Mount Sinai Center for Computational Psychiatry

# Reinforcement learning as a model of cognitive dynamics in healthy aging

Angela Radulescu, Ph.D.
Pronouns: they/them

Assistant Professor
Department of Psychiatry
Mt. Sinai Center for Computational Psychiatry

# Roadmap

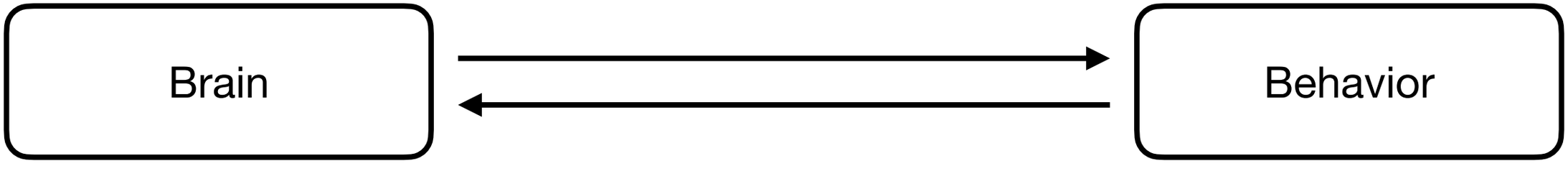
I. RL background

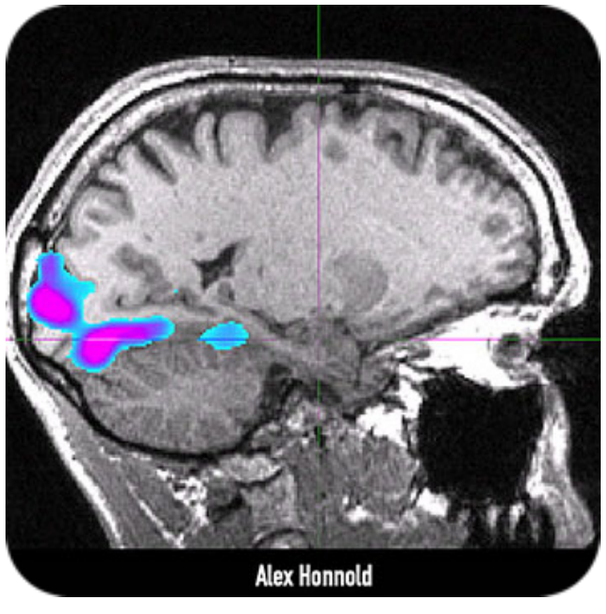
II. RL perspectives on aging

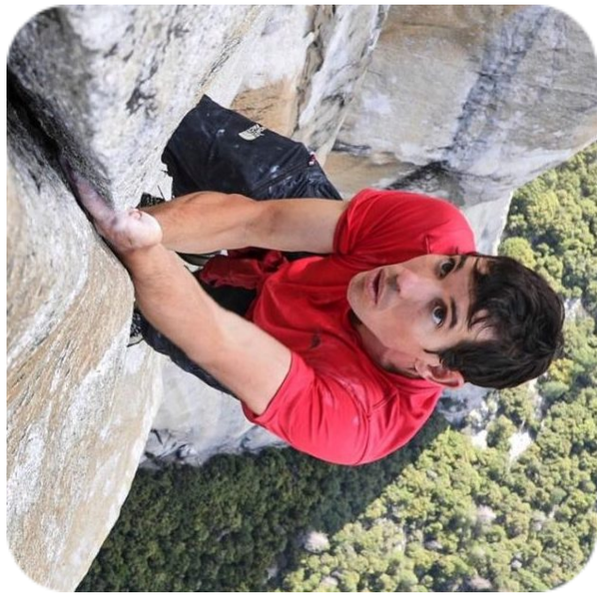
## I. RL background

## II. RL perspectives on aging

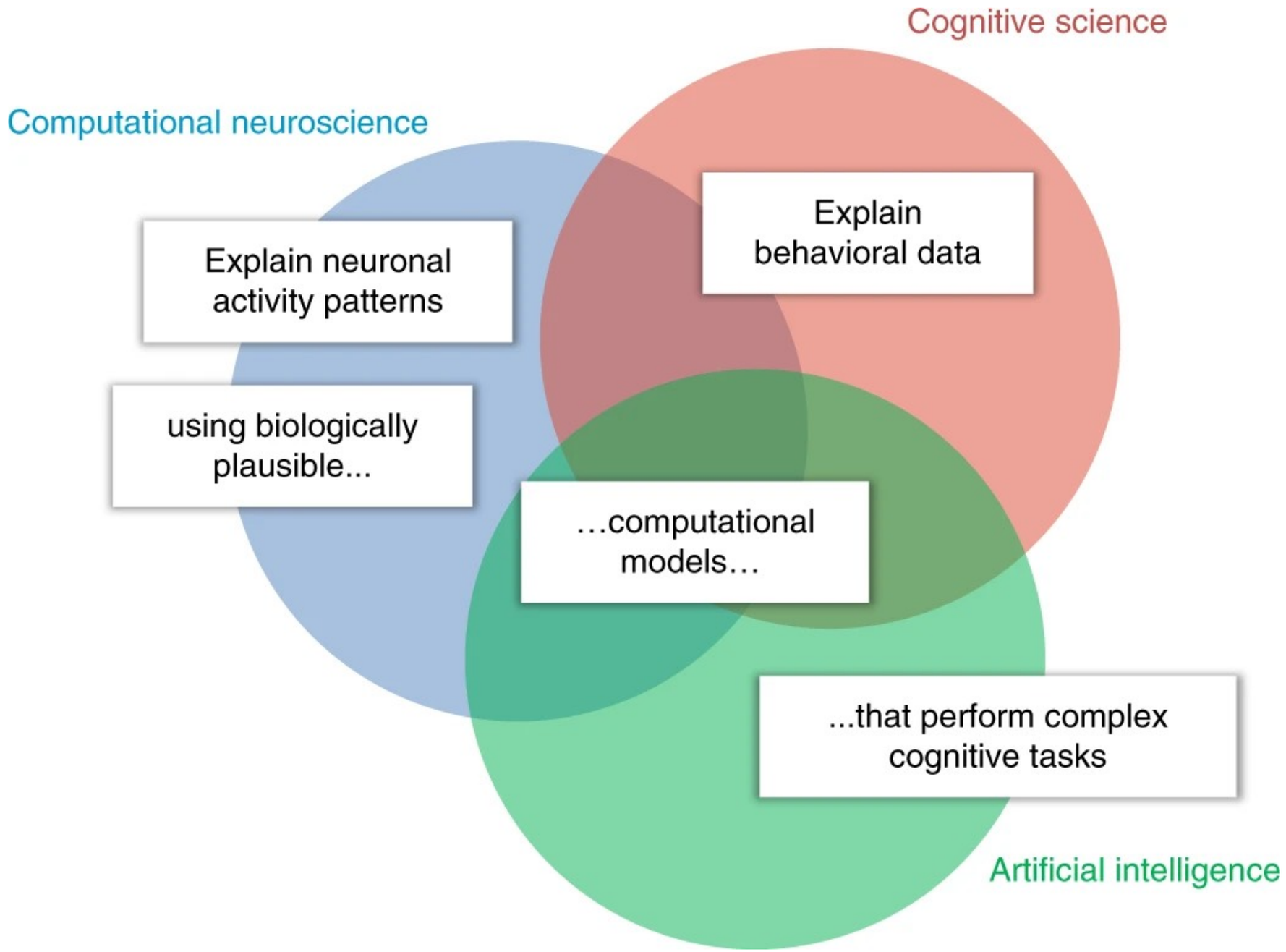
# How does the brain generate behavior?



# Computational models bridge brain, mind and behavior




Kriegeskorte and Douglas, 2018

**Computational cognitive modeling** has led to recent advances in understanding of processes such as learning, memory, and attention.

These processes **change** with healthy aging.

Models allow us to **quantify** cognitive and affective processes

…and to track their **dynamics** over time.

# Current approaches

## Computational

Why do things work the way they do?
What is the goal of the computation?
What are the unifying principles?

## Algorthmic

What representations can implement such computations?
How does the choice of representations determine the algorithm?

## Implementational

How can such a system be built in hardware?
How can neurons carry out the computations?

# Current approaches

## Computational

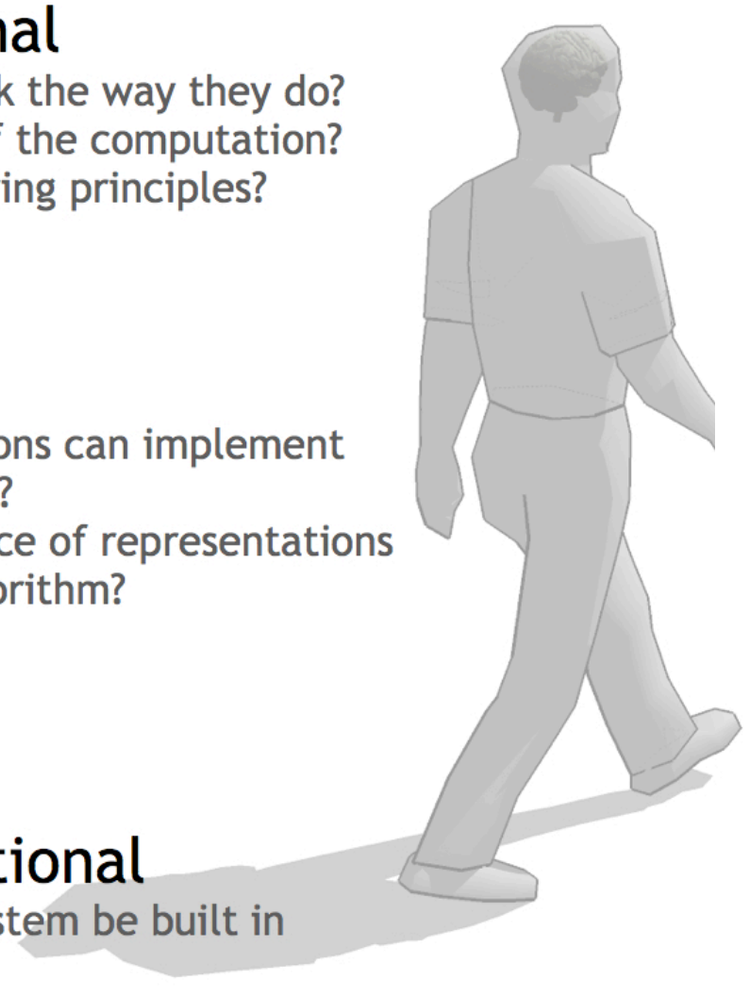
Why do things work the way they do?
What is the goal of the computation?
What are the unifying principles?

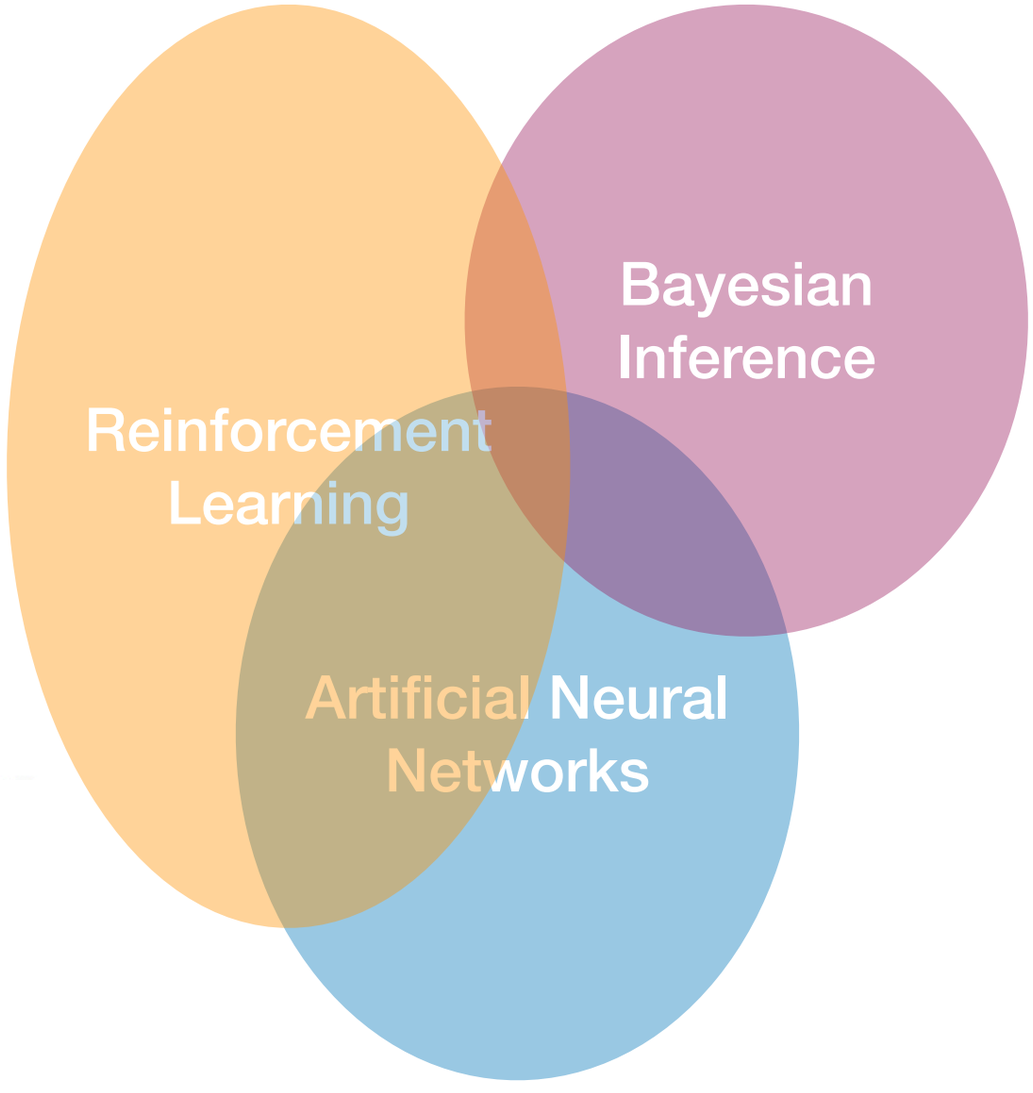
## Algorthmic

What representations can implement such computations?
How does the choice of representations determine the algorithm?

## Implementational

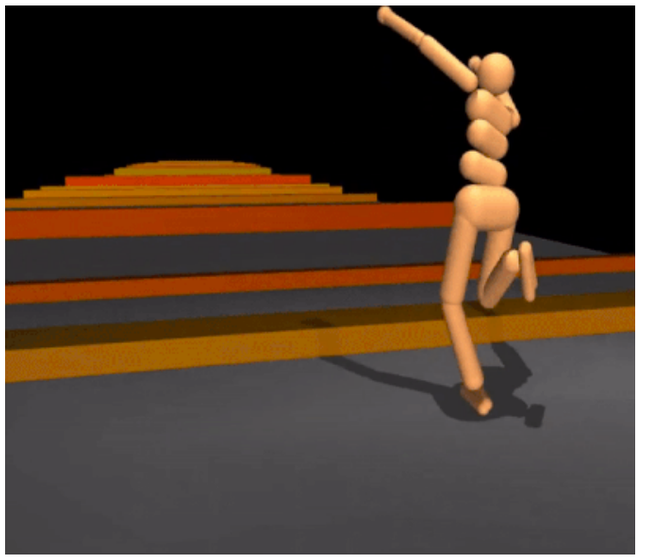
How can such a system be built in hardware?
How can neurons carry out the computations?





# Reinforcement learning as a generative model of behavior

RL: learning to make decisions over time to achieve a goal, given sparse, delayed and uncertain feedback.

# Markov Decision Processes (MDPs)

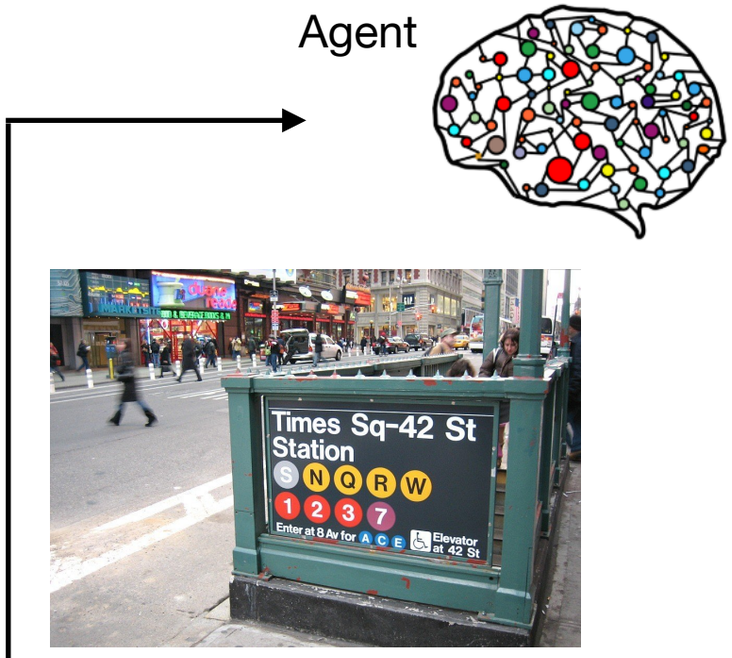
Agent

# Markov Decision Processes (MDPs)

Agent

Environment

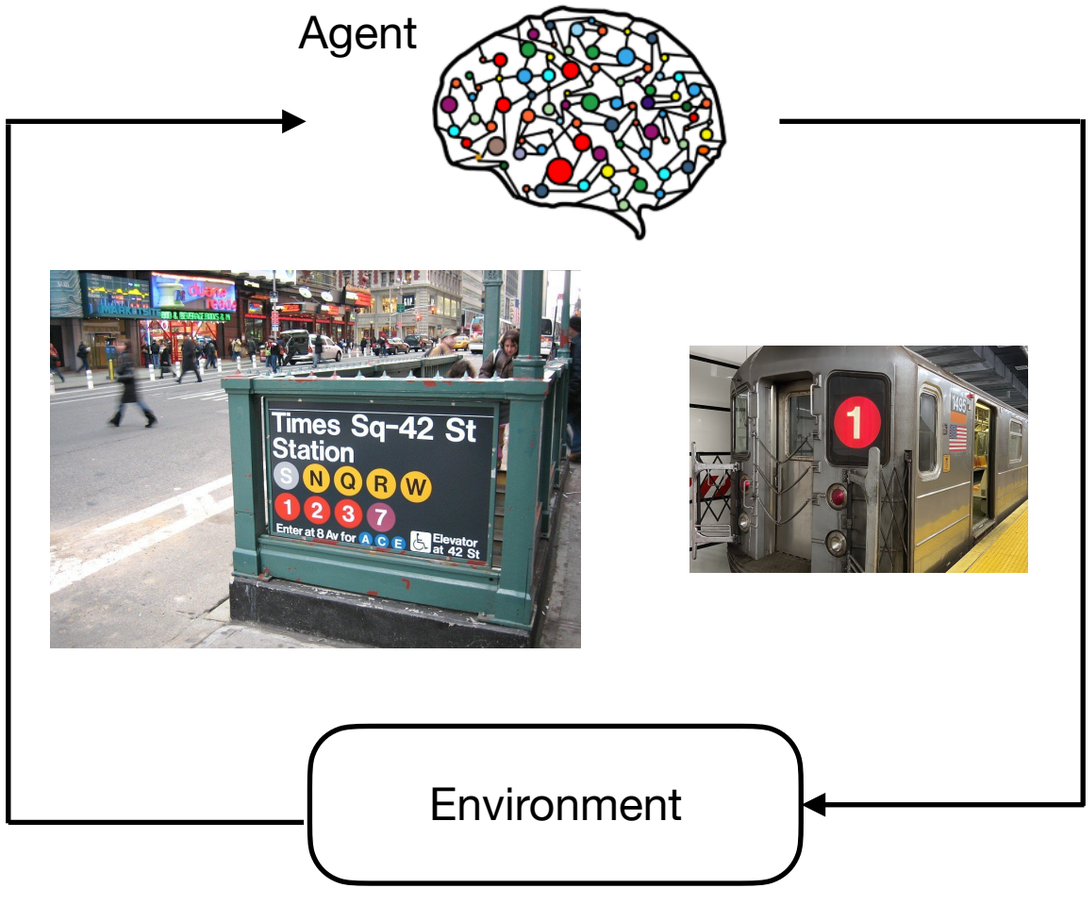
# Markov Decision Processes (MDPs)




state $s_t$

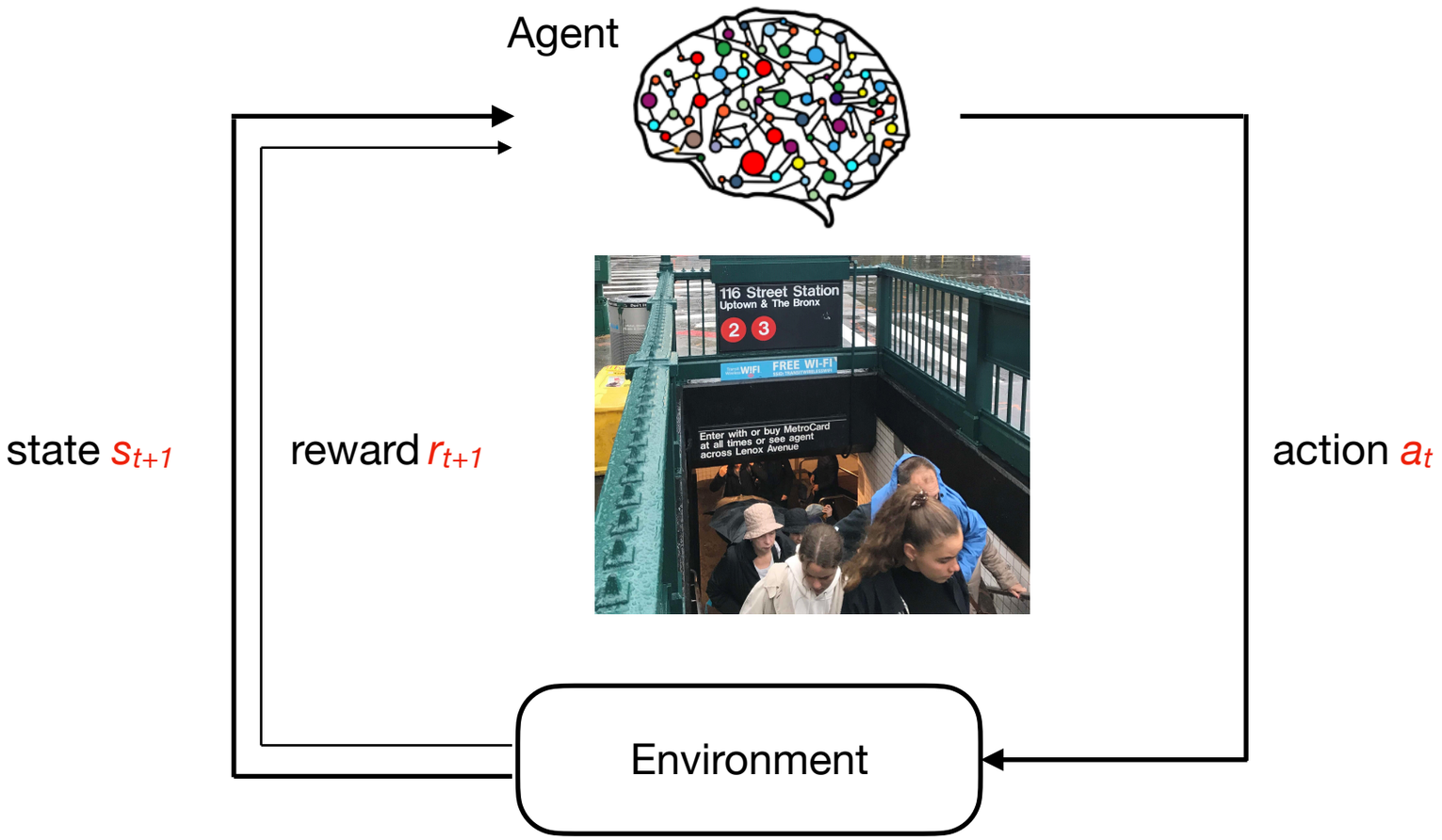
- $s_t$ - what features of the environment will change how much reward I get?

