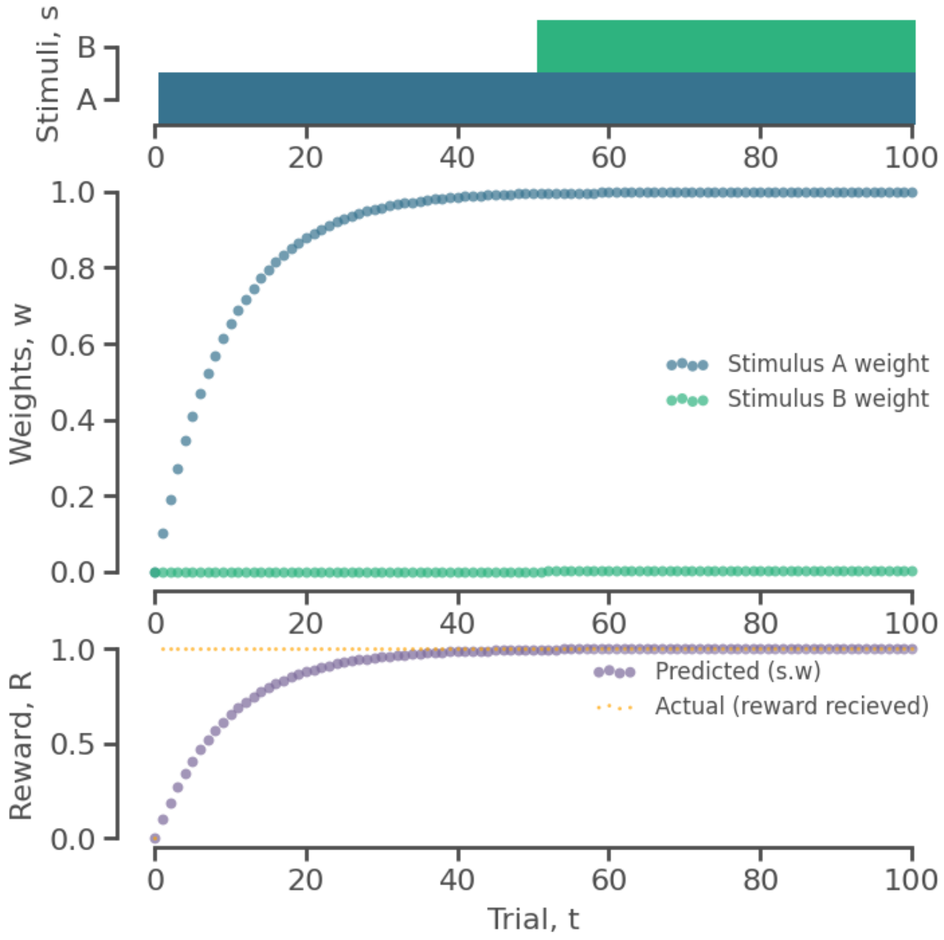
Blocking results in reduced learning of the second association



Blocking results in reduced learning of the second association



Blocking results in reduced learning of the second association

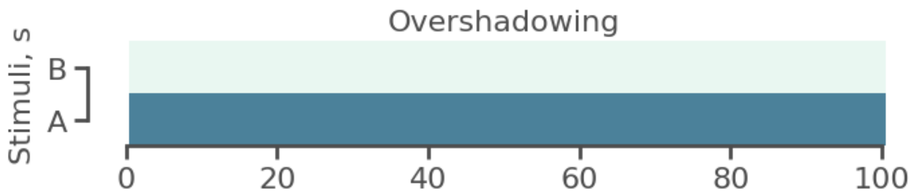


Multi-stimulus experiment 2

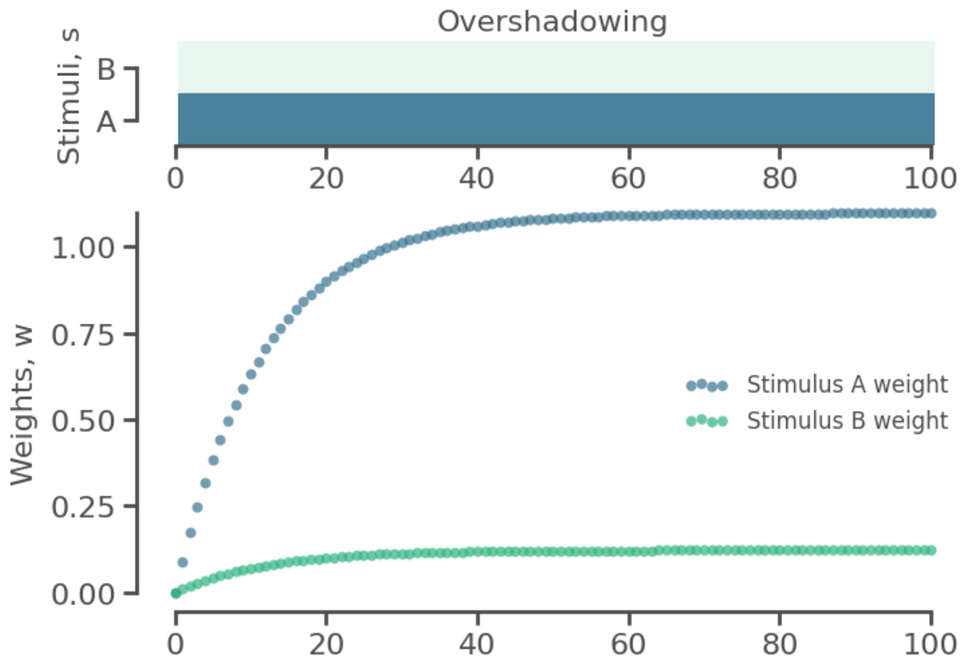
	Cues	Reward	Expectation
Phase 1	[🔔, 💡] = [20,1]	R🧀 × 1	High
Test	[~~🔔~~, 💡] = [0,1]		?

Kamin (1969)

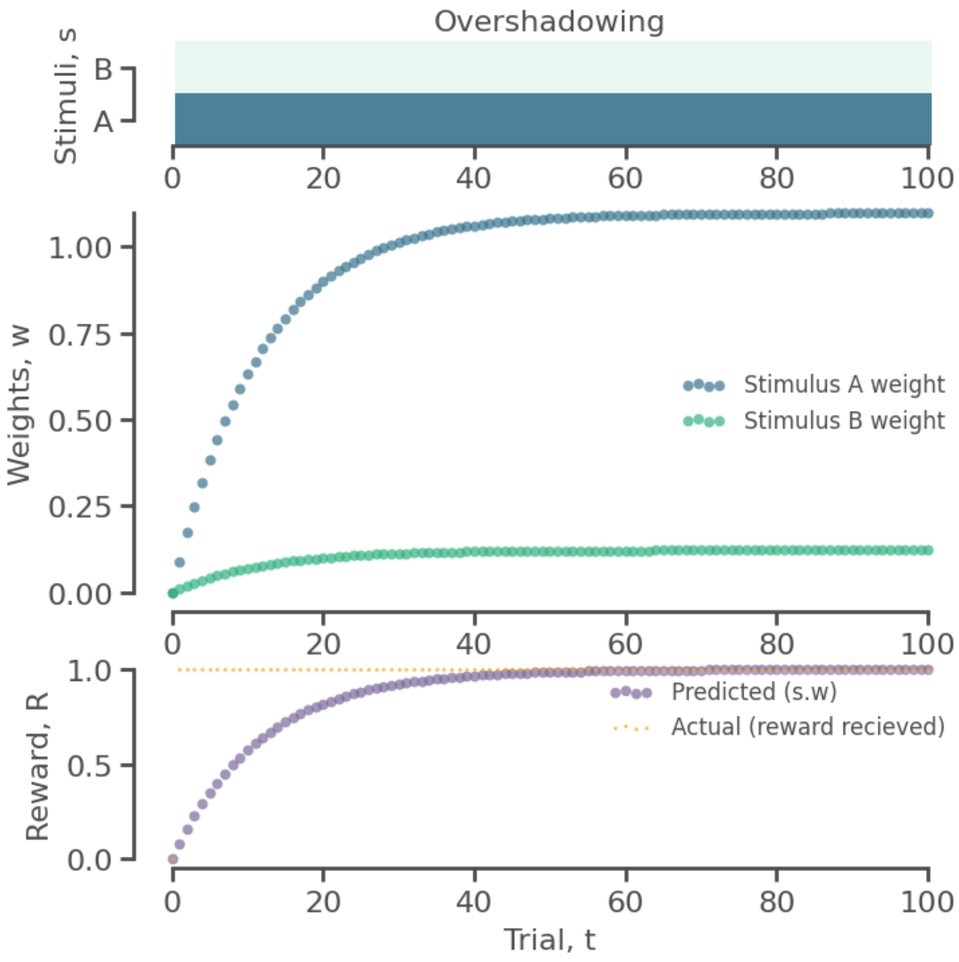
Salient features overshadow less salient features



Salient features overshadow less salient features



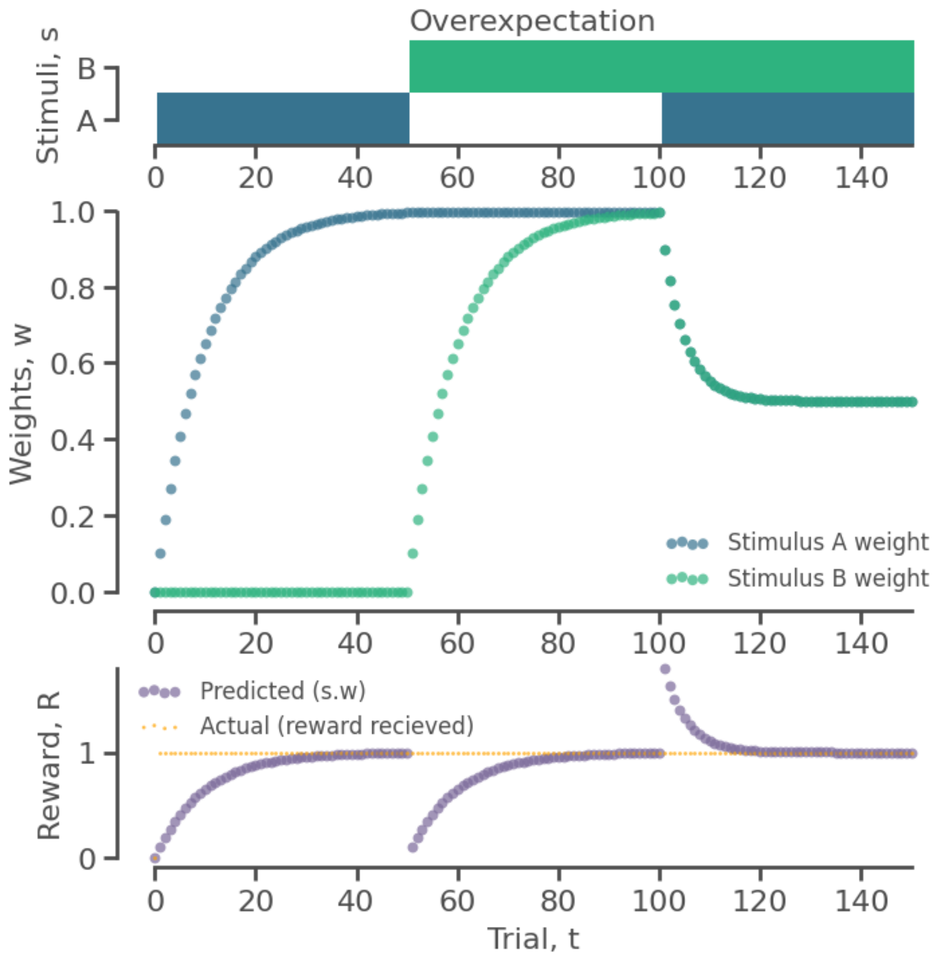
Salient features overshadow less salient features



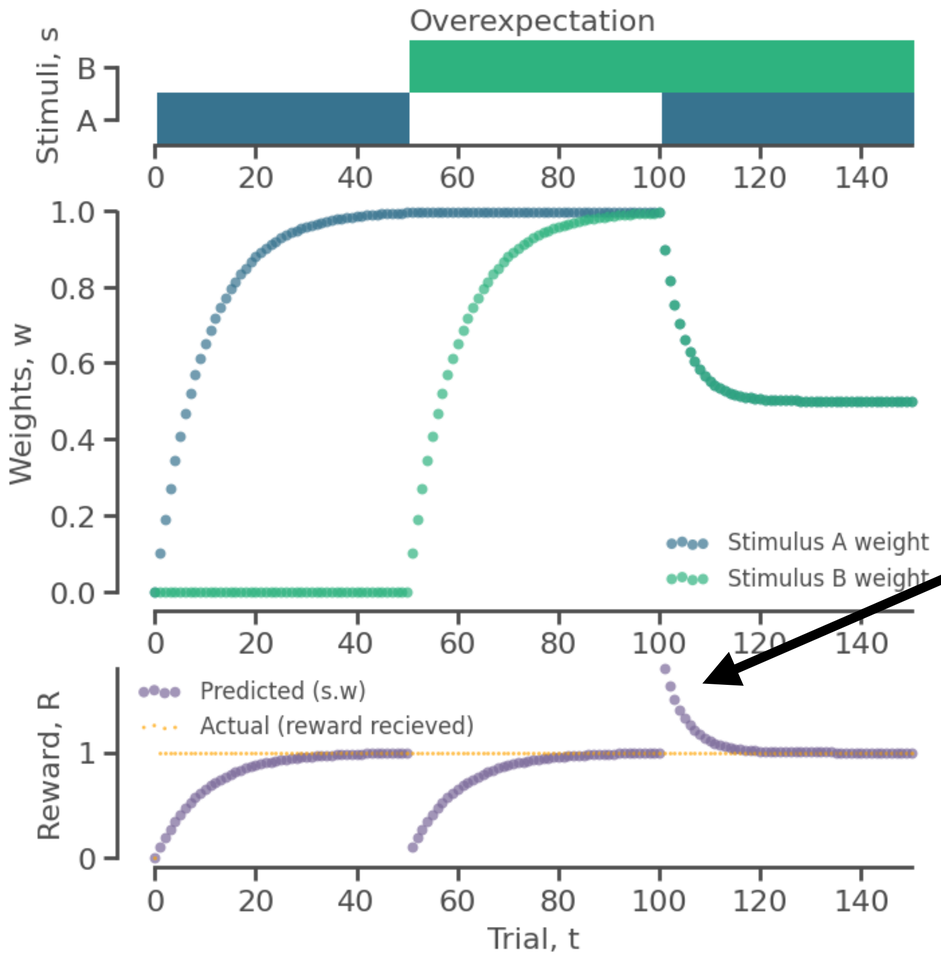
Multi-stimulus experiment 3

	Cues	Reward	Expectation
Phase 1	S_A🔔	R🧀 $\times 1$	High
Phase 2	S_B💡	R🧀 $\times 1$	High
Test	S_A🔔 $+ S_B$💡		?

Summation can lead to over-expectation



Summation can lead to over-expectation



over-expectation

Plan for today

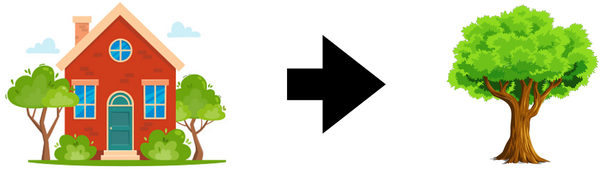
- **How do we predict reward?**
  - Classical conditioning and associative learning theory
- **How do we predict *future* reward?**
  - Temporal difference learning and dopamine
- **How do we optimise actions to maximise future reward?**
  - Policy iteration, SARSA
- **What to do in large or continuous environments?**
  - Function approximation and representations of place
- **Representation learning for RL**
  - Learning a predictive representation to support more flexible RL
- **The limits of model-free RL and the case for a cognitive map**
- Afternoon: TUTORIAL

Temporal difference learning for predicting *future* rewards

S $\rightarrow R$

Predicting future rewards in RL using *value*

Predicting future rewards in RL using *value*



Predicting future rewards in RL using *value*

Predicting future rewards in RL using *value*

Predicting future rewards in RL using *value*

Predicting future rewards in RL using *value*

$$S_0 \longrightarrow S_1 \longrightarrow S_2 \longrightarrow S_3 \longrightarrow S_4$$

Predicting future rewards in RL using *value*

$$S_0 \xrightarrow[R_1 = 0]{} S_1 \xrightarrow[R_2 = 0]{} S_2 \xrightarrow[R_3 = 0]{} S_3 \xrightarrow[R_4 = 1]{} S_4$$

Predicting future rewards in RL using *value*

$$S_0 \xrightarrow[R_1 = 0]{} S_1 \xrightarrow[R_2 = 0]{} S_2 \xrightarrow[R_3 = 0]{} S_3 \xrightarrow[R_4 = 1]{} S_4$$

$$V(S_t) = \mathbb{E}\left[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots\right]$$

value = discounted sum of *future* rewards

How can we learn the value function?

Offline method

$$V(S_t) = \mathbb{E}\left[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots\right] = \mathbb{E}\left[G_t\right]$$

value = discounted sum of *future* rewards

$$S_0 \xrightarrow[R_1 = 0]{} S_1 \xrightarrow[R_2 = 0]{} S_2 \xrightarrow[R_3 = 0]{} S_3 \xrightarrow[R_4 = 1]{} S_4$$

How can we learn the value function?

Offline method

$$V(S_t) = \mathbb{E}\left[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots\right] = \mathbb{E}\left[G_t\right]$$

value = discounted sum of *future* rewards

$$S_0 \xrightarrow{R_1 = 0} S_1 \xrightarrow{R_2 = 0} S_2 \xrightarrow{R_3 = 0} S_3 \xrightarrow{R_4 = 1} S_4$$

$$G_3 = 1$$

How can we learn the value function?

Offline method

$$V(S_t) = \mathbb{E}\left[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots\right] = \mathbb{E}\left[G_t\right]$$

value = discounted sum of *future* rewards

$$S_0 \xrightarrow{R_1 = 0} S_1 \xrightarrow{R_2 = 0} S_2 \xrightarrow{R_3 = 0} S_3 \xrightarrow{R_4 = 1} S_4$$

$G_2 = \gamma \cdot 1$ $\quad$ $G_3 = 1$

How can we learn the value function?

Offline method

$$V(S_t) = \mathbb{E}\left[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots\right] = \mathbb{E}\left[G_t\right]$$

value = discounted sum of *future* rewards

$$S_0 \xrightarrow{R_1 = 0} S_1 \xrightarrow{R_2 = 0} S_2 \xrightarrow{R_3 = 0} S_3 \xrightarrow{R_4 = 1} S_4$$

$G_1 = \gamma^2 \cdot 1$ $\quad$ $G_2 = \gamma \cdot 1$ $\quad$ $G_3 = 1$

How can we learn the value function?

Offline method

$$V(S_t) = \mathbb{E}\left[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots\right] = \mathbb{E}\left[G_t\right]$$

value = discounted sum of *future* rewards

$$S_0 \xrightarrow{R_1 = 0} S_1 \xrightarrow{R_2 = 0} S_2 \xrightarrow{R_3 = 0} S_3 \xrightarrow{R_4 = 1} S_4$$

$G_0 = \gamma^3 \cdot 1$ $\quad$ $G_1 = \gamma^2 \cdot 1$ $\quad$ $G_2 = \gamma \cdot 1$ $\quad$ $G_3 = 1$

How can we learn the value function *online*?

$$S_0 \xrightarrow[R_1 = 0]{} S_1 \xrightarrow[R_2 = 0]{} S_2 \xrightarrow[R_3 = 0]{} S_3 \xrightarrow[R_4 = 1]{} S_4$$

How can we learn the value function *online*?

$$S_0 \xrightarrow[R_1 = 0]{} S_1 \xrightarrow[R_2 = 0]{} S_2 \xrightarrow[R_3 = 0]{} S_3 \xrightarrow[R_4 = 1]{} S_4$$

$$V(S_t) = \mathbb{E}\left[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots\right]$$

How can we learn the value function *online*?

$$S_0 \xrightarrow[R_1 = 0]{} S_1 \xrightarrow[R_2 = 0]{} S_2 \xrightarrow[R_3 = 0]{} S_3 \xrightarrow[R_4 = 1]{} S_4$$

$$V(S_t) = \mathbb{E}\left[R_{t+1} + \gamma V(S_{t+1})\right]$$

How can we learn the value function *online*?

$$S_0 \xrightarrow[R_1 = 0]{} S_1 \xrightarrow[R_2 = 0]{} S_2 \xrightarrow[R_3 = 0]{} S_3 \xrightarrow[R_4 = 1]{} S_4$$

$$V(S_t) = \mathbb{E}\left[R_{t+1} + \gamma V(S_{t+1})\right]$$

value of current state = (expect.) instantaneous reward + value of next state

How can we learn the value function *online*?

$$S_0 \xrightarrow[R_1 = 0]{} S_1 \xrightarrow[R_2 = 0]{} S_2 \xrightarrow[R_3 = 0]{} S_3 \xrightarrow[R_4 = 1]{} S_4$$

How can we learn the value function *online*?

$$S_0 \xrightarrow[R_1 = 0]{} S_1 \xrightarrow[R_2 = 0]{} S_2 \xrightarrow[R_3 = 0]{} S_3 \xrightarrow[R_4 = 1]{} S_4$$

$$V(S_t) = \mathbb{E}[R_{t+1} + \gamma V(S_{t+1})] \approx \mathbb{E}[R_{t+1} + \gamma \hat{V}(S_{t+1})]$$

How can we learn the value function *online*?

$$S_0 \xrightarrow[R_1 = 0]{} S_1 \xrightarrow[R_2 = 0]{} S_2 \xrightarrow[R_3 = 0]{} S_3 \xrightarrow[R_4 = 1]{} S_4$$

$$\hat{V}(S_t) \leftarrow \hat{V}(S_t) + \alpha \left[\underbrace{R_{t+1} + \gamma \hat{V}(S_{t+1}) - \hat{V}(S_t)}_{\text{TD error } \delta_t} \right]$$

How can we learn the value function *online*?

TD error $\delta_t = R_{t+1} + \gamma \hat{V}(S_{t+1}) - \hat{V}(S_t)$

$$S_0 \xrightarrow[R_1 = 0]{} S_1 \xrightarrow[R_2 = 0]{} S_2 \xrightarrow[R_3 = 0]{} S_3 \xrightarrow[R_4 = 1]{} S_4$$

How can we learn the value function *online*?

TD error $\delta_t = R_{t+1} + \gamma \hat{V}(S_{t+1}) - \hat{V}(S_t)$

$$S_0 \xrightarrow[R_1 = 0]{} S_1 \xrightarrow[R_2 = 0]{} S_2 \xrightarrow[R_3 = 0]{} S_3 \xrightarrow[R_4 = 1]{} S_4$$

Episode 1 $\quad \delta_0 = 0 \quad \delta_1 = 0 \quad \delta_2 = 0 \quad \delta_3 = 1$

How can we learn the value function *online*?

TD error $\delta_t = R_{t+1} + \gamma \hat{V}(S_{t+1}) - \hat{V}(S_t)$

$$S_0 \xrightarrow[R_1 = 0]{} S_1 \xrightarrow[R_2 = 0]{} S_2 \xrightarrow[R_3 = 0]{} S_3 \xrightarrow[R_4 = 1]{} S_4$$

Episode 1	$\delta_0 = 0$	$\delta_1 = 0$	$\delta_2 = 0$	$\delta_3 = 1$
Episode 2	$\delta_0 = 0$	$\delta_1 = 0$	$\delta_2 > 0$	$\delta_3 > 0$

How can we learn the value function *online*?

