nature COMMUNICATIONS

ARTICLE

https://doi.org/10.1038/s41467-021-21696-1 OPEN

Check for updates

Predictive learning as a network mechanism for extracting low-dimensional latent space representations

Stefano Recanatesi[1✉], Matthew Farrell[2], Guillaume Lajoie[3,4], Sophie Deneve[5], Mattia Rigotti[6,8] & Eric Shea-Brown[1,2,7,8]

Artificial neural networks have recently achieved many successes in solving sequential processing and planning tasks. Their success is often ascribed to the emergence of the task's low-dimensional latent structure in the network activity – i.e., in the learned neural representations. Here, we investigate the hypothesis that a means for generating representations with easily accessed low-dimensional latent structure, possibly reflecting an underlying semantic organization, is through learning to predict observations about the world. Specifically, we ask whether and when network mechanisms for sensory prediction coincide with those for extracting the underlying latent variables. Using a recurrent neural network model trained to predict a sequence of observations we show that network dynamics exhibit low-dimensional but nonlinearly transformed representations of sensory inputs that map the latent structure of the sensory environment. We quantify these results using nonlinear measures of intrinsic dimensionality and linear decodability of latent variables, and provide mathematical arguments for why such useful predictive representations emerge. We focus throughout on how our results can aid the analysis and interpretation of experimental data.

[1] University of Washington Center for Computational Neuroscience and Swartz Center for Theoretical Neuroscience, Seattle, WA, USA. [2] Department of Applied Mathematics, University of Washington, Seattle, WA, USA. [3] Department of Mathematics and Statistics, Université de Montréal, Montreal, QC, Canada. [4] Mila-Quebec Artificial Intelligence Institute, Montreal, QC, Canada. [5] Group for Neural Theory, Ecole Normal Superieur, Paris, France. [6] IBM Research AI, Yorktown Heights, NY, USA. [7] Allen Institute for Brain Science, Seattle, WA, USA. [8]These authors contributed equally: Mattia Rigotti, Eric Shea-Brown. ✉email: stefanor@uw.edu

Neural network representations are often described as encoding latent information from a corpus of data[1–7]. Similarly, the brain forms representations to help it overcome a formidable challenge: to organize episodes, tasks, and behavior according to a priori unkown latent variables underlying the experienced sensory information. In this paper, motivated by the literature suggesting that these efficient representations are instrumental for the brain's ability to solve a variety of tasks[8–10], we ask: How does such an organization of information emerge?

In the context of artificial neural networks, two related bodies of work have shown that this can occur due to the process of prediction—giving rise to predictive representations. First, neural networks are able to extract latent semantic characteristics from linguistic corpora when trained to predict the context in which a given word appears[11–14]. The resulting neural representations of words (known as word embeddings) have emergent geometric properties that reflect the semantic meaning of the words they represent[15]. Second, models learning to encode for future sensory information give rise to internal representations that encode task-related maps useful for goal-directed behavior[9,16–18].

As predictive mechanisms have been conjectured to be implemented across distinct neural circuits[19–21], characterizing predictive representations can then shed light on where and how the brain exploits such mechanisms to organize sensory information. Our goal is to build theoretical and data-analytic tools that explain why a predictive learning process leads to low-dimensional maps of the latent structure of the underlying tasks —and what the general features of such maps in neural recordings might be. This links predictive learning in neural networks with existing mechanisms of extracting latent structure[22–24] and low-dimensional representations from data[25].

We begin with an introductory example of how predictive learning enables the extraction of latent variables characterizing the regularity of transitions among a set of discrete "states", each of which generates a different observation about the world. Then we focus on a model where observations are generated from continuous latent variables embedded in a low-dimensional manifold. We focus on the special case of spatial exploration, in which the latent variables are the position and orientation of an agent in the spatial environment, and the observations are high-dimensional sensory inputs specific to a given position and orientation. The predictive learning task we study is to predict future observations. Our central question is whether a recurrent neural network (RNN) trained on this predictive learning task will extract representations of the underlying low-dimensional latent variables.

We develop analytical tools to reveal the low-dimensional structure of representations created by predictive learning. Crucial to this is the distinction between linear[26–30] and nonlinear dimensionality[31,32], which allows us to uncover what we call latent space signal transfer, wherein latent variables become increasingly linearly decodable from the top principal components of the neural representation as learning progresses. Latent space signal transfer is accompanied by clear trends in the linear and nonlinear dimensionality of the underlying representation manifold, and potentially gives rise to the the formation of neurons with localized activations on the nonlinear manifold, manifold cells[33]. Importantly, while each of these phenomena could separately find its origin in a mechanism different from predictive learning, they altogether provide a strong measurable feature of predictive learning that expect further testing in both neural and machine-learning experiments. We conclude by extending our framework to the analysis of both neural data and a second task—arm-reaching movements.

Results

Predictive learning and latent representations: a simple example. In predictive learning a neural network learns to minimize the errors between its output at the present time and a stream of future observations. This is a predictive framework in the temporal domain, where the prediction is along the time axis[20]. At each time t an agent observes the state of a system o_t and takes an action a_t out of a set of possible actions. The agent is prompted to learn that, given (o_t, a_t), it will next observe o_{t+1}.

We begin by illustrating our core idea— that predictive learning leads neural networks to represent the latent spaces underlying their inputs—in a simple setting. We study the task shown in Fig. 1a, where the state of the system is in one of $N_s =$ 25 states. To each state is associated a unique set of five random cards that the agent observes whenever it is in that state. The states are organized on a two-dimensional lattice—the latent space. Observations have no dependence on the lattice structure, as they are randomly assigned to each state with statistics that are completely independent from one state to the next. On the other hand, actions are defined on the lattice: at each time t the agent either randomly moves to one out of the four neighboring states by selecting the corresponding action or remains in the same state. Movements, when they occur are thus along the four cardinal directions N, S, W, E used to indicate the corresponding action. Meanwhile, 0 denotes the action corresponding to no movement, for a total of $N_a = 5$ possible actions.

The agent solves this predictive task when, prompted with a pair (o_t, a_t), it correctly predicts the upcoming observation o_{t+1}. A priori, this task does not require the agent to extract information about the underlying lattice structure of the state space. Indeed the agent could solve the task with at least two possible strategies: (1) by associating with each observation (set of cards) the next observation via a collection of $N_s \times N_a$ distinct relationships $(o_t, a_t) \mapsto o_{t+1}$ (combinatorial solution), or (2) via a simple set of relationships that exploit the underlying lattice structure of the state space. In this second scenario the agent would uncover the lattice structure while using it to map actions to predictions. This solution thus presupposes an internal representation of the latent space and we refer to it as predictive representation solution. The critical difference between the combinatorial and predictive representation solutions is that the latter extracts a representation of the latent space while the former doesnot, cfr. Fig. 1b.

We train a simple two-layer network on this card-game task: to predict the future observation given inputs of the current observation and action, Fig. 1c. We focus on the first layer that receives the joint input of actions and observations. In this example observations are encoded with a one-hot representation, formally turning the problem into a classification task. Upon learning, by means of Stochastic Gradient Descent (SGD), the network develops an internal representation in the hidden layer for each of the 125 input pairs (o_t, a_t).

Visualizing these internal representations in the space of principal components of neural activations, the underlying latent structure of the state space appears (Fig. 1d.) This lattice-like structure is a joint representation of observations and actions. This representation emerges over the course of learning: initially, the representation of each observation-action pair (o_t, a_t) does not reflect the underlying latent space, see Fig. 1d. The development of the latent space representation can be clearly visualized across stages of the learning process (see Fig. S1).

Additionally, if we remove the actions from the input to the network but still training it to perform prediction, the network still learns a representation that partially reflects the latent space, Fig. 1e, though this time it is distorted (cf. Fig. S2).

Below, we will demonstrate this phenomenon in other more complex settings, but we first pause to build intuition for why it

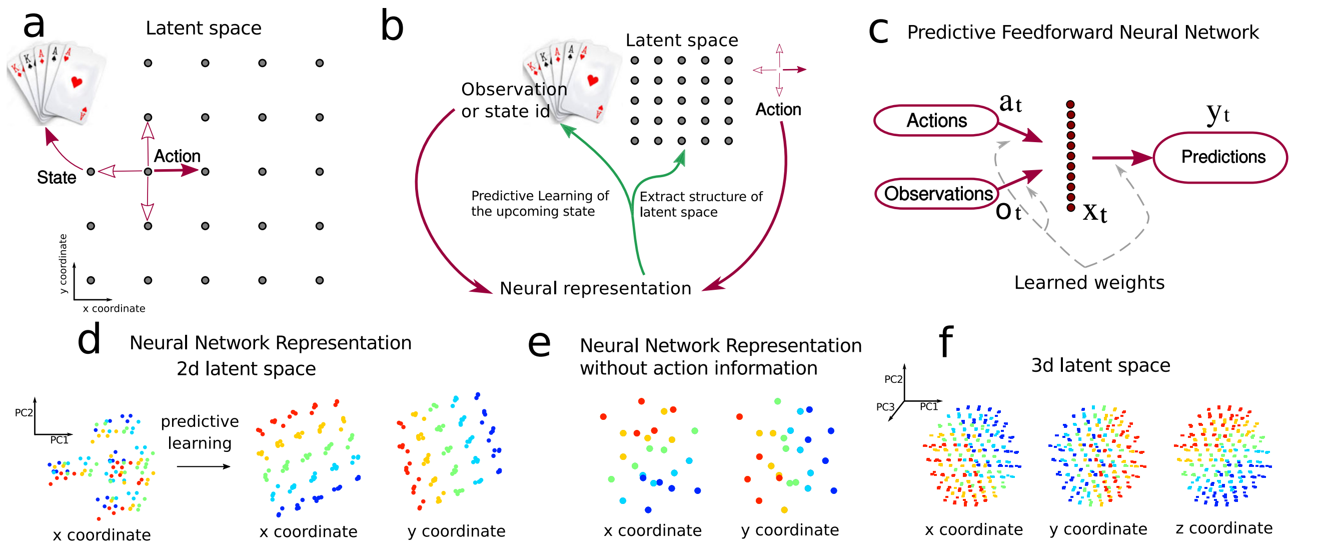

Fig. 1 Predictive network solving a card-game task. a Description of the latent space underlying the task. **b** Illustration of the task and information flow diagram: the neural representation receives state observations and actions and extracts the latent space structure by means of predicting upcoming observations. **c** Diagram of the network's structure. The diagram highlights the layer studied here, although the network has a two layers, where the second layer serves as a decoder. **d** The network's neural representation: activity in the hidden layer plotted vs. principal components PCs 1 and 2 of hidden layer activity. For each observation-action pair (o_t, a_t), the corresponding activation is colored by the position of the state that the network predicts: x-coordinate (left plot, before and after learning) and y-coordinate (right plot). **e** Same as panel **d** in the absence of the action as a input to the network. **f** Same as panel **d** for a three-dimensional latent space.

occurs within neural networks. We start by noticing that upon learning the five actions $a \in \{N, S, W, E, \theta\}$ are mapped to a fixed vector $\boldsymbol{w}_a$, which is added to the state representation $\boldsymbol{w}_s$ every time the corresponding action is selected:

$$\boldsymbol{x}_{s,a} = \tanh(\boldsymbol{w}_s + \boldsymbol{w}_a + \boldsymbol{b}) \,, \tag{1}$$

where $\boldsymbol{b}$ is a learned bias parameter. Specifically, consider the representation $\boldsymbol{x}$ in the network for predicting a state s' located immediately above (to the N) of the state s, in two scenarios. In the first, s' is arrived at from s, after the action $a = N$. This gives the representation

$$\boldsymbol{x}_{s,N} = \tanh(\boldsymbol{w}_s + \boldsymbol{w}_N + \boldsymbol{b}) \tag{2}$$

In the second, s' is arrived at from s', after the null action:

$$\boldsymbol{x}_{s',0} = \tanh(\boldsymbol{w}' + \boldsymbol{w}_0 + \boldsymbol{b}) \tag{3}$$

Both of these activations must be read out to return the same prediction: s'. While this could occur in principle if the readout operation learned to collapse different representations to the same readout, the network learns a simpler solution in which the representations (Eq. (2)) and (Eq. (3)) are equal (cf.[34,35]), so that $\boldsymbol{x}_{s',0} = \boldsymbol{x}_{s,\mathrm{N}}$ implies:

$$\boldsymbol{w}_{s'} - \boldsymbol{w}_s = \boldsymbol{w}_{\mathrm{N}} - \boldsymbol{w}_0 \tag{4}$$

for any pair of states s, s' linked by the action $a = N$. This implies that (up to the hyperbolic tangent non-linearity), the representation of the states is acted upon by the action in a translational invariant way in the direction of the action $\boldsymbol{w_N} - \boldsymbol{w}_0$. This is true for any of the actions N, S, W, E, and for any starting state s. Thus, the representation inherits an approximate translation invariance—the characteristic property of a lattice structure. This invariance confers a geometrical structure upon the learned neural representation that reflects the latent space. This phenomenon directly generalizes to lattices of higher dimension, as shown in Fig. 1f.

We note that this analysis holds precisely when the learning process enforces representations of the same decoded state to be nearly identical—which occurs in all of our simulations and is predicted by other numerical and theoretical studies[34,35]—and holds approximately when it tends to cluster these together. By contrast, in a general combinatorial solution of Eq. (4) each observation action pair could be linked to the upcoming state independently, $\boldsymbol{x}_{s,N} \neq \boldsymbol{x}_{s',0}$.

We can apply related ideas to begin to understand more challenging case in which the prediction task is performed without knowledge of the action, so that only observations are passed as input to the network. As we showed in figure (Fig. 1e) above, in this case the internal representation still partially reflects the latent space. This is not because the set of observations as a whole carries any information about the latent space, but because the effect of the actions—to bind nearby states together—is reflected in the statistics of the sequence of observations. Thus, through making predictions about future observations, the network still learns to bind states that occur nearby in time together, extracting the latent space (cf. Suppl. Mat. Sec. 2.4).

We next generalize the predictive learning framework to two different, more complex benchmark tasks of neuroscientific interest: spatial exploration and arm-reaching movements.

Predictive learning extracts latent space representations in a spatial exploration task. We focus on predictive learning in a spatial exploration task in order to generalize the previous example to show how predictive learning extracts the low-dimensional latent structure from a high-dimensional sensory stream (Fig. 2a) and to introduce novel metrics, which quantify such process.

In the spatial exploration task an agent traverses a square open arena. Traversing the environment, the actions taken determine a trajectory in three spaces: the latent space, which defines the agent's (or animal's) state in the environment, the observation space of the agent's sensory experience, and the neural activation space of its neural representation. We introduce the task defining these three spaces.

The latent space, similarly to the card-game example, is the set of spatial coordinates that identifies the agent's state, (x, y, θ),

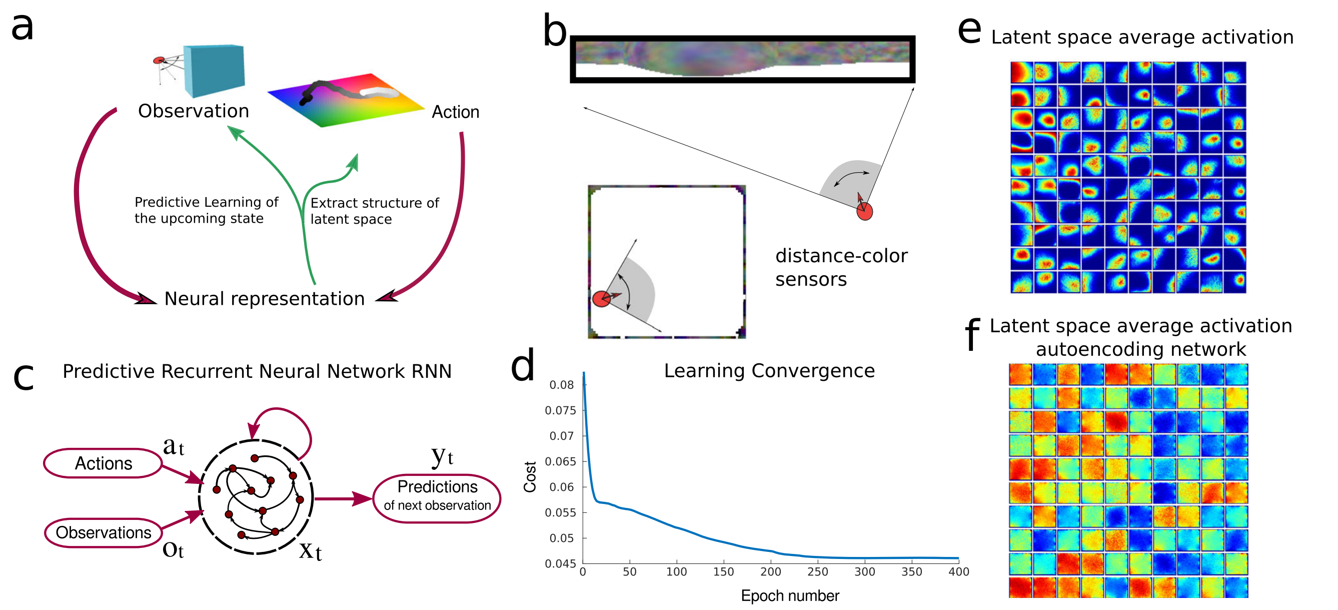

Fig. 2 Predictive network solving an exploration task. a Information flow diagram of the task: an agent explores a two-dimensional environment (latent space) through actions and receives observations regarding it. The network's task is to predict the next sensory observation. By learning to do so it recovers information regarding the underlying hidden latent space. **b** Illustration of the agent with sensors in the square environment where the walls have been colored (cfr. Methods). The sensors span a 90° degree angle and register the color and distance of the wall along their respective directions. **c** Diagram of the predictive recurrent neural network: the network receives actions and observations as inputs and is trained to output the upcoming sensory observation. **d** Cost during training for the network (cf. Methods). **e** Average activity of 100 neurons (each of the 100 neurons average activity is showed in one of the small 100 quadrants) against the *x*, *y* coordinates of the environment, showing place-related activity. **f** Same as panel **e** for a RNN trained to autoencode its input observations.

where x and y are position and θ is its direction. The observation space is defined in terms of the agent's ability to sense the surrounding environment. To model this we consider the case where the agent senses both visual and distal information from the environment's walls—the agent is equipped with sensors that span a 90° visual cone centered on its current direction θ reporting distance and color of the environment's wall along their directions, Fig. 2b. The environment the agent navigates is a discrete grid of 64×64 locations. Each wall tile, one at each wall location, is first colored randomly and then a narrow spatial autocorrelation is applied, see Fig. 2b. The number of sensors N_s is chosen so that observations across sensors are independent $N_s = 5$.

We consider the case where the agent's actions are correlated in time but do not depend on the observations—random exploration. At each step the agent's direction θ is updated by a small random angle $d\theta$ drawn from a Gaussian distribution centered at zero and with a variance of 30°. The agent then moves to the discrete grid location most aligned with the updated direction $\theta + d\theta$ (unless it is occupied by a wall; cfr. Methods for details). Actions are performed by the agent with respect to its allocentric framework, so that there are nine possible choices: for each location there are eight neighboring ones plus the possibility of remaining in the same location. While the agent moves in the environment it collects a stream of observations.

In predictive learning, the agent learns to predict the upcoming sensory observation, Fig. 2c. It achieves this by minimizing the difference between its prediction y_t at time t and the upcoming observation o_{t+1}: $C = \Sigma_t ||y_t - o_{t+1}||^2$, Fig. 2d. We refer to the activations of the units of the trained RNN as its internal predictive representation. The RNN can be thought as a model of the agent's brain area carrying out the task. As the agent learns to predict the next observation, its representation is influenced both by the observation space (since the task is defined purely in terms of observations) and by the latent space (since the actions are defined on it); a priori, it is not obvious which space's influence will be stronger. In this example, we used a more general recurrent network rather than the simplest-possible feedforward setup in the first example of Fig. 1; this allows information from the stream of sensory observations to be integrated over time, a feature especially important in more challenging settings when instantaneous sensory information may be only partially informative of the current state.

A first indication that, by the end of learning, neurons encode the latent space is given by the fact that individual neurons develop spatial tuning Fig. 2e. The neural representation has extracted information about the latent space from the observations, without any explicit prompt to do so. In the Suppl. Mat. (Figs. S7–S12), we show how this phenomenon is robust to alterations of the sensory observations and network architecture.