# Markov Decision Processes (MDPs)



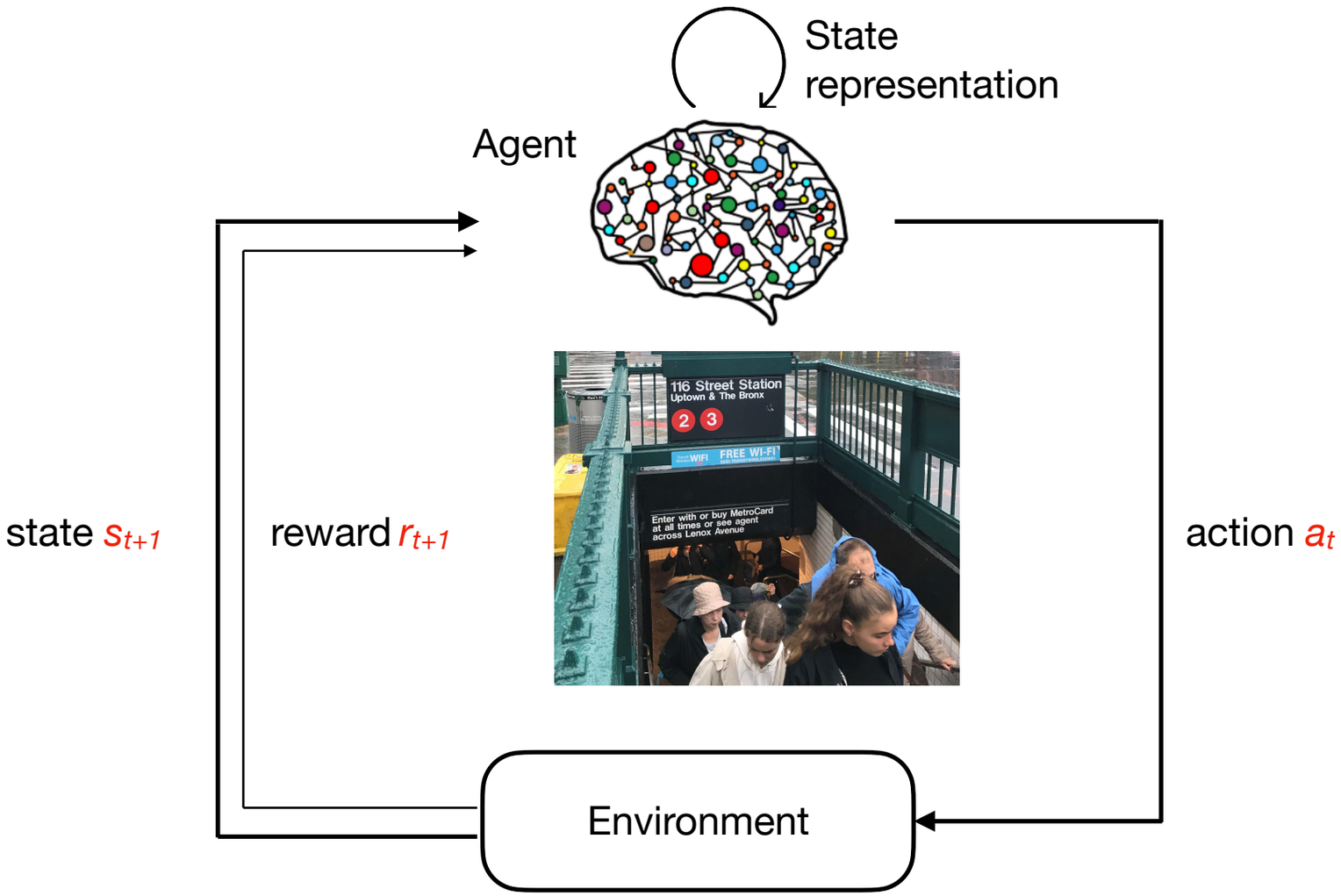

- $s_t$ - what features of the environment will change how much reward I get?
- $a_t$ - what do I choose to do?

# Markov Decision Processes (MDPs)



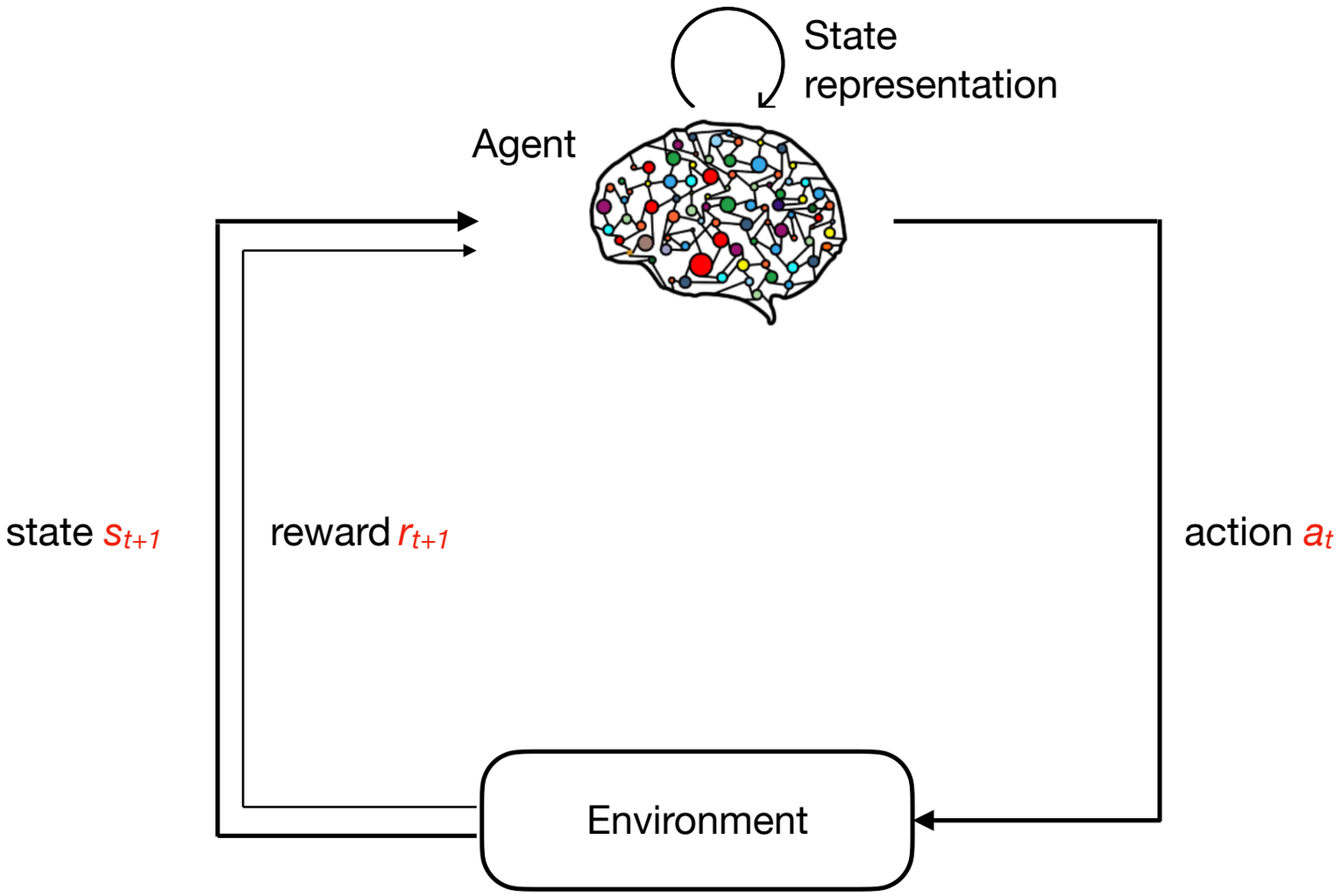

- $s_t$ - what features of the environment will change how much reward I get?
- $a_t$ - what do I choose to do?
- $r_{t+1}$ - how good was my action?
- $s_{t+1}$ - what changed in the environment after my action?

# Markov Decision Processes (MDPs)




- $s_t$ - what features of the environment will change how much reward I get?
- $a_t$ - what do I choose to do?
- $r_{t+1}$ - how good was my action?
- $s_{t+1}$ - what changed in the environment after my action?

# Markov Decision Processes (MDPs)




- $s_t$ is in finite set of states $S$
- $a_t$ is in finite set of actions $A$
- $T$ is a state transition function
- $R$ is a reward function
- $\gamma \in [0,1]$ is a discount factor

A *Markov Decision Process* is defined as a tuple $\langle S, A, T, R, \gamma \rangle$.

Sample state-action-reward sequence (experience):

$s_0, a_0, r_1, s_1, a_1, r_2, s_2, a_2, r_3, \ldots$

Problem specification

Lever pressing

Caching nuts

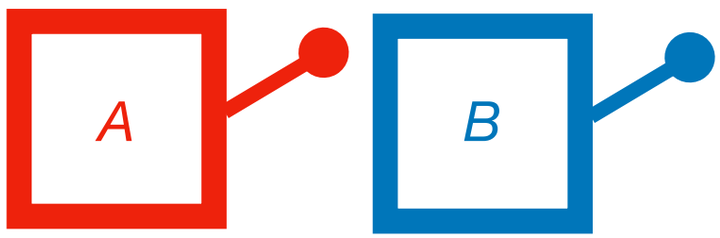
Choosing a restaurant

Playing tic-tac-toe

# Example MDPs

# Example MDPs

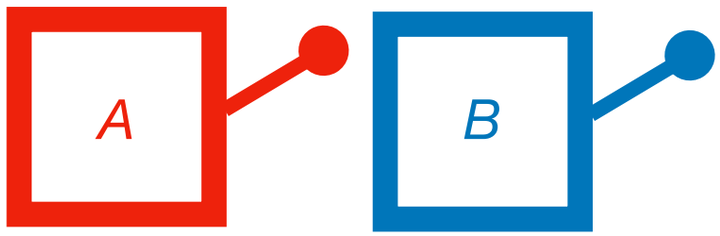
2-armed bandit




***S**: n/a*
***A**: {choose A, choose B}*
***T**: n/a*
***R**: {+1 if A, 0 if B}*

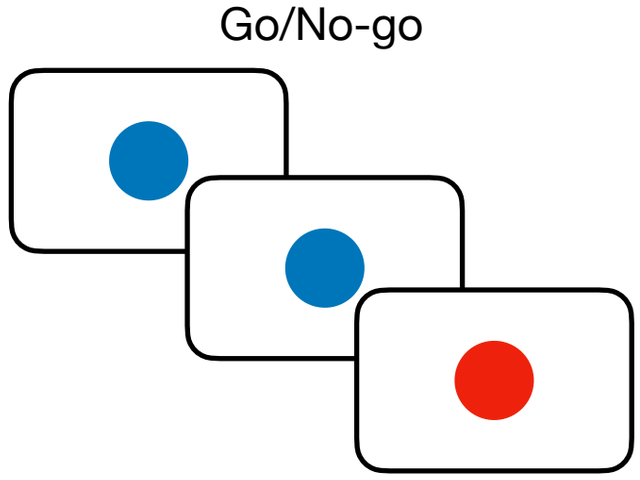
# Example MDPs

2-armed bandit




***S**: n/a*
***A**: {choose A, choose B}*
***T**: n/a*
***R**: {+1 if A, 0 if B}*

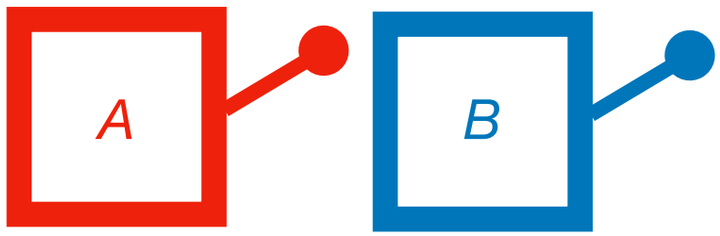
Go/No-go



***S**: { , }*
***A**: {go, no-go}*
***T**: p (red | blue) = p(blue | red) = 0.5*
***R**: {+1 if go in blue, 0 if no-go in blue, +1 if no-go in red, -1 if go in red}*

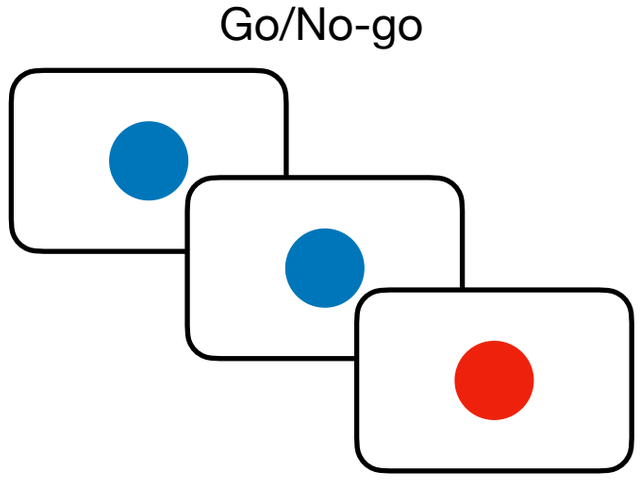
# Example MDPs

2-armed bandit



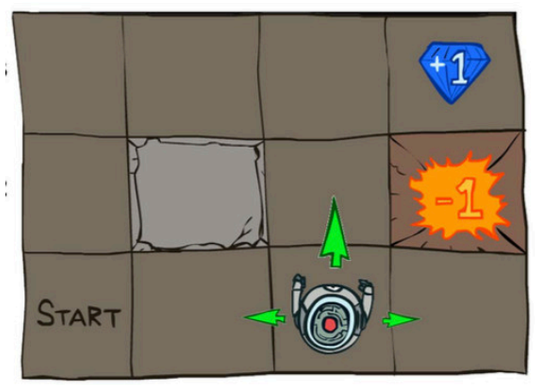

***S**: n/a*
***A**: {choose A, choose B}*
***T**: n/a*
***R**: {+1 if A, 0 if B}*

Go/No-go

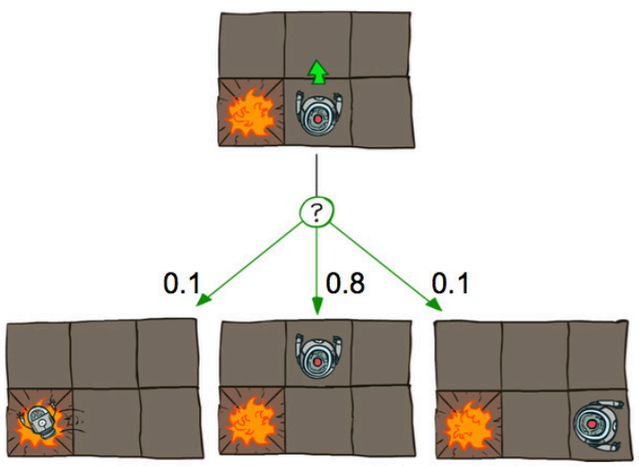


***S**: { , }*
***A**: {go, no-go}*
***T**: p (red | blue) = p(blue | red) = 0.5*
***R**: {+1 if go in blue, 0 if no-go in blue, +1 if no-go in red, -1 if go in red}*

Gridworld




***T**:*




***S**: cells of the grid world*
***A**: {up, down, left, right}*
***R**: {+1 if diamond, -1 if pit, -0.04 for every non-terminal state}*

# Key definitions

The *discounted return* is defined as:

$$G_t = r_{t+1} + \gamma\, r_{t+2} + \ldots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$$

*How much reward will I get in the future?*

# Key definitions

The *discounted return* is defined as:

$$G_t = r_{t+1} + \gamma\, r_{t+2} + \ldots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$$

*How much reward will I get in the future?*

The *policy* $\pi$ defines a distribution over actions given states:

$$\pi(a \mid s) = P\{A_t = a \mid S_t = s\}$$

*What action should I take in each state?*

# Key definitions

The *discounted return* is defined as:

$$G_t = r_{t+1} + \gamma\, r_{t+2} + \ldots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$$

*How much reward will I get in the future?*

The *policy* $\pi$ defines a distribution over actions given states:

$$\pi(a \mid s) = P\{A_t = a \mid S_t = s\}$$

*What action should I take in each state?*

The value: expected return when starting in state *s*:

$$v_\pi(s) = \mathbb{E}_\pi[G_t \mid S_t = s]$$

*Future expected reward from this state onward (how 'good' is this state)*

# Key definitions

The *discounted return* is defined as:

$$G_t = r_{t+1} + \gamma\, r_{t+2} + \ldots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$$

*How much reward will I get in the future?*

The *policy* $\pi$ defines a distribution over actions given states:

$$\pi(a \mid s) = P\{A_t = a \mid S_t = s\}$$

*What action should I take in each state?*

The value: expected return when starting in state *s*:

$$v_\pi(s) = \mathbb{E}_\pi[G_t \mid S_t = s]$$

*Future expected reward from this state onward (how 'good' is this state)*

The Q-value*:* expected return when starting in state *s,* and taking action *a*:

$$q_\pi(s, a) = \mathbb{E}_\pi[G_t \mid S_t = s, A_t = a]$$

*Future expected reward from this state onward if I do an action*

# Optimal policy maximizes value of every state

The *optimal policy* $\pi^*$ satisfies:

$$v_*(s) = \max_\pi v_\pi(s) \qquad \text{, for all} \quad s \in \mathcal{S}$$

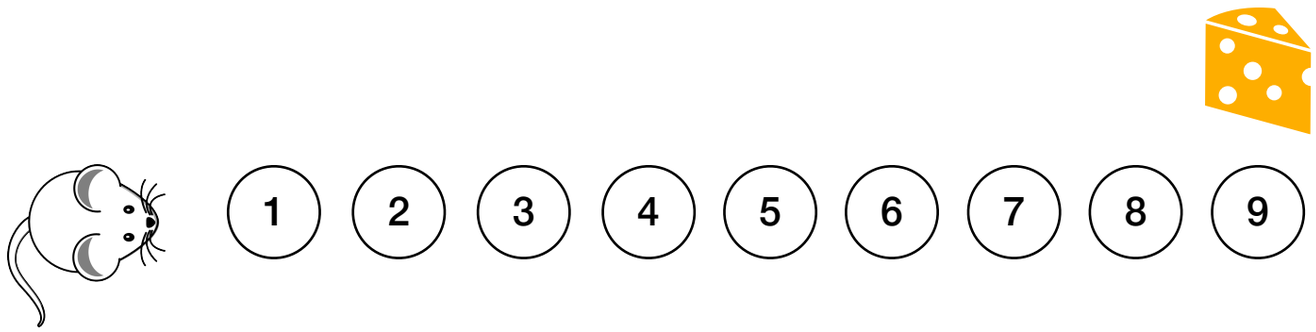
$$q_*(s,a) = \max_\pi q_\pi(s,a) \qquad \text{, for all} \quad s \in \mathcal{S} \quad a \in \mathcal{A}$$

‘Solving’ an MDP: finding the optimal policy.

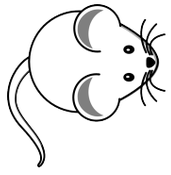
# What is an RL algorithm?

- Given **problem definition** (MDP), set of **iterative steps** for arriving at optimal policy
- Relies on some internal **representation**
    - Value, beliefs (e.g. about state)
- Often, in practice, needs **experience**
- Can be constrained by **implementation** (e.g. neural substrates)
- A **hypothesis** about how behavior is generated
    - Pavlovian vs. instrumental learning
    - “Model-free” vs. “model-based” RL

# Pavlovian conditioning: what is going to happen?

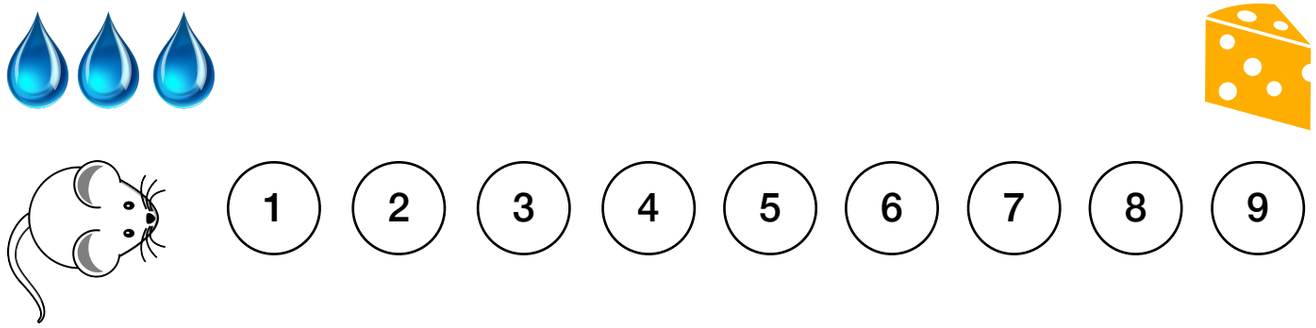


# Pavlovian conditioning: what is going to happen?



1 2 3 4 5 6 7 8 9

# Pavlovian conditioning: what is going to happen?




## Temporal-Difference Learning

TD-Error ($\delta_t$)

$$V(S_t) \leftarrow V(S_t) + \alpha\,[R_{t+1} + \gamma V(S_{t+1}) - V(S_t)]$$

Intuition: “backing up” errors

$V(S_t)$: value of current state = future expected reward

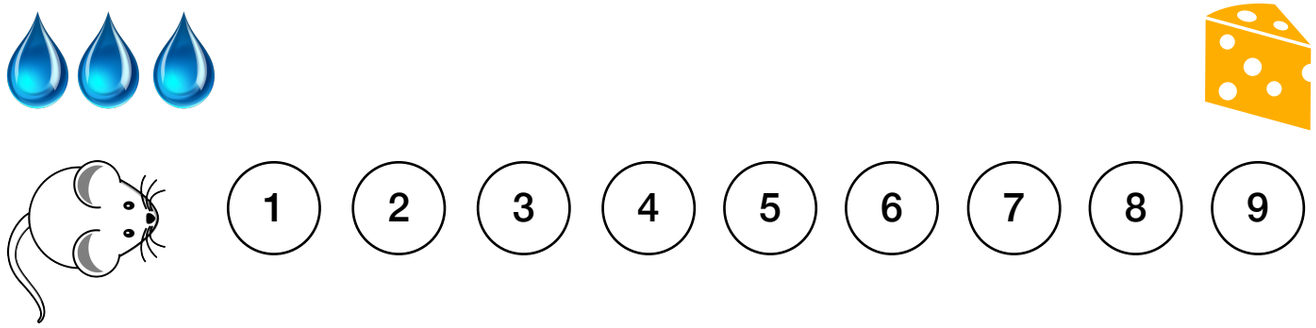
$R_{t+1}$: reward

$V(S_{t+1})$: value of next state

$\alpha$: step size (how fast to update expectations)

$\gamma$: discount factor (how much to discount the future)

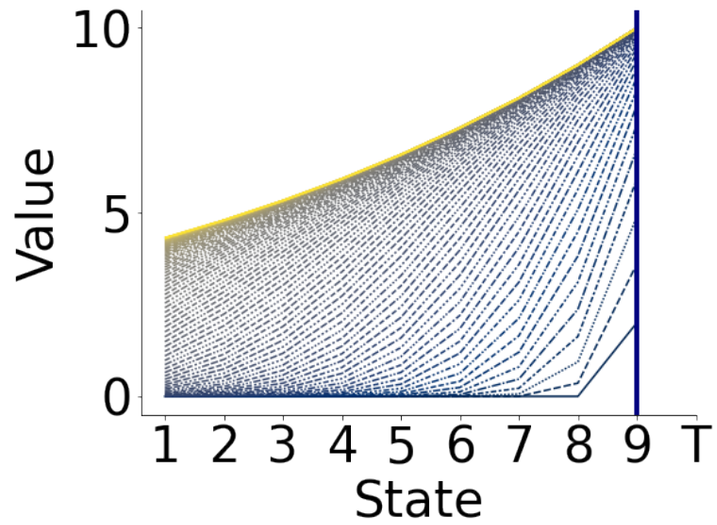
# Pavlovian conditioning: what is going to happen?