TD error $\delta_t = R_{t+1} + \gamma \hat{V}(S_{t+1}) - \hat{V}(S_t)$

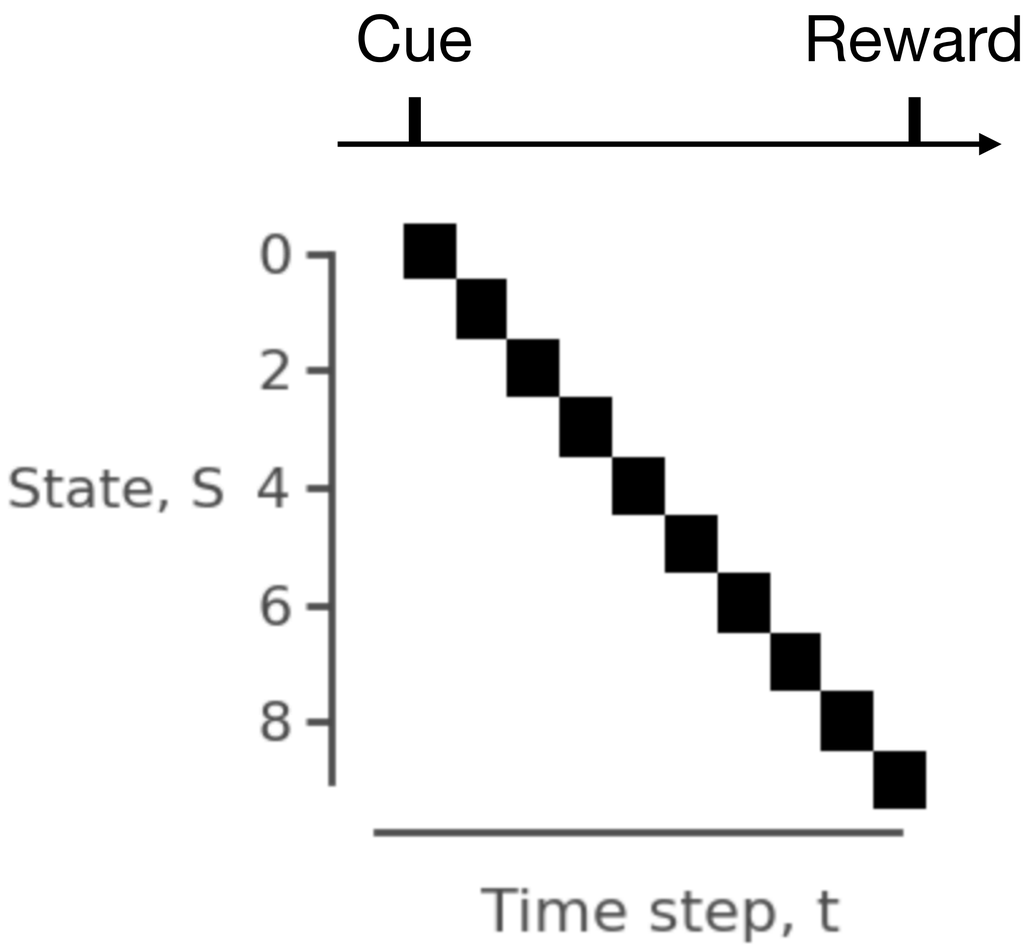
$$S_0 \xrightarrow{R_1 = 0} S_1 \xrightarrow{R_2 = 0} S_2 \xrightarrow{R_3 = 0} S_3 \xrightarrow{R_4 = 1} S_4$$

Episode 1	$\delta_0 = 0$	$\delta_1 = 0$	$\delta_2 = 0$	$\delta_3 = 1$
Episode 2	$\delta_0 = 0$	$\delta_1 = 0$	$\delta_2 > 0$	$\delta_3 > 0$
Episode 3	$\delta_0 = 0$	$\delta_1 > 0$	$\delta_2 > 0$	$\delta_3 > 0$

Modelling classical conditioning with TD RL

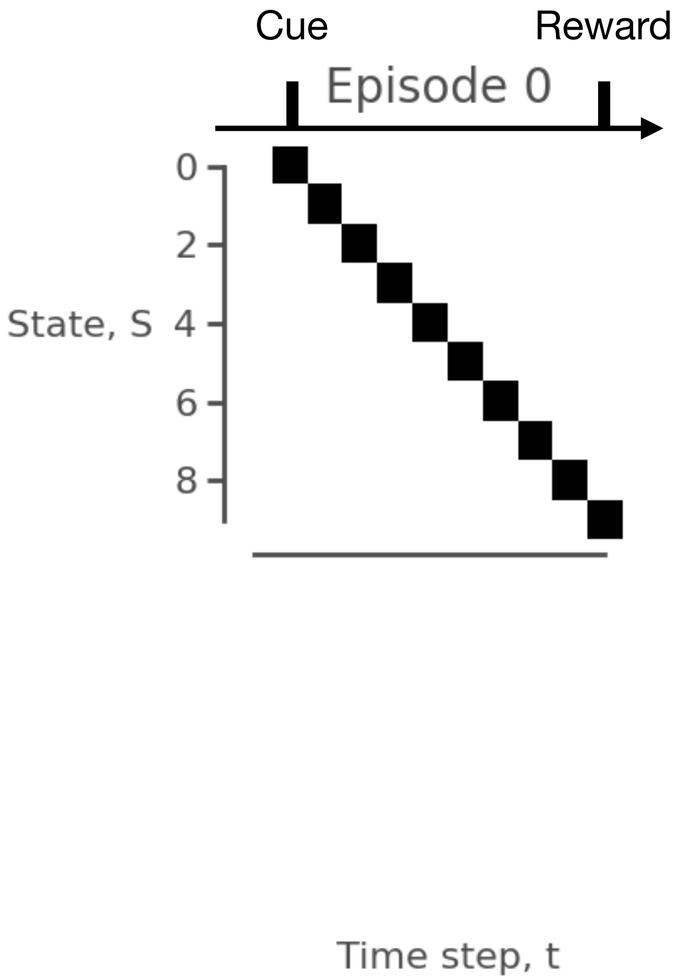
Time step, t

Modelling classical conditioning with TD RL



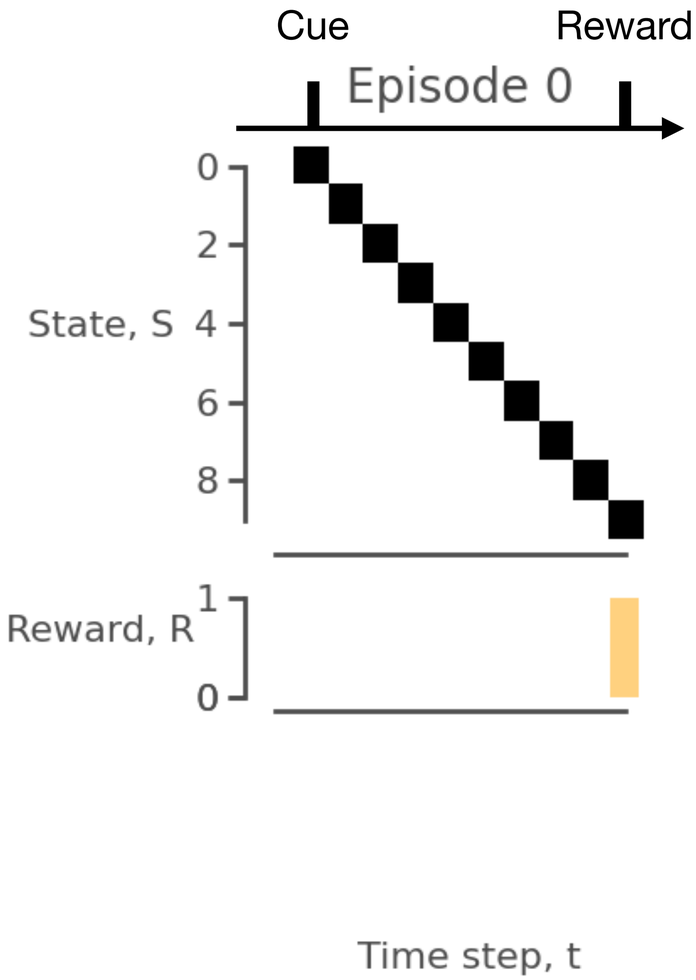
Schultz, Dayan & Montague (1998)

Dopamine as TD reward prediction error



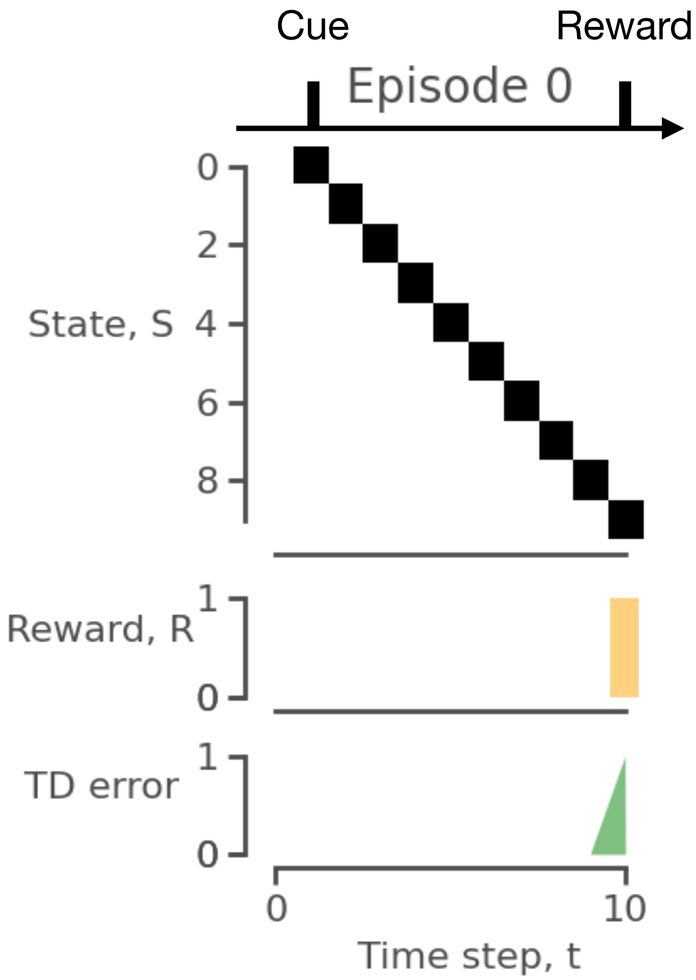
Schultz, Dayan & Montague (1998)

Dopamine as TD reward prediction error



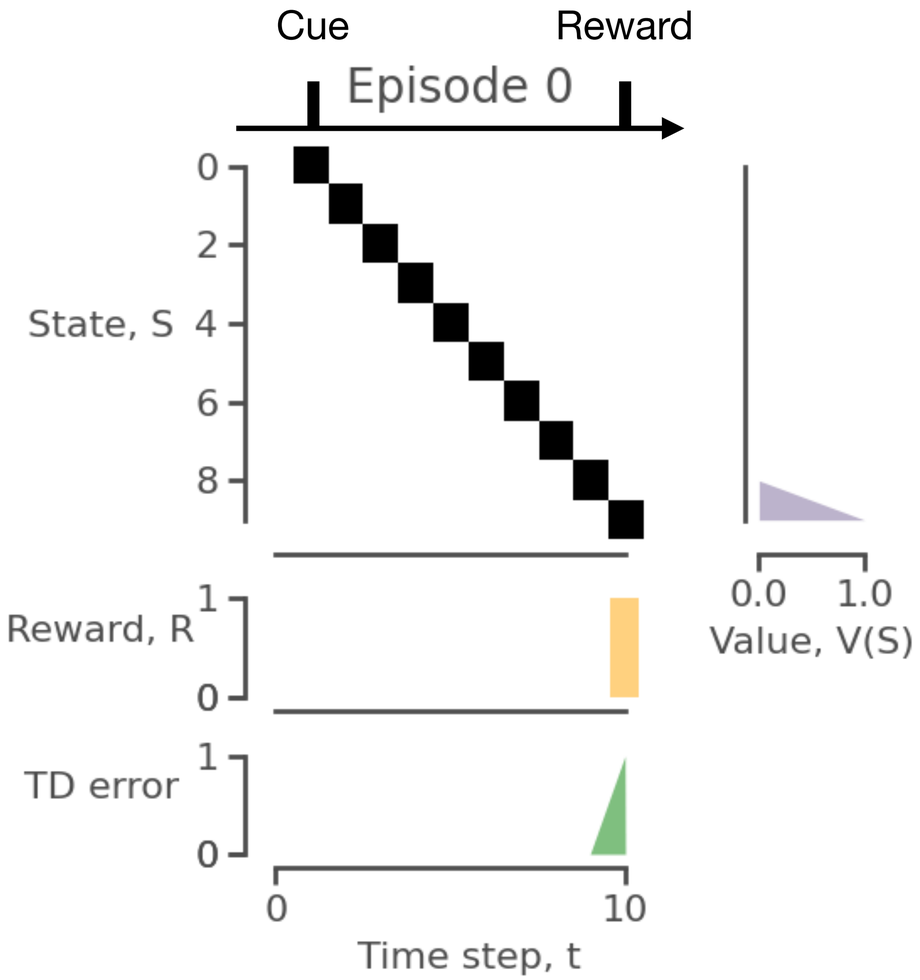
Schultz, Dayan & Montague (1998)

Dopamine as TD reward prediction error



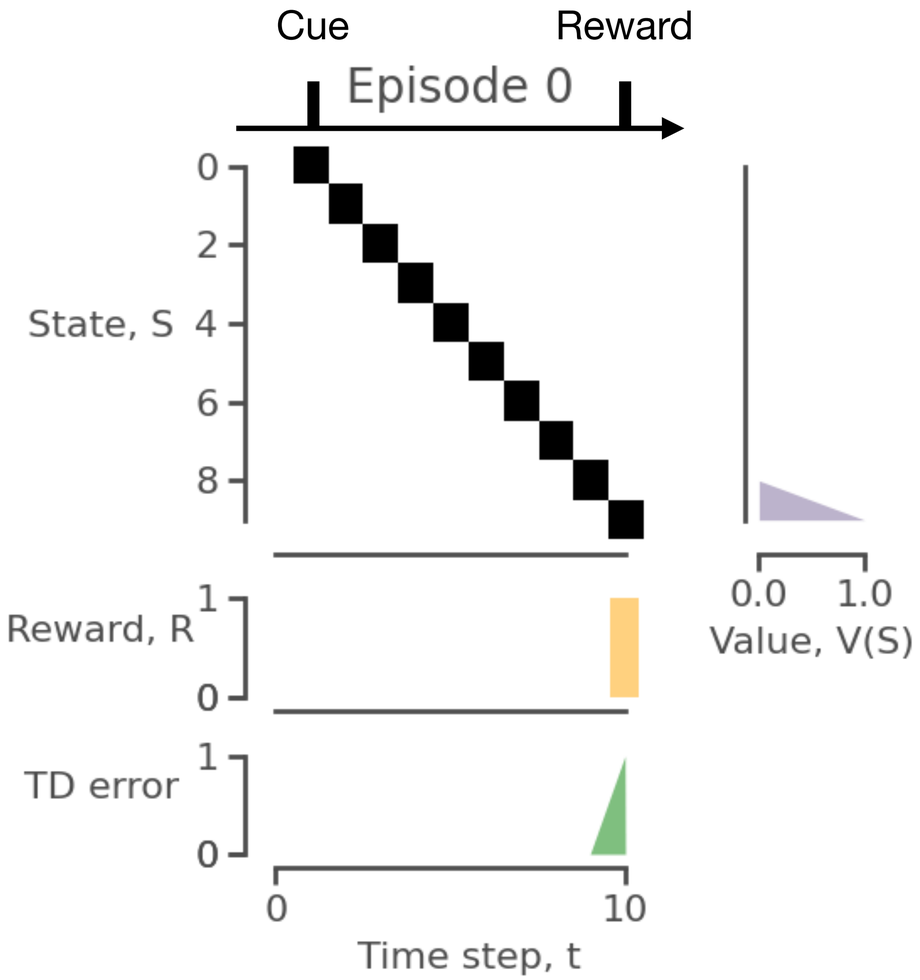
Schultz, Dayan & Montague (1998)

Dopamine as TD reward prediction error



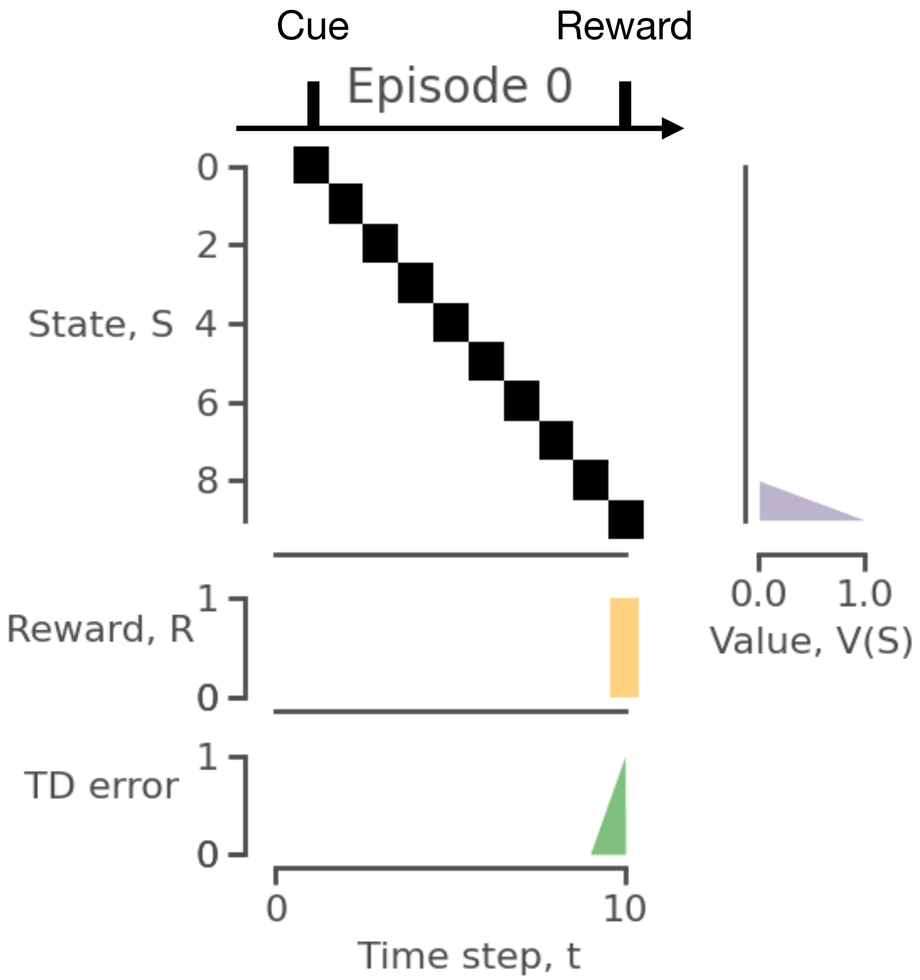
Schultz, Dayan & Montague (1998)

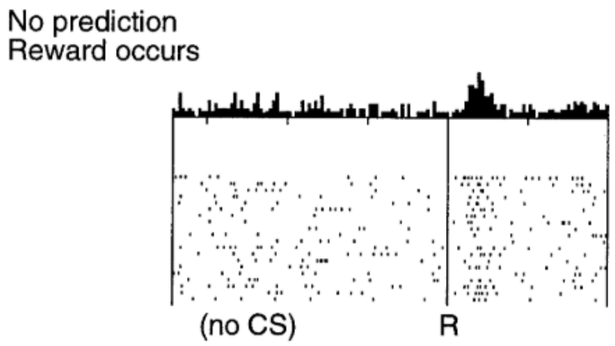
Dopamine as TD reward prediction error



Schultz, Dayan & Montague (1998)

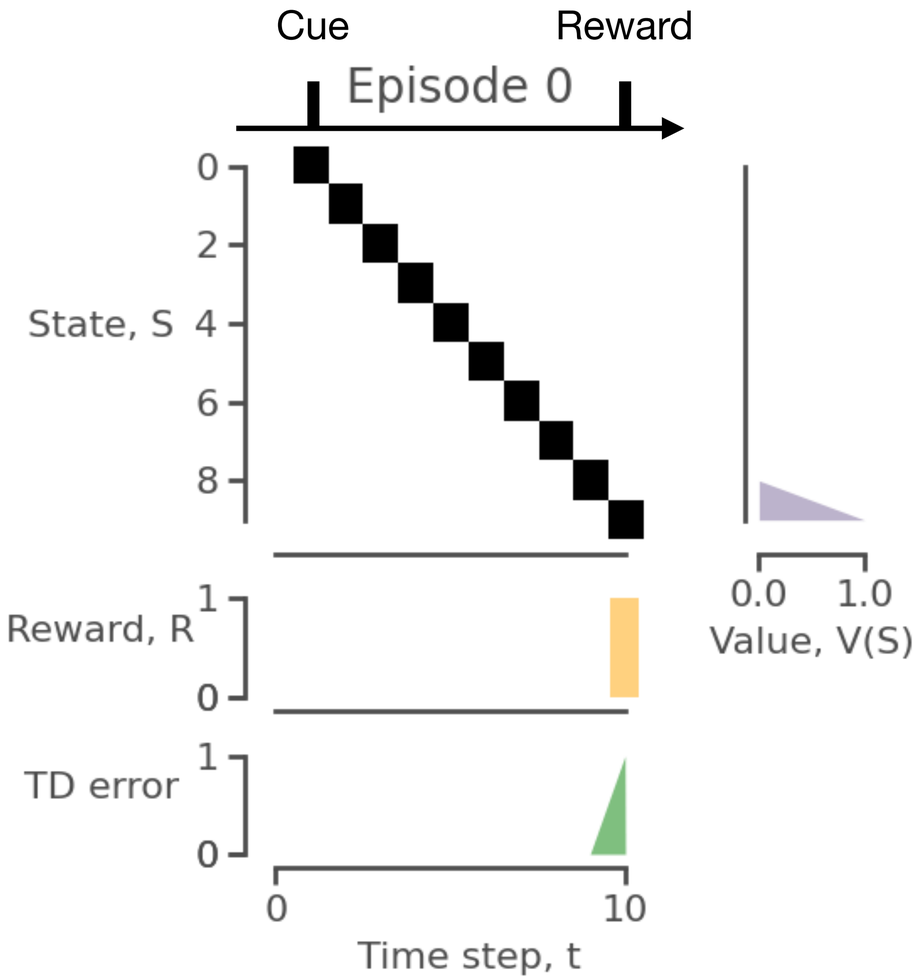
Dopamine as TD reward prediction error

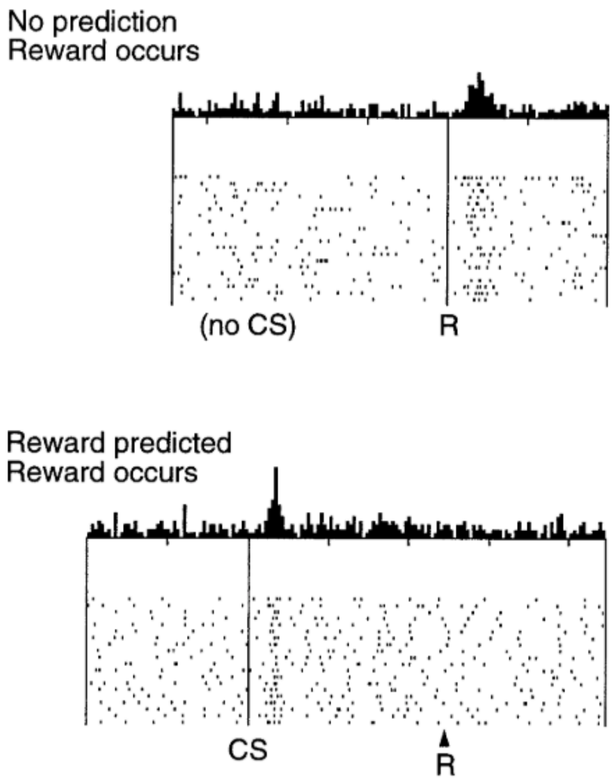




Schultz, Dayan & Montague (1998)

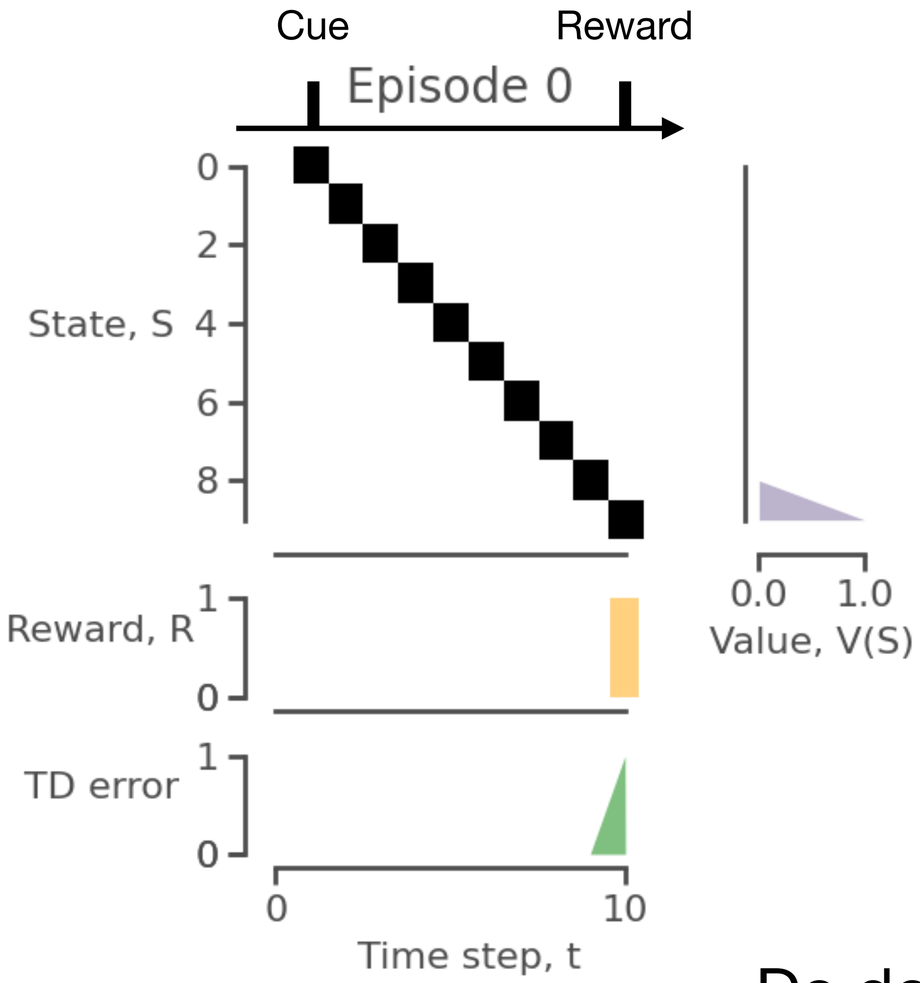
Dopamine as TD reward prediction error

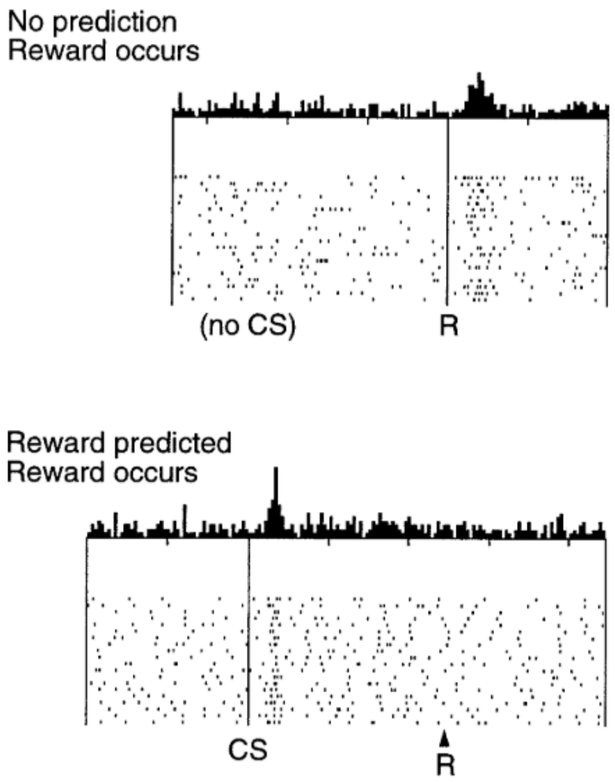




Schultz, Dayan & Montague (1998)

Dopamine as TD reward prediction error

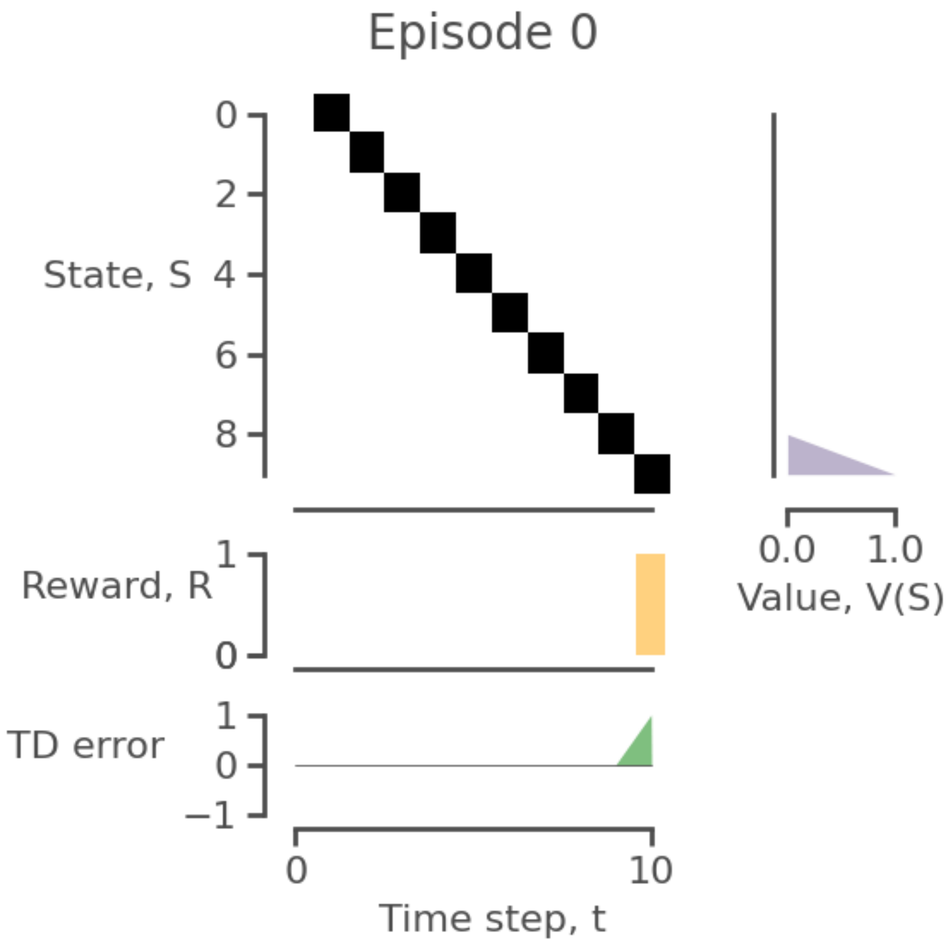




Do dopamine neurons encode the TD error?