However, when the same network learns, based on the same input sequence, to reproduce the current observation (autoencoding framework corresponding to a cost $C = \Sigma_t ||y_t - o_t||^2$) rather than predict the upcoming one, individual neurons do not appear to develop spatial tuning, Fig. 2f and Suppl. Mat. (Figs. S10–11).

Metrics for predictive learning and latent representations. How—and to what extent—does the neural population as a whole represent the latent space? This question demands quantitative answers. To this end we develop novel methods for analyzing neural representation manifolds, and three metrics that capture the dynamical and geometrical properties of the representation manifold. These are predictive error, latent signal transfer and dimensionality gain. While the first of these is specific to predictive frameworks, the other two could be interpreted as general metrics to quantify the process of extraction of a low-dimensional latent space from data. Below we illustrate these metrics in the context of the spatial exploration task (cf. Figs. S3–5 for a detailed analysis and more examples of such metrics).

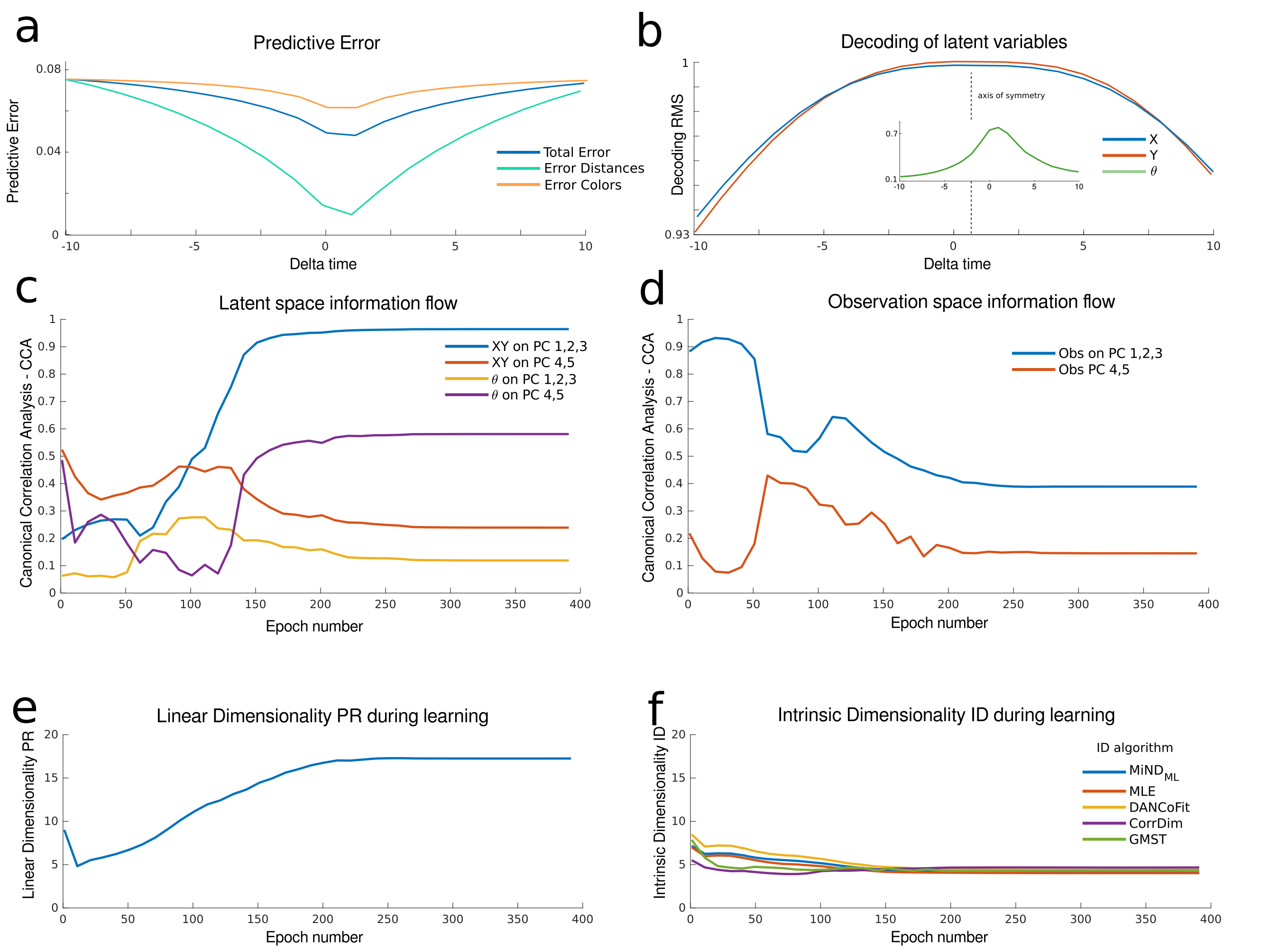

Fig. 3 Learning the predictive representation. **a** Predictive error (L_2 norm) in blue between the network's output and the observation as a function of the lag (Delta t). In red average L_2 norm between the observation at time 0 and at a lag Delta t. **b** Linear decoding of latent variables. RMS measure of the linear decoding of (x, y, θ) at time Delta t from the neural representation at time 0. The dotted line highlights the axis of symmetry of the curves. **c** Signal transfer analysis: Canonical Correlation Analysis between PCs of the neural representation and the latent space. The lines correspond to the average of the canonical correlations between the highlighted variables. **d** Same as panel **c** but for the observation space. **e** Participation ratio of the representation during learning. **f** Intrinsic dimensionality (ID) of the representation during learning. Five different intrinsic dimensionality estimators are used (cfr. Methods).

Predictive error. The network's task is to predict future observations. Owing to correlations in the sensory input itself from one timestep to the next, to verify that the network is actually making predictions we first ask whether the network's output is most similar to the upcoming observation rather than current or previous ones[36]. This can be captured by the absolute difference between the current output of the network and the stream of observations at any time, which we refer to as predictive error. If this is skewed towards the upcoming observation (see Fig. 3a blue line), it suggests that the network predicts elements of upcoming observations. This measure relies on knowledge of the network's output and of the stream of observations. An allied measure of this effect relies on the ability to decode past vs. future latent states from the current neural representation. If the decoding error is skewed for future (vs. past) latent states, this also suggests that the network predicts future states. Figure 3b shows that this is the case for the spatial exploration network: it codes for future latent variables as well as current and past ones, with the axis of symmetry for decoding the spatial coordinates x, y located close to the future value $\Delta t = 1$ (cf. Fig. S13 for a comparison with neural data). Similarly the axis of symmetry for the angle θ is located closer to $\Delta t = 1$, although in this case the analysis is confounded by the fact that actions carry partial information regarding θ.

Latent signal transfer. We next introduce a feature of predictive learning that tracks how the neural representation reflects the latent space over the course of learning. This quantifies the phenomenon visible by eye in the introductory example of Fig. 1d. To define the latent signal transfer metric, at each stage of learning we compute the average of the canonical correlation (CC) coefficients between the representation projected into its PCs, and latent space variables x, y, θ. The blue line in Fig. 3c shows the average of the CC coefficients between the representation in PCs 1 to 3 and the position x, y of the agent in latent space. When the average CC coefficient is 1, all the signal regarding x, y has been transferred onto PCs 1 to 3 in a linear fashion. A similar interpretation holds for the other curves: in sum, they track the formation of explicit representations of latent variables that are accessible via linear decoding .

Figure 3c shows that, between epoch 50 and 150, most of the information regarding the latent space moves onto the first few PC modes of the neural representation. The same analysis can be carried out with respect to observation space variables. This is shown in Fig. 3d, where the decreasing trend indicates that the observation space signal flows out of the first few PC components as learning progresses. Altogether Fig. 3c, d show that the representation, as interpreted through PC components, encodes more information about the latent space as opposed to the observation space as learning progresses (blue and red lines).

Dimensionality gain. Finally, motivated by the fact that the latent spaces of interest are lower-dimensional, we introduce metrics that allow us to quantify the extent to which the learned neural representations have a similar dimension.

We begin by noting that the latent signal transfer analysis (Fig. 3c, d) suggests that predictive learning might have formed a low-D neural representation. However, when we measure the dimensionality of the neural representation with a linear dimensionality metric, the participation ratio (PR), we observe that dimensionality actually increases over the course of learning Fig. 3e. Instead, measuring the dimensionality of the neural representaion with nonlinear techniques sensitive to the local curvature of the representation manifold—yielding the intrinsic dimensionality (ID)—shows that the dimensionality rather than increasing at most decreases through learning.

This dichotomy can be interpreted by means of two different demands that shape network representations. On one hand, the representation is prompted to encode high-dimensional observations; on the other, it extracts the regularity of a low-dimensional latent space. While the high dimensionality of the observations is a global property, referring to the collection of many observations, the regularity of the latent space is induced on a local scale, as neural representations relate to their possible neighbors via the action. These demands lead the linear dimensionality PR, measuring a global property of the representation manifold, and the nonlinear dimensionality ID, measuring more local properties, to have opposite trends. This interpretation is supported by further experiments and the next example we study, that arm-reaching movements, in which the network is prompted to predict a lower-dimensional observation signal. To encapsulate this phenomenon we suggest the metric of dimensionality gain (DG), which is the ratio between the linear global dimensionality and the nonlinear local dimensionality of the representation manifold. Higher values of DG thus capture the network's ability to extract a low-dimensional representation of a high-dimensional stream of observations. In the example of Fig. 3e, f, DG ≈ 3.5 upon learning.

The role of prediction in extracting latent representations. To show how the three metrics just described characterize predictive learning, we compare representations learned in the same networks but without the demand for prediction (as in Fig. 2f). In Fig. 4 we show how predictive error, latent signal transfer and dimensionality properties of the network differ in these two cases. The comparison is carried out by training 50 different networks of smaller size (100 neurons) on either the predictive task or a non-predictive version in which the network outputs observations received on the current timestep. In sum, comparing each of the metrics introduced above for the predictive vs. non-protective cases shows that, while predictive learning extracts a low-dimensional manifold encoding for the latent variables, non-predictive learning in these networks does not.

One point here bears further discussion. While Fig. 4e shows that ID is lower in the predictive vs. non-predictive case, this may seem surprising because there are grounds to expect that ID would be equal in these cases. These grounds are that the observations are produced as a map from a low-dimensional latent space in both cases, so that if the network directly encodes them, it should admit a similar low-dimensional parametrization and hence similar ID in both cases as well. The resolution comes from the fact that ID, despite being a local measure, is based on statistical properties of points sampled from a manifold (cf. "Methods"). So if the manifold appears higher dimensional, despite having a parametrization, which is low-dimensional, then ID would point to a higher dimension. In other terms ID is sensitive to the manifold's smoothness and can be taken as a measure of it for manifolds parameterized by a fixed number of variables. This problem is known in the literature as multiscaling and different ID measures are more or less robust to it[31].

Finally we note that, in Suppl. Mat. Figs. S7–12, we describe a series of 12 other control networks that show how results on the role of prediction are robust against a number of factors such as noise. These results show that predictive models outperform non-predictive models in the encoding of latent variables, at least when such encoding is probed by means of linear measures (cf. Fig. S7).

Visualizing the structure of learned neural population manifolds: signal transfer and neural manifold cells. The metrics just introduced capture properties of the neural representation at the population level via useful numbers that can be plotted over the course of learning. Here, we pause to visualize the underlying population representations in two complementary ways.

The first visualization is directly related to the metric of latent space signal transfer. In Fig. 5a the neural representation projected into the space of its first three PCs, colored according to each of the three latent variables x, y, and θ. Each point in these plots corresponds to the neural representation at a specific moment in time, and the color of the point is determined by the position or orientation of the agent in the latent environment at that moment. This shows visually that, after learning, the agent's location x, y is systematically encoded in the first three PCs, while PCs four and five encode the agent's orientation θ, Fig. 5b. This corresponds to the high values of latent space signal transfer seen at the end of learning in Fig. 3c. We next turn to visualize whether the observation variables are similarly encoded in the network representation. Figure 5c shows that, while the first three PCs do encode distance, they do not appear to encode the sensor-averaged color in any of the three RGB (red, green, blue) channels. Intriguingly, this is a consequence of learning: average color information is encoded in the first PCs in the beginning of learning as suggested by the signal transfer measure (cfr. Fig. 3c), but less in the end of it. Taken together, the visualizations in Fig. 5a, b support the conclusion from the signal transfer metrics that the network allocates most of its internal variability to the encoding of latent variables.

These visualizations of population level neural coding, as well as plots of single neuron tuning as in Fig. 2e, require foreign knowledge of the latent space variables. However, in many settings, neither the values or nature of these variables maybe known in advance. How can we proceed in these cases? We now introduce a second strategy for visualizing neural activity, via an emerging concept that we refer to as neural manifold cells[33,37].

Figure 5d shows the activity of the same 100 neurons in Fig. 2e averaged over "locations" in the space spanned by the first two PCs of the neural population activity itself. This shows tuning of individual neurons, but not with respect to motor, stimulus, or environmental variables as is typically studied—but rather with respect to population level neural activity. The approach reveals a similarity between the well known phenomenon of place cells tuned to a location in the environment and neural manifold cells tuned to a "location" on the principal components of their neural population manifold (we make this relationship made more

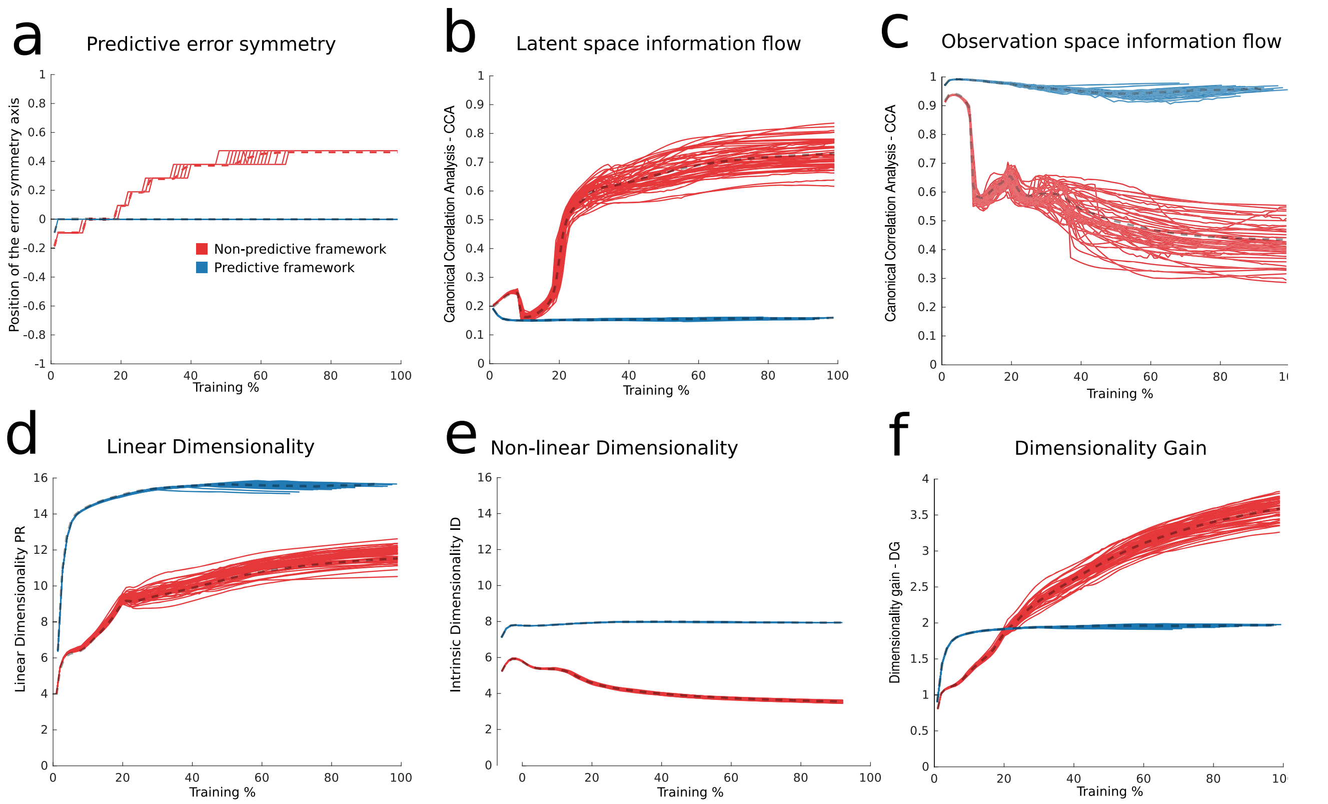

Fig. 4 Comparison between predictive and non-predictive learning. We train 50 networks of 100 neurons in each of the predictive and non-predictive conditions and equalize the learning axis between the two to highlight the trends of the different measures. **a** Predictive error. The position of the predictive error symmetry axis plotted throughout learning for the predictive and non-predictive network ensembles. The symmetry axis position is the one that minimizes a L2 norm between the predictive error curve (cf. Fig. 3a) and its reflection through the symmetry axis. **b** Latent signal transfer analysis. A canonical correlation analysis is performed between the latent space and the top PCs of the neural representation at every epoch, and the average of the two canonical correlations (for coordinates x and y) is shown. **c** Observation signal transfer analysis. The canonical correlation analysis, same as panel **b**, is performed between the top PCs of the observations and the top PCs of the network's representation. **d** Linear dimensionality (PR) throughout learning. **e** Non-linear dimensionality (ID) throughout learning. **f** Dimensionality gain (DG) throughout learning.

explicit in the context of hippocampal data in Fig. S13). Overall, this shows that receptive fields localized not just in the latent, but also in the principle component, spaces can arise naturally through predictive learning.

Predictive learning extracts latent representations of arm-reaching movements. While the spatial exploration task studied above is a useful proving ground, given the clear role played by latent spatial variables, we wished to illustrate the broader scope of the effects of predictive learning. Thus, we next apply this framework to a different task, that of predicting arm-reaching movements. We model arm movements as a dynamical system with forward and inverse kinematics according to the mitrovic model[38,39]. In this model, movements in the 2d sagittal plane of the upper right limb are modeled as a function of six muscles, Fig. 6a. The muscles control, by means of dynamical equations, two angles: the angle in between the upperarm and the line of the shoulders, and the angle in between the forearm and upperarm. The position of the elbow and wrist is then a nonlinear trigonometric function of these angles and of the lengths of the upperarm and forearm.

We cast this system into predictive learning by generating randomly correlated binary input pulses, which signal the contraction of one of the six muscles through the forward kinematics equations, resulting in exploratory movements of the arm.

We train the predictive recurrent network to predict future (x,y) locations of both the elbow and the wrist given their current locations and the input to the six muscles. This replicates the spatial exploration task description in terms of observations and actions, where observations are in this case thought to be the current locations of the elbow and wrist with respect to the shoulder Fig. 6b and actions are muscular contraction signals.

Upon learning, the network successfully predicts future observations and extracts in its neural representation the values of the underlying latent variables that ultimately regulate the movements: the two angles, see Fig. 6c, e. Owing to the low dimensionality of the observations compared with the spatial exploration task, and the fact that they are partially colinear with the latent variables, latent space signal transfer increases over the course of learning as before, but observation space signal transfer does not decay.

For the same reason, the linear dimensionality (PR), as it increases through learning, achieves a lower final value. The latent variable extraction is accompanied by the localization of neural activations on the neural population manifold and on the latent space as shown in Fig. 6f, g replicating the results shown for the spatial exploration task. Furthermore, we analyzed neural recordings in the primary motor cortex[40,41] during a motor task, as an example of how our analysis of representations of arm-reaching movements could inform future data analyses and experiments, cf. Fig. S14.

