

Reinforcement Learning in the Brain

Logan Cross, Postdoctoral Scholar

Decision Neuroscience: How do brains learn to make decisions?



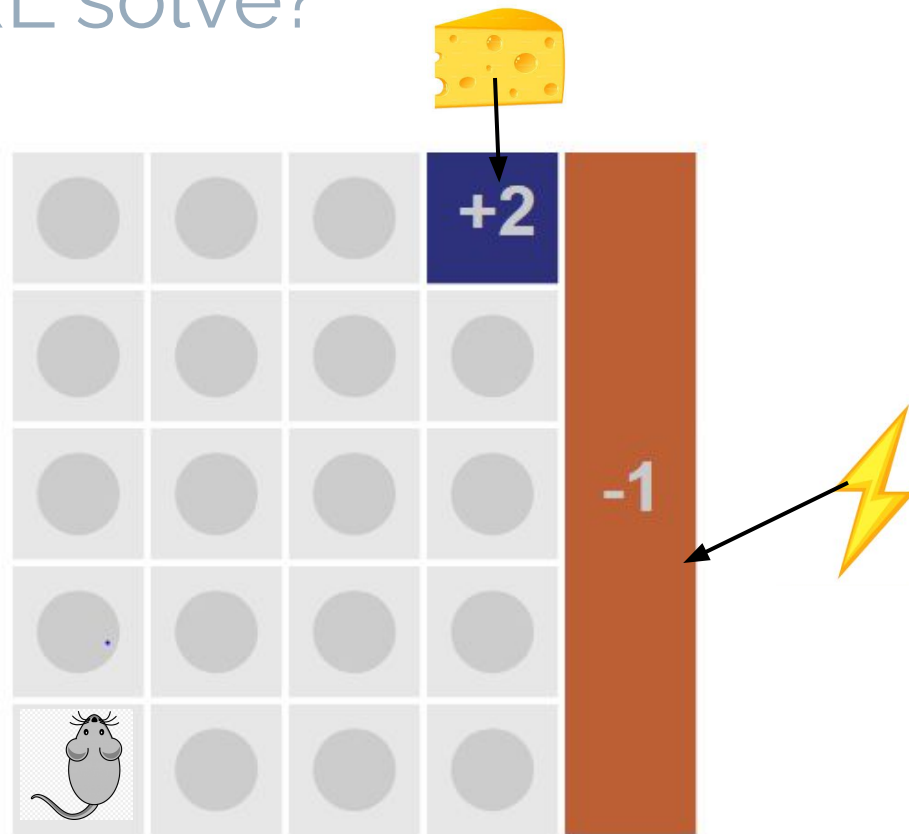
Reinforcement Learning (RL)

- ▷ RL - a poster child of influential crosstalk between AI and decision neuroscience

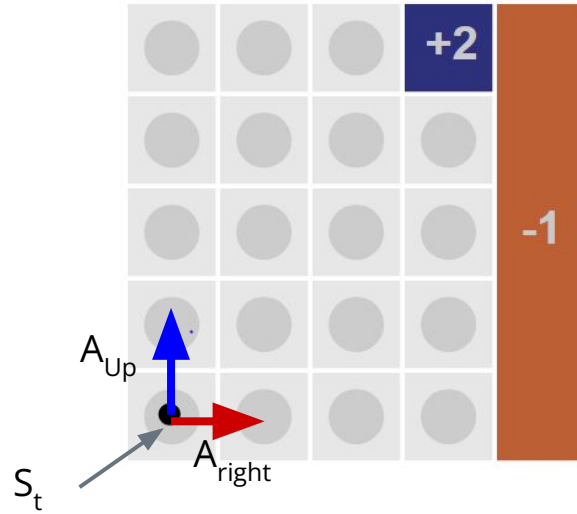
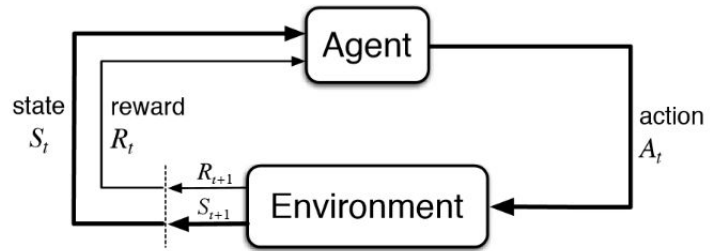


What problem does RL solve?

**Computational
problem: Mapping
states to actions**



Temporal Difference Learning



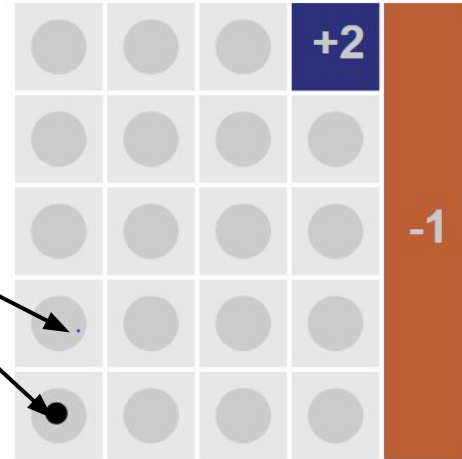
Temporal Difference Learning

Estimate the value of being in every state

$V(s)$ = discounted sum of expected future reward

$$V(s_t) \leftarrow r_t + \gamma V(s_{t+1})$$

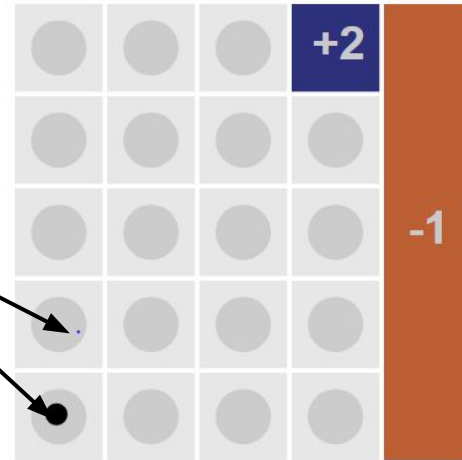
State value Reward Next state value



Temporal Difference Learning

Estimate the value of
being in every state
 $V(s)$

Update the value through
prediction errors

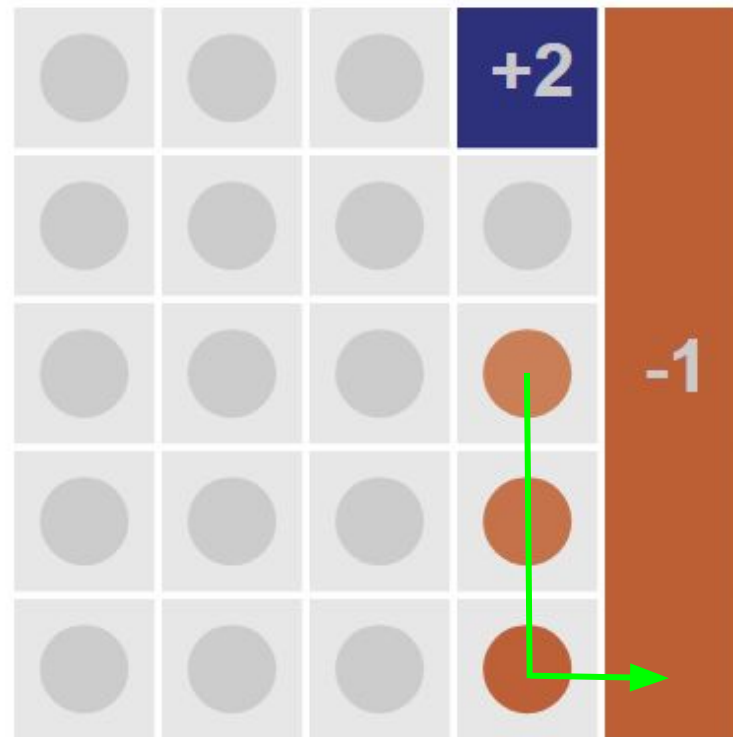


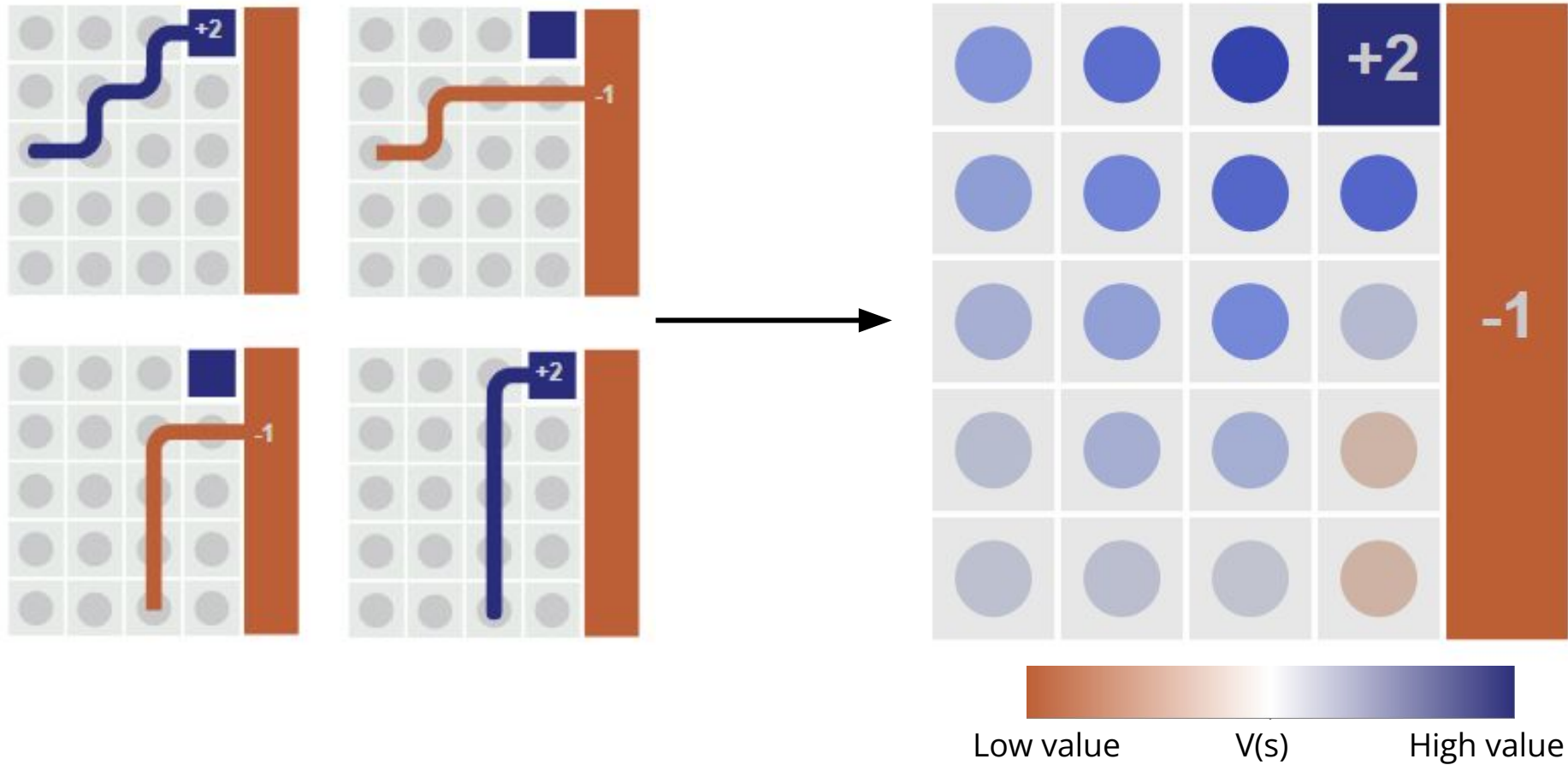
$$\text{TD Learning } V(S_t) \leftarrow V(S_t) + \alpha [R_{t+1} + \gamma V(S_{t+1}) - V(S_t)]$$

Previous estimate Reward t+1 Discounted value on the next step TD Target

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

Previous estimate
Reward t+1
Discounted value on the next step
TD Target





Q-Learning

- ▷ Turn a value function into a decision-making policy
- ▷ Learn an action-value function $Q(s,a)$ for every state-action pair, take the action with the highest Q-value

$Q(s, a)$



$$Q(S_t, A_t) = R_{t+1} + \gamma \max_a Q(S_{t+1}, a)$$

Q-Learning

- ▷ Turn a value function into a decision-making policy
- ▷ Learn an action-value function $Q(s,a)$ for every state-action pair, take the action with the highest Q-value
- ▷ Update Q-values with TD prediction errors

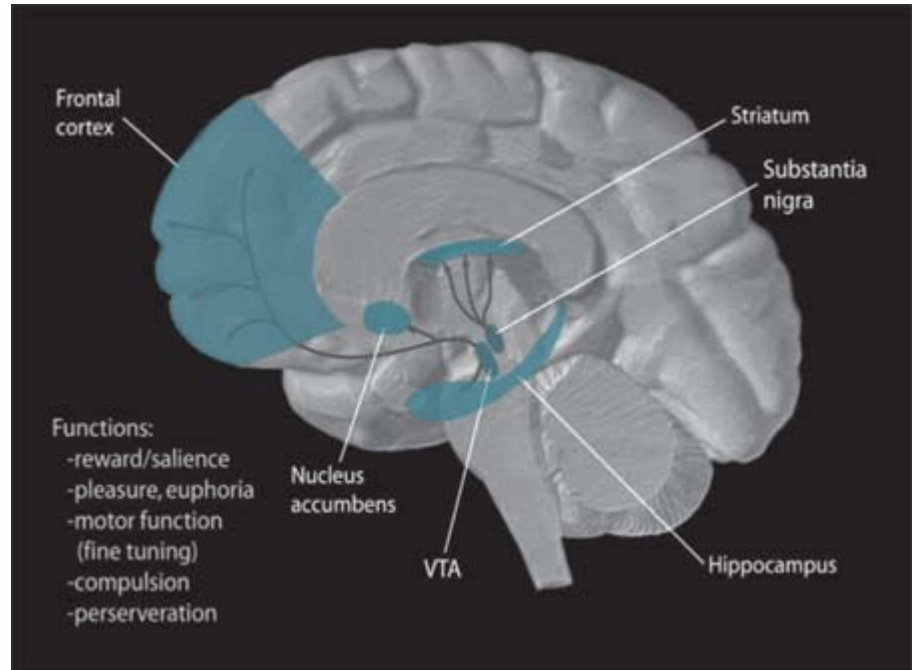
$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha [R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)]$$

The diagram illustrates the Q-learning update equation with color-coded components and labels:

- New Q-value estimation** (green bar) is represented by $Q(S_t, A_t)$ on the left side of the equation.
- Former Q-value estimation** (blue bar) is represented by $Q(S_t, A_t)$ on the right side of the equation.
- Learning Rate** (red bar) is represented by α .
- Immediate Reward** (orange bar) is represented by R_{t+1} .
- Discounted Estimate optimal Q-value of next state** (purple bar) is represented by $\gamma \max_a Q(S_{t+1}, a)$.
- TD Target** (teal bar) is represented by the sum $R_{t+1} + \gamma \max_a Q(S_{t+1}, a)$.
- TD Error** (yellow bar) is represented by the entire right-hand side of the equation: $[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)]$.