## Temporal-Difference Learning

TD-Error ($\delta_t$)

$$V(S_t) \leftarrow V(S_t) + \alpha\,[R_{t+1} + \gamma V(S_{t+1}) - V(S_t)]$$




Intuition: “backing up” errors

$V(S_t)$: value of current state = future expected reward

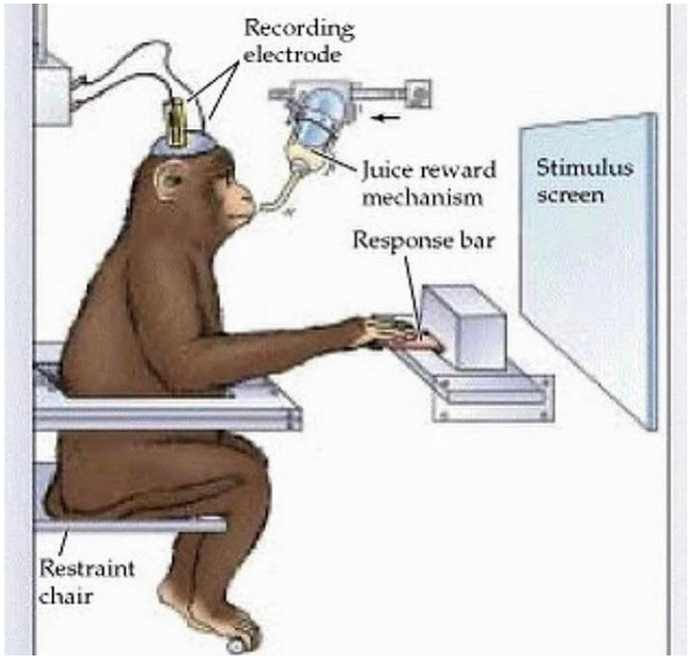
$R_{t+1}$: reward

$V(S_{t+1})$: value of next state

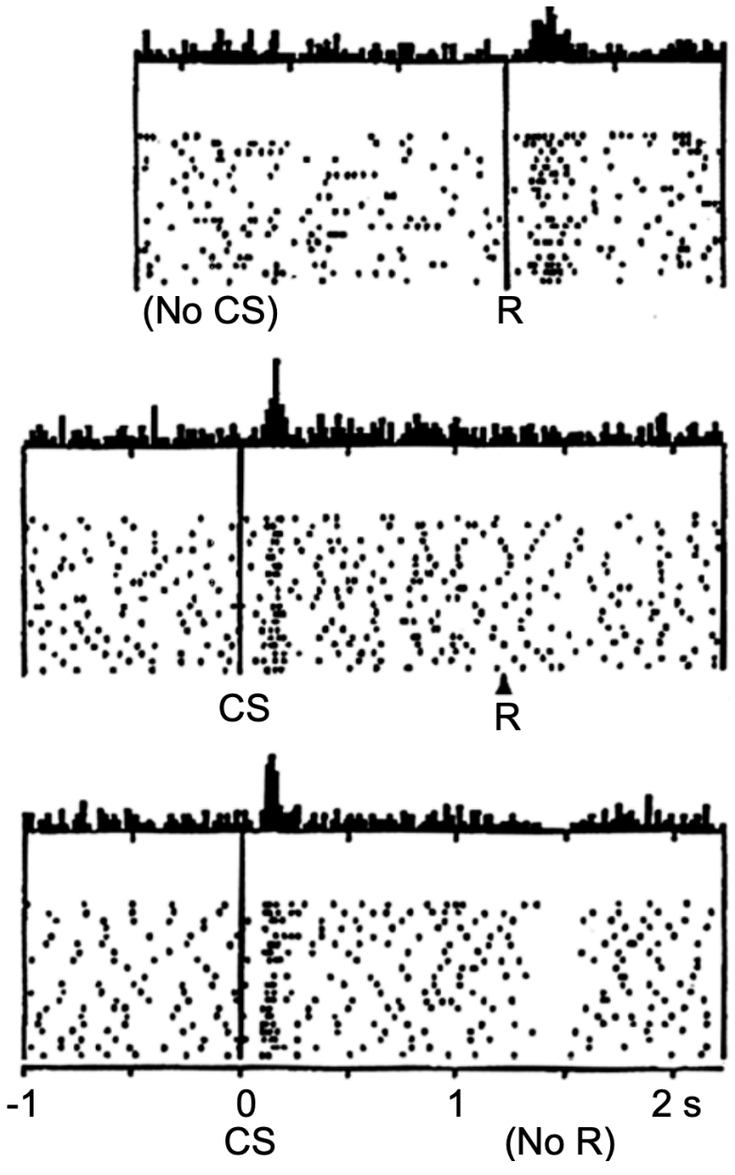
$\alpha$: step size (how fast to update expectations)

$\gamma$: discount factor (how much to discount the future)

# Reward Prediction Error theory of dopamine




<u>Idea</u>: Dopamine encodes a temporal-difference prediction error TD-Error ($\delta_t$)




$\delta_t = R$

$\delta_t = V(S_t)$

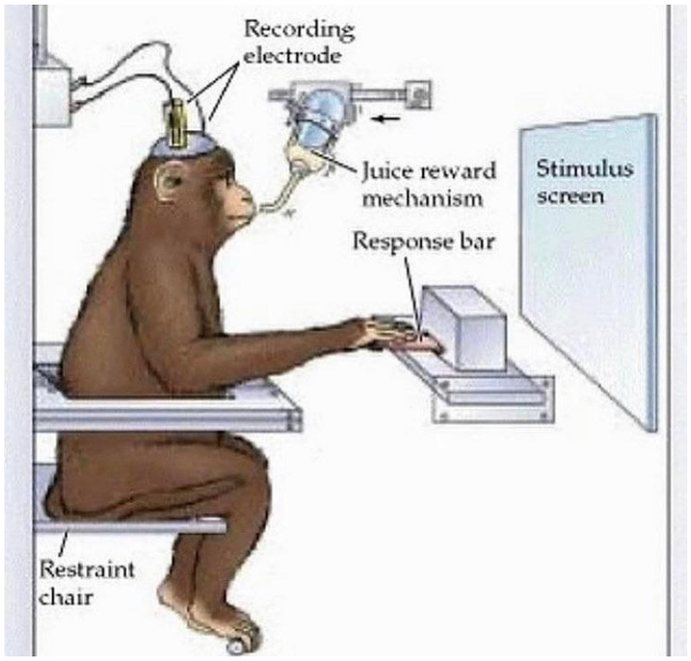
$\delta_t = R - V(S_{t-1})$

$\delta_t = V(S_t)$

$\delta_t = 0 - V(S_{t-1})$

Schultz, Dayan and Montague, 1997; Bayer & Glimcher 2005; Niv 2009; Gershman et al. 2024

# Reward Prediction Error theory of dopamine




nature neuroscience



Perspective | Published: 25 July 2024

## Explaining dopamine through prediction errors and beyond

Idea: Dopamine encodes a temporal-difference prediction error TD-Error ($\delta_t$)

Schultz, Dayan and Montague, 1997; Bayer & Glimcher 2005; Niv 2009; Gershman et al. 2024

# Instrumental conditioning: what can I do to make things better?

$S_1, A_1$

$S_1, A_2$

# Instrumental conditioning: what can I do to make things better?

$S_1, A_1$

$S_1, A_2$

$S_2, A_1$

$S_2, A_2$

# Instrumental conditioning: what can I do to make things better?

$S_1, A_1$

$S_1, A_2$

$S_2, A_1$



$S_2, A_2$

# Instrumental conditioning: what can I do to make things better?

$S_1, A_1$

$S_1, A_2$

$S_2, A_1$



$S_2, A_2$



## Q-Learning

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left[R_{t+1} + \gamma \max Q(S_{t+1}, a) - Q(S_t, A_t)\right]$$

| States \ Actions | $A_1$ | $A_2$ | … | $A_n$ |
|---|---|---|---|---|
| $S_1$ | $q_{1,1}$ | $q_{1,2}$ | … | $q_{1,n}$ |
| $S_2$ | $q_{2,1}$ | $q_{2,2}$ | … | $q_{2,n}$ |
| ⋮ | ⋮ | ⋮ | ⋱ | ⋮ |
| $S_m$ | $q_{m,1}$ | $q_{m,2}$ | … | $q_{m,n}$ |

# “Model-free” RL

Rescorla-Wagner Learning
TD-Learning
Q-Learning
Actor-Critic

Experience

*Behavior*

Value functions
Optimal policies

“It’s better to be indoors when it storms”

“Take umbrella if it’s raining”

…

# “Model-based” RL

*Model of the environment (T)*

$$\sum_{s',r} p(s', r \mid s, a)$$

“If I turn left, I will get to the train station”

Tree Search
Successor Representation

Planning

*Behavior*

Value functions
Optimal policies

# “Model-based” RL

*Model of the environment (T)*

$$\sum_{s',r} p(s', r \mid s, a)$$

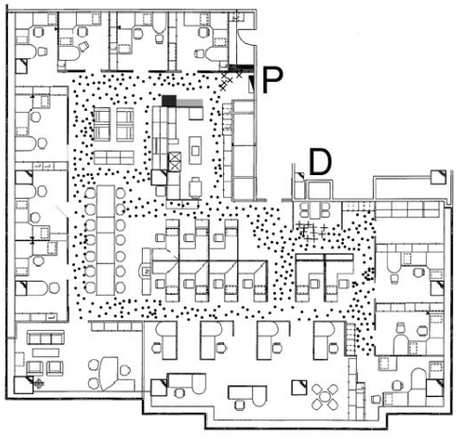
“If I turn left, I will get to the train station”

Tree Search
Successor Representation

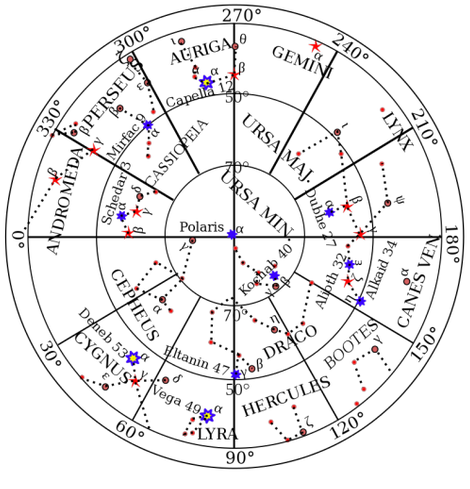
Planning

*Behavior*

Value functions
Optimal policies







# “Model-based” RL

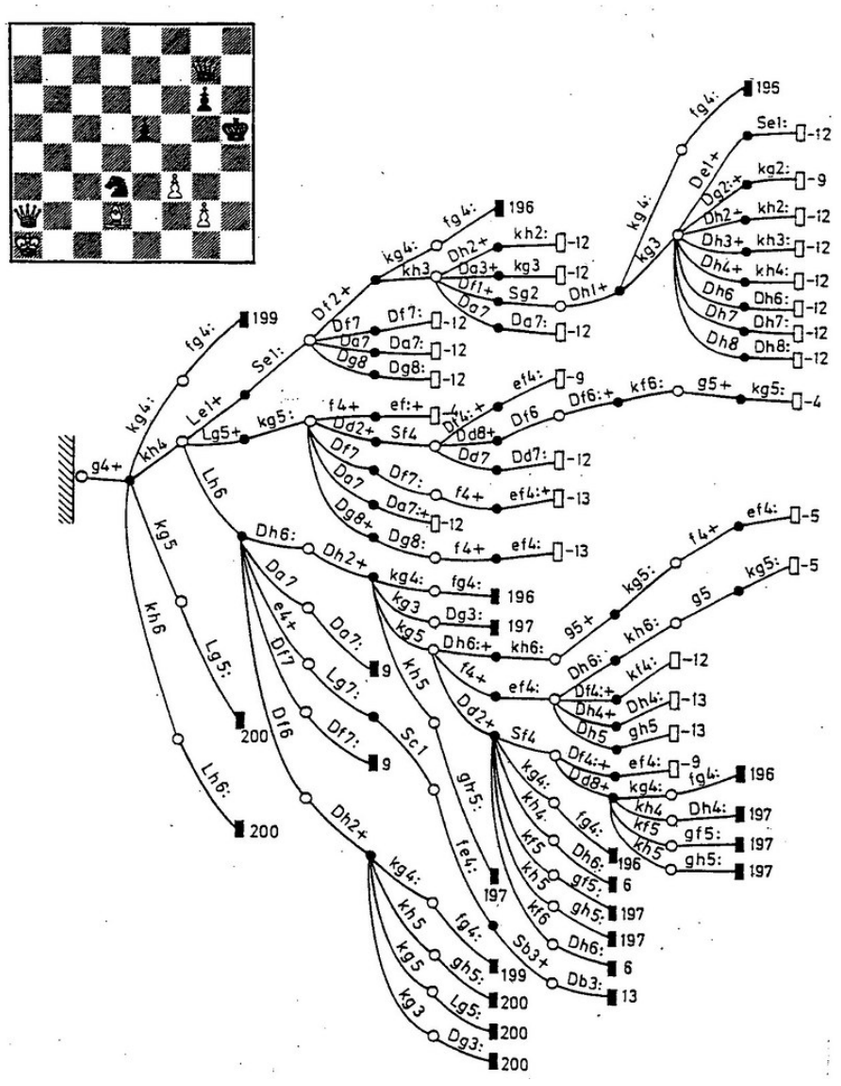
*Model of the environment (T)*

$$\sum_{s',r} p(s', r \mid s, a)$$

Model allows agent to **plan**.

**Planning -** predict what will happen in the future… “what state will I end up in, what reward will I get?”

Update value functions from **simulated** experience (instead of through learning).



# “Model-based” RL

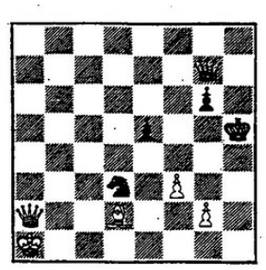
*Model of the environment (T)*

$$\sum_{s',r} p(s', r \mid s, a)$$

Model allows agent to **plan**.

**Planning -** predict what will happen in the future… “what state will I end up in, what reward will I get?”

Update value functions from **simulated** experience (instead of through learning).



“Rollout”

| Problem specification | Class of solutions / algorithms |
| --- | --- |
| Lever pressing | Model-free RL |
| Caching nuts | Model-based RL |
| Choosing a restaurant | Rescorla-Wagner |
| Playing tic-tac-toe | Planning |

I. RL background

**II. RL perspectives on aging**

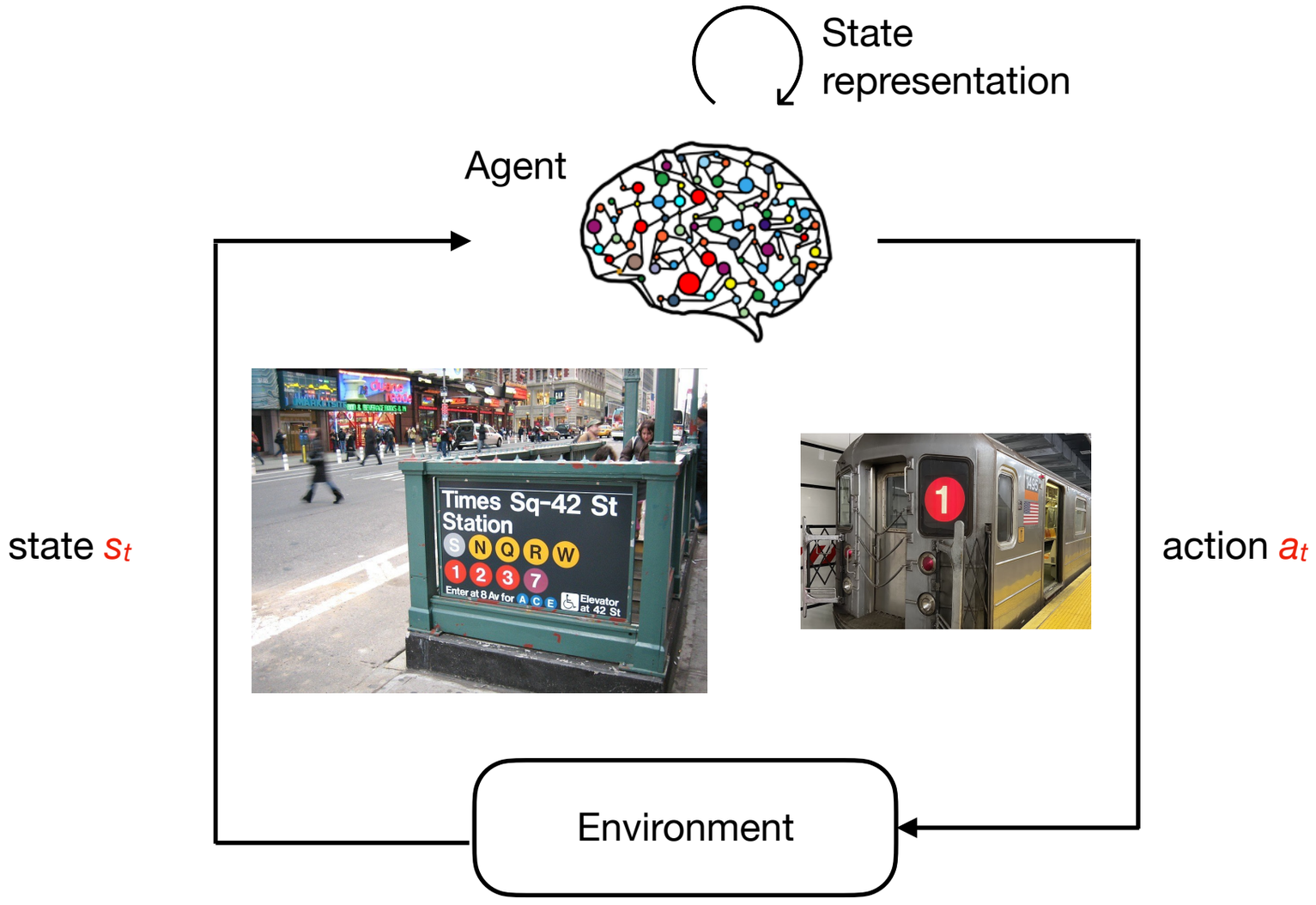
# Why is RL hard?

- **Curse of dimensionality**: learning and action selection in a *multidimensional* world

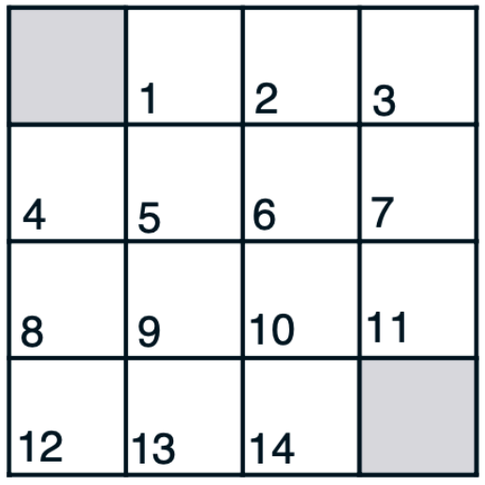
- **Credit assignment problem**:
  - Outcomes may depend on a *series of actions*…

  …we experience *many states* before we know where rewards are

# Solution: learn both what to do, and how to represent the world



# A simple state representation


| States \ Actions | $A_1$ | $A_2$ | … | $A_n$ |
|---|---|---|---|---|
| $S_1$ | $q_{1,1}$ | $q_{1,2}$ | … | $q_{1,n}$ |
| $S_2$ | $q_{2,1}$ | $q_{2,2}$ | … | $q_{2,n}$ |
| ⋮ | ⋮ | ⋮ | ⋱ | ⋮ |
| $S_m$ | $q_{m,1}$ | $q_{m,2}$ | … | $q_{m,n}$ |

$S_t$ is in {1, 2, … 14}

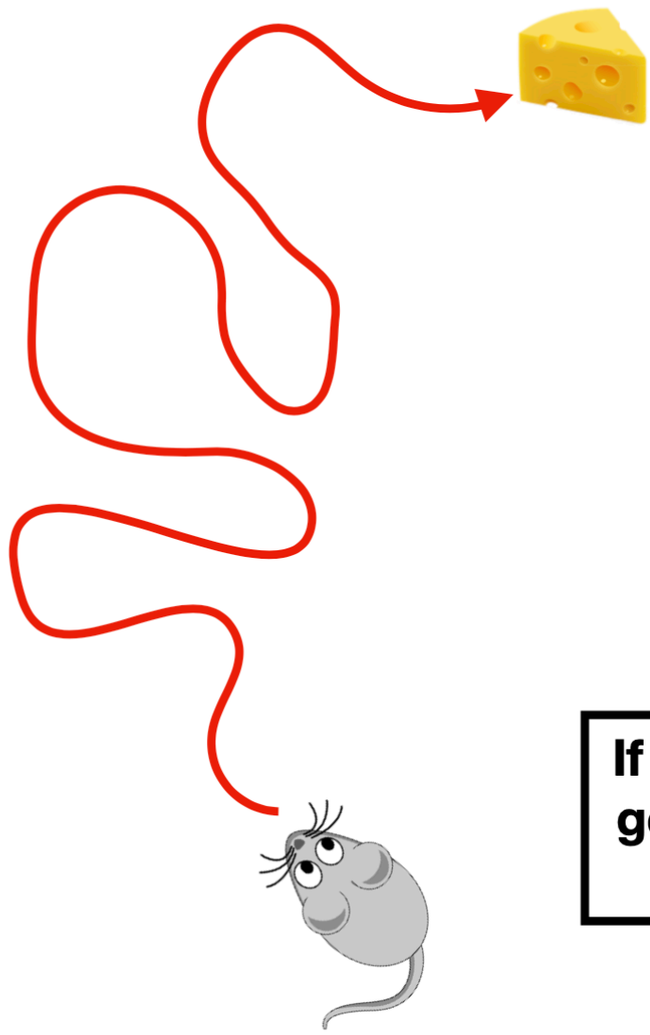
# A more complex state representation


$S_t$ = pixels?

# A real-world observation

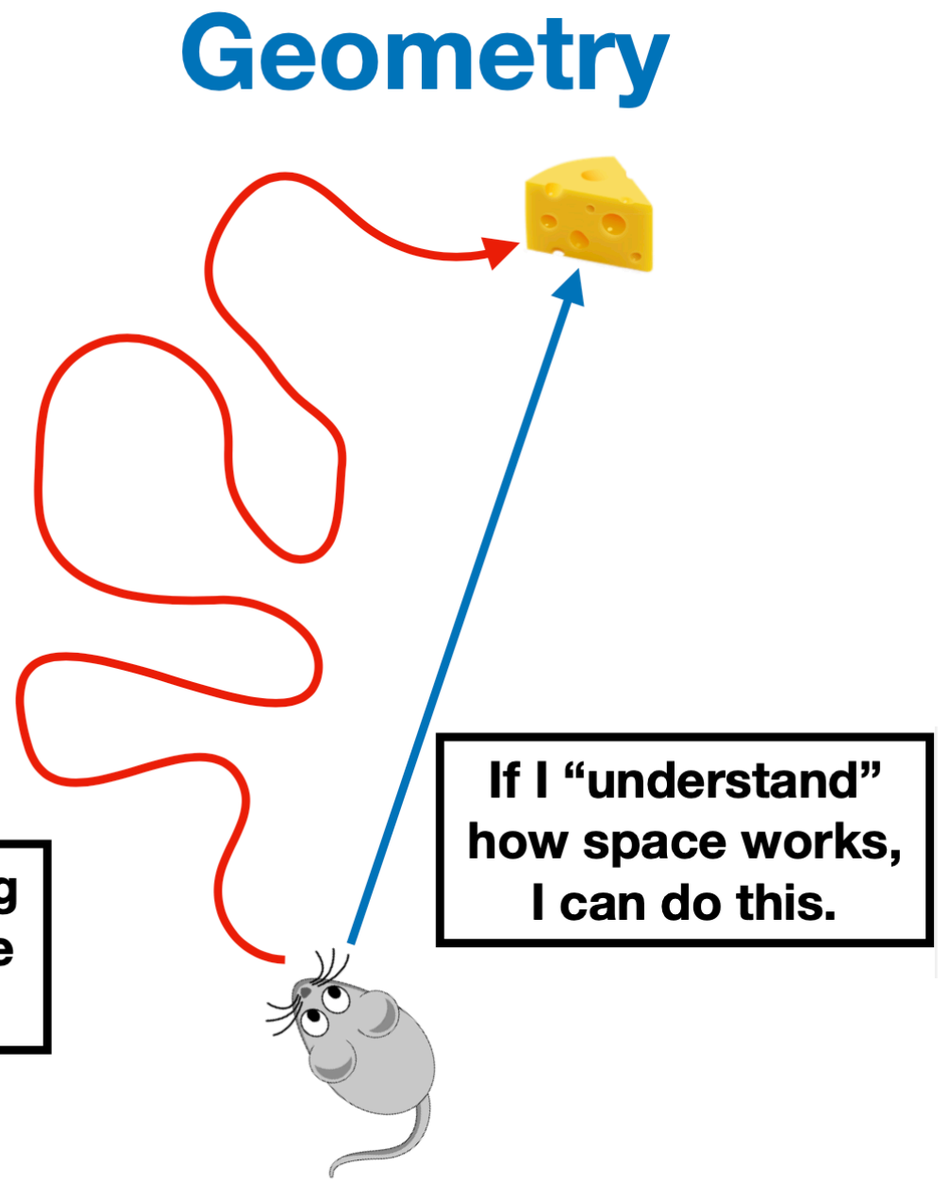

$S_t$ = ??