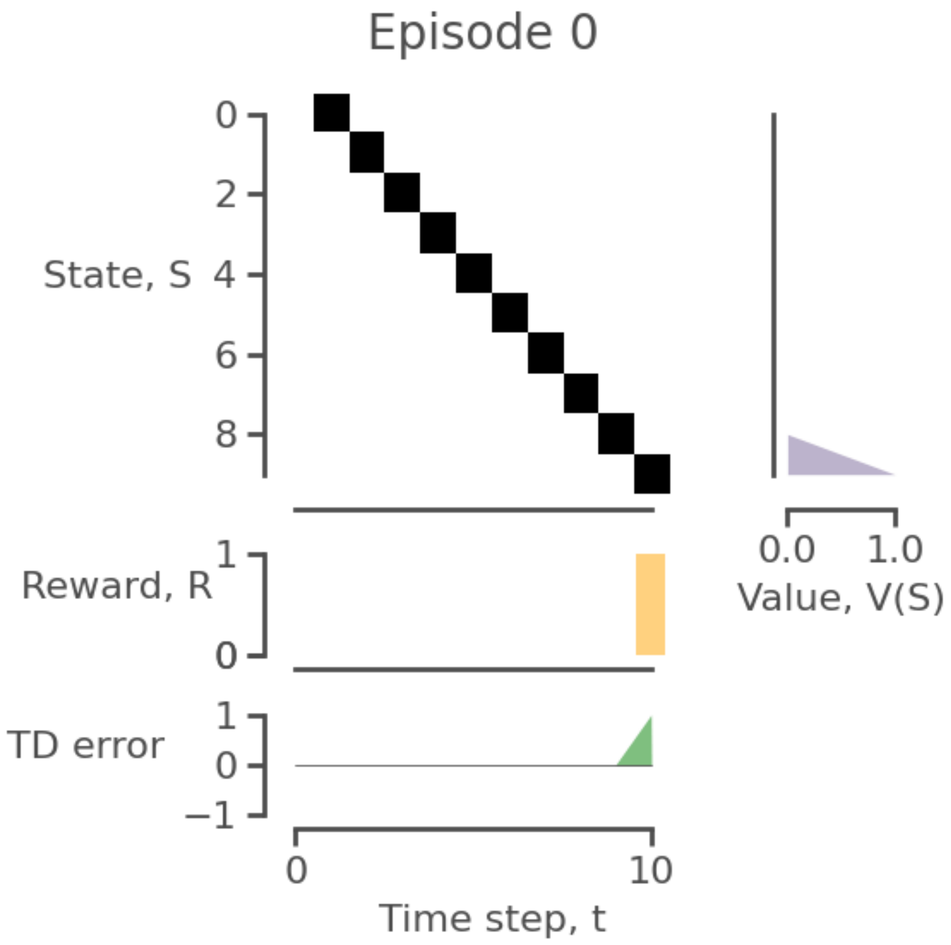
Schultz, Dayan & Montague (1998)

Dopamine as TD reward prediction error



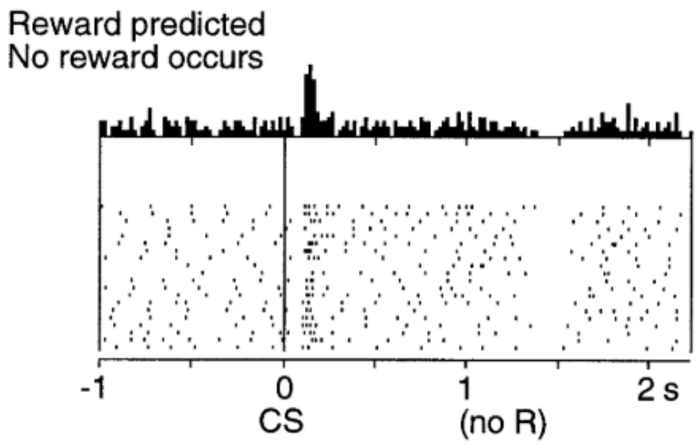
Schultz, Dayan & Montague (1998)

Dopamine as TD reward prediction error



Schultz, Dayan & Montague (1998)

Dopamine as TD reward prediction error



Play for today

- **How do we predict reward?**
  - Classical conditioning and associative learning theory
- **How do we predict *future* reward?**
  - Temporal difference learning and dopamine
- **How do we optimise actions to maximise future reward?**
  - Policy iteration, SARSA
- **What to do in large or continuous environments?**
  - Function approximation and representations of place
- **Representation learning for RL**
  - Learning a predictive representation to support more flexible RL
- **The limits of model-free RL and the case for a cognitive map**
- Afternoon: TUTORIAL

Learning to act: policy evaluation and improvement

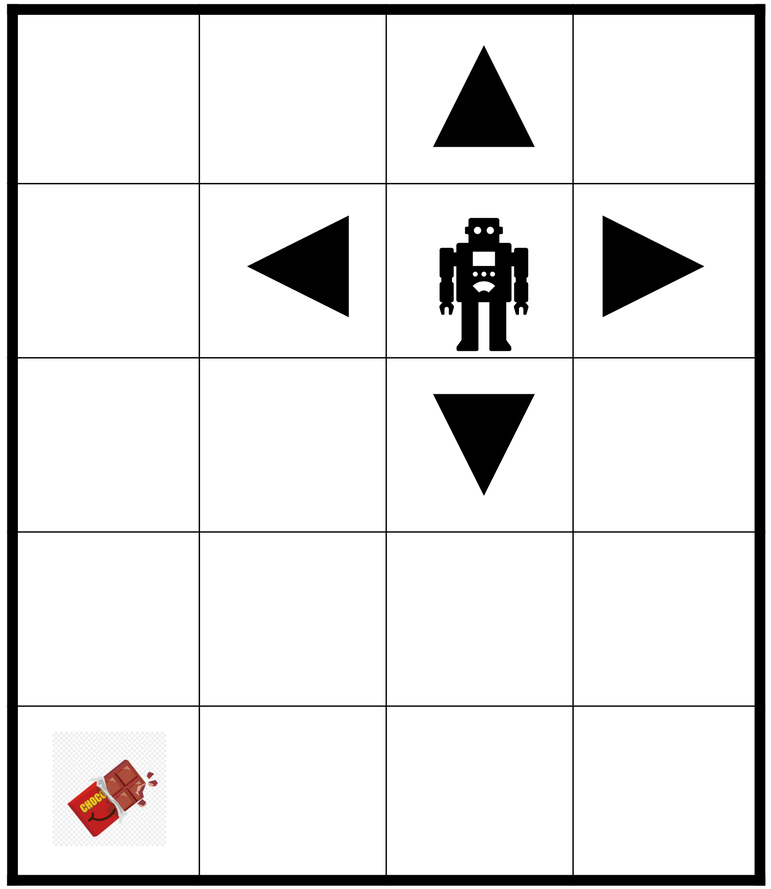
Markov reward process

$$S_0 \xrightarrow[R_1 = 0]{} S_1 \xrightarrow[R_2 = 0]{} S_2 \xrightarrow[R_3 = 0]{} S_3 \xrightarrow[R_4 = 1]{} S_4$$

Markov decision process: Grid World

Find a *policy* $\pi : S \rightarrow A$

which actions in which states?

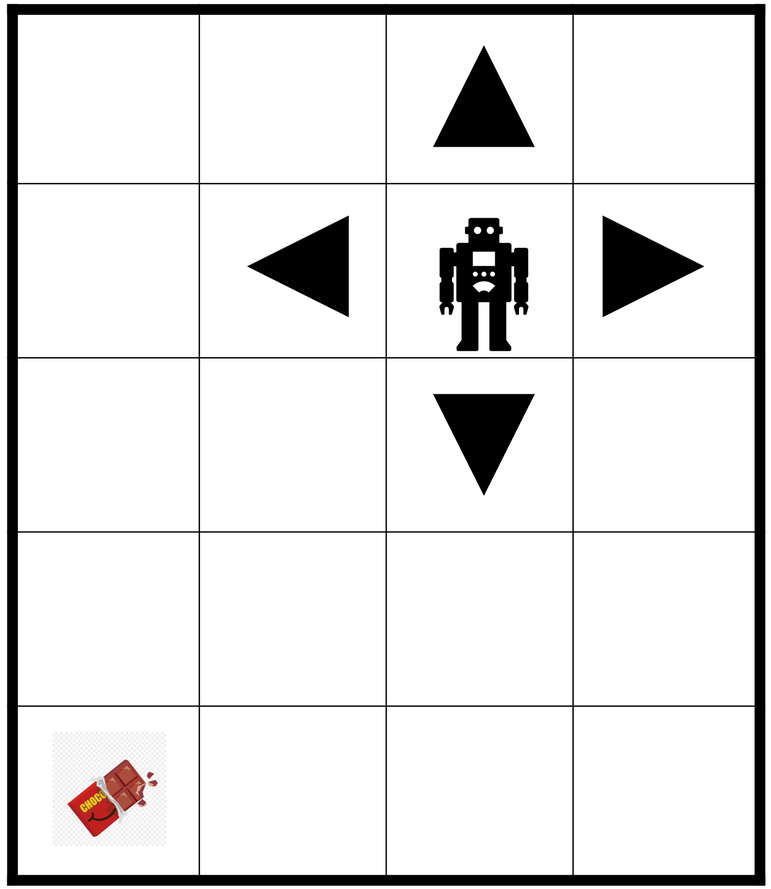


Markov decision process: Grid World

Find a *policy* $\pi : S \rightarrow A$

which actions in which states?

1. Policy evaluation

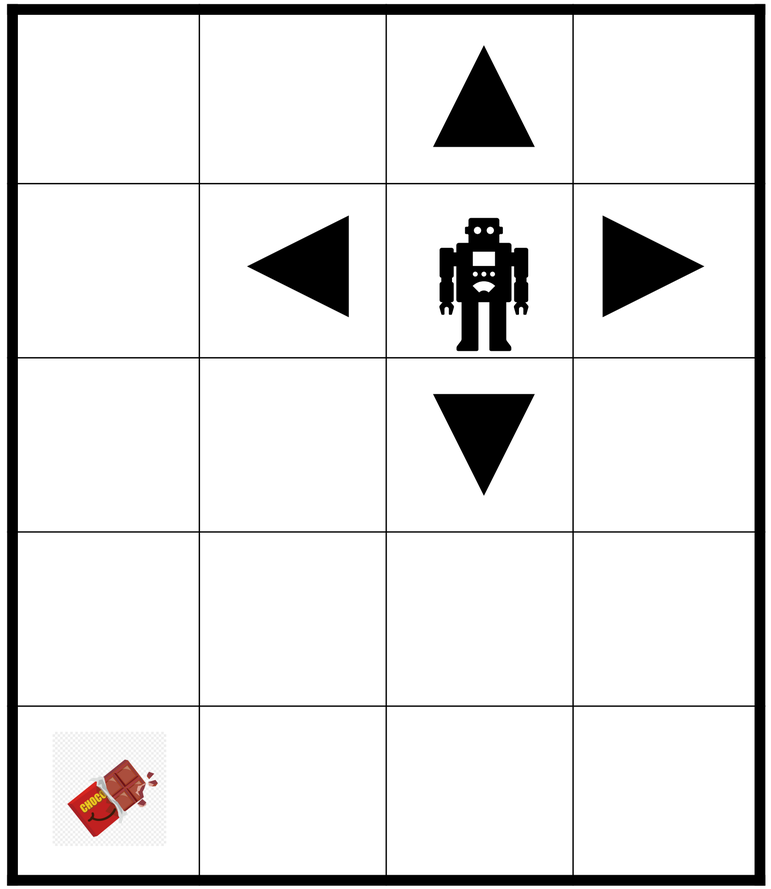


Markov decision process: Grid World

Find a *policy* $\pi : S \rightarrow A$

which actions in which states?

1. Policy evaluation
2. Policy improvement

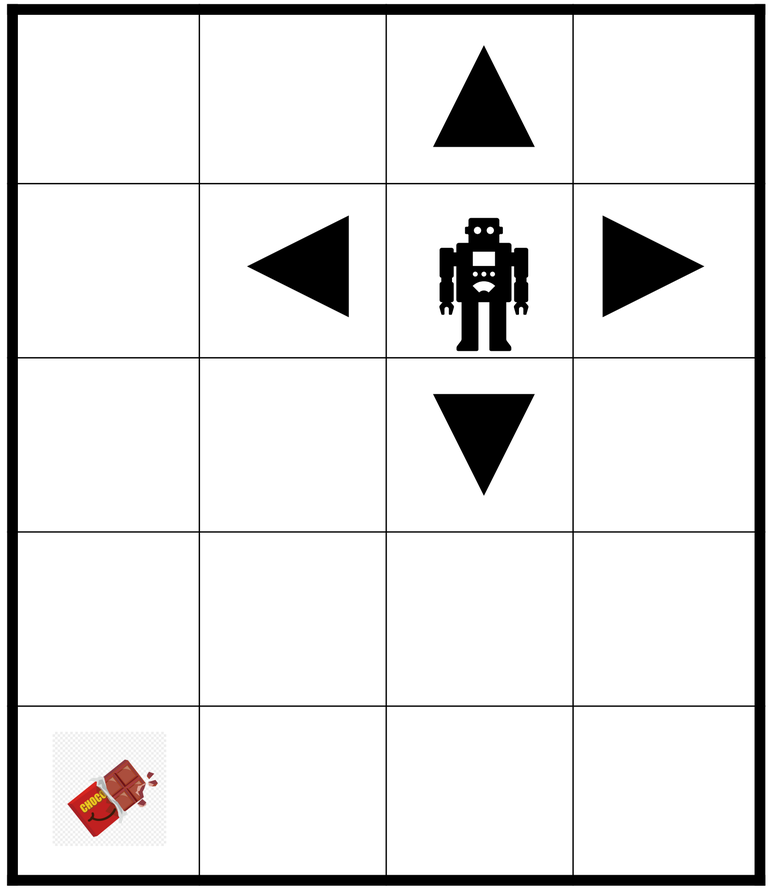


Markov decision process: Grid World

1: Policy evaluation

$$Q_\pi(s,a) = \mathbb{E}_\pi\left[R_{t+1} + \gamma R_{t+2} + \ldots \,\middle|\, S_t = s, A_t = a\right]$$

how good is each action,
under the current policy?



Markov decision process: Grid World

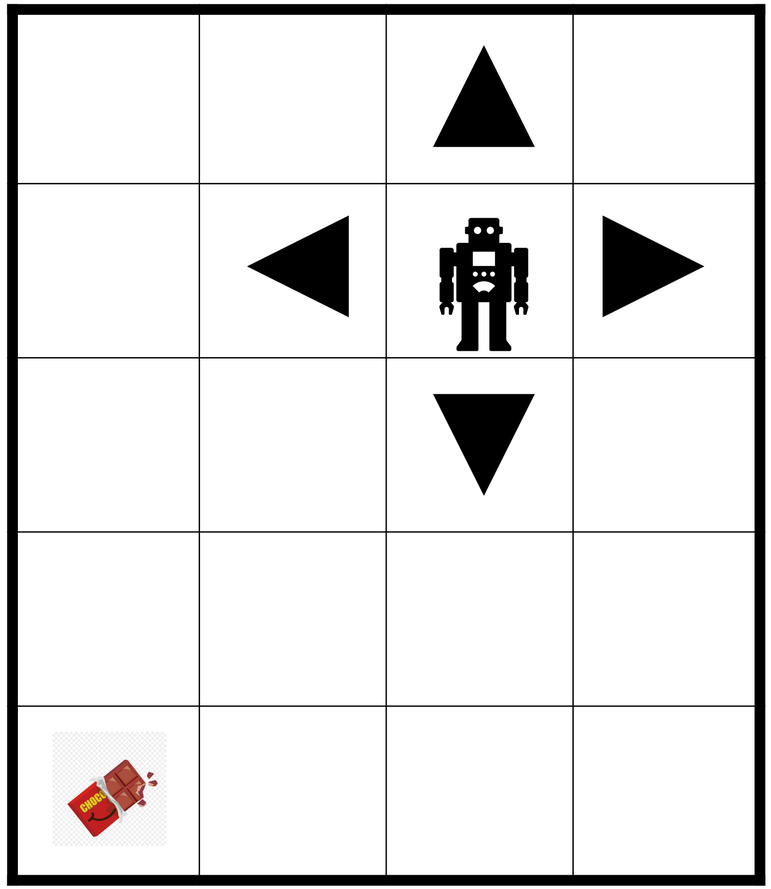
1: Policy evaluation

$$Q_\pi(s,a) = \mathbb{E}_\pi\left[R_{t+1} + \gamma R_{t+2} + \ldots \,\middle|\, S_t = s, A_t = a\right]$$

how good is each action,
under the current policy?

Learn Q using TD learning

$$\hat{Q}_\pi(S_t, A_t) \leftarrow \hat{Q}_\pi(S_t, A_t) + \alpha\left[R_{t+1} + \gamma \hat{Q}_\pi(S_{t+1}, A_{t+1}) - \hat{Q}_\pi(S_t, A_t)\right]$$

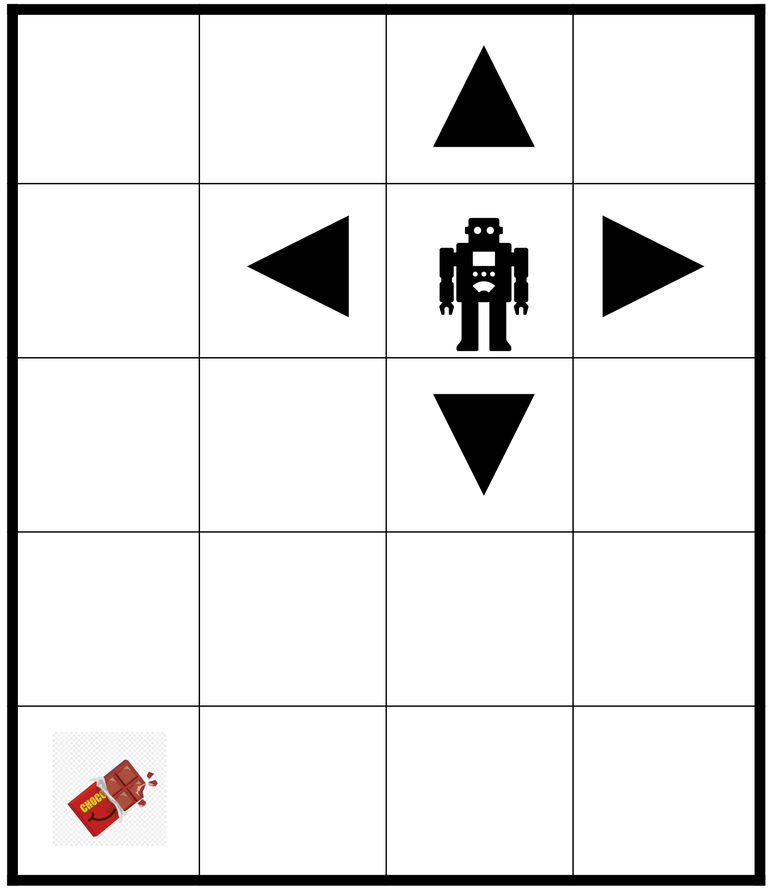


Markov decision process: Grid World

1: Policy evaluation

$$Q_\pi(s,a) = \mathbb{E}_\pi\left[R_{t+1} + \gamma R_{t+2} + \ldots \,\middle|\, S_t = s, A_t = a\right]$$

how good is each action,
under the current policy?



Learn Q using TD learning

$$\hat{Q}_\pi(S_t, A_t) \leftarrow \hat{Q}_\pi(S_t, A_t) + \alpha\left[R_{t+1} + \gamma \hat{Q}_\pi(S_{t+1}, A_{t+1}) - \hat{Q}_\pi(S_t, A_t)\right]$$

SARSA

Markov decision process: Grid World

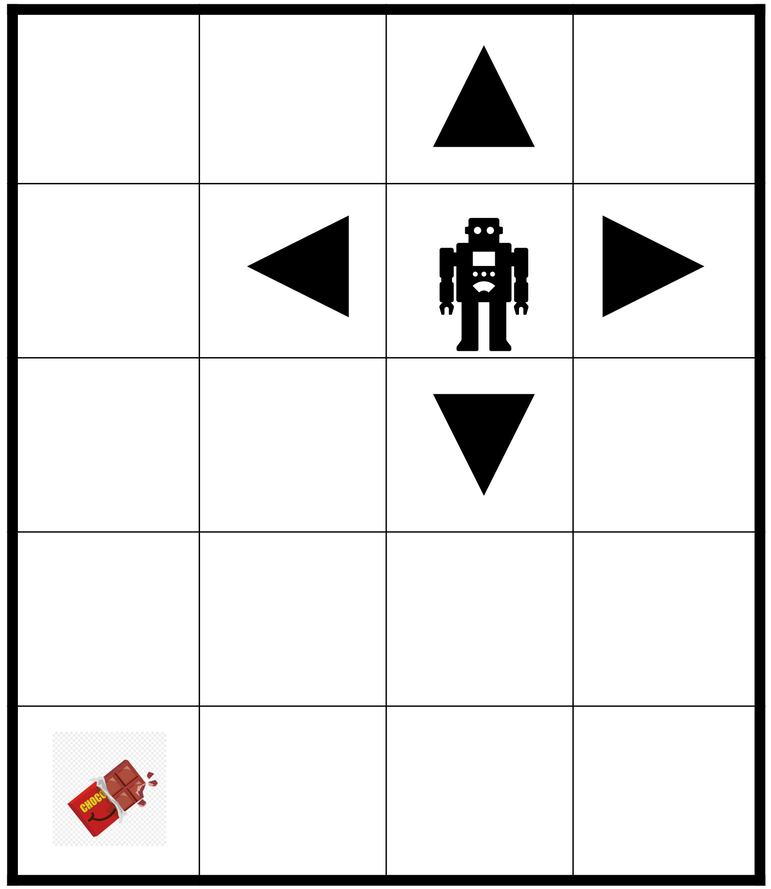
1: Policy evaluation

$$Q_\pi(s,a) = \mathbb{E}_\pi\left[R_{t+1} + \gamma R_{t+2} + \ldots \,\middle|\, S_t = s, A_t = a\right]$$

how good is each action,
under the current policy?

$Q_\pi(s,a)$

	N	W	E	S
S_1				
S_2				
...				