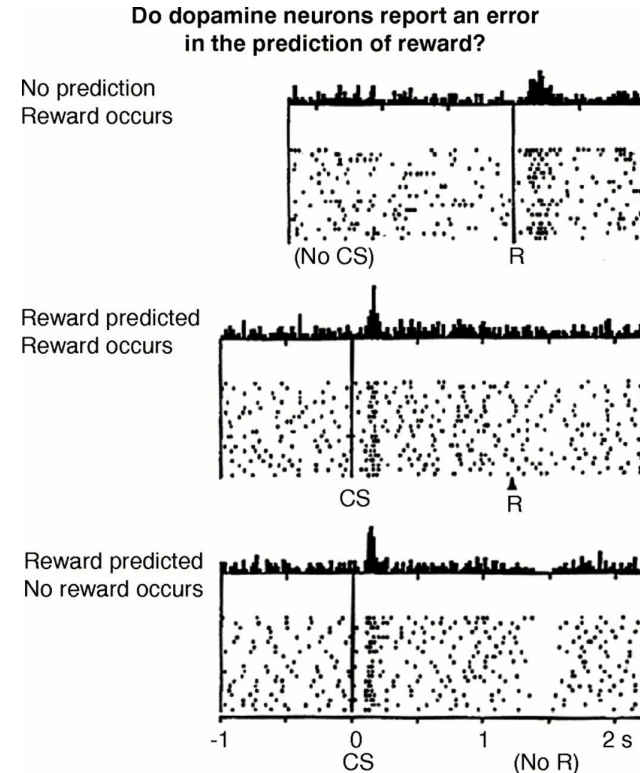
Dopamine and the reward circuit

- ▶ Dopamine neurons in the Ventral Tegmental Area (VTA) signal reward
- ▶ Addictive drugs act on these dopamine regions
- ▶ VTA projects to striatum/nucleus accumbens and frontal cortex



Evidence for TD Learning signals in the brain

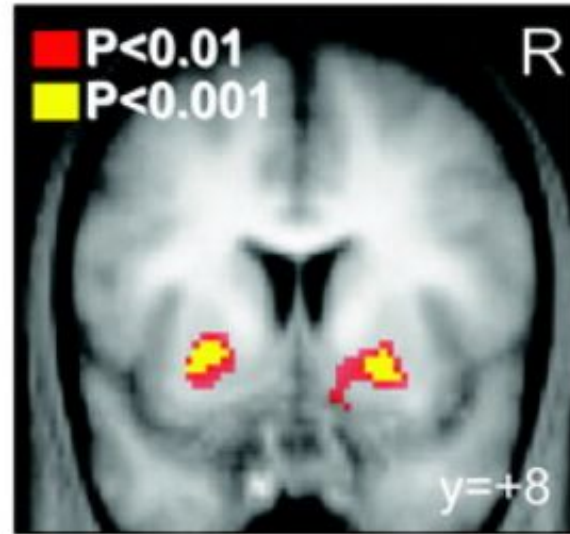
- ▷ Dopamine neuron responses reflect prediction errors as in TD learning algorithm



Evidence for TD Learning signals in the brain

Correlates of prediction error signals are found in the striatum of humans using fMRI

Image from O'Doherty et al., 2004

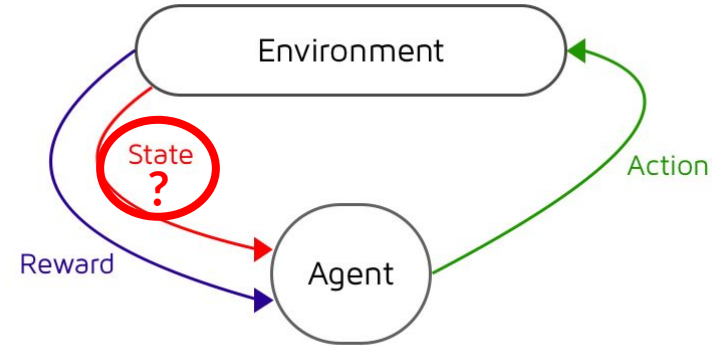


Classic RL in the brain

- ▷ RL theories describe algorithmic solutions for selecting actions to maximize long-term reward
- ▷ Evidence found for implementation of these algorithms in the brain

Classic RL breaks in environments with real-world complexity

- ▷ In the real-world states are not discrete
 - They are continuous and high-dimensional
- ▷ Animals, humans, and robots have to use perception to know where they are
- ▷ Can't learn about every state
 - Have to generalize



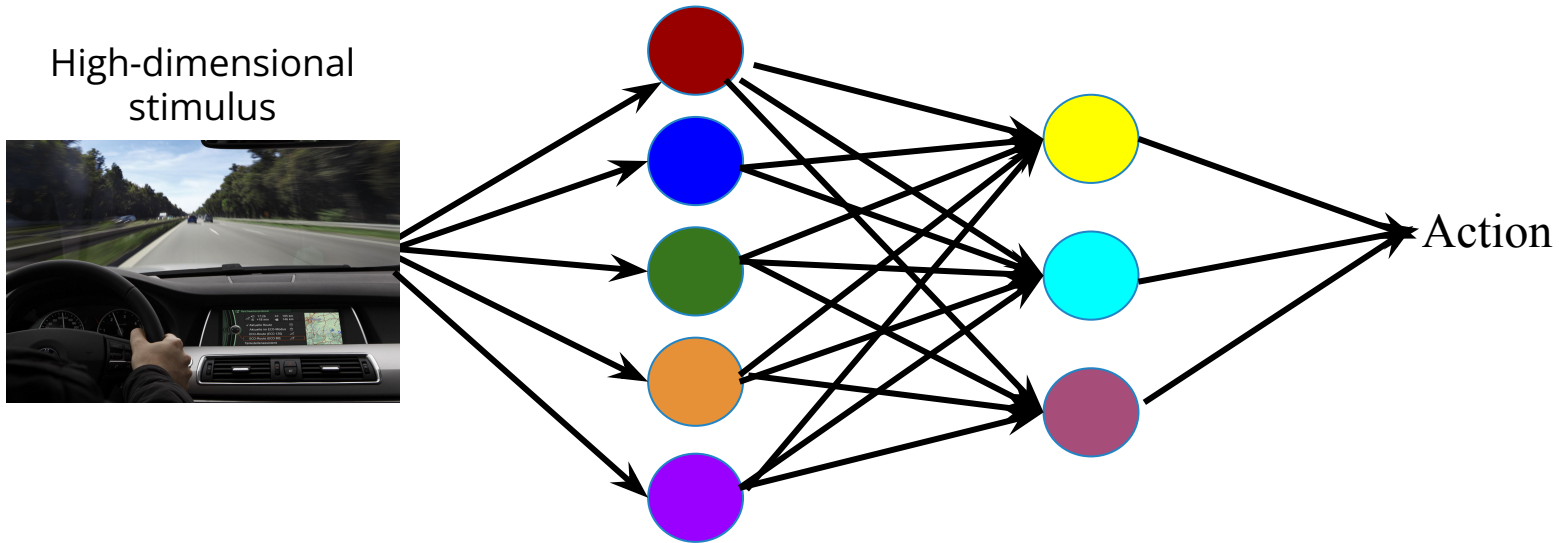
What does the brain do in environments of real-world complexity?

- ▷ Classic RL theory defines a new state for any change in sensory input
- ▷ But, humans can drive a car in new environments seamlessly, even with novel sensory input
- ▷ The brain must construct a lower dimensional, more abstract state space that allows it to generalize to new environments and conditions

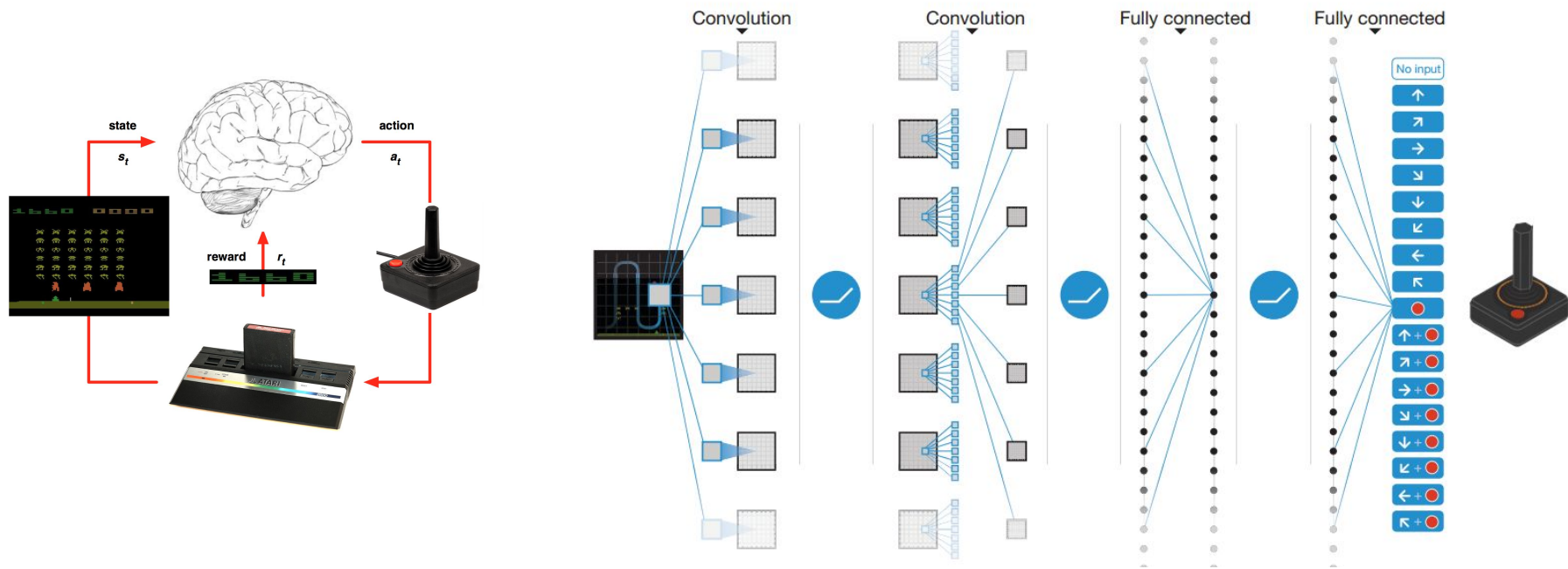


What does the brain do in environments of real-world complexity?

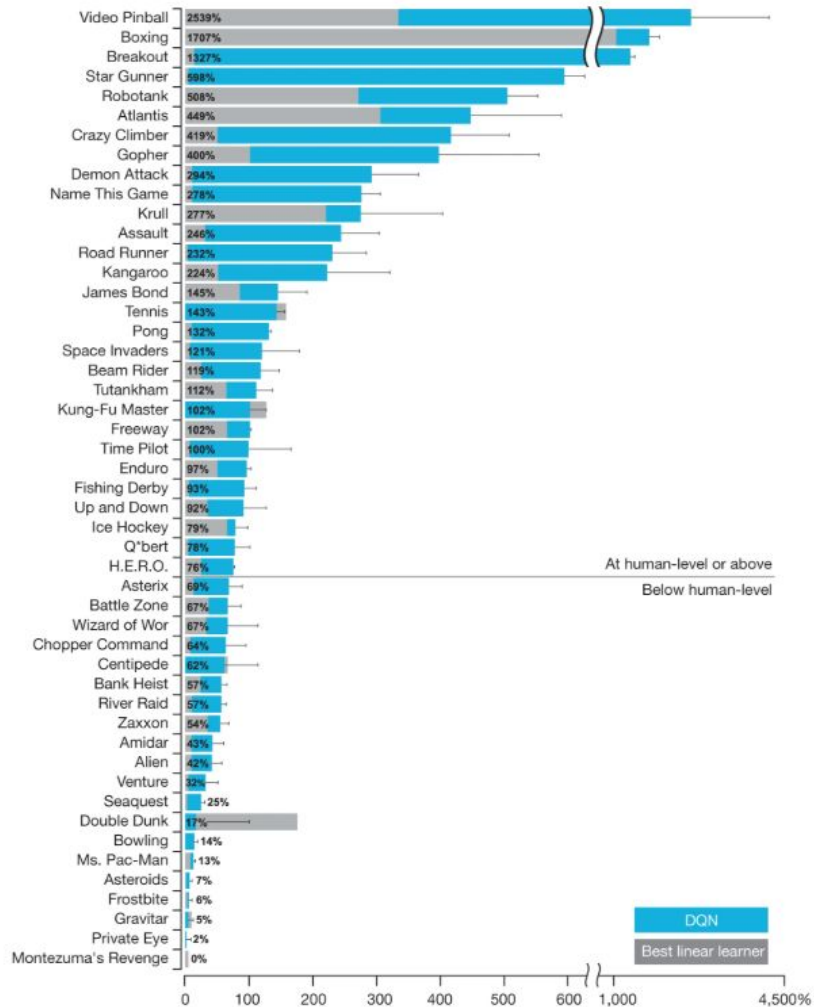
- ▷ The mapping of input -> action becomes highly non-linear
- ▷ Can we get inspiration from modern deep learning tools?



Atari video games and Deep-Q-Network as a model for this problem

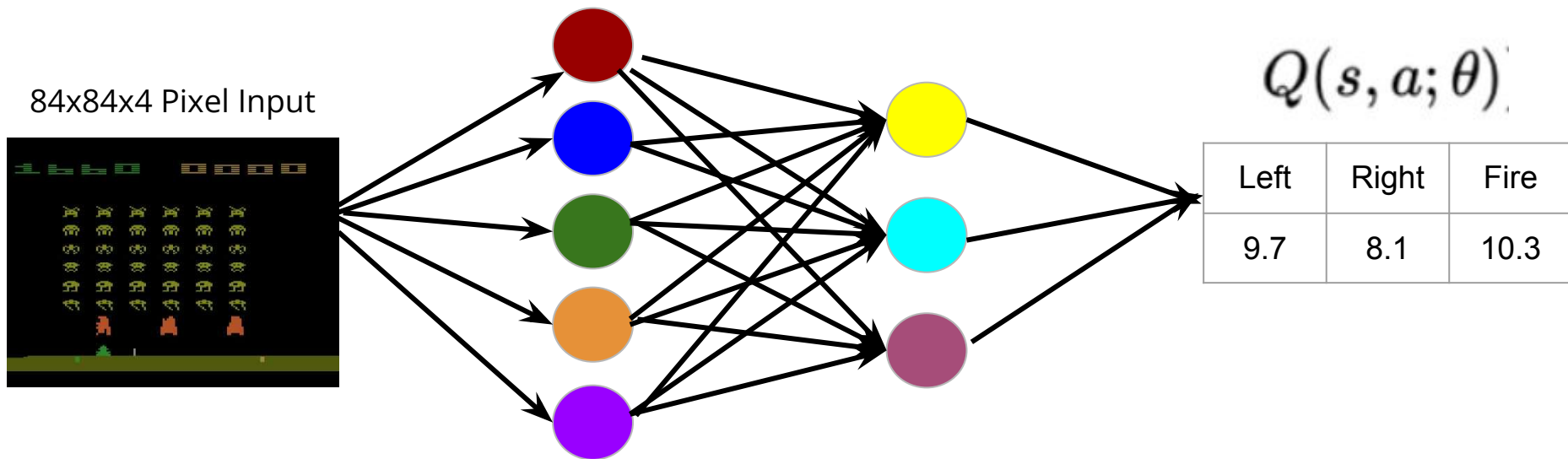


From Mnih et al., 2015



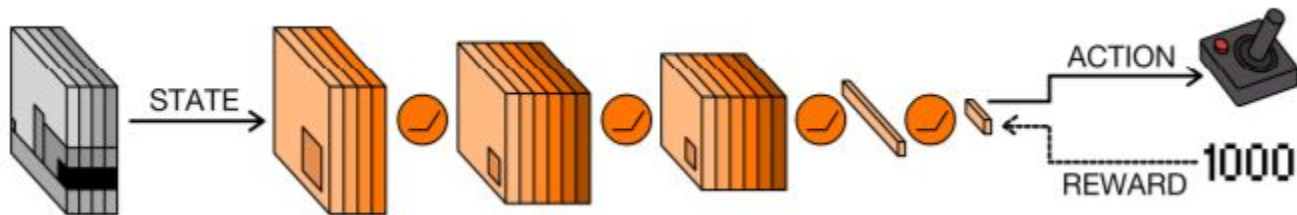
Deep-Q-Network

- Key idea: use a deep neural network to approximate the Q-value function



Deep-Q-Network

- ▶ Three convolutional layers -> one fully-connected layer -> output Q-value for every action
- ▶ Take action with the highest Q-value
 - Take random action with ϵ probability for exploration (ϵ -greedy)



Deep-Q-Network

- ▷ To learn: use TD target and turn it into regression problem

$$\underbrace{r + \gamma \max_{a'} Q(s', a'; \theta_i^-)}_{\text{target}}$$

$$Loss = (r + \gamma \max_{a'} Q(s', a'; \theta') - Q(s, a; \theta))^2$$

Other Challenges that DQN solves

- ▷ We want the data to be i.i.d.
- ▷ But in RL, data points are highly correlated between time points



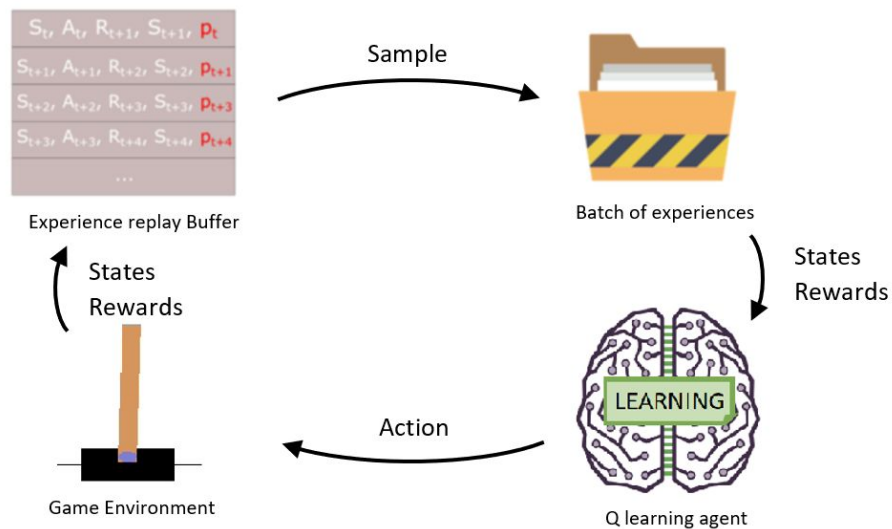
t



$t+1$

Experience Replay Buffer

- ▷ Run policy to collect experiences to store in replay buffer
 - (s, a, r, s')
- ▷ To train, randomly sample these experiences to decorrelate samples



Deep RL in the brain

- ▷ How does an AI, human, or animal evaluate actions in a high-dimensional environment?
 - Use deep learning to approximate the value function and do state representation
- ▷ **Is there a network in the brain that constructs state representations similarly to deep RL algorithms?**

Article

Neuron

Using deep reinforcement learning to reveal how the brain encodes abstract state-space representations in high-dimensional environments