# Representation learning



If I am just repeating good actions, I have to do this again

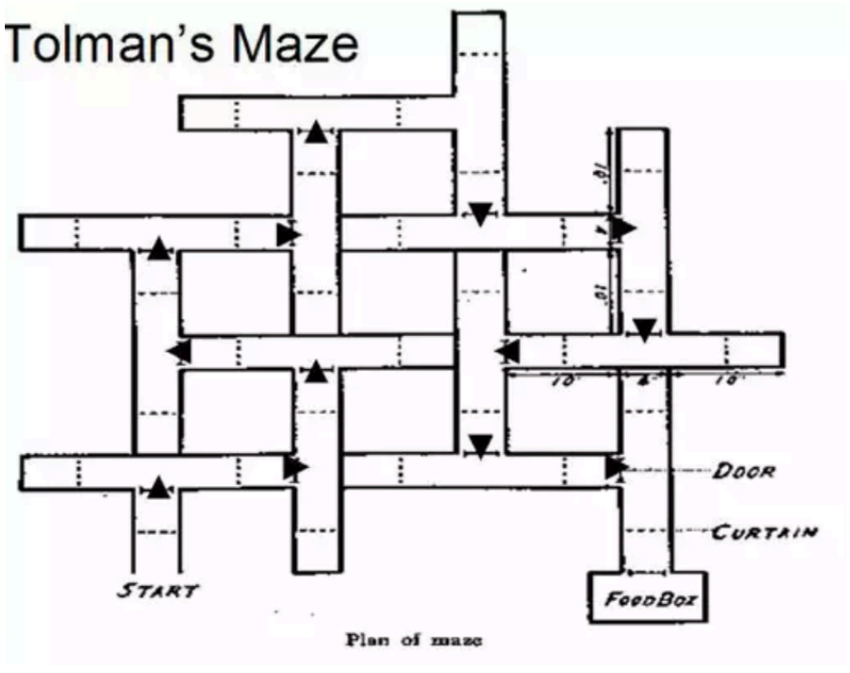
**Experienced paths**



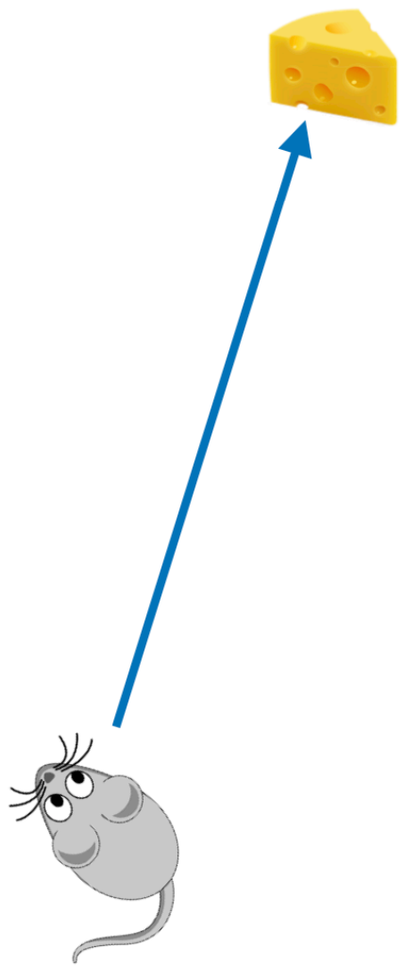

Next time I want cheese.

Tolman 1948

# Representation learning



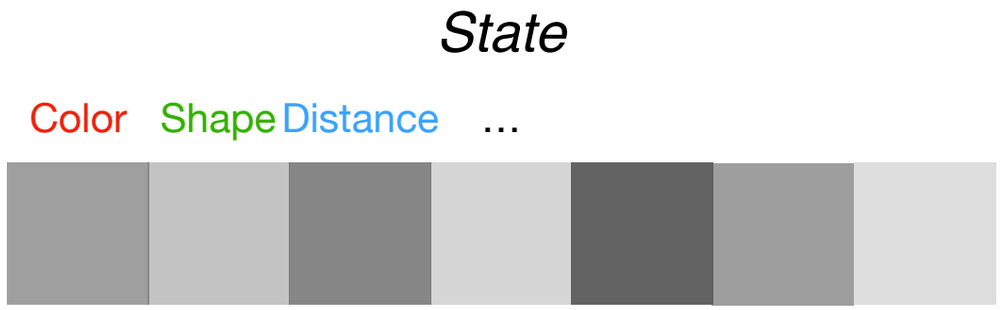

**Experienced paths**



**Generalisable features of maps**

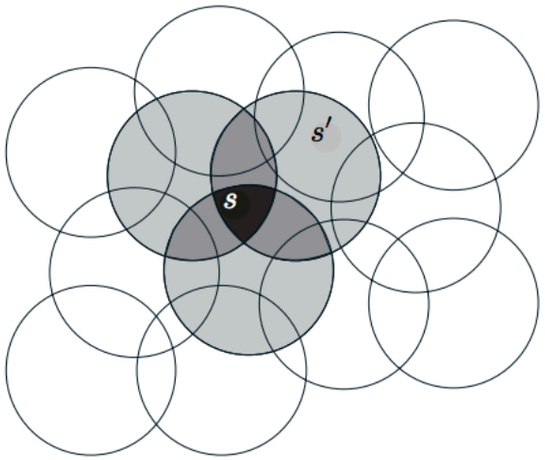
# States represented as a feature vector

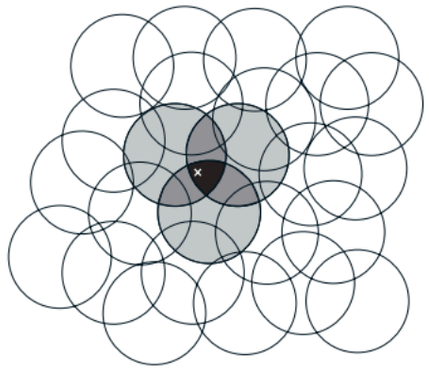
- Instead of directly looking up the states in a table, we can represent different features of the states



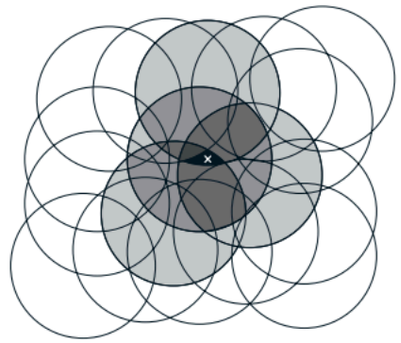
# Feature-based generalization

- States that contain similar features are grouped together

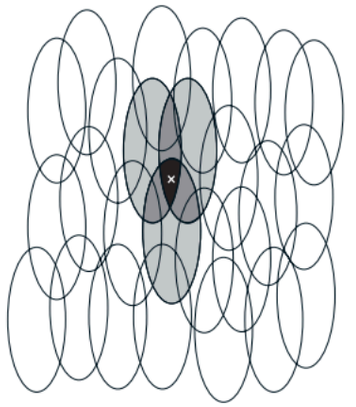


Narrow generalization



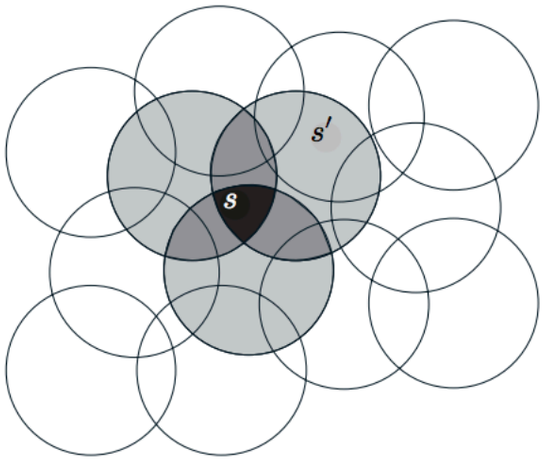
Broad generalization

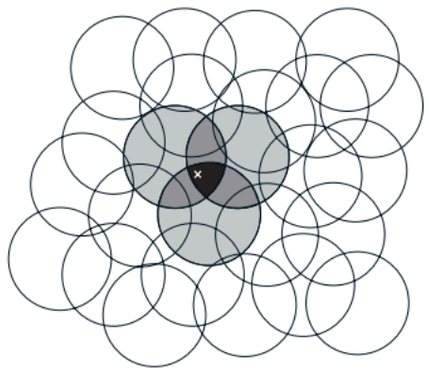


Asymmetric generalization

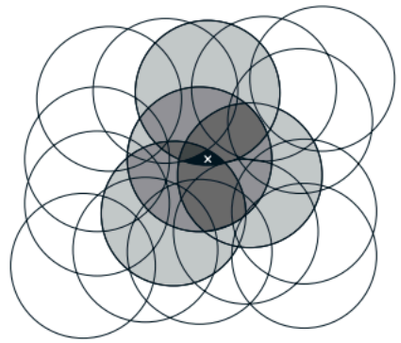
# Feature-based generalization

- States that contain similar features are grouped together

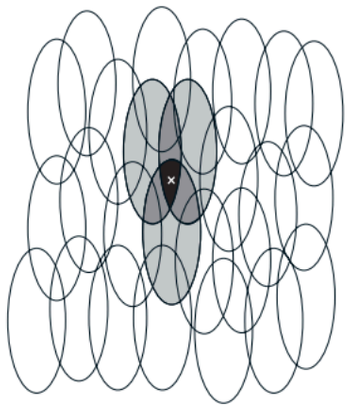


Narrow generalization



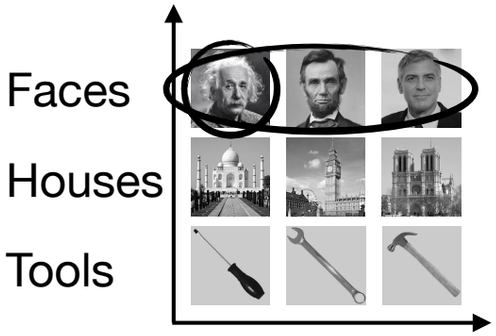
Broad generalization



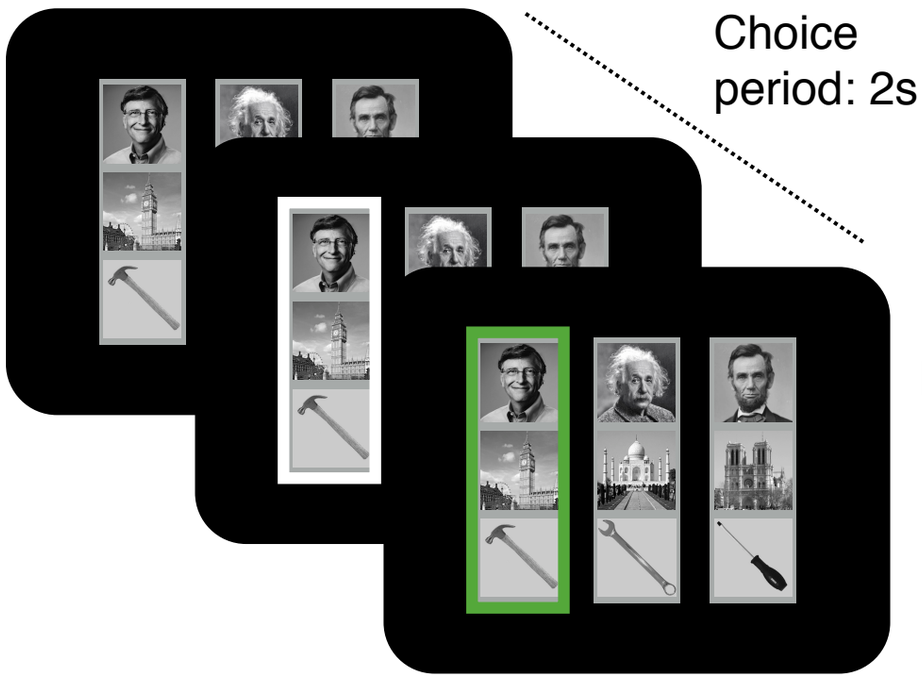
Asymmetric generalization

*How do humans do this?*

# Multidimensional reinforcement learning task




**1 relevant dimension**
**1 target feature f***
p(reward | f* in chosen) = 0.75
p(reward | f* not in chosen) = 0.25

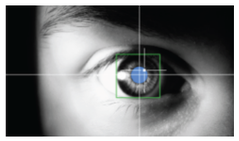


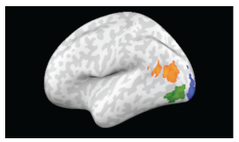
Choice
period: 2s

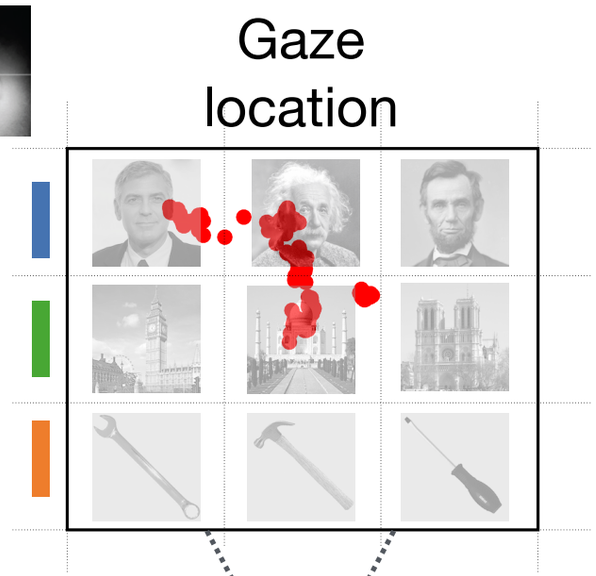
Outcome
period: 2s

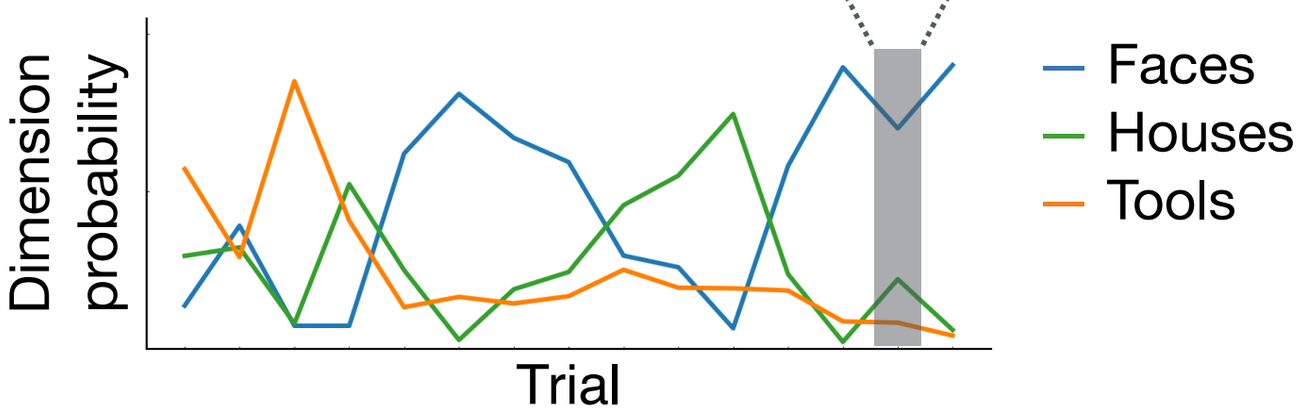
Leong et al. 2017

# Trial-by-trial attention decoding








$$\begin{bmatrix} A_{t,\,faces} \\ A_{t,\,houses} \\ A_{t,\,tools} \end{bmatrix}$$

Leong et al. 2017

# Behavioral model: attention-weighted RL

Value

$V(\,)_t = v(\,)_t + v(\,)_t + v(\,)_t$ $\begin{bmatrix} A_{t,\,faces} \\ A_{t,\,houses} \\ A_{t,\,tools} \end{bmatrix}$

Attention at choice and learning

$v[\,,\,,\,]_t \leftarrow v[\,,\,,\,]_t + \alpha\,[R_{t+1} - V(\,)_t]$ $\begin{bmatrix} A_{t,\,faces} \\ A_{t,\,houses} \\ A_{t,\,tools} \end{bmatrix}$

Reward Prediction Error

Leong et al. 2017

# Models evaluated based on how well they predict choice

Reinforcement learning model *M(θ)*

# Models evaluated based on how well they predict choice

Reinforcement learning model *M(θ)*

Softmax
action selection

$$p(c_t = \text{[image]}) \propto e^{\beta V t(\text{[image]})}$$

# Models evaluated based on how well they predict choice

Reinforcement learning model $M(\theta)$

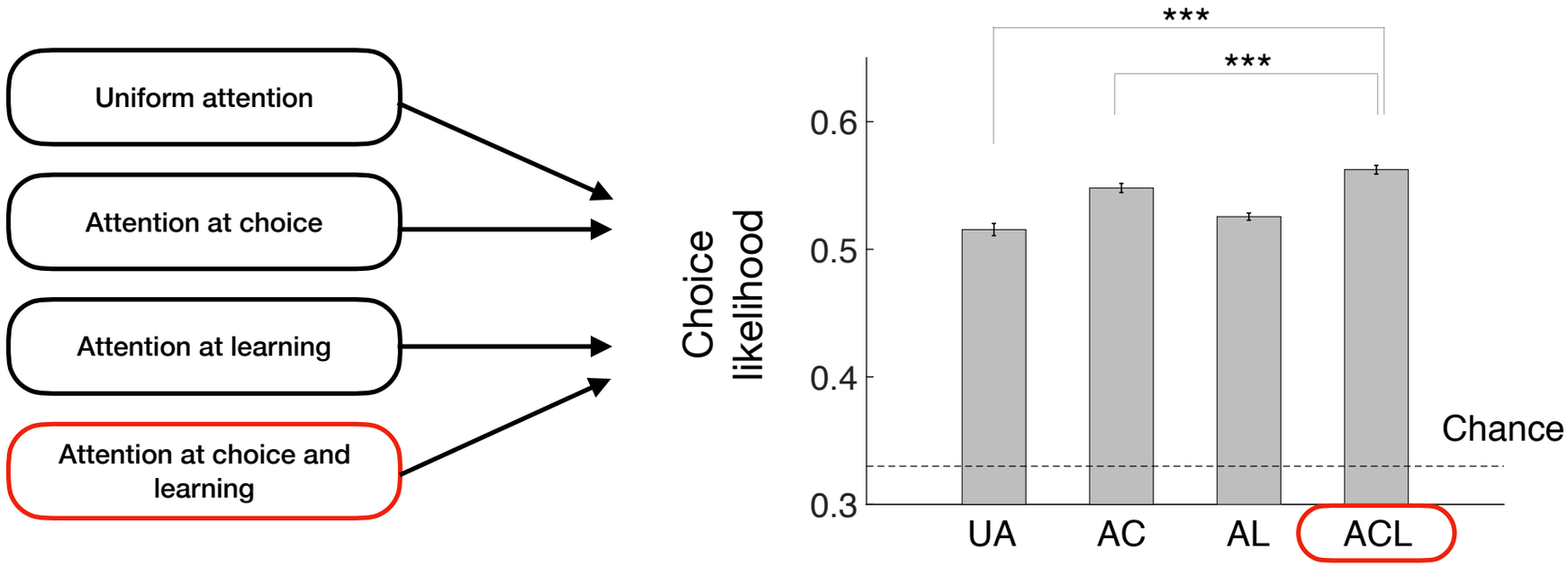
Softmax action selection

$$p(c_t = \text{  }) \propto e^{\beta V_t(\text{  })}$$



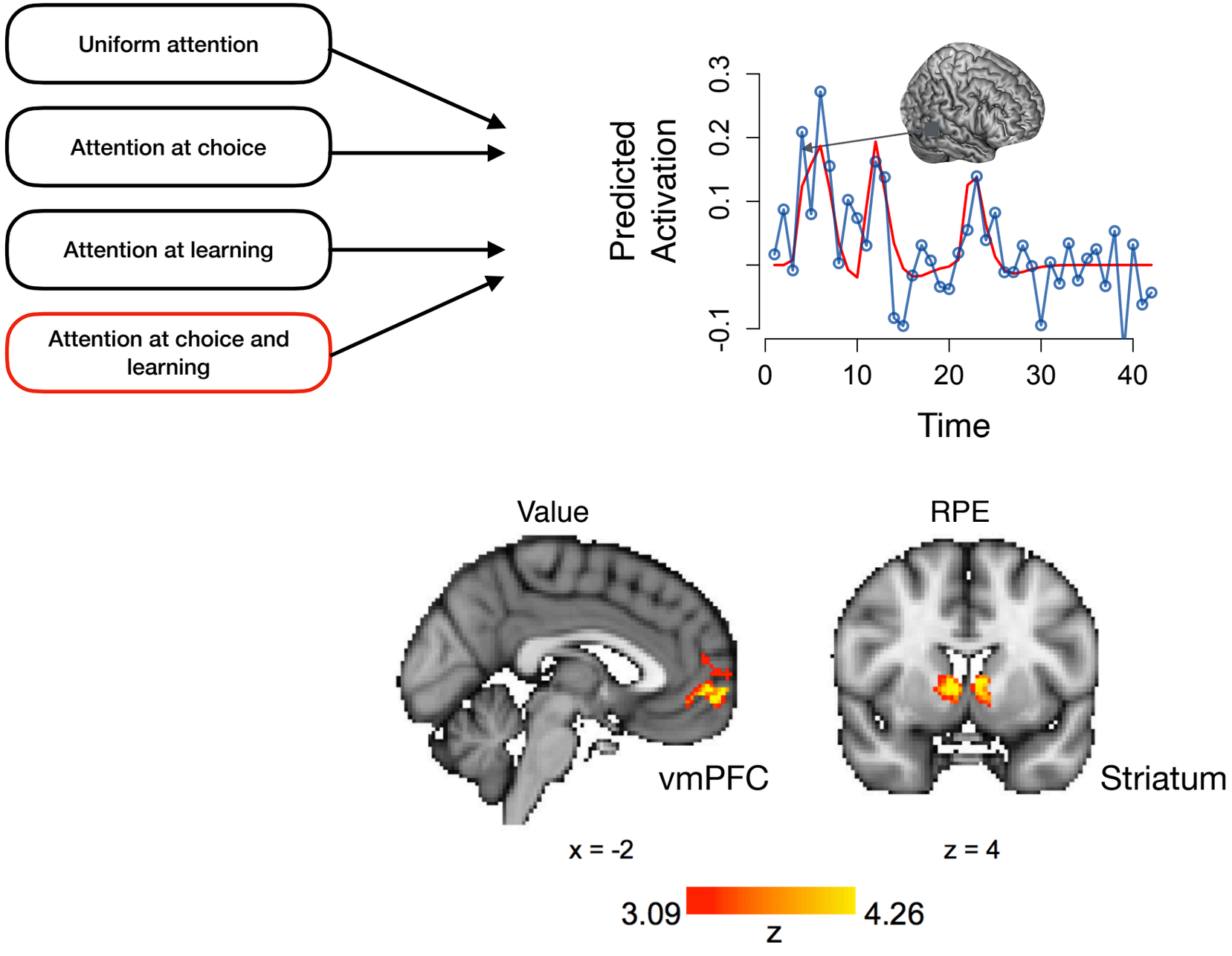
$$\arg\min_{\theta} \prod_{t} p(c_t = \text{  } \mid V_t(\text{  }), V_t(\text{  }), V_t(\text{  }))$$