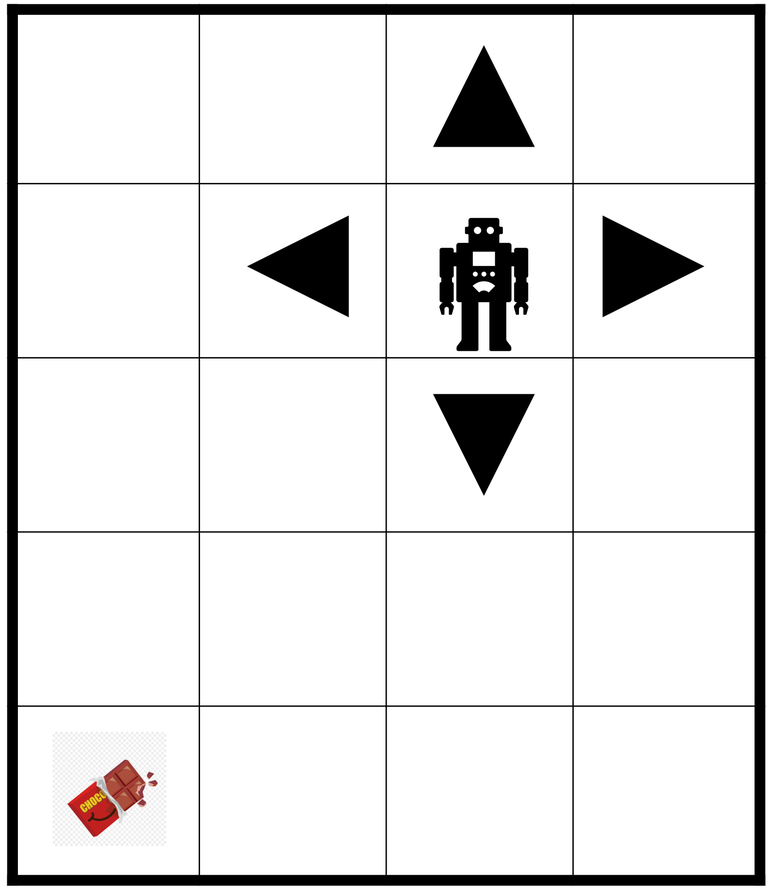


Markov decision process: Grid World

2: Policy improvement: greedy

$$\pi'(s) = \arg\max_a \hat{Q}_\pi(s, a)$$

in each state, take action with maximum estimated value



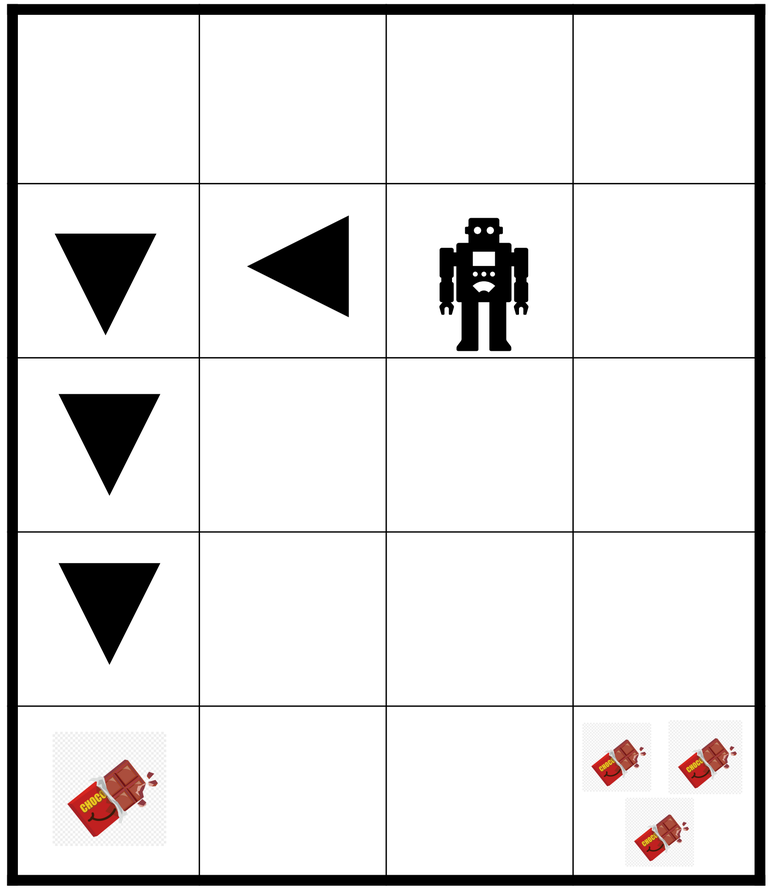
Markov decision process: Grid World

2: Policy improvement: greedy

$$\pi'(s) = \arg\max_a \hat{Q}_\pi(s, a)$$

in each state, take action with maximum estimated value

explore-exploit dilemma

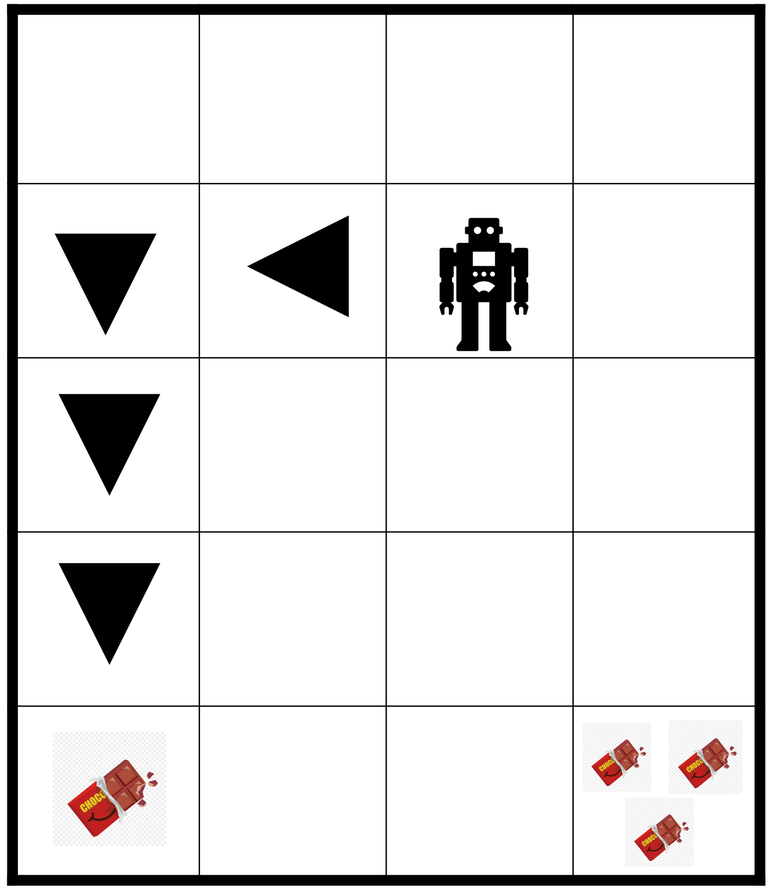


Markov decision process: Grid World

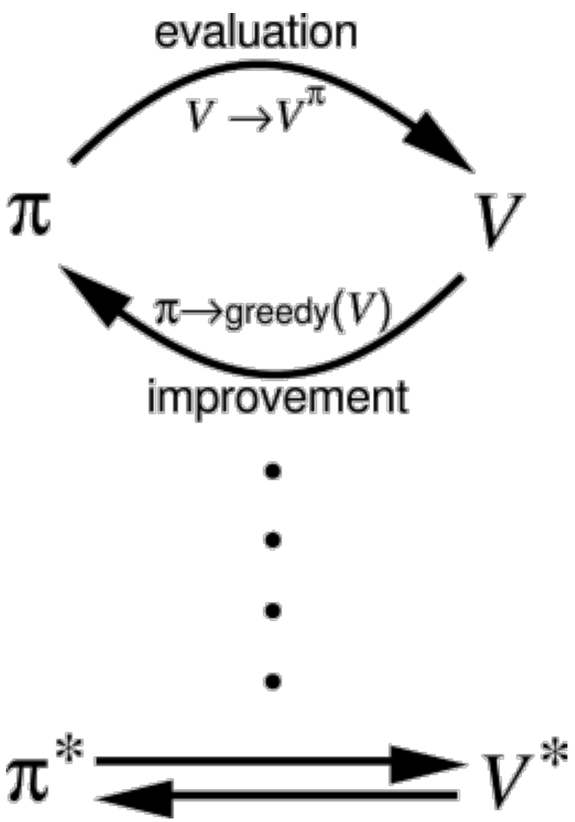
2: Policy improvement: epsilon greedy

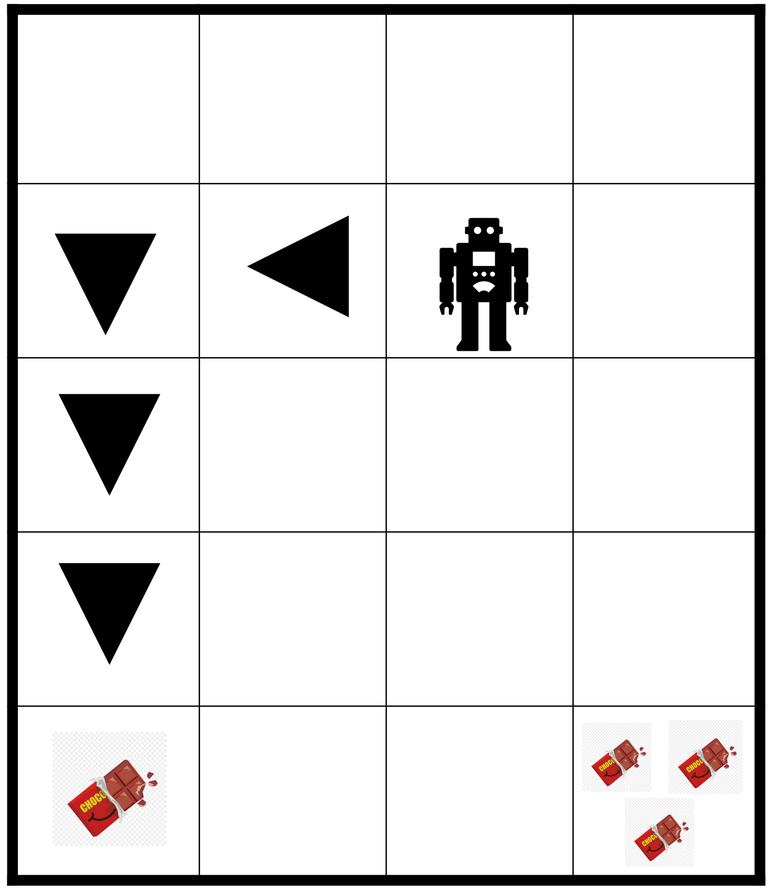
$$\pi'(s) = \begin{cases} \arg\max_a Q_\pi(s,a) & \text{with probability } 1-\epsilon \\ \text{random action} & \text{with probability } \epsilon \end{cases}$$

in each state, take action with maximum estimated value (with probability 1-epsilon)



Policy iteration





We’ve only shown one of MANY methods

We’ve only shown one of MANY methods

- Action value methods versus policy gradient

We’ve only shown one of MANY methods

- Action value methods versus policy gradient
  - Combination: actor-critic

We’ve only shown one of MANY methods

- Action value methods versus policy gradient
  - Combination: actor-critic
- On-policy versus off-policy learning

We’ve only shown one of MANY methods

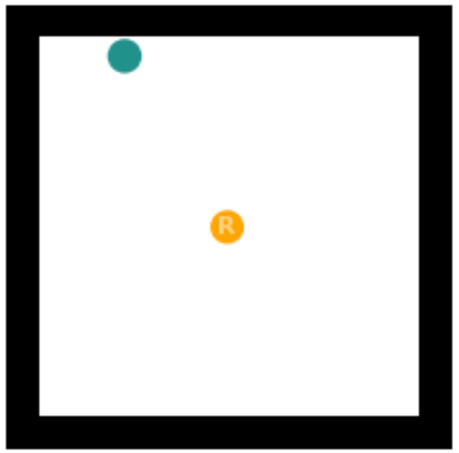
- Action value methods versus policy gradient
  - Combination: actor-critic
- On-policy versus off-policy learning
- Random exploration versus directed exploration

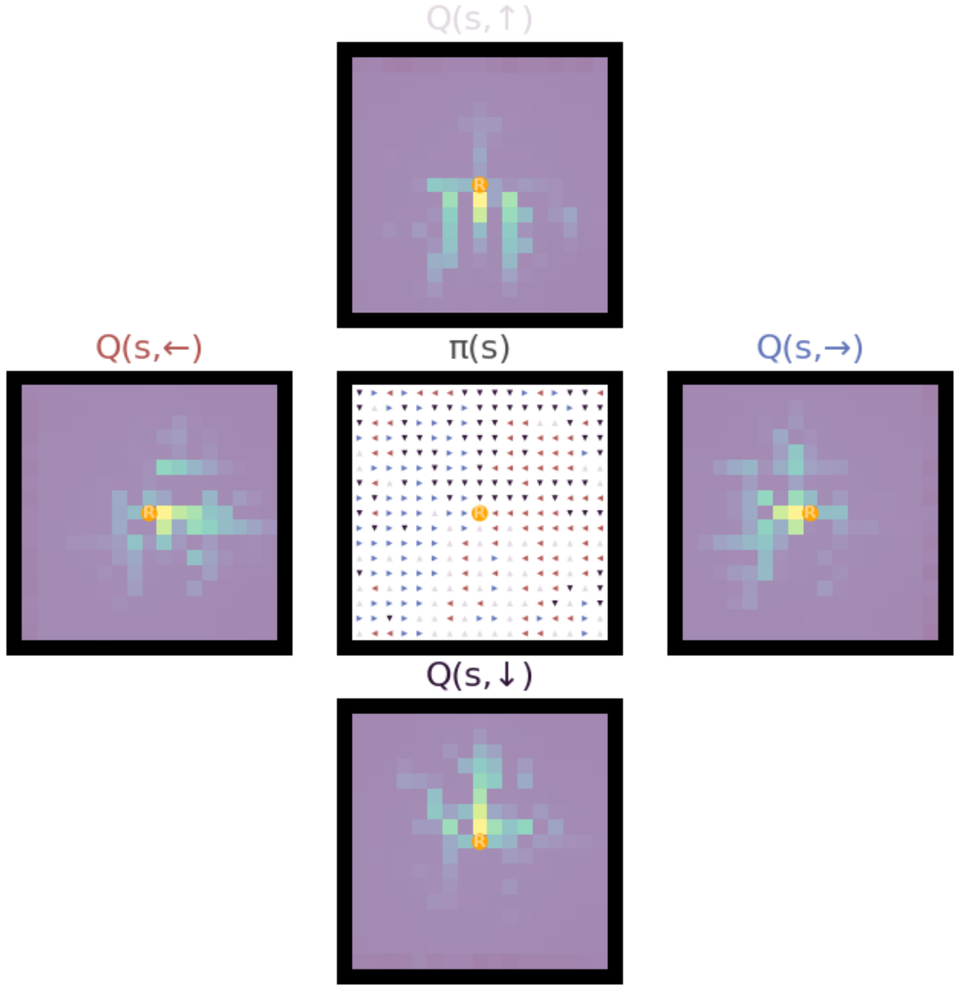
We’ve only shown one of MANY methods

- Action value methods versus policy gradient
  - Combination: actor-critic
- On-policy versus off-policy learning
- Random exploration versus directed exploration
- Etc

Tutorial exercise: implement SARSA on a grid world navigation task

A minigrid environment





Plan for today

- **How do we predict reward?**
  - Classical conditioning and associative learning theory
- **How do we predict *future* reward?**
  - Temporal difference learning and dopamine
- **How do we optimise actions to maximise future reward?**
  - Policy iteration, SARSA
- **What to do in large or continuous environments?**
  - Function approximation and representations of place
- **Representation learning for RL**
  - Learning a predictive representation to support more flexible RL
- **The limits of model-free RL and the case for a cognitive map**
- Afternoon: TUTORIAL

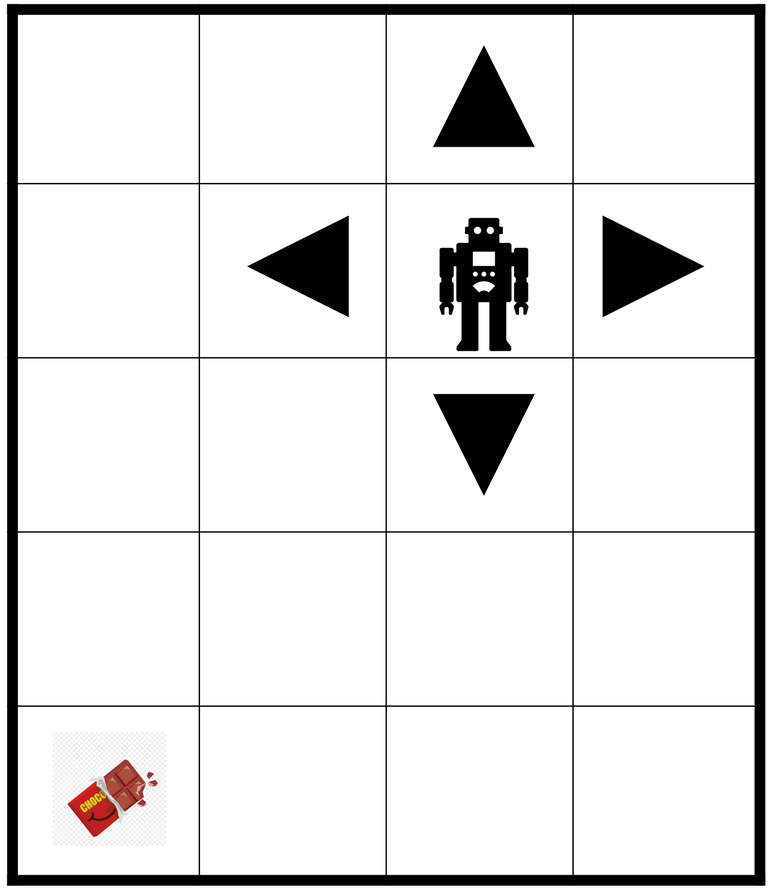
RL in large or continuous state spaces

Tabular RL: represent states as table entries

$Q_\pi(s, a)$

	N	W	E	S
S_1				
S_2				
...				

20 rows

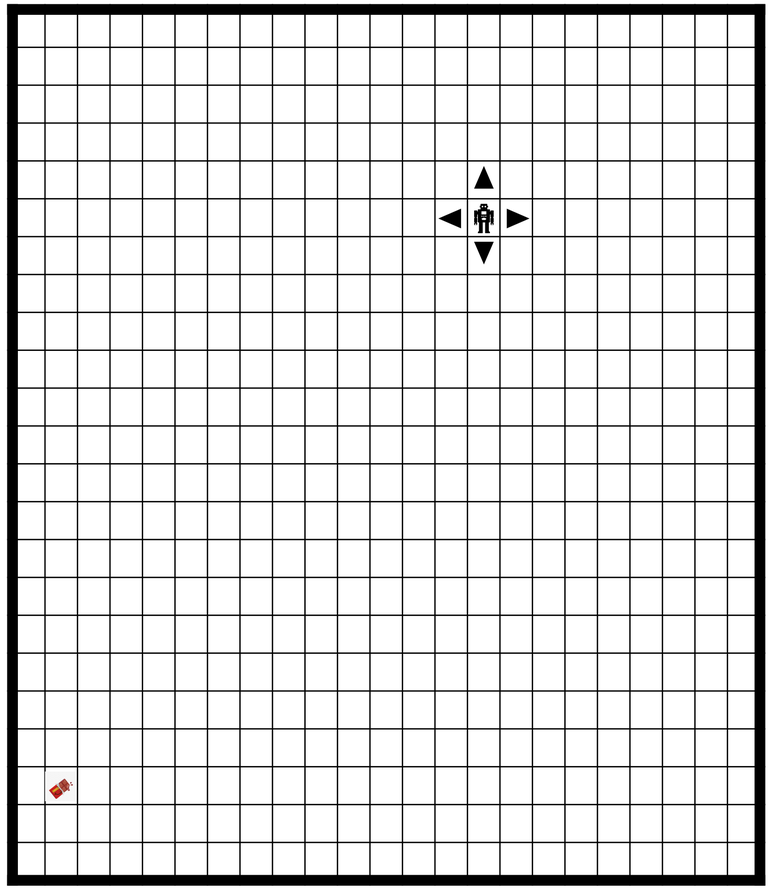


Tabular RL does not work for large environments

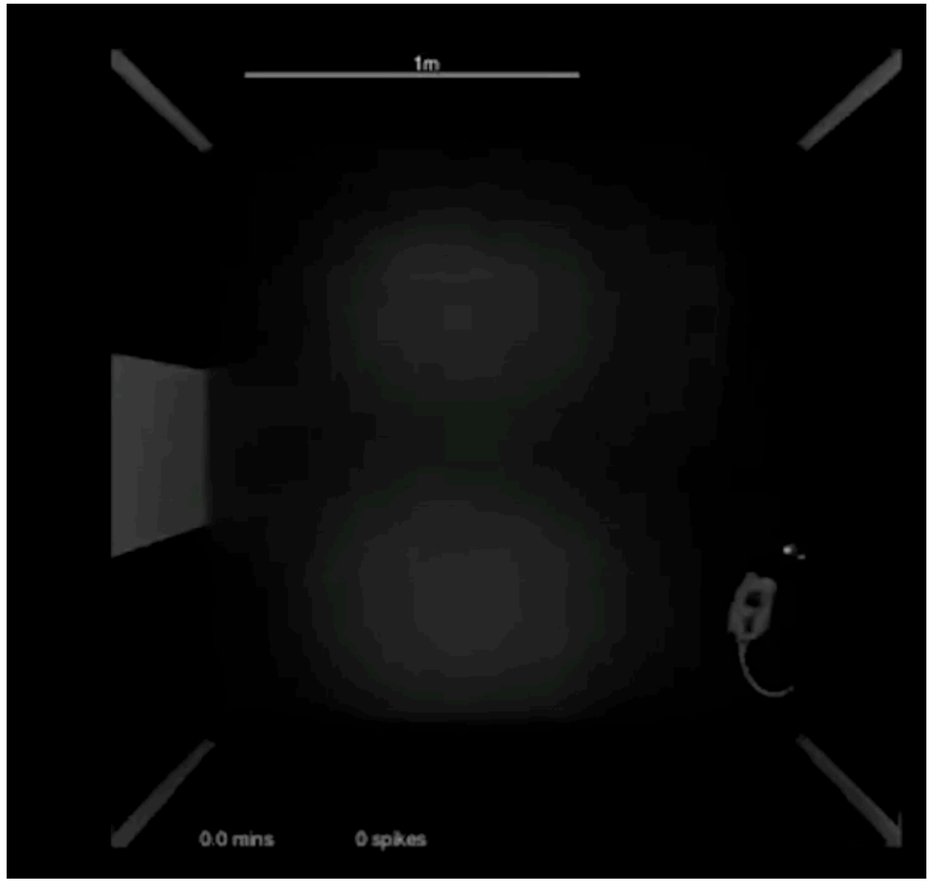
$Q_\pi(s, a)$

	N	W	E	S
S_1				
S_2				
...				

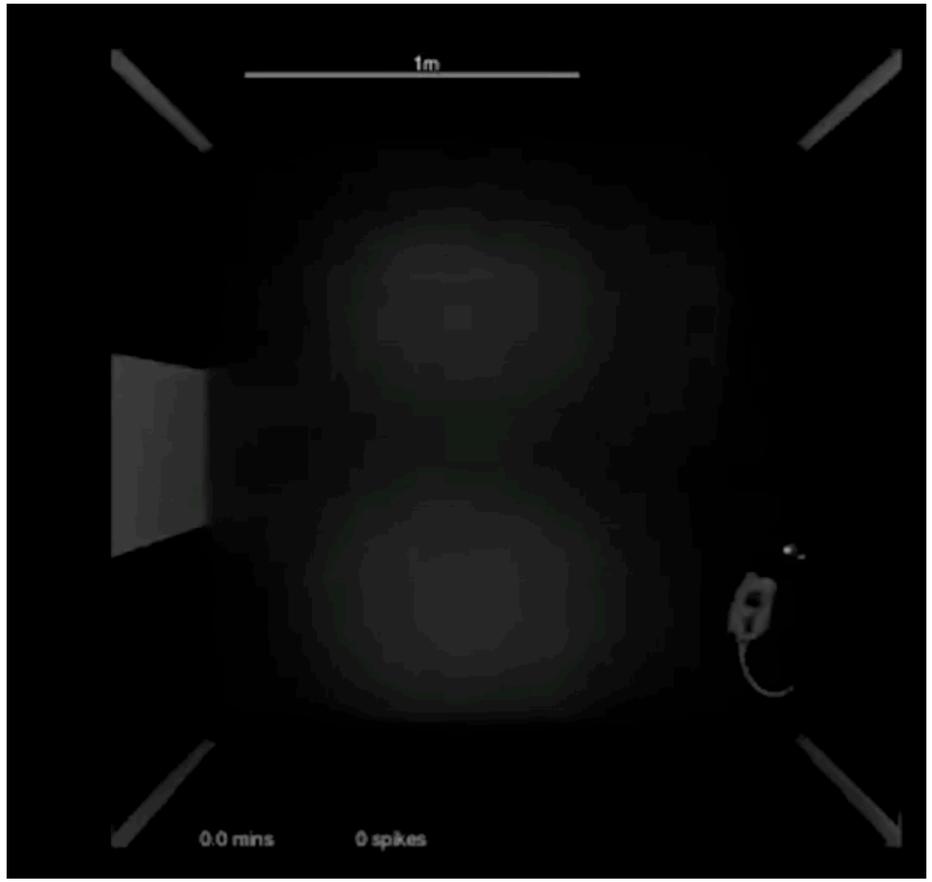
529 rows

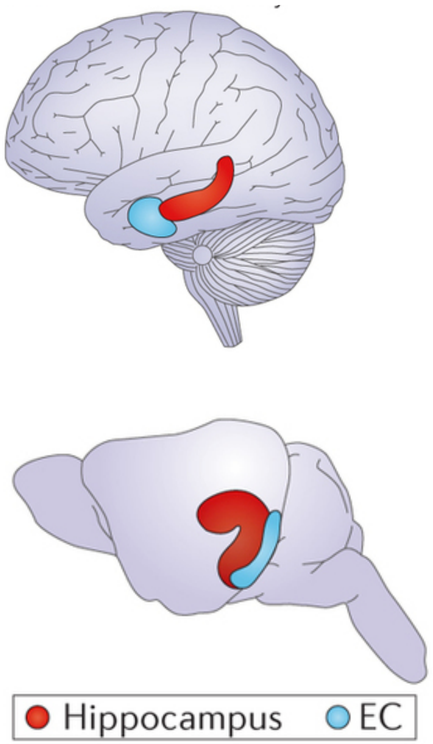


Some inspiration from neuroscience: place cells

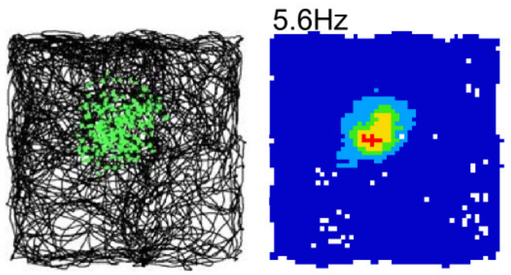


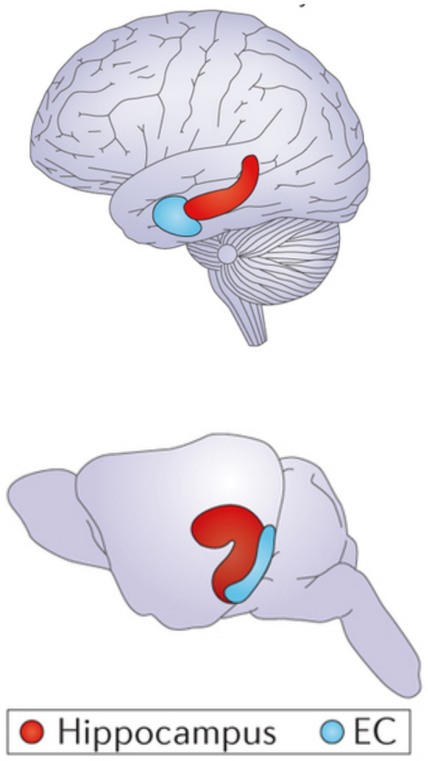
Some inspiration from neuroscience: place cells