Authors

Logan Cross, Jeff Cockburn,
Yisong Yue, John P. O'Doherty

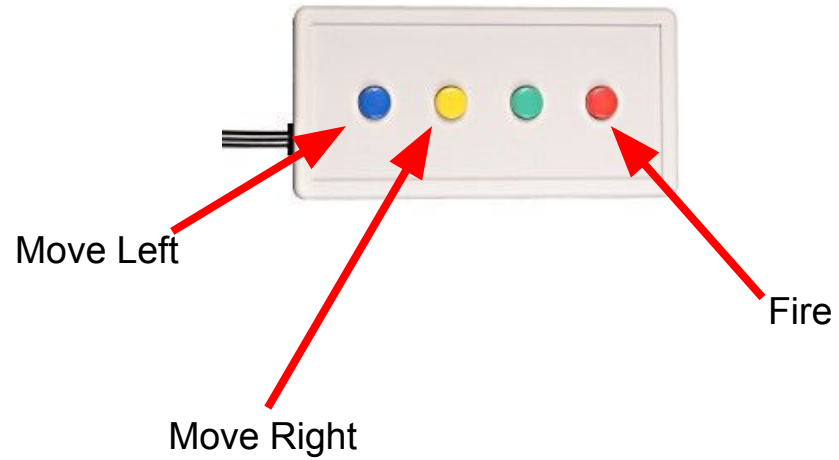
Experimental Design

- ▷ Subjects freely play Atari video games in the fMRI scanner
- ▷ Subjects are scanned in 4 separate days for a total of 4.5 hours of gameplay
 - N=6
- ▷ Games
 - Enduro
 - Pong
 - Space Invaders



Cross et al., 2021

Controller



Enduro

- ▷ You control a race car that must avoid other cars and go as fast as possible
- ▷ Pass 200 cars before the day is over
- ▷ Weather conditions change throughout the day



Pong

- ▷ You control the green paddle on the right
- ▷ Move the paddle up and down to hit the white ball
- ▷ Get the white ball past your opponent to be awarded 1 point



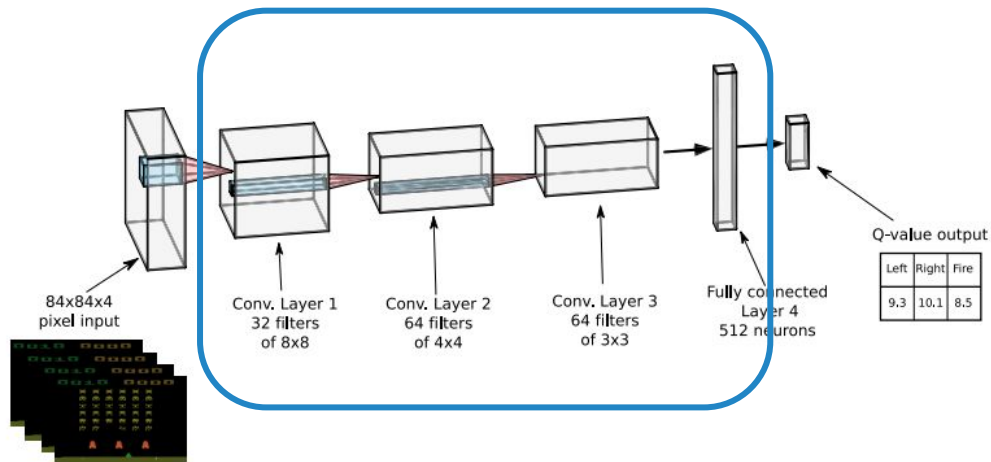
Space Invaders

- ▷ You control a ship that can move from left to right at the bottom of a screen
- ▷ You must destroy enemy ships above you and avoid being hit by missiles



Can DQN hidden layers model state-space representation?

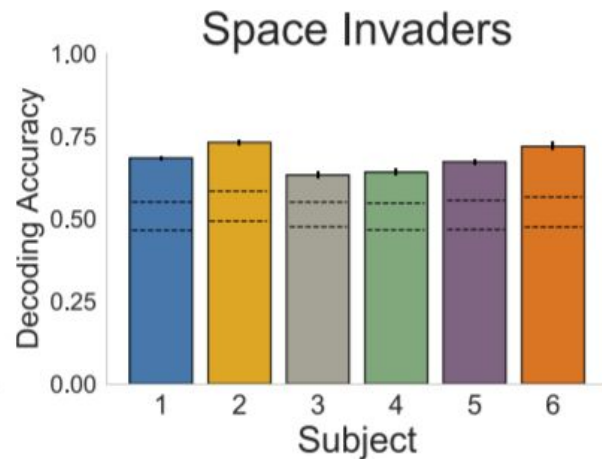
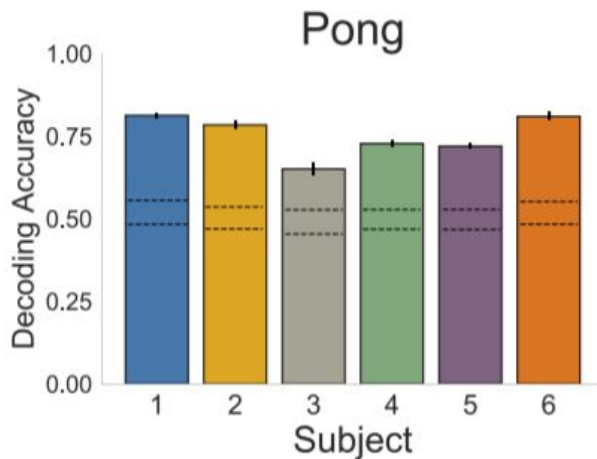
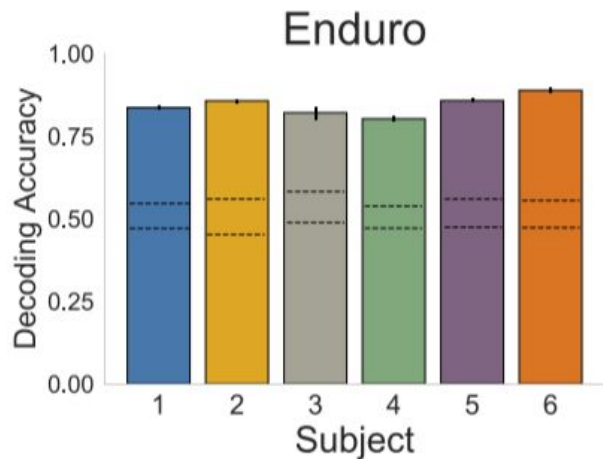
- ▶ Four hidden layers represent the internal state representations in DQN
- ▶ Can human behavior and brain activity be predicted from these layers?



Predicting human behavior using DQN hidden layers

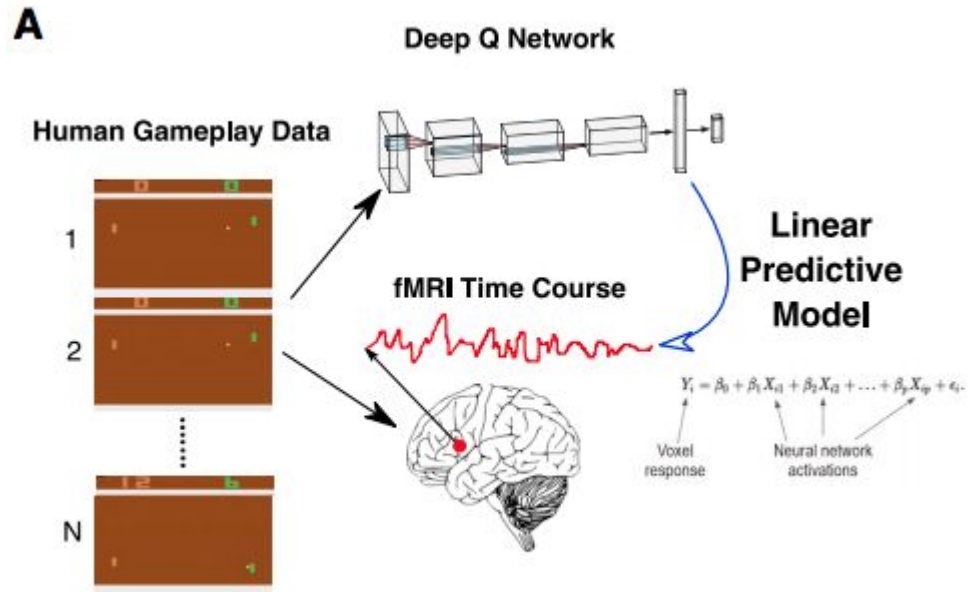
1. Run human gameplay data through trained DQN to produce stimulus features represented by the activations in the hidden layers
2. Take 100 principal components from each layer (400 total features)
3. Predict human left vs right actions from these features using logistic regression

Human Action Decoding Using DQN Hidden Layer Features



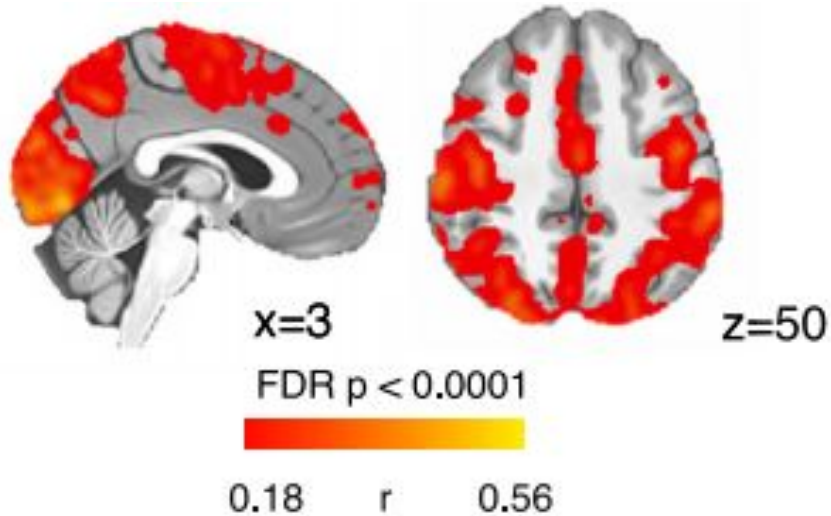
Predicting brain activity using DQN hidden layers - Encoding Model

- ▷ Use ridge regression to model the response of a voxel at each timepoint as a linear combination of neural network activations

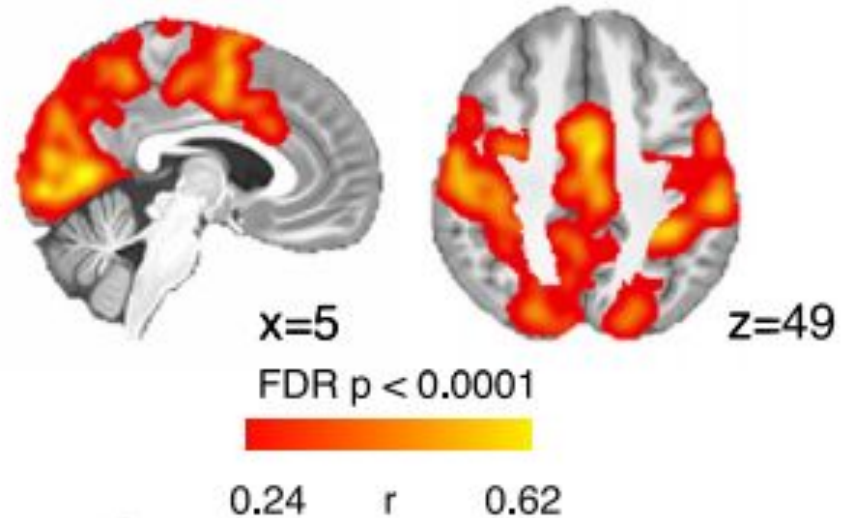


Pong - Encoding model results

Sub001



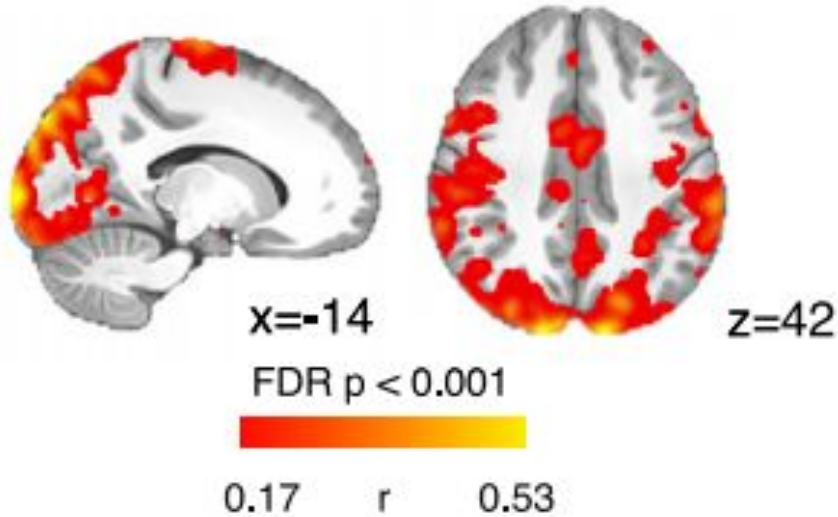
Sub006



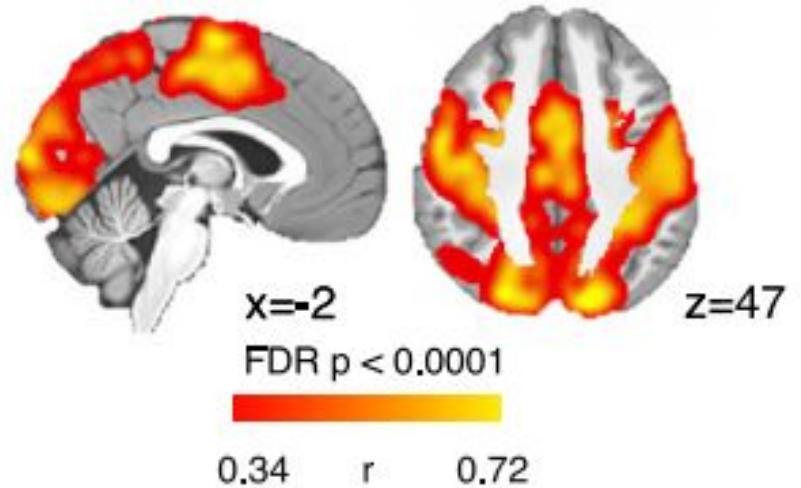
Pong regions: Distributed sensorimotor pathway extending from dorsal visual pathway, posterior parietal cortex (PPC) to premotor cortex

Enduro - Encoding model results

Sub001



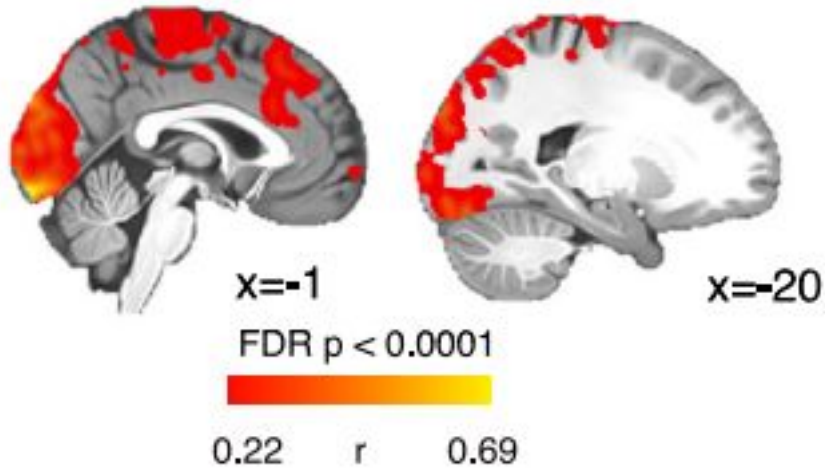
Sub006



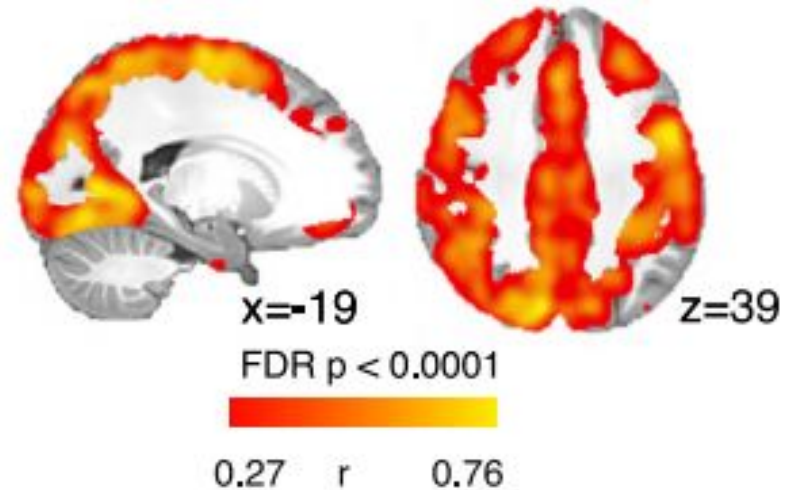
Enduro regions: Distributed sensorimotor pathway extending from dorsal visual pathway, posterior parietal cortex (PPC), and to premotor cortex

