Neuron

Article

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

Highlights

- Naturalistic decision-making tasks modeled by a deep Q-network (DQN)
- Task representations encoded in dorsal visual pathway and posterior parietal cortex
- Computational principles common to both DQN and human brain are characterized

Authors

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

Correspondence

lcross@caltech.edu

In Brief

Cross et al. scanned humans playing Atari games and utilized a deep reinforcement learning algorithm as a model for how humans can map high-dimensional sensory inputs in actions. Representations in the intermediate layers of the algorithm were used to predict behavior and neural activity throughout a sensorimotor pathway.

Cross et al., 2021, Neuron *109*, 1–15
February 17, 2021
https://doi.org/10.1016/j.neuron.2020.11.021

CellPress

Please cite this article in press as: Cross et al., Using deep reinforcement learning to reveal how the brain encodes abstract state-space representations in high-dimensional environments, Neuron (2020), https://doi.org/10.1016/j.neuron.2020.11.021

Neuron

CellPress

Article

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

Logan Cross,[1,4,*] Jeff Cockburn,[2] Yisong Yue,[3] and John P. O'Doherty[2]

[1]Computation and Neural Systems, California Institute of Technology, Pasadena, CA 91125, USA

[2]Division of Humanities and Social Sciences, California Institute of Technology, Pasadena, CA 91125, USA

[3]Department of Computing and Mathematical Sciences, California Institute of Technology, Pasadena, CA 91125, USA

[4]Lead Contact

*Correspondence: lcross@caltech.edu

https://doi.org/10.1016/j.neuron.2020.11.021

SUMMARY

Humans possess an exceptional aptitude to efficiently make decisions from high-dimensional sensory observations. However, it is unknown how the brain compactly represents the current state of the environment to guide this process. The deep Q-network (DQN) achieves this by capturing highly nonlinear mappings from multivariate inputs to the values of potential actions. We deployed DQN as a model of brain activity and behavior in participants playing three Atari video games during fMRI. Hidden layers of DQN exhibited a striking resemblance to voxel activity in a distributed sensorimotor network, extending throughout the dorsal visual pathway into posterior parietal cortex. Neural state-space representations emerged from nonlinear transformations of the pixel space bridging perception to action and reward. These transformations reshape axes to reflect relevant high-level features and strip away information about task-irrelevant sensory features. Our findings shed light on the neural encoding of task representations for decision-making in real-world situations.

INTRODUCTION

The framework of reinforcement learning (RL) has illuminated how agents learn to make adaptive choices from trial and error feedback (Niv and Langdon, 2016). Efficient algorithmic strategies have been identified for learning which actions to take in a given state of the world (Sutton and Barto, 2018; Watkins and Dayan, 1992), which in turn has helped reveal neural substrates for these processes (O'Doherty et al., 2004; Schultz, 1998; Schultz et al., 1997; Steinberg et al., 2013).

To date, most research has been on learning and value representations, divorced from the perceptual systems coupled to these mechanisms in the real world. In a typical neuroscience study, state-spaces are low dimensional and discrete, characterized by a small set of distinctive stimuli and actions. Yet, in more naturalistic environments, the brain faces a continuous stream of high-dimensional input and has to efficiently identify relevant states from this complex input stream by constructing a lower-dimensional state-space internally (Botvinick et al., 2020; Niv, 2019). Actions can then be selected with even novel sensory inputs by generalizing from past experience given what previously worked well in similar states in this space. The goal of the present study is to probe how the human brain can solve this state-space representation problem.

This computational problem was a major barrier to progress in artificial intelligence, until the recent emergence of deep RL. The marriage of RL and deep learning provides an end-to-end framework for solving the task representation problem by linking sensory processing to action selection. For instance, the deep Q-network (DQN) is capable of learning high-dimensional tasks like Atari video games with human-level performance (Mnih et al., 2015). Here, we explore the possibility that the human brain may utilize similar computational principles in dynamic decision-making environments.

To address this question, we scanned human participants with fMRI while they played three different classic Atari video games: Pong, Enduro, and Space Invaders. We used DQN as a model for how the brain might solve the state representation and action evaluation problems humans face when mapping high-dimensional pixel inputs to actions.

We first tested whether DQN converges on a similar behavioral policy to that used by human participants during gameplay. We next examined the relationship between the features encoded in the hidden layers of the DQN agent and patterns of activity in the human brain while human participants played the Atari games. This enabled us to test whether the human brain utilizes similar mechanisms for encoding state-space representations as DQN.

Additionally, comparing the neural predictivity of various control models and different features within DQN helped reveal which computational principles the brain uses to encode a compact state-space representation and how this representation changes between regions. We reasoned that abstract

Please cite this article in press as: Cross et al., Using deep reinforcement learning to reveal how the brain encodes abstract state-space representations in high-dimensional environments, Neuron (2020), https://doi.org/10.1016/j.neuron.2020.11.021

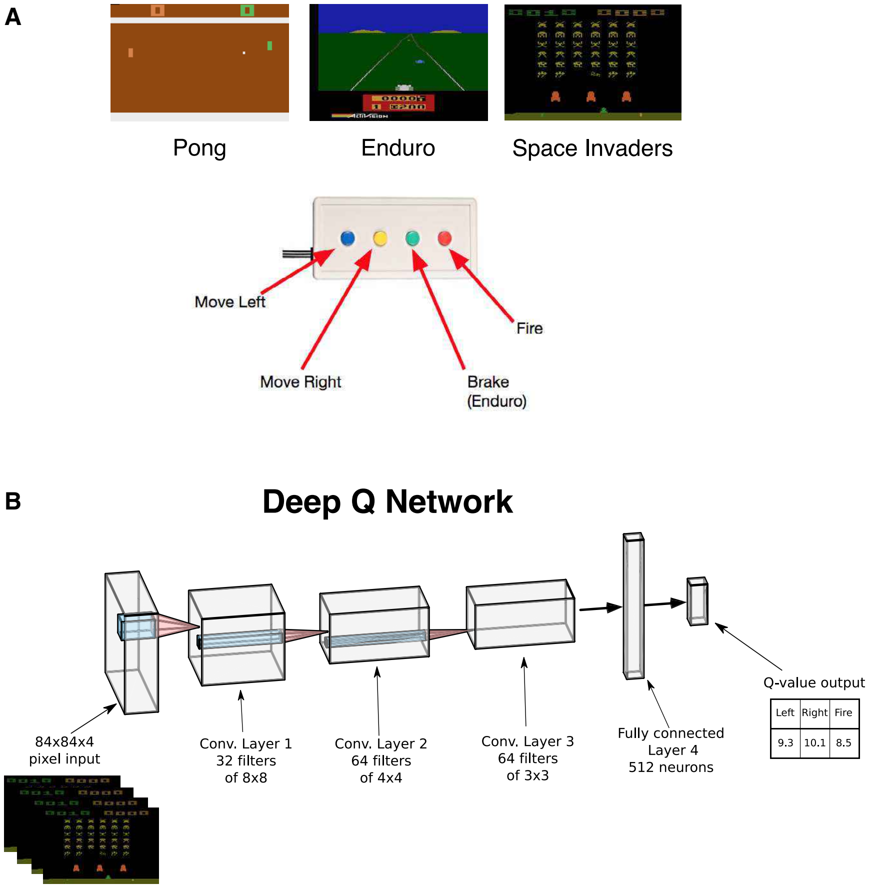


Figure 1. Atari game setup and DQN

(A) Participants played Atari games in the fMRI scanner (Pong, Enduro, and Space Invaders). A button box was used as a controller.

(B) DQN is used as a model for how the brain maps high-dimensional inputs to actions. See Mnih et al. (2015) and STAR methods for more details.

state-space representations should only encode sensory information relevant for gameplay behavior by encoding the most important high-level features and ignoring irrelevant low-level nuisance variables. The richness of Space Invaders and Enduro also enabled us to determine that abstract features that generalize across perceptually different inputs are mapped to posterior parietal cortex (PPC).

RESULTS

We used three Atari tasks of varied complexity (Figure 1A). The relatively simple game of Pong involves getting the ball past your opponent's paddle while avoiding being scored against. Enduro is a driving game where a player needs to drive as fast as possible while avoiding other cars, and Space Invaders is a fixed shooter game where a player shoots enemy spaceships. The trained DQN reaches human-level performance on all three games (Mnih et al., 2015; Table S1). Therefore, we hypothesized that the DQN agent could be utilized as an end-to-end model for how the brain maps high-dimensional inputs to actions and that its hidden layers could serve as a model for state-space representation (Figure 1B).

We acquired fMRI data from six participants, each completing 4.5 h of gameplay (1.5 h on each game). Rather than testing a large group of participants for a short time, as is typical in group fMRI studies, we obtained sufficiently large amounts of data in a small set of participants to robustly establish the relationship between each participant's data and DQN representations (see Kay et al., 2008 for a similar approach). For our analyses, we ran the frames from human gameplay data through DQN models trained independently from human data. This yielded Q-value outputs and a large set of nonlinear stimulus features represented by the activations in the hidden layers.

DQN state-space representations resemble human state-space representations

Since DQN training was independent of human behavior, it is unclear whether its state-space representations or policy would resemble that of humans. The distribution of human actions appeared to diverge from the DQN's when fed human gameplay frames (Figure S1A). However, these differences are largely trivial due to an increased propensity for humans to take NOOP actions (meaning no operation or no action) and a reduced tendency for action combinations. This is expected, since unlike DQN, humans encounter a metabolic cost for taking actions, and physical constraints limit rapid switching from one action to another. Consequently, we focused on DQN action values when human participants take a "move left" or "move right" action (or any combination with fire or brake). Across all games, DQN action values were significantly higher for the corresponding human action (Figure 2A). For example, when a human participant moves left to avoid hitting a car in Enduro, DQN also values moving left more than right. This suggests that the DQN mirrors human policies at these crucial decision points.

The DQN's state-space is encoded not by the output action value layer but rather in the internal representations of the four preceding hidden layers. Therefore, we tested if these representations are predictive of human actions. Using a linear decoder, human actions (move left versus move right) could be reliably predicted from the hidden representations in all games, demonstrating that DQN encodes stimulus features that can be used to model human actions (average accuracy: Enduro = 84.3, Pong = 75.0%, Space Invaders = 67.9%; cross-validated by run; chance level accuracy = 50%; $p < 0.001$, block permutation test; Figure 2B). We also isolated contributions from different layers by

Please cite this article in press as: Cross et al., Using deep reinforcement learning to reveal how the brain encodes abstract state-space representations in high-dimensional environments, Neuron (2020), https://doi.org/10.1016/j.neuron.2020.11.021

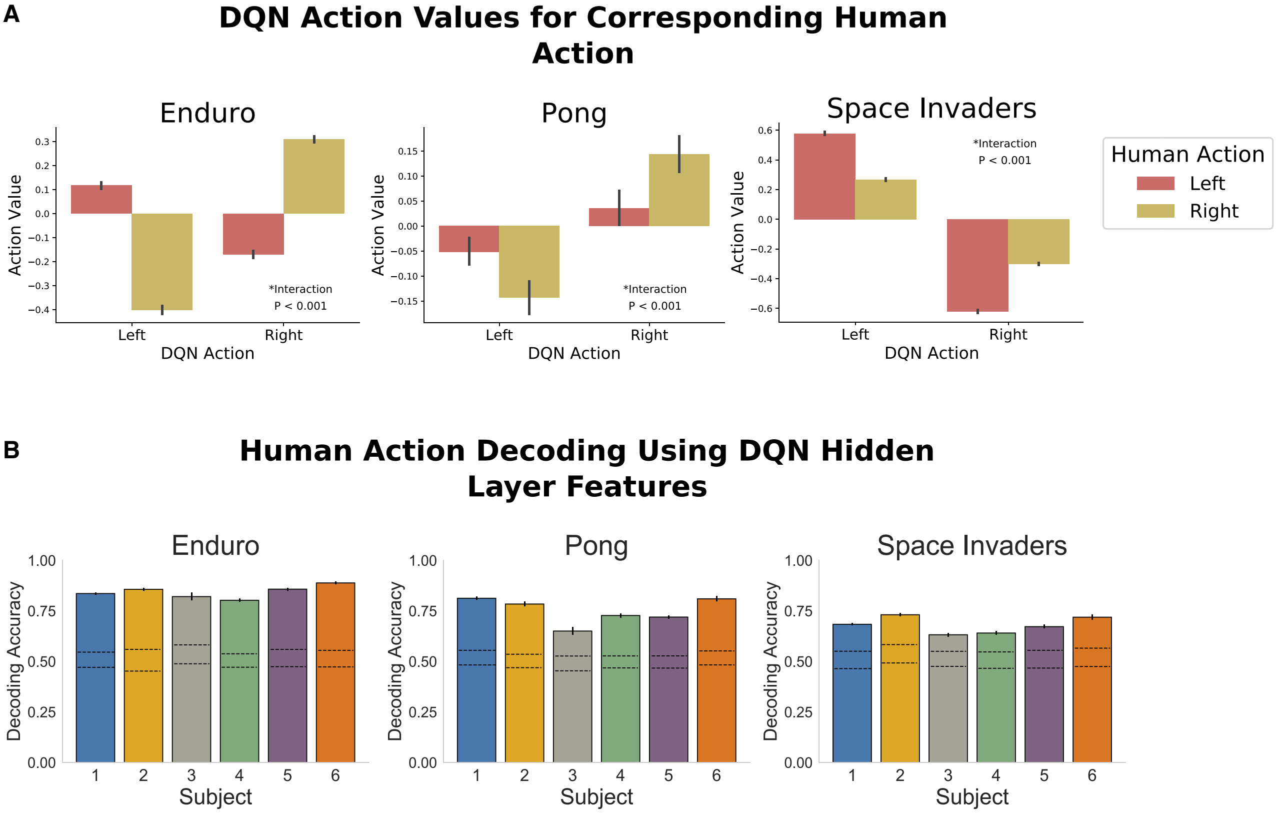


Figure 2. Predicting human behavior using DQN hidden layers

(A) DQN action values are higher for actions that participants chose. DQN action values depicted for “left” and “right” actions for frames where human participants took either a “left” or “right” action of any combination with fire or brake. Action values correspond to normalized action advantages (see STAR methods).