# Attention helps compress states…




Leong et al. 2017

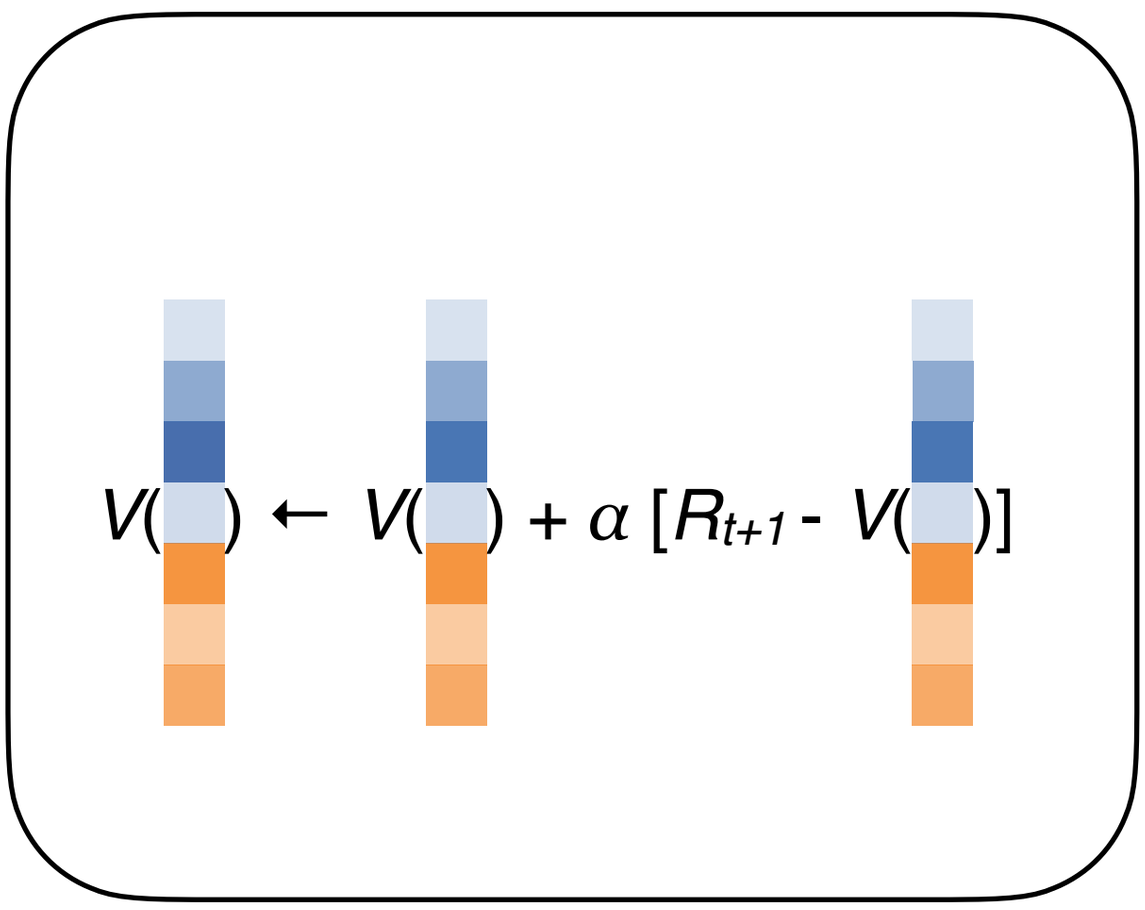
# Attention helps compress states…




Leong et al. 2017

# … and supports generalization

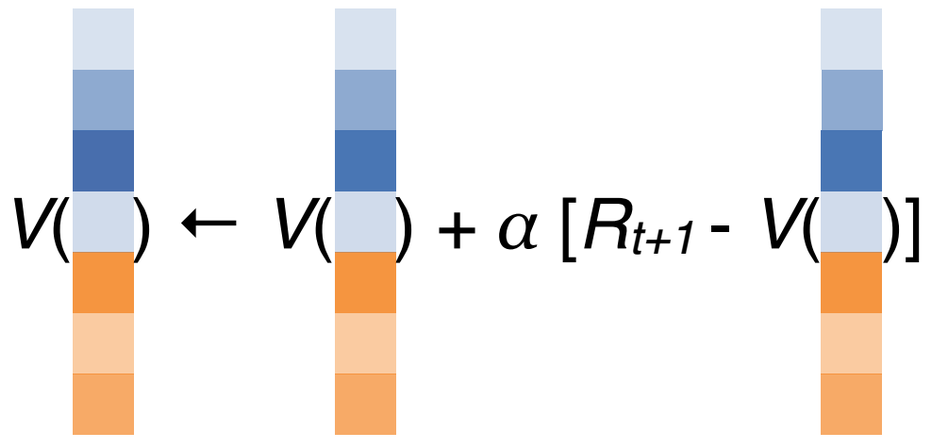
*Reinforcement learning*



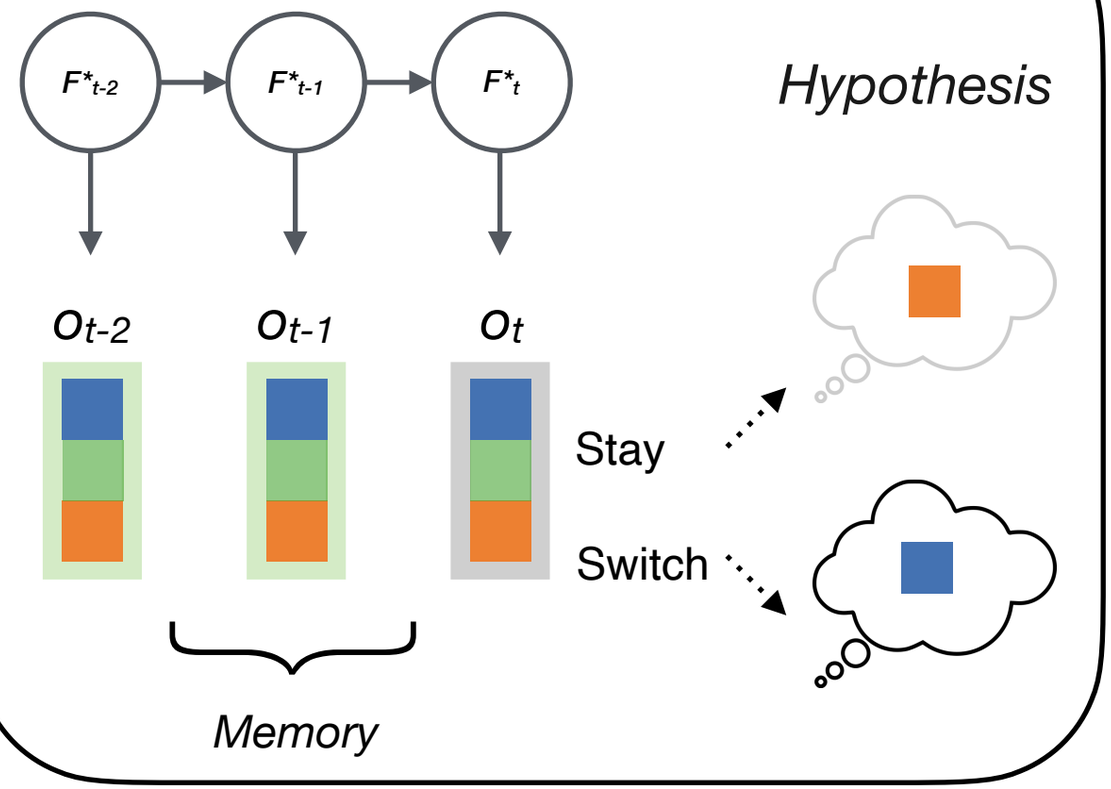
# Attention learning as sampling

*Attention*

*Reinforcement learning*




*Sequential sampling*




Leong et al. 2017
Wilson & Niv, 2011
Radulescu, Niv and Ballard, 2019
Song et al. 2022

# Representation learning in healthy aging

# Representation learning in healthy aging

Harder to learn stimulus-reward associations

Changes in DA signaling

Mell et al., 2005
Eppinger et al., 2013

# Representation learning in healthy aging

Harder to learn stimulus-reward associations

Changes in DA signaling

Mell et al., 2005
Eppinger et al., 2013

Harder to control attention, increased distractibility

Braver & Barch, 2002
Schmitz et al., 2010

# Representation learning in healthy aging

## Changes in representation learning?

Harder to learn stimulus-reward associations

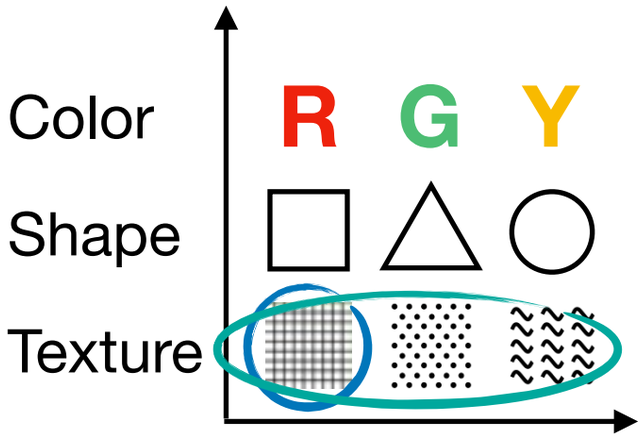
Changes in DA signaling

Mell et al., 2005
Eppinger et al., 2013

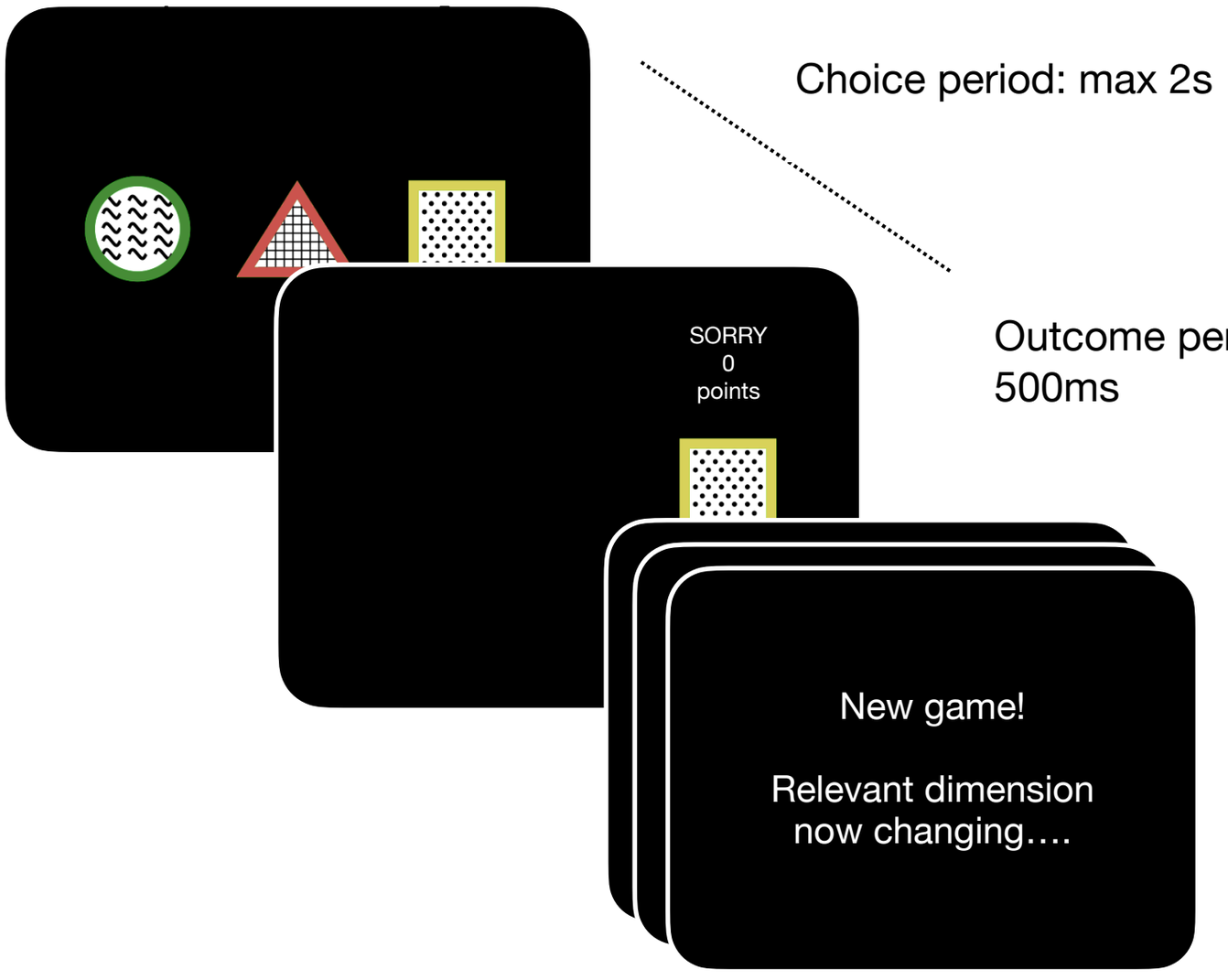
Harder to control attention, increased distractibility

Braver & Barch, 2002
Schmitz et al., 2010

# Representation learning in healthy aging




**1 relevant dimension**
**1 target feature f***
p(reward | f* in chosen) = 0.75
(reward | f* not in chosen) = 0.25

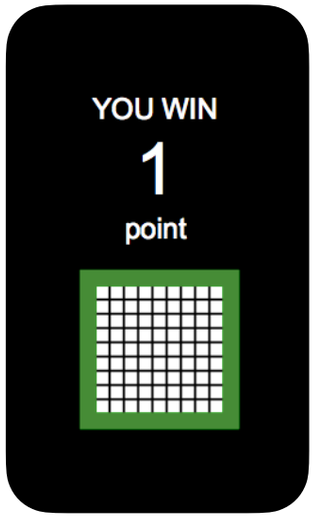



Choice period: max 2s

Outcome period:
500ms

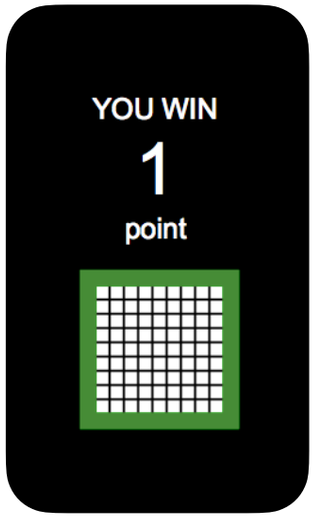
Radulescu et al. 2016

# Behavioral model: feature RL with decay




Radulescu et al. 2016

# Behavioral model: feature RL with decay

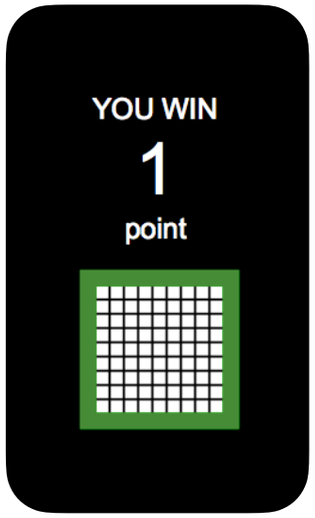



$$V(\blacksquare) = v(\text{green}) + v(\square) + v(\blacksquare)$$

*value*

Radulescu et al. 2016

# Behavioral model: feature RL with decay

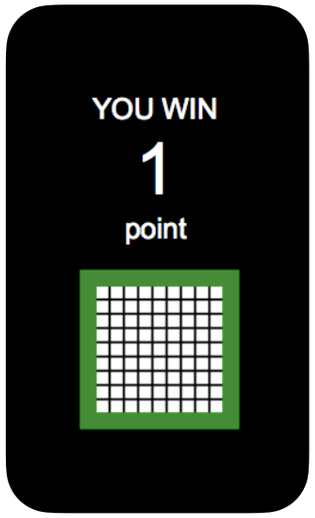



$V(\blacksquare) = v(\text{green}) + v(\square) + v(\blacksquare)$ *value*

$\delta = \text{reward} - V(\blacksquare)$ *prediction error*

Radulescu et al. 2016

# Behavioral model: feature RL with decay




$V(\blacksquare) = v(\text{green}) + v(\square) + v(\blacksquare)$ *value*

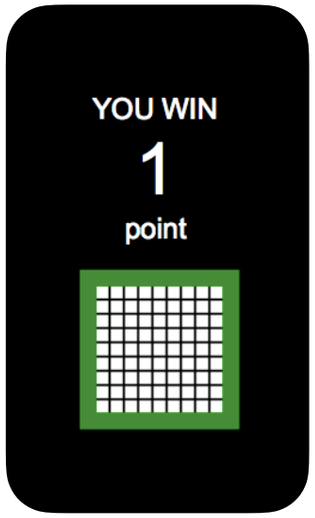
$\delta = \text{reward} - V(\blacksquare)$ *prediction error*

chosen features

$v(\text{green}) \leftarrow v(\text{green}) + \eta \cdot \delta$ *learning ($\eta$<1)*

Radulescu et al. 2016

# Behavioral model: feature RL with decay




$V(\blacksquare) = v(\text{green}) + v(\square) + v(\blacksquare)$ *value*

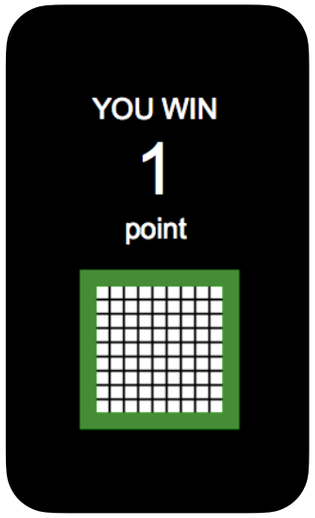
$\delta = \text{reward} - V(\blacksquare)$ *prediction error*

chosen features: $v(\text{green}) \leftarrow v(\text{green}) + \eta \cdot \delta$ *learning ($\eta$<1)*

other features: $v(\text{red}) \leftarrow k \cdot v(\text{red})$ *decay to 0 (k<1)*

Radulescu et al. 2016

# Behavioral model: feature RL with decay




$V(\blacksquare) = v(\text{green}) + v(\square) + v(\blacksquare)$ *value*

$\delta = \text{reward} - V(\blacksquare)$ *prediction error*

chosen features: $v(\text{green}) \leftarrow v(\text{green}) + \eta \cdot \delta$ *learning ($\eta$<1)*

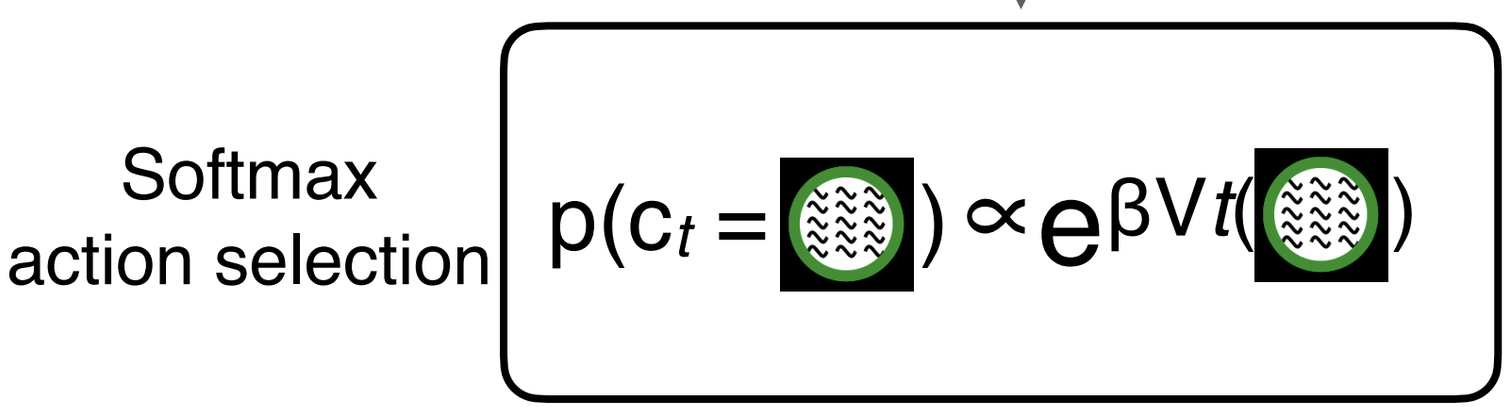
other features: $v(\text{red}) \leftarrow k \cdot v(\text{red})$ *decay to 0 ($k$<1)*

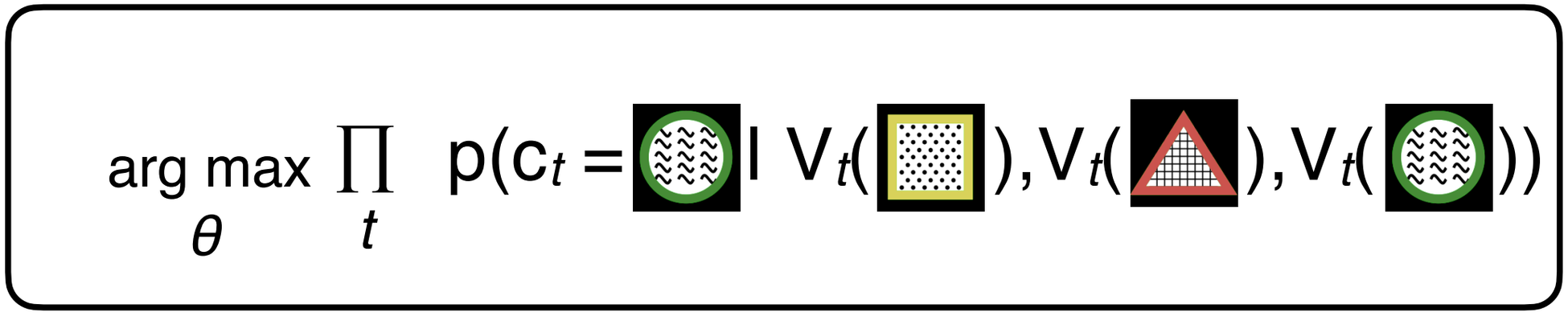
***"attentional selection"***

Radulescu et al. 2016

# Models evaluated based on how well they predict choice

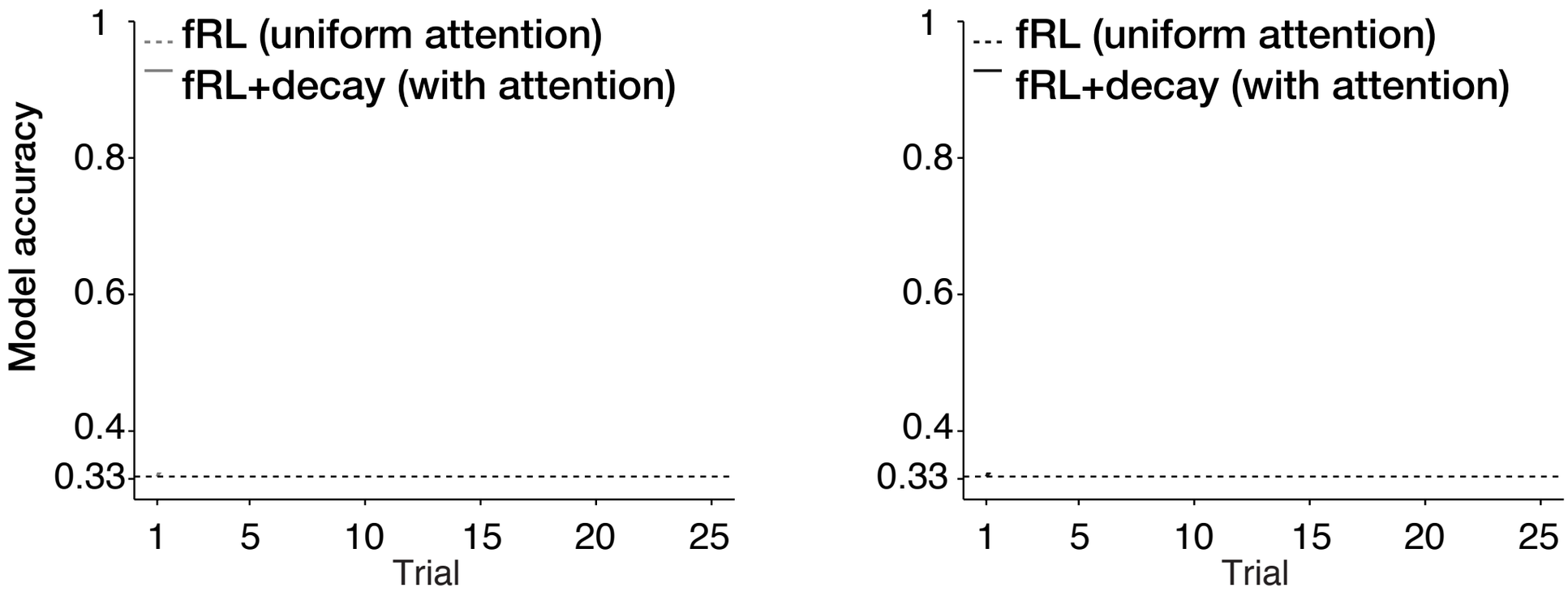
Reinforcement learning model $M(\theta)$