Space Invaders - Encoding model results

Sub001



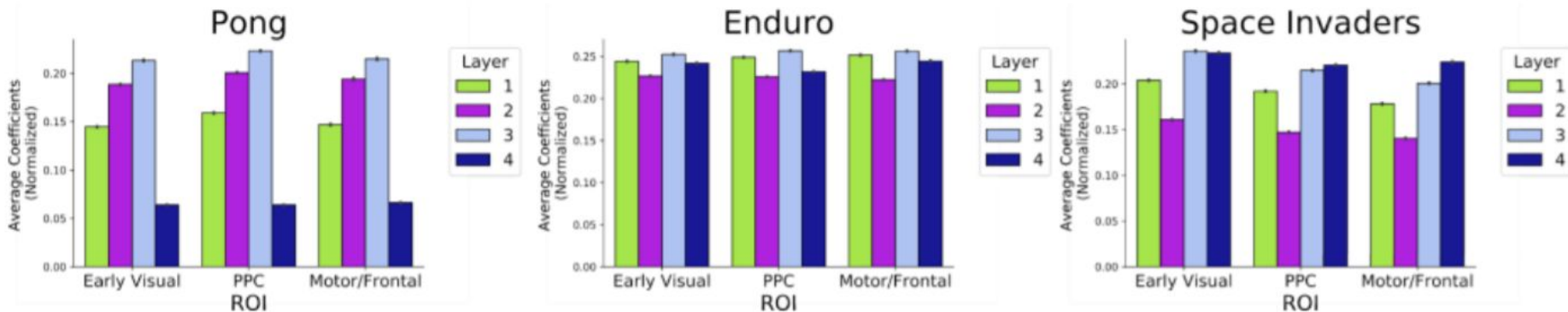
Sub006



Space Invaders regions: Distributed sensorimotor pathway extending from dorsal stream, PPC, and prefrontal cortex regions

Layers 3 & 4 have highest coefficients even in early visual cortex

Average Encoding Model Coefficient Magnitude By Layer



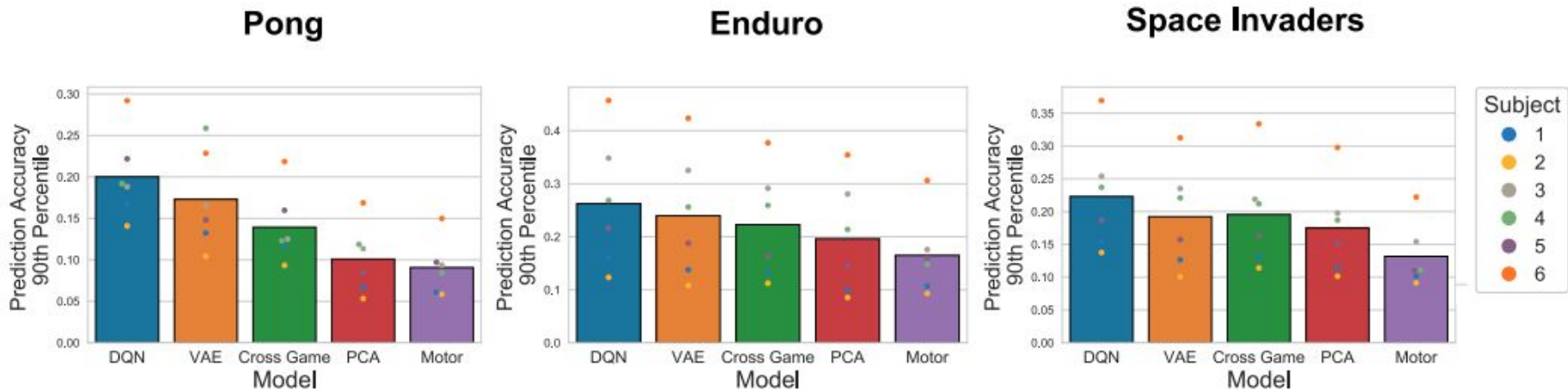
*PPC = posterior parietal cortex

Control Analyses

- ▷ Is DQN just picking up on basic visual and motor responses
- ▷ DQN tested against control models
 - Variational autoencoder (VAE)
 - DQN trained on another game
 - Principal components of pixel space
 - Motor regressors

DQN outperformed all control models for all subjects across games (except one game in one subject)

DQN Model vs. Control Models



Control Analyses

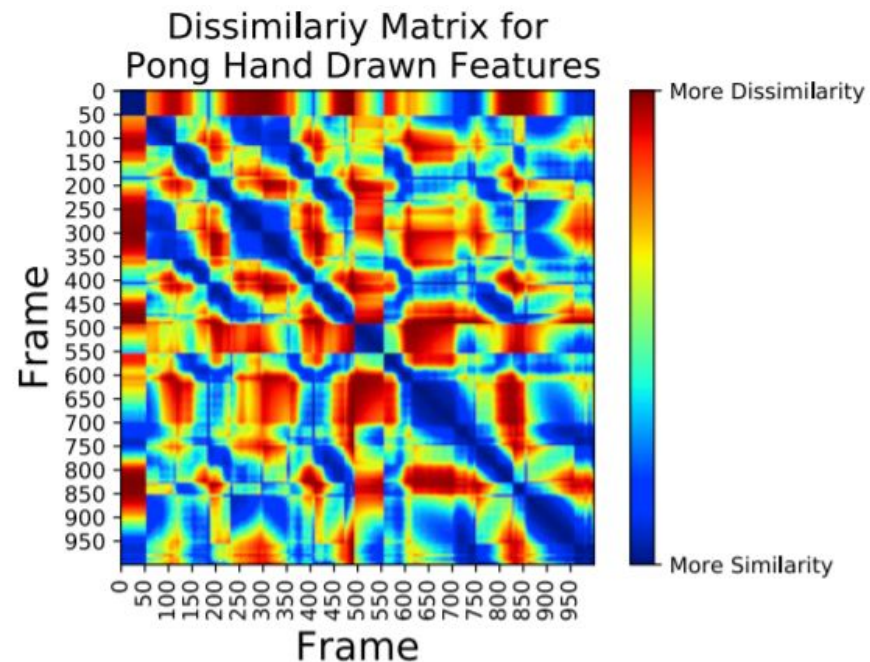
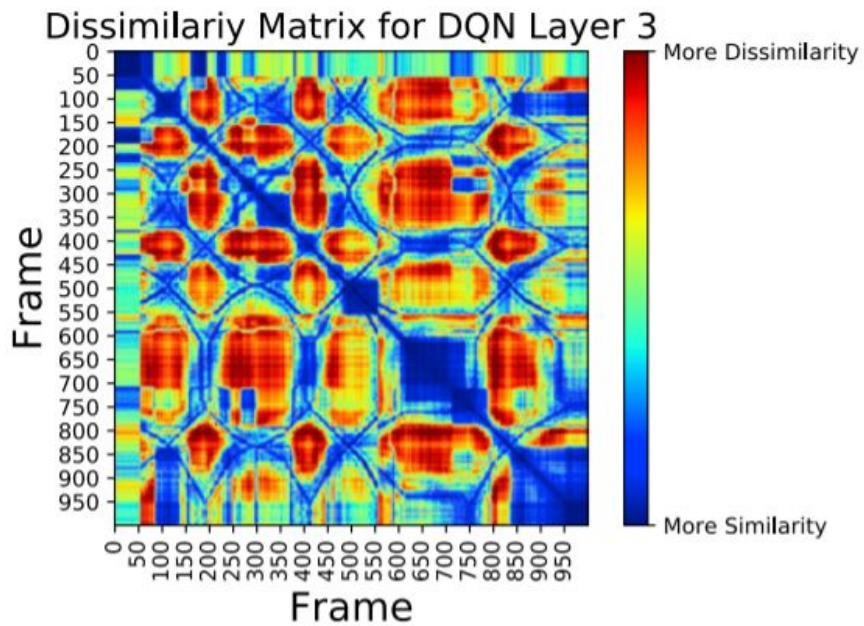
- ▷ Nonlinear feature representations outperformed linear ones
 - VAE and DQN two best models
- ▷ DQN outperforms VAE by linking perception to action and reward

What exactly is DQN encoding?

- ▷ We performed representational similarity analysis (RSA) on DQN to characterize its internal representations

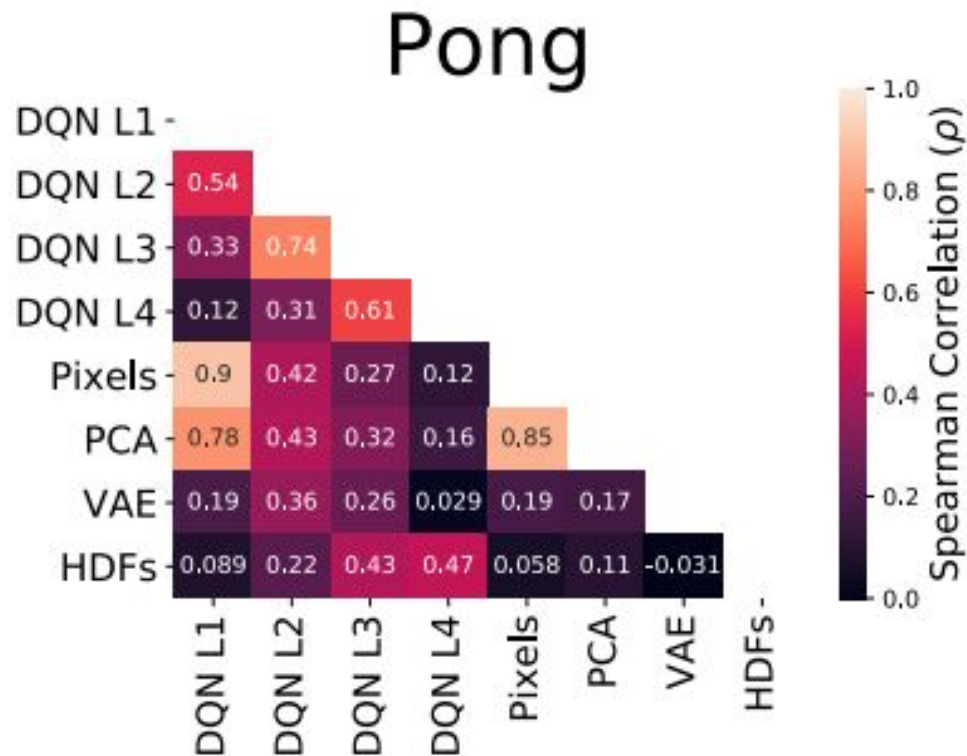
RSA on DQN - Pong

- ▷ Annotated high-level features for Pong
 - Ball position
 - Ball velocity
 - Paddle positions
- ▷ Construct dissimilarity matrices for DQN layers and compare it to these hand drawn features

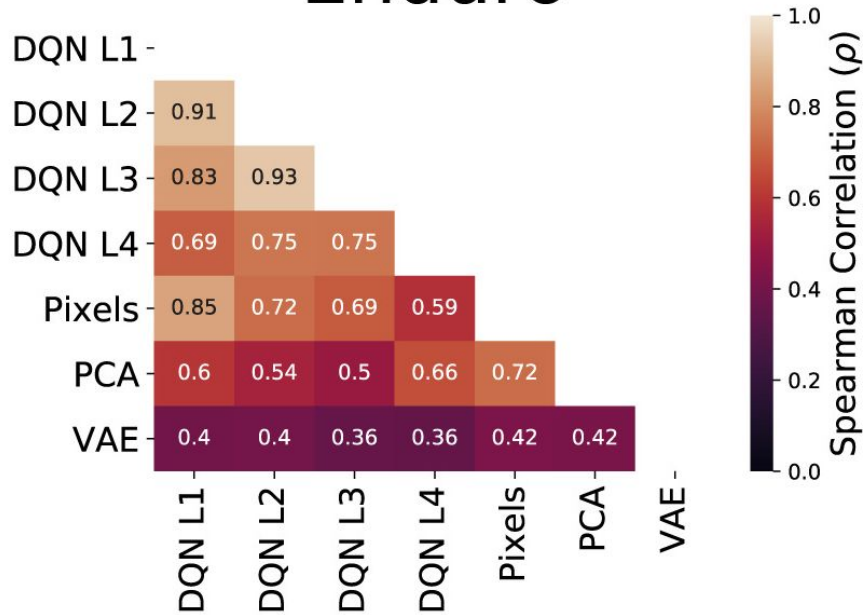


Early DQN layers
correlated to pixel space

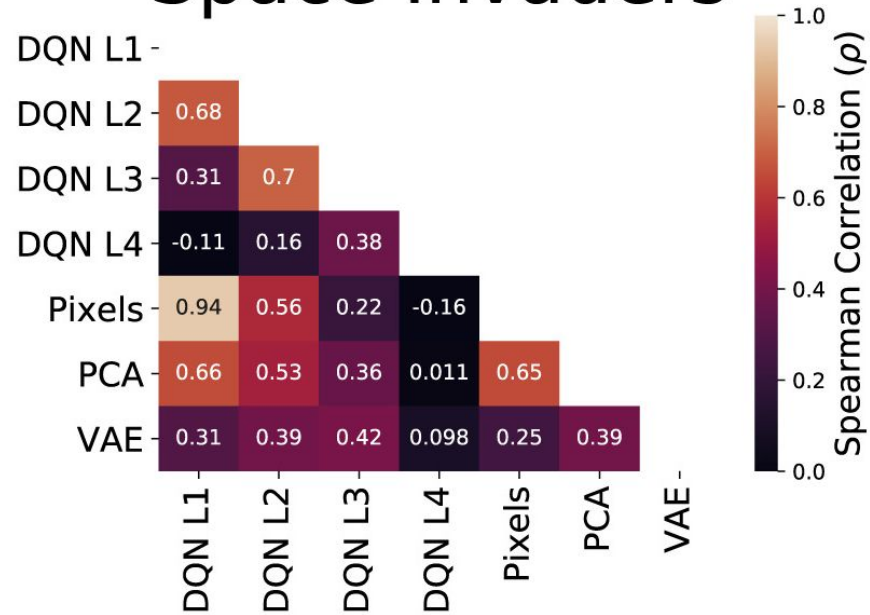
Later DQN layers
correlated to hand drawn
features



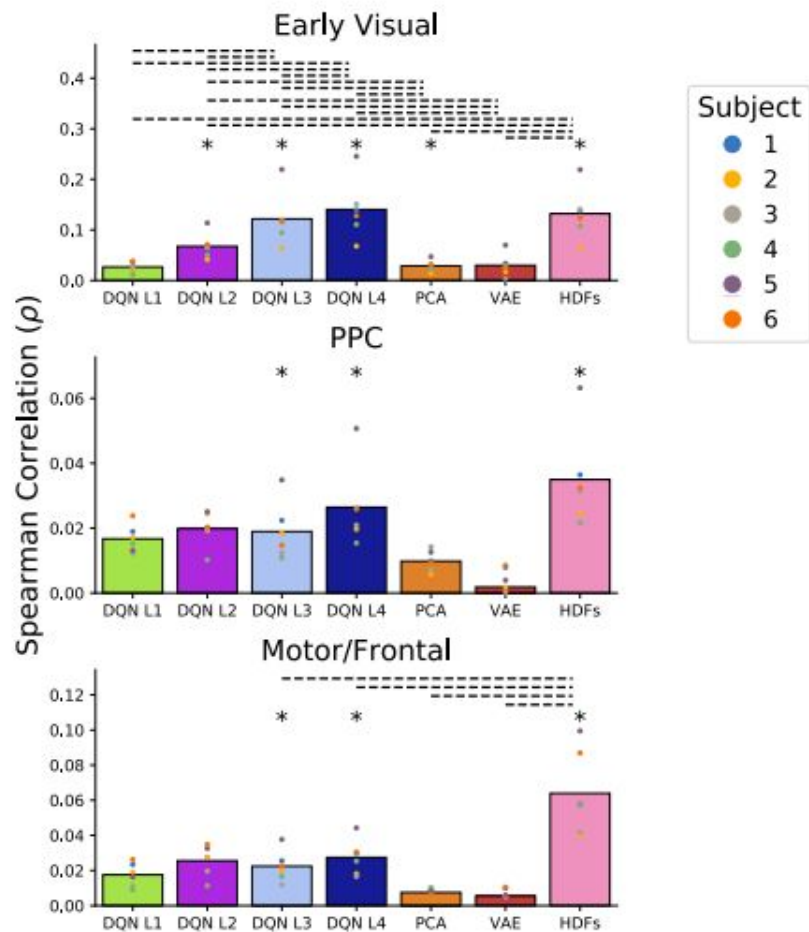
Enduro



Space Invaders



Representational Similarity Analysis of Pong fMRI Data



RSA in Pong

The shared task representation between the brain and DQN in Pong corresponds to a mutual encoding of the spatial positions of objects

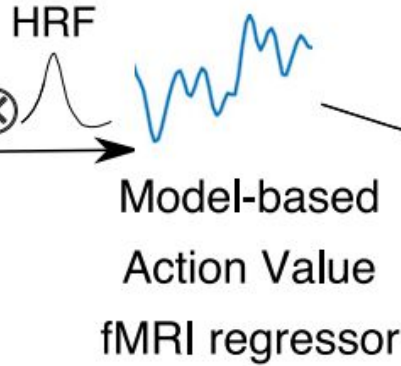
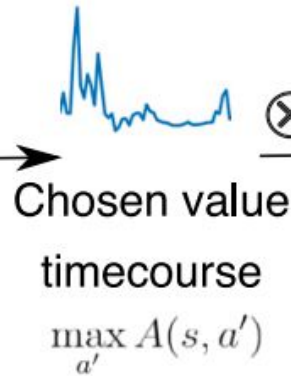
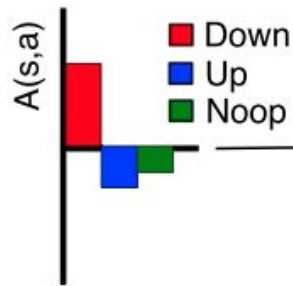
Are there correlates of DQN's action value outputs in the brain?