(B) Human actions are linearly decodable from the features in DQN hidden layers. Logistic regression models were trained to predict left versus right actions in all games. Features in the model included 100 principal components (PCs) of each DQN layer. Graphs depict cross-validated classification accuracy. Error bars depict SE across 11 cross-validation folds. Dashed lines correspond to the max and min accuracies of null distributions computed with block permutation tests of 1,000 shuffles.

averaging the absolute value of the coefficients across a layer (Figure S1B). For Enduro and Space Invaders, features from the last two hidden layers were most useful for predicting actions. For the simplest game, Pong, layers 1 and 2 contributed more, and the contribution of each layer was more varied across participants.

Encoding model reveals a distributed network representing a state-space

After validating the use of DQN hidden layers as a model for human state-space representation that could predict behavior, we next aimed to localize brain regions involved in encoding this state-space. We employed an encoding model to create a linear mapping of neural network activations to voxel responses, as done previously (Güçlü and van Gerven, 2015; Yamins et al., 2014). Neural network activations from all hidden layers were used to model and predict the response of individual voxels with ridge regression (Figure 3A).

Across games, DQN significantly predicted voxel responses throughout the dorsal visual stream and PPC (cross-validated by run; p < 0.001, false discovery rate [FDR] corrected; block permutation tests; Figures 3B–3E and S2). Prediction accuracies were higher in the dorsal visual stream extending into parietal cortex than in the ventral stream extending into temporal cortex, suggesting a specific role for the dorsal visual pathway in state-space representation for naturalistic visuomotor tasks (two-sample t test, p < 1e-10; Figure S3A). The encoding model also captured responses in motor and premotor cortex, supplementary motor area (SMA), and superior frontal gyrus in all games. Outside of primary sensory and motor areas, additional regions of PPC were mapped to DQN hidden layers, including the superior parietal lobule, supramarginal gyrus, and precuneus.

To determine whether early visual regions prefer early DQN layers and more anterior regions prefer later layers (a representational gradient), we examined coefficients in each layer. No clear gradient was identified for Enduro and Pong (Figure S3B). For Space Invaders, coefficients for layers 1 and 2 were lower in PPC, motor, and frontal regions than in early visual regions. For all games, every region had very high-magnitude coefficients for the last convolutional layer (layer 3).

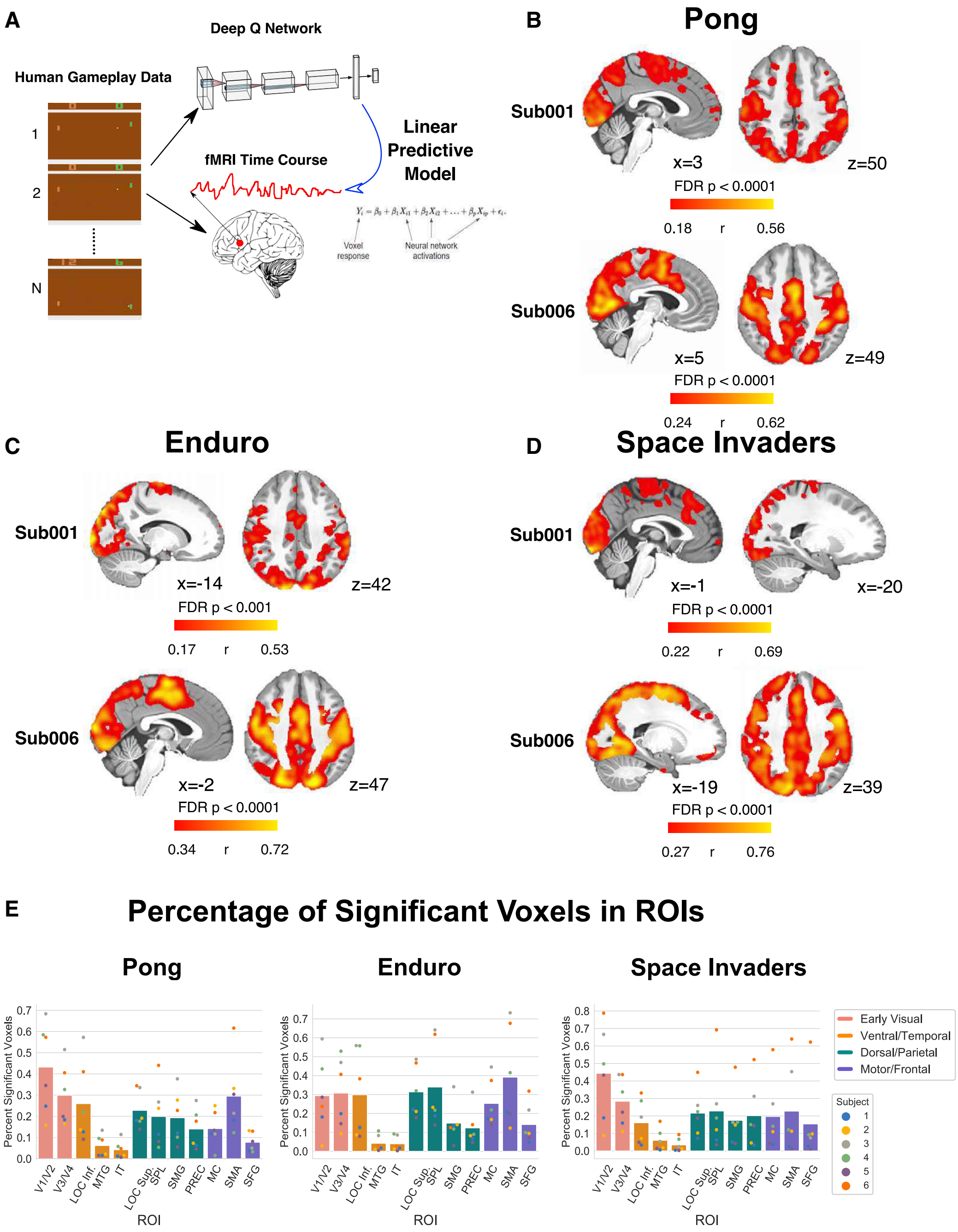


Figure 3. Encoding model: DQN hidden layers mapped to distributed network across the brain, including dorsal stream

(A) Visualization of encoding model analysis. Human gameplay frames were run through a trained DQN to extract neural network activations in the hidden layers at every time point in an fMRI run. Voxel responses were modeled using ridge regression. The explanatory features included the first 100 PCs from each DQN hidden layer.

(B) Voxels mapped to hidden layers for Pong. Cross-validated prediction accuracy uses Pearson correlation between the predicted and actual voxel responses. Whole-brain threshold at $p < 0.001$ or $p < 0.0001$ FDR corrected. Thresholds are determined via cross-validated prediction accuracy against the null distribution using block permutation testing on a subset of voxels. Data are from two participants; others are shown in Figure S2A.

(C) Same as in (B), but for Enduro.

(D) Same as in (B), but for Space Invaders.

(E) Percentage of voxels in a region of interest that are significant in the respective thresholds in (B)–(D). ROIs are noted as V1/V2, V3/V4, LOC Inf. (inferior lateral occipital cortex), MTG (middle temporal gyrus), IT (inferior temporal lobe), LOC Sup. (superior lateral occipital cortex), SPL (superior parietal lobule), SMG (supramarginal gyrus), PREC (precuneus), MC (motor cortex), SMA (supplementary motor area), and SFG (superior frontal gyrus). Plots for individual participants are shown in Figure S2B.

Control analyses

An alternative explanation for the encoding model results is that they reflect basic visual features and not information related to reward or action evaluation. To test this, we performed control analyses with feature representations of variable complexity. We used motor regressors as a basic motor control and principal components (PCs) of the pixel space to control for low-level visual properties. We also included two deep neural network (DNN) controls: a DQN agent trained on a separate game, and a variational autoencoder (VAE) (Kingma and Welling, 2014), an unsupervised representation learning method used previously to extract state representations (Ha and Schmidhuber, 2018; Higgins et al., 2018b; Watter et al., 2015) (see STAR methods and Figure S4A for examples of VAE outputs). Since the VAE does not encode value or action information, this allows us to test whether this information is needed to reach the prediction accuracies of the DQN encoding model.

DQN outperformed all control models ($p < 1e\text{-}10$, paired t test across voxels) across games except in one participant (Figures 4A and S4B). Furthermore, DQN was best in all regions of interest (ROIs) (except in one participant), especially in PPC (Figures 4B and S5A). The relative performance of different feature sets reveals the computational principles accounting for DQN's ability to explain neural activity. Nonlinear feature representations outperformed linear ones, as both the DQN trained on another game and the VAE consistently showed higher prediction accuracies than a linear principal-component analysis (PCA) model. Additionally, the original DQN surpasses the other two DNN models by linking perception to action and reward.

We next examined whether neural to DQN feature correlations are maintained when all models are included in the same analysis to compete for variance. This reveals whether DQN offers unique predictive information even after controlling for basic visual and motor activity and alternative sensory models. For this, we constructed a general linear model with the first 10 PCs of the most relevant models (DQN layers 1–4, VAE, and PCA) and other regressors of no interest such as game events.

We found that many voxels within each ROI are significantly modulated by unique variance in each model, particularly DQN layers 3 and 4 ($p < 0.001$ family-wise error rate [FWER] corrected, cluster level, F-test; Figure 4C). In Figure 4C, the results show the proportion of voxels per ROI correlating with a given model above and beyond variance explained by every other model. After controlling for both VAE and PCA, all DQN layers still explain significant variance in a substantial proportion of the voxels per ROI. Additionally, VAE and PCA models explain significant variance after controlling for the effects of the DQN layers. Since early visual and motor regions encode features in DQN layers 3 and 4 when controlling for the other models, this suggests that even these primary sensory regions process more complex sensorimotor features than in conventional visual and motor models.

Representational geometry of DQN's internal representations

The highly distributed representation and numerous parameters within a DNN make its representation rather opaque. To shed light on what DQN is encoding, we utilized representational similarity analysis (RSA). RSA allows comparison of the representational space of many different data types and models of varying dimensionality (e.g., deep network, fMRI patterns, and hand-drawn features), helping to illustrate how a model's representation changes throughout a task as well as aiding comparison across models (Haxby et al., 2014; Kriegeskorte et al., 2008).

We first examined Pong, which can be fully characterized with a few high-level features that we manually annotated frame by frame: the positions of the two paddles, the ball position (X and Y), and the ball's velocity (X and Y). A useful and compact state-space should encode this information in some form. An exemplar dissimilarity matrix (DSM; see STAR methods) for these hand-drawn features is illustrated in Figure 5A alongside the DSM of the last convolutional layer in DQN (layer 3) for the same game frames. Similarity is high between two time points when feature vectors in those time points are close in a distance metric (i.e., Euclidean). The representational geometry of DQN resembles the hand-drawn feature DSM, suggesting that it may encode these game-relevant features directly.

To quantify similarities among different DQN layers, hand-drawn features, and other models, we correlated the model DSMs with each other. In Pong, the internal representations in DQN start to become highly similar to hand-drawn features in layers 3 and 4 (Figure 5B; Spearman $\rho = 0.53, 0.55$, respectively), suggesting that DQN constructs a compact state-space representation by realigning its axes to code for these high-level features in later layers. Although this object information is present in the input pixels, they share a relatively low correlation with the pixel space ($\rho = 0.058$), suggesting some form of nonlinear transformation is required to disentangle this information from the input (DiCarlo and Cox, 2007; Higgins et al., 2018a). Additionally, the first layer of DQN in Pong is highly similar to the pixel space and PCA model ($\rho = 0.9$; $\rho = 0.78$), suggesting that the input data are not yet highly compressed in the first layer of DQN. In contrast, the later layers become increasingly dissimilar to the pixel and PCA representation as they start encoding a lower-dimensional subspace for game-relevant features. A similar pattern is seen in Space Invaders, where the first DQN layer is highly correlated to the pixel space and PCA model ($\rho = 0.91$; $\rho = 0.69$) but the last layer is highly dissimilar ($\rho = -0.16$; $\rho = 0.04$). In Enduro, representations in all four layers are highly similar to each other, suggesting that differences between them might be more subtle, raising the possibility that there may be more interesting variance within a layer rather than between layers. In all games, the VAE representations are moderately similar to the DQN's, especially for the first three DQN layers.