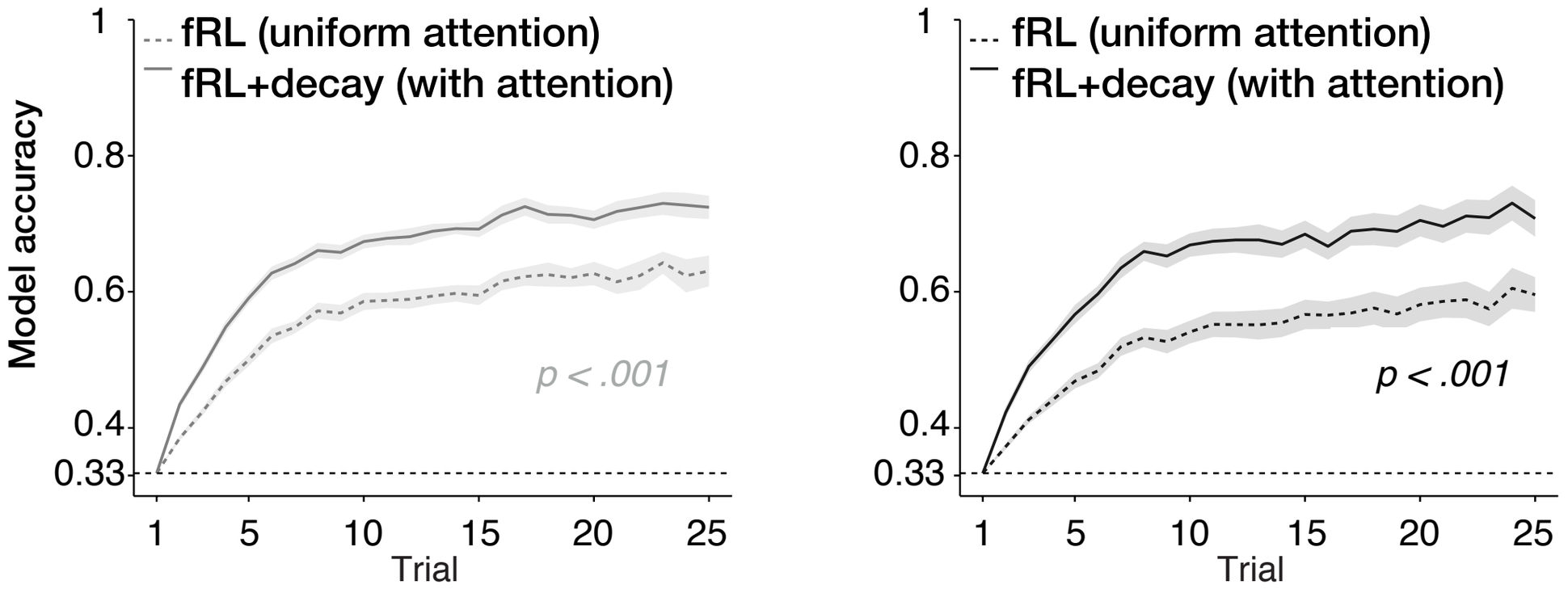

Maximum Likelihood
Estimation

# Results: model accuracy




Radulescu et al. 2016

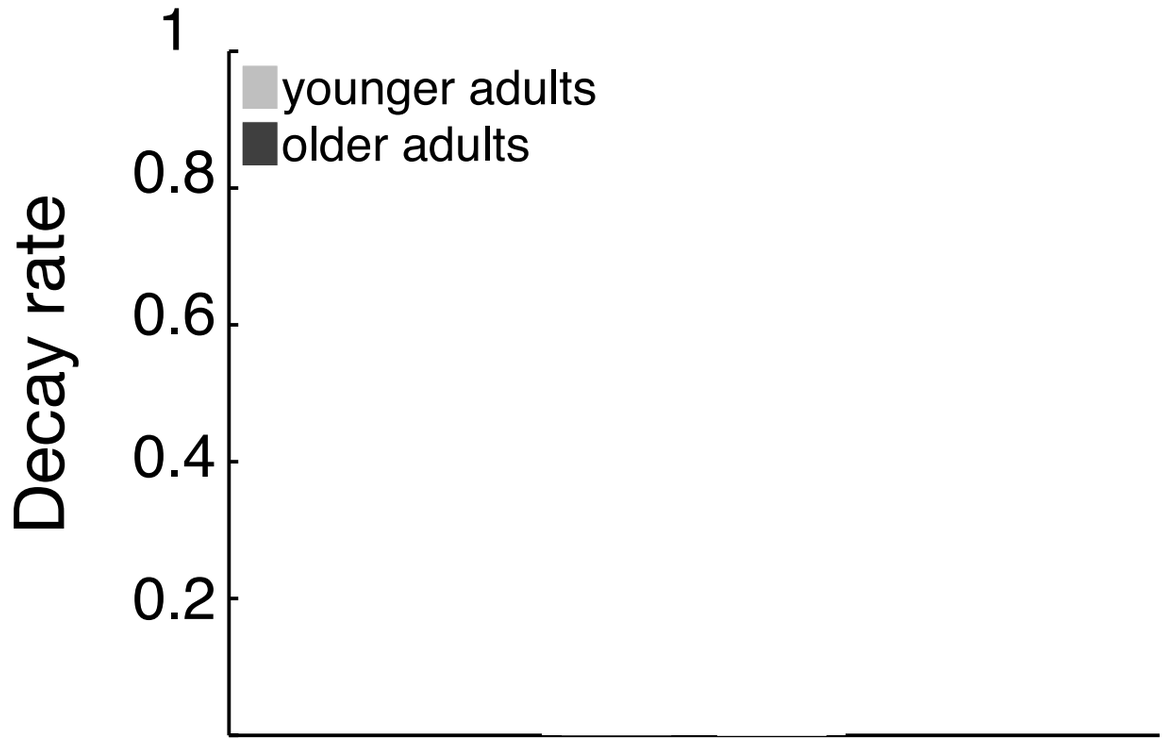
# Results: model accuracy




Radulescu et al. 2016

# Results: group comparison

Narrow focus of attention

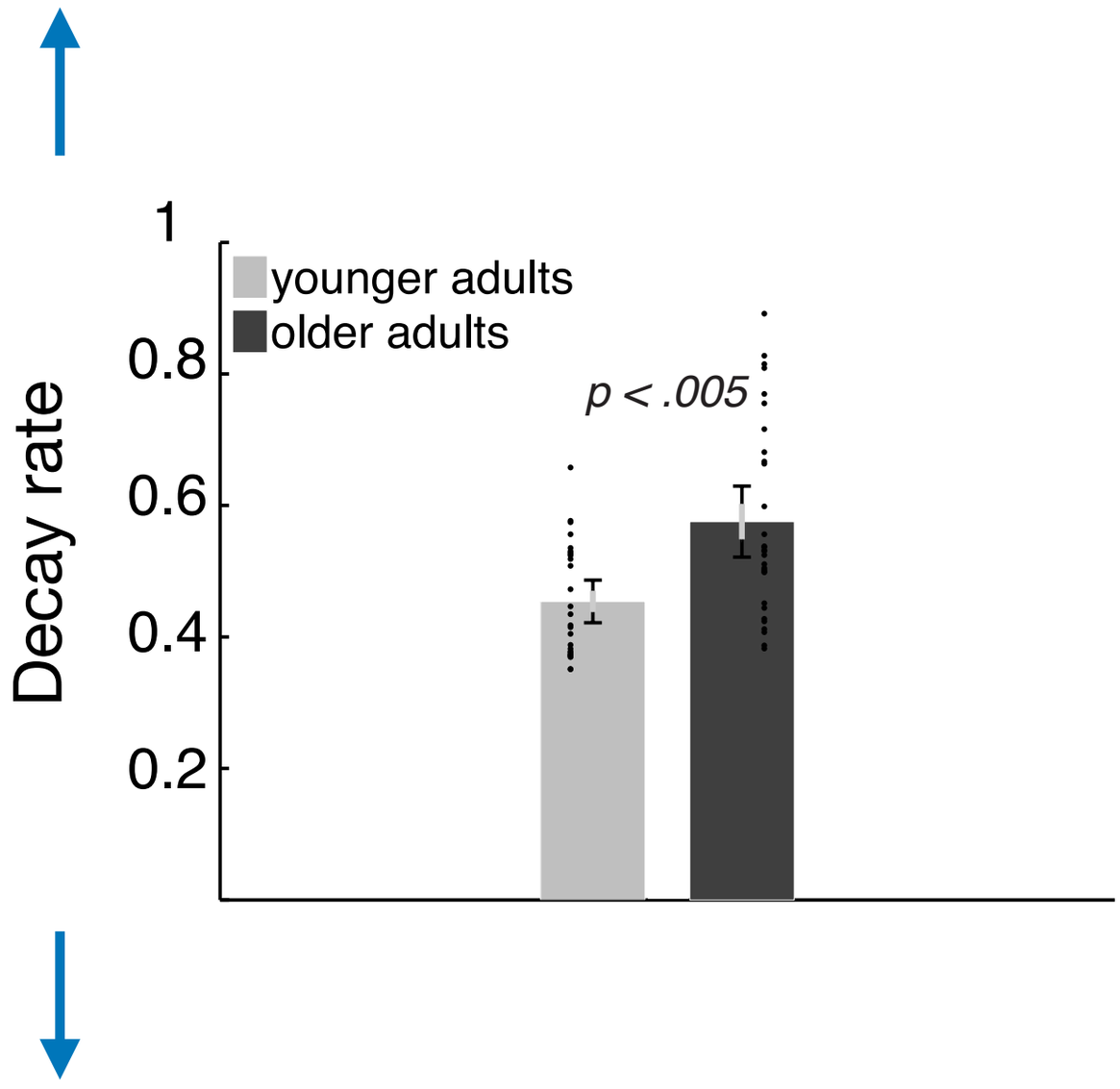



Broad focus of attention

Radulescu et al. 2016

# Results: group comparison

Narrow focus of attention

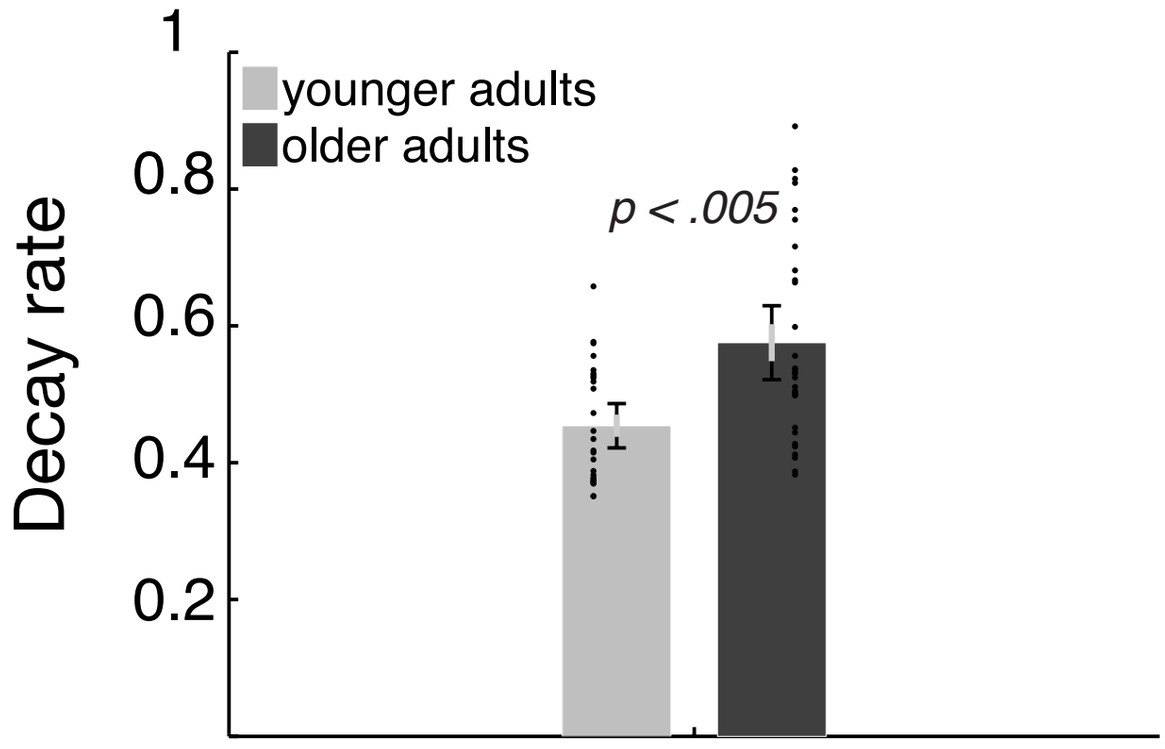


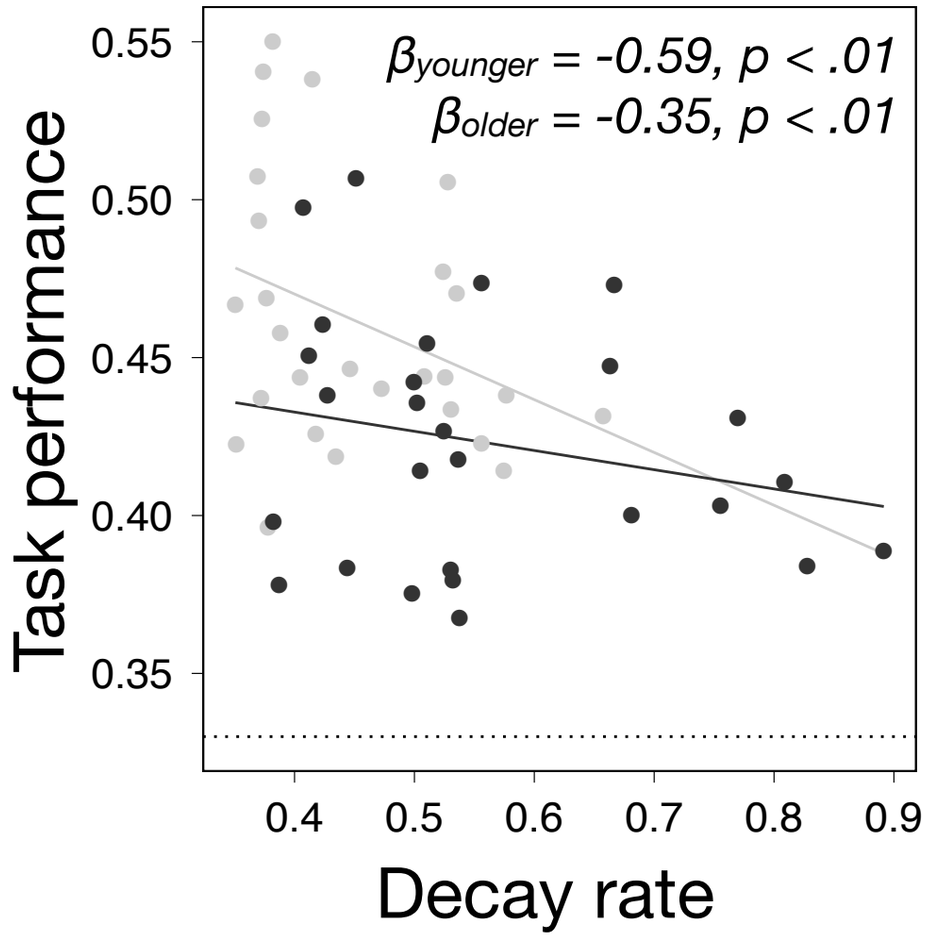

Broad focus of attention

Radulescu et al. 2016

# Results: group comparison

Narrow focus of attention

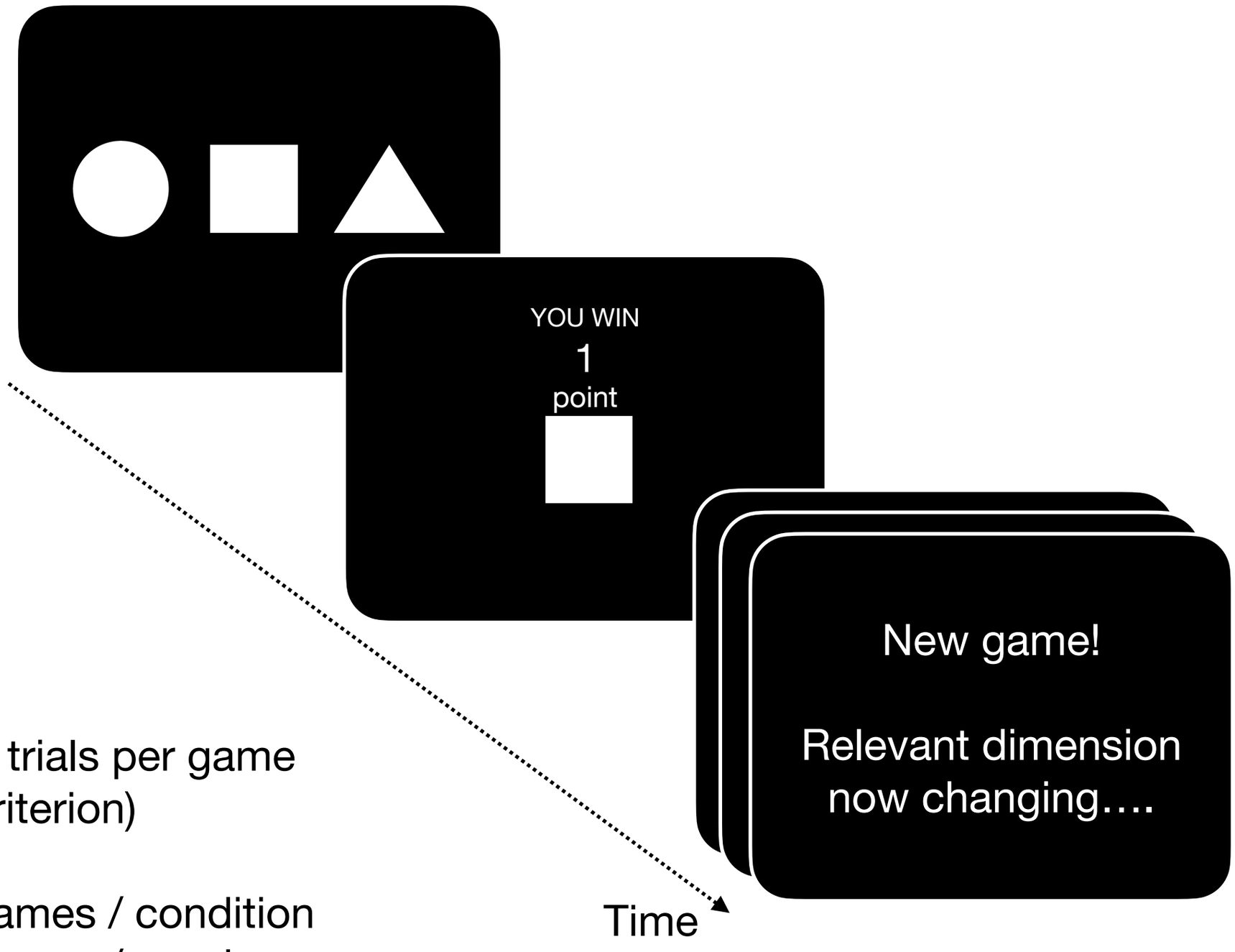



Broad focus of attention

Radulescu et al. 2016

# Component processes of representation learning

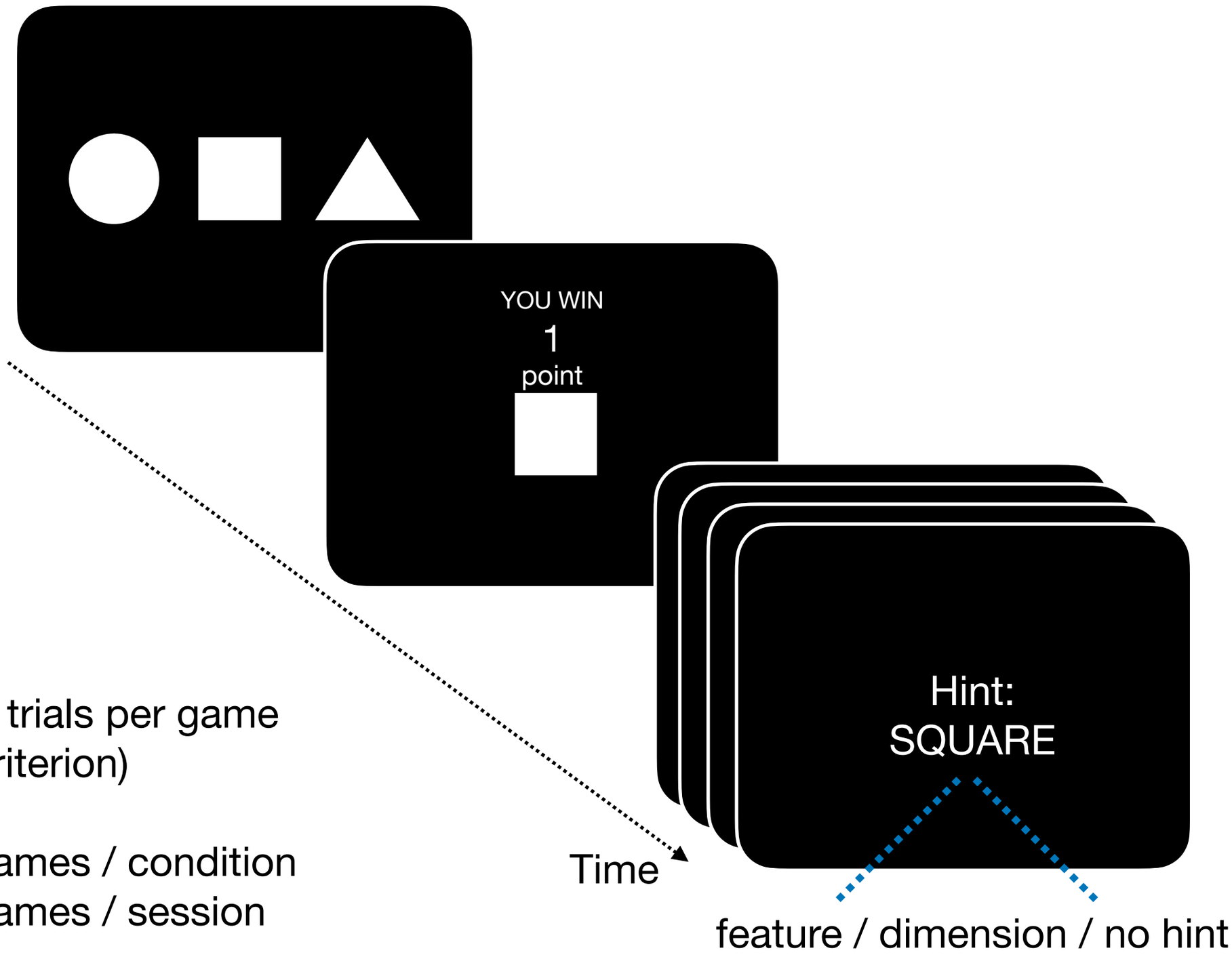



8-25 trials per game
(to criterion)