Human Gameplay

Data



Action Values

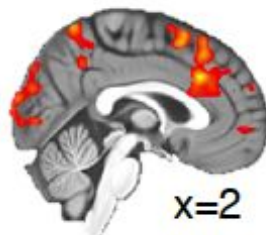


Action Value

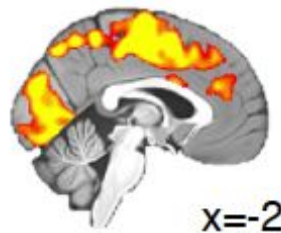
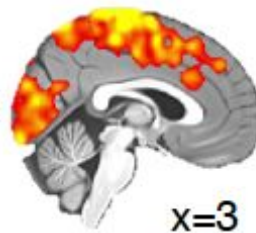
Sub001

Sub006

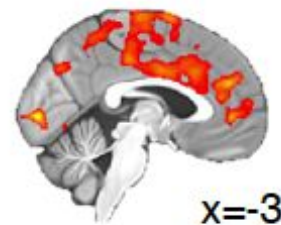
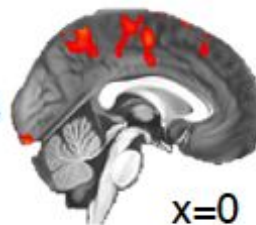
Pong



Enduro



Space
Invaders



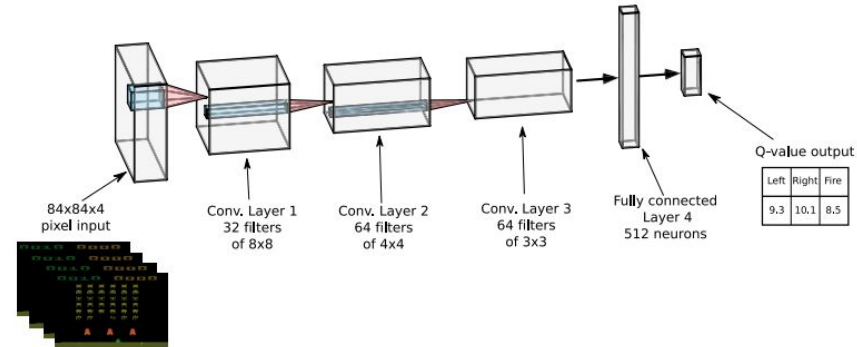
FWER corr. $p < 0.001$ at cluster level



Filter Analyses

How can we further interpret what representation is shared between the DQN and various brain regions?

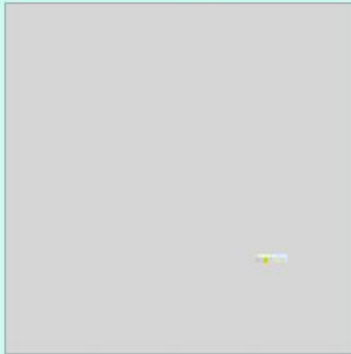
Investigate the filters in the convolutional layers



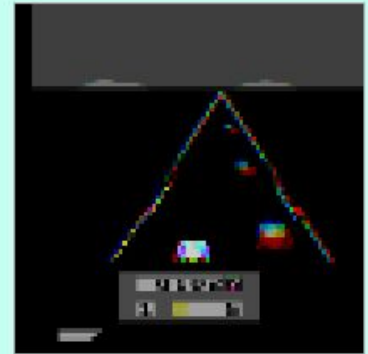
Use deconvolution to visualize the filters

Enduro 1st Convolutional Layer - Detects Edges

Feature Map 0

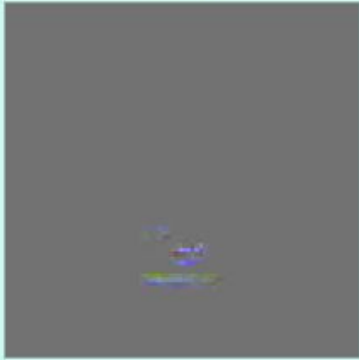


Feature Map 1



Enduro 2nd Convolutional Layer - Detects Object Parts

Feature Map 0

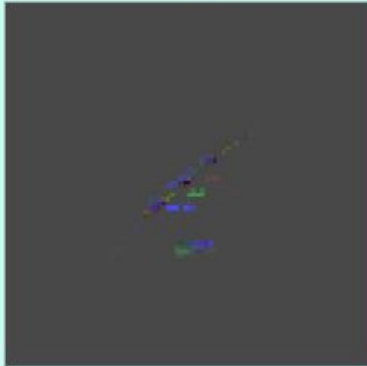


Feature Map 1



Enduro 3rd Convolutional Layer - Detects Cars and Road

Feature Map 0



Feature Map 1

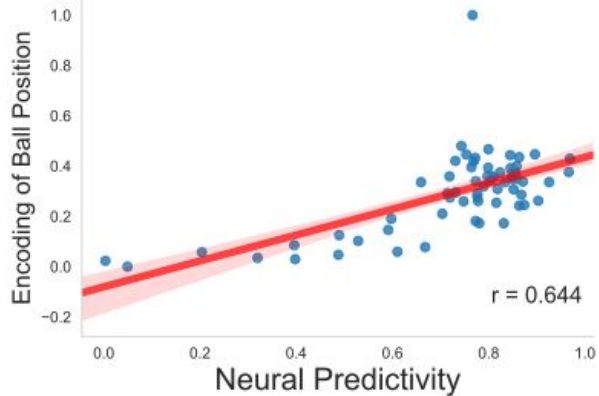


Which filters best explain brain activity?

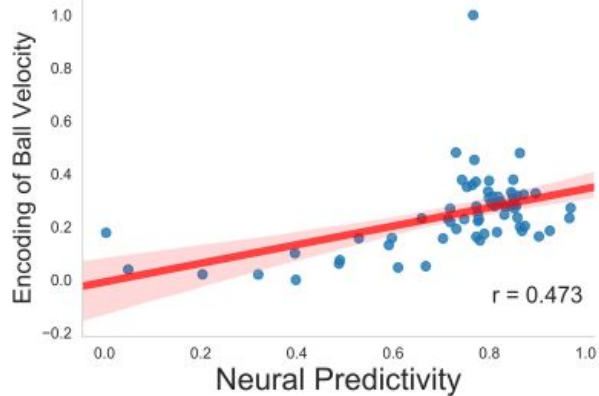
- ▷ Retrained encoding model on each convolutional filter in the last convolutional layer separately (layer 3, 64 filters)
- ▷ Each filter gets a Neural Predictivity score based on how well it predicts voxel responses in a region of interest

In Pong, the most neurally predictive filters encode the hand drawn features

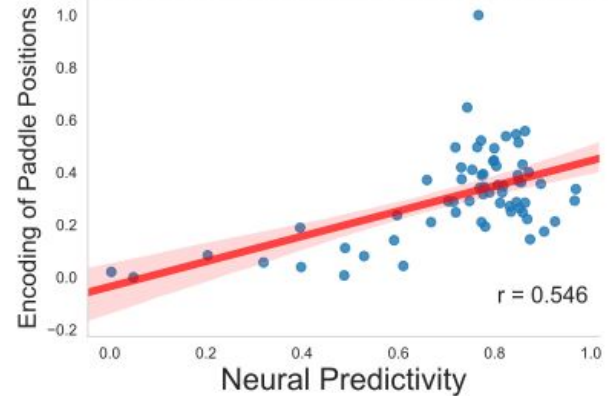
Filter Score for Brain vs. Filter Encoding of Ball Position



Filter Score for Brain vs. Filter Encoding of Ball Velocity

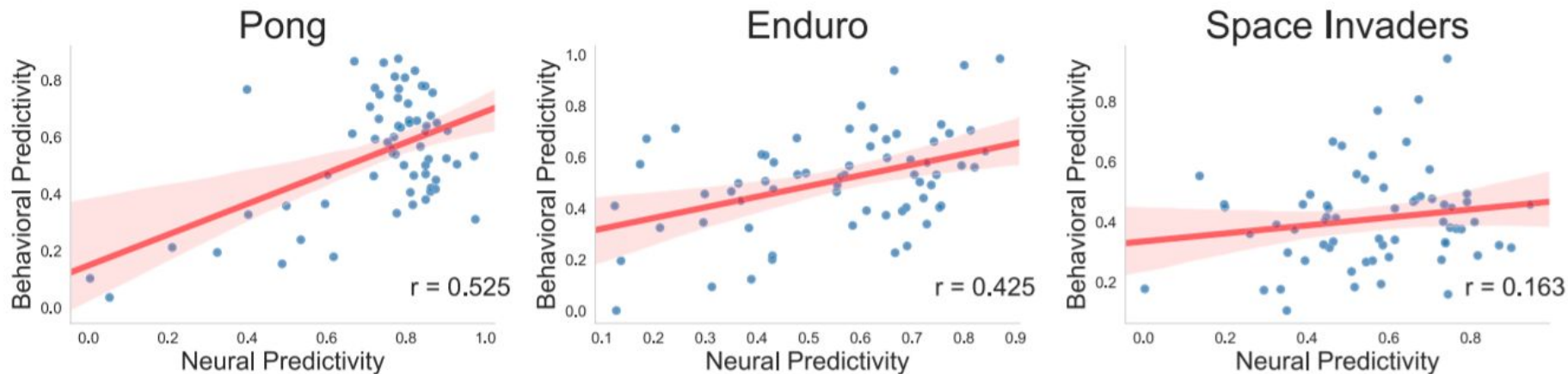


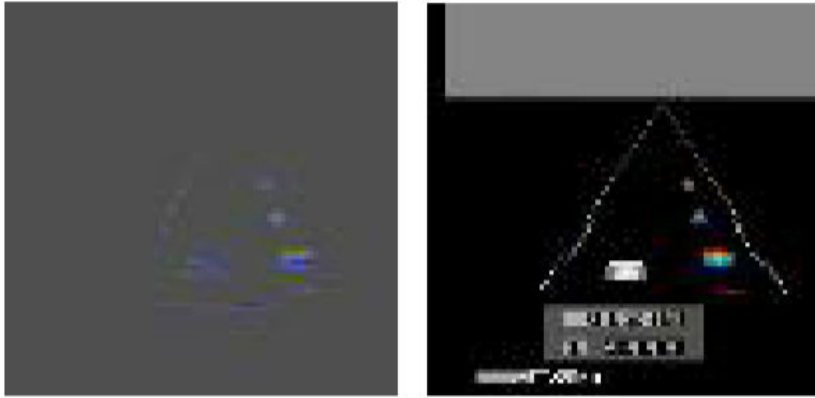
Filter Score for Brain vs. Filter Encoding of Paddle Positions



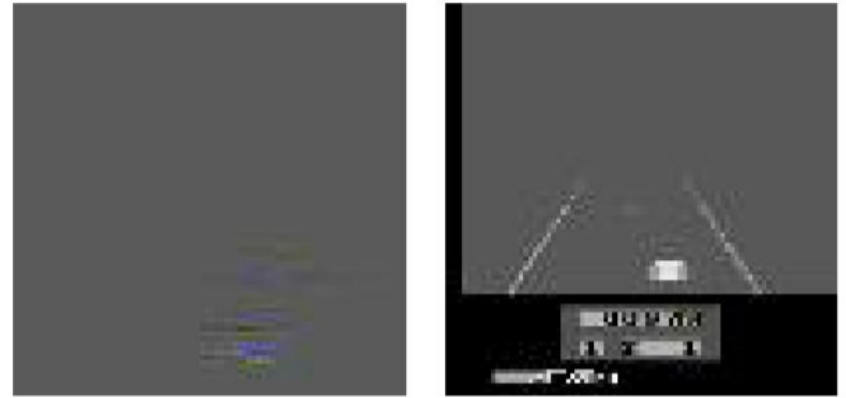
The most neurally predictive filters are also predictive of behavior

Correlation of Filter Predictivity for Modeling Actions and Voxels



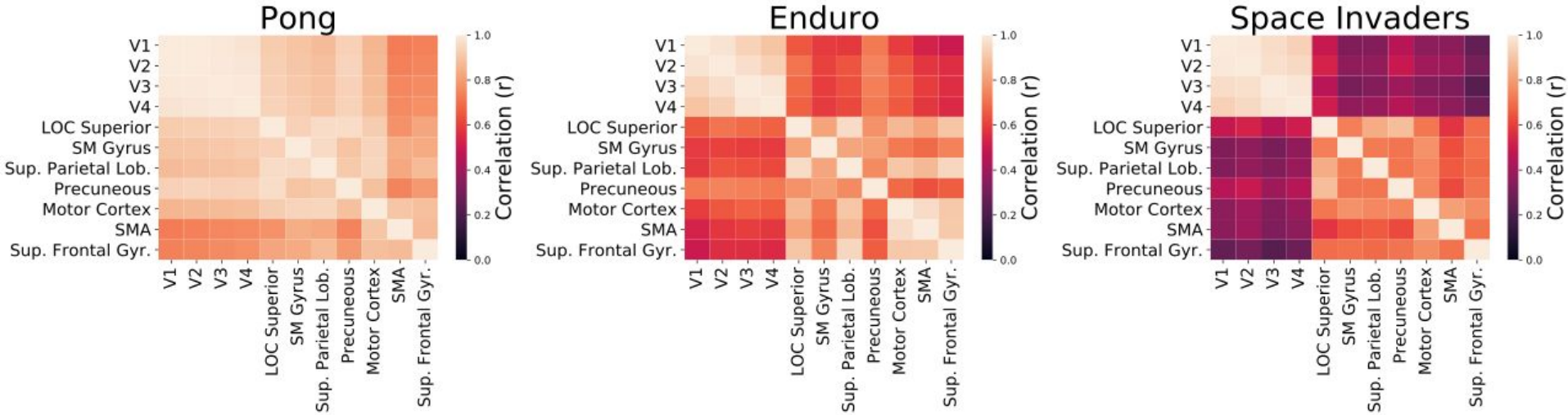


Good filter example:
Layer 3 Filter 40 in Enduro
detects side of the road and cars
Neural Predictivity Rank: 5



Bad filter example:
Layer 3 Filter 56 in Enduro
detects score on bottom of screen
Neural Predictivity Rank: 56

Correlation of Filter Neural Predictivity Across Regions



More heterogeneity of filter selectivity across regions for Enduro and Space Invaders

Abstract State-Spaces

- ▷ Abstract state-representations should be invariant to irrelevant sensory information
- ▷ For Pong, this involves encoding spatial features about the relevant objects in the game
- ▷ Can we get metrics for abstract representations for Enduro and Space Invaders?