The brain's state-space representation in Pong encodes the spatial information about objects

Next, we tested whether the brain similarly encodes the spatial positions of the objects in Pong by computing DSMs from voxel activity and correlating these DSMs with a hand-drawn feature DSM (downsampled to fMRI resolution). For all subjects, the hand-drawn feature DSM was significantly correlated to all brain areas in the sensorimotor pathway previously identified in the encoding model analyses (Figures 5C and S6 for individual subjects; block permutation tests, $p < 0.01$, FWER corrected for multiple comparisons). This suggests that similarly to DQN, the brain's

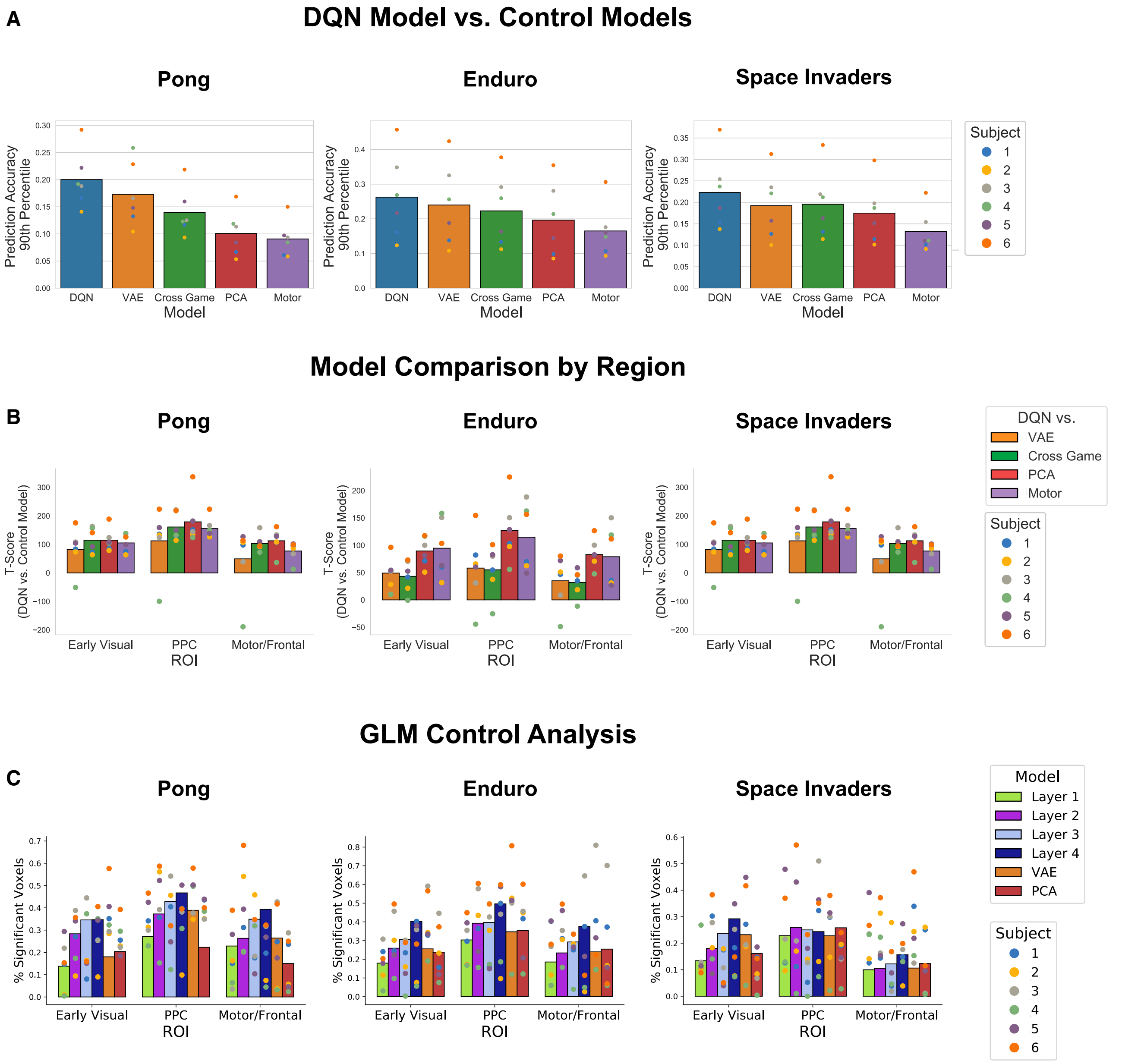


Figure 4. Control models

(A) Encoding analysis control models: motor regressors, PCA on the input pixels, DQN trained on one of the other games, and a VAE. Bar plots show prediction accuracies for the 90th percentile of prediction accuracies across the whole brain for each model (averaged across six participants with each participant's values shown). Boxplots for distribution of scores in the upper 20th percentile for each model and participant shown in Figure S4B.

(B) T-scores by region of interest comparing DQN prediction accuracies to prediction accuracies from control models. T-values reflect average T-scores across participants with each participant's T-scores shown. Plots for individual participants are depicted in Figure S5A.

(C) Percentage of significant voxels for each ROI in a GLM where all DQN layers, VAE model, and PCA model compete for variance ($p < 0.001$ FWER corrected, cluster level, F-test across 10 PCs representing a model's regressors).

state-space representation in Pong involves coding for high-level features tracking the spatial positions of the relevant objects.

Additionally, brain DSMs are significantly correlated to DQN layers 3 and 4 for all subjects in early visual, PPC, and motor/frontal ROIs (and to DQN layer 2 for early visual regions). Representations in early visual areas are already highly correlated to hand-drawn features, which may explain why these regions prefer DQN layers 3 and 4 rather than earlier layers.

Action values encoded in motor and premotor areas

DQN hidden layers encode a state-space to compute Q-values in the output of the network for action evaluation. To identify

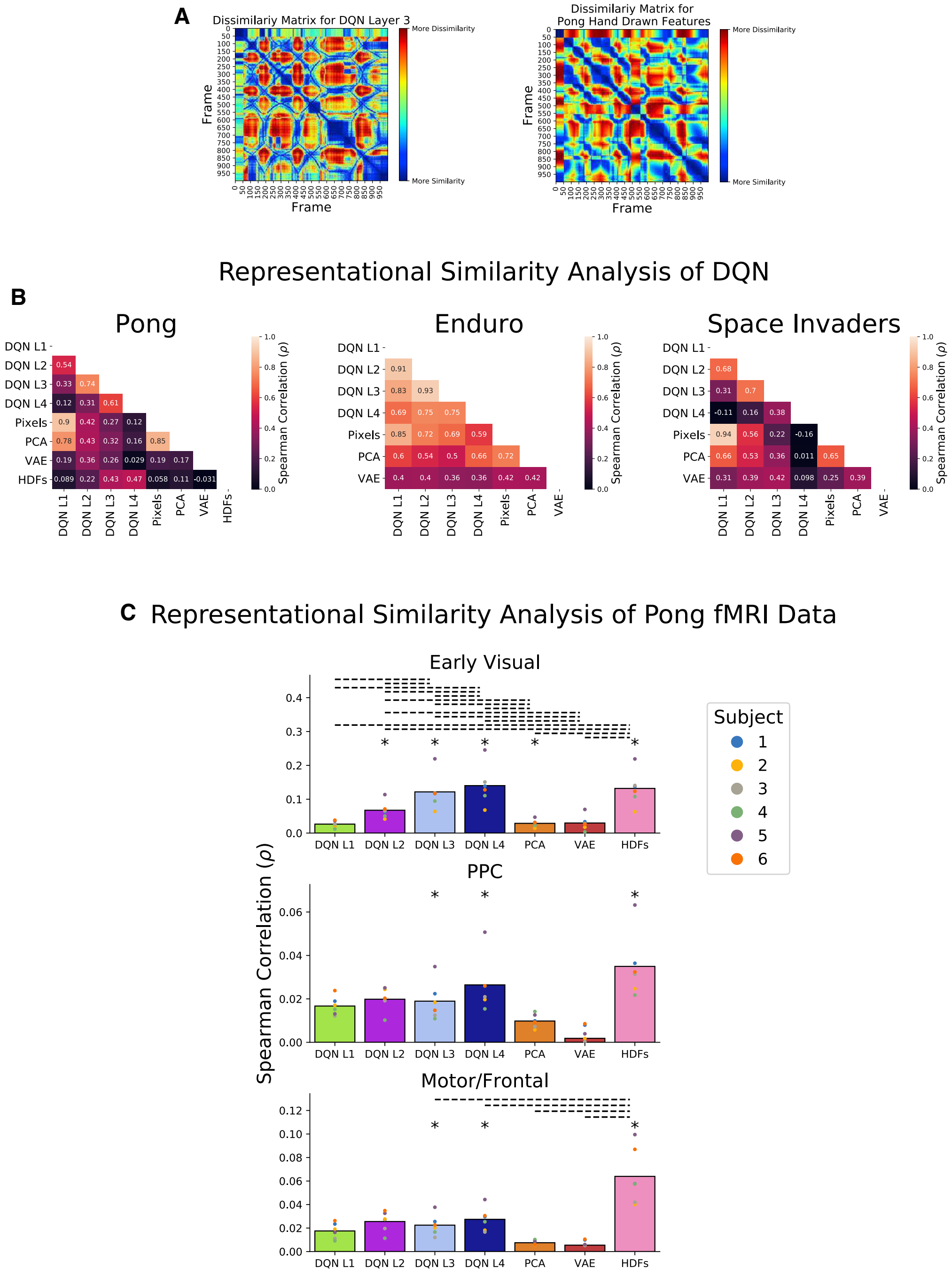


Figure 5. Representational similarity analysis

(A) Illustrations of what dissimilarity matrices (DSMs) look like for Pong. DSMs represent pairwise comparisons of model representations across time, depicted here for the first 1,000 frames in an example Pong run. The DSM on the left represents the DSM for DQN layer 3 and the DSM on the right represents the DSM for the hand-drawn features in Pong: the positions of the two paddles, the ball position, and the ball's velocity.

(B) Representational similarity analysis on DQN. Correlations of all the model DSMs for all games and also the hand-drawn features (HDFs) for Pong. The internal representations in Pong become more dissimilar to the pixel space and PCA model and more similar to the hand-drawn features from DQN layers 1–4. DQN representations in later layers also become more dissimilar to the input space in Space Invaders.

(C) Representational similarity analysis on fMRI data for Pong. fMRI DSMs for three ROIs were correlated with model DSMs including HDFs, each layer of DQN, PCA, and VAE. Asterisks (*) above bars indicate significance in six out of six subjects (block permutation tests, $p < 0.01$, FWER corrected for multiple comparisons). Dotted lines above bars indicate significant differences between models in six out of six subjects (block permutation tests, $p < 0.01$, FWER corrected for multiple comparisons). All brain areas in all subjects were significantly correlated to the HDF DSM and DQN layers 3 and 4. See Figure S6 for individual subject plots.

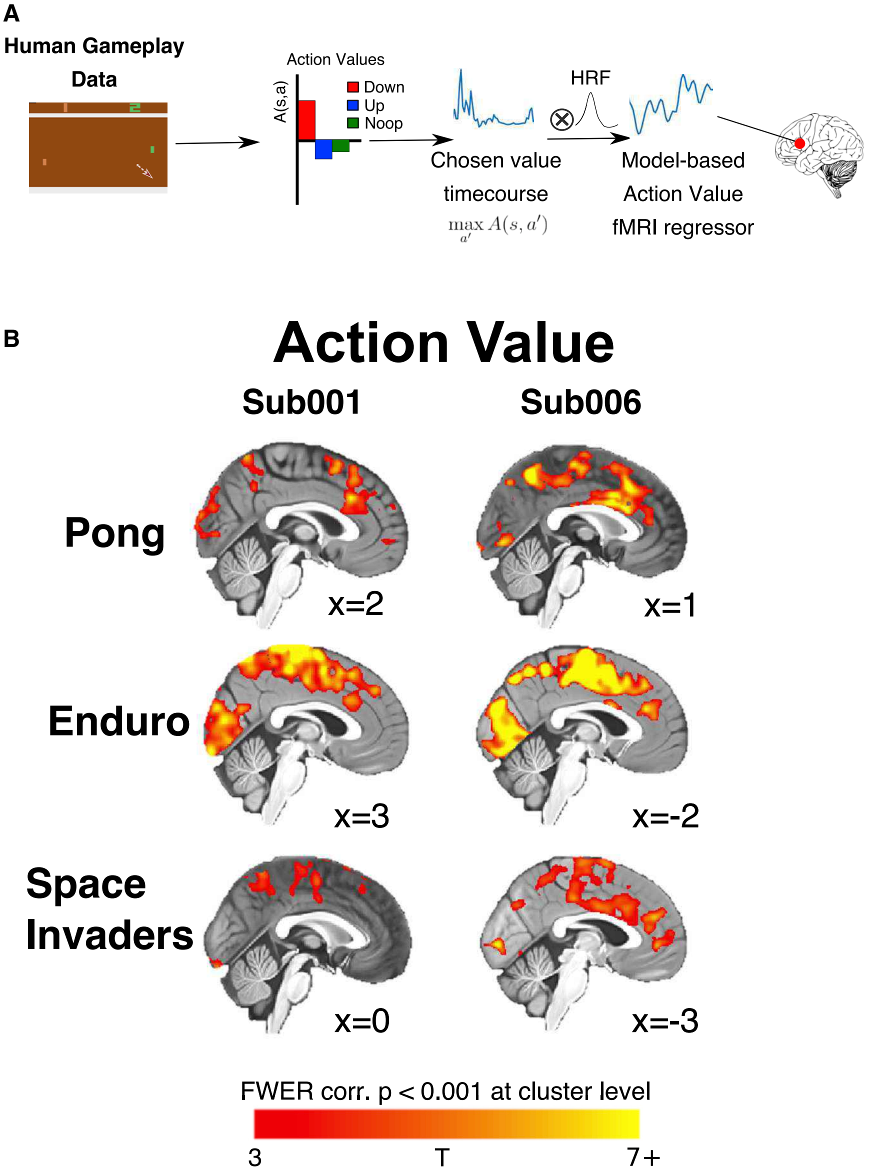


Figure 6. Action value results

(A) Depiction of action value GLMs. Human gameplay frames were run through DQN to evaluate action/chosen values. Traces were then downsampled to 10 Hz and convolved with a hemodynamic response function to reveal GLM regressors for action values.