Place cells

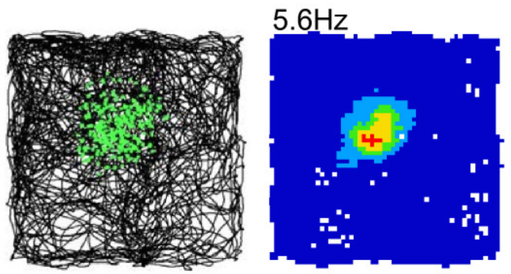


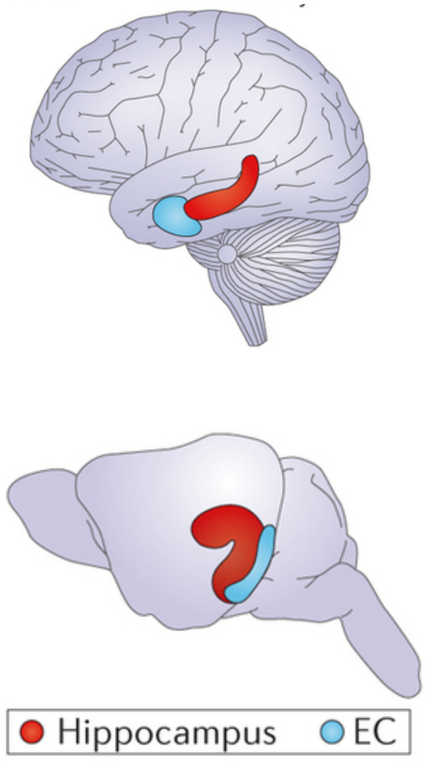


O'Keefe & Dostrovsky (1971)

Place cells

- Found in hippocampus
- Firing is dependent on location

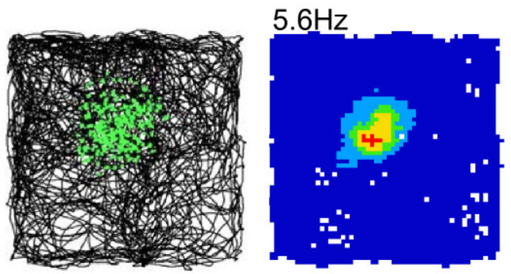


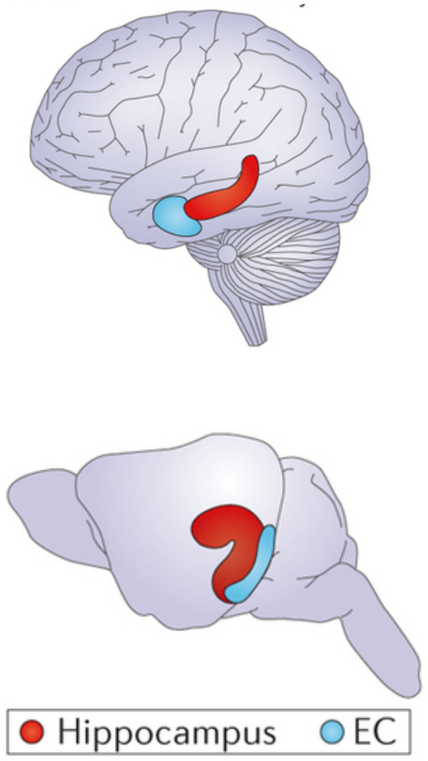


O'Keefe & Dostrovsky (1971)

Place cells

- Found in hippocampus
- Firing is dependent on location
  - Place field centre has highest firing

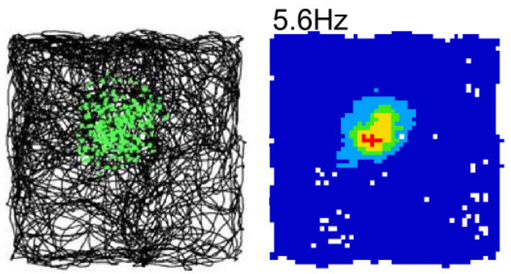


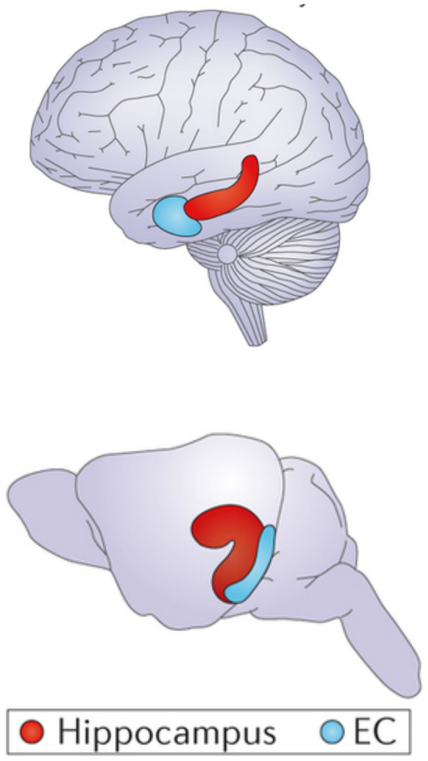


O'Keefe & Dostrovsky (1971)

Place cells

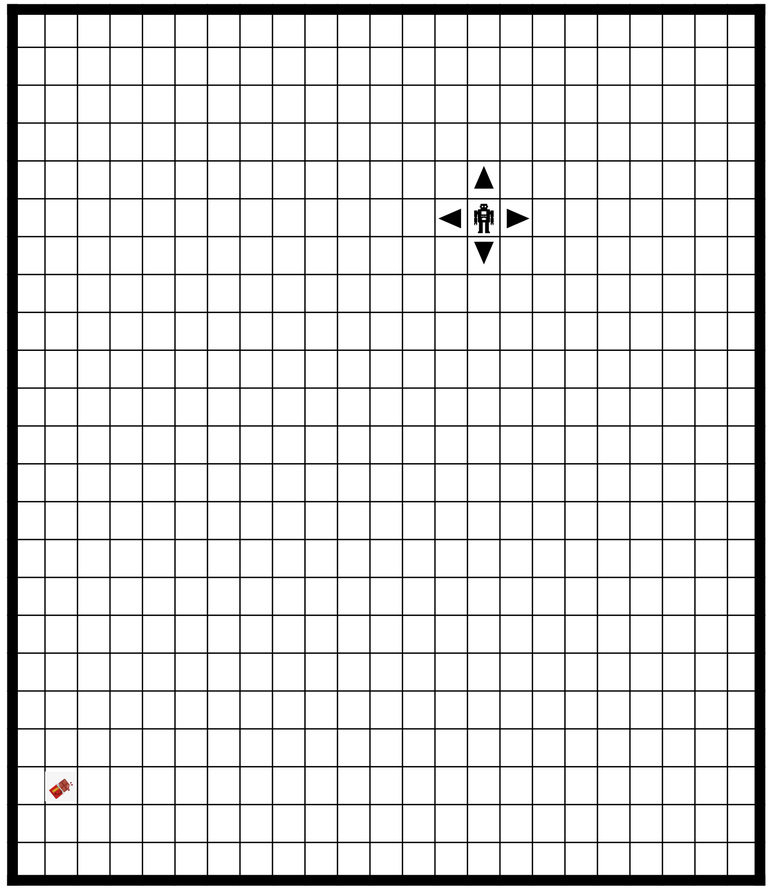
- Found in hippocampus
- Firing is dependent on location
  - Place field centre has highest firing
  - Some firing when animal is close to the centre



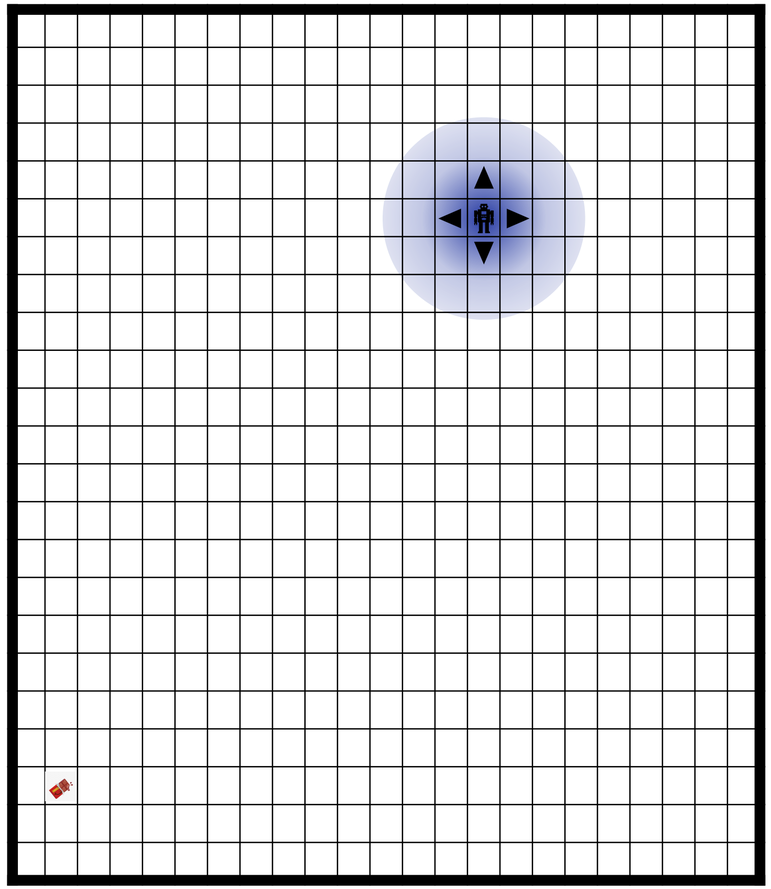


O'Keefe & Dostrovsky (1971)

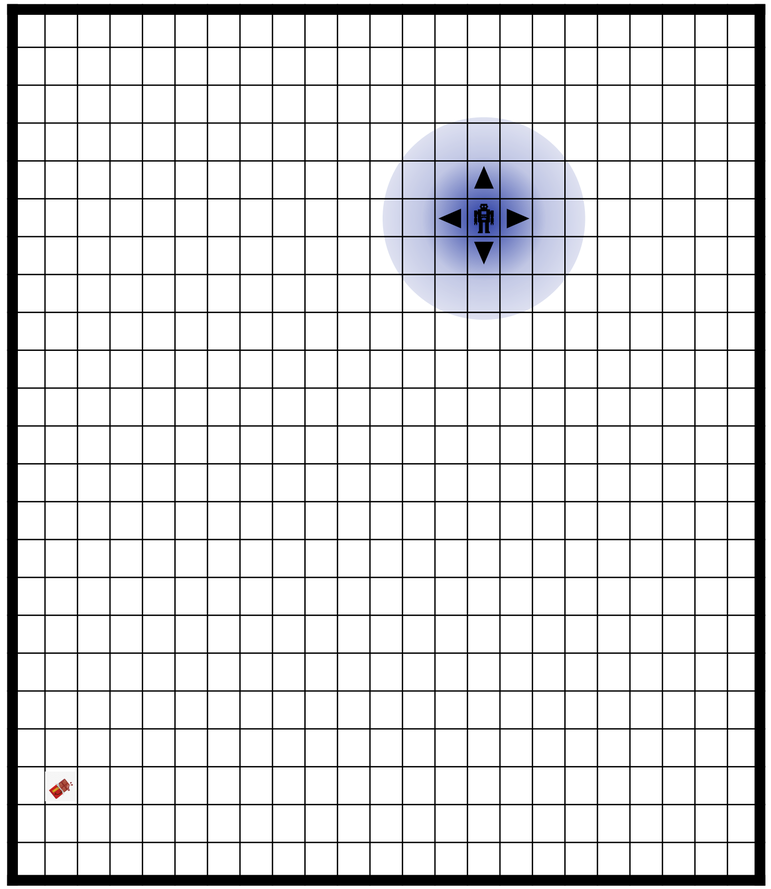
Approximating the value function using place cell-like features



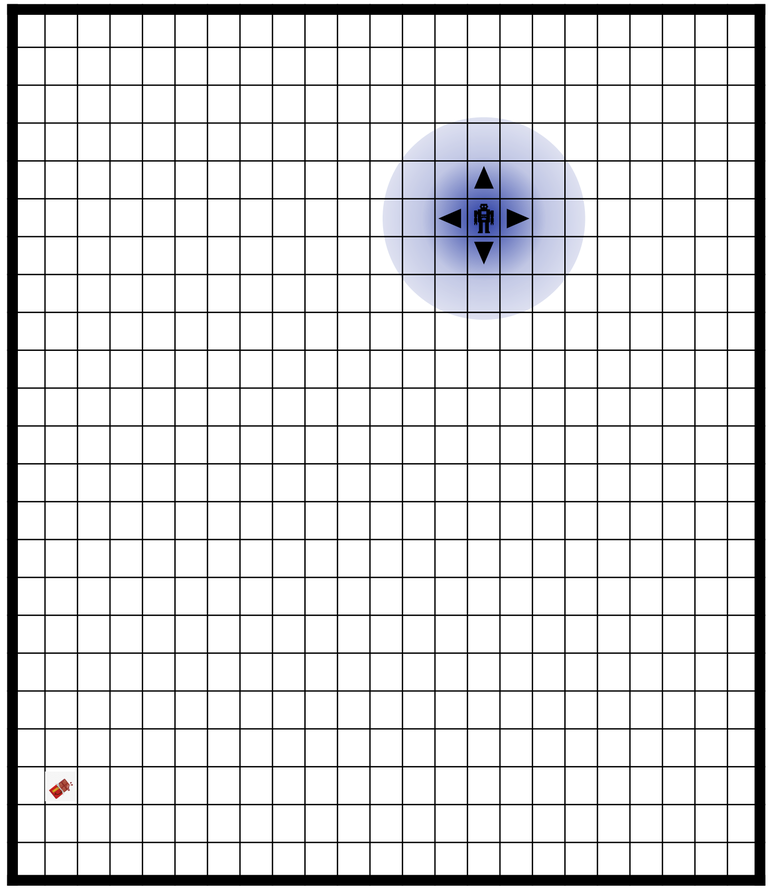
Approximating the value function using place cell-like features



Approximating the value function using place cell-like features

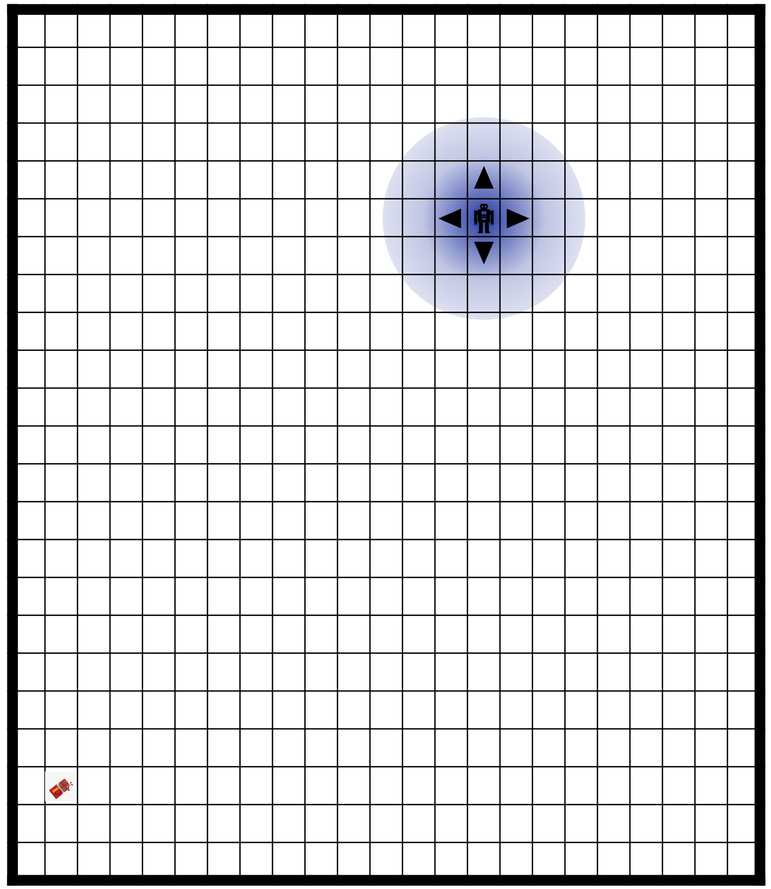


Approximating the value function using place cell-like features



Approximating the value function using place cell-like features

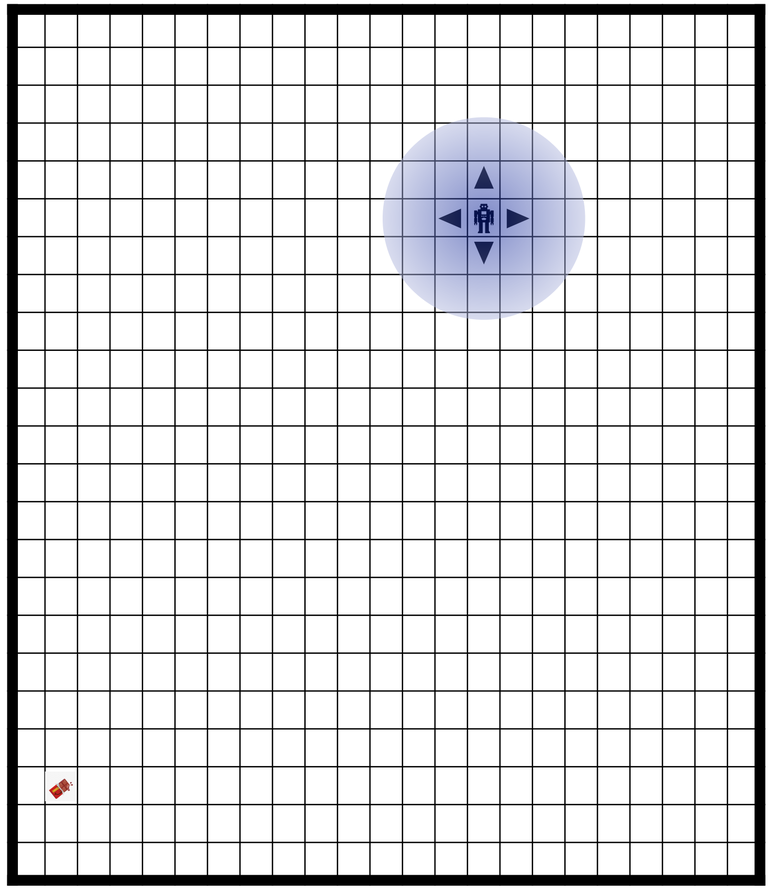
- Input: state features
  $s \rightarrow \phi(s) = [\phi_1(s), \phi_2(s), \ldots, \phi_{N_f}(s)]^\top$
- Approximation: $\hat{Q}_\theta(s, a)$
- Here we keep actions discretised



Approximating the value function using state features

- Approximation: $\hat{Q}_\theta(s, a)$

- $\hat{Q}_W(s, a) = [W^\top \boldsymbol{\phi}(s)]_a$

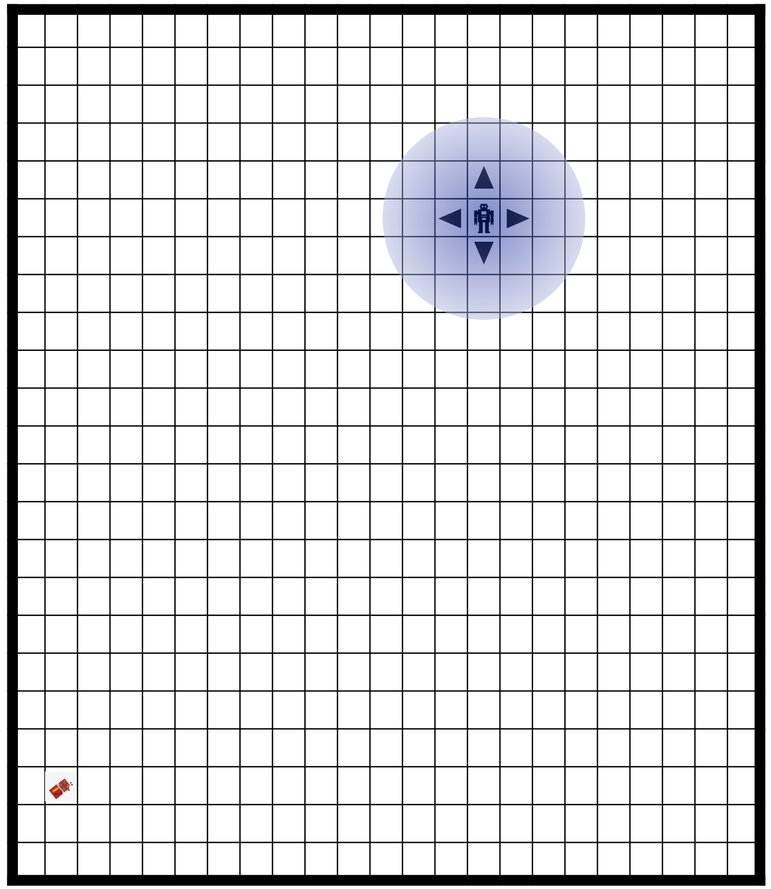


Approximating the value function using state features

- Approximation: $\hat{Q}_\theta(s, a)$

- $\hat{Q}_W(s, a) = [W^\top \boldsymbol{\phi}(s)]_a$

$$W \leftarrow W + \alpha \left[R_{t+1} + \gamma \hat{Q}_W(S_{t+1}) - \hat{Q}_W(S_t) \right] \boldsymbol{\phi}(S_t)$$

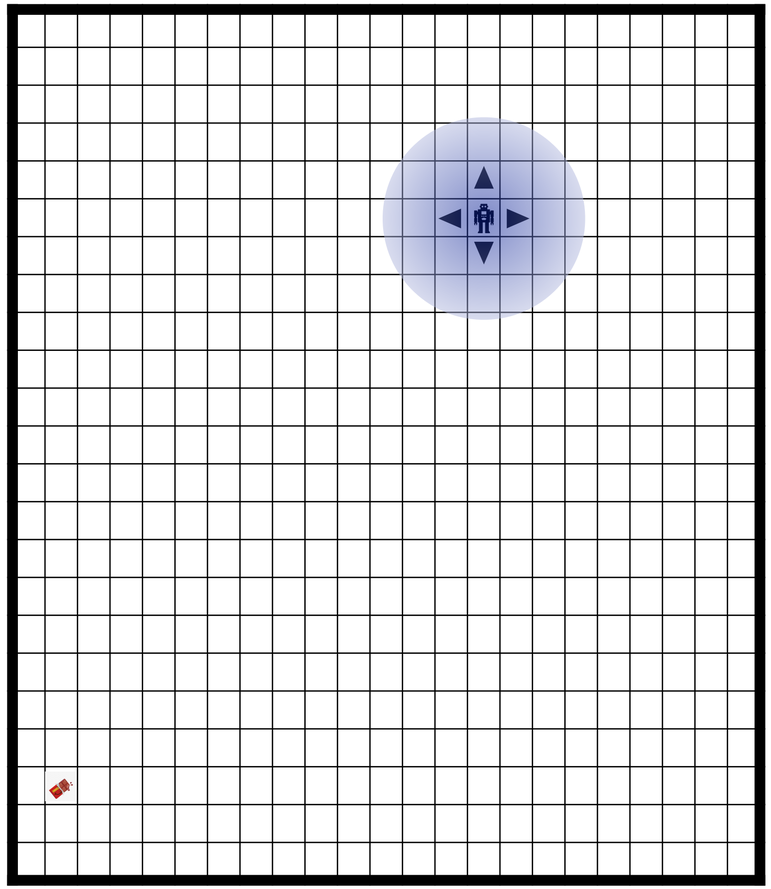


Approximating the value function using state features

- Approximation: $\hat{Q}_\theta(s, a)$

- $\hat{Q}_W(s, a) = [W^\top \boldsymbol{\phi}(s)]_a$

$$W \leftarrow W + \alpha \left[\underbrace{R_{t+1} + \gamma \hat{Q}_W(S_{t+1}) - \hat{Q}_W(S_t)}_{\delta_t} \right] \boldsymbol{\phi}(S_t)$$

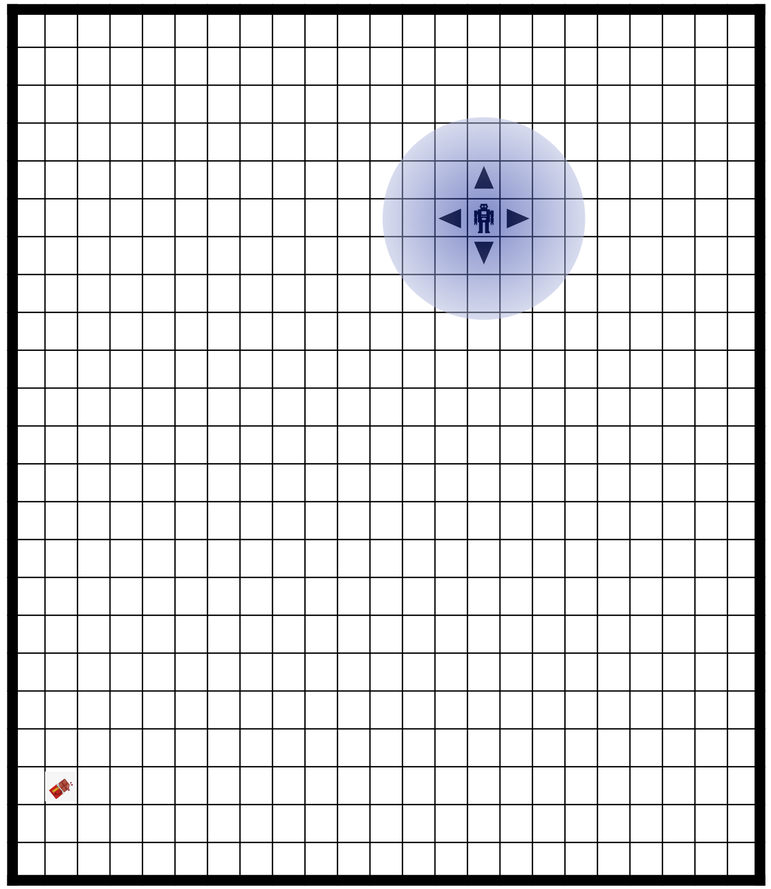


Approximating the value function using state features

- Approximation: $\hat{Q}_\theta(s, a)$

- $\hat{Q}_W(s, a) = [W^\top \boldsymbol{\phi}(s)]_a$

$$W \leftarrow W + \alpha \left[\underbrace{R_{t+1} + \gamma \hat{Q}_W(S_{t+1}) - \hat{Q}_W(S_t)}_{\delta_t} \right] \boldsymbol{\phi}(S_t)$$