Nuisance Invariance

Nuisance: Any random variable that affects the data x ; but is irrelevant to the task y

$$y \perp\!\!\!\perp n, \text{ or equivalently } I(y; n) = 0.$$

Translation Invariance



Size Invariance



Rotation/Viewpoint Invariance



Illumination Invariance



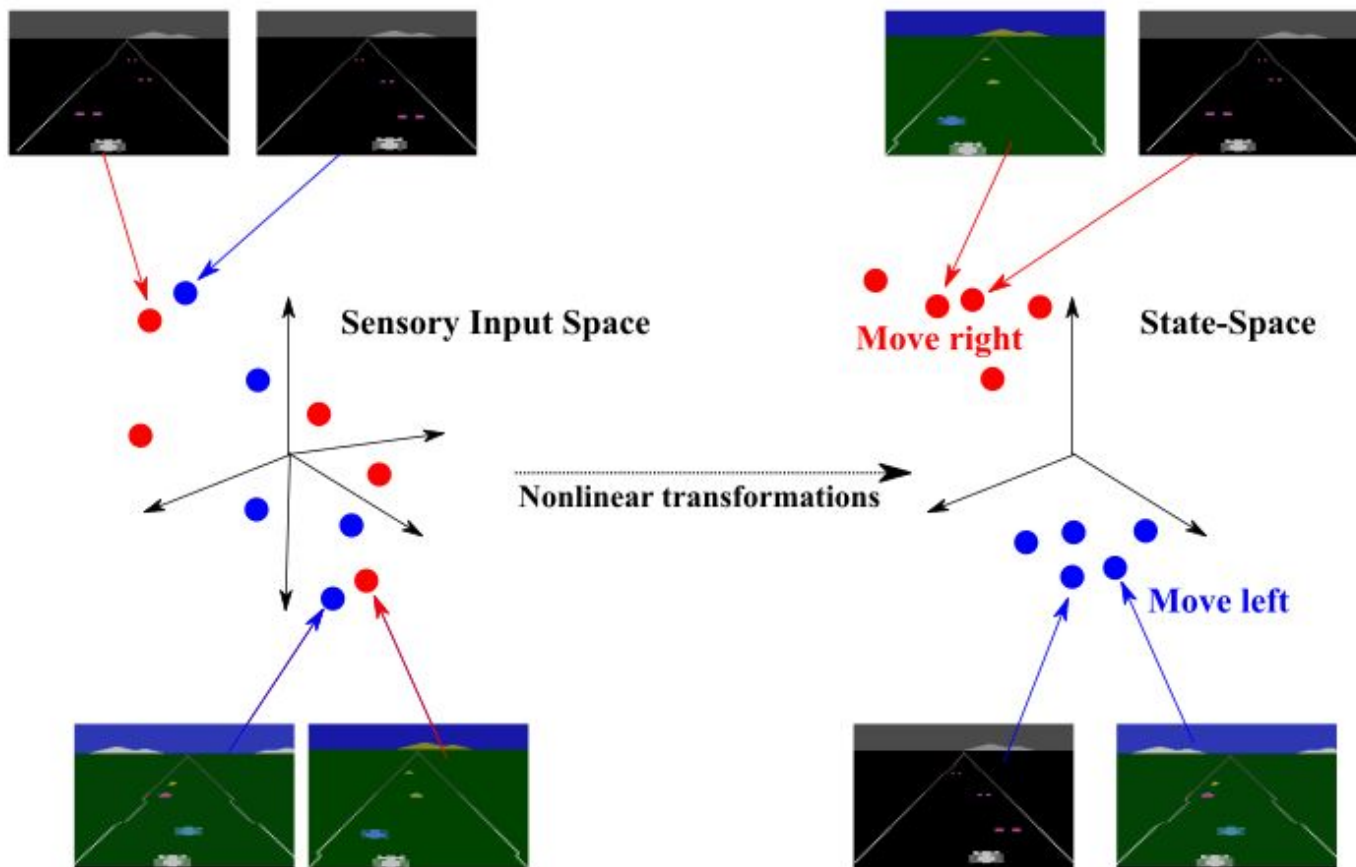
Nuisance Invariance

In Enduro, weather/time of day is a nuisance

$$I(\text{human action}; \text{weather}) = 0$$

Dramatic changes in pixel space is independent of how agent should act



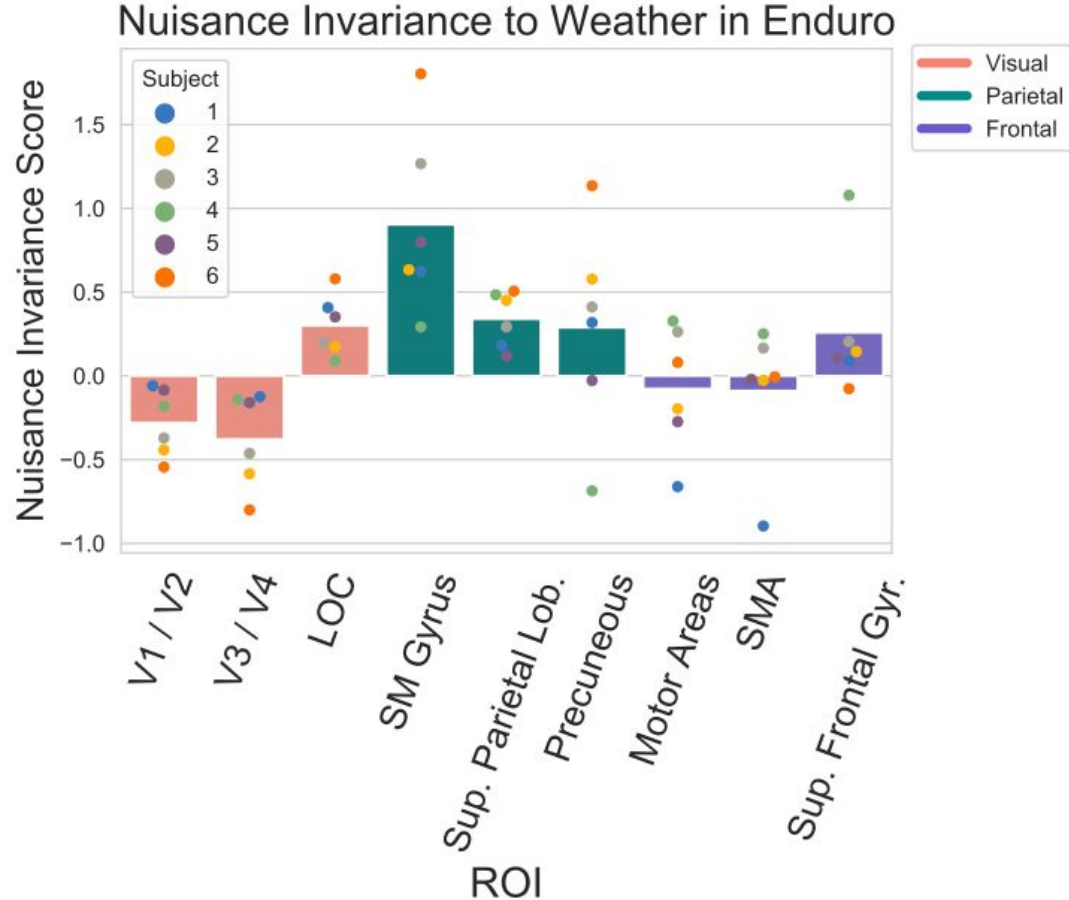


Nuisance Invariance Score

- ▷ Estimate each filter's nuisance invariance to weather with mutual information
- ▷ Correlate this metric with that filter's neural predictivity in a region
 - **Negative correlation suggests more representation of weather in a region**
 - **High correlation suggests more nuisance invariance in a region**

Filters mapped to posterior parietal cortex have less representation of weather

- Supramarginal gyrus, superior parietal lobule, precuneus
- **Less representation of weather suggests a more abstract representation in these regions that are invariant to nuisances like color of pixels**



Nuisance Invariance

In Space Invaders, the number of invaders on the screen does not have much effect on actions (what the invaders near you are doing is what matters)

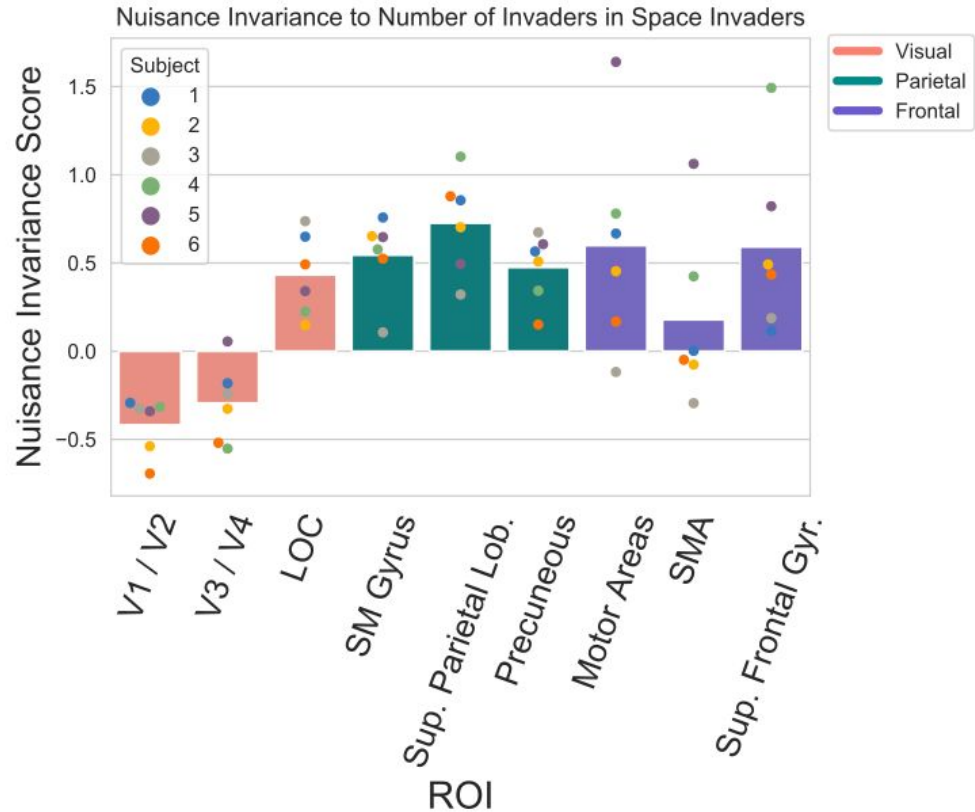
$$I(\text{human action}; \text{number of invaders}) = 0.07$$

Dramatic changes in pixel space is independent of how agent should act



Filters mapped to posterior parietal cortex have less representation of the number of invaders

- Supramarginal gyrus, superior parietal lobule, precuneus
- **Less representation of the number of invaders suggests a more abstract representation in these regions**



Representation learning in the artificial and biological neural networks underlying sensorimotor integration

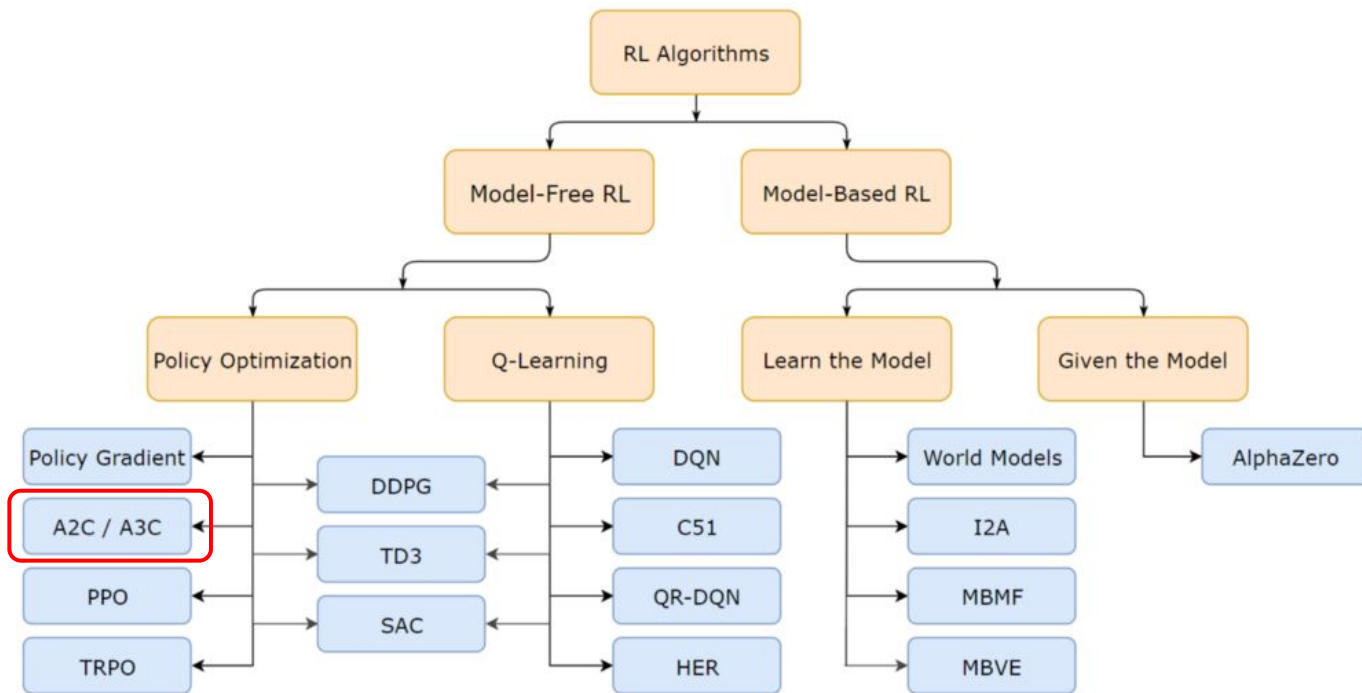
Ahmad Suhaimi, Amos W. H. Lim, Xin Wei Chia, Chunyue Li, Hiroshi Makino*

Science Advances, 2022

Compares deep RL agents and mouse brains performing the same sensorimotor task

Deep RL agent: A2C

Aside: What is Advantage Actor Critic? (A2C)



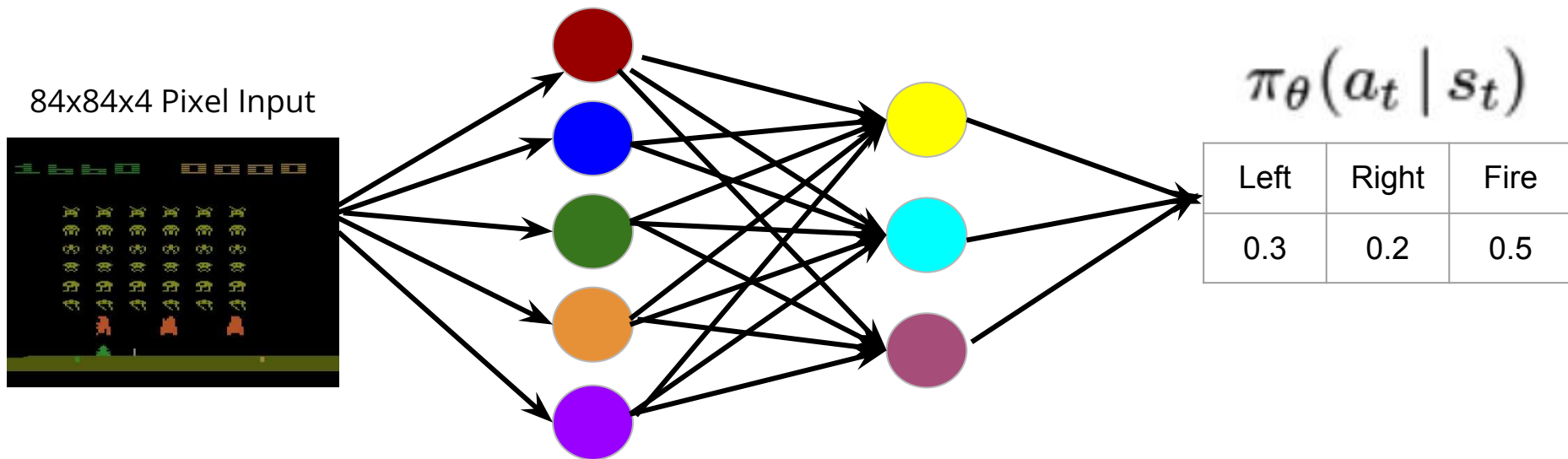
Policy-Based Reinforcement Learning

- ▷ Rather than learning a value function, directly optimize a policy to maximize the objective of expected reward

$$J(\pi_{\theta}) = \mathbf{E}_{\tau \sim \pi_{\theta}} [R(\tau)]$$

Policy Gradient/REINFORCE Algorithm

- ▷ Output of network is a policy - a probability distribution over actions



Policy Gradient/REINFORCE Algorithm

- ▷ Use gradient ascent to optimize objective

$$\max_{\theta} J(\pi_{\theta}) = \mathbb{E}_{\tau \sim \pi_{\theta}} [R(\tau)]$$

$$\theta \leftarrow \theta + \alpha \nabla_{\theta} J(\pi_{\theta})$$



Policy gradient term


Policy Gradient/REINFORCE Algorithm

- ▷ Use gradient ascent to optimize objective

$$\max_{\theta} J(\pi_{\theta}) = \mathbb{E}_{\tau \sim \pi_{\theta}} [R(\tau)]$$

$$\theta \leftarrow \theta + \alpha \nabla_{\theta} J(\pi_{\theta})$$

Policy gradient term


$$\nabla_{\theta} J(\pi_{\theta}) = \mathbb{E}_{\tau \sim \pi_{\theta}} \left[\sum_{t=0}^T R_t(\tau) \nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \right]$$