(B) Neural encoding of action value in premotor/SMA areas. Whole-brain maps were thresholded at $p < 0.001$ (FWER corrected, cluster level). Significant representation of action value was also found in primary visual and motor cortex. Other participants are shown in Figure S7.

whether similar action value computations occur in the brain, we implemented a computational model-based general linear model (GLM) analysis (O'Doherty et al., 2007) using the DQN output as the computational model.

The action value regressor identifies regions encoding continuous values for the chosen DQN action as a function of the state the participant sees (action advantages used; see STAR methods). Significant encoding of action values was found in premotor, SMA, and primary visual and motor cortex in all games (Figures 6B and S7). Significant clusters at $p < 0.001$ (FWER corrected, cluster level) are located in motor or SMA/premotor regions for all participants in Enduro, five out of six participants in Pong (six out of six at uncorrected $p < 0.001$), and three out of six participants in Space Invaders. These results indicate that action values are computed in SMA and premotor cortex during Atari gameplay.

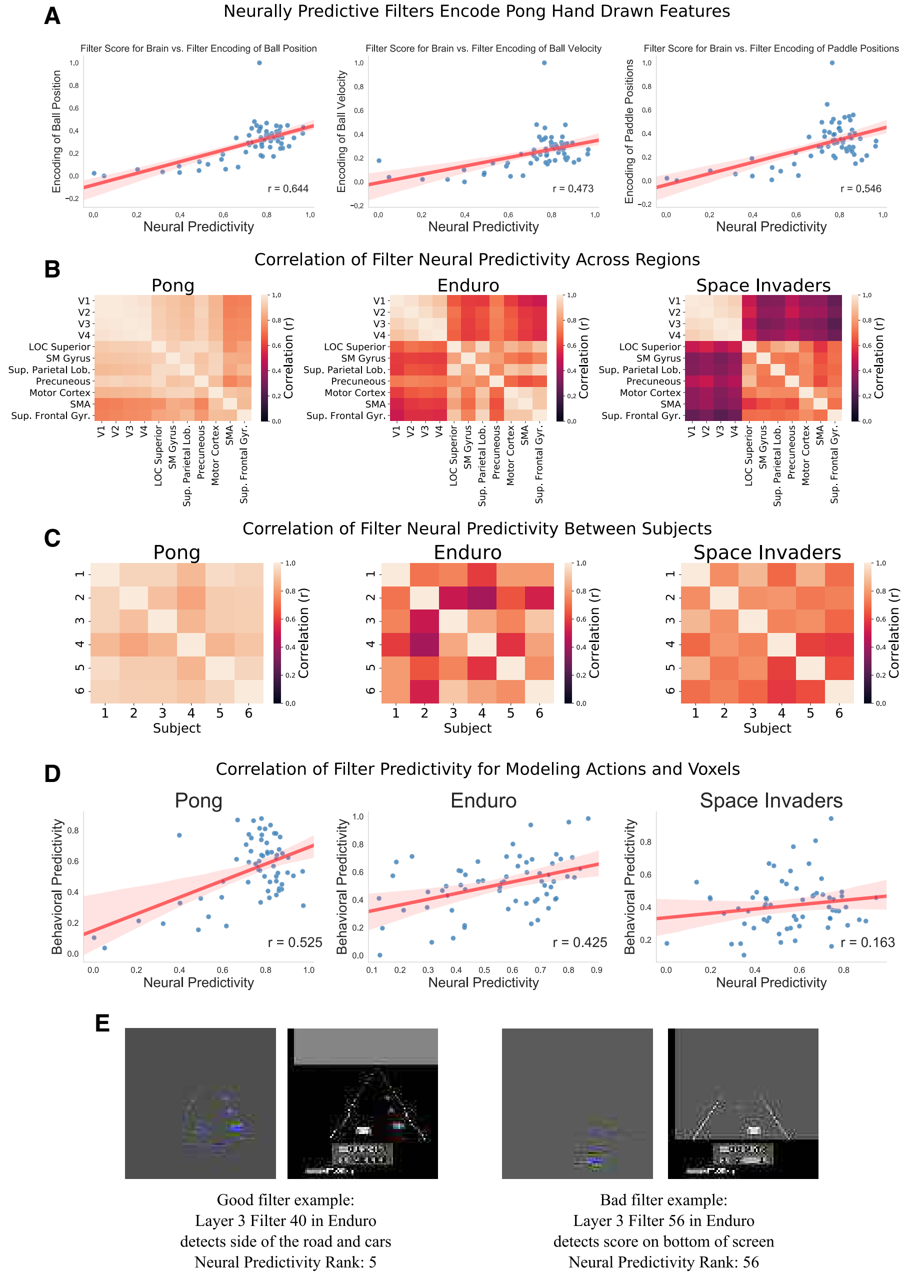
Convolutional filter analyses

Thus far, we have shown that a brain-like representation emerges most notably in DQN layers 3 and 4. We see that all ROIs, even early visual regions, prefer these last two DQN layers, suggesting multiple nonlinear transformations of the input pixels are necessary to derive features most predictive of cortical responses during Atari gameplay. However, even though the last two layers best predict voxels across the brain, different regions might prefer different artificial neurons or features within these layers. If so, could we leverage this variability to further shed insight into the features the brain is encoding and how the brain's internal representations transform from one region to another?

We test this by retraining the encoding model on each convolutional filter in the last convolutional layer separately (layer 3, 64 filters; DQN architecture illustrated in Figure 1B). The convolutional filter of a convolutional neural network (CNN) represents a feature the network is looking to detect in the input, and this feature can be somewhat visualized with guided backpropagation/deconvolution (Springenberg et al., 2015; Zeiler and Fergus, 2013) (Figure 7E). For example, early layers in a typical CNN encode low-level features such as edges and contours.

We then estimated how well each filter predicted voxel responses by averaging prediction accuracies across voxels in our ROIs, a metric we term "neural predictivity." This quantifies how well each filter explains neural responses in general and enables us to test whether neural predictivity changes across different ROIs.

The RSA results in Pong suggested that the shared representation between the brain and DQN in Pong corresponds to a mutual encoding of the spatial positions of objects. We tested this explicitly with our neural predictivity metric, as convolutional filters containing more information about high-level features may better explain brain responses. To quantify this, we calculate the degree to which a layer 3 filter encodes the Pong hand-drawn features with a mutual information metric.



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

Figure 7. Filter analyses on brain activity

(A) Neurally predictive filters in Pong encode the spatial positions of objects. Encoding models were run separately on each layer 3 filter to estimate filter neural predictivity. Separately, each filter was assessed on how much it encoded the hand labeled features in Pong. Significant correlations were found between the filter neural predictivity scores and the metric about how much information the filters encoded about the hand-labeled features in every participant ($p < 0.0001$). The average scores and correlations across participants are plotted.

(B) Correlations in filter neural predictivity scores across regions. Neural predictivity scores were correlated across regions to estimate whether the same filters are useful for predicting all neural responses or whether the mapping is more heterogeneous. In both Enduro and Space Invaders, more clustering occurs separating visual, parietal, and motor networks.

(C) Filter scores are correlated across participants. There are high correlations across all participants and nearly perfect correlations for Pong.

(D) Correlations between neural predictivity and behavioral predictivity. Axes represent normalized scores with worst filter at 0 and best filter at 1. Data aggregated across participants are depicted.

(E) Visualization of two example filters using guided backpropagation in Neon. Images to the right of each example represent an image from the human gameplay data that activate the filter the most. Gray images to the left of each example represent which parts of the pixel space affect the activation of the filter the most from this input image. Red, green, and blue reflect pixels that changed across the frames in the input. Five randomly selected filters for each game are also visualized in Figure S8A.

We found that filters with higher neural predictivity encode more information about the hand-drawn features. These correlations are significant for ball position, ball velocity, and paddle positions in every participant ($p < 0.0001$; Figure 7A), indicating that the nature of the DQN to brain mapping in Pong lies at the representation of the high-level features.

Filter neural predictivity across regions

To estimate whether different regions prefer different filters, we averaged prediction accuracies for each filter across each ROI. We then computed correlations between the 64 filter scores across regions. For Pong, high correlations between filter scores were found across all regions, suggesting that the same filters are useful for explaining responses uniformly across the brain (Figure 7B).

However, in Enduro and Space Invaders, different ROIs only have partially overlapping sets of filters mapped to them, suggesting a more heterogeneous representation across regions (Figure 7B). We found visual, parietal, and motor clusters of filter encoding with high correlations within cluster and moderate correlations between cluster. These patterns may differ from the more homogeneous filter selectivity in Pong because of the increased complexity of these games.

Neurally predictive filters generalize across participants and can predict behavior

To investigate if all our participants converge on similar useful representations for solving the task, we correlated each filter's neural predictivity score across participants. We observed high correlations between all participants in all games (Figure 7C), meaning the same filters were mapped to the brain across participants.

This result also suggests that some filters in the network are universally useful for explaining neural responses and some are universally useless. Enduro layer 3 filter 40 was one of the best-fitting filters for explaining brain activity in every participant. Through guided backpropagation (Springenberg et al., 2015), we could see that the filter detects cars and the sides of the road, which are useful features for acting in the game (Figure 7E). By contrast, Enduro layer 3 filter 56 was one of the worst-fitting filters for explaining brain activity in five out of six participants. This filter detects the score at the bottom of the screen, which is correlative of reward, since the score board changes when reward is received, but not causally related to reward.

A sample of filter deconvolutions for five random filters in each game is also plotted in Figure S8A.

Next, we evaluated how well each filter modeled human behavior by retraining the decoding human behavior model (Figure 2B) on every filter in layer 3 separately. Similar to the neural predictivity analysis, this allows us to probe how useful every layer 3 filter is for predicting human actions. We found correlations between how well a filter explains voxel activity (the neural predictivity score) and how well a filter explains human behavior (Figure 7D). This correlation was most pronounced for Enduro and Pong ($p < 0.05$ in six out of six participants in Enduro and six out of six participants in Pong but only two out of six participants in Space Invaders). Thus, the brain encodes the features most relevant for behavior, and DQN encodes features that not only are brain-like in a universal way across participants but also predict human actions.

State-space representations are nuisance invariant in PPC

An abstract state-space representation should ideally be pruned of sensory features not necessary for learning or behavior. For Pong, this involves encoding high-level features about the relevant objects in the game. However, the other two games are more complex and involve a large number of features that are difficult to hand label. Thus, rather than isolating relevant high-level features in these games, we next identify irrelevant features that an abstract state-space should ignore.

We wanted to find brain regions where the state-space encoding is insensitive to sensory information irrelevant for task performance, a pattern known as nuisance invariance (Lenc and Vedaldi, 2015). For Enduro, one nuisance variable is the weather and time of day. Driving gameplay starts off during the day and gradually becomes nighttime with various weather patterns. The colors of the pixels and visual input dramatically change, while the overall gameplay remains mostly the same. Formally, this weather variable had no relationship with the participant's actions in an information-theoretic sense (see STAR methods). A good state-space representation should localize objects independently of colors in the game. Thus, it should often project inputs that are very far away in the pixel space to similar regions of the latent state-space if an agent should act similarly across them (illustrated in Figure 8A). In contrast, even small changes in pixel space may necessitate opposite actions. For example, in Figure 8A, an agent should move left or right depending on the location of the car in front of it, even though the two pairs of frames are perceptually similar.

For Space Invaders, the number of on-screen invaders explains a lot of variance in the pixel space but has a marginal effect on what actions participants take (see STAR methods). This is because as an agent kills more invaders, the screen becomes more and more black. This information does not heavily impact which actions an agent should take, because the relative positions of the invaders above an agent matter the most.

To estimate whether ROI representations are nuisance invariant, we quantified the mutual information between a filter and the nuisances identified for Enduro and Space Invaders, giving each filter a metric for how insensitive it was to the nuisances (see STAR methods). We computed the correlation between each filter's nuisance invariance, and its neural predictivity in a ROI, which we define as a nuisance invariance score for each region (normalized across voxels; see STAR methods). Simply put, this score estimates how each region prefers the filters that are nuisance invariant.