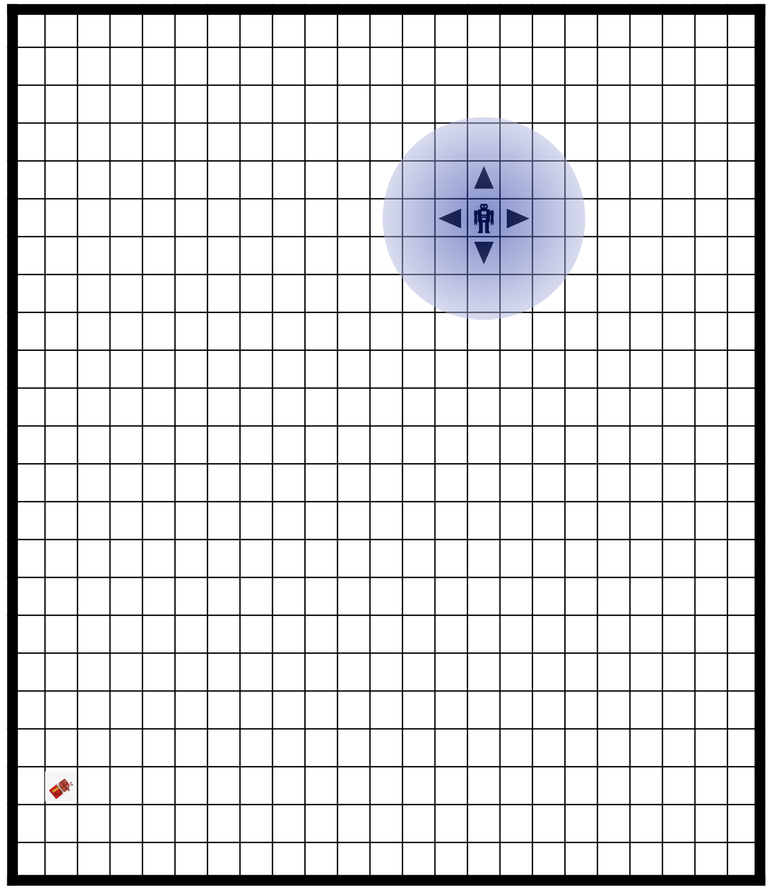


Approximating the value function using state features

- Approximation: $\hat{Q}_\theta(s, a)$

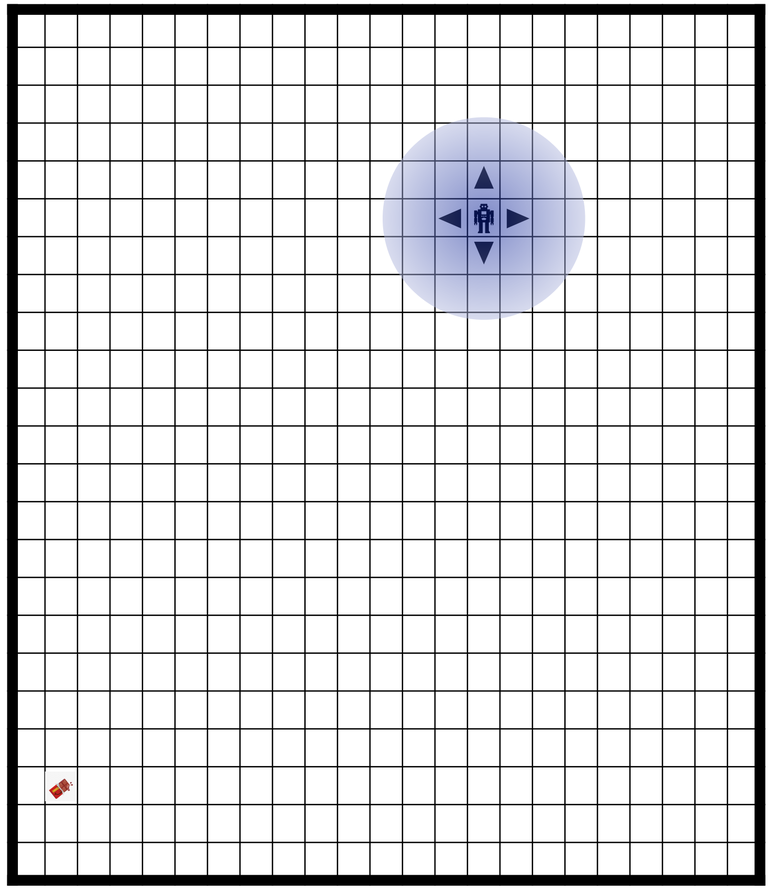
- $\hat{Q}_W(s, a) = [W^\top \boldsymbol{\phi}(s)]_a$

$$W \leftarrow W + \alpha \left[\underbrace{R_{t+1} + \gamma \hat{Q}_W(S_{t+1}) - \hat{Q}_W(S_t)}_{\delta_t} \right] \boldsymbol{\phi}(S_t)$$



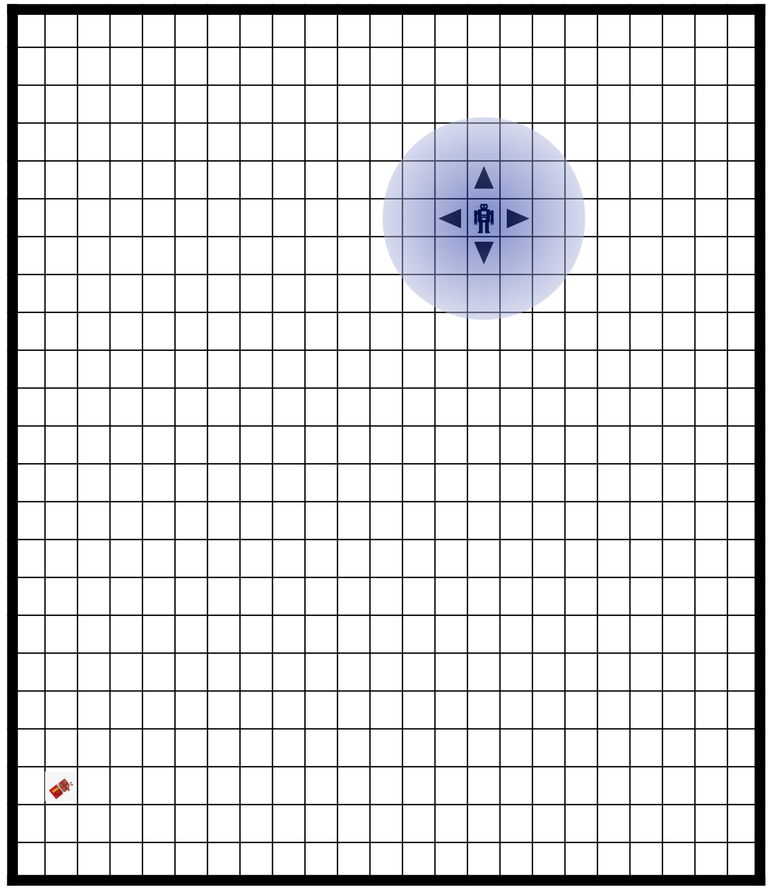
Approximating the value function using state features

- Advantages:
  - Generalisation across nearby states
  - $N_f << N_S$



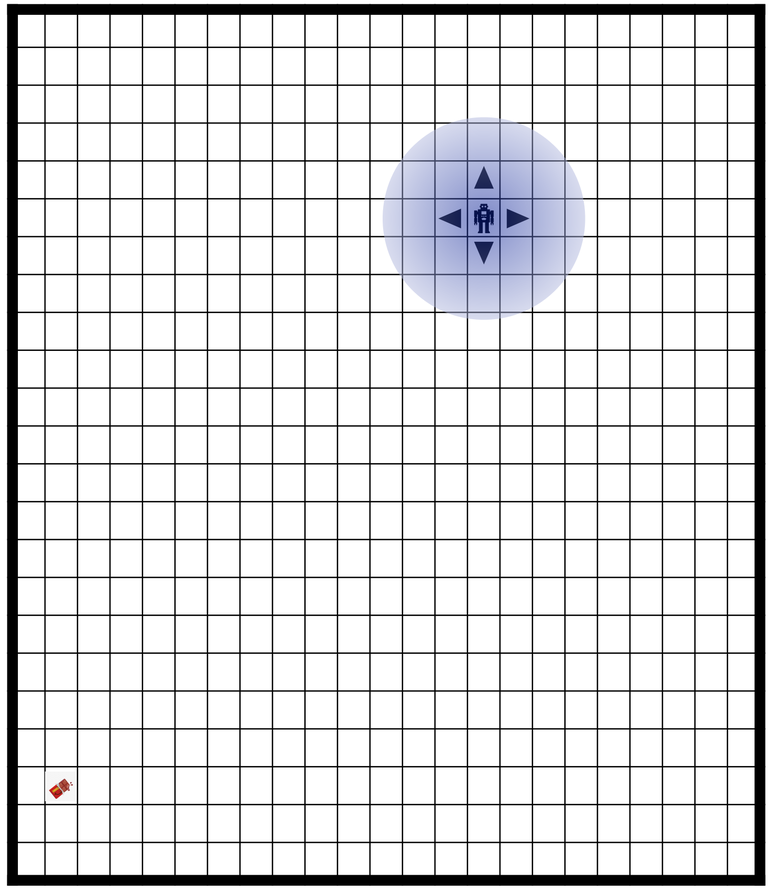
Approximating the value function using state features

- Advantages:
  - Generalisation across nearby states
  - $N_f << N_S$

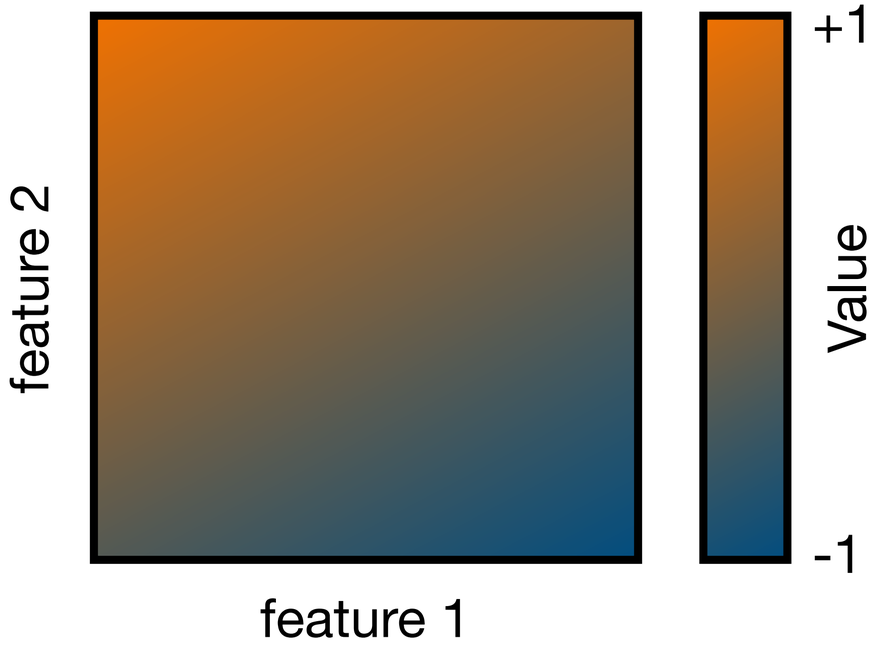


Approximating the value function using state features

- Advantages:
  - Generalisation across nearby states
  - $N_f << N_S$

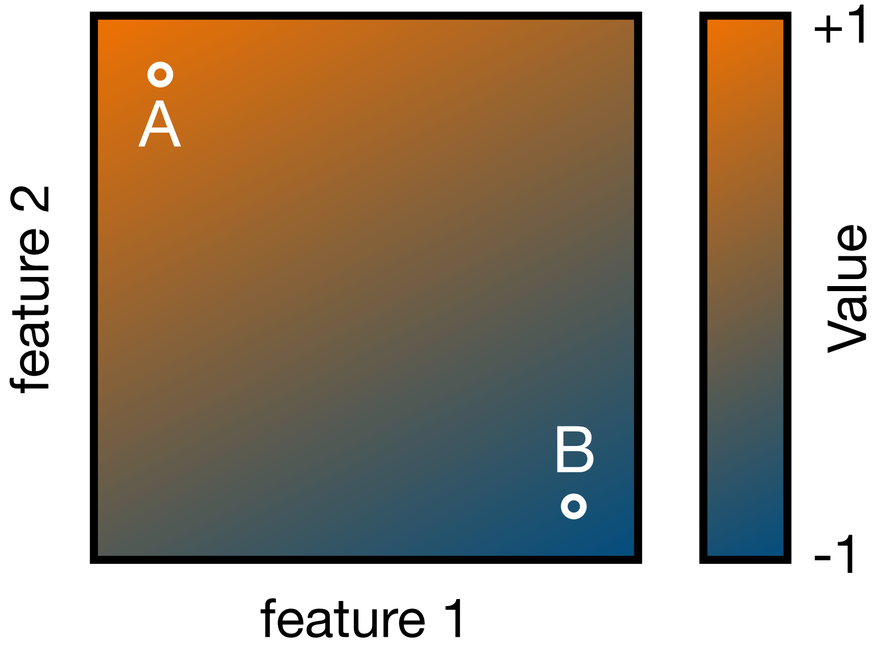


What is a good choice of features for generalization?



What is a good choice of features for generalization?

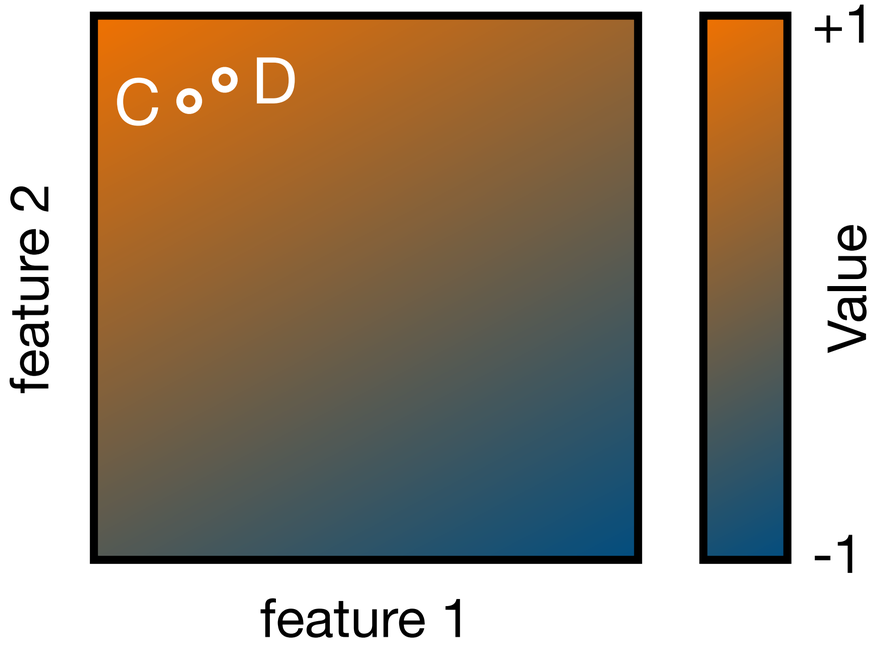
Distant states can have different value



$$V(S_A) \neq V(S_B)$$

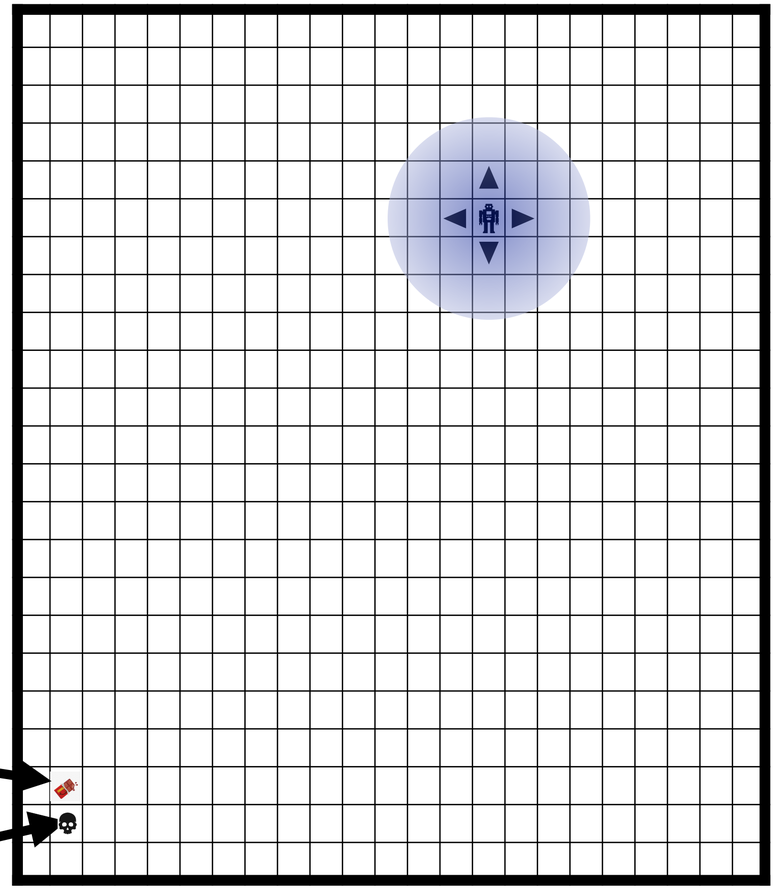
What is a good choice of features for generalization?

Nearby states are constrained to have similar values



$$V(S_C) \approx V(S_D)$$

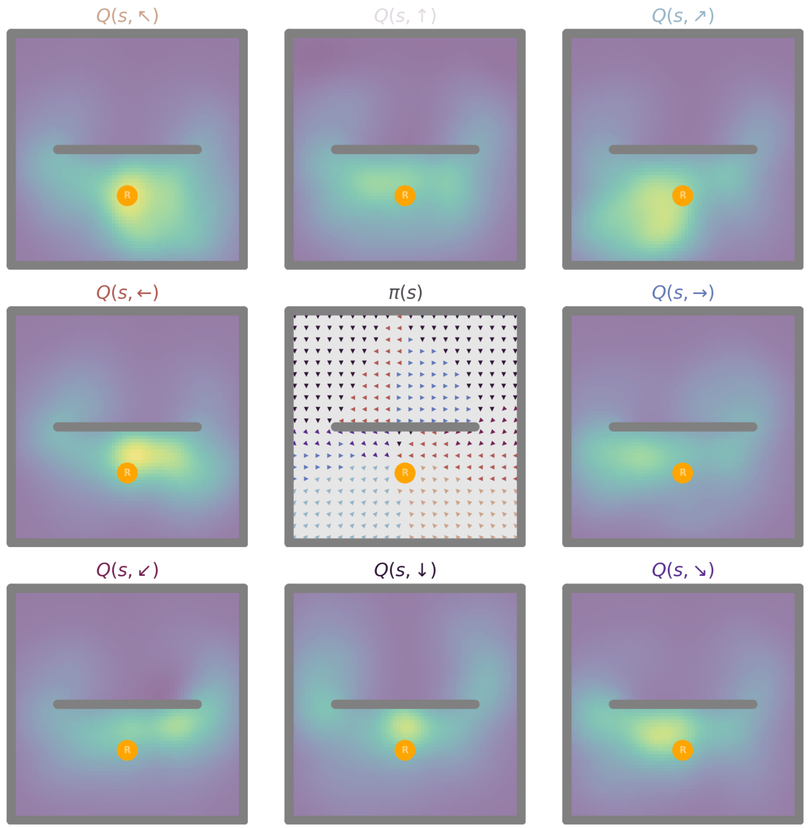
Feature engineering is hard!



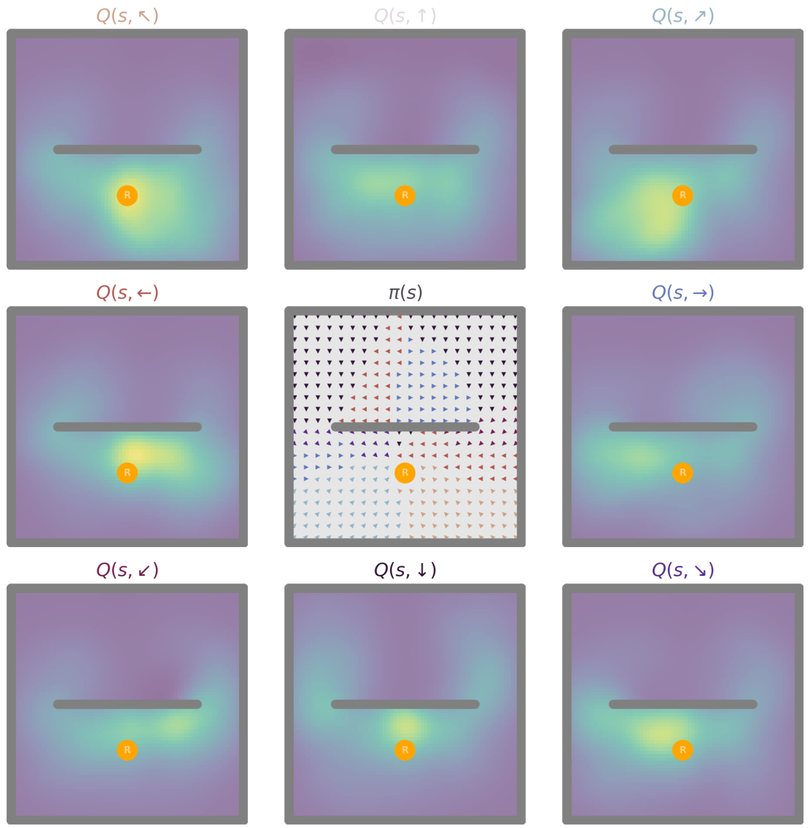
this state is very good

this state is very bad

Tutorial exercise: SARSA in a continuous navigation domain with place cell features



Tutorial exercise: SARSA in a continuous navigation domain with place cell features

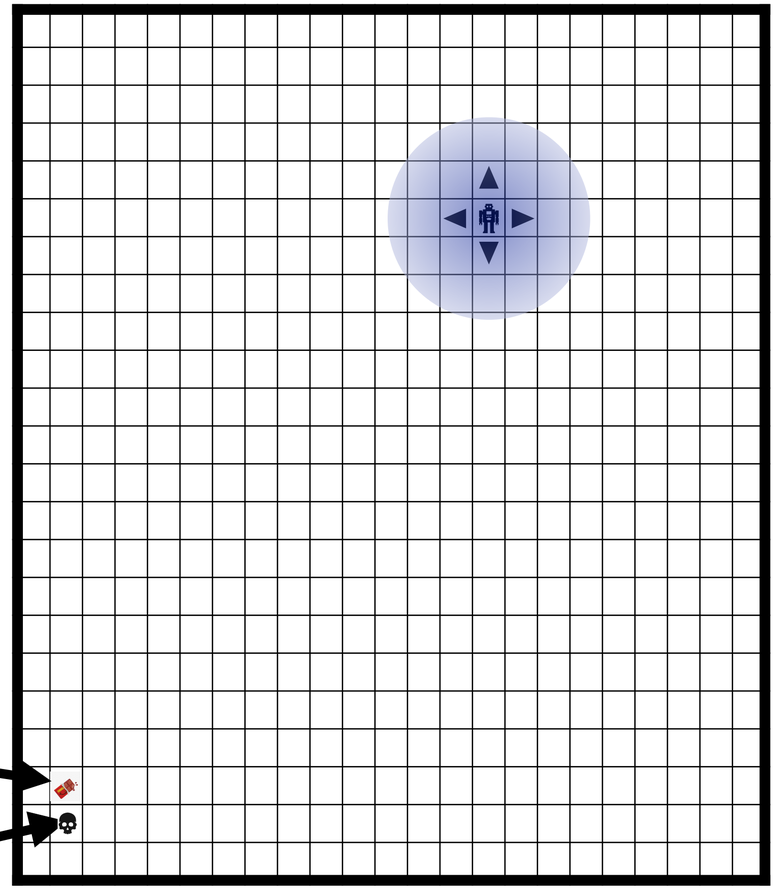


Play for today

- **How do we predict reward?**
  - Classical conditioning and associative learning theory
- **How do we predict *future* reward?**
  - Temporal difference learning and dopamine
- **How do we optimise actions to maximise future reward?**
  - Policy iteration, SARSA
- **What to do in large or continuous environments?**
  - Function approximation and representations of place
- **Representation learning for RL**
  - Learning a predictive representation to support more flexible RL
- **The limits of model-free RL and the case for a cognitive map**
- Afternoon: TUTORIAL

Representation learning for more flexible RL

Feature engineering is hard!



this state is very good

this state is very bad

Predictive representations for RL

$$V^{\pi}(S) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R_{t+1} \,|\, S_0 = S\right]$$

value = sum expected future rewards

Dayan (1993)

Predictive representations for RL

$$V^{\pi}(S) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R_{t+1} \,|\, S_0 = S\right]$$

value = sum expected future rewards

$$V^{\pi}(S) = \sum_{S'} M^{\pi}(S, S') R(S')$$

value = sum expected future state visits · reward at each state

Dayan (1993)

Predictive representations for RL

$$M^{\pi}(S, S') = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \mathbb{I}\left(S_t = S'\right) | S_0 = S\right]$$

successor representation [s, s'] = sum expected future state visits to s' after s

- states that predict similar future states, have similar SR!
- when rewards change, we can quickly recompute value
- SR is dependent on policy

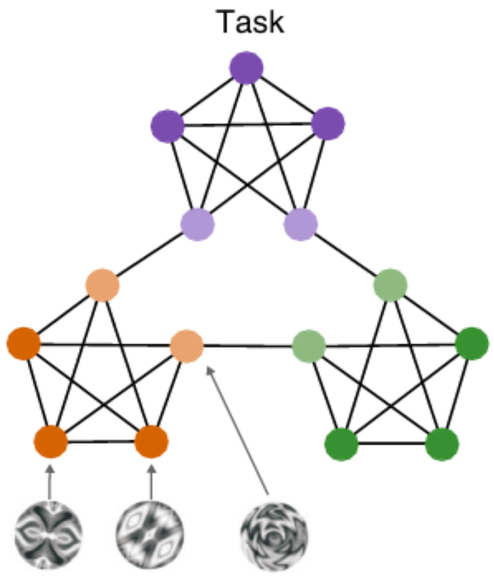
Dayan (1993)

Generalisation to states that predict similar futures

$$M^{\pi}(S, S') = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \mathbb{I}\left(S_t = S'\right) \mid S_0 = S\right]$$

successor representation

Schapiro *et al.* (2015)



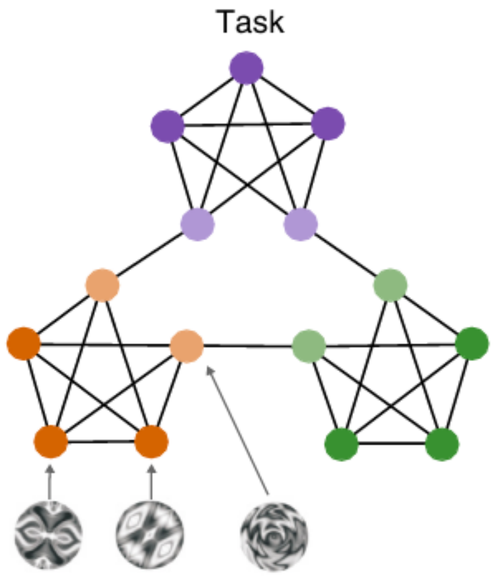
Stachenfeld et al. (2017)