Network mechanisms that create low-D representations through prediction. In our introductory example of the card-game, we gave some mathematical reasoning for how simple

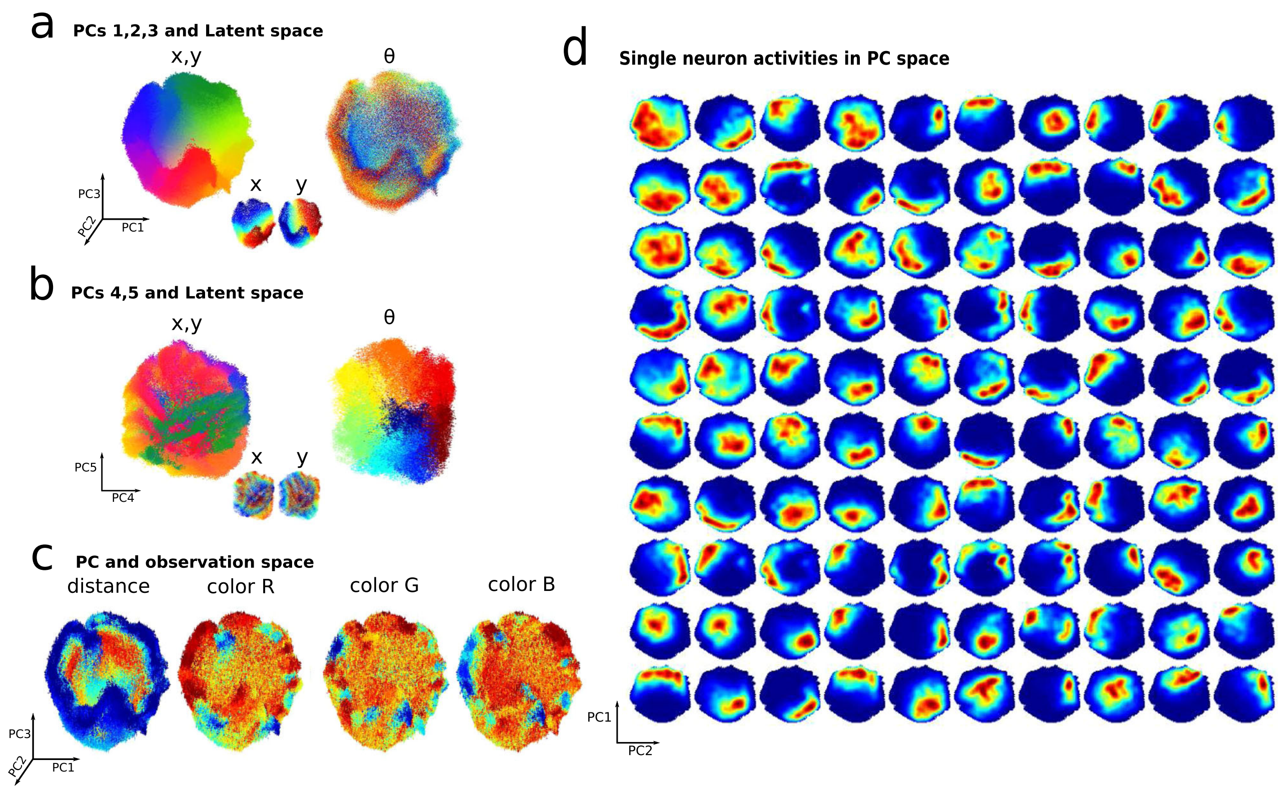

Fig. 5 Features of the learned predictive representation. a 100,000 points of the neural network representation, corresponding to an equal number of steps for the agent's exploration, are shown projected into the space spanned by PCs 1 to 3 of the learned representation, and colored, respectively, according to *x*, *y* latent variables (cfr. Fig. 1a for color code) and *θ*. **b** Same as panel **b** but for PCs 4 and 5. **c** Same as panel **a** but colored with respect to the mean distance or color activations of the agent's sensors. In this specific example, the first five PC components explain, respectively, 13.7%, 11.4%, 10.2%, 5.5%, 5.4% of the total neural variance. **d** Manifold cell activations: average activity of 100 neurons on the manifold (here displayed for the first PCs 1 and 2.). The activity of each neuron (one per quadrant) is averaged as the population activity is in a specific "location" on the neural manifold in the space spanned by PCs 1 and 2.

feedforward networks trained to predict their future inputs (observations) can extract the structure of the latent space underlying those observations. Here, we formalize this idea and extend it to recurrent networks, as used for the more general spatial and motor exploration settings studied above. Here, the RNN is governed by the equations:

$$\begin{aligned} \boldsymbol{r}_t &= g\left(\boldsymbol{W}\boldsymbol{r}_{t-1} + \boldsymbol{W}_o\boldsymbol{o}_t + \boldsymbol{W}_a\boldsymbol{a}_t\right) \\ \boldsymbol{y}_t &= g\left(\boldsymbol{W}_{\text{out}}\boldsymbol{r}_t\right) \end{aligned} \tag{5}$$

where $\boldsymbol{W}$, $\boldsymbol{W}_o$, $\boldsymbol{W}_a$, $\boldsymbol{W}_{\text{out}}$ are the weight matrices and y is the output exploited to minimize the predictive cost $\mathcal{C}_{\text{pred}} = \sum_t |y_t - o_{t+1}|^2$. Following the same logic as for the card-game task, we consider two independent network updates, denoted by A and B respectively, which lead up to the same observation o_{t+1}, read out from identical representations $r_t^A = r_t^B$. Again, up to nonlinear corrections, this gives the condition:

$$\boldsymbol{r}_{t-1}^A - \boldsymbol{r}_{t-1}^B = \boldsymbol{W}^{-1}\left(\boldsymbol{W}_o(\boldsymbol{o}_t^A - \boldsymbol{o}_t^B) + \boldsymbol{W}_a(\boldsymbol{a}_t^A - \boldsymbol{a}_t^B)\right) \tag{6}$$

which is an analogous to Eq. (4). From here, we consider two different scenarios.

In the first, the action term dominates. This gives an identical case to the one already analyzed in the introductory section Eq. (4): the action acts on the neural representation in a translationally invariant way. As before, this results in representations corresponding to different observations being translated with respect to one another similarly to how the action translates among them in the underlying latent space. For the spatial exploration task this corresponds to the product of a two-dimensional lattice and a circle (angle); for the arm-reaching task this corresponds to the product of two angles.

In the second scenario, the observation term dominates. Observations at the current time define a set of possible observations at the next timestep, those related to the current observation via one of the possible actions from the current point in the latent space. Extending the reasoning above suggests that representations r_A and r_B of latent states A and B should be similar according to the overlap in this set of possible next-timestep observations. This again suggests that the structure of latent space will be inherited by representations, as it is only states that are related by one action that can map to the same next-timestep observation. This is indeed what we find: Fig. 1e and Figs. S7–10 (case without actions) show how the latent space emerges in neural representations in predictive networks even in the absence of action inputs. However, the Supplemental Sec. S1.2 does show that these representations carry latent information in a less regular way when actions are not provided to the networks.

Taken together, these results show that the network's representation is shaped by the latent space by means of learning to predict future inputs. This connects to novel approached that have recently led to important progress in the theory of deep learning[42–44] by applying group theory to analyze neural networks[45,46]. Through this emerging perspective predictive networks, when prompted with the current observation of the state of a system ($\boldsymbol{o}$) can be analyzed as if they were asked to output the transformed observation upon applying the action of a group element g_a: $\boldsymbol{o} \mapsto g_a(\boldsymbol{o})$. In our setup we use the generators of the group instead of all possible group elements. As the network

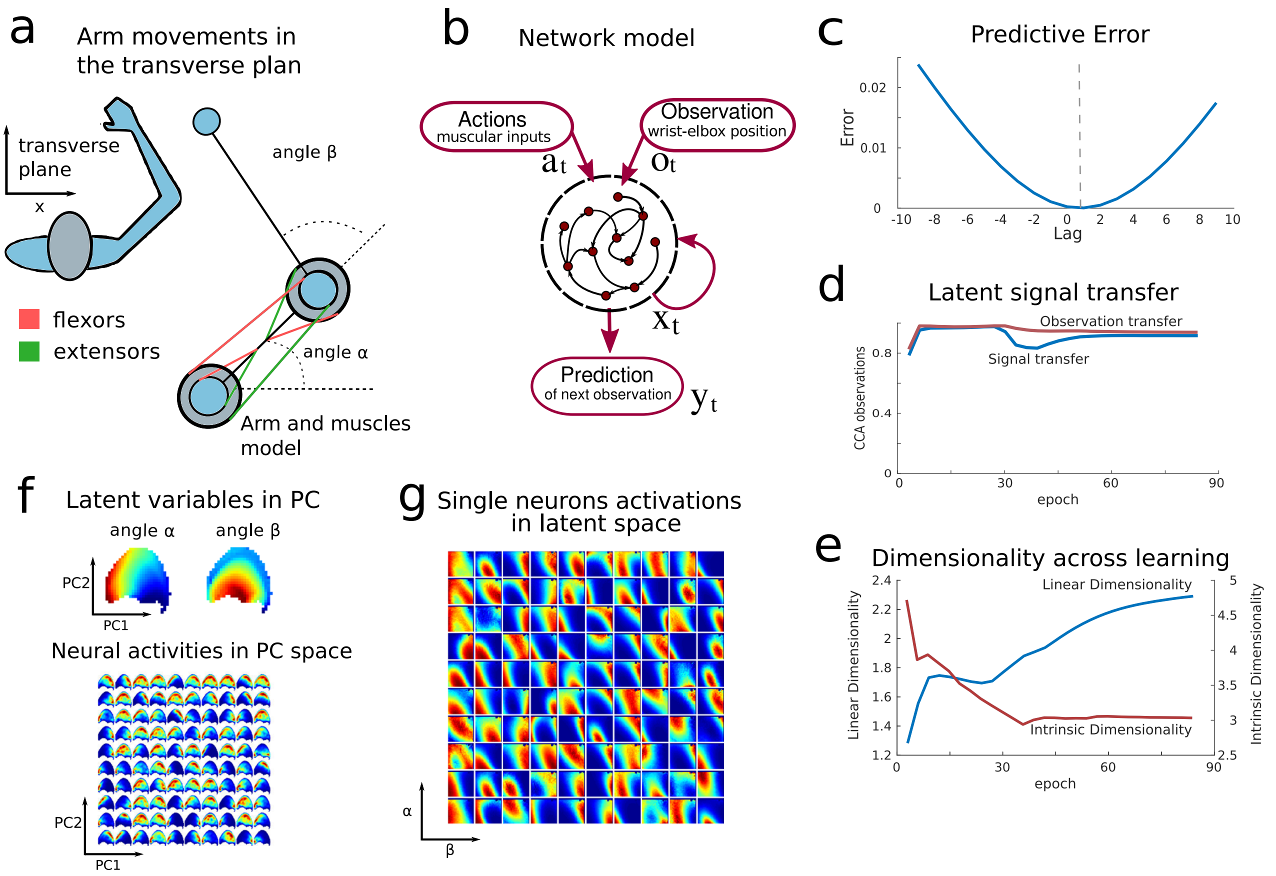

Fig. 6 Predictive representations of arm-reaching movements. a Plane transverse to the dynamic of arm-reaching movements. The muscle model is shown and the two latent angular variables α and β. **b** Recurrent network model. **c** Predictive error upon training. The symmetry axis is around lag +1 indicating that the network is carrying out the prediction correctly. **d** Latent signal transfer and observation signal transfer. **e** Dimensionality trends across learning for both linear (PR) and nonlinear (ID) dimensionality measures. **f** Top: principal components space (PCs 1–2) colored by the average angles α, β for each location. Bottom: average activity of neurons in the space spanned by the top 2 PCs. Each subplot represents the average activity of a single neurons. Neurons are ranked according to their average firing rates. The most active neuron is in the top left corner, the second in the first column second row and so on for all the neurons. **g** Average activity of neurons in latent space α, β. Each subplot corresponds to the neuron in panel **f**.

learns to apply group actions g_a to its representation, it transforms, through its layers, the given observation $\boldsymbol{o}$ into a neural representation onto which the action acts as a group element.

At this stage the network's representation inherits the geometry of what is called the group's representation. For example, in the spatial exploration example, the states in which the agent can be found are defined by the Special Euclidean group of rotations and translations in two dimensions SE(2). In our framework the actions of the agent correspond to the group generators for translations—reflecting minimal translational movements of the agent (the angle, corresponding to the rotation degree, is not directly provided). Thus, the action passed to the network is formally the one relative to the translation subgroup, and it is provided in vectorial form. As these group generators act as vectorial translations on the neural representations, a definite geometry is inherited by the network representation: the translation subgroup of SE(2) is encoded as a two-dimensional lattice[47]. This is a more general way to arrive at the conclusions of the direct calculations taken above.

The analysis above shows how the structure of the latent space shapes the structure of neural representations. This structure can be clearly visualized in many of the plots presented above. Moreover, it is reflected in the metrics we introduce in at least two ways. First, we expect that states being represented in a transitionally invariant way will lead to the ability to decode states from neural representations; how this plays out for the principal components of neural activity that are used for plotting neural activity above and for the metric of latent space signal transfer is described using results from the linear algebra of Toeplitz matrices in Supplemental Secs. S2.1–2.3. Second, states being represented in a transitionally invariant way leads to an approximate parameterization of neural activity via terms of the latent space, corresponding to the lower values of intrinsic dimensionality also measured above.

By contrast, as an autoencoder does not compute the action of a group element on its input, is not generally expected to build a representation with structure induced by that group. Nonetheless a group theoretic approach to autoencoders still enables insights into why autoencoders develop activations reminiscent of receptive fields[48]. In the Suppl. Mat. Sec. 2.5 we provide further considerations on the locality of receptive fields mainly inspired by ref. [37].

Discussion

How the brain extracts information about the latent structures of the external world, given only its sensory observations, is a longstanding question. Here, we show that the computation of predicting future inputs can contribute to this process, giving rise to to low-dimensional neural representation of the underlying latent spaces in artificial neural networks. We demonstrate this phenomenon in a sequence of gradually more complex simulations and by providing basic mathematical arguments that indicate its generality.

What features of neural responses, or representations, characterize predictive learning? When the observations to be predicted arise from an environment with an underlying low-dimensional latent structure, e.g., in the case of spatial exploration or arm-reaching movements, our work suggests several distinct features. First, the predictive error shows that neural representations are biased towards encoding upcoming observations or latent variables. Second the latent structure underlying the observations is transferred onto the representation progressively through learning (Latent Signal Transfer, cf. Fig. 5). Finally, the dimensionality of the set of neural responses will likely appear high when assessed with standard linear measures, such as participation ratio[28,29]. However, when assessed through nonlinear metrics sensitive to the dimensionality of curved manifolds, the dimensionality will be lower, in the ideal case tending to the number of independent latent variables.

This last feature is the result of neural responses being strongly tuned to the variables, which parameterize the neural representation manifold (cfr. Fig. 5d). An established example of such strong coding is the locality of neural receptive fields in latent space (e.g., place fields). Here, we observe an allied phenomenon, that of manifold cells with local receptive fields on the manifold of population-wide neural responses. This is a feature that can be explored in artificial network studies of complex data, or in experimental settings (cf. proof-of-concept data analysis in Suppl. Mat. Fig. S13) where the underlying latent variables do not need to be known in advance. This feature connects to recent work on understanding neuronal representations through the lens of dimensionality[27–29,37,49]. Overall, these features provide a quantitative framework to compare representations across conditions that can be applied both in machine learning (e.g., to compare learning schemes and overall mechanisms of extracting latent signals from data) and in brain circuits (e.g., to compare coding in distinct brain areas).

Our findings should not be taken as a theory of a specific brain area but rather as a formulation of a general connection between predictive coding and the extraction of latent information from sensory data. For example, our model falls short in explaining mechanistically key elements of spatial maps individuated in hippocampal recordings, such as the emergence of place cells and their relation to direction or grid cells. However, it does suggest that predictive learning is a mechanism that enables the binding of sensory information beyond spatial exploration and towards the more general notion of semantically related episodes. While traditionally distinct theories of hippocampus involve declarative memory[50]) and spatial exploration[51], considerable effort has been devoted to reconciling these apparently contrasting views[52–55]. In particular, Eichenbaum[54] proposed that the hippocampus supports a semantic relational network that organizes related episodes to subserve sequential planning[8,9,56]. Here, we posit that prediction—with its ability to extract latent information—may serve as such a mechanism to generate semantic relational networks. In particular, we speculate that relevant semantic relations are encoded by neural representations of low intrinsic dimensionality, which are constructed by predictive learning to reflect the relevant latent variables in a task. Our results substantiate and build on the importance of allied frameworks in constructing such relational networks[15,16,57]. Overall the predictive learning framework provides a potential alternative of generating hippocampal representations, which differs from both attractor[58,59] and path-integration models[60,61], while maintaining elements of both these models. Discerning the underlying differences and similarities will require careful future investigations.

From an algorithmic and computational perspective, our proposal is motivated by the recent success of predictive models in machine-learning tasks that require vector representations reflecting semantic relationships in the data. Information retrieval and computational linguistics have benefited enormously from the geometric properties of word embeddings learned by predictive models[11–13,62]. Furthermore, prediction over observations has been used as an auxiliary task in reinforcement learning to acquire representations favoring goal-directed learning[9,16–18]. Alongside these studies there are other emerging frameworks that are related to the predictive learning networks we analyze: contrastive predictive coding[63,64], information theoretic approaches[65,66] and world models[67]. Furthermore, our contribution shall also be seen in light of computational models studying neurons with optic flow selectivity[68,69].

Predictive learning is a general framework that goes beyond the examples analyzed here, and future work can expand in other directions (text, visual processing, behavioral tasks, etc.) that may open new theoretical advances and new implications for learning and generalization. It will also be exciting to adapt and test these ideas for the analysis of large-scale population recordings of in vivo neural data—ideally longitudinally, so that the evolution of learned neural representations can be tracked with metrics such as the emergence of a low-D neural representation manifold, predictive error, latent signal transfer and dimensionality gain. A very interesting possibility is that this might uncover the presence of latent variables in tasks where they were previously unsuspected or unidentified. Our techniques require no advance knowledge of the latent variables. The consequence is that both the number and identity of latent variables can be discovered by analysis of a learned neural response manifold, as studied in other settings[62,70–72].

Furthermore, it will be important to develop a formal connection between predictive learning mechanisms[73,74] and reinforcement learning (RL) paradigms[9,75] in both model-free and model-based schemes[76–78]. This has the potential to build a general framework that could uncover predictive learning behavior in both animals and humans. One step here would be to extend existing RL paradigms to scenarios where making predictions is important even in the absence of rewards[79–82].

Methods

Card-game network. We generate a two-dimensional 5 x 5 grid of states, which is the latent space. To each state, we randomly assign a random set of five cards from a deck of 40, sampled with no repetition. This serves as an example of observations associated to states, which are fully random, independent, and of arbitrary complexity. In particular the dimensionality of the observation is not tied to the dimensionality of the latent space. We generate 10^6 state transitions following the five actions as defined in the main text. Upon generating such sequence of states we train a feedforward network to predict upcoming obeservations given current ones. The network is a two-layer network with 100 neurons in both layers, the first with sigmoidal transfer function and the second with hyperbolic tangent followed by a binary cross-entropy cost function. Both actions and observations have a one-hot encoding. All weights are initialized with random normal matrices. Training is performed on 80% of the sequence and validated on the remaining 20% utilizing a RMSprop optimizer (parameters: learning constant = 0.0001, $\alpha = 0.95$, ϵ regularizer $= 1 \cdot 10^{-7}$). The learning rate was reduced of a factor 0.5 if the validation loss did not decrease for eight consecutive epochs (reducing on plateau scheme). Training was stopped after 25 epochs with no improvement in the validation loss (min delta of variation 5e-5). The neural network used for Fig. 2e is identical to the one just described, except that the output is read out at the second layer (the hyperbolic tangent layer) with mean-squared error. This is to account for the fact that the prediction, when actions are not passed to the network, is probabilistic towards neighboring states. All simulations were performed in Keras.

Neural network model for the spatial exploration task. We study a recurrent neural network (RNN) that generates predictive neural representations during the exploration of partially observable environments. RNNs are suited to processing sequence-to-sequence tasks[83] and the state of a recurrent network is a function of the history of previous inputs and can thus be exploited to learn contextually appropriate responses to a new given input[84–86].

Figure 2c illustrates the RNN model: at a given time t the RNN receives as input an observation vector $\overrightarrow{o}$ and a vector representation of the action $\overrightarrow{a}$. The internal state $\overrightarrow{r}^t$ of the network is updated and used to generate the network's output through the following set of equations:

$$\begin{aligned} \boldsymbol{r}_t &= g(\boldsymbol{W}\boldsymbol{r}_{t-1} + \boldsymbol{W}_o\boldsymbol{o}_t + \boldsymbol{W}_a\boldsymbol{a}_t) \\ \boldsymbol{y}_t &= g(\boldsymbol{W}_{\text{out}}\boldsymbol{r}_t) \end{aligned} \tag{7}$$

The RNN is trained to predict the observation at the next timestep by minimizing the first cost function, or alternatively to autoencode its input, via the predictive and non-predictive cost functions, respectively:

$$\begin{aligned} \mathcal{C}_{\text{pred}} &= \frac{1}{T}\sum_{t=0}^{T-1} ||\boldsymbol{o}_{t+1} - \boldsymbol{y}_t||^2 , \\ \mathcal{C}_{\text{non-pred}} &= \frac{1}{T}\sum_{t=0}^{T-1} ||\boldsymbol{o}_t - \boldsymbol{y}_t||^2 . \end{aligned} \tag{8}$$

Networks were trained by minimizing the cost function in Eq. (8) via backpropagation through time[87]. While RNNs are known to be difficult to train in many cases[88], a simple vanilla RNN model with hyperbolic tangent activation function was able to learn our task, Fig. 2d.