10 games / condition
50 games / session

Radulescu et al. 2016

# Component processes of representation learning

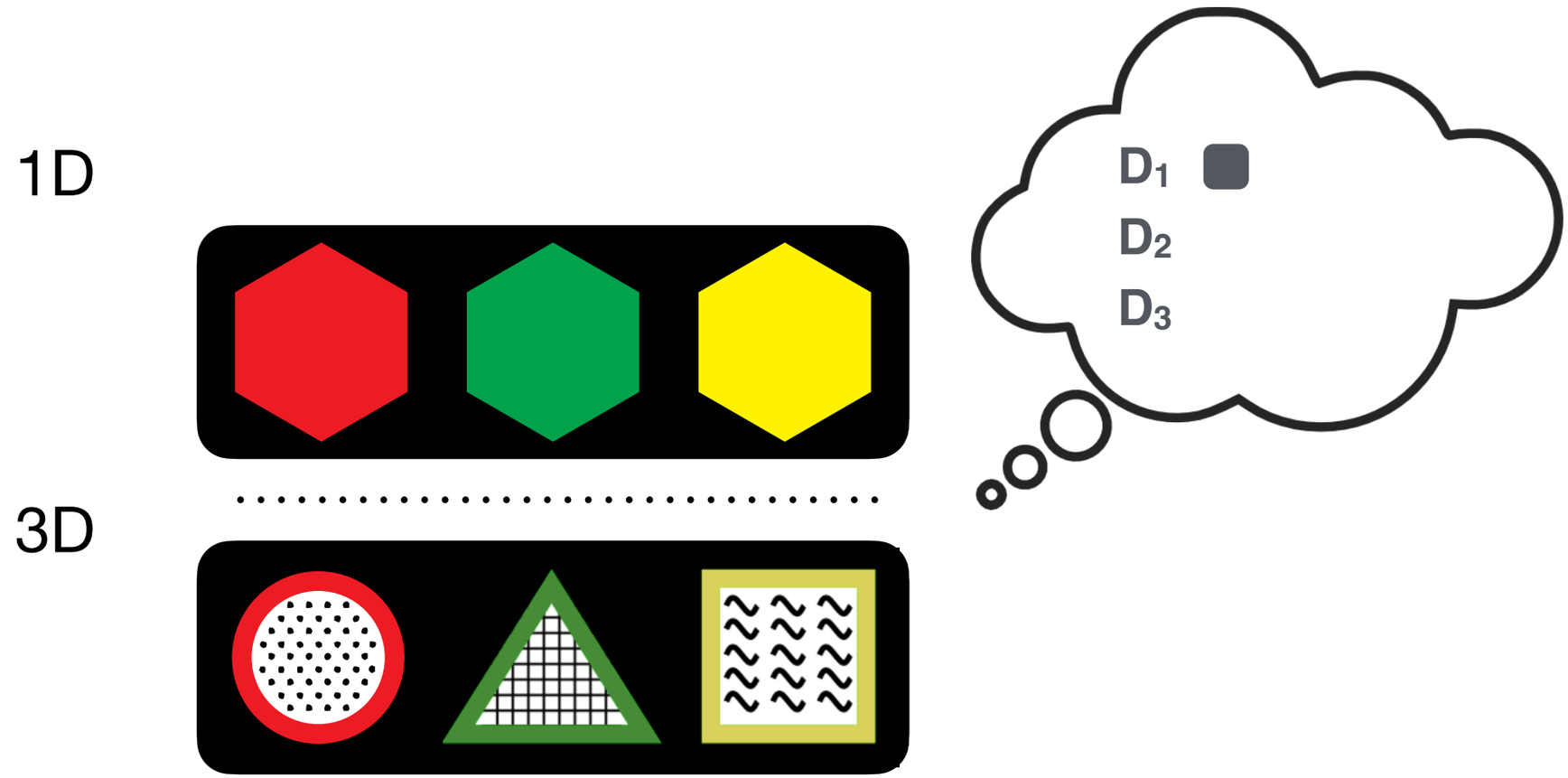



8-25 trials per game
(to criterion)

10 games / condition
50 games / session

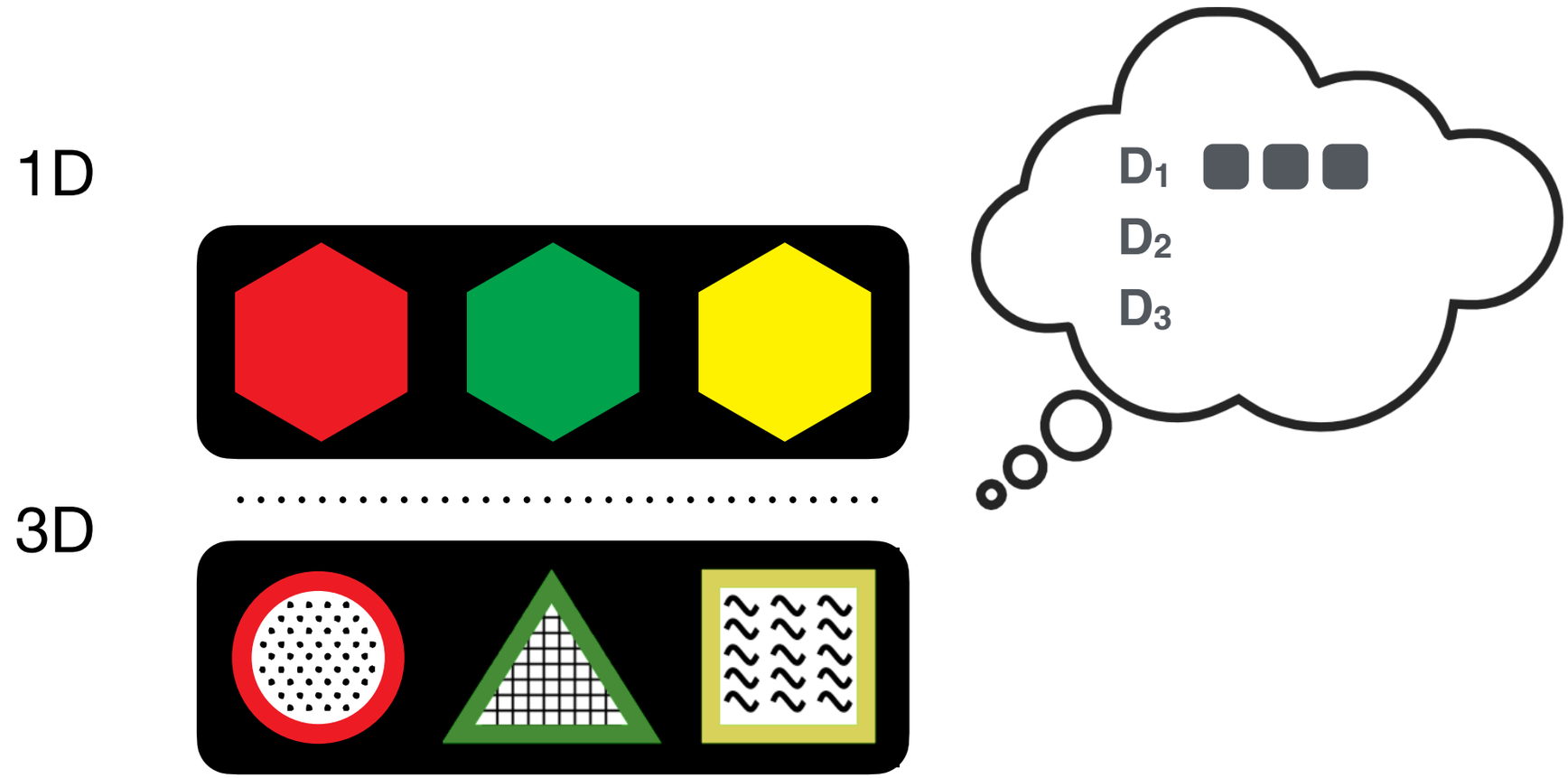
Radulescu et al. 2016

# Component processes of representation learning



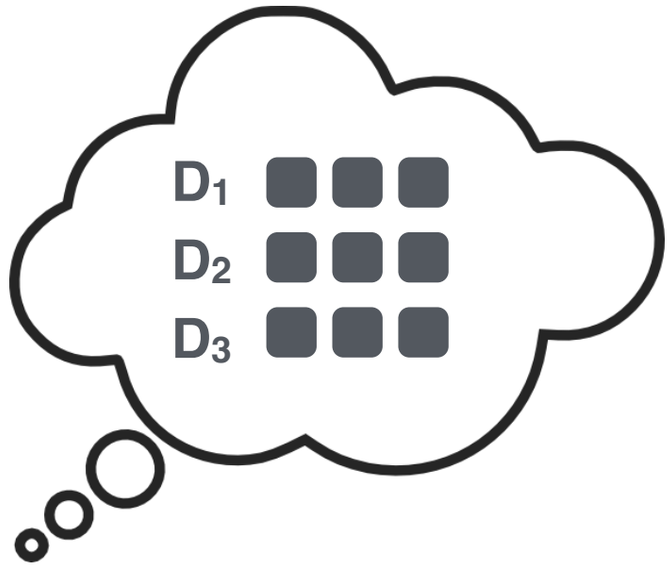

**Hint: RED**

# Component processes of representation learning



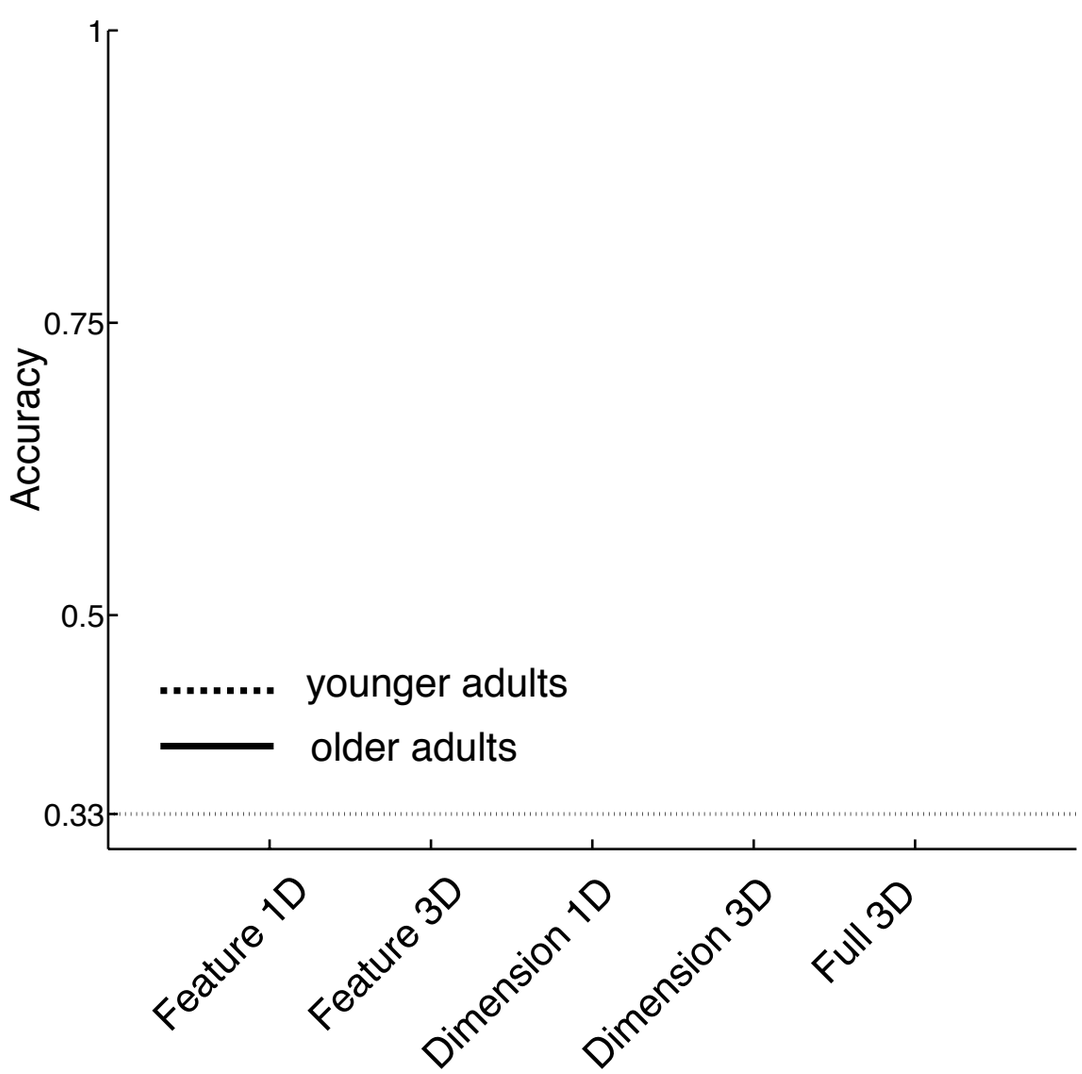

**Hint: COLOR**

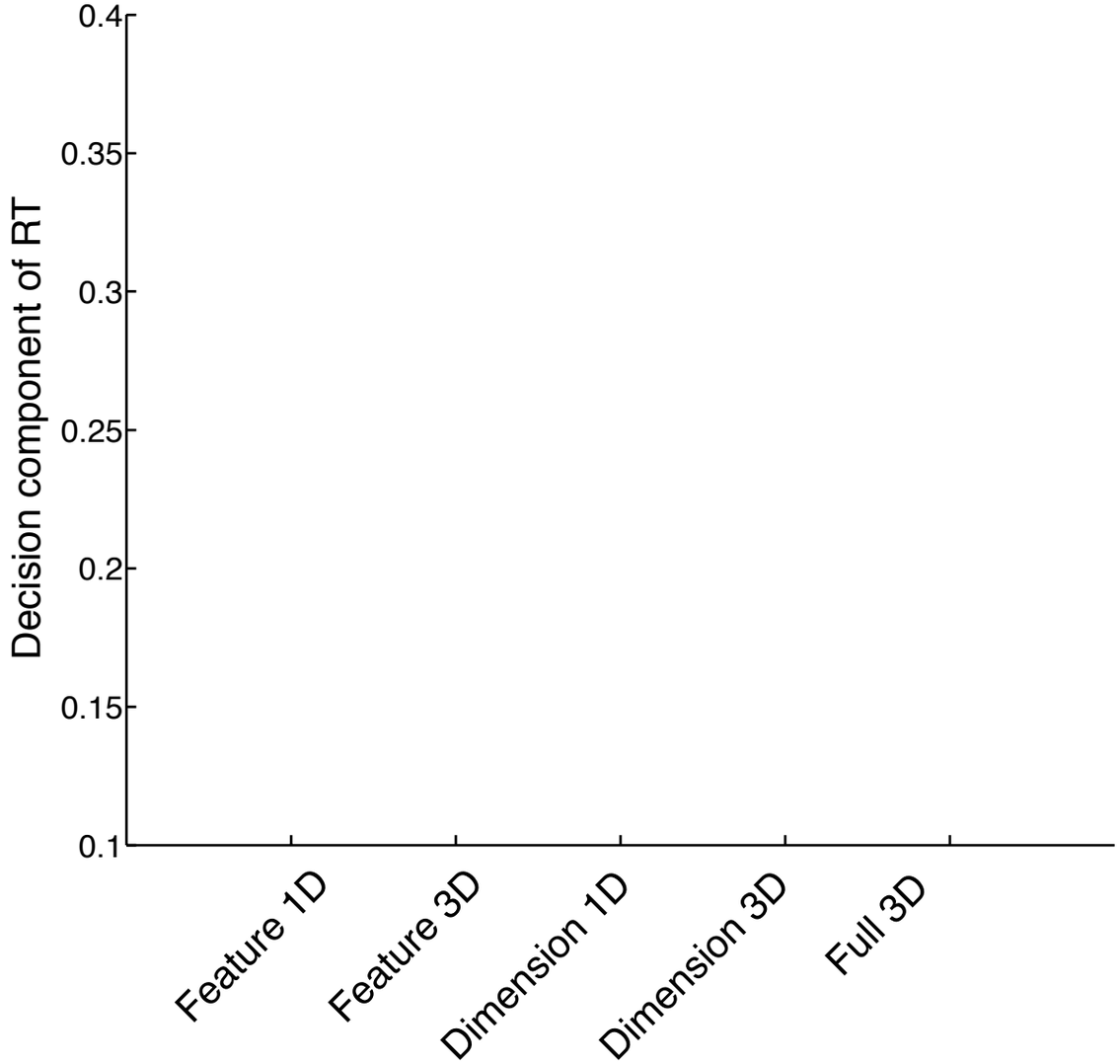
# Component processes of representation learning