Regions in PPC and in the late dorsal visual stream (i.e., lateral occipital cortex [LOC]) were more insensitive to nuisances than early visual cortex regions V1–V4 (Figures 8B, 8C, S8B, and S8C) in both games. Early visual cortex regions exhibited the lowest nuisance invariance scores in both games, suggesting that filters mapped to these regions still encoded the low-level nuisance variables. Additionally, LOC, which is later in the dorsal

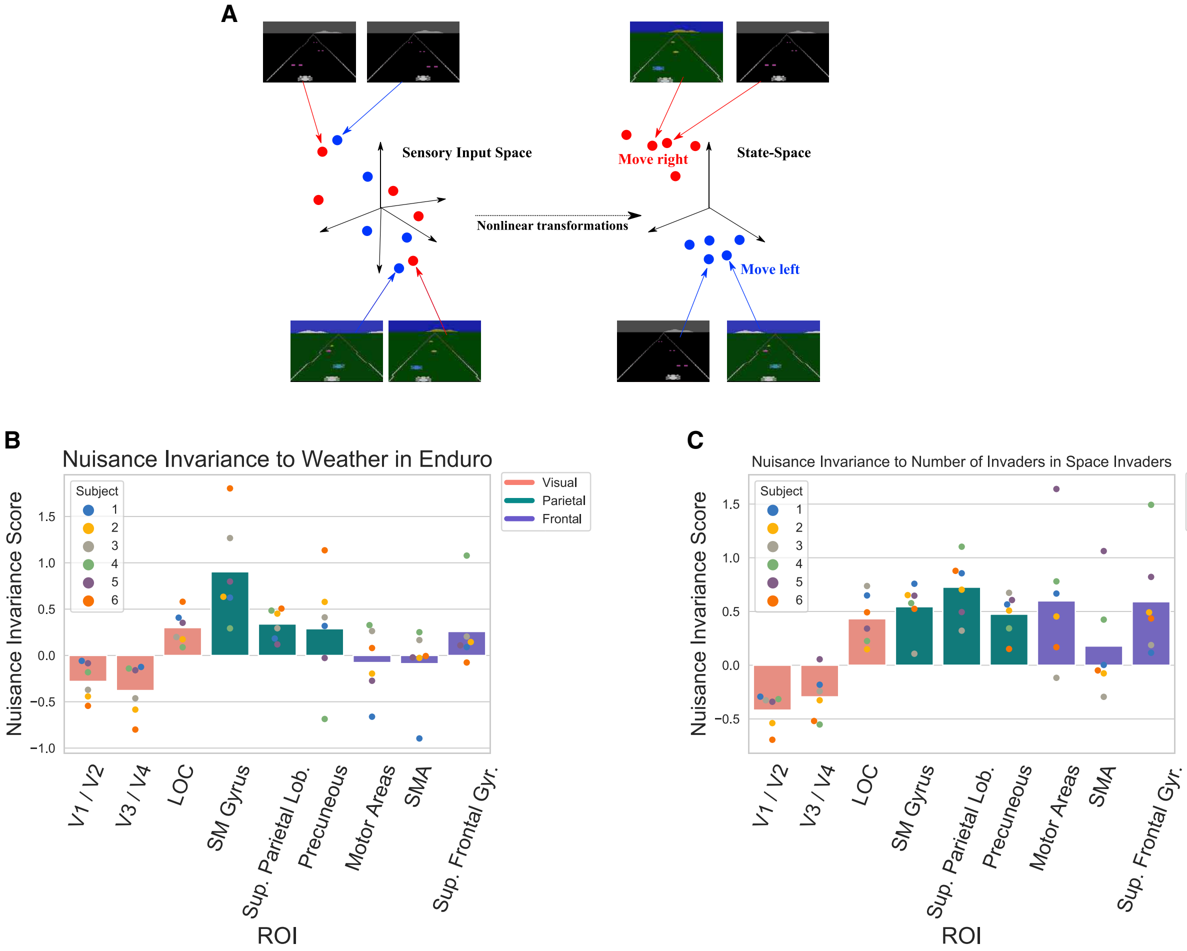


Figure 8. Representations become more insensitive to nuisances in posterior parietal cortex

(A) Illustration of what a useful representation would do in Enduro.

The sky color changes frequently, but these changes have no effect on human actions. The input space on the left depicts how situations are clustered by perceptual features such as color in the pixel space. Within each night/day cluster, there are examples of a car in front of an agent on both the right and left. Therefore, one must take opposite actions in each scenario to avoid a collision. A good state-space localizes the positions of the relevant objects independently of visual nuisances. The resulting state-space representation on the right clusters together perceptually dissimilar situations if they share the same underlying semantic meaning for the policy.

(B) Nuisance invariance to weather/time of day in Enduro. We calculate a nuisance invariance score in every region. This score is defined as the correlation of a filter's neural predictivity in a region and that filter's nuisance invariance to weather. The motor area ROI includes both the primary motor cortex and premotor cortex.

(C) Nuisance invariance to number of invaders on the screen in Space Invaders. We similarly calculate a nuisance invariance score for every region as defined in (B). For the game Space Invaders, the proxy nuisance variable was the number of invaders on the screen.

visual pathway, had a higher nuisance invariance score than these earlier visual regions. For Enduro, a PPC region exhibited the highest or second highest score of any region in every participant. In five out of six participants in Space Invaders, premotor/prefrontal cortex regions also exhibited high nuisance invariance scores.

These results suggest that irrelevant visual input is stripped from the neural code as information passes through the dorsal visual stream to the PPC. This leads to a lower-dimensional, compressed, and abstract representation that projects similar game situations to the same part of the state-space as depicted in Figure 8A.

DISCUSSION

One of the major unresolved questions in decision neuroscience is how relevant sensory features are identified and structured to aid action evaluation and selection in real-world scenarios. Here, we addressed this question by having humans play complex Atari games (Pong, Enduro, and Space Invaders) in an fMRI scanner. Taking our cue from advances in artificial intelligence (Mnih et al., 2015), we utilized a deep RL algorithm as a model for how to solve the task representation problem inherent in these tasks. We demonstrate that representations in DQN show a remarkable similarity to those used by humans. Features in DQN hidden layers predict human actions and fMRI activity in a distributed sensorimotor network extending from the dorsal visual stream and PPC to premotor areas. Not only does the DQN model significantly outperform control models of varying levels of complexity, but DQN features also explain unique variance in these ROIs when controlling for the other models. These results suggest that these regions do not simply encode low-level sensory information but produce a state representation that links sensory information to reward and action selection. Further validating our approach, we found an encoding of the action value output of DQN in premotor cortex/SMA, along with primary visual and motor cortex. In alignment with a traditional trial-based study (Wunderlich et al., 2009), our results support a role for SMA in action valuation while generalizing these findings to an environment with high-dimensional state dynamics.

Our findings build on a growing catalog of intriguing similarities between DNNs and the brain (Eickenberg et al., 2017; Güçlü and van Gerven, 2015; Iigaya et al., 2020; Khaligh-Razavi and Kriegeskorte, 2014; Wang et al., 2018; Wen et al., 2018; Yamins et al., 2014; Yamins and DiCarlo, 2016). Unlike some studies, we did not find a gradient of abstraction mapping early layers to early visual regions and later layers to later regions. All ROIs consistently preferred DQN layers 3 and 4. By examining the representational geometry of different DQN layers, we could identify principles accounting for this pattern. For Pong and Space Invaders especially, the internal representation is not dissimilar to the pixel space until DQN layers 3 and 4, whereas the information reaching early visual cortex may already be heavily compressed. Prior research suggests considerable compression and nonlinear processing of visual inputs occurs before the cortex, via the retina, lateral geniculate nucleus (LGN), eye movements, and feedback connections (Gollisch and Meister, 2010; Hayhoe and Ballard, 2005; Hosoya et al., 2005; Kietzmann et al., 2019). For Pong, both the brain and later layers of DQN represent high-level features about the spatial positions of the ball and paddles. Early visual regions may have more similarity to layers 3 and 4, because DQN only disentangles these features from the pixel space in later layers. Additionally, many of the DNNs used in visual neuroscience have eight or more layers, with layers 2–4 often constituting the most similarity to early visual cortex rather than layer 1 (Khaligh-Razavi and Kriegeskorte, 2014; Seeliger et al., 2018; Wen et al., 2018). If the network had more layers, representational gradients at the layer level might emerge, perhaps with early visual regions still preferring layers 3 and 4 but more anterior regions mapping to deeper layers. Yet, for our purposes, DQN provides a satisfactory account of both behavior and neural data, with relevant variance for explaining cortical activity packed into layers 3 and 4. Thus, we analyzed how different features within layer 3 (the last convolutional layer) explain activity across the brain.

To do so, we retrained separate encoding models on stimulus features from the convolutional filters in DQN layer 3. This analysis showed that filters most predictive of voxel activity are also predictive of human behavior, suggesting that these features are used by the brain to guide behavior. Filter selectivity is highly correlated between participants, indicating a common task representation across individuals. For Pong, the filter analysis provided more evidence that this common state-space represents high-level features such as the spatial positions of the relevant objects. This is in line with a recent proposal that the dorsal stream and PPC encode spatial positions of objects by projecting high-dimensional inputs onto a low-dimensional manifold of physical space (Summerfield et al., 2020).

For Enduro and Space Invaders, the mapping of DQN features to the brain was more heterogeneous between regions, suggesting that different regions prefer different underlying features in the network. PPC areas encoded features that are more generalizable and nuisance invariant than early visual regions. Thus, PPC is able to ignore and abstract away information from the sensory stream that is not relevant for behavioral performance, such as changing colors and backgrounds in Enduro. This suggests that PPC may be a central nexus for isolating behaviorally relevant stimuli by integrating visual, cognitive, and motor information (Freedman and Ibos, 2018). A substantial literature in motor neuroscience also implicates PPC in sensorimotor transformations, linking perception to decision-making and action (Andersen and Buneo, 2002; Andersen and Cui, 2009; Gold and Shadlen, 2007). The present work suggests that these past findings and proposed theories can be integrated into a broader conceptualization of PPC as encoding abstract state-space features linking perception to learning and action selection.

Overall, our results point toward key properties fostering an effective state-space for tasks of real-world complexity. Initially, compression to a lower-dimensional space takes place to avoid the curse of dimensionality, where learning complexity scales exponentially with the number of states to learn about. However, exploiting the raw statistical properties of the input data, as in unsupervised learning techniques, is not enough; it must also disentangle a purely sensory manifold into appropriate axes linked to rewards and the actions that deliver them (DiCarlo and Cox, 2007; Higgins et al., 2018a). For Pong, these axes code for relevant data-generating factors, the spatial positions of the ball and paddles. In addition, a state-space would likely benefit from being invariant to nuisances irrelevant for task performance (Lenc and Vedaldi, 2015). This property further reduces state-space dimensionality by only transmitting useful signals through an information bottleneck (Achille and Soatto, 2018; Shwartz-Ziv and Tishby, 2017). This added compression helps protect against overfitting by shaping an abstract task representation orthogonal to low-level sensory properties that can change in future settings. Humans are clearly equipped with abstract representations with this property (Behrens et al., 2018), as they can seamlessly adapt to novel circumstances, such as driving on new roads without having to relearn the driving process.

It should be noted that the DQN objective does not explicitly promote the learning of nuisance-invariant representations, and most filters still retain information about the nuisances we highlighted. Additionally, DQN performance is not robust to visual changes (such as in image contrast) during testing if the change was not in the training distribution. Most deep RL algorithms are not explicitly trained to learn a representation but are trained to approximate value-based and policy-based functions and thereby learn a task representation as a side effect. These approaches suffer sample efficiency and generalization issues (Kaiser et al., 2020; Lake et al., 2016). Therefore, deep

RL algorithms likely would benefit from explicitly learning a representation with the principles we previously outlined, alongside other inductive biases humans possess about the structure of the world (Botvinick et al., 2019). Methods for accomplishing this goal are being developed in the emerging field of state representation learning (Anand et al., 2019; Higgins et al., 2018b; Jaderberg et al., 2016; Lesort et al., 2018; van den Oord et al., 2019; Srinivas et al., 2020; Zhang et al., 2020). We also espouse more cross-talk between decision neuroscientists and artificial intelligence researchers at the level of representations for a RL system; thus far, most of the interaction between these fields has occurred at the level of learning signals (Botvinick et al., 2020; Dabney et al., 2020; Niv and Langdon, 2016; Wang et al., 2018).

The present findings suggest that even with notable architectural differences between the human brain and deep RL models, DQN still does remarkably well in capturing variance in both human behavior and brain activity throughout the dorsal visual stream and the parietal and premotor cortices in high-dimensional decision-making contexts. These findings further help to establish the deep and sustained relationship between progress in artificial intelligence and in computational neuroscience. Our results suggest that this interdisciplinary interplay is continuing to evolve and that in particular, a synergy between deep RL and decision neuroscience offers the continuing prospect to yield rich insights about the internal representations of intelligent systems.

STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- KEY RESOURCES TABLE
- RESOURCE AVAILABILITY
  - ○ Lead contact
  - ○ Materials availability
  - ○ Data and code availability
- EXPERIMENTAL MODEL AND SUBJECT DETAILS
- METHOD DETAILS
  - ○ Experimental paradigm/Atari gameplay
  - ○ fMRI data acquisition
- QUANTIFICATION AND STATISTICAL ANALYSIS
  - ○ fMRI preprocessing
  - ○ Deep Q-network training
  - ○ Human actions analyses
  - ○ Encoding model
  - ○ Regions of interest and atlases
  - ○ Encoding model control analyses
  - ○ Motor
  - ○ PCA
  - ○ Cross game DQN
  - ○ VAE
  - ○ General linear model (GLM) control analysis
  - ○ Control region analysis
  - ○ Representational similarity analysis
  - ○ Computational model-based GLMs
  - ○ Filter analyses
  - ○ Nuisance invariance scores

SUPPLEMENTAL INFORMATION

Supplemental Information can be found online at https://doi.org/10.1016/j.neuron.2020.11.021.

ACKNOWLEDGMENTS

This work is supported by NIDA (grant R01DA040011 to J.O.D. and L.C.) and the NIMH Caltech Conte Center for the Neurobiology of Social Decision-Making (grant P50 MH094258 to J.O.D.). We would like to thank Kiyohito Iigaya and other members of the O'Doherty lab for helpful feedback and discussions.

AUTHOR CONTRIBUTIONS

L.C., J.C., and J.P.O. designed the project. L.C. and J.C. developed experimental protocol and collected data. L.C. performed the analyses and wrote the draft of the manuscript. L.C., J.C., Y.Y., and J.P.O. discussed analyses and edited the manuscript. J.P.O. acquired funding.

DECLARATION OF INTERESTS

The authors declare no competing interests.