The connectivity matrix of the recurrent network was initialized to the identity[89,90], while input and output connectivity matrices were initialized to be random matrices. Individual weights were sampled from a normal distribution with mean zero and standard deviation 0.02. The network had 500 recurrent units (with the exception noted below), while the input and output size depended on the task as defined by the environment. Each epoch of training corresponded to $T = 10^6$ time steps.

All other training details were the same as reported for the card-game example. For the simulations of Fig. 5, we trained 100 networks of 100 neurons: 50 networks in the predictive case and 50 networks in the non-predictive case (cf. Eq. (8) with equal instantiation of the rest of parameters.

Description of the spatial environment. modeled the spatial exploration task in two dimensions. We simulated the exploration of the agent in a square maze tessellated by a grid of evenly spaced cells ($64 \times 64 = 4096$ locations). At every time t the agent was in a given location in the maze and headed in a direction $\varphi \in [0, 2\pi)$. The agent executed a random walk in the maze, which was simulated as follows. At every step in the simulation an action was selected by updating the direction variable θ stochastically with $d\theta$ (*i.i.d.* sampled from a Gaussian distribution with variance $\sigma^2_{\text{theta}} = 0.5\ \ \text{rad}$). The agent then attempted a move to the cell, among the eight adjacent ones, that was best aligned to θ. The move occurred unless the target cell was occupied by a wall, in which case the agent remained in the current position but updated its angle with an increment twice the size of a regular one: $\sigma^2_{\text{theta}} = 1.0\ \ \text{rad}$. To ensure coherence between updates in the direction θ and the cell towards which the agent just moved, we required each update in $d\theta$ to be towards the direction of the agent's last movement d_a so that $d\theta \cdot (\theta - d_a)$ would always be positive, where d_a assumed one of 8 values depending on the action taken by the agent.

The chosen action was encoded in a one-hot vector that indexed the movement. The actions were discrete choices $a_t \in [0..8]$ correlating with the head direction but distinct from it. This was indeed a continuous variable $\theta_t \in [0, 2\pi)$. Moreover, knowledge of the action didn't provide direct information about the agent's direction and observation; in other words, there was no direct correspondance between the action taken and the observation collected as for each location and action there were many possible directions the agent could point towards and consequently as many possible observations.

As the agent explored the environment it collected, through a set of $N_s = 5$ sensors, observations of the distance and color of the walls along five different directions equally spaced in a 90 degree visual cone centered at φ. Thus it recorded, for each sensor, four variables at every timestep: the distance from the wall and the RGB components of the color of the wall. This information was represented by a vector o_t of size $5 \times 4 = 20$. Such a vector, together with the action represented as a one-hot representation, was fed as input into the network and used for the training procedure. The walls were initially colored so that each tile corresponding to a wall carried a random color (i.e., three uniformly randomly generated numbers in the interval [0,1]). A Gaussian filter of variance two tiles was then used, for each color channel, to make the color representations smooth. Figure 2b shows an example of such an environment.

Predictive error. The predictive error is a direct generalization of Eq. (8) as a function of a time lag variable:

$$\mathcal{C}_{\text{pred}}(\text{lag}) = \frac{1}{T}\sum_{t=0}^{T-1} ||\boldsymbol{o}_{t+\text{lag}} - \boldsymbol{y}_t||^2, \tag{9}$$

so that it is possible to verify that the output of the network y is most similar, on average, to the upcoming observation rather than the current observation.

Latent signal transfer. The latent signal transfer measure was obtained by performing a canonical correlation analysis (CCA) between two spaces: the top 3 PC components of the network's representation and other variables as specified in the text, e.g., latent variables (x,y). CCA extracts the directions of maximal correlation between the two spaces returning a set of canonical correlations. Latent signal transfer is then taken to be the average of these canonical correlations, which are as many as the minimum between the ranks of the two spaces.

Nonlinear dimensionality: intrinsic dimensionality. While research on estimating intrinsic dimensionality (ID) is advancing, there is still no single decisive algorithm to do so; rather, we adopt the recommended practice of computing and reporting several (here, five) different estimates of ID based on distinct ideas[31,32]. The set of techniques we use include: MiND$_{\text{ML}}$[91], MLE[92], DancoFit[93], CorrDim[94], and GMST[95,96]. These techniques follow the selection criteria illustrated in ref. [31], emphasizing the ability to handle high-dimensional data (in our case hundreds of dimensions) and being robust, efficient, and reliable; we refer the reader to ref. [25] for a useful comparison. We implement these techniques using the code from the the authors available online[31,92,93], "out of the box" without modifying hyperparameters.

A simple intuition regarding for some of the selected techniques builds on the notion of correlation dimension, which derives from the following idea. Consider a manifold $\mathcal{M}$ of dimensionality d embedded in IR^N and a set of points uniformly sampled from the manifold. For each point build a ball of radius r (denoted as B_r), then the number of points within B_r (denoted as $\#B_r$) can be analyzed as a function of r and be found to scale as $\#B_r \sim r^d$ at least for small r. This scaling can be exploited to estimate d.

Description of arm-reaching movements model. To model arm-reaching movements we used a kinematic model of the arm muscles[97,98]. The arm kinematics were modeled in the transverse plane by analyzing the effect of six muscles on the arm dynamics, cf. Fig. 6a. The activation signals for the muscles were used as actions in our model. For each of the six muscles, we used a pulsed binary signal where at each instant in time the pulse can be turned on or off. These activation signals are filtered and passed to the equations of inverse kinematics of the muscles, which regulate muscular contraction. Such muscle dynamics drives the arm dynamics according to the Mitrovic model[38,99,100]. All the details regarding the implementations of this model can be found on the Github repository we adopted for the simulations https://github.com/jeremiedecock/pyarm and in the code we provide. The most relevant feature of this model for our study is the fact that the six-dimensional muscle activity drives nonlinear dynamics in the two-dimensional latent space described by the two angles α, β in Fig. 6a.

Reporting summary. Further information on research design is available in the Nature Research Reporting Summary linked to this article.

Data availability

All data generated through the simulations generated is made available from the corresponding author upon reasonable request.

Code availability

All code is made available from the corresponding author upon reasonable request.

Received: 12 July 2019; Accepted: 22 January 2021;
Published online: 03 March 2021

References

1. Bengio, Yoshua. in *Statistical Language and Speech Processing, number 7978 in Lecture Notes in Computer Science* (eds Dediu, A.-H., Martín-Vide, C., Mitkov, R. & Truthe, B.) 1–37. (Springer, 2013).
2. Laje, R. & Buonomano, D. V. Robust timing and motor patterns by taming chaos in recurrent neural networks. *Nat. Neurosci.* **16**, 925–933 (2013).
3. Mnih, V. et al. Human-level control through deep reinforcement learning. *Nature* **518**, 529–533 (2015).
4. Silver, D. et al. Mastering the game of Go with deep neural networks and tree search. *Nature* **529**, 484–489 (2016).
5. Graves, A. et al. Hybrid computing using a neural network with dynamic external memory. *Nature* **538**, 471 (2016).
6. Kulkarni, T. D., Saeedi, A., Gautam, S. & Gershman, S. J. Deep successor reinforcement learning. https://arxiv.org/abs/1606.02396 (2016).
7. Konovalov, A. & Krajbich, I. Neurocomputational dynamics of sequence learning. *Neuron* **98**, 1282–+ (2018).
8. Banino, A. et al. Vector-based navigation using grid-like representations in artificial agents. *Nature* **557**, 429–433 (2018).
9. Wayne, G. et al. Unsupervised predictive memory in a goal-directed agent. Preprint at https://arxiv.org/abs/1803.10760 (2018).

10. Yang, G. R., Joglekar, M. R., Song, H. F., Newsome, W. T. & Wang, X.-J. Task representations in neural networks trained to perform many cognitive tasks. *Nat. Neurosci.* **22**, 297–306 (2019).
11. Bengio, Y., Ducharme, R., Vincent, P. & Jauvin, C. A neural probabilistic language model. *J. Mach. Learn. Res.* **3**, 1137–1155 (2003).
12. Turian, J., Ratinov, L. & Bengio, Y. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 384–394. (Association for Computational Linguistics, 2010).
13. Collobert, R. et al. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.* **12**, 2493–2537 (2011).
14. Mikolov, T., Chen, K., Corrado, G. & Dean, J. Efficient estimation of word representations in vector space. Preprint at https://arxiv.org/abs/1301.3781 (2013).
15. Arora, S., Li, Y., Liang, Y., Ma, T. & Risteski, A. Rand-walk: a latent variable model approach to word embeddings. Preprint at https://arxiv.org/abs/1502.03520arxiv (2015).
16. Dayan, P. Improving generalization for temporal difference learning: the successor representation. *Neural Comput.* **5**, 613–624 (1993).
17. Stachenfeld, K. L., Botvinick, M. & Gershman, S. J. in *Advances in Neural Information Processing Systems* 27 (eds Ghahramani, Z., Welling, M., Cortes, C. Lawrence, N. D. & Weinberger, K. Q.) 2528–2536 (Curran Associates, Inc., 2014).
18. Russek, E. M., Momennejad, I., Botvinick, M. M., Gershman, S. J. & Daw, N. D. Predictive representations can link model-based reinforcement learning to model-free mechanisms. *PLoS Computat. Biol.* **13**, e1005768 (2017).
19. Huang, Y. & Rao, R. P. N. Predictive coding. *Wiley Interdiscip. Rev.: Cognit. Sci.* **2**, 580–593 (2011).
20. Spratling, M. W. A review of predictive coding algorithms. *Brain Cogn.* **112**, 92–97 (2017).
21. Koren, V. & Denève, S. Computational account of spontaneous activity as a signature of predictive coding. *PLoS Computat. Biol.* **13**, e1005355 (2017).
22. Blei, D. M. Build, compute, critique, repeat: data analysis with latent variable models. *Ann. Rev. Stat. Appl.* **1**, 203–232 (2014).
23. Salakhutdinov, R. Learning deep generative models. *Ann. Rev. Stat. Appl.* **2**, 361–385 (2015).
24. Kim, B., Lee, K. H., Xue, L. & Niu, X. A review of dynamic network models with latent variables. *Stat. Surv.* **12**, 105 (2018).
25. Van Der Maaten, L., Postma, E. & Van den Herik, J. Dimensionality reduction: a comparative. *J. Mach. Learn. Res.* **10**, 66–71 (2009).
26. Abbott, L. F, Rajan, K. & Sompolinsky, H. in *The Dynamic Brain: an Exploration of Neuronal Variability and Its Functional Significance.*1–16 (OUP, 2011).
27. Rigotti, M. et al. The importance of mixed selectivity in complex cognitive tasks. *Nature* **497**, 585 (2013).
28. Mazzucato, L., Fontanini, A. & Camera, G. L. Stimuli reduce the dimensionality of cortical activity. *Front. Syst. Neurosci.* **10**, 11 (2016).
29. Litwin-Kumar, A., Harris, K. D., Axel, R., Sompolinsky, H. & Abbott, L. F. Optimal degrees of synaptic connectivity. *Neuron* **93**, 1153–1164.e7 (2017).
30. Gao, P. et al. A theory of multineuronal dimensionality, dynamics and measurement. Preprint at *bioRxiv*, https://doi.org/10.1101/214262page (2017).
31. Camastra, F. & Staiano, A. Intrinsic dimension estimation: advances and open problems. *Information Sci.* **328**, 26–41 (2016).
32. Campadelli, P., Casiraghi, E., Ceruti, C. & Rozza, A. Intrinsic dimension estimation: relevant techniques and a benchmark framework. *Math. Probl. Eng.* **2015**, 759567 (2015).
33. Low, R. J, Lewallen, S., Aronov, D., Nevers, R. & Tank, D. W. Probing variability in a cognitive map using manifold inference from neural dynamics. Preprint at *bioRxiv*, https://doi.org/10.1101/418939 (2018).
34. Farrell, M., Recanatesi, S., Lajoie, G. & Shea-Brown, E. Recurrent neural networks learn robust representations by dynamically balancing compression and expansion. Preprint at *bioRxiv* https://doi.org/10.1101/564476 (2019).
35. Recanatesi, S. et al. Dimensionality compression and expansion in deep neural networks. Preprint at https://arxiv.org/abs/1906.00443 (2019).
36. Palmer, S. E., Marre, O., Berry, M. J. & Bialek, W. Predictive information in a sensory population. *Proc. Natl Acad Sci* **112**, 6908–6913 (2015).
37. Sengupta, A., Tepper, M., Pehlevan, C., Genkin, A. & Chklovskii, D.. Manifold-tiling localized receptive fields are optimal in similarity-preserving neural networks. Preprint at *bioRxiv* https://doi.org/10.1101/338947 (2018).
38. Mitrovic, D., Klanke, S., Osu, R., Kawato, M. & Vijayakumar, S. A computational model of limb impedance control based on principles of internal model uncertainty. *PLoS ONE*, **5**, e1360 (2010).
39. Mitrovic, D. *Stochastic Optimal Control with Learned Dynamics Models.* Edinburgh Research Archive (2011).
40. Lawlor, P. N., Perich, M. G., Miller, L. E. & Kording, K. P. Linear-nonlinear-time-warp-poisson models of neural activity. *J. Comput. Neurosci.* **45**, 173–191 (2018).
41. Perich, M. G., Lawlor, P. N., Kording, K. P., & Miller, L. E. *Extracellular Neural Recordings from Macaque Primary and Dorsal Premotor Motor Cortex during A Sequential Reaching Task.* (CNRS.org, 2018).
42. Kondor, R. & Trivedi, S. On the generalization of equivariance and convolution in neural networks to the action of compact groups. *International Conference on Machine Learning*, 2747–2755 (2018).
43. Cohen, T., Geiger, M. & Weiler, M. A general theory of equivariant cnns on homogeneous spaces. Preprint at https://arxiv.org/abs/1811.02017 (2018).
44. Esteves, C. Theoretical aspects of group equivariant neural networks. Preprint at https://arxiv.org/abs/2004.05154 (2020).
45. Ravanbakhsh, S., Schneider, J. & Póczos, B. Equivariance through parameter-sharing. *International Conference on Machine Learning*, 2892–2901 (2017).
46. Keriven, N. et al. *Advances in Neural Information Processing Systems 32*, pages 7092–7101 (Curran Associates, Inc., 2019).
47. Gallier, J. & Quaintance, J. *Aspects of Harmonic Analysis and Representation Theory.* (2019). https://www.seas.upenn.edu/~jean/nc-harmonic.pdf.
48. Paul, A. & Venkatasubramanian, S. *Why does Deep Learning work?-A perspective from Group Theory.* Preprint at https://arxiv.org/abs/1412.6621 (2015).
49. Cayco-Gajic, N. A., Clopath, C. & Silver, R. A. Sparse synaptic connectivity is required for decorrelation and pattern separation in feedforward networks. *Nat. Commun.* **8**, 1116 (2017).
50. Cohen, N. J. & Squire, L. R. Preserved learning and retention of pattern-analyzing skill in amnesia: dissociation of knowing how and knowing that. *Science* **210**, 207–210 (1980).
51. O'Keefe, J. & Dostrovsky, J. The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat. *Brain Res.* **34**, 171–175 (1971).
52. Buzsáki, G. & Moser, E. I. Memory, navigation and theta rhythm in the hippocampal-entorhinal system. *Nat. Neurosci.* **16**, 130–138 (2013).
53. Milivojevic, B. & Doeller, C. F. Mnemonic networks in the hippocampal formation: from spatial maps to temporal and conceptual codes. *J. Exp. Psychol.* **142**, 1231 (2013).
54. Eichenbaum, H. & Cohen, N. J. Can we reconcile the declarative memory and spatial navigation views on hippocampal function? *Neuron* **83**, 764–770 (2014).
55. Schiller, D. et al. Memory and space: towards an understanding of the cognitive map. *J. Neurosci*. **35**, 13904–13911 (2015).
56. Kanitscheider, I. & Fiete, I. in *Advances in Neural Information Processing Systems*, 4529–4538, (MIT Press, 2017).
57. Stachenfeld, K. L., Botvinick, M. M & Gershman, S. J. The hippocampus as a predictive map. *Nat. Neurosci.* **20**, 1643–1653 (2017).
58. Tsodyks, M. Attractor neural network models of spatial maps in hippocampus. *Hippocampus* **9**, 481–489 (1999).
59. Rolls, E. T. An attractor network in the hippocampus: theory and neurophysiology. *Learn. Memory* **14**, 714–731 (2007).
60. McNaughton, B. L., Battaglia, F. P., Jensen, O., Moser, E. I. & Moser, M. Path integration and the neural basis of the 'cognitive map'. *Nat. Rev. Neurosci.* **7**, 663–678 (2006).
61. Savelli, F. & Knierim, J. J. Origin and role of path integration in the cognitive representations of the hippocampus: computational insights into open questions. *J. Exp. Biol.* **222**, jeb188912 (2019).
62. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. & Dean, J. in *Advances in Neural Information Processing Systems*, 3111–3119 (MIT Press, 2013).
63. van den Oord, A., Li, Y. & Vinyals, O. Representation learning with contrastive predictive coding. Preprint at https://arxiv.org/abs/1807.03748 (2018).
64. Hénaff, O. J. Data-efficient image recognition with contrastive predictive coding. *International Conference on Machine Learning*, 4182–4192 (2020).
65. Bachman, P., Devon Hjelm, R. & Buchwalter, W. Learning representations by maximizing mutual information across views. Preprint at https://arxiv.org/abs/1906.00910 (2019).
66. Trinh, T. H., Luong, M.-T. & Le, Q. V. Selfie: self-supervised pretraining for image embedding. Preprint at https://arxiv.org/abs/1906.02940 (2019).
67. Freeman, C. D., Metz, L. & Ha, D. Learning to predict without looking ahead: world models without forward prediction. Preprint at https://arxiv.org/abs/1910.13038 (2019).
68. Olshausen, B. A. & Field, D. J. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature* **381**, 607–609 (1996).
69. Beardsley, S. A. & Vaina, L. M. Computational modelling of optic flow selectivity in MSTd neurons. *Network (Bristol, England)* **9**, 467–493 (1998).
70. Hinton, G. E. & Salakhutdinov, R. R. Reducing the dimensionality of data with neural networks. *Science* **313**, 504–507 (2006).
71. Hastie, T., Tibshirani, R. & Friedman, J. in *The Elements of Statistical Learning*, 485–585. (Springer, 2009).
72. Weinberger, K. Q. & Saul, L. K. Unsupervised learning of image manifolds by semidefinite programming. *Int. J. Comput. Vis.* **70**, 77–90 (2006).