Policy Gradient/REINFORCE Algorithm

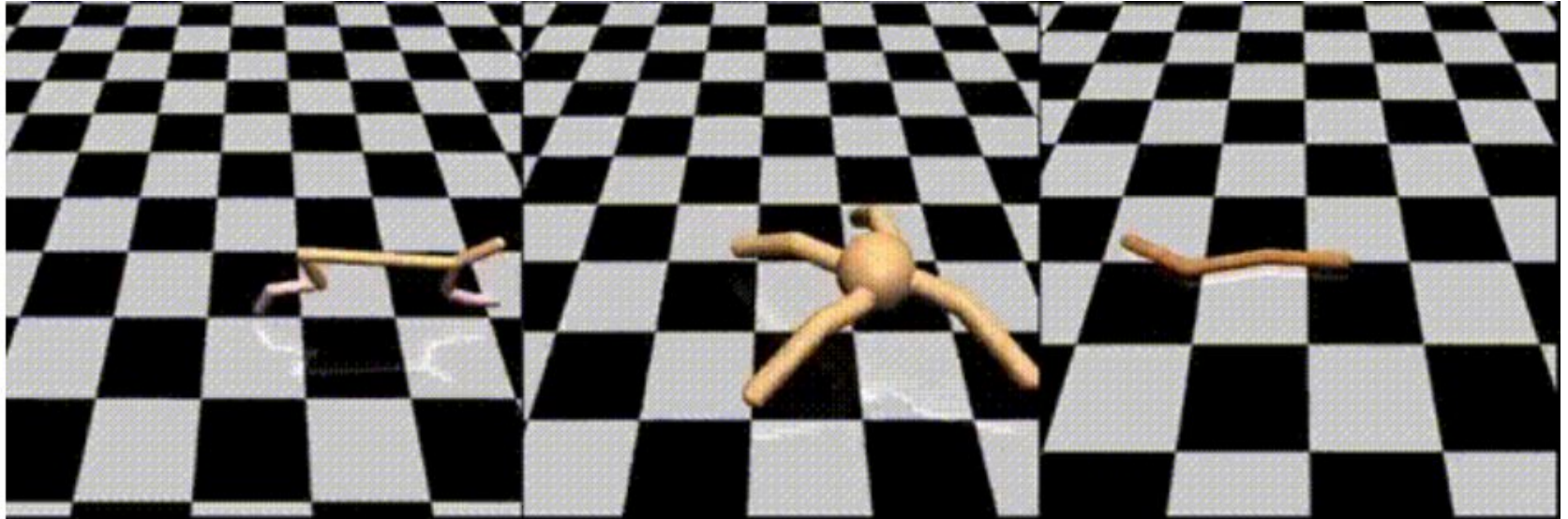
- ▷ Probability of action $\pi_{\theta}(a_t | s_t)$ increases if $R_t > 0$
- ▷ Probability of action $\pi_{\theta}(a_t | s_t)$ decreases if $R_t < 0$

$$\nabla_{\theta} J(\pi_{\theta}) = \mathbb{E}_{\tau \sim \pi_{\theta}} \left[\sum_{t=0}^T R_t(\tau) \nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \right]$$

Policy Gradient/REINFORCE Algorithm

- ▷ Known as on-policy RL because you have to train using data from current policy
- ▷ Data is thrown away after training step
- ▷ This makes training more inefficient and makes high variance gradients

Policy Gradients Useful for Continuous Action Spaces



Policy-Based vs Value-Based RL

- ▷ Policy-based
 - Pros:
 - General class of optimization
 - Can be applied to any domain like continuous action spaces
 - PG Theorem guarantees convergence
 - Cons:
 - Sample inefficient/slow learning because you throw away data (on-policy)
 - High variance

Policy-Based vs Value-Based RL

- ▷ Value-based
 - Pros:
 - More sample efficient because data is reused through replay buffer (off-policy)
 - Lower variance
 - Cons:
 - Stable training requires numerous practical tricks
 - High bias
 - Overestimation of value function

Actor-critic methods

- ▷ Combine policy-based and value-based methods to get the best of both worlds
- ▷ Learn two components
 - Actor: Learn a policy
 - Critic: Learn the value of states and provide reinforcing signal to actor

Actor-critic methods


- ▷ How can we solve the high variance problem in policy-based methods?

$$\nabla_{\theta} J(\pi_{\theta}) = \mathbb{E}_{\tau \sim \pi_{\theta}} \left[\sum_{t=0}^T R_t(\tau) \nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \right]$$

This reward to go term is highly variable from trajectory to trajectory making the gradients unstable

Actor-critic methods

- ▷ How can we solve the high variance problem in policy-based methods?

$$\nabla_{\theta} J(\pi_{\theta}) = \mathbb{E}_{\tau \sim \pi_{\theta}} \left[\sum_{t=0}^T R_t(\tau) \nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \right]$$


Also this term is not normalized

What if reward is always negative?
Every action will be negatively reinforced

Actor-critic methods

- ▷ Solution: use an action value function as the reinforcing signal

$$\nabla_{\theta} J(\pi_{\theta}) = \mathbb{E}_{\tau \sim \pi_{\theta}} \left[\sum_{t=0}^T Q(s_t, a_t) \nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \right]$$

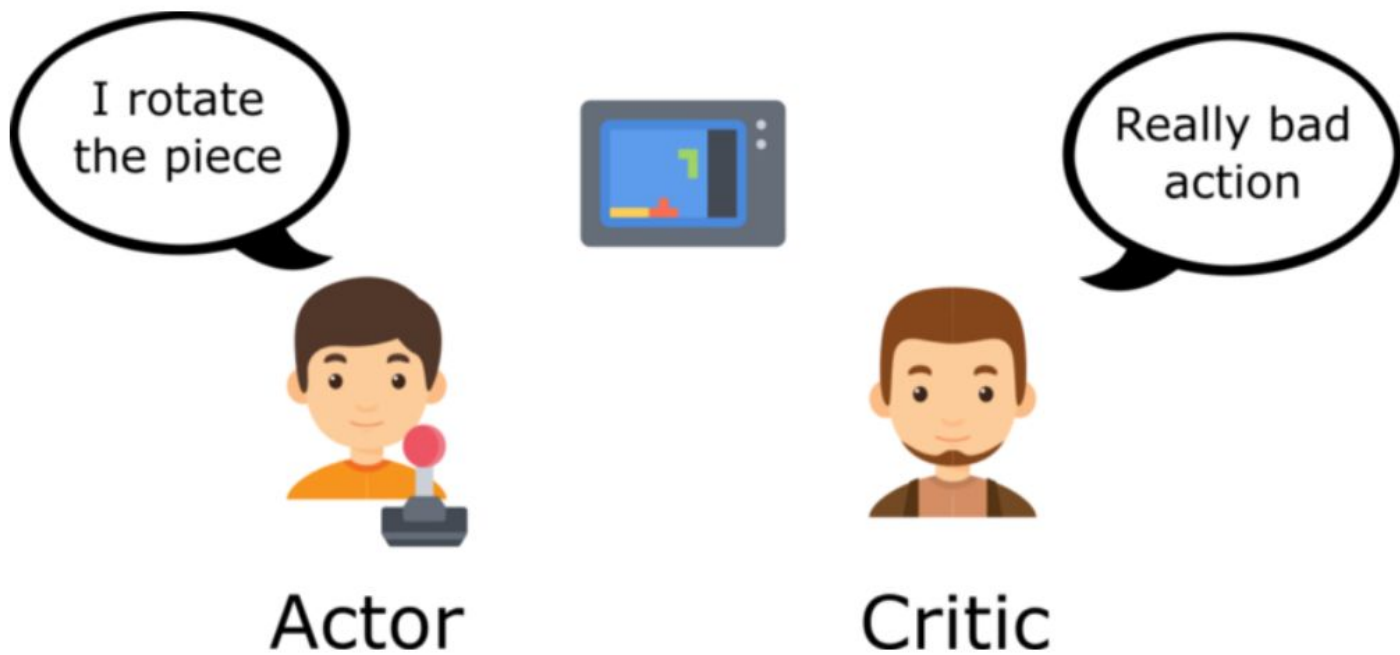
Separately learn Q-value function



Actor-critic methods

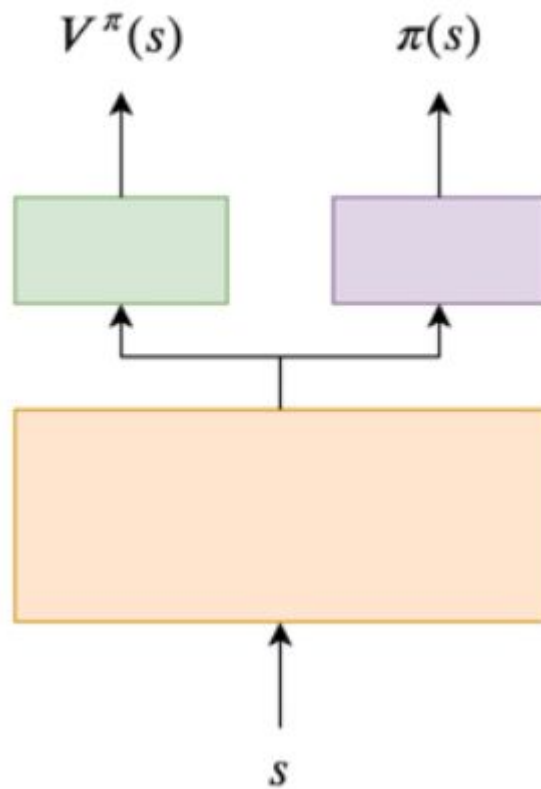
- ▷ Solution: use an action value function as the reinforcing signal

$$\nabla_{\theta} J(\pi_{\theta}) = \mathbb{E}_{\tau \sim \pi_{\theta}} \left[\sum_{t=0}^T \underbrace{Q(s_t, a_t)}_{\text{Critic}} \nabla_{\theta} \underbrace{\log \pi_{\theta}(a_t | s_t)}_{\text{Actor}} \right]$$



$$\nabla_{\theta} J(\pi_{\theta}) = \mathbb{E}_{\tau \sim \pi_{\theta}} \left[\sum_{t=0}^T \underbrace{Q(s_t, a_t)}_{\text{Critic}} \nabla_{\theta} \underbrace{\log \pi_{\theta}(a_t | s_t)}_{\text{Actor}} \right]$$

Actor-Critic
shared network



Actor-critic methods

- ▷ Can we do even better?
- ▷ Reinforce actions that produce outcomes that are better than other actions

Advantage Actor-critic (A2C)

- ▷ Reinforce actions that produce outcomes that are better than other actions
- ▷ Use the advantage function

$$A(s, a) = \underline{Q(s, a)} - \underline{V(s)}$$

q value for action a
in state s

average
value
of that
state

Advantage Actor-critic (A2C)

- ▷ We want to reinforce actions relative to how they perform vs the other actions - independent of the value of a state

$$Q^\pi(s, a) = 110, \quad V^\pi(s) = 100, \quad A^\pi(s, a) = 10$$

$$Q^\pi(s, a) = -90, \quad V^\pi(s) = -100, \quad A^\pi(s, a) = 10$$

Advantage Actor-critic (A2C)

- ▶ Use the advantage function as reinforcing signal

$$\nabla_{\theta} J(\pi_{\theta}) = \mathbb{E}_{\tau \sim \pi_{\theta}} \left[\sum_{t=0}^T \underbrace{A(s_t, a_t)}_{\text{Critic}} \nabla_{\theta} \underbrace{\log \pi_{\theta}(a_t | s_t)}_{\text{Actor}} \right]$$

Advantage Actor-critic (A2C)

- ▷ Probability of action $\pi_{\theta}(a_t|s_t)$ increases if $A(s_t, a_t) > 0$
- ▷ Probability of action $\pi_{\theta}(a_t|s_t)$ decreases if $A(s_t, a_t) < 0$

$$\nabla_{\theta} J(\pi_{\theta}) = \mathbb{E}_{\tau \sim \pi_{\theta}} \left[\sum_{t=0}^T \underbrace{A(s_t, a_t)}_{\text{Critic}} \nabla_{\theta} \underbrace{\log \pi_{\theta}(a_t|s_t)}_{\text{Actor}} \right]$$

Representation learning in the artificial and biological neural networks underlying sensorimotor integration

Ahmad Suhaimi, Amos W. H. Lim, Xin Wei Chia, Chunyue Li, Hiroshi Makino*

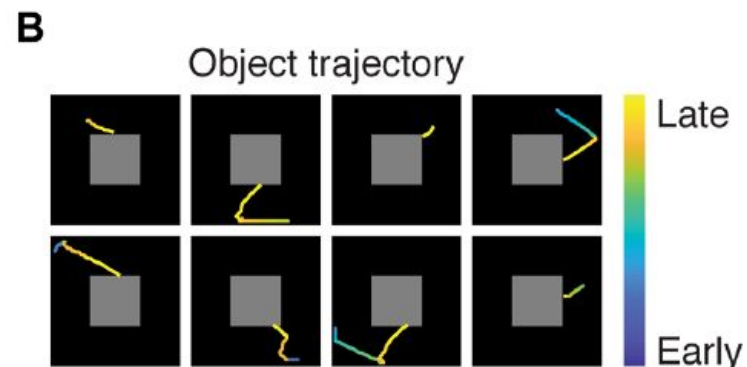
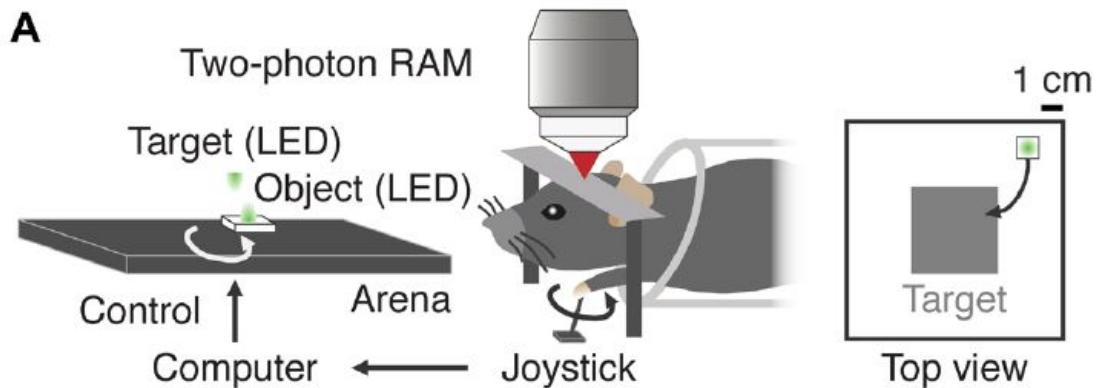
Science Advances, 2022

Compares deep RL agents and mouse brains performing the same sensorimotor task

Deep RL agent: A2C

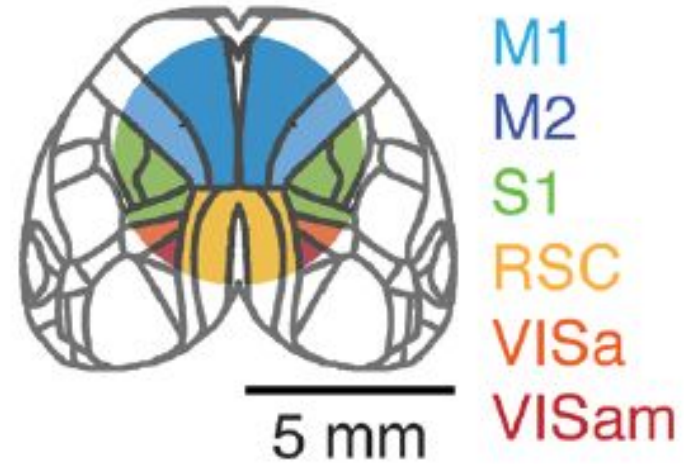
Novel sensorimotor task designed for both mice and deep RL agents (A2C)

Task: Control object with joystick/actions to reach reward zone



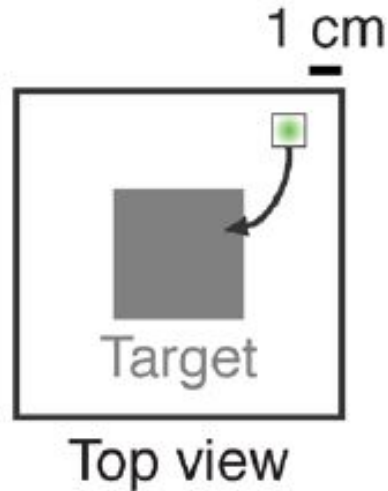
Recorded from

- ▷ Motor cortex (M1 and M2)
- ▷ Primary Somatosensory cortex (S1)
- ▷ Retrosplenial cortex (RSC)
- ▷ Posterior Parietal Cortex (PPC)
 - Anterior visual cortex (VISa)
 - Anteromedial visual cortex (VISam)



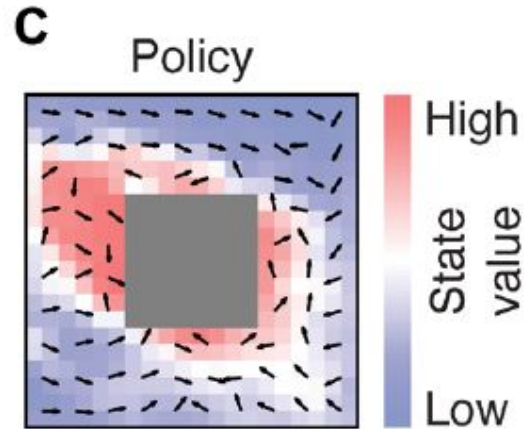
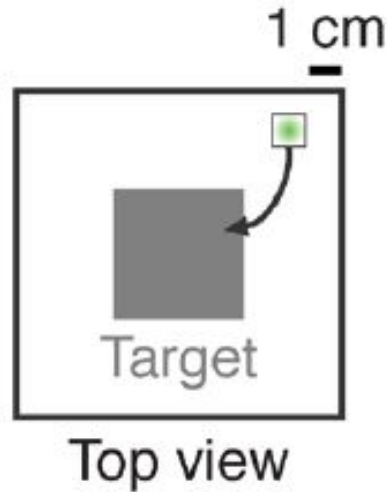
RL Task: Control object with actions to reach reward zone

- ▷ Continuous input space (x, y coords)
- ▷ High-dimensional action space (64): 8 speed x 8 direction bins