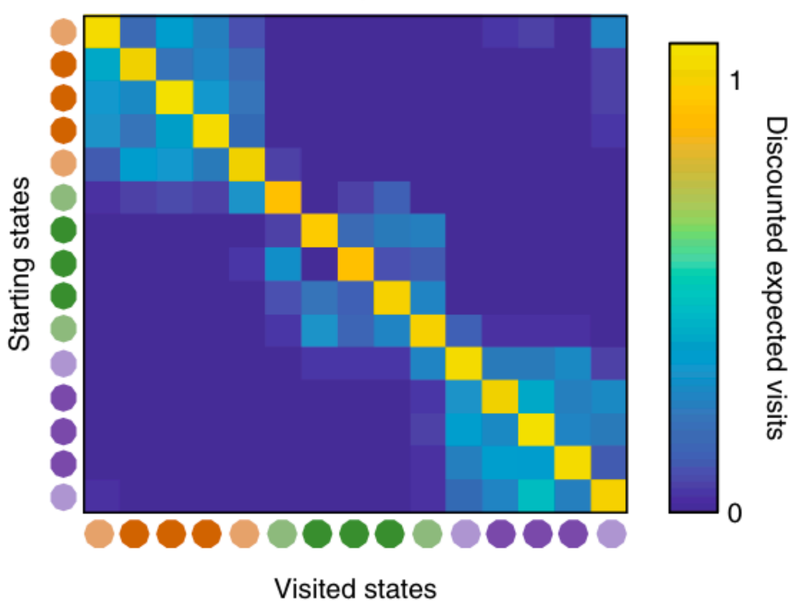
Generalisation to states that predict similar futures

$$M^{\pi}(S, S') = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \mathbb{I}\left(S_t = S'\right) \mid S_0 = S\right]$$

successor representation

Schapiro *et al.* (2015)





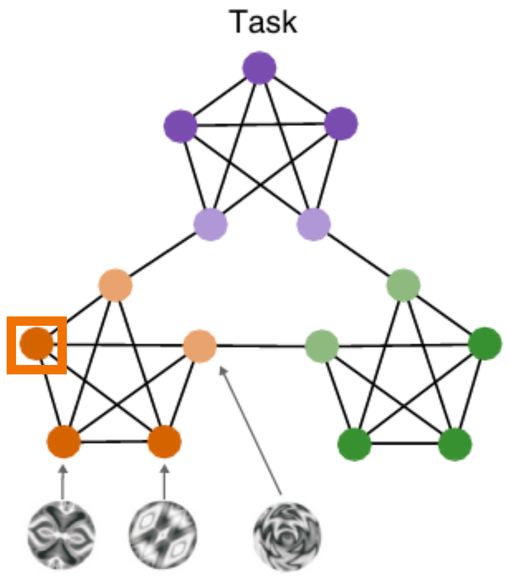
Stachenfeld et al. (2017)

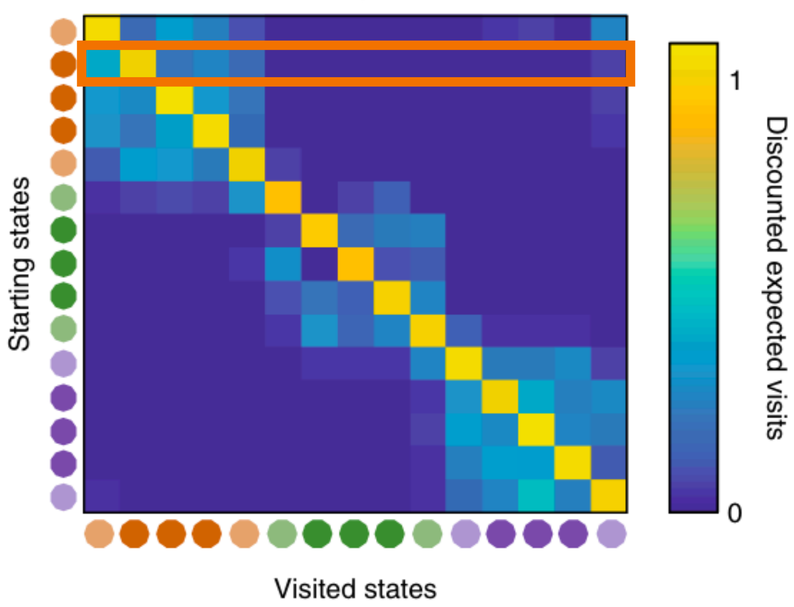
Generalisation to states that predict similar futures

$$M^{\pi}(S, S') = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \mathbb{I}\left(S_t = S'\right) \mid S_0 = S\right]$$

successor representation

Schapiro *et al.* (2015)





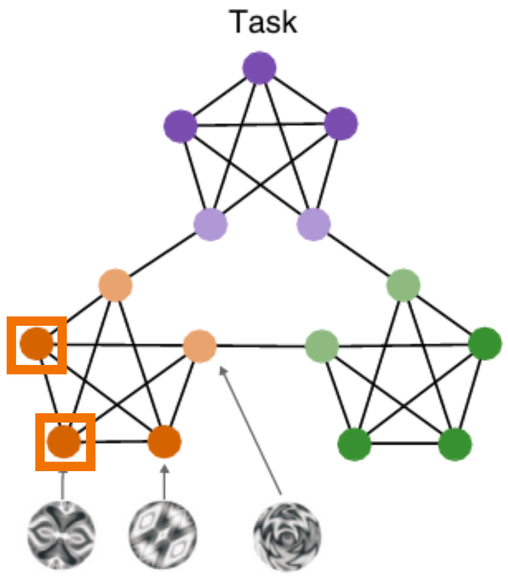
Stachenfeld et al. (2017)

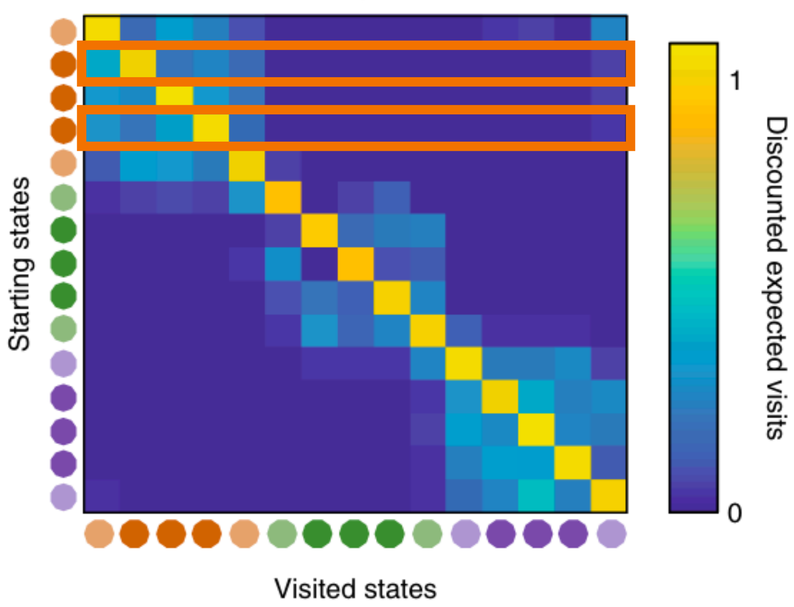
Generalisation to states that predict similar futures

$$M^{\pi}(S, S') = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \mathbb{I}\left(S_t = S'\right) \mid S_0 = S\right]$$

successor representation

Schapiro *et al.* (2015)





Stachenfeld et al. (2017)

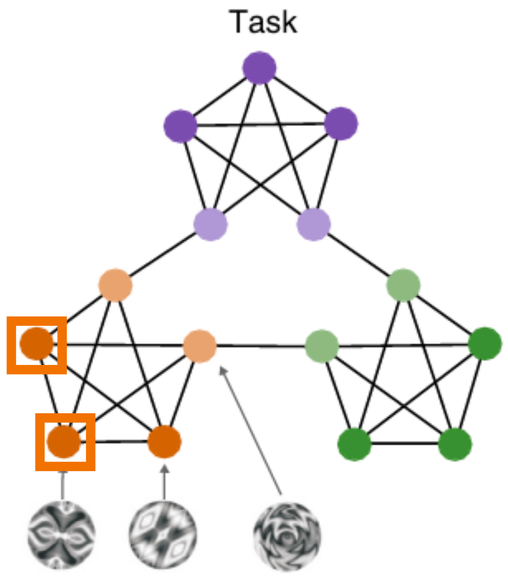
Generalisation to states that predict similar futures

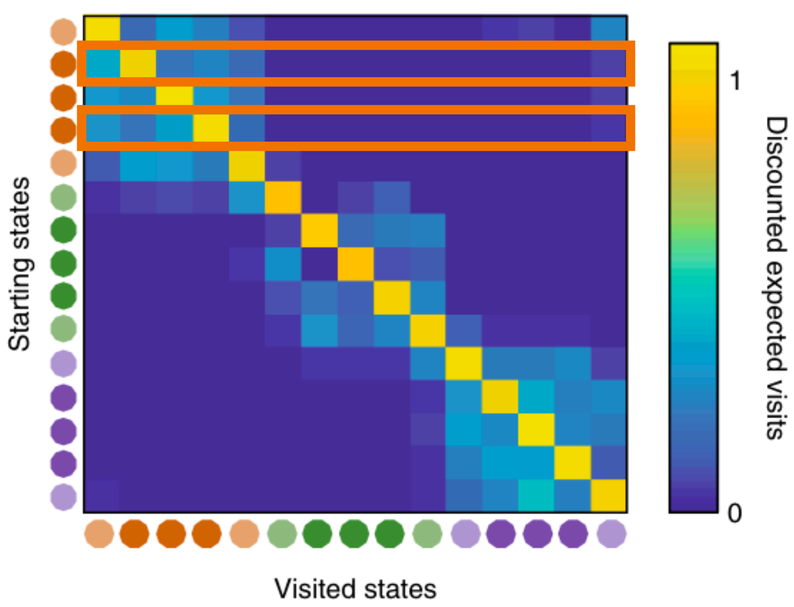
$$M^{\pi}(S, S') = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \mathbb{I}\left(S_t = S'\right) \mid S_0 = S\right]$$

successor representation

similar

Schapiro *et al.* (2015)





Stachenfeld et al. (2017)

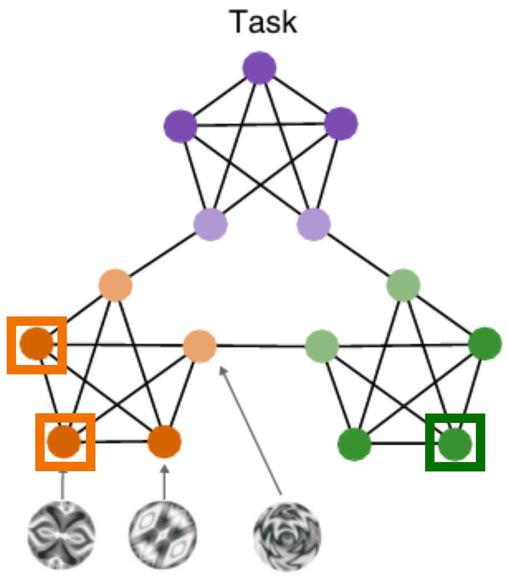
Generalisation to states that predict similar futures

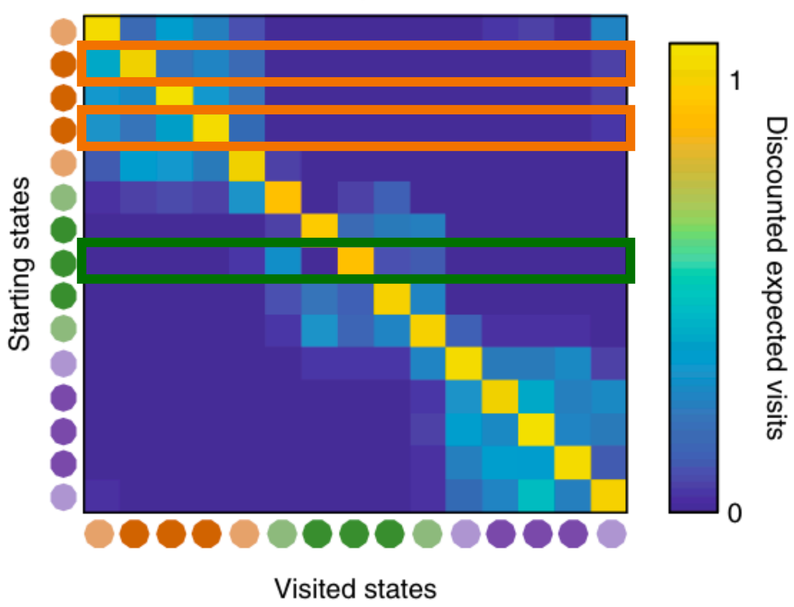
$$M^{\pi}(S, S') = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \mathbb{I}\left(S_t = S'\right) \mid S_0 = S\right]$$

successor representation

similar

Schapiro *et al.* (2015)





Stachenfeld et al. (2017)

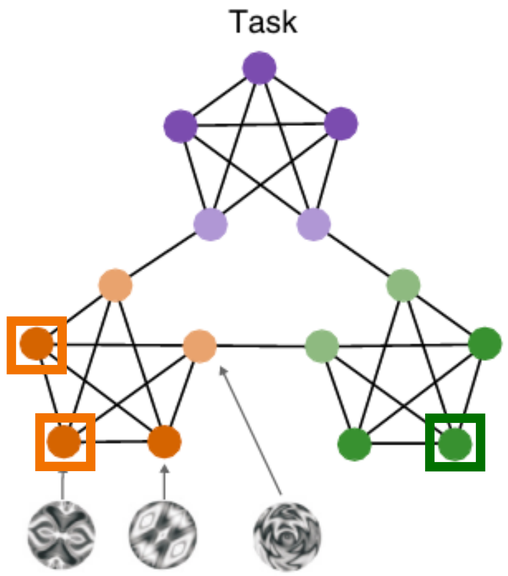
Generalisation to states that predict similar futures

$$M^{\pi}(S, S') = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \mathbb{I}\left(S_t = S'\right) | S_0 = S\right]$$

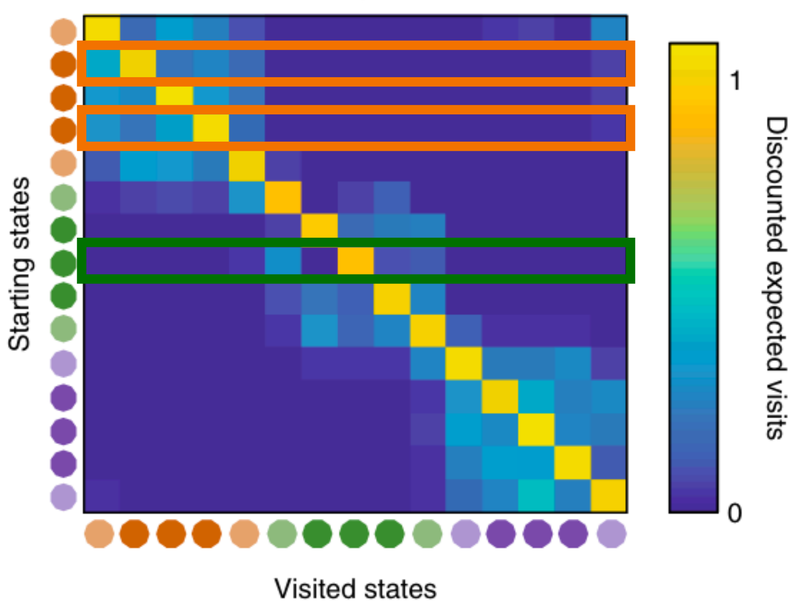
successor representation

similar

Schapiro *et al.* (2015)



Not so similar

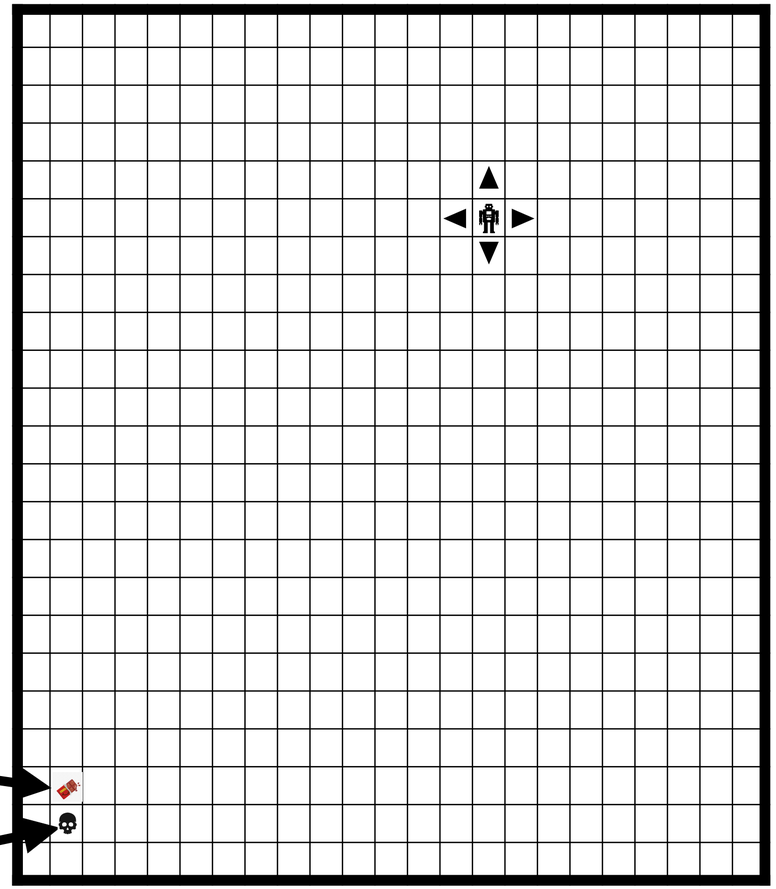


Stachenfeld et al. (2017)

We can learn an SR that does not generalise between these states

$M(S_t, S_{\text{good}}) = \text{high}$

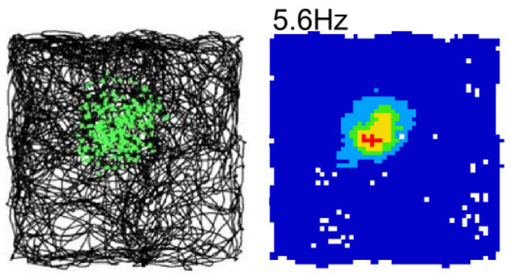
$M(S_t, S_{\text{bad}}) = \text{low}$

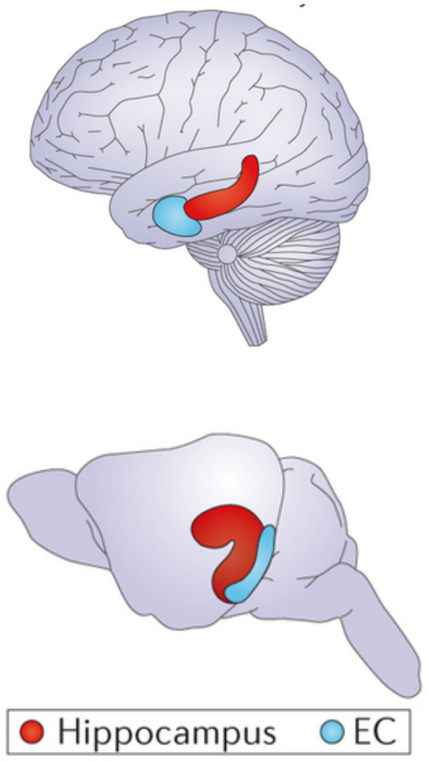


this state is very good

this state is very bad

Place cells

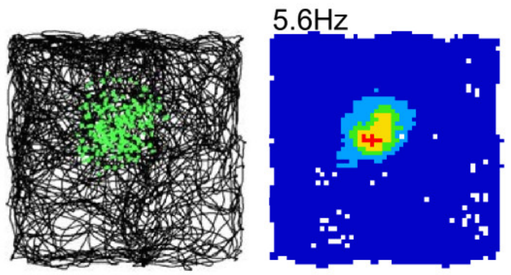


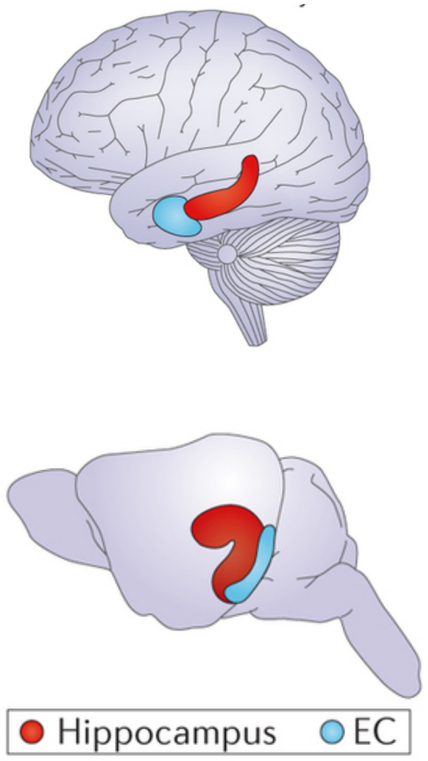


O'Keefe & Dostrovsky (1971)

Place cells

- Found in hippocampus
- Firing is dependent on location
  - Place field centre has highest firing
  - Some firing when animal is close to the centre

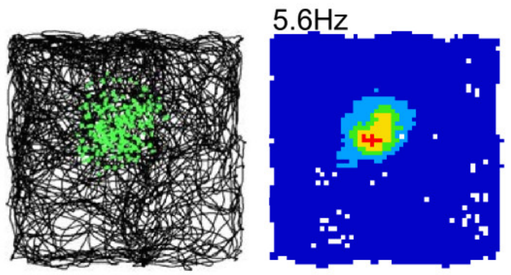


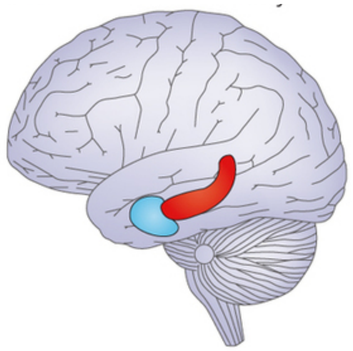


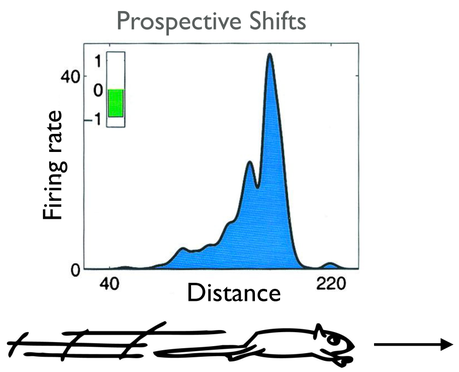
O'Keefe & Dostrovsky (1971)

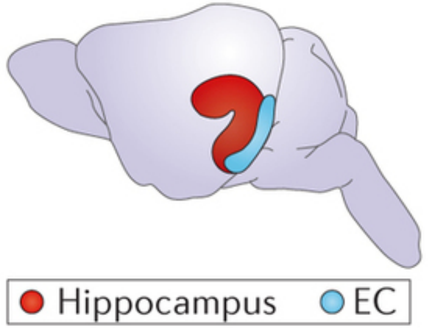
Place cells

- Found in hippocampus
- Firing is dependent on location
  - Place field centre has highest firing
  - Some firing when animal is close to the centre
- Place cells show “prospective” shifts depending on animal behaviour



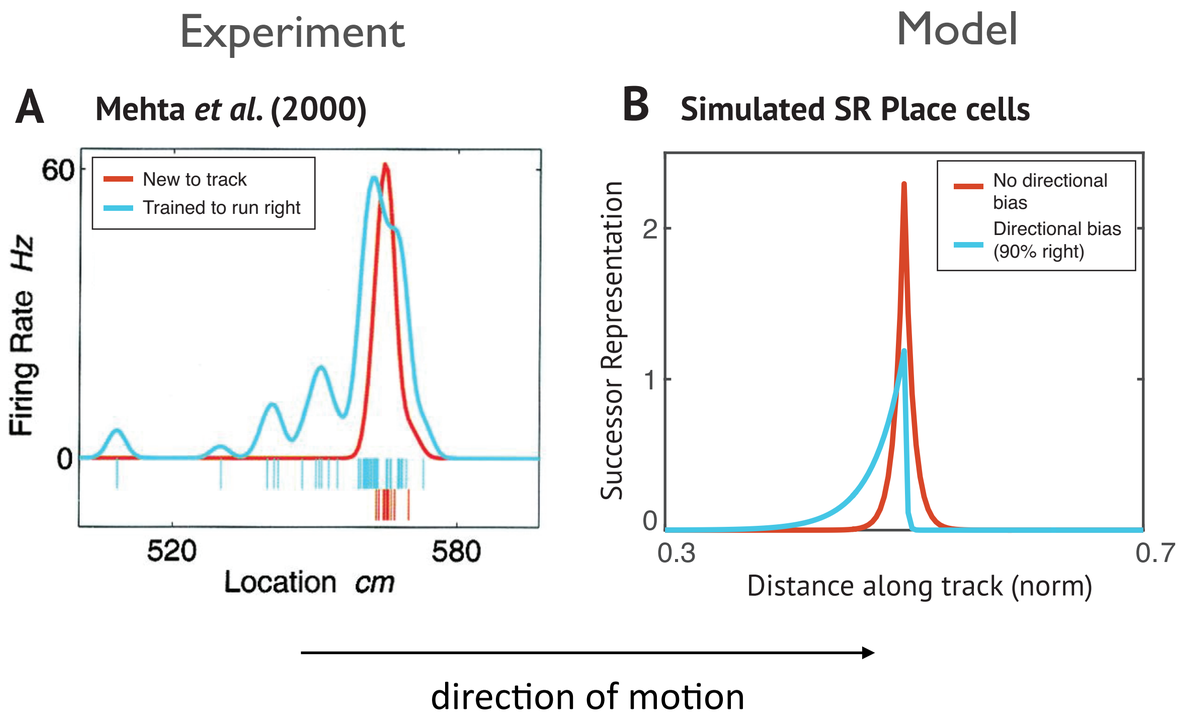






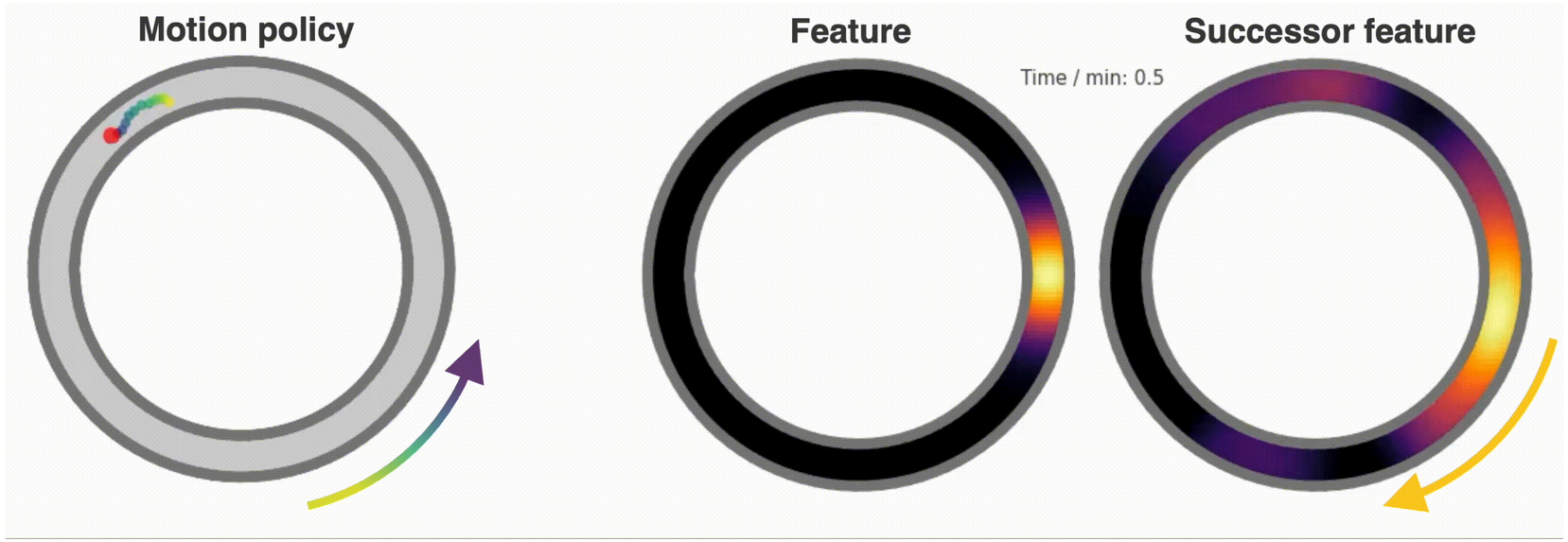
O’Keefe & Dostrovsky (1971)