73. Huang, Y. & Rao, R. P. N. Predictive coding. *Wiley Interdiscipl. Rev. Cognit. Sci.* **2**, 580–593 (2011).
74. Denève, S., Alemi, A. & Bourdoukan, R. The brain as an efficient and robust adaptive learner. *Neuron* **94**, 969–977 (2017).
75. Gershman, S. J. & Daw, N. D. Reinforcement learning and episodic memory in humans and animals: an integrative framework. *Ann. Rev. Psychol.* **68**, 101–128 (2017).
76. Russek, E. M., Momennejad, I., Botvinick, M. M., Gershman, S. J. & Daw, N. D. Predictive representations can link model-based reinforcement learning to model-free mechanisms. *PLoS Comput. Biol.* **13**, e1005768 (2017).
77. Momennejad, I. et al. The successor representation in human reinforcement learning. *Nat. Human Behav.***1**, 680–692 (2017).
78. Vikbladh, O. M. et al. Hippocampal contributions to model-based planning and spatial memory. *Neuron* **102**, 683–693 (2019).
79. O'Doherty, J. P., Dayan, P., Friston, K., Critchley, H. & Dolan, R. J. Temporal difference models and reward-related learning in the human brain. *Neuron* **38**, 329–337 (2003).
80. Duncan, K., Semmler, A. & Shohamy, D. Modulating the use of multiple memory systems in value-based decisions with contextual novelty. *J. Cognit. Neurosci.* **31**, 1455–1467 (2019).
81. Biderman, N., Bakkour, A. & Shohamy, D. What are memories for? the hippocampus bridges past experience with future decisions. *Trend. Cognit. Sci.* https://doi.org/10.1016/j.tics.2020.04.004 (2020).
82. Webb, T., Dulberg, Z., Frankland, S., Petrov, A., O'Reilly, R. & Cohen, J. Learning representations that support extrapolation. *International Conference on Machine Learning*, 10136–10146 (2020).
83. Sutskever, I., Vinyals, O. & Le, Q. V. in *Advances in Neural Information Processing Systems* 3104–3112 (MIT Press, 2014).
84. Rigotti, M., Rubin, D. B. D., Wang, Xiao-Jing & Fusi, S. Internal representation of task rules by recurrent dynamics: the importance of the diversity of neural responses. *Front. Computat. Neurosci.* **4**, 29 (2010).
85. Rigotti, M., Rubin, D. B. D., Morrison, S. E., Salzman, C. D. & Fusi, S. Attractor concretion as a mechanism for the formation of context representations. *Neuroimage* **52**, 833–847 (2010).
86. Lipton, Z. C. A critical review of recurrent neural networks for sequence learning. *Preprint at* https://arxiv.org/abs/1506.00019 (2015).
87. Werbos, P. J. Backpropagation through time: what it does and how to do it. *Proc. IEEE* **78**, 1550–1560 (1990).
88. Pascanu, R., Mikolov, T. & Bengio, Y. On the difficulty of training recurrent neural networks. *International conference on machine learning*, 1310–1318 (2013).
89. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
90. Collins, J., Sohl-Dickstein, J. and Sussillo, D. Capacity and trainability in recurrent neural networks. Preprint at https://arxiv.org/abs/1611.09913 (2016).
91. Lombardi, G., Rozza, A., Ceruti, C., Casiraghi, E. & Campadelli, P. Minimum neighbor distance estimators of intrinsic dimension. In *Proceedings of the 2011 European Conference on Machine Learning and Knowledge Discovery in Databases-Volume Part II*, ECML PKDD'11, 374–389, (Springer-Verlag, 2011).
92. Levina, E. & Bickel, P. J. in *Advances in Neural Information Processing Systems 17* (eds Saul, L. K., Weiss, Y. & Bottou, L.) 777–784 (MIT Press, 2005).
93. Ceruti, C. et al. DANCo: dimensionality from angle and norm concentration. Preprint at https://arxiv.org/abs/1206.3881 (2012).
94. Grassberger, P. & Procaccia, I. Measuring the strangeness of strange attractors. *Physica D* **9**, 189–208 (1983).
95. Tenenbaum, J. B., De Silva, V. & Langford, J. C. A global geometric framework for nonlinear dimensionality reduction. *Science* **290**, 2319–2323 (2000).
96. Costa, J. & Hero, A. Manifold learning with geodesic minimal spanning trees. Preprint at https://arxiv.org/abs/cs/0307038 (2003).
97. Marin, D., Decock, J., Rigoux, L. & Sigaud, O. Learning cost-efficient control policies with XCSF: generalization capabilities and further improvement. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, GECCO '11, 1235–1242 (Association for Computing Machinery, Dublin, Ireland, 2011).
98. Lanzi, P. L. & Loiacono, D. XCSF with tile coding in discontinuous action-value landscapes. *Evol. Intell.* **8**, 117–132 (2015).
99. Mitrovic, D., Klanke, S. & Vijayakumar, S. Adaptive optimal control for redundantly actuated arms. In *International Conference on Simulation of Adaptive Behavior*, 93–102. (Springer, 2008).
100. Mitrovic, D., Klanke, S. & Vijayakumar, S. in *From Motor Learning to Interaction Learning in Robots*, 65–84. (Springer, 2010).

Acknowledgements

E.S.B. is supported by NSF Grant 1514743, wishes to thank the Allen Institute for Brain Science founders, Paul and Jody Allen, for their vision, encouragement, and support. G.L. is supported by an NSERC Discovery Grant (RGPIN-2018-04821) and the FRQNT Young Investigator Startup Program (2019NC253251). Part of this work was conducted during an internship at IBM Research. The authors would like to acknowledge the numerous colleagues who have helped generate the ideas of the paper. In particular, we thank Luca Mazzucato (University of Oregon, USA), Kameron Decker Harris (University of Washington, USA), Stefan Mihalas (Allen Institute for Brain Science, USA), Greg Wayne (DeepMind, UK), and Alon Rubin (Weizmann Institute, Israel), Cengiz Pehlevan (Harvard University).

Author contributions

S.R.: conceptualization, formal analysis, software, validation, visualization, writing. M.F.: conceptualization, formal analysis, review and editing. G.L.: conceptualization, formal analysis, review and editing. S.D.: conceptualization, formal analysis, review and editing. M.R.: conceptualization, project administration, supervision, writing, review and editing. E.S.B.: conceptualization, project administration, supervision, writing, review and editing.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41467-021-21696-1.

Correspondence and requests for materials should be addressed to S.R.

Peer review information *Nature Communications* thanks Florian Raudies and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. Peer reviewer reports are available.

Reprints and permission information is available at http://www.nature.com/reprints

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Supplementary Material: Predictive learning as a network mechanism for extracting low-dimensional latent space representations.

Contents

1 Predictive learning and representations in the simple "card game" example: Further analysis

1.1 Learning neural representations in the card game example

In Fig. S1 we show how the neural representation, projected in the Principal Component space of PCs 1-2, develops throughout learning for the card game task. In Fig. S1a we show the learning progression for the same data as in plot of Fig.1d in the main manuscript. In Fig. S1b we color the same plot by the previous state while in Fig. S1c we color the plot by the action.

These plots show how the grid of states is a prospective grid. This means that the states represented in it are not the states of input of the network but rather states of output. This means that the latent structure extracted by the network is the latent structure of the outputs and not the inputs. These have the same latent structure in terms of lattice ordering but the points that are in proximity are not the ones that are generated with the same observation o_s as input but rather with the same observation as output. This is a critical difference in the predictive learning representation, as we explore further below.

1.2 Analysis of the regularity of representations

In Fig.1 of the main manuscript we show that, even when the underlying network is trained without actions, its representations still develop some regularity, but less than in the case when actions are provided. We here quantify this regularity. To do so we analyse Euclidean distances between the representations of different points in the network trained with and without actions. We compute this as a function of the state distance on the 2d lattice, where nearby states are considered to be at distance 1 while further states follow the Euclidean distance on the lattice. For example starting from a state and taking 2 move right (East) and one up (North) leads to a second state at a distance $\sqrt{5}$ from the original. In Fig. S2a we show the distributions of distances between the representations of all states at distance 2. The representation with actions displays a smaller variance and a higher average. In Fig. S2b we show the scaling of the average norms as a function of the distance between states. We see that the scaling in the network trained with actions appears perfectly linear. The fact that the scaling of distances in the network trained without actions also displays a linear relationship is indicative of the fact

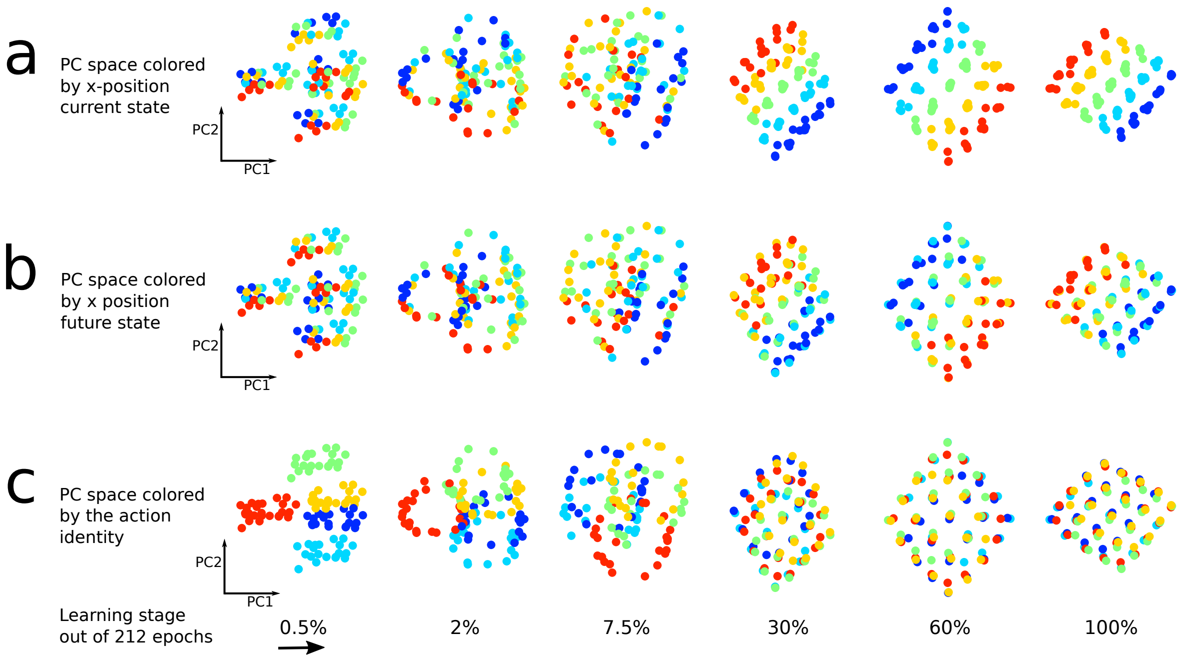

Figure S1. Learning the predictive neural representation a) Principal component space of the neural representation colored by the x-coordinate of the input latent space. b) Principal component space of the neural representation colored by the x-coordinate of the output latent space. c) Principal component space of the neural representation colored by the input action.

that the representation is "partially ordered." The quantification of this partial ordering or *regularity* is given in Fig. S2c where we show the average divided by the standard deviation of the distances between states (these are averages and standard deviations of all distance distributions as in Fig. S2a). We highlight how, in the network with actions, the linear trend is maintained, following from the fact that while the norm increases (Fig. S2b) the standard deviation is fairly constant. By contrast, for the network trained without actions, the standard deviation (the “noise” in this analysis) increases so that the relative increase in the average norm (the "signal") is damped.

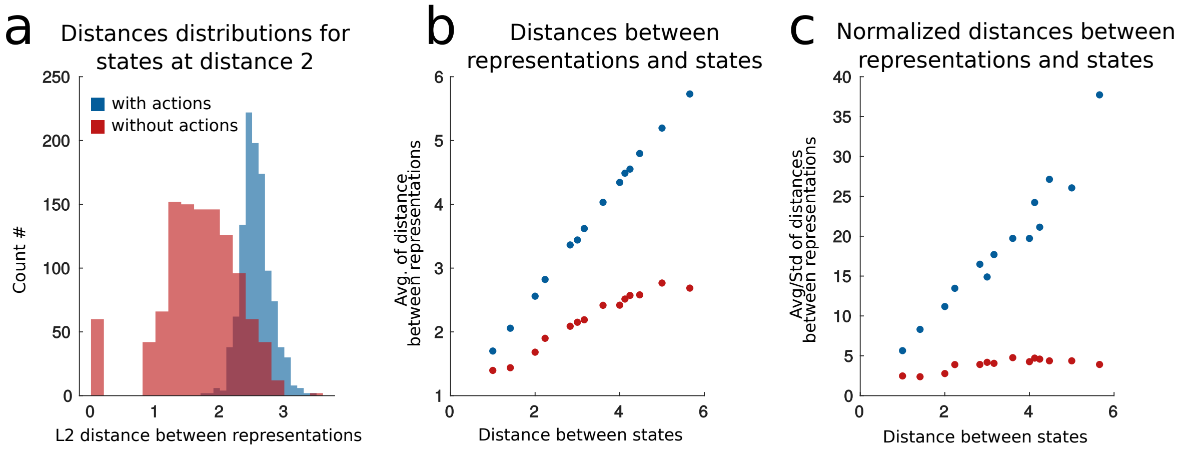

Figure S2. Regularity analysis. a) Distribution of distances for the representation of states at a lattice distance of 2 from one another. b) Average of the distribution of Euclidean distances of neural representations as a function of the distances between the two corresponding states. c) Same as b) but normalized by the standard deviation of each representation, i.e. displaying mean/std.

2 Theoretical analysis of predictive learning and latent space representations

2.1 Low-dimensional neural representation manifolds and how they code latent variables

We begin by defining and characterising the dimensionality of a representation manifold in an idealized, pre-prescribed setting. This is a simplified, concrete model of latent space coding. Low-dimensional (Low-D) representation manifolds occur when a large number of neurons are strongly and consistently tuned to a small set of latent variables. Place and grid cells are examples of such coding [14, 20–22].

In the following, we consider the following specific setting. Given two continuous variables x, y that parametrize a latent space, Fig. S3a, consider an ensemble of N neurons with Gaussian tuning curves that are centered over uniformly distributed locations on the latent space. For example a neuron may be centered at location (x_0, y_0) and have a gaussian radial basis tuning curve as shown in Fig. S3b, $\mathcal{G}_\sigma(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(x-x_0)^2+(y-y_0)^2}{2\sigma^2}\right)$. The responses of an ensemble of N neurons map the latent space manifold (Fig. S3a) to a neural response manifold embedded in neural representation space (that is, the N-dimensional space spanned by the activity of all neurons in the population. To visualize the response manifold, we project it onto its first three Principal Components (PCs), Fig. S3c. As the agent traverses a trajectory $\boldsymbol{x}_t$ in the 2d latent space (Fig. S3a, grayscale), the representation $\boldsymbol{r}_t$ traces out a trajectory on the response manifold (Fig. S3c, grayscale). We can view the tuning curve of a single neuron (Fig. S3b) on the response manifold to obtain the *manifold tuning curve* of this neuron (Fig. S3d), as in Fig. 5 in the main text. In the next section we will analyze in more depth the meaning and properties of manifold tuning curves.

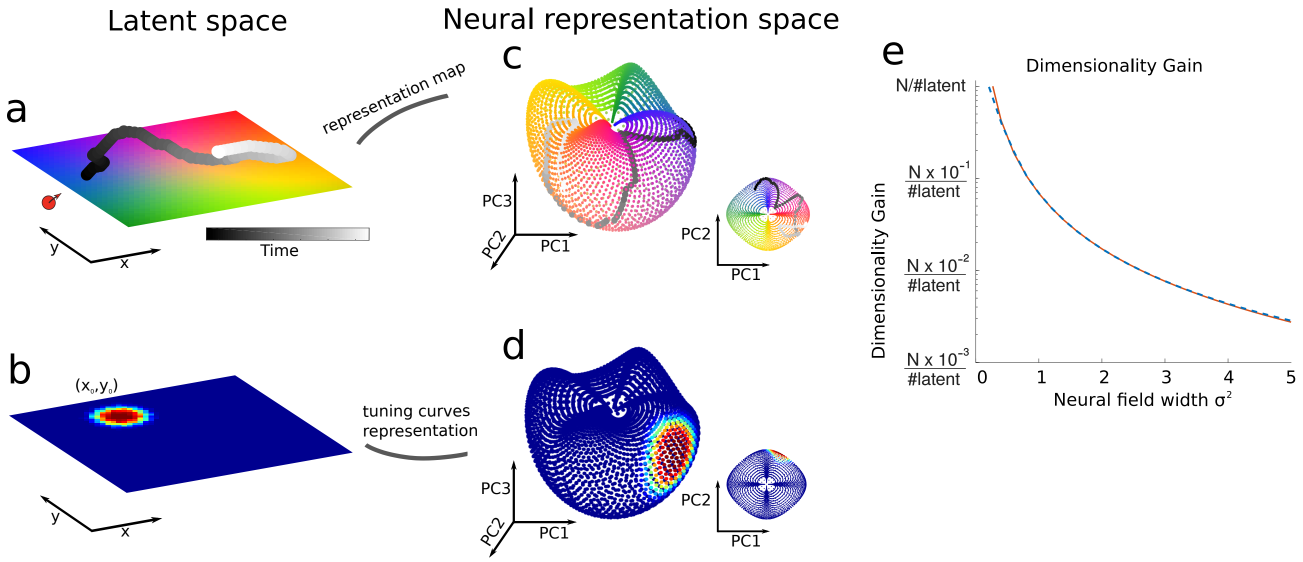

Figure S3. Analysis of neural representation manifolds with pre-prescribed neural tuning. a) Example of a two dimensional environment in which the agent moves. We assign a unique color to each location of the environment. A segment of the agent's trajectory is represented in gray scale, with shade standing for time. b) Example tuning of a neuron with gaussian receptive field centered on (x_0, y_0). c) Neural representation manifold projected onto PCs 1 to 3, under the assumptions that neurons have gaussian receptive fields which uniformly cover the environment and that the agent uniformly explores the environment. Displayed points are uniformly sampled from the manifold. Each point of this representation manifold is colored according to the corresponding location in latent space. The agent's trajectory is represented on the manifold; the inset shows the top view (first two PCs). d) Example of a neural response field on the manifold. The same neuron shown in b) is now shown, with its receptive field with respect to manifold coordinates. e) Dimensionality Gain dependence on the size of the gaussian field σ. The red line represents the DG as computed for 4096 neurons tiling the latent space. The blue dotted line represents the theoretical analysis. In this case DG = PR/2 as the Intrisic Dimensionality ID = 2.

The two dimensions of the latent space completely parametrize the response manifold, resulting in a two-dimensional curved surface. The fact that the representation manifold has two dimensions is revealed by a measure known as Intrinsic Dimensionality (ID), whose formal definition relies on concepts of Riemannian geometry for smooth manifolds [5].

While the ID of the representation manifold is two, due to its curvature, many linear components are necessary to cover it in the N-dimensional neural space. This linear dimensionality can be captured by a second measure of dimensionality: the Participation Ratio (PR) of the manifold. This metric is defined over the eigenvalues $\lambda_{1..N}$ of the covariance matrix $\boldsymbol{C}$ of the neural activity:

$$\mathrm{PR} = \frac{(Tr\boldsymbol{C})^2}{Tr(\boldsymbol{C}^2)} = \frac{(\sum_{i=1}^{N}\lambda_i)^2}{\sum_{i=1}^{N}\lambda_i^2} = \frac{1}{\sum_{i=1}^{N}\tilde{\lambda}_i^2} \tag{1}$$

where $\tilde{\lambda}_i = \lambda_i / \sum_{j=1}^{N}\lambda_j$, see Fig. S4a. [1, 7, 10, 13].

The two most important aspects of these measures of dimension are:

- ID of the representation manifold is determined by the latent variables underlying the inputs. As such, it does not depend on specific details of the neural code.
- PR, by contrast, is a property of the neural code. The more *localized* the neural fields are (i.e. the smaller the response curve width σ is), the more decorrelated the neural activations are, and, in turn, the higher the linear dimensionality PR is.

Thus, the difference between PR and ID carries information about the non-linear embedding of latent variables in the representation. We suggest a novel metric, *Dimensionality Gain* (DG), to capture such difference which measures the extent to which a given representation linearly expands the "true" (i.e. intrinsic) dimensionality of the manifold:

$$\mathrm{DG} = \frac{\textit{linear dimensionality measure}}{\textit{non-linear dimensionality measure}} = \frac{\mathrm{PR}}{\mathrm{ID}}\ . \tag{2}$$

Fig. S3e shows a key observation, that we will return to in the context of predictive representations: that the Dimensionality Gain (DG) increases as the width σ of the neural fields decreases. Thus a higher DG is regarded as a signature of low-D coding. We now give an analytical formula for this relationship as well as a more thorough explanation of relationships among ID, PR, and DG.