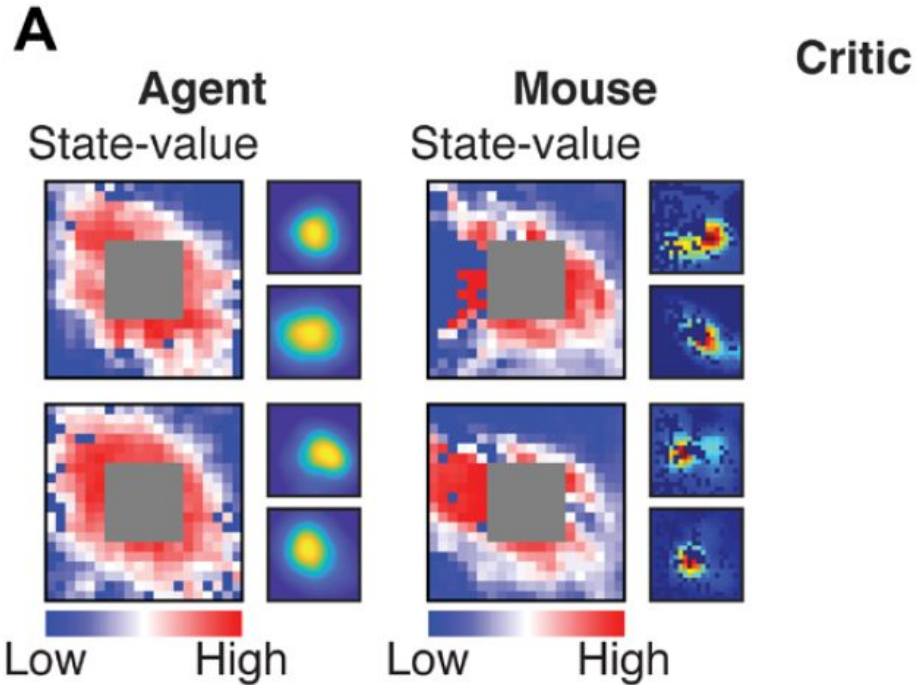


RL Task: Control object with actions to reach reward zone

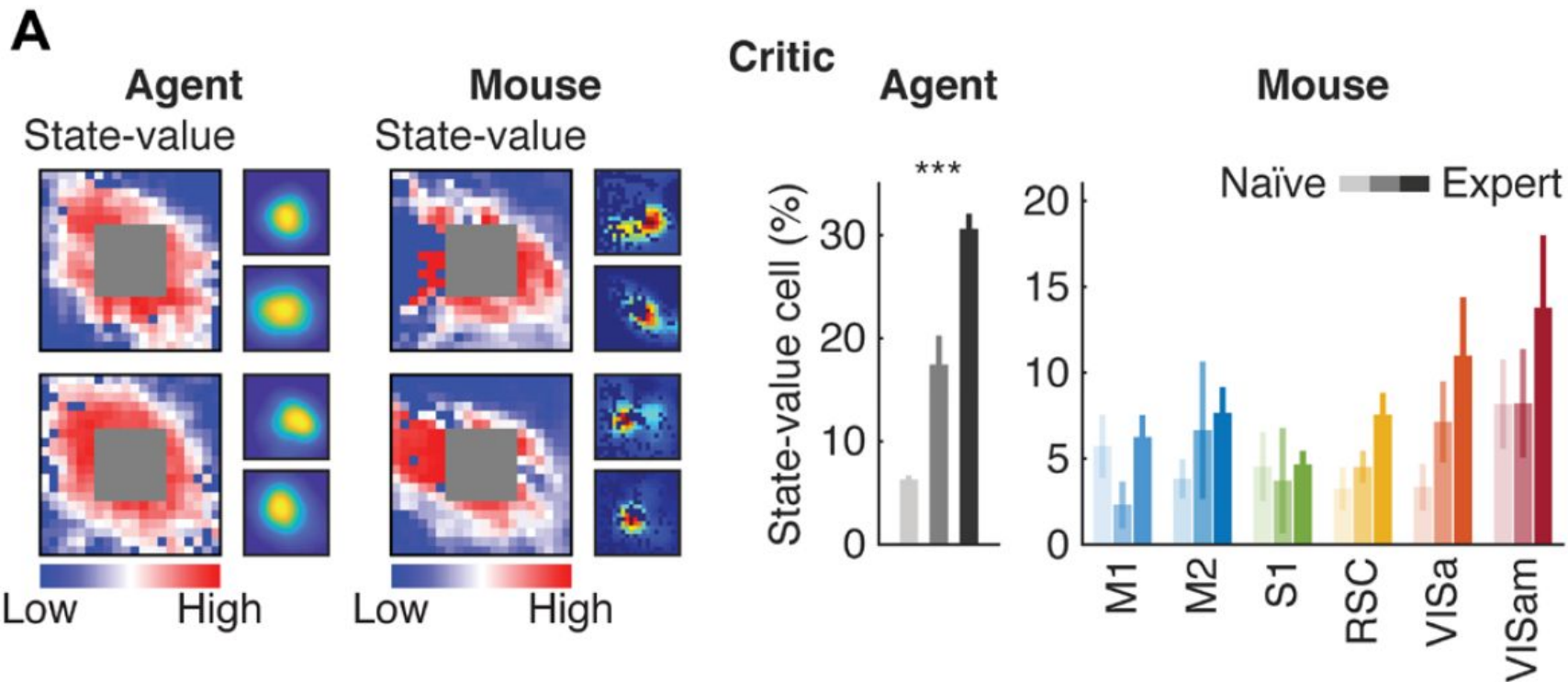
- ▷ Continuous input space (x, y coords)
- ▷ High-dimensional action space (64): 8 speed x 8 direction bins



State-Value Neurons

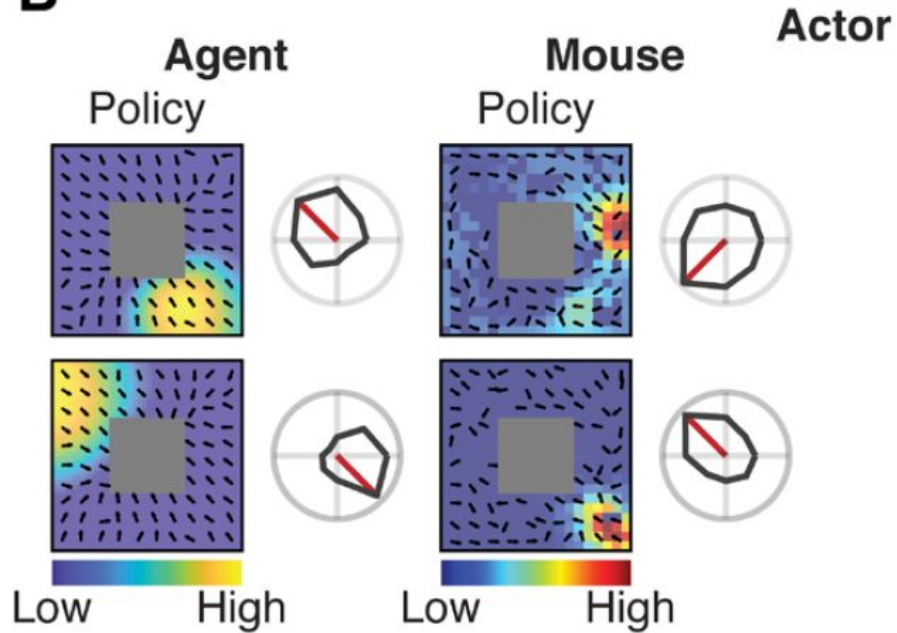


State-Value Neurons



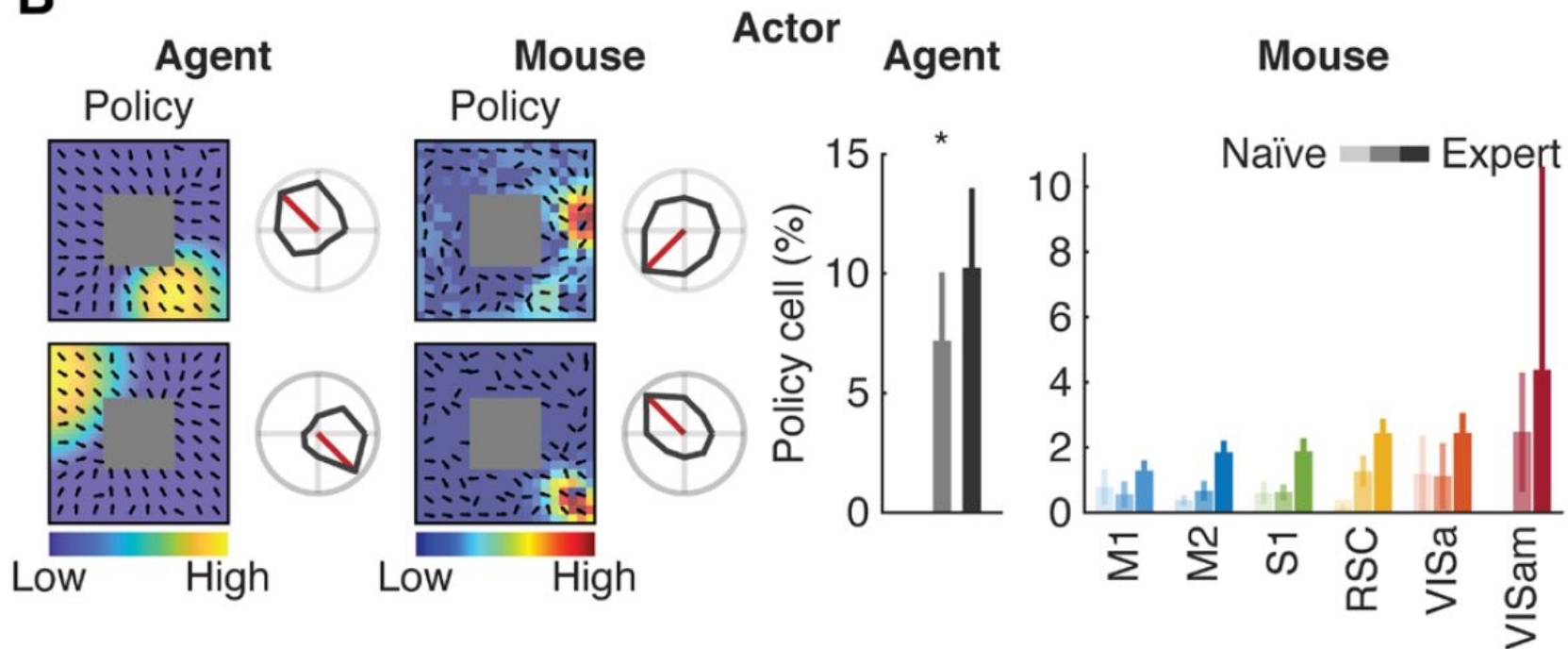
Policy Neurons

B

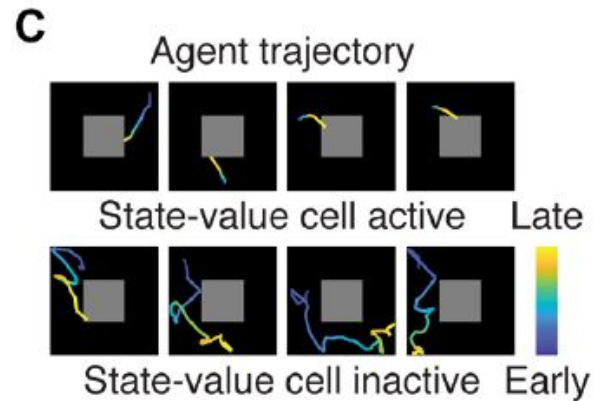
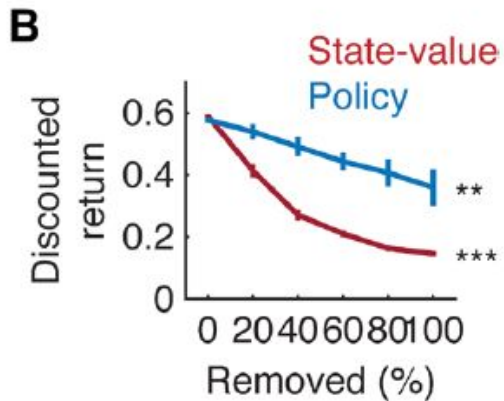
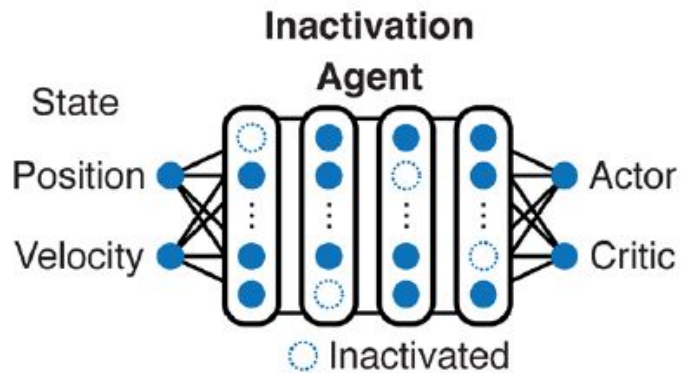


Policy Neurons

B

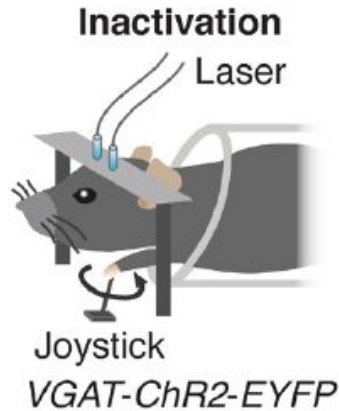


Causal Manipulations



Causal Manipulations

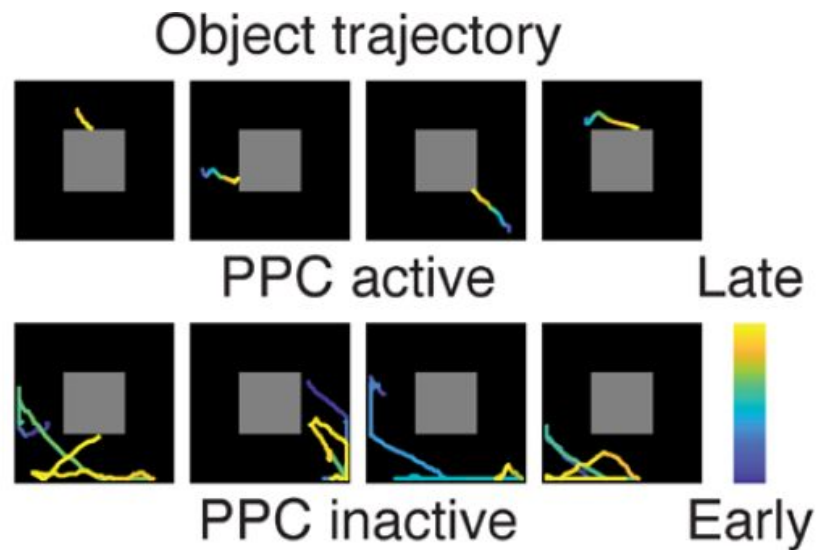
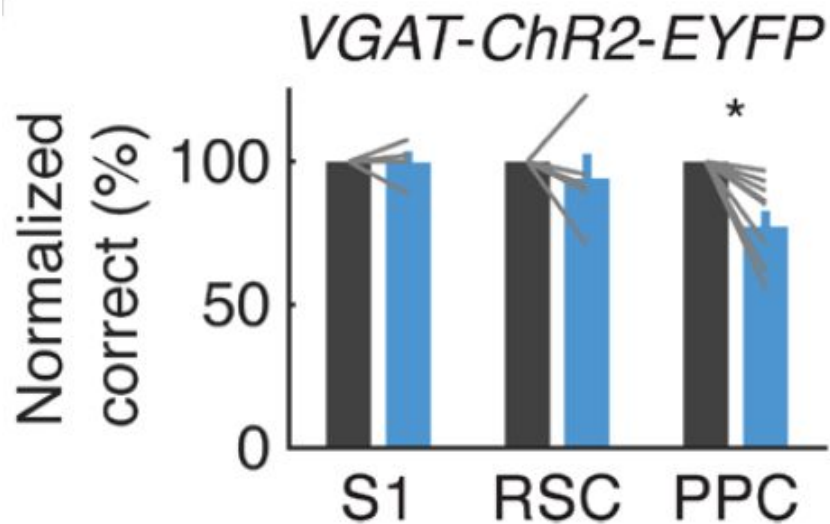
E



Optogenetic inhibition of
neural activity



Causal Manipulations



Summary

- ▶ **RL has a long history of interaction with neuroscience and psychology**

Summary

- ▷ RL has a long history of interaction with neuroscience and psychology
- ▷ **To scale the RL framework up to real-world environments, we need to combine them with modern tools like deep learning**

Summary

- ▷ RL has a long history of interaction with neuroscience and psychology
- ▷ To scale the RL framework up to real-world environments, we need to combine them with modern tools like deep learning
- ▷ **Features from deep RL algorithms can be used to significantly predict brain responses in sensorimotor regions during naturalistic tasks**

Summary

- ▷ RL has a long history of interaction with neuroscience and psychology
- ▷ To scale the RL framework up to real-world environments, we need to combine them with modern tools like deep learning
- ▷ Features from deep RL algorithms can be used to significantly predict brain responses in sensorimotor regions during naturalistic tasks
- ▷ **Deep RL and the brain converge to similar representations**
 - **Encoding an abstract state representation in PPC**
 - **Representing state-value and policy information in PPC**



Thank you for listening!