Place cells and SR show a policy-dependent skew

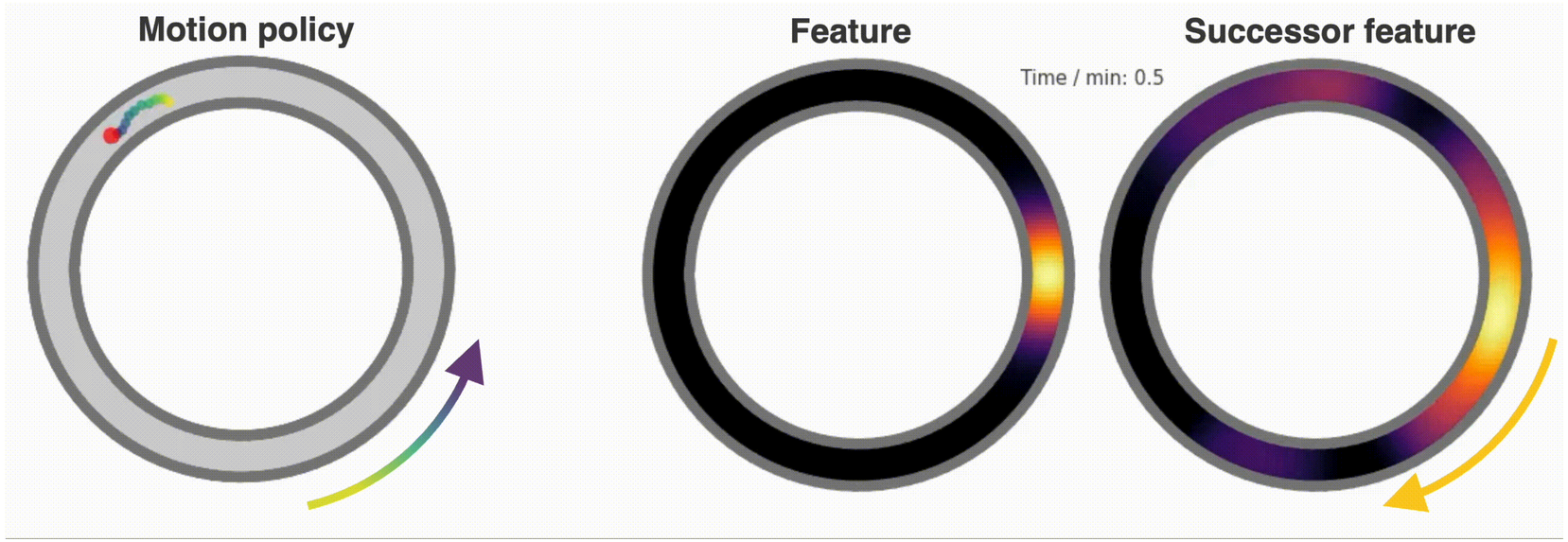


Stachenfeld et al. (2017)

The successor representation is *predictive* and *policy-dependent*



The successor representation is *predictive* and *policy-dependent*



Interim summary

Interim summary

- The SR is a *predictive* representation and a *model of CA1 place cells*

Interim summary

- The SR is a *predictive* representation and a *model of CA1 place cells*
- It generalises value to states that predict similar futures

Interim summary

- The SR is a *predictive* representation and a *model of CA1 place cells*
- It generalises value to states that predict similar futures
- When the reward function changes (but nothing else) it allows quick revaluation

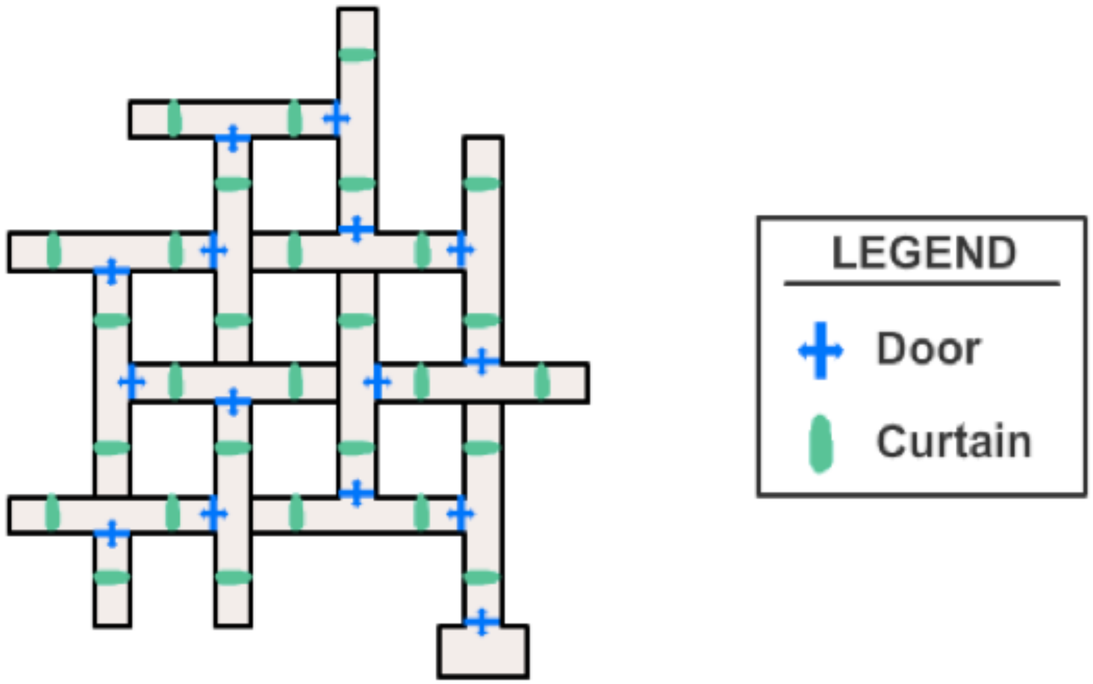
Interim summary

- The SR is a *predictive* representation and a *model of CA1 place cells*
- It generalises value to states that predict similar futures
- When the reward function changes (but nothing else) it allows quick revaluation
- When policy or transitions change, the SR must change too

Plan for today

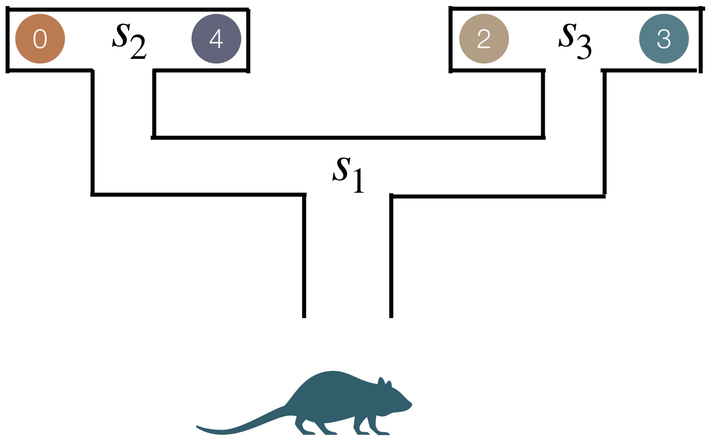
- **How do we predict reward?**
  - Classical conditioning and associative learning theory
- **How do we predict *future* reward?**
  - Temporal difference learning and dopamine
- **How do we optimise actions to maximise future reward?**
  - Policy iteration, SARSA
- **What to do in large or continuous environments?**
  - Function approximation and representations of place
- **Representation learning for RL**
  - Learning a predictive representation to support more flexible RL
- **The limits of model-free RL and the case for a cognitive map**
- Afternoon: TUTORIAL

Model-free learning methods do not explain shortcut behaviours

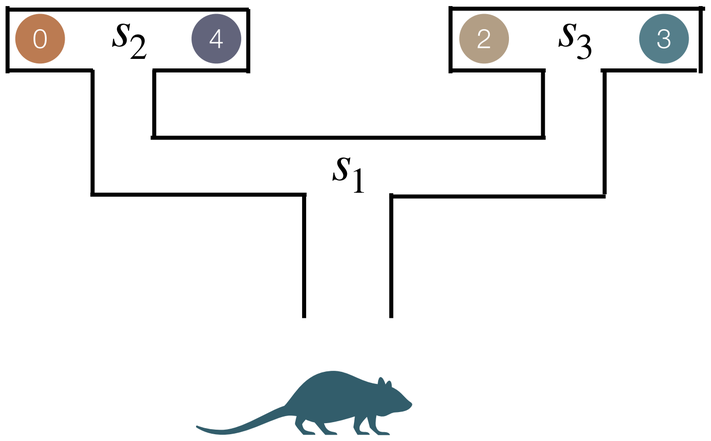


Tolman (1948)

How to solve the detour maze?



How to solve the detour maze?

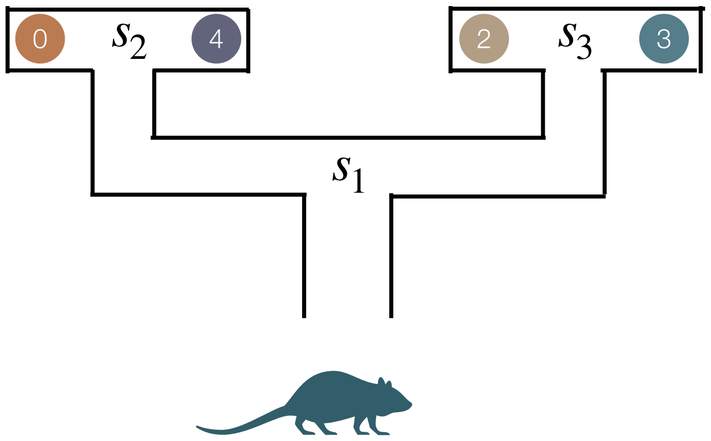


$Q(s, a)$

	L	R
S_1	4	3
S_2	0	4
S_3	2	3

Model-free

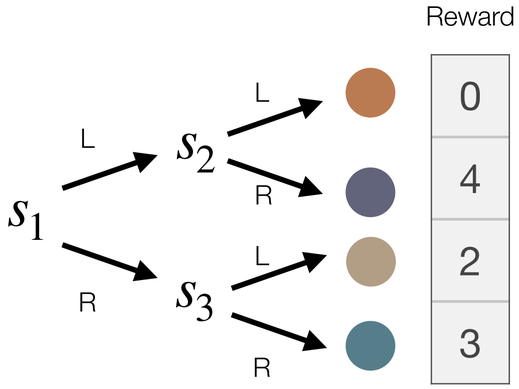
How to solve the detour maze?



$Q(s, a)$

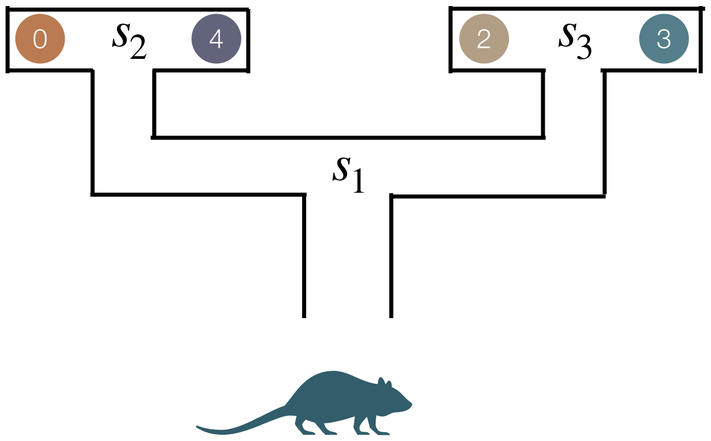
	L	R
S_1	4	3
S_2	0	4
S_3	2	3

Model-free



Model-based

How to solve the detour maze?

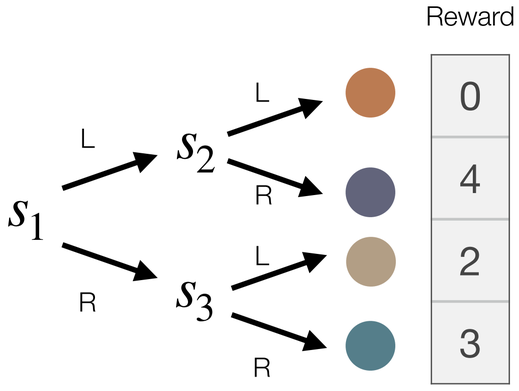


Fast
Inflexible

$Q(s, a)$

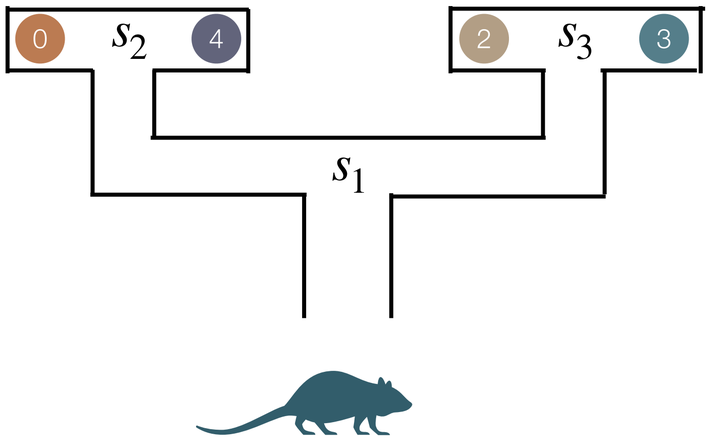
	L	R
S_1	4	3
S_2	0	4
S_3	2	3

Model-free



Model-based

How to solve the detour maze?

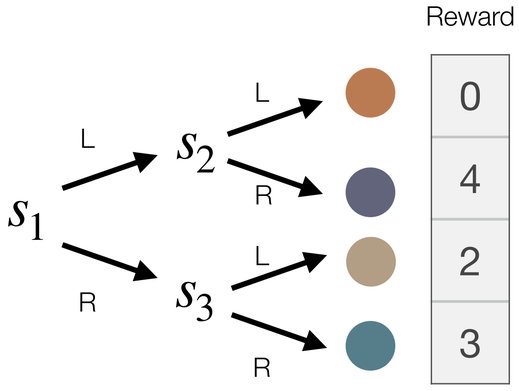


Fast
Inflexible

$Q(s, a)$

	L	R
S_1	4	3
S_2	0	4
S_3	2	3

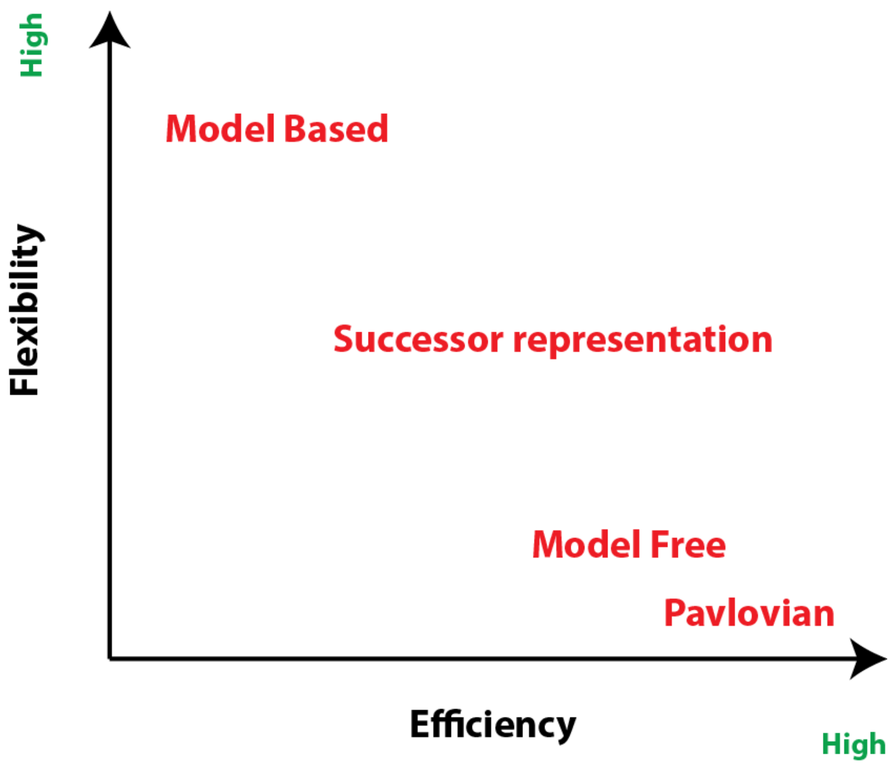
Model-free



Model-based

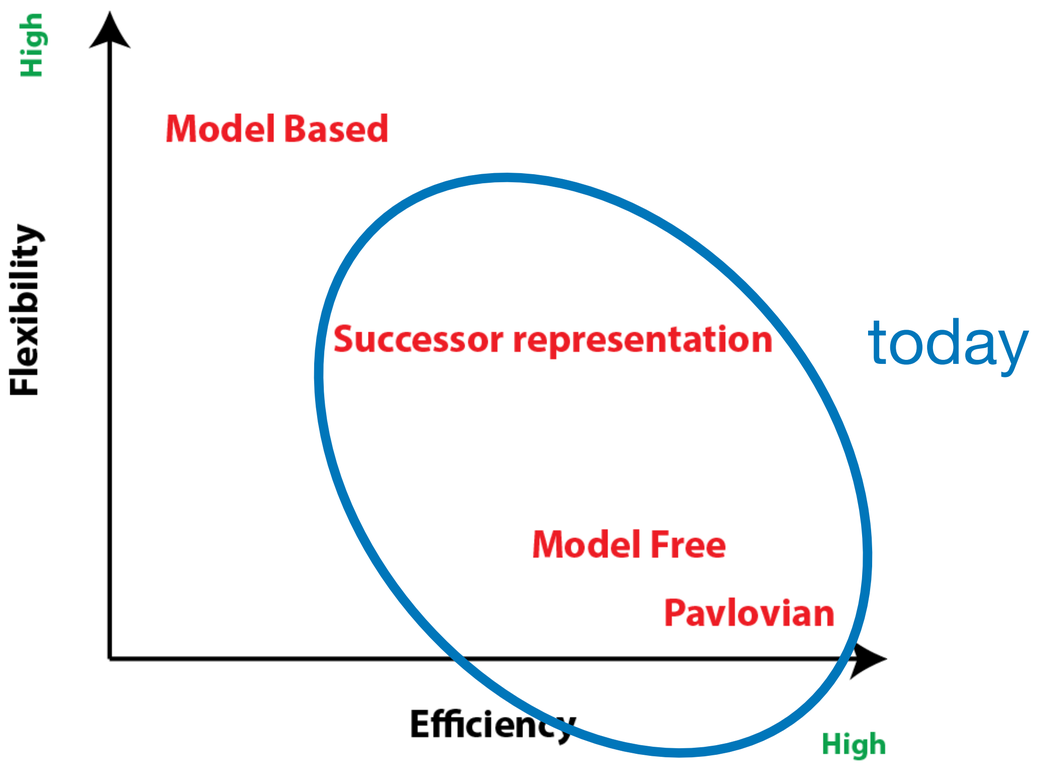
Slow
Flexible

Model-free and model-based trade off efficiency and flexibility



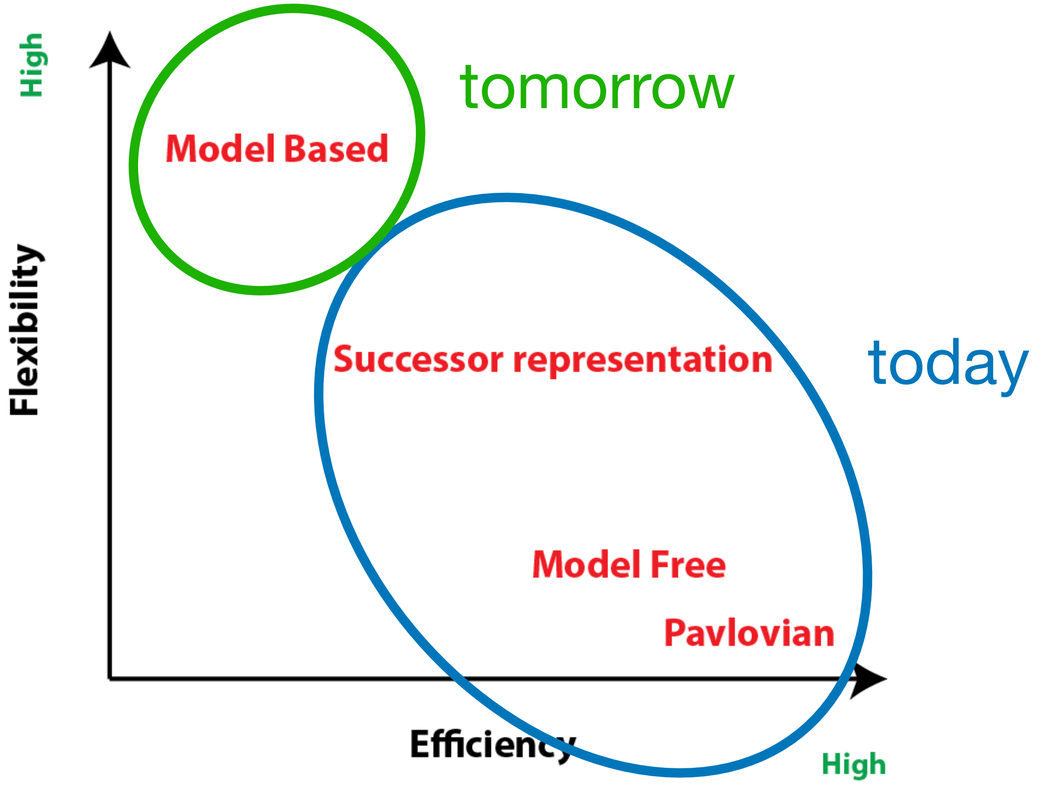
Gershman (2018)

Model-free and model-based trade off efficiency and flexibility



Gershman (2018)

Model-free and model-based trade off efficiency and flexibility



Gershman (2018)

Plan for today

- **How do we predict reward?**
  - Classical conditioning and associative learning theory
- **How do we predict *future* reward?**
  - Temporal difference learning and dopamine
- **How do we optimise actions to maximise future reward?**
  - Policy iteration, SARSA
- **What to do in large or continuous environments?**
  - Function approximation and representations of place
- **Representation learning for RL**
  - Learning a predictive representation to support more flexible RL
- **The limits of model-free RL and the case for a cognitive map**
- Afternoon: TUTORIAL

Summary

Summary

- Rescorla-Wagner predicts animal behaviour at the trial level

Summary

- Rescorla-Wagner predicts animal behaviour at the trial level
- Model-free TD learning is a good model for sequential decision making

Summary

- Rescorla-Wagner predicts animal behaviour at the trial level
- Model-free TD learning is a good model for sequential decision making
- Model-based learning is more flexible but harder and less well understood

Summary

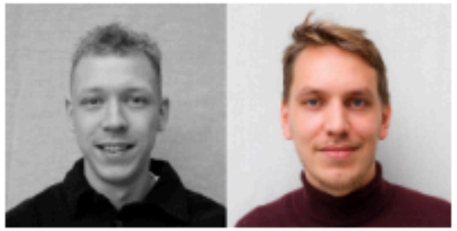
- Rescorla-Wagner predicts animal behaviour at the trial level
- Model-free TD learning is a good model for sequential decision making
- Model-based learning is more flexible but harder and less well understood
  - There are alternative strategies that lie in between these two extremes, such as the SR

Summary

- Rescorla-Wagner predicts animal behaviour at the trial level
- Model-free TD learning is a good model for sequential decision making
- Model-based learning is more flexible but harder and less well understood
  - There are alternative strategies that lie in between these two extremes, such as the SR
- For neuroscientists, RL gives a formal (normative) framework to model autonomous learning from neurons to behaviour

github.com/jessegeerts/NeuroRLTutorial

made by: **Tom George (UCL) and Jesse Geerts (Imperial)**



In this tutorial we'll study and build **reinforcement learning models inspired by the brain**. By the end you'll be able to construct a series of simple but surpringly powerful model of how agents and animals learn to navigate and make decisions in complex environments.

Further reading

- **Courses**
  - DeepMind x UCL RL lecture series on YouTube
  - NeuroMatch module on Reinforcement Learning
- **Textbooks**
  - Sutton & Barto: Reinforcement learning, an introduction