2.2 Linear Dimensionality analysis: Participation Ratio and Dimensionality Gain

Participation Ratio is a measure of dimensionality that is based on the distributions of eigenvalues $(\lambda_1, \lambda_2 ...)$ of the covariance matrix $\boldsymbol{C}$:

$$PR = \frac{(\mathrm{Tr}\boldsymbol{C})^2}{\mathrm{Tr}(\boldsymbol{C}^2)} = \frac{(\sum_{i=1}^{N}\lambda_i)^2}{\sum_{i=1}^{N}\lambda_i^2} = \frac{1}{\sum_{i=1}^{N}\tilde{\lambda}_i^2} \tag{3}$$

where $\tilde{\lambda}_i = \lambda_i / \sum_{j=1}^{N}\lambda_j$. In the case of the example of Fig. S3, if we assume that all the locations of the latent space $\mathcal{X}$ are visited with the same probability, then we can compute the covariance matrix of the representation $\boldsymbol{C}$. The entry of the covariance matrix that corresponds to two neurons, i and j, with neural fields centered respectively in position $\boldsymbol{x}_i \equiv (x_i, y_i)$ and $\boldsymbol{x}_j \equiv (x_j, y_j) = \boldsymbol{x}_j + \Delta\boldsymbol{x} = (x_i + \Delta x, y_i + \Delta y)$ and with isotropic variance $\boldsymbol{\sigma}^2 \equiv (\sigma_x^2, \sigma_y^2) = (\sigma^2, \sigma^2)$ is given by:

$$\boldsymbol{C}_{ij} = \frac{1}{T}\int_0^T dt\ (\mathcal{G}_\sigma(\boldsymbol{x}_i - \boldsymbol{x}_t) - \frac{1}{T}\int_0^T \mathcal{G}_\sigma(\boldsymbol{x}_i - \boldsymbol{x}_s)ds)(\mathcal{G}_\sigma(\boldsymbol{x}_j - \boldsymbol{x}_t) - \frac{1}{T}\int_0^T \mathcal{G}_\sigma(\boldsymbol{x}_j - \boldsymbol{x}_s)ds) \tag{4}$$

As each location of the latent space is visited uniformly then this time integral is equivalent to a spatial average over the area A of the latent space $\mathcal{X}$:

$$\begin{aligned}
\boldsymbol{C}_{ij} &= \frac{1}{A}\int_A dt(\mathcal{G}_\sigma(\boldsymbol{x}_i - \boldsymbol{x}_t) - \frac{1}{A})(\mathcal{G}_\sigma(\boldsymbol{x}_j - \boldsymbol{x}_t) - \frac{1}{A}) = \frac{1}{A}\int_A dt\ \mathcal{G}_\sigma(\boldsymbol{x}_i - \boldsymbol{x}_t)\mathcal{G}_\sigma(\boldsymbol{x}_j - \boldsymbol{x}_t) - \frac{1}{A} = \\
&= \frac{1}{4\pi\sigma^2}\frac{1}{A}e^{-\frac{\Delta^2}{4\sigma^2}}\int_A dt\ \mathcal{G}_{\sigma/\sqrt{2}}((\boldsymbol{x}_i + \boldsymbol{x}_j)/2 - \boldsymbol{x}_t) - \frac{1}{A} = \\
&= \frac{1}{4\pi\sigma^2 A}e^{-\frac{\Delta^2}{4\sigma^2}} - \frac{1}{A^2}\ .
\end{aligned} \tag{5}$$

where we recall that $\mathcal{G}_\sigma$ is a Gaussian with variance σ^2 normalized to 1 over the area A. Eq. 5 shows that C_{ij} has a banded structure; in particular it is in Toeplitz form, with entries that decay with the distance between neurons in latent space [7].

We can now compute the terms in Eq. 3 that determine the PR. Specifically by considering the approximation $A \gg 4\pi\sigma^2$ we obtain:

$$\begin{aligned}
(\boldsymbol{C}^2)_{ij} &= \sum_{k=1}^{N} C_{ik}C_{jk} \approx \int_A \mathcal{G}_\sigma(i-k)\mathcal{G}_\sigma(k-j)dk = \\
&= \frac{1}{8\pi^2\sigma^2 A}e^{-\frac{\Delta_{ij}^2}{8\sigma^2}}\ .
\end{aligned} \tag{6}$$

Thus the PR in the limit of large N is:

$$PR = \frac{(Tr\boldsymbol{C})^2}{Tr(\boldsymbol{C}^2)} = \left(\frac{N}{4\pi\sigma^2 A}\right)^2 \frac{8\pi^2\sigma^2 A}{N} = \frac{NA}{2\pi\sigma^2} \,. \tag{7}$$

This shows that the PR dimensionality grows with the inverse of the width of the Gaussian kernel and is proportional to the number of neurons N. Furthermore we also see that it scales as $\frac{A}{s\pi\sigma^2}$ which is the area divided by the width of the field which matches the intuition of the problem.

If all the principal components of neural representations are independent and have equal variance, all the eigenvalues of the covariance matrix have the same value and PR$(\boldsymbol{C}) = N$. Alternatively, if the components are correlated so that the variance is evenly spread across M dimensions, then $\lambda_1 = \lambda_2 = \lambda_3 = ...\lambda_M$ with $\lambda_M > 0$ and $\lambda_m = 0$ for $m > M$ so that the data points are arranged in an M-dimensional subspace of the full N-dimensional space. In this case only M eigenvalues would be nonzero and PR$(\boldsymbol{C}) = M$ (Fig. S4a). For other cases, this measure interpolates between these two regimes. As a rule of thumb, [7] establishes that the PR dimensionality can be thought as the number of dimensions required to explain about 80% of the total population variance in many applications.

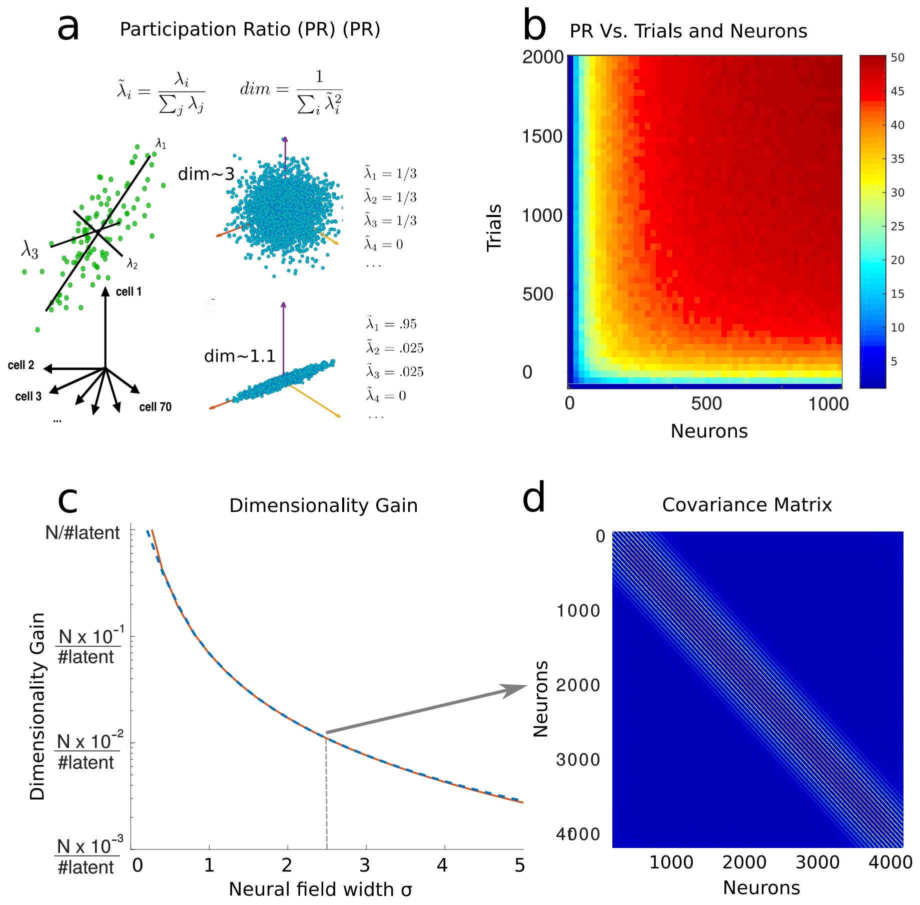

Figure S4. Linear dimensionality analysis. a) Illustration of the Participation Ratio (PR) dimensionality measure. The mathematical expression in terms of the eigenvalues of the covariance is illustrated for a few distributions in PC space. The left part shows an example of point cloud distribution and the leading eigenvalues $\lambda_{1,2,3}$. The right part shows a symmetric spherical distribution with PR=3 and an elongated one with PR=1.1. The eigenvalues of the covariance matrix are shown next to each example. b) PR estimation from a finite number of neurons or trials for the manifold example of Fig. S3 with $\sigma = 2.5$. c) PR dependence on the size of the gaussian field σ^2, same as figure Fig. S3e. The red line represents the DG as computed for 4096 neurons tiling the latent space shown in Fig.2 Main Text. The blue dotted line represents the theoretical analysis. d) Example of the covariance matrix for $\sigma = 2.5$.

2.3 How latent space signal transfer follows from translation-invariant representations of neural states

This section explains the theory behind the results on latent space signal transfer shown in Figs. 3-6 of the main manuscript.

The analysis of the covariance matrix $\boldsymbol{C}$ developed above shows that it is in the Toeplitz form, due to the evenly spaced Gaussian tuning curves (cf. Fig. S4). Specifically, in the case shown in Fig. S4d, it is a Toeplitz tensor because, for each of the two variables, the Toeplitz structure is encoded in the representation as described in Fig. S3. Signal transfer measures the colinearity of the projection of the neural activity on the top eigenvectors of the covariance matrix with the latent variables. In the case analyzed above, x and y are the latent variables and signal transfer measures whether these two variables can be expressed as a linear combination of the projections on the top eigenvectors of the covariance matrix. To see when this is the case we need to compute the eigenvectors of the covariance in terms of x and y. We first restrict our analysis to a nearly Toeplitz matrix in a single variable, Fig. S5a. The eigenvectors of such a Toeplitz matrix have recently been determined to be in a form approaching $\xi_i = a\ cos\left(\frac{\pi k i}{N+1}\right)$ for N large enough, where a is the normalization coefficient and k indicates the kth eigenvector [3,4], Fig. S5b. The eigenvalues are shown in Fig. S5c which displays the relative importance of the first few eigenvectors, Fig. S5d.

The projections on the top eigenvectors are the elements of the representation that most contribute to the value of the signal transfer measure. The top eigenvector is the constant vector $\boldsymbol{n} = \frac{1}{\sqrt{N}}(1, 1, ..1)$. Projecting on this vector is equivalent to taking the average of the representation vector. In neuroscientific terms this would be the average activity or average firing rate across all neurons. The contribution of this eigenvector is subtracted when we consider a mean substracted covariance, the case displayed in the figure is for a Toeplitz matrix with rows normalized to have sum one rather than zero.

The second eigenvector follows the cosine function. Suppose as above that the response of the network to the inputs is similar to the response of a set of Gaussian-bump units responding selectively to the position latent variable x. Projecting the activity of the network onto the second eigenvector approximately returns the position at which the active bump is centered, but shifted by a constant and possibly negated (depending on the sign of the eigenvector). The reason for this is that projecting onto $\boldsymbol{\xi}$ such that $\xi_i = a\ cos\left(\frac{\pi i}{N+1}\right)$ for $i \in [1, N]$, is similar to projecting onto $\xi_i = -\frac{\pi i}{N+1} + 1$, since $\cos(x) \approx -x + 1$ in the interval $[0, \pi]$. Dropping the shift by $+1$, the magnitude of the correlation coefficient between $\cos(x)$ and x in the interval $[0, \pi]$ is also large, equaling $\frac{4\sqrt{6}}{\pi^2} = 0.9927$, Fig. S5d (this is because $\cos(x)$ and x are strongly anti-correlated).

Thus, if we assume a Gaussian response in the activity $f(\boldsymbol{x})$ with the form $f(\boldsymbol{x}) = \mathcal{G}_\sigma(\boldsymbol{x} - \boldsymbol{x}_0)$ around the true location $\boldsymbol{x}_0$ in the latent space $\mathcal{X}$, then the projection over the top eigenvectors of the covariance matrix in Toeplitz form returns a value strongly correlated (in magnitude) with the position x_0 of such gaussian bump that is the latent variable encoded by it. To see this consider the convolution, similar to this projection operation, between a gaussian and a linear variable:

$$\frac{1}{\sqrt{2\pi\sigma}} \int_{-\infty}^{+\infty} \mathcal{G}_\sigma(x-y)x\ dx = \frac{1}{\sqrt{2\pi\sigma}} \int_{-\infty}^{+\infty} e^{-\frac{(x-y)^2}{2\sigma^2}}\ ((x-y)+y)\ dx = y\ . \tag{8}$$

This suggests that projecting over the PCs for a low-D code will lead to recovery of the latent variables. A condition for this to occur is that many cells are tuned to the underlying latent variables.
Now we consider the full case of the network responding to two position variables x and y. The tensoring of multiple variables doesn't affect the argument above as the tensored space will have, as leading eigenvectors, the leading tensored eigenvectors of the individual spaces. The tensored covariance will be in the form:

$$\boldsymbol{C}_{xy} = \boldsymbol{C}_x \otimes \boldsymbol{C}_y$$

where the Kronecker tensor product is denoted by ⊗. Thus, for the case of two variables analyzed in depth in the previous section (Fig. S4), projecting on the first few eigenvectors still serves the role of recovering latent variables. For a deeper analysis and understanding of these phenomena we point the interested reader to more exhaustive reviews [3, 6, 18]. The most important caveat to this analysis is that the spectral properties of the Toeplitz matrix described above depend on the boundary conditions. The case we considered here, where the rows are normalized to sum to one, falls outside the common definition of Toeplitz matrix where the rows are truncated at the boundaries. This latter choice, with different boundary conditions, would lead to eigenvectors of the form $\xi_i = a\ sin\left(\frac{\pi k i}{N+1}\right)$ rather than $\xi_i = a\ cos\left(\frac{\pi k i}{N+1}\right)$, where a is the normalization coefficient and k indicates the kth eigenvector. Thus, in this case the leading eigenvectors would be sine rather than cosine functions. This difference, however, doesn't interfere with the argument we illustrated above, although in this case is necessary to project on multiple eigevectors to reconstruct the latent variable. To this end a Canonical Correlation Analysis between the latent variables and the leading iegenvectors, as we perform in the main text in defining Latent Space Signal Transfer, comes in handy. For example, considring the canonical correlation coefficient between the underlying variable x and the top four eigenvectors as sine functions ($k \in \{1, 2, 3, 4\}$) leads to a correlation coefficient of 0.86.

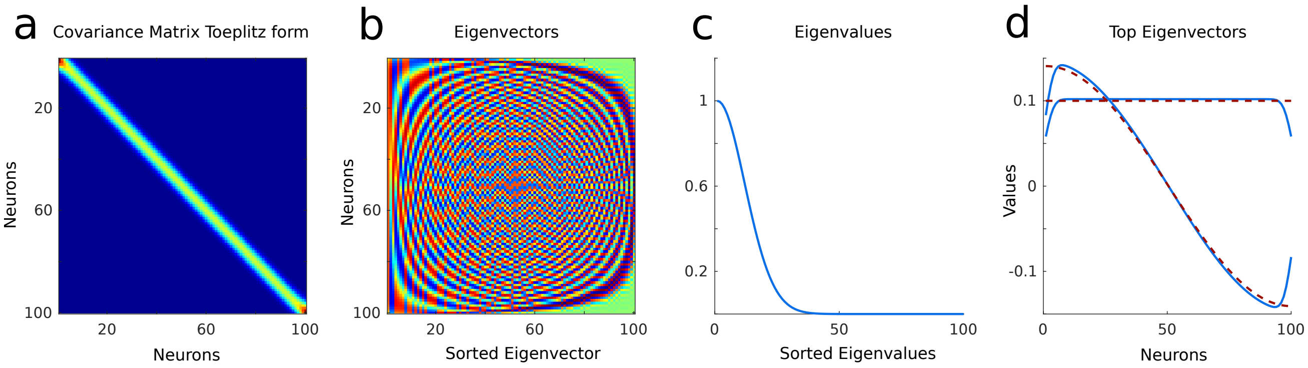

Figure S5. a) Covariance matrix in Toeplitz form. The normalization of the rows (summing up to one) is such that the boundary conditions for this matrix are not exactly in the Toeplitz. b) Sorted eigenvectors of the Toeplitz matrix in a). c) Sorted eigenvalues. d) Top two eigenvectors of the matrix: constant and cosine shaped. Numerical solutions are in blue and theory in red.

2.4 Participation ratio and linear dimensionality

The arguments above imply that predictive representations will have low ID (i.e., low nonlinear dimensionality). We next give reasoning for why such predictive representations develop localized receptive fields. As shown in Fig. S3f, this leads, in turn, to high PR (i.e., high linear dimensionality) and hence high DG, all phenomena that we have observed in our network simulations above.

We begin with the assumption that the low-dimensional predictive representations are a smooth map of the latent space. A consequence is Lipschitz continuity, which guarantees that nearby points in the latent space $(\boldsymbol{x}, \boldsymbol{x'})$ map onto nearby points $(\boldsymbol{r}, \boldsymbol{r'})$ in representation space, at least up to a given radius:

$$d_{\boldsymbol{r},\boldsymbol{r'}} \leq \kappa d_{\boldsymbol{x},\boldsymbol{x'}} \tag{9}$$

where κ is the Lipschitz constant and d indicates distance. This preservation of distances, or similarities – together with the positivity constraint ($r_i \geq 0$ for each neuron i) – is known to lead to localized manifold fields [16, 19]. Interestingly, in our framework this result appears to be true for both positive representations (when the activation function is a sigmoid) and other ones although in such cases the localization of the receptive fields appears to be different and, in general, less localized than in the case where a sigmoid (positive) transfer function is used, Fig. S6.

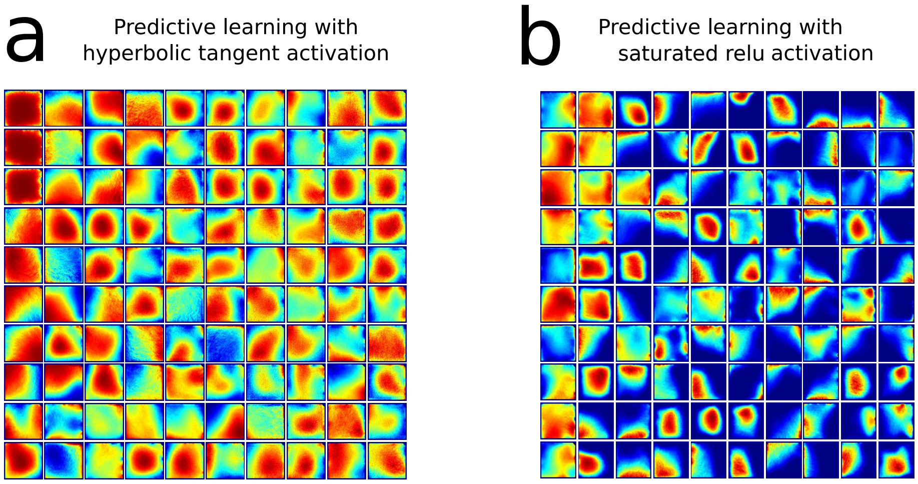

Figure S6. Neural activations comparison across activation functions. a) Average neural activations for a predictive network trained with hyperbolic tangent activation function. b) Same as panel a for a network trained with a hard sigmoid: $f(x) = 0$ if x≤-2.5, $f(x) = 1$ if x≥2.5, $f(x) = 0.2x + 0.5$ otherwise.

The arguments above indicate that predictive learning leads to increases in linear dimensionality, as observed in our learning simulations (Fig.3 main manuscript). But when should this increase stop? A possible answer is: when the linear dimensionality of the neural representation matches that of the outputs that the network is seeking to produce. We give a simplified argument based on linear readout that suggests why this answer might be correct. Rewriting the cost function for a linear readout we obtain $\mathcal{C}_{pred} = \frac{1}{T} \sum_{t=0}^{T-1} ||\boldsymbol{o}_{t+1} - \boldsymbol{y}_t||^2 =$

$\frac{1}{T}\sum_{t=0}^{T-1}||\boldsymbol{o}_{t+1}-\boldsymbol{W}_{out}\boldsymbol{r}_t||^2$, and recognize that (for $\boldsymbol{W}_{out}$ randomly distributed or orthogonal), the linear dimensionality of the representation tends to match the linear dimensionality of the output as they are directly related through the linear transformation $\boldsymbol{W}_{out}$ (cf. [2, 8, 12]). Our numerical studies lend evidence to this: the PR increases through learning until it saturates at about the PR dimensionality of the output, which is 16.2, Fig.3 main manuscript.