Received: July 10, 2020
Revised: October 15, 2020
Accepted: November 17, 2020
Published: December 15, 2020

REFERENCES

Achille, A., and Soatto, S. (2018). Emergence of invariance and disentanglement in deep representations. arXiv, 1706.01350 https://arxiv.org/abs/1706.01350.

Anand, A., Racah, E., Ozair, S., Bengio, Y., Côté, M.A., and Hjelm, R.D. (2019). Unsupervised state representation learning in Atari. Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), 8769–8782.

Andersen, R.A., and Buneo, C.A. (2002). Intentional maps in posterior parietal cortex. Annu. Rev. Neurosci. *25*, 189–220.

Andersen, R.A., and Cui, H. (2009). Intention, action planning, and decision making in parietal-frontal circuits. Neuron *63*, 568–583.

Avants, B., Tustison, N., and Song, G. (2009). Advanced Normalization Tools: V1.0. Insight J. *2009*, 681.

Behrens, T.E.J., Muller, T.H., Whittington, J.C.R., Mark, S., Baram, A.B., Stachenfeld, K.L., and Kurth-Nelson, Z. (2018). What is a cognitive map? Organizing knowledge for flexible behavior. Neuron *100*, 490–509.

Bellemare, M.G., Naddaf, Y., Veness, J., and Bowling, M. (2013). The arcade learning environment: an evaluation platform for general agents. J. Artif. Intell. Res. *47*, 253–279.

Botvinick, M., Ritter, S., Wang, J.X., Kurth-Nelson, Z., Blundell, C., and Hassabis, D. (2019). Reinforcement learning, fast and slow. Trends Cogn. Sci. *23*, 408–422.

Botvinick, M., Wang, J.X., Dabney, W., Miller, K.J., and Kurth-Nelson, Z. (2020). Deep reinforcement learning and its neuroscientific implications. Neuron *107*, 603–616.

Chang, L., and Tsao, D.Y. (2017). The code for facial identity in the primate brain. Cell *169*, 1013–1028.e14.

Dabney, W., Kurth-Nelson, Z., Uchida, N., Starkweather, C.K., Hassabis, D., Munos, R., and Botvinick, M. (2020). A distributional code for value in dopamine-based reinforcement learning. Nature *577*, 671–675.

DiCarlo, J.J., and Cox, D.D. (2007). Untangling invariant object recognition. Trends Cogn. Sci. *11*, 333–341.

Eickenberg, M., Gramfort, A., Varoquaux, G., and Thirion, B. (2017). Seeing it all: convolutional network layers map the function of the human visual system. Neuroimage *152*, 184–194.

Freedman, D.J., and Ibos, G. (2018). An integrative framework for sensory, motor, and cognitive functions of the posterior parietal cortex. Neuron *97*, 1219–1234.

Gold, J.I., and Shadlen, M.N. (2007). The neural basis of decision making. Annu. Rev. Neurosci. *30*, 535–574.

Gollisch, T., and Meister, M. (2010). Eye smarter than scientists believed: neural computations in circuits of the retina. Neuron *65*, 150–164.

Güçlü, U., and van Gerven, M.A.J. (2015). Deep neural networks reveal a gradient in the complexity of neural representations across the ventral stream. J. Neurosci. *35*, 10005–10014.

Ha, D., and Schmidhuber, J. (2018). World models. arXiv, 1803.10122 https://arxiv.org/abs/1803.10122.

Hanke, M., Halchenko, Y.O., Sederberg, P.B., Hanson, S.J., Haxby, J.V., and Pollmann, S. (2009). PyMVPA: A python toolbox for multivariate pattern analysis of fMRI data. Neuroinformatics *7*, 37–53.

Haxby, J.V., Connolly, A.C., and Guntupalli, J.S. (2014). Decoding neural representational spaces using multivariate pattern analysis. Annu. Rev. Neurosci. *37*, 435–456.

Hayhoe, M., and Ballard, D. (2005). Eye movements in natural behavior. Trends Cogn. Sci. *9*, 188–194.

Higgins, I., Amos, D., Pfau, D., Racaniere, S., Matthey, L., Rezende, D., and Lerchner, A. (2018a). Towards a definition of disentangled representations. arXiv, 1812.02230 https://arxiv.org/abs/1812.02230.

Higgins, I., Pal, A., Rusu, A.A., Matthey, L., Burgess, C.P., Pritzel, A., Botvinick, M., Blundell, C., and Lerchner, A. (2018b). DARLA: improving zero-shot transfer in reinforcement learning. Proceedings of the 34th International Conference on Machine Learning *70*, 1480–1490.

Hosoya, T., Baccus, S.A., and Meister, M. (2005). Dynamic predictive coding by the retina. Nature *436*, 71–77.

Iigaya, K., Yi, S., Wahle, I.A., Tanwisuth, K., and O'Doherty, J.P. (2020). Aesthetic preference for art emerges from a weighted integration over hierarchically structured visual features in the brain. bioRxiv. https://doi.org/10.1101/2020.

Jaderberg, M., Mnih, V., Czarnecki, W.M., Schaul, T., Leibo, J.Z., Silver, D., and Kavukcuoglu, K. (2016). Reinforcement learning with unsupervised auxiliary tasks. arXiv, 1611.05397 https://arxiv.org/abs/1611.05397.

Kaiser, L., Babaeizadeh, M., Milos, P., Osinski, B., Campbell, R.H., Czechowski, K., Erhan, D., Finn, C., Kozakowski, P., Levine, S., et al. (2020). Model-based reinforcement learning for Atari. arXiv, 1903.00374 https://arxiv.org/abs/1903.00374.

Kay, K.N., Naselaris, T., Prenger, R.J., and Gallant, J.L. (2008). Identifying natural images from human brain activity. Nature *452*, 352–355.

Khaligh-Razavi, S.-M., and Kriegeskorte, N. (2014). Deep supervised, but not unsupervised, models may explain IT cortical representation. PLoS Comput. Biol. *10*, e1003915.

Kietzmann, T.C., Spoerer, C.J., Sörensen, L.K.A., Cichy, R.M., Hauk, O., and Kriegeskorte, N. (2019). Recurrence is required to capture the representational dynamics of the human visual system. Proc. Natl. Acad. Sci. USA *116*, 21854–21863.

Kingma, D.P., and Welling, M. (2014). Auto-encoding variational Bayes. arXiv, 1312.6114 https://arxiv.org/abs/1312.6114.

Kriegeskorte, N., Mur, M., and Bandettini, P. (2008). Representational similarity analysis: connecting the branches of systems neuroscience. Front. Syst. Neurosci. *2*, 4.

Lake, B.M., Ullman, T.D., Tenenbaum, J.B., and Gershman, S.J. (2016). Building machines that learn and think like people. arXiv, 1604.00289 https://arxiv.org/abs/1604.00289.

Lenc, K., and Vedaldi, A. (2015). Understanding image representations by measuring their equivariance and equivalence. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 991–999.

Lesort, T., Díaz-Rodríguez, N., Goudou, J.I., and Filliat, D. (2018). State representation learning for control: An overview. Neural Netw. *108*, 379–392.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. Nature *518*, 529–533.

Mohamed, S., and Rezende, D.J. (2015). Variational information maximisation for intrinsically motivated reinforcement learning. arXiv, 1509.08731 https://arxiv.org/abs/1509.08731.

Niv, Y. (2019). Learning task-state representations. Nat. Neurosci. *22*, 1544–1553.

Niv, Y., and Langdon, A. (2016). Reinforcement learning with Marr. Curr. Opin. Behav. *11*, 67–73.

O'Doherty, J., Dayan, P., Schultz, J., Deichmann, R., Friston, K., and Dolan, R.J. (2004). Dissociable roles of ventral and dorsal striatum in instrumental conditioning. Science *304*, 452–454.

O'Doherty, J.P., Hampton, A., and Kim, H. (2007). Model-based fMRI and its application to reward learning and decision making. Ann. N Y Acad. Sci. *1104*, 35–53.

Olshausen, B.A., and Field, D.J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Nature *381*, 607–609.

Penny, W., Friston, K., Ashburner, J., Kiebel, S., and Nichols, T. (2011). Statistical Parametric Mapping: The Analysis of Functional Brain Images (Elsevier).

Schultz, W. (1998). Predictive reward signal of dopamine neurons. J. Neurophysiol. *80*, 1–27.

Schultz, W., Dayan, P., and Montague, P.R. (1997). A neural substrate of prediction and reward. Science *275*, 1593–1599.

Seeliger, K., Fritsche, M., Güçlü, U., Schoenmakers, S., Schoffelen, J.-M., Bosch, S.E., and van Gerven, M.A.J. (2018). Convolutional neural network-based encoding and decoding of visual object recognition in space and time. NeuroImage *180*, 253–266.

Shwartz-Ziv, R., and Tishby, N. (2017). Opening the black box of deep neural networks via information. arXiv, 1703.00810 https://arxiv.org/abs/1703.00810.

Smith, S.M., Jenkinson, M., Woolrich, M.W., Beckmann, C.F., Behrens, T.E.J., Johansen-Berg, H., Bannister, P.R., De Luca, M., Drobnjak, I., Flitney, D.E., et al. (2004). Advances in functional and structural MR image analysis and implementation as FSL. Neuroimage *23* (*Suppl 1*), S208–S219.

Springenberg, J.T., Dosovitskiy, A., Brox, T., and Riedmiller, M. (2015). Striving for simplicity: the all convolutional net. arXiv, 1412.6806 https://arxiv.org/abs/1412.6806.

Srinivas, A., Laskin, M., and Abbeel, P. (2020). CURL: contrastive unsupervised representations for reinforcement learning. arXiv, 2004.04136 https://arxiv.org/abs/2004.04136.

Steinberg, E.E., Keiflin, R., Boivin, J.R., Witten, I.B., Deisseroth, K., and Janak, P.H. (2013). A causal link between prediction errors, dopamine neurons and learning. Nat. Neurosci. *16*, 966–973.

Summerfield, C., Luyckx, F., and Sheahan, H. (2020). Structure learning and the posterior parietal cortex. Prog. Neurobiol. *184*, 101717.

Sutton, R.S., and Barto, A.G. (2018). Reinforcement Learning: An Introduction (MIT Press).

Tyszka, J.M., and Pauli, W.M. (2016). In vivo delineation of subdivisions of the human amygdaloid complex in a high-resolution group template. Hum. Brain Mapp. *37*, 3979–3998.

van den Oord, A., Li, Y., and Vinyals, O. (2019). Representation learning with contrastive predictive coding. arXiv, 1807.03748 https://arxiv.org/abs/1807.03748.

Wang, J.X., Kurth-Nelson, Z., Kumaran, D., Tirumala, D., Soyer, H., Leibo, J.Z., Hassabis, D., and Botvinick, M. (2018). Prefrontal cortex as a meta-reinforcement learning system. Nat. Neurosci. *21*, 860–868.

Watkins, C.J.C.H., and Dayan, P. (1992). Q-learning. Mach. Learn. *8*, 279–292.

Watter, M., Springenberg, J.T., Boedecker, J., and Riedmiller, M. (2015). Embed to control: a locally linear latent dynamics model for control from raw images. arXiv, 1506.07365 https://arxiv.org/abs/1506.07365.

Wen, H., Shi, J., Zhang, Y., Lu, K.-H., Cao, J., and Liu, Z. (2018). Neural encoding and decoding with deep learning for dynamic natural vision. Cereb. Cortex *28*, 4136–4160.

Wunderlich, K., Rangel, A., and O'Doherty, J.P. (2009). Neural computations underlying action-based decision making in the human brain. Proc. Natl. Acad. Sci. USA *106*, 17199–17204.

Yamins, D.L.K., and DiCarlo, J.J. (2016). Using goal-driven deep learning models to understand sensory cortex. Nat. Neurosci. *19*, 356–365.

Yamins, D.L.K., Hong, H., Cadieu, C.F., Solomon, E.A., Seibert, D., and DiCarlo, J.J. (2014). Performance-optimized hierarchical models predict neural responses in higher visual cortex. Proc. Natl. Acad. Sci. USA *111*, 8619–8624.

Zeiler, M.D., and Fergus, R. (2013). Visualizing and understanding convolutional networks. arXiv, 1311.2901 https://arxiv.org/abs/1311.2901.

Zhang, A., McAllister, R., Calandra, R., Gal, Y., and Levine, S. (2020). Learning invariant representations for reinforcement learning without reconstruction. arXiv, 2006.10742 https://arxiv.org/pdf/2006.10742.

Please cite this article in press as: Cross et al., Using deep reinforcement learning to reveal how the brain encodes abstract state-space representations in high-dimensional environments, Neuron (2020), https://doi.org/10.1016/j.neuron.2020.11.021

STAR★METHODS

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited Data		
fMRI Data	This paper	TBD when uploaded to NIH database
Software and Algorithms		
MATLAB_R2019a	MathWorks	https://www.mathworks.com/
SPM12	Penny et al., 2011	https://www.fil.ion.ucl.ac.uk/spm/software/spm12/
FSLv5.0	Smith et al. 2004	https://fsl.fmrib.ox.ac.uk/fsl/fslwiki
Advanced Normalization Tools (ANTs) v1.9	Avants et al., 2009	http://stnava.github.io/ANTs/
Python 2.7 and Python 3.5	Python	https://www.python.org/
PyMVPA Version 2.6.0	Hanke et al., 2009	http://www.pymvpa.org/index.html
Simple DQN	Tambet Matiisen	https://github.com/tambetm/simple_dqn
Tensorflow 2.1	Tensorflow	https://www.tensorflow.org/
Arcade Learning Environment	Bellemare et al., 2013	https://github.com/mgbellemare/Arcade-Learning-Environment#:~:text=The%20Arcade%20Learning%20Environment%20(ALE)%20is%20a%20simple%20object%2D,of%20emulation%20from%20agent%20design.
Custom code	This paper	https://github.com/locross93/Atari-Project

RESOURCE AVAILABILITY

Lead contact

Any additional information and requests for resources or data should be directed to and will be fulfilled by the Lead Contact, Logan Cross at lcross@caltech.edu.