3D

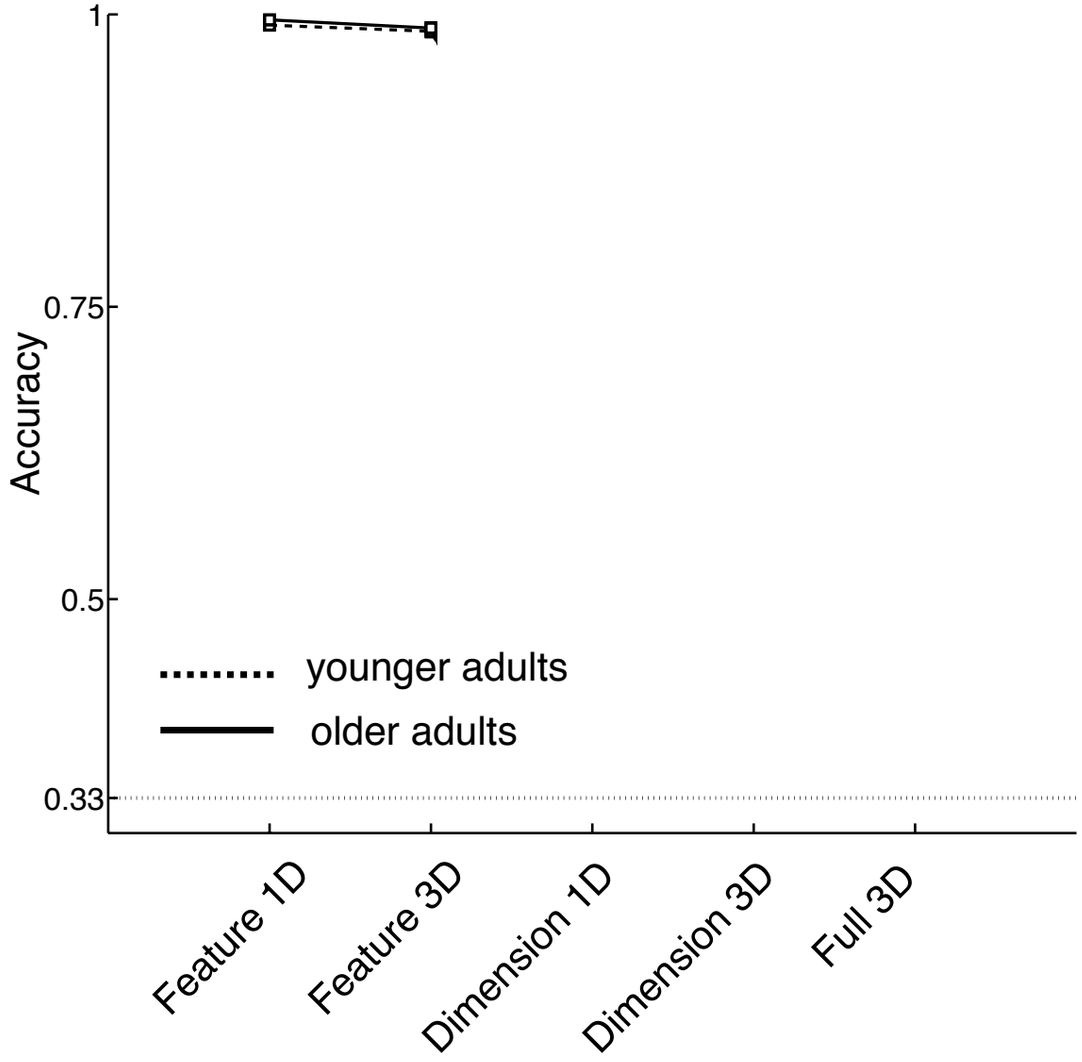
**Hint: NO HINT**

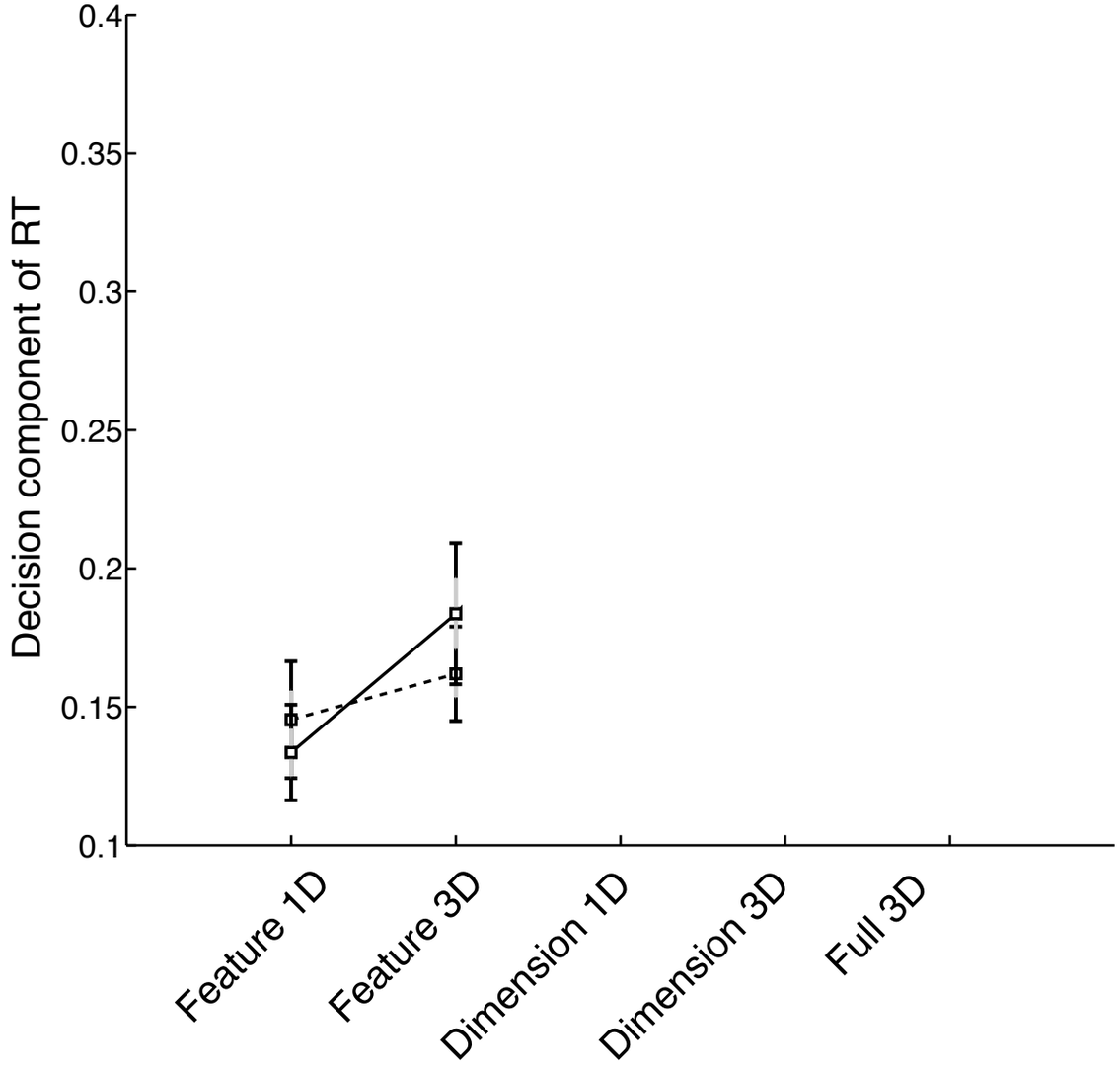



Radulescu et al. 2016

# Extra aging cost of ignoring distractors in RT (but not accuracy)



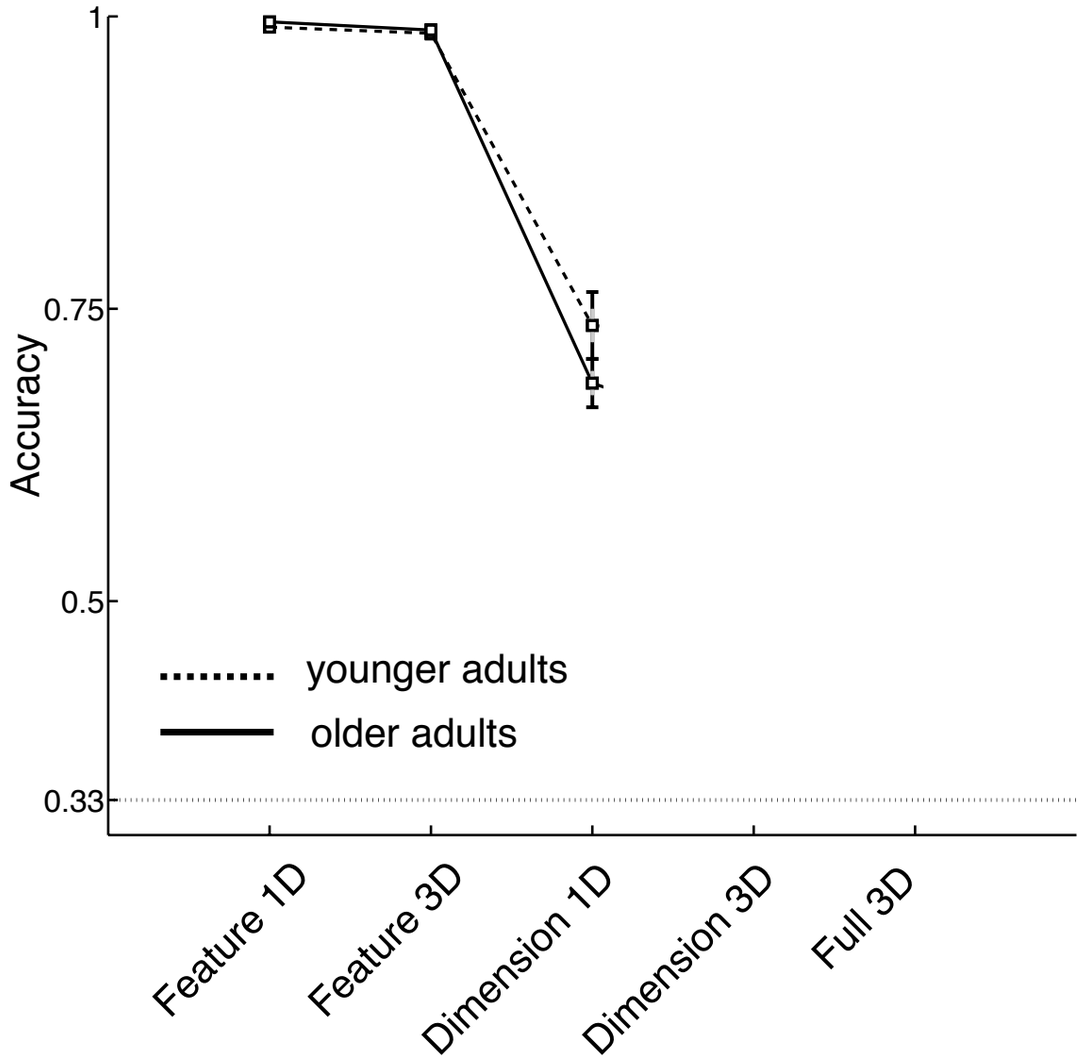



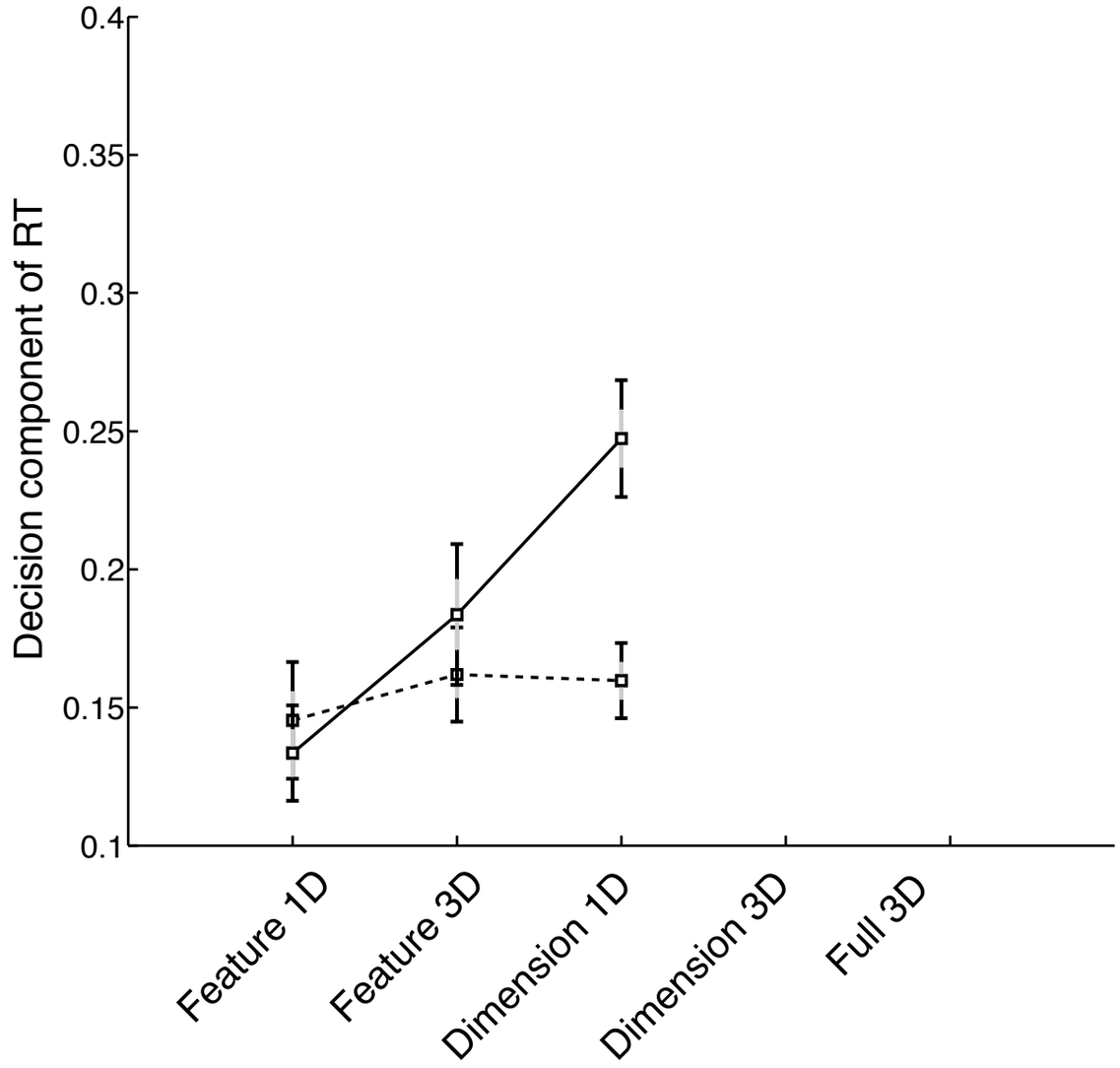

**Hint: RED**

**Hint: RED**

Radulescu et al. 2016

# Extra aging cost of learning



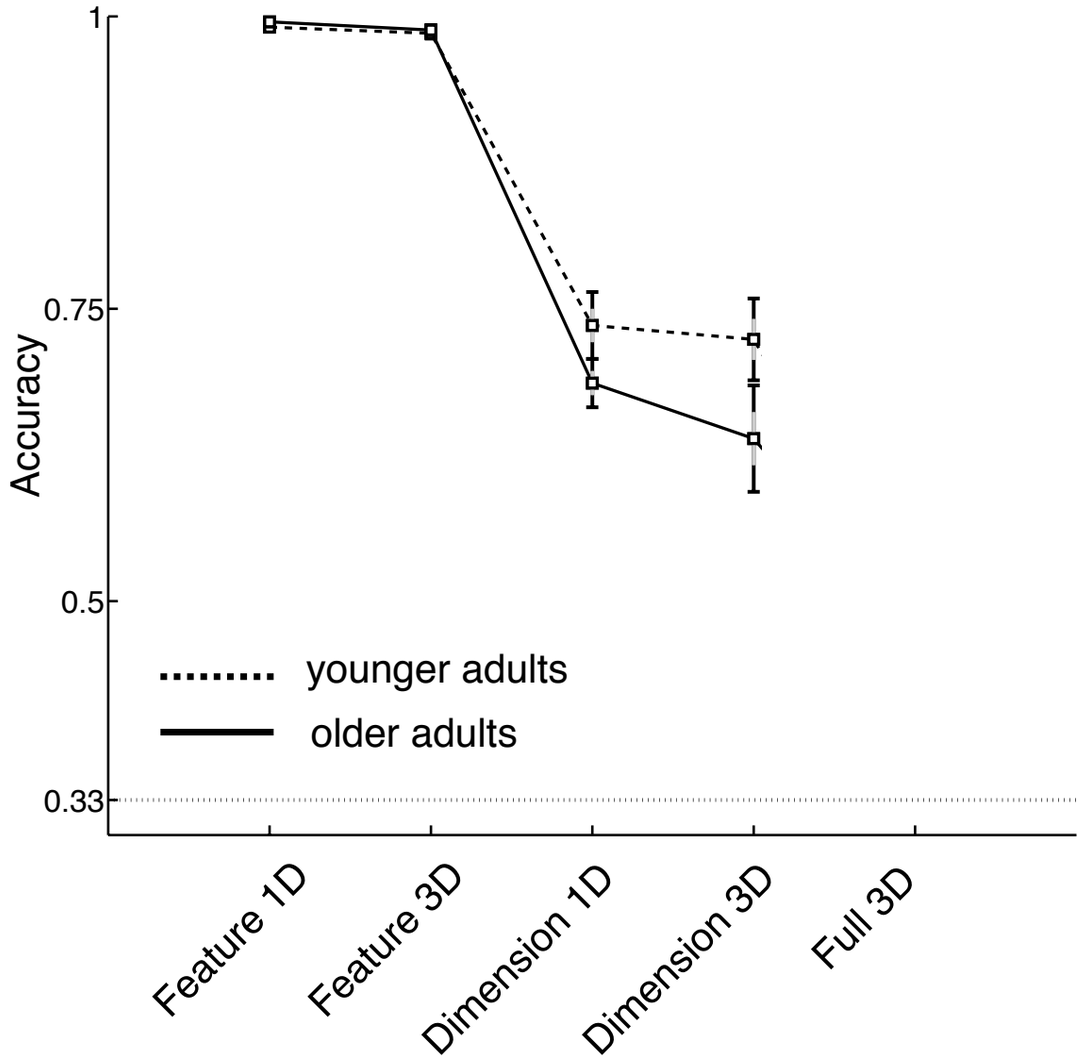

**Hint: RED**

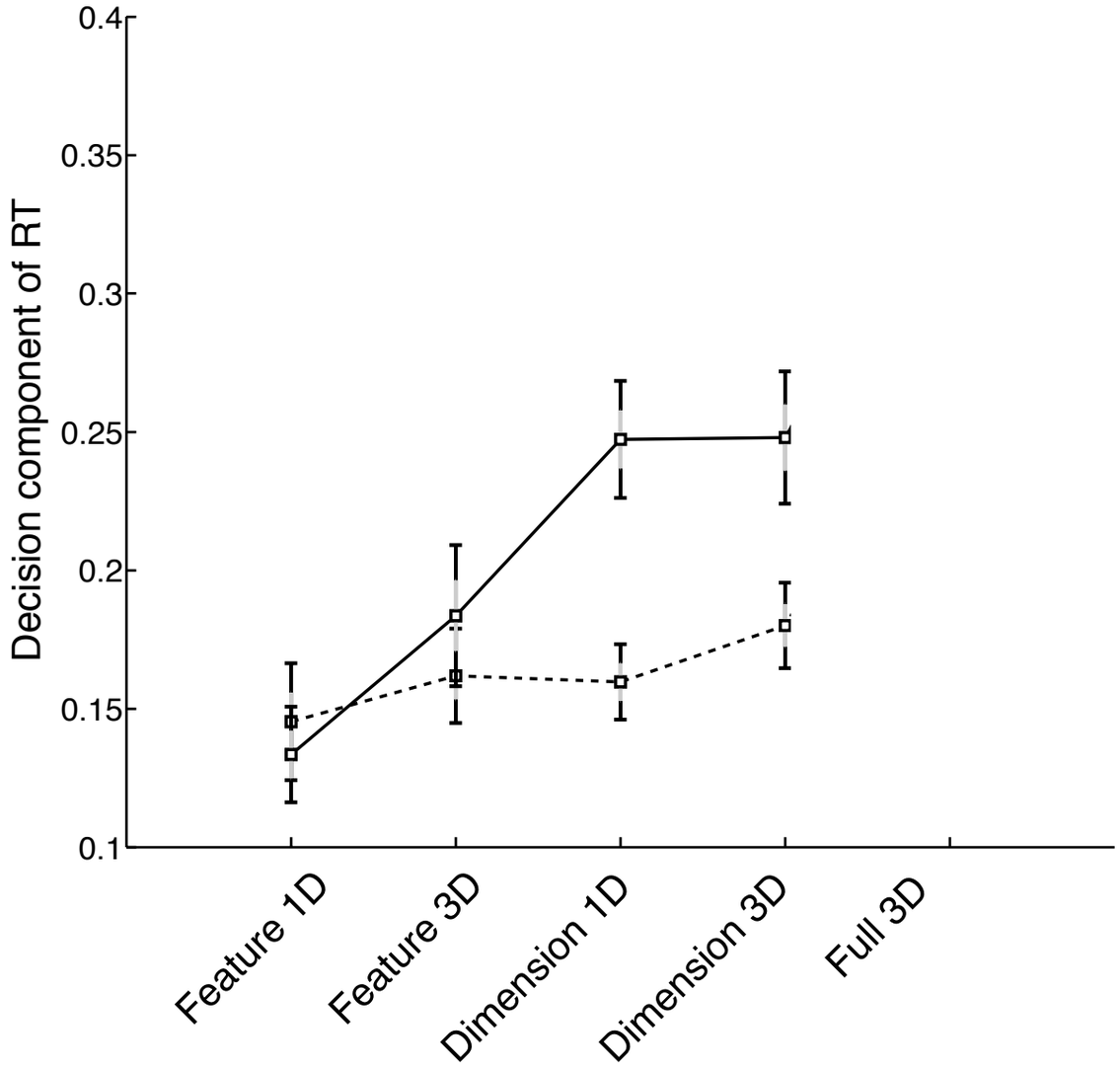
**Hint: RED**

**Hint: COLOR**

Radulescu et al. 2016

# Extra aging cost of selective attention




**Hint: RED**

**Hint: RED**

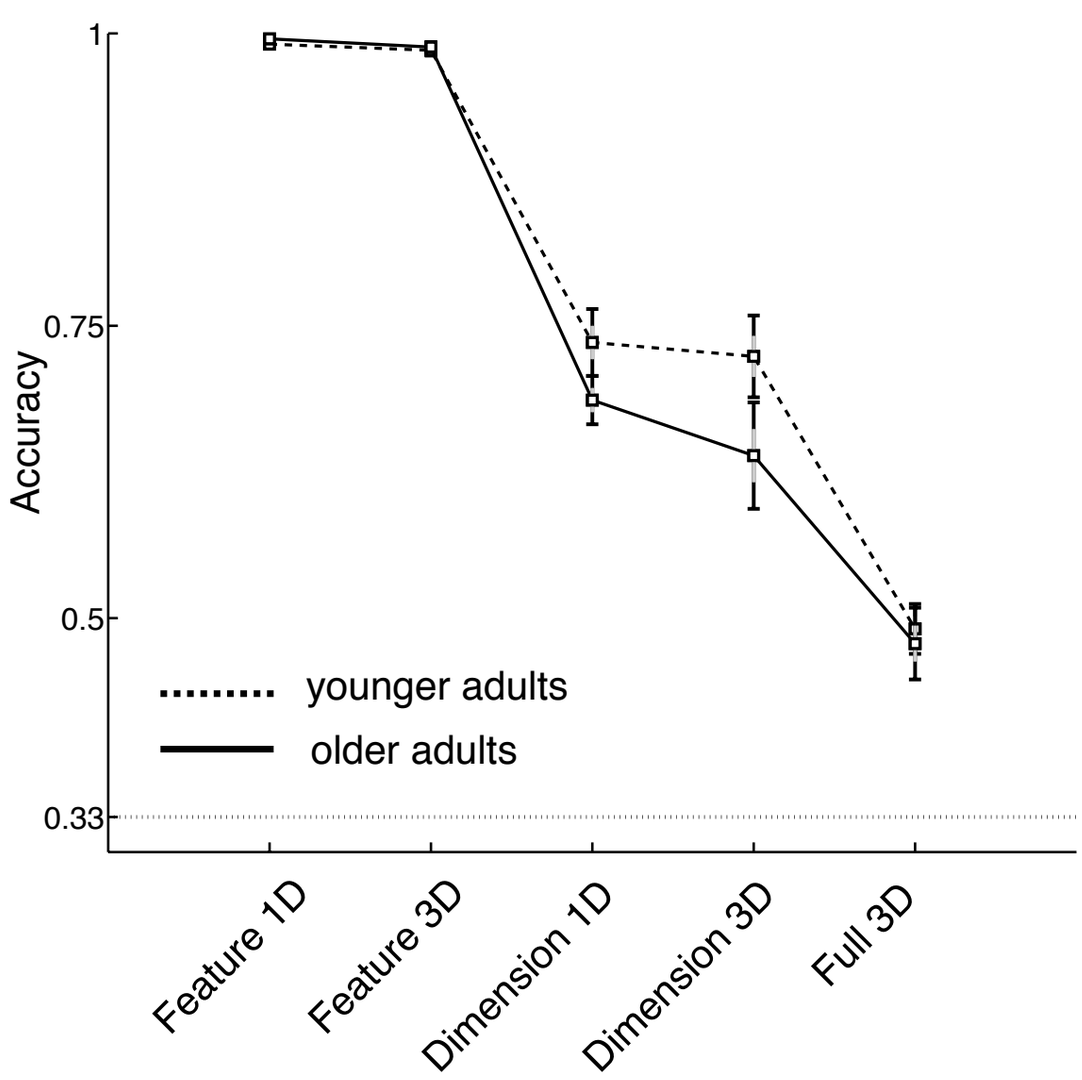
**Hint: COLOR**

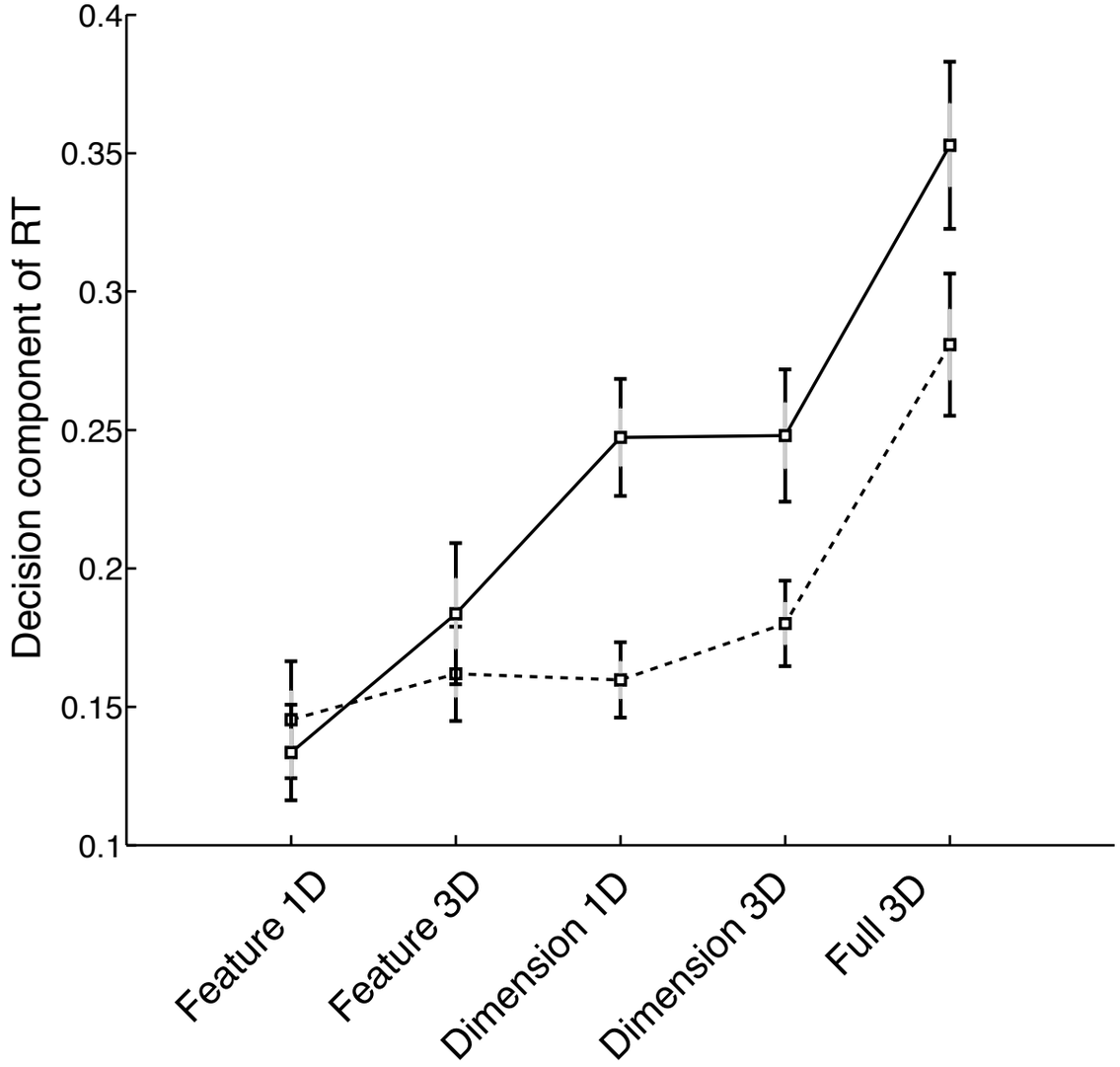
**Hint: COLOR**

Radulescu et al. 2016

# Age-differences in RT, but not accuracy during representation learning

(accuracy differences in separate experiment with more data and “No hint” games only)




**Hint: RED**

**Hint: RED**

**Hint: COLOR**

**Hint: COLOR**

**No hint**

Radulescu et al. 2016

# Conclusions

- Reinforcement learning as a model of cognitive dynamics
- Bridges algorithm and implementation
- Can be used to reveal latent age-related differences in cognitive processes
  - Narrower focus of attention during learning in older adults
  - Normative explanation: more beneficial to overgeneralize with more experience

# More examples of RL in aging

- Bolenz, F., Kool, W., Reiter, A. M., & Eppinger, B. (2019). Metacontrol of decision-making strategies in human aging. Elife, 8, e49154.

- Chowdhury, R., Guitart-Masip, M., Lambert, C., Dayan, P., Huys, Q., Düzel, E., & Dolan, R. J. (2013). Dopamine restores reward prediction errors in old age. Nature neuroscience, 16(5), 648-653.

- Cutler, J., Wittmann, M. K., Abdurahman, A., Hargitai, L. D., Drew, D., Husain, M., & Lockwood, P. L. (2021). Ageing is associated with disrupted reinforcement learning whilst learning to help others is preserved. Nature communications, 12(1), 4440.

- Daniel, R., Radulescu, A., & Niv, Y. (2020). Intact reinforcement learning but impaired attentional control during multidimensional probabilistic learning in older adults. Journal of Neuroscience, 40(5), 1084-1096.

- Hämmerer, D., Li, S. C., Müller, V., & Lindenberger, U. (2011). Life span differences in electrophysiological correlates of monitoring gains and losses during probabilistic reinforcement learning. Journal of Cognitive Neuroscience, 23(3), 579-592.

- Radulescu, A., Daniel, R., & Niv, Y. (2016). The effects of aging on the interaction between reinforcement learning and attention. Psychology and aging, 31(7), 747.

- van de Vijver, I., Ridderinkhof, K. R., Harsay, H., Reneman, L., Cavanagh, J. F., Buitenweg, J. I., & Cohen, M. X. (2016). Frontostriatal anatomical connections predict age-and difficulty-related differences in reinforcement learning. Neurobiology of aging, 46, 1-12.

# Thank you!


**Mt. Sinai Center for Computational Psychiatry:** Laura Berner, Vincenzo Fiore, Xiaosi Gu, Ignacio Saez, Daniela Schiller, et al.

**Lab:** Jacqueline Beltrán, Catherine Kim, Jing Li, Christina Maher, Marjorie Xie

**Collaborators:** Daniel Bennett, Fred Callaway, Tom Griffiths, James Hillis, Laurel Morris, Bas van Opheusden, Sam Zorowitz