2.5 Further considerations on the locality of receptive fields

Consider the case where the movement of the agent in the latent space $\mathcal{X}$ is governed by a discrete-time dynamical system, similar to the case in the main text:

$$\boldsymbol{x}_{t+1} = F(\boldsymbol{x}_t) \tag{10}$$

where $\boldsymbol{x} = (x, y, \theta)$ and $F(\boldsymbol{x})$ is a vector field on $\mathcal{X}$. Above we argued that the recurrent network representation $\boldsymbol{r}_t = f^{RNN}(\boldsymbol{o_t}, \boldsymbol{r}_{t-1})$ through learning becomes a direct function of the latent space $\mathcal{X}$ as predictive learning extracts the latent variables: $\boldsymbol{r}_t = f(\boldsymbol{x_t})$. We now ask the question of whether this representation has localized neural activity.

Considering the local expansion at second order around a point $\boldsymbol{x}^* \in \mathcal{X}$ we obtain:

$$f(\boldsymbol{x}^*) - f(\boldsymbol{x}) = f(\boldsymbol{x}^*) + D_f(\boldsymbol{x}^*)\cdot(\boldsymbol{x}-\boldsymbol{x}^*) + (\boldsymbol{x}-\boldsymbol{x}^*)\cdot H_f(\boldsymbol{x}^*)\cdot(\boldsymbol{x}-\boldsymbol{x}^*) + ... \tag{11}$$

where D_f and H_f are respectively the Jacobian and Hessian. Assuming that the function f is Lipschitz continous then:

$$d_{\boldsymbol{r}^*,\boldsymbol{r}} = ||f(\boldsymbol{x}^*) - f(\boldsymbol{x})|| \leq \kappa_m||\boldsymbol{x}-\boldsymbol{x}^*|| \ , \tag{12}$$

where κ_m is the Lipschitz constant. Furthermore if the inverse is Lipschitz, as expected if the representation manifold is smooth, then we have the bi-Lipschitz property:

$$\kappa_l d_{\boldsymbol{x},\boldsymbol{x}^*} \leq d_{\boldsymbol{r}^*,\boldsymbol{r}} = ||f(\boldsymbol{x}^*) - f(\boldsymbol{x})|| \leq \kappa_m d_{\boldsymbol{x},\boldsymbol{x}^*} \ . \tag{13}$$

These bounds suggest that local similarities in latent space $\mathcal{X}$ translate in local similarities in representation space $\mathcal{R}$. Furthermore, depending on the order of the Taylor series which dominates the local expansion of the function $f(\boldsymbol{x})$, we obtain a stronger form of Lipschitz continuity – Holder continuity:

$$\kappa_l d^{\alpha_l}_{\boldsymbol{x},\boldsymbol{x}^*} \leq d_{\boldsymbol{r}^*,\boldsymbol{r}} = ||f(\boldsymbol{x}^*) - f(\boldsymbol{x})|| \leq \kappa_m d^{\alpha_m}_{\boldsymbol{x},\boldsymbol{x}^*} \ . \tag{14}$$

These relationships control how representations of similar latent variables map onto similarities in the representation space, up to a certain radius. As latent variables become more and more distant, the corresponding representations tend to orthogonalize:

$$d^2_{\boldsymbol{x},\boldsymbol{x}^*} = ||\boldsymbol{x}-\boldsymbol{x}^*||^2 = ||\boldsymbol{x}||^2 + ||\boldsymbol{x}||^2 - 2\langle\boldsymbol{x},\boldsymbol{x}^*\rangle \ , \tag{15}$$

which shows that as the scalar product $\langle\boldsymbol{x},\boldsymbol{x}^*\rangle$ increases, the distance $d_{\boldsymbol{x},\boldsymbol{x}^*}$ decreases. On a spherical surface, where the norm $||\boldsymbol{x}||$ of each point is equal, the scalar product is in 1-1 correspondence with the distance.

An example of a code which varies continuously locally but orthogonalizes globally is a representation with localized gaussian fields, cf. Fig.2a-d in the main text. This phenomenon has been studied, with the extra condition of the representation being positive ($\boldsymbol{r} \geq 0$) in [19] where the authors show that preserving local similarities with a positivity constraint builds a representation whose receptive fields tile the representation manifold.

In sum, the arguments above indicate why activity on the representation manifold becomes localized in terms of the latent variables $\mathbf{x}$.

We close by emphasizing that the representations produced by the underlying neural networks will also be local in time. For example, consider a Wiener process in the latent space. If $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \boldsymbol{\xi}$ and $\boldsymbol{\xi}$ is isotropically i.i.d. according to a Gaussian distribution $\mathcal{G}(0, \sigma^{\mathcal{X}})$ for each coordinate, then we obtain the relations:

$$d_{\boldsymbol{x(t)},\boldsymbol{x(t^*)}} = ||\boldsymbol{x}(t^*) - \boldsymbol{x}(t)|| = d^{\mathcal{X}}\sigma^{\mathcal{X}}\sqrt{t^*-t} \ , \tag{16}$$

where $d^{\mathcal{X}}$ is the dimensionality of the latent space. Such relations lead to

$$\kappa_l d^{\alpha_l}_{\boldsymbol{x},\boldsymbol{x}^*} = \kappa_l(d^{\mathcal{X}}\sigma^{\mathcal{X}}\sqrt{t^*-t})^{\alpha_l} \leq d_{\boldsymbol{r}^*,\boldsymbol{r}} = ||f(\boldsymbol{x}(t^*)) - f(\boldsymbol{x}(t))|| \leq \kappa_m d^{\alpha_m}_{\boldsymbol{x},\boldsymbol{x}^*} = \kappa_m(d^{\mathcal{X}}\sigma^{\mathcal{X}}\sqrt{t^*-t})^{\alpha_m} \ . \tag{17}$$

This equation highlights how similarities scale with time. They also scale with the dimensionality of the representation manifold $d^{\mathcal{R}}$, so that considering the effective random dynamics induced on it, we have:

$$d_{\boldsymbol{r_t},\boldsymbol{r_{t^*}}} \geq d^{\mathcal{R}}\sigma^{\mathcal{R}}\sqrt{t^*-t} \ . \tag{18}$$

Here $\sigma^{\mathcal{R}}$ denotes the average variance, per dimension, of the induced Wiener process in representation space. As the dimensionality of the manifold $d^{\mathcal{R}}$ decreases then the bounds become tighter and the similarity between neighbouring points increases. These considerations will drive future research aimed at fully describing how similarities explored dynamically across time lead to the learning of similarities across space on the representation manifold.

3 Control studies: Numerical simulations

3.1 Robustness of our findings: comparing results for multiple tasks and network structures

Several controls are required to assess that our findings are robust to the structure of the RNN, are robust to its input statistics, are robust to other modeling assumptions, and continue to depend on the task being predictive. We describe here a set of controls for the spatial exploration task.

Each control model is trained for a total of 200 epochs, enough for all models to converge. Our focus is not on optimizing performance and therefore we do not employ an Early Stopping Rule here, although we reduce the learning rate on plateau when the validation loss doesn't decrease for more than 10 epochs. We first describe the overall control analysis and detail later the individual models. The key difference for the models is given in their respective names, where we use the abbreviation 'w' for with and 'wo' for without. For example 'wo distance information' refers to the same predictive model trained without distances from the walls in its observations. The models are sorted into three categories: predictive models, non-predictive models and predictive models with critical modifications. These last ones include modifications to the network that differ from the architecture of the main model presented. Some of these have minor differences from the original framework (e.g. adding a sparsity constraint) while others are critically different, like in the case of predicting the previous step (the past instead of the future).

We show how for these models the metrics introduced in the main manuscript - predictive error, latent signal transfer and dimensionality analysis (DG) - capture differences across these models. In Fig. S7 we show how the models converge in their cost function through learning, Fig. S7a. We then show the predictive error symmetry, Fig. S7b based on the position of the axis of symmetry for the predictive error. For a network trained to predict the next step, this should be roughly 1 while for a network trained to predict the previous step, this should be roughly -1. In Fig. S7c we show the linear regression coefficient for a linear regression between the hidden representations of these networks and the spatial variables x, y. The linear decoding of position is the average of the two regressors for coordinate x and y. Models linearly encode for position in their hidden representation have a linear decoding measure closer to one. The results of these controls are in line with those in Fig. 3 of the main manuscript.

Following the same analysis presented in the main manuscript (Fir. 3) we next analyse latent signal transfer, performing a Canonical Correlation Analysis between the position variables and the top 3 Principal Components of the hidden representation for every epoch, Fig. S8a. The same analysis is repeated for the observation signal in Fig. S8b. In predictive models, while the former grows through learning, the latter declines – indicating that the top Principal Components in the hidden representation represent the position (latent signal) rather than observations.

Finally we analyze dimensionality trends across learning for both linear and nonlinear dimensionality measures. Fig. S9a shows the linear dimensionality (PR) across learning while Fig. S9b shows the average of nonlinear dimensionality measures. The trends of predictive and nonpredictive models are highlighted with brackets and generally agree with the trends pointed out in the main manuscript. It is clear that the variability across models is high: these metrics can be affected by several different factors. For example, enforcing sparsity - which can be achieved in several different ways - may modify the dimensionality of the representations. Finally in Fig. S9c we show the dimensionality gain, being the ratio between linear (Fig. S9a) and nonlinear (Fig. S9b dimensionalities.

Having analyzed the metrics introduced in the main manuscript in Figs. S7 to S9 we then turn to the question of place cell coding. As highlighted in the main manuscript the emergence of place cell activation is a possible way to explain and interpret the trends in the metrics established this far. In Fig. S10 we show the place selectivity in the neural activities of 100 neurons across all models. The 100 neurons are sorted to be the 100 neurons with maximum average activity. From this figure it appears that all predictive models develop some form of localized activations while non-predictive models do not. We also aimed at capturing the overall statistics across all cells for their sparsity. We analyzed two forms of sparsity: temporal sparsity and spatial sparsity. For temporal sparsity (Fig. S11a) we compute the average across time of the total activation (L1 norm of activity population vector, given positive activity). For spatial sparsity we compute the average activations of neurons, once such activations have been averaged over space (Fig. S11b). This is the average, for each neuron, of the values shown in Fig. S11a. We also show in Fig. S12 four examples of how different hidden representations appear in PC space. Here the top three Principal Components of the hidden representation are colored by the x-position of the agent in the environment, similar to Fig. 5 in the main text.

Overall the results here displayed confirm the principal results presented in the main manuscript. They also introduce several nuances and avenues for interesting future study. We provide the code to generate these models and analyses.

We now explain the details of the 14 models we compared above. Each model lists only the differences from the original one, which we refer to as "predictive learning."

Predictive networks

- Noise in RNN activations. In this model gaussian noise with std of 0.1 is added on top of the activations of every unit at each step.
- Predictive learning with input noise. We add noise to the input as a control that the phenomena describe are not dependent on the absence of noise or on overfitting. We add time independent zero mean gaussian noise to each input channel with an amplitude σ^2_{noise} which is 10% of the total variance of the channel: $\sigma_{noise} = 0.1\sigma_{channel}$ for all channels. The model shows the same signatures of predictive learning.
- GRU. In this model the network, instead of being a recurrent vanilla network has units which are GRU.
- LSTM. In this model the network, instead of being a recurrent vanilla network has units which are LSTM.

Non-predictive networks

- Autoencoder without bottleneck. In place of the RNN we use a feedforward layer of size 200 and we train the model not on predicting the upcoming observations but rather on replicating them (autoencoding framework). This model can be trained, but it doesn't display all the phenomena highlighted in the main text. The linear dimensionality increases and the intrinsic dimensionality decreases but the latent variables do not seem to be extracted as in the predictive case. Both CCA metrics fail to show the extraction of latent variables and place cell tuning curves do not appear.
- Autoencoder with bottleneck. In place of the RNN we use a feedforward layer of size 10.
- Non-predictive, recurrent autoencoder. This model, as discussed in the main text, has the same structure of the predictive learning one but is trained in replicating (autoencoding) the input observations.

Other models

- Sparsity, predictive learning with a sparsity constraint. We add a L1 sparsity constraint with a penalty of $5 \ 10^{-8}$ on the activations of the recurrent network. This constraint doesn't appear to sparsify the network in a straightfoward way. Rather it seems to strongly reduce the overall activity and introduce a code where some units tend to be more active than others overall. This is a signature of a less distributed neural code.
- Predictive learning without actions. In this case actions are not fed as input to the network but the network is still trained to reproduce both distance and color information. The task is more difficult but the network still seems to be able to extract a representation with similar features to the full case, as long as it is trained to perform predictive learning.
- Predictive learning without color information. We train the same predictive learning model without color information from the sensors in the input and output: color information is not passed as input and is not decoded from the output. Sensors receive only distance information. The model minimizes the cost function but it doesn't display the features analyzed in the main text.
- Predictive learning without distance information: same as above but without distance information. The model seems to learn with similar characteristics, showing robustness to the lack of distance information. This is an important feature as one may say that having precise, "hard-coded" distance information in the sensors is not biological. In the main text we study the case of both distance and color information to include reasonably available visual information, but the present control is important to highlight the robustness of our results.
- Autoencoder with angle. We train a network to autoencode its observations where to the observations the current angle of the agent is added. The model didn't train particularly well across several repetitions we tried; it is included as an example of model which fails to train in outputting the angle, as compared to other autoencoding models listed above.
- Predictive learning on the previous timestep. We train the same model but to predict the previous time step in time rather than the future one. This model doesn't extract the latent space.

4 Pilot analysis of neural data

Here we run two preliminary data analysis on both hippocampal and motor cortical neural activity to directly link our findings to the analysis of neurophysiology data.

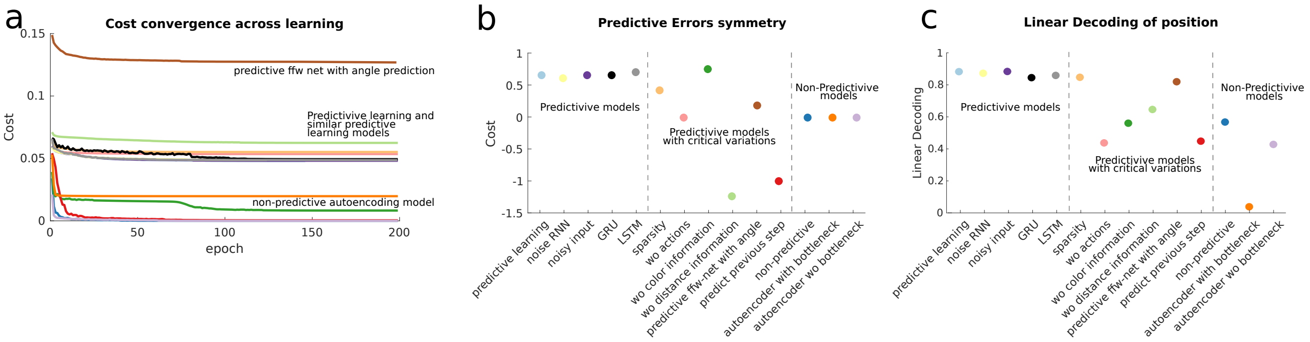

Figure S7. Cost and Predictive Error metrics. a) Cost convergence across all models. b) Predictive error symmetry axis position upon learning across all models. This is the same measure used for Fig. 4 main manuscript. c) Linear Decoding performance of the latent variables, from the network representation.

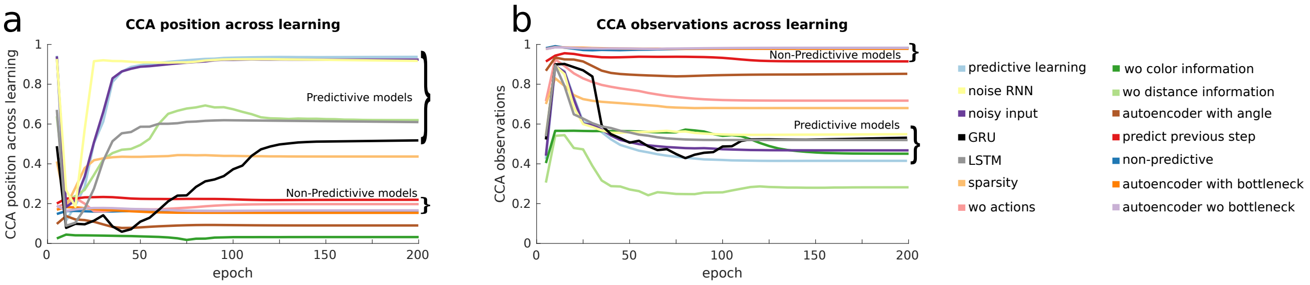

Figure S8. Canonical Correlation Analysis. a) Canonical correlation analisys through learning between the top 3 PCs of the hidden representation and the position of the agent in (x,y) coordinates. The average of the two canonical correlations found by the analysis is plotted for each epoch. b) Same as panel a but between the hidden representation and the top 3 PCs of the observations.

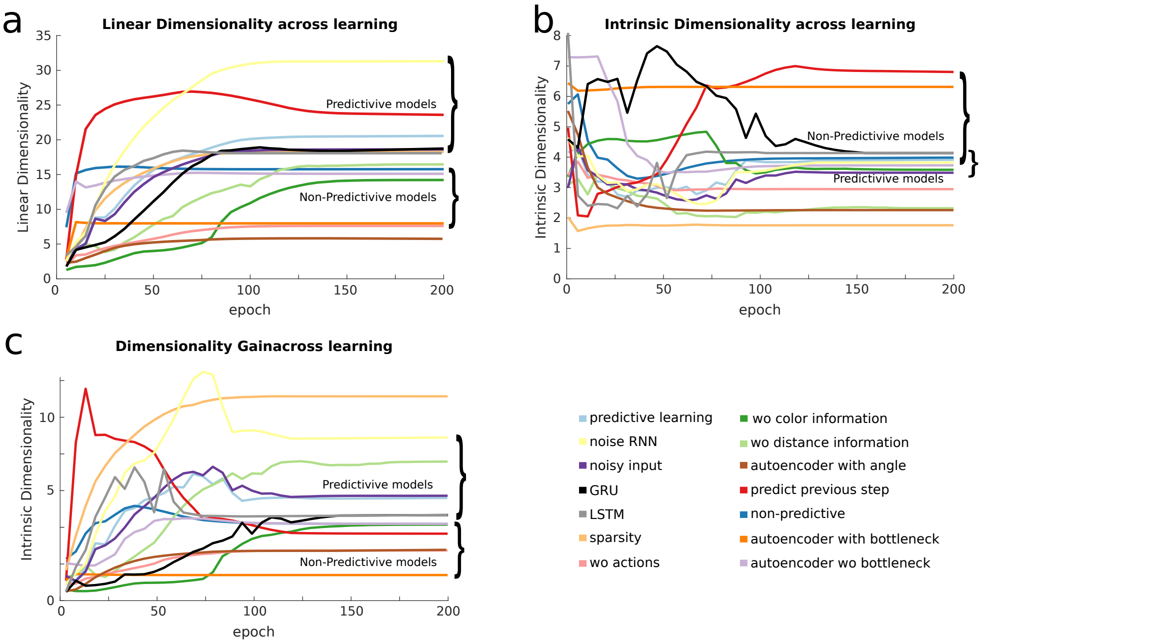

Figure S9. Dimensionality analysis. a) Linear dimensionality (PR) across learning. b) Non-linear dimensionality (ID) across learning. Each curve is the average of the 4 ID estimation methods introduced in the main manuscript, cf. Fig. 3. c) Dimensionality Gain across learning.