Materials availability

This study did not generate new unique materials.

Data and code availability

Due to Conte Center NIH funding, the fMRI data will be uploaded to the NIH database in the near future.

The code for this project is available upon request from the corresponding authors. The most fundamental parts of our code base will be released on GitHub (https://github.com/locross93/Atari-Project).

EXPERIMENTAL MODEL AND SUBJECT DETAILS

We recruited six healthy participants from the Caltech and Pasadena community (4 male and 2 females, age 26 ± 3.4). All participants performed the tasks over the course of four separate days and received a participation fee of $40 a day. The Caltech Institutional Review Board approved the protocol, and all participants gave their informed consent on each day of the experiment.

METHOD DETAILS

Experimental paradigm/Atari gameplay

Across the four days of the experiment, each participant went through 33 runs of gameplay. The runs were 10 minutes in duration, with 8 minutes of gameplay in between a minute of rest and a fixation cross before and after gameplay. Eyetracking was recorded, but not analyzed for this paper. Each participant played the games Space Invaders, Pong, and Enduro 11 times each. On day 1, each game was played twice, in random order with the one constraint of never playing the same game twice in a row. The six runs were then followed by anatomical scans on day 1. On days 2-4, each game was played three times, in random order with the same constraint of never playing the same game twice in a row. Before scanning on the first day, each participant went through a training session to become familiar with each game by playing each game for 5 minutes on a laptop.

The Atari games were presented through the Arcade Learning Environment (Bellemare et al., 2013), with modified code to log actions, rewards, MRI pulses, and frames with proper timestamps. A button box with four buttons was used as an Atari controller (Figure 1A). Participants held the button box with two hands, using their left thumb to press the 1 and 2 buttons corresponding to move left and move right respectively, and using their right thumb to press the 3 and 4 buttons to hit brake and fire respectively. Brake is only used in Enduro, and fire is only used in Enduro and Space Invaders.

In Enduro, participants control a race car that must move as fast as possible while avoiding other cars on the road. Participants get a reward of 1 for every car they pass, and the main objective is to pass a certain number of cars before the end of the day (200 cars in level 1 and 300 cars in level 2). The sky and weather patterns change throughout the gameplay to simulate the passing of time in the day (‘sunny’, ‘snow’, ‘blue dusk’, ‘red dusk’, ‘night’, ‘fog’, ‘sunrise’), with the sky eventually becoming black and the sun beginning to rise before time runs out after 13312 frames.

In Pong, points are awarded to a player when the white ball moves past their opponent’s paddle. Participants control the green paddle on the right side of the screen and try to defend their goal and score on their opponent’s goal by moving their paddle up and down in the white ball’s path.

In Space Invaders, participants control a green ship that can move from left to right at the bottom of the screen. The objective is to destroy enemy ships to get reward and avoid being hit by missiles from the enemy ships while having 3 lives before the game ends.

fMRI data acquisition

We collected two datasets on two separate scanners at the Caltech Brain Imaging Center (Pasadena, CA). The first dataset included two participants and was collected using a 3T Siemens Magneto TrioTim scanner. After an upgrade to a Siemens Prisma, a second dataset was collected with four participants. Both datasets used a 32-channel radio frequency coil. These parameters were shared across the two sequences: whole-brain BOLD signal acquired using multiband acceleration of 4, 56 slices, voxel size = 2.5mm isotropic, TR = 1,000 ms, TE = 30 ms, FA = 60°, FOV = 200mm x 200mm. At the end of the first day of scanning, T1 and T2 weighted anatomical high-resolution scans were collected with 0.9mm isotropic resolution.

QUANTIFICATION AND STATISTICAL ANALYSIS

fMRI preprocessing

Data was preprocessed using a standard pipeline for preprocessing of multiband data. Using FSL (Smith et al., 2004), images were brain extracted, realigned, high-pass filtered (100 s threshold), and unwarped. Images were denoised by ICA component removal. Components were extracted using FSL’s Melodic, classified into signal or noise with a classifier trained on separate datasets for the first dataset, and manually classified for the second dataset since the scanner was different from the one used in the classifier training set. T2 images were aligned to T1 images with FSL FLIRT, and then both were normalized to standard space using ANTs (using CIT168 high resolution T1 and T2 templates (Avants et al., 2009; Tyszka and Pauli, 2016)). The functional data was first co-registered to anatomical images using FSL’s FLIRT, then registered to the normalized T2 using ANTs. For GLMs in SPM 12 (Penny et al., 2011) (encoding model control GLM and action value analysis) the data was spatially smoothed in FSL with a 5-mm FWHM Gaussian kernel. Smoothing was not initially applied to the fMRI images for the voxelwise encoding model analyses to preserve fine-grained detail at the voxel level but was applied with a 5-mm kernel for visualization.

Deep Q-network training

Deep Q-networks were trained separately for each of the three games using the Neon deep learning library, by making modifications to open source code (https://github.com/tambetm/simple_dqn). As in the original paper (Mnih et al., 2015), DQN takes a tensor of four input frames as input, has three convolutional layers (Layer 1: 32 filters of 8x8 with stride of 4; Layer 2: 64 filters of 4x4 with stride of 2; Layer 3: 64 filters of 3x3 with stride of 1) followed by one fully connected layer (512 units), and outputs Q-values for every available action. DQN takes the action with the highest Q-value. Convolutional layers are locally connected with each neuron having a receptive field. Convolutional filters learn visual features which are then convolved across the input to detect the presence of that feature. Fully connected layers do not have this local connectivity as every neuron is connected to every neuron in the previous layer.

The Arcade Learning Environment was used as the Atari environment during training (Bellemare et al., 2013). The training consisted of 100 epochs of 250,000 steps in each epoch for each game. One modification was made for Pong by restricting the action set to noop, up, and down, since the default available action set for this game includes redundant actions up/right, and down/left.

To output Q-values and hidden unit activations that are used for all analyses, the human gameplay frames were run through the trained network. Since the input to DQN is a tensor of four consecutive images, a frame from the human data is concatenated with its three preceding frames. Thus, the fourth frame in a run is the first one put through DQN. In Enduro, each level is won after passing 200 cars in the first level and 300 cars in the second level, signified by flags appearing on the scoreboard. When this happens, the game engine no longer gives reward until the day ends/clock stops even though the participant is still tasked with controlling the car and trying to avoid other cars. Thus, the network would detect the flags and predict 0 reward when this happened, resulting in meaningless Q-value traces. This would happen occasionally in a participant’s run and would last a couple of minutes. To ensure that the activations and Q-values we extracted from the network were useful, we altered the images from Enduro human gameplay before they were put through DQN so that the scoreboard would never change. Specifically, the scoreboard from a reference image midway through a run was copied into every frame.

Human actions analyses

To analyze the human state-space in relation to the DQN's state space, we analyzed the actions participants took and how these compare to the actions DQN selects when fed the human gameplay data. This initially involved plotting the distributions for the actions executed by DQN and human participants (Figure S1A). To analyze DQN's action values with respect to human actions, the Q-values for every participant were included after downsampling by 10 and removing the first 100 frames in a run. Action values were computed with an action advantage function by subtracting the average Q-value as an action-independent baseline (Sutton and Barto, 2018). This allows us to isolate action related variance from state value related variance. Action values/advantages were then LOWESS smoothed across frames (using the Statsmodels python package with the "frac" parameter = 0.005) and normalized with sklearn's StandardScaler. All the frames involving a "move-left" or "move-right" human action were selected, including combination actions (ie. "fire left"). Then average action values for the corresponding frames are computed across a human action category. For Enduro and Space Invaders that have combination actions, the maximum Q-value for a "move" action was taken (ie. for the "move left" Q-value in a frame in Enduro, we take the max between "move left," "brake left," and "fire left"). To test for significance, we test the interaction term in the linear model Action Value ~C(DQN Action Value) + C(Human Action) + C(DQN Action Value):C(Human Action) with Statsmodels.

For decoding human actions, we model human actions with the hidden layers of DQN using LASSO logistic regression (L1 regularization) using scikit learn functions and custom python code. Each hidden layer was projected to a dimensionality of 100 using PCA, giving a concatenated feature set of 400. Time points were downsampled by a factor of 10 to ease computation. The PCA transformation matrices were estimated using the frames for Sub001. These transformation matrices were used in every participant, to ensure that the PCs of every participant would be in the same space. LASSO logistic regression classifiers were then trained to predict left versus right actions, after frames where no action or other actions occurred were removed. The time points when other actions were selected in combination with left versus right were also included. Decoding accuracy was determined by cross-validating across runs. Optimal regularization parameters were found through grid search and were fixed across participants per game. Decoding accuracies were tested against a null distribution created from permutation tests of 1000 permutations. To maintain the autocorrelation of action trajectories, the cross validated data was shuffled in blocks of 40 time points (Wen et al., 2018). The predicted responses from the model were then compared against these shuffled datasets. The accuracy of every model in every participant exceeded the accuracy of the maximum value in the null distributions. To determine which layers were most useful for decoding actions, the model was trained on all runs (no cross-validation) and coefficients were absolute valued and averaged by layer.

Encoding model

To map hidden representations in DQN to voxels in the brain, we performed deep learning based encoding model analyses (Güçlü and van Gerven, 2015). All analyses were run in custom python code using functions from PyMVPA (Hanke et al., 2009) and scikit learn. First, image frames from the participant's gameplay data were run through the trained DQNs in order to generate neural network activations in every layer at every time point. As done in the decoding human actions analysis, PCA is used to reduce the dimensionality to 400 (100 PCs per layer). To downsample from the video game framerate to the TR of 1 Hz, each feature's values are averaged over a second. Then, copied time courses are shifted by both 5 s and 6 s to account for the hemodynamic delay of the fMRI signal. These two shifted time courses are concatenated into a feature set of 800. Next, voxelwise ridge regression (L2 regularization) is performed to predict each voxel's responses as a linear combination of this feature set. Optimal regularization parameters were found using grid search. Voxels are preprocessed as described above without spatial smoothing. Each voxel's response is z-scored to ensure every voxel is on the same scale. Accuracy is estimated using cross-validation across runs and calculating the Pearson correlation between predicted and actual time courses.

Statistical significance was quantified through permutation tests (since fMRI data may not be normally distributed) methods similar to previous approaches where 100,000 permutation tests are performed on 14 random voxels (Eickenberg et al., 2017). In each permutation, the time course of the held-out validation set was shuffled in a blockwise manner of blocks of 40 TRs to keep autocorrelation intact (Wen et al., 2018). The Pearson correlation between the shuffled time course and the predicted responses from the model were then computed. These permuted distributions are then concatenated, and voxel accuracy scores are compared to this concatenated null distribution to obtain one-sided p values for every voxel. Rather than selecting 14 completely random voxels to estimate a global null hypothesis for all brain voxels, we took a more conservative approach and selected 14 random voxels who were in the 90th percentile or above of scores in the encoding model analysis. This condition ensured that voxels with strong signal were selected. Voxels were then multiple comparisons corrected using FDR and plotted at the corrected threshold as indicated. Maps are transformed to standard space and spatially smoothed (5mm kernel) for visualization. To estimate layer selectivity, the coefficients from the models were absolute valued, averaged across layer, and then averaged across region. Average coefficients across participants are shown in Figure S3B.

Regions of interest and atlases

To define regions of interests for visualization and further analyses, we used the Harvard-Oxford Atlas. To distinguish V1, V2, V3, and V4 in the visual cortex, we used the Juelich Histological Atlas. Both atlases were accessed with FSLview. The early visual ROI consists of V1-V4; PPC includes LOC superior, superior parietal lobule, supramarginal gyrus, precuneus; Motor/Frontal includes motor and premotor cortex, SMA, and superior frontal gyrus.

Encoding model control analyses

Various control models were tested in the encoding model to help identify what computational principles play a role in the DQN model explaining neural responses. In an identical pipeline as the DQN encoding model analysis, these control feature sets were downsampled and time shifted by 5 s and 6 s (other than motor regressors where this preprocessing has already taken place) before cross-validated ridge regression was performed to compute prediction accuracies for every voxel.

Motor

Two motor regressors corresponding to making responses with the left and right hands were used. These regressors were taken directly from the GLMs in SPM for action value that are described below.

PCA

To construct a control model for basic visual features that represented the statistical structure of the images, the 84x84x4 pixel tensor was linearly projected to dimensionality 100 with principal component analysis using scikit learn. Although the DQN encoding model includes 400 features and we match this dimensionality with the cross game DQN and VAE control models, using 100 principal components outperformed using 400. This linear projection of the input uncovers features that explain the low-level statistical structure in the input that vary the most during gameplay without any representation of reward. Similar approaches have been used to explain neural responses to a remarkable degree throughout the visual pathway (Chang and Tsao, 2017; Olshausen and Field, 1996). Additionally, since we perform PCA on the tensor of 4 consecutive frames that are input into DQN, the principal components uncover statistical properties of motion and change detection that are appropriate to model the dorsal visual pathway. As with the other PCA analyses, transformation matrices were estimated using sub001's data and used across participants to project every all data to the same space.