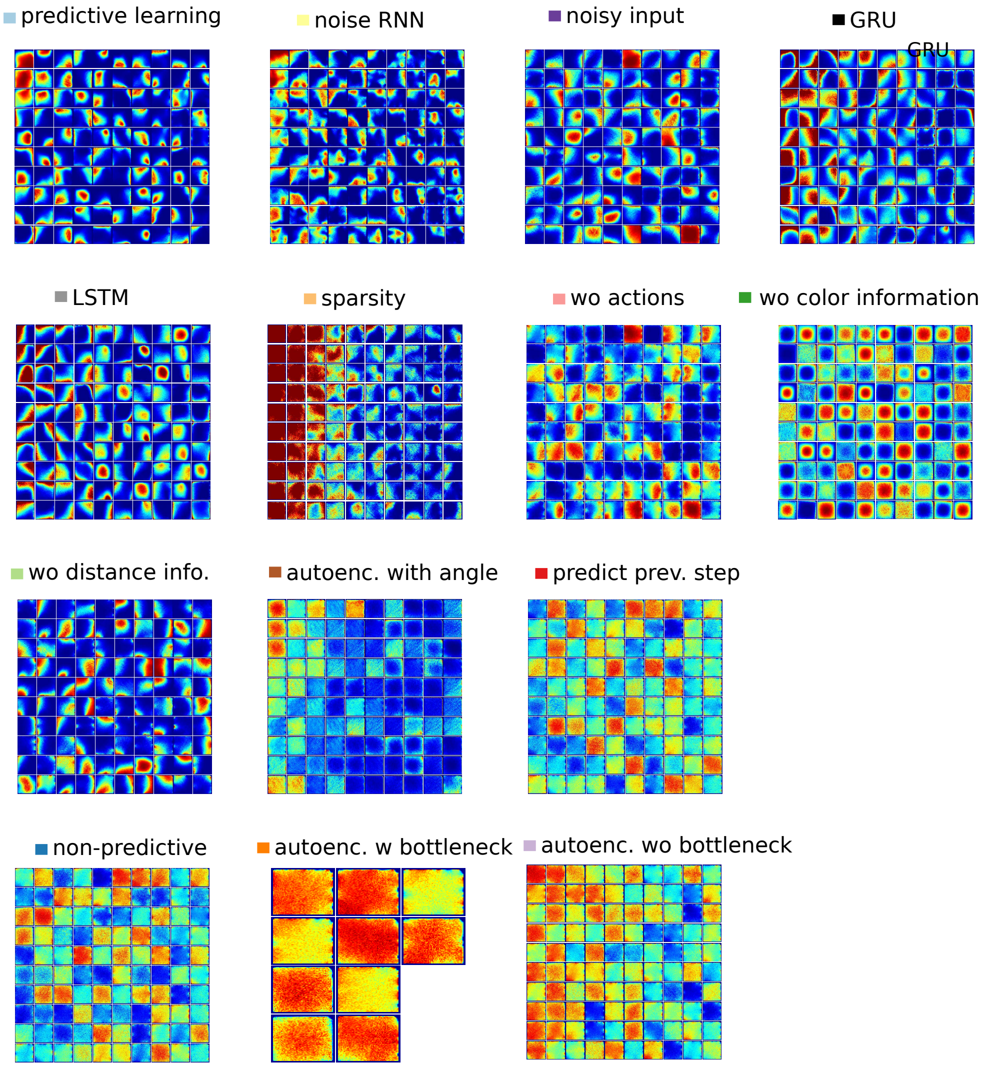

Figure S10. Average activity of 100 neurons in the latent (environment) space across all models.

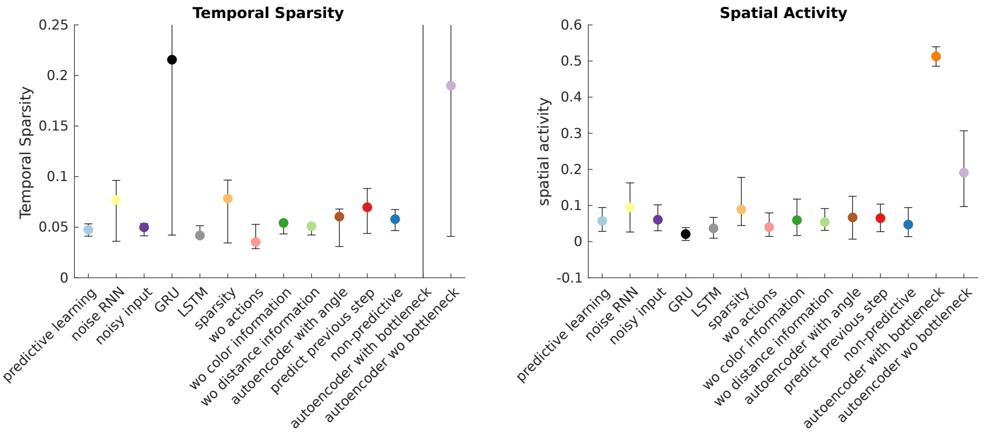

Figure S11. Sparsity analysis. a) Temporal sparsity. Average L1 norm of the population vector across time. For each time step the L1 norm of the activity of all neurons is computed and the mean and standard deviation of the distribution of such sparsity measure are displayed. b) Spatial sparsity. We compute the L1 norm of the spatial averages of individual neurons. These are the ones plotted in Fig. S10. The average and standard deviation of the distributions of L1 norms are used for the plot across all models.

4.1 Hippocampal recordings during spatial navigation: neural data reveal partial evidence of predictive learning.

We analyze a publicly available neural dataset [15], collected in the Buszaki lab, consisting of recordings from the hippocampal area CA1. In the analyzed session I15, rat i01 performed free exploration of an open square

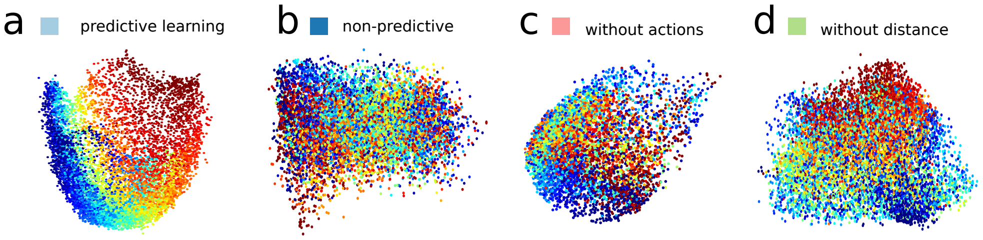

Figure S12. Principal Component Space. a) We show the hidden representation in PCs1-2 colored by the x coordinate of the latent space (environment). This is the same plot as in Fig. 5 of the main manuscript. b) Same as panel a for the non-predictive model. c) Same as panel a for the predictive model without actions. d) Same as panel a for the predictive model without distance information.

environment for about 60min. Over 102 channels 164 CA1 neurons were recorded and identified. We didn't preprocess the data except for binning spikes into a moving window of 100ms (we repeated the procedure for a moving window of 50ms and obtained similar results).

First, we decoded the future and past position and head direction of the animal from the neural population at the current time, as a function of the time difference Delta t. Fig. S13a shows that the decoding of the spatial coordinates, but not the angle, appeared to be prospective in time by about 100ms. This result is in line with our findings regarding predictive error and decoding of latent variables from the representation, cf. Figs. 3a,b.

We then fit, by means of a quadratic Generalized Linear Model, receptive fields to each neuron. We repeated the same procedure both in the spatial domain of the environmental coordinates and in the Principal Component space spanned by the first two PCs. We measured the size of the field by the negative exponent of the fit constant (exponential decay of the field). A higher exponent indicates a faster decay and thus a sparser code, Fig. S13b. Neural receptive fields fit to spatial (blue) and PCs coordinates (red) have a generally similar form. This result is in line with our analysis developed in the main manuscript Fig. 4. There we showed how, in our simulations, single neurons developed localized receptive fields on the neural population (PC) manifold.

Finally, we tested different measures of intrinsic dimensionality on the neural data, in Fig. S13c. We reported only measures which displayed numerical stability. Interestingly, several measures which appeared very stable in simulations seemed unstable on neural data. This could be due to the lack of data (e.g. the green curve in Fig. S13c appears numerically stable when at least 40 neurons are used for the computation), or to the intrinsic noise of neural data (which was not modeled in our simulations). This suggests that careful future analyses are needed to understand the problem of estimating the dimensionality of neural data, a topic which recent work suggests could be of crucial importance to understand hippocampal coding [11].

We close by pointing out that the metrics we developed (latent signal transfer and dimensionality analysis) are mainly geared towards understanding learning and the formation of manifold structures through the learning process (Fig. 3 main manuscript). We thus look forward to future analysis of datasets with more neurons and to attendant tests of how our methods may reveal how the geometrical properties of neural representations evolve through task learning.

4.2 Motor Cortex recordings during virtual target reaching task.

We analyze a publicly available neural dataset [9, 17], collected in the Miller lab, consisting of recordings from the Primary Motor Cortex (M1). In the analyzed session (session n.1), a monkey controlled an on-screen cursor being rewarded for moving it to an indicated reach target. Multiple targets were presented during each trial. The kinematic demands of the task were minimal (e.g., very brief hold times), so that the monkey typically completed the task with a smooth sequence of reaches. The position, velocity, and acceleration of the cursor were recorded while electrophysiological recordings were collected with Utah multielectrode arrays yielding 97 neurons in M1 for session N.1.

Similarly to the analysis performed on hippocampal data, we binned neural activity every 100ms to obtain spike counts vectors on which we performed a similar analysis to the one just described and performed on hippocampal data. We first sought to identify whether behavioral variables were encoded in the neural activity, Fig. S14a. All behavioral variables appeared to have a decoding lag, quantified by the symmetry axis of the decoding curve, skewed towards the future in the range of 100-300ms. This can be interpreted as a signature that M1 neural activity encodes for the upcoming movements of the cursor.

We then attempted to characterize neural receptive fields on the behavioral latent spaces (position, velocity, accelaration) and principal component space of the neural activity. Given the differences between experimental paradigms (cursor moving vs arm reaching movements in predictive learning simulations) we opted for showing raw-data rather than fitted receptive fields as in Fig. S13b. In Fig. S14b we display the average neural activity projected on the spaces of the cursor coordinates, cursor speed and top two principal components of the neural

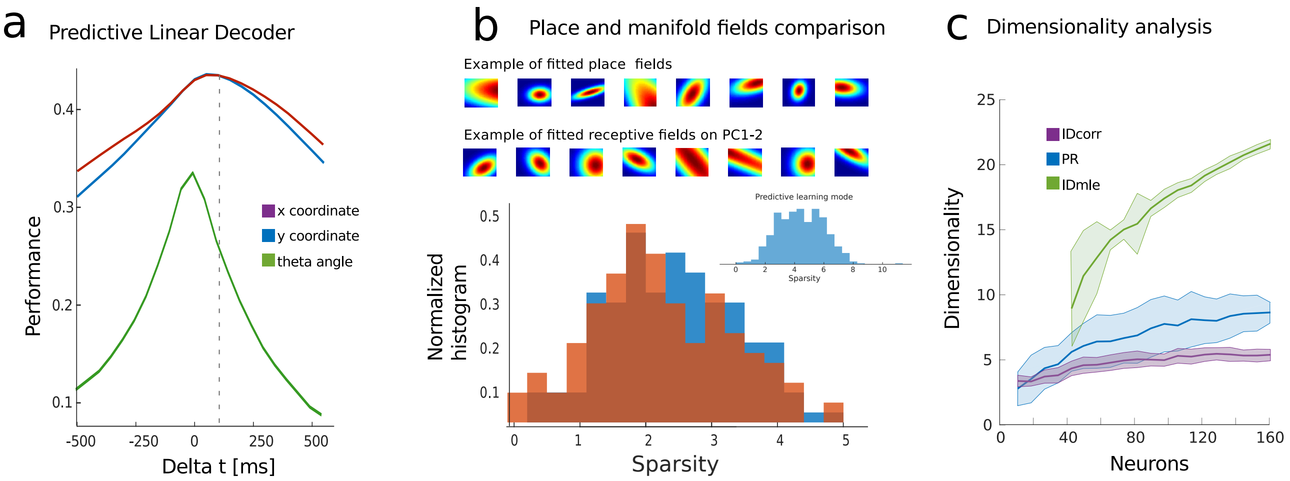

Figure S13. Neural data analysis of Hippocampal data. a) Linear decoding of latent variables from neural population data. b) Comparison between localization of receptive fields in the latent space vs Principal Components space. Top example of localization of extracted receptive fields (quadratic GLM model) on both latent space variables (x,y) and PCs 1-2. Bottom comparison of the extracted tuning in the two cases (red for fitted place fields and blue for fitted fields on PCs). Inset: distribution for the predictive learning model. c) Three measures of dimensionality estimation applied to neural data.

population activity. Further analysis with more neural statistics and careful extraction of receptive field tuning is due to understand the similarity and differences of the tuning of individual neurons over these different spaces. Finally we characterized the dimensionality of the neural activity manifold, Fig. S14c similarly to the case of the hippocampus.

Altogether the presented analysis shows a way of characterizing neural activity in M1 which has the potential to both enable comparative characterizations across brain areas (hippocampus and motor cortex) and with different learning algorithms, e.g. predictive learning. Surely a wider and detailed data analysis is due to yield such consistent characterization. These pilot analyses have the limited scope to allow building tools and intuition for identifying similarities and differences across both neural recordings and between such recordings and learning simulations.

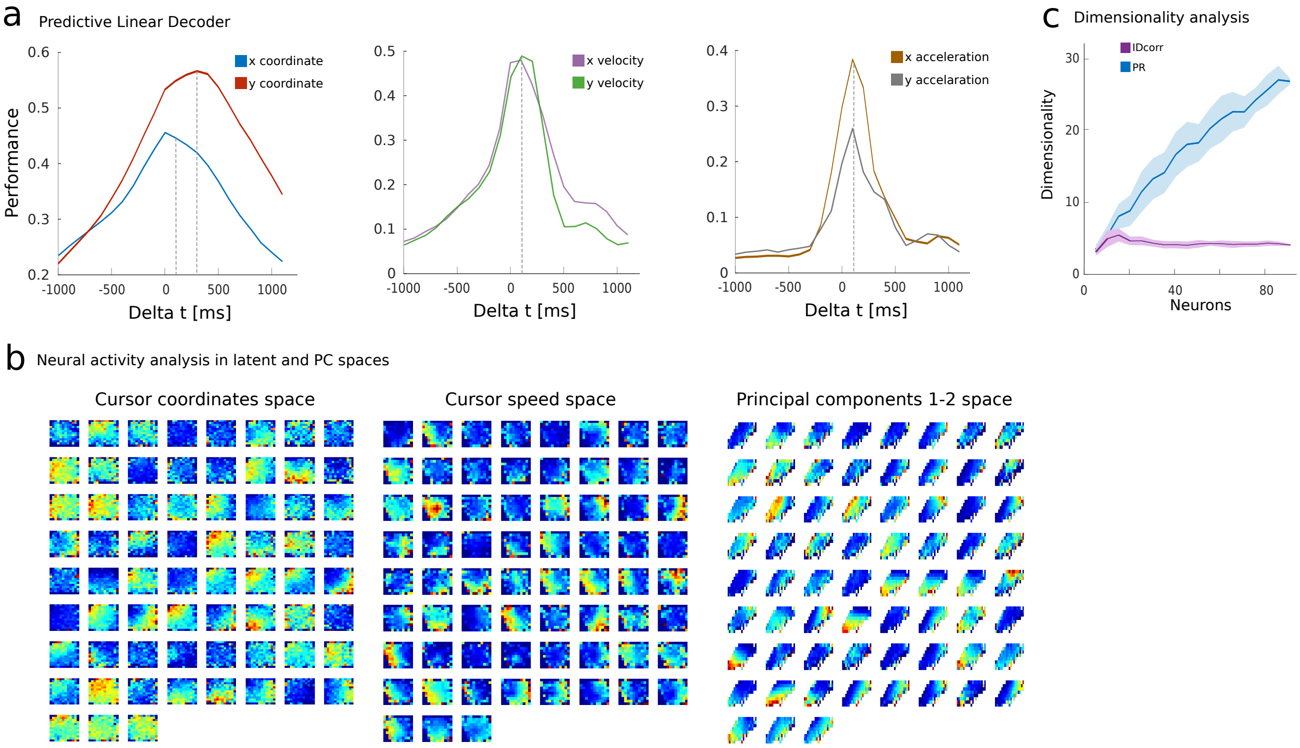

Figure S14. Neural data analysis of Motor Cortical data. a) Linear decoding of latent variables from neural population data. b) Comparison between localization of receptive fields in the latent space vs Principal Components space. (Left) Average activity of individual neurons on the space spanned by the cursor (x,y cursor coordinates respectively). (Center) Same as left panel for cursor velocity along x and y axis. (Right) Average activity of indivudal neurons in the space of population activity as spanned by the top two principal components of the same. c) Two measures of dimensionality estimation applied to neural data.

References

1. L. F. Abbott, K. Rajan, and H. Sompolinsky, *Interactions between intrinsic and stimulus-evoked activity in recurrent neural networks*, The dynamic brain: an exploration of neuronal variability and its functional significance, (2011), pp. 1–16.

2. A. Agarwal, A. Anandkumar, P. Jain, P. Netrapalli, and R. Tandon, *Learning sparsely used overcomplete dictionaries*, in Conference on Learning Theory, 2014, pp. 123–137.

3. A. Böttcher, J. M. Bogoya, S. Grudsky, and E. A. Maximenko, *Asymptotics of eigenvalues and eigenvectors of toeplitz matrices*, Sbornik: Mathematics, 208 (2017), p. 1578.

4. A. Böttcher, S. M. Grudsky, and E. A. Maksimenko, *On the structure of the eigenvectors of large hermitian toeplitz band matrices*, in Recent Trends in Toeplitz and Pseudodifferential Operators, Springer, 2010, pp. 15–36.

5. F. Camastra and A. Staiano, *Intrinsic dimension estimation: Advances and open problems*, Information Sciences, 328 (2016), pp. 26–41.

6. H. Dai, Z. Geary, and L. P. Kadanoff, *Asymptotics of eigenvalues and eigenvectors of toeplitz matrices*, Journal of Statistical Mechanics: Theory and Experiment, 2009 (2009), p. P05012.

7. P. Gao, E. Trautmann, B. M. Yu, G. Santhanam, S. Ryu, K. Shenoy, and S. Ganguli, *A theory of multineuronal dimensionality, dynamics and measurement*, bioRxiv, (2017), p. 214262.

8. S. Lahiri, P. Gao, and S. Ganguli, *Random projections of random manifolds*, arXiv preprint arXiv:1607.04331, (2016).

9. P. N. Lawlor, M. G. Perich, L. E. Miller, and K. P. Kording, *Linear-nonlinear-time-warp-poisson models of neural activity*, Journal of computational neuroscience, 45 (2018), pp. 173–191.

10. A. Litwin-Kumar, K. D. Harris, R. Axel, H. Sompolinsky, and L. F. Abbott, *Optimal Degrees of Synaptic Connectivity*, Neuron, 93 (2017), pp. 1153–1164.e7.

11. R. J. Low, S. Lewallen, D. Aronov, R. Nevers, and D. W. Tank, *Probing variability in a cognitive map using manifold inference from neural dynamics*, bioRxiv, (2018).

12. J. Mairal, F. Bach, J. Ponce, and G. Sapiro, *Online dictionary learning for sparse coding*, in Proceedings of the 26th annual international conference on machine learning, ACM, 2009, pp. 689–696.

13. L. Mazzucato, A. Fontanini, and G. La Camera, *Stimuli Reduce the Dimensionality of Cortical Activity*, Frontiers in Systems Neuroscience, 10 (2016).

14. J. O'Keefe and J. Dostrovsky, *The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat.*, Brain Res, 34 (1971), pp. 171–175.

15. E. Pastalkova, Y. Wang, K. Mizuseki, and G. Buzsáki, *Simultaneous extracellular recordings from left and right hippocampal areas ca1 and right entorhinal cortex from a rat performing a left/right alternation task and other behaviors*, CRCNS, (2015).

16. C. Pehlevan, A. M. Sengupta, and D. B. Chklovskii, *Why do similarity matching objectives lead to hebbian/anti-hebbian networks?*, Neural computation, 30 (2018), pp. 84–124.

17. M. G. Perich, P. N. Lawlor, K. P. Kording, and L. E. Miller, *Extracellular neural recordings from macaque primary and dorsal premotor motor cortex during a sequential reaching task.*, CNRS.org, (2018).

18. M. Rezghi and L. Elden, *Diagonalization of tensors with circulant structure*, Linear Algebra and its Applications, 435 (2011), pp. 422–447.

19. A. Sengupta, M. Tepper, C. Pehlevan, A. Genkin, and D. Chklovskii, *Manifold-tiling localized receptive fields are optimal in similarity-preserving neural networks*, bioRxiv, (2018).

20. T. Solstad, C. N. Boccara, E. Kropff, M.-B. Moser, and E. I. Moser, *Representation of Geometric Borders in the Entorhinal Cortex*, Science, 322 (2008), pp. 1865–1868. WOS:000261799400061.

21. H. Stensola, T. Stensola, T. Solstad, K. Froland, M.-B. Moser, and E. I. Moser, *The entorhinal grid map is discretized*, Nature, 492 (2012), pp. 72–78. WOS:000311893400047.

22. T. J. Wills, F. Cacucci, N. Burgess, and J. O'Keefe, *Development of the Hippocampal Cognitive Map in Preweanling Rats*, Science, 328 (2010), pp. 1573–1576. WOS:000278859200051.