These principal components were also used to estimate their representation of the nuisance variables. Scikit learn's 'mutual_info_classif' function was used to calculate the mutual information between the first principal component and the nuisance variables.

Cross game DQN

We also compared our encoding model results with a DQN trained on a different game. The Space Invaders network was used as this control for Enduro, Enduro for Pong, and Pong for Space Invaders. Other than shifting the networks, the regressors were constructed identically to the original encoding model.

VAE

To compare DQN with another state of the art method for state representation learning using a deep neural network (Higgins et al., 2018b; Mohamed and Rezende, 2015; Watter et al., 2015), we trained variational autoencoders in Tensorflow for each game by modifying an existing template (https://github.com/tensorflow/docs/blob/master/site/en/tutorials/generative/cvae.ipynb). The architecture we used was designed to be as similar as possible to DQN.

This consisted of an encoder of three convolutional layers (Layer 1: 32 filters of 8x8 with stride of 4; Layer 2: 64 filters of 4x4 with stride of 2; Layer 3: 64 filters of 3x3 with stride of 1), followed by a fully connected layer to output the set of mean and log-variance parameters for the latent representation of dimensionality 400. The decoder architecture consisted of a fully connected layer followed by four convolution transpose layers (Layer 1: 64 filters of 4x4 with stride of 1; Layer 2: 64 filters of 4x4 with stride of 2; Layer 3: 32 filters of 8x8 with stride of 2; Layer 4: 1 filter of 8x8 with stride of 1). All activation functions are rectified linear units (ReLU). The network was trained on each game separately by maximizing the evidence lower bound (ELBO) on the marginal log-likelihood of the training data. Data frames of the first 8 runs of the first participant were used as the training set, and frames from the last 3 runs were used as the test set for tracking generalization (training sets and test sets were downsampled by 5 to ease computation). Training included 1000 epochs over the entire training set, but converged well before that for every game (training loss for first 500 epochs plotted in Figure 4A). After training, performance on the test set was nearly equivalent to performance on training set.

The human frames from every participant were then run through the trained encoder to map them to the latent distribution, which outputs 400 means and log-variances for the latent dimensions. The means were then used as a 400 dimensional stimulus feature set for the control encoding model, and preprocessed with downsampling and time lags identically to the other feature sets used for encoding models.

General linear model (GLM) control analysis

In order to test for whether brain responses could still be predicted by DQN when controlling for the other models and game events, we constructed GLMs in SPM12 similarly to previous approaches (Iigaya et al., 2020). The first 10 principal components for each DQN layer, VAE, and PCA models were added as parametric modulators to the same onset at the temporal resolution of the 1 Hz TR after averaging across volumes. Orthogonalization was turned off. Other regressors of no interest included all of the regressors described in the computational model-based GLM section below, including regressors for motor responses, reward/punishment, and action values. To quantify a voxel's correlation to the unique variance in each of the six models (four DQN layers, VAE, PCA) F-tests were computed on the betas for the 10 PCs in each model, which tests whether a voxel is significantly modulated by at least one principal component in a model. The percentage of significant voxels in a region of interest for each model is reported in Figure 4C.

Control region analysis

To rule out the possibility that our analyses are picking up on artifacts such as head motion that affect the entire properties of the fMRI images, we completed a control region analysis with one subject (sub001). A control region was represented by two spheres of air drawn directly in front of the brain. The encoding model was then run on every voxel within those spheres and the distribution of prediction accuracies were plotted alongside comparison ROIs (V1 and superior parietal lobule) in Figure S5B. No voxels in these spheres had significant prediction accuracies and the whole distribution of scores were very close to zero.

Representational similarity analysis

We performed representational similarity analyses (RSA) to examine how the representations transform throughout the DQN layers. Dissimilarity matrices (DSMs) were constructed at the frame level for DQN layers 1-4, VAE, PCA, the pixel space, and the hand drawn features for Pong. Each model was first downsampled by 20 and data was concatenated across runs within subject. DSMs were constructed by computing pairwise comparisons across frames for each model with pyMVPA. Within day comparisons were removed to avoid potential confounds due to similarity being driven by patterns being in the same run or day. For the pixel space, the 84x84x4 tensor of images that are fed to DQN were reshaped into a 28224 dimensional response vector. For the PCA model, weights fit to the data of sub001 were again used to transform the pixel space into a 100 dimensional space. In Pong, each hand drawn feature (the positions of the two paddles, the ball position X and Y, and the ball's velocity X and Y) was z-scored and input into one response vector. Euclidean distance was used as the distance metric for the Pong hand drawn features, and correlation distance was used for every other model. Every DSM was rank-ordered to compare model DSMs without assuming a linear relationship between models. Models were then compared with Spearman correlations (the Pearson correlation on the rank-ordered DSMs).

For comparing model DSMs and fMRI DSMs in Pong, each DSM was created at the TR level. This involved using the same feature sets that were used in the encoding model, where responses were averaged across volumes to downsample to TR resolution (1 Hz) and shifted by 6 s to account for hemodynamic delay. Again, correlation distance was used for every model except the hand drawn features (Euclidean) and DSMs were rank-ordered.

For fMRI data, DSMs for three brain areas were constructed, early visual, posterior parietal cortex (PPC), and motor/frontal. Early visual regions included all visual cortex ROIs. PPC included superior lateral occipital cortex, superior parietal lobule, supramarginal gyrus, and precuneus. Motor/frontal included motor and premotor cortex, SMA, and superior frontal gyrus.

To test for significance we performed block permutation tests for every model, since the data may not be normally distributed. Similarly to the encoding model permutation tests, fMRI data volumes were shuffled blockwise in blocks of 40 TRs to keep autocorrelation intact (Wen et al., 2018), then DSMs were reconstructed and correlated to the non shuffled model DSMs. Then, to test if the correlation in a model was significantly different than zero, the correlation score had to be greater than the maximum correlation in the permutation test distribution (one-sided). To test if the differences between models were significant, this difference was tested (two-sided) against a distribution based on computing the differences between the models in every permutation. All scores were corrected for multiple comparisons.

Computational model-based GLMs

To localize the neural correlates of action value computations, we conducted computational model-based generalized linear model (GLM) analyses (O'Doherty et al., 2007). This novel analysis differs from previous approaches in two ways: a deep neural network is used to approximate the value function that is used to construct regressors, and the model is trained independently of any human behavioral data.

All univariate GLMs were conducted using SPM12 software. Initially the image frames from the human gameplay data were run through the trained DQN to output Q-values at every frame in a run as described above. Next, the Q-values were decomposed into action advantages/values to separate action related variance from reward related variance. Taking inspiration from actor critic approaches to isolate action advantages (Sutton and Barto, 2018), we define state value (V(s), s = state) as the average of all Q-values, and action advantages (A(s,a), s = state, a = action) as the difference between an actions Q-value and the state value.

$$A(s,a) = Q(s,a) - V(s)$$

$$V(s) = \frac{1}{|A|}\sum_{a'} Q(s, a')$$

Similar to the analysis in a previous study (Wunderlich et al., 2009), the action value regressor here is computed as the chosen value (the maximum) between move left value and move right value. Chosen value is then LOWESS smoothed across frames, and downsampled to 10 Hz for every volume (TR = 1 s). The regressor is then z-scored and entered into a GLM where it is convolved with a hemodynamic response function. Across the games, other covariates included left and right hand motor responses, parametric regressors for both positive reward and negative reward, game presentation (8 minutes of gameplay per run with one minute of rest before and after), run, and day. Losing a life was included as the negative reward regressor in Space Invaders, although the Atari engine does not explicitly deliver negative reward for loss of life, and the negative ramifications are reflected in the opportunity cost of gaining more points. Additional regressors for Space Invaders also included fire action value and the number of invaders left on the screen. For Enduro, the action value for the brake action was also included (which simultaneously approximates the anti-correlated fire action value, thus the fire action value was not also included).

Filter analyses

To further interpret the encoding model results, we wanted to identify which filters were useful for modeling neural responses, and whether this varied between regions of interest. To do this, we retrained the encoding model on each filter in layer 3 (the last convolutional layer) on each voxel that was significant in the encoding model analyses. This layer had 64 filters of 7x7 receptive field size. We use cross-validated prediction accuracy of a voxel response using a convolutional filter's explanatory features to quantify that filter's Neural Predictivity. This Neural Predictivity score was averaged across a region to estimate how well that filter predicted responses in a region. With 64 Neural Predictivity scores per region, correlations of these scores across regions were computed to evaluate the variability of filter selectivity between regions and to construct a similarity matrix (Figure 7B). This similarity matrix reflects the average similarity matrix across the six participants. A similar procedure was used to compute correlations of Neural Predictivity across participants, where in this case filter scores were averaged across all voxels in all ROIs in a participant rather than by region (Figure 7C). For computing a filter score for decoding human actions (Figure 7D), we similarly retrain the model from ''Decoding Human Actions'' on each filter separately in layer 3. These filter scores were then rescaled with min-max normalization for subsequent correlation analyses and visualizations. Thus, the best filter has a score of 1 and the worst filter has a score of 0.

To visualize the features encoded by the filters (Figure 7E), we use Neon's deconvolution visualization function and modified code from (https://github.com/tambetm/simple_dqn). This procedure finds frames from the human gameplay data that activate a filter the most (which is depicted on the right side), then uses guided backpropagation to identify parts of the image that led to this activation (left side). The colors reflect changes and motion across the image tensor of three frames, meaning the filter detects motion in this location.

We annotate six high-level features in Pong using custom python code that localizes the corresponding objects in the pixel space: ball X position, ball Y position, ball X velocity, ball Y velocity, left paddle position, and right paddle position. To assess how much each filter encodes each feature, we use scikit learn's 'mutual_info_regression' function to calculate the mutual information between a filter and these continuous variables. The mutual information scores were averaged across ball X and Y positions to get one score for ball position. We similarly averaged across the ball X and Y velocity and the left and right paddle position to get scores for ball velocity and the paddle positions respectively. This outputs a MI score for each of the 7x7 receptive fields in a filter, which were then averaged to get one metric per filter for each high-level feature. These metrics are then correlated with each filter's Neural Predictivity across the whole brain in Pong (Figure 7A).

Nuisance invariance scores

We completed additional analyses to identify how the regions of interest encode sensory information that is irrelevant for task performance. To uncover this, we utilized a concept from the machine-learning sub-field of representation learning: nuisance invariance (Lenc and Vedaldi, 2015). A nuisance variable is any variable in the input that is irrelevant to the task, and is mathematically defined as any variable where the mutual information between it and the task output is zero (I(y;n) = 0), where y is a task label and n is a nuisance variable). Common examples include translation and illumination invariance in object recognition, as the location of an object on an image and the overall brightness of a picture are usually unrelated to classifying it correctly. Thus, nuisance invariance in neural networks suggests that a compressed and abstract representation has been learned.

The game Enduro has a unique feature that we leveraged to study nuisance invariance in the gameplay environment. The colors on the screen constantly change as the weather and time of day in the game frequently changes. These stages include sunny, snowy, foggy, dusk, and night-time. Therefore, the pixel space changes dramatically while the overall gameplay dynamics are stable. In fact, we calculated that the mutual information between human left and right actions and the weather/time of day variable equals zero using scikit-learns 'mutual_info_classif' function (I(time of day; actions) = 0), which indicates that weather/time of day is a nuisance variable. This metric can only be zero if and only if two random variables are independent. To put this in perspective, the mutual information between weather/time of day and the first principal component of the pixel space is 1.70 (I(time of day; PC 1) = 1.70), and the mutual information of weather/time of day with itself is 1.81. This shows that a large amount of variance in the pixel space is due to these changing weather patterns as the first principal component codes for these conditions.

Although there was no factor that is as obviously a nuisance variable in Space Invaders as the changing colors on the screen was in Enduro, the total number of invader ships on the screen explains a lot of variance in the visual pixel space, and has a high mutual information with the first principal component of the pixel space (I(num. invaders; PC 1) = 1.52). However, in this game the relative

Please cite this article in press as: Cross et al., Using deep reinforcement learning to reveal how the brain encodes abstract state-space representations in high-dimensional environments, Neuron (2020), https://doi.org/10.1016/j.neuron.2020.11.021

positions of the invaders above an agent matter more than their absolute position and the global features, as the invaders above an agent will be in the agent's line of fire and the agent will be in the invader's line of fire. One exception is when there is one invader left and it starts to speed up faster than usual. To quantify this pattern, we calculated that the mutual information between the number of invaders on the screen and left and right actions is relatively low (I(num. invaders; action) = 0.07).

To compute a nuisance invariance score for each filter, we again use scikit learn's 'mutual_info_classif' function to calculate the mutual information between a filter and a nuisance variable (weather/time of day for Enduro, number of invaders on the screen for Space Invaders - calculated with downsampled data from sub001 to ease computation).This function outputs a MI score for each of the 7x7 receptive fields, thus these scores were averaged to get a single score per filter. This score was multiplied by -1 to get the inverse of this MI metric, to denote insensitivity of the nuisance rather than encoding of the nuisance. Next, the 64 filter nuisance invariance scores are Pearson correlated with the 64 Neural Predictivity scores in a region. Intuitively, this analysis estimates whether a region prefers filters that are more insensitive to the nuisances (positive correlation) or filters that code for the nuisance (negative correlation). To increase interpretability and enhance the variability across regions that we are most interested in assessing, we z-score this metric across voxels in a participant. Thus, a nuisance invariance score of 0 is average with respect to the other voxels in a participant and the magnitude of the score reflects how many standard deviations it is from the mean.