Predictive learning extracts latent space representations from sensory observations

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

[1]University of Washington Center for Computational Neuroscience and Swartz Center for Theroetical Neuroscience; Seattle, WA
[2]Department of Applied Mathematics, University of Washington; Seattle, WA
[3]Department of Mathematics and Statistics, Université de Montréal; Montreal, Canada
[4]Mila - Quebec Artificial Intelligence Institute; Montreal, Canada
[5]Group for Neural Theory, Ecole Normal Superieur; Paris, France
[6]IBM Research AI; Yorktown Heights, NY
[7]Allen Institute for Brain Science; Seattle, WA
[+]These authors share senior authorship
[*]Corresponding author stefanor@uw.edu

Neural networks have achieved many recent 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*. Similarly, biological neural circuits and in particular the hippocampus may produce representations that organize semantically related episodes. Here, we investigate the hypothesis that representations with low-dimensional latent structure, reflecting such semantic organization, result from 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 in a simulated spatial navigation task, we show that network dynamics exhibit low-dimensional but nonlinearly transformed representations of sensory inputs that capture the latent structure of the sensory environment. We quantify these results using nonlinear measures of intrinsic dimensionality which highlight the importance of the *predictive* aspect of neural representations, and provide mathematical arguments for when and why these representations emerge. We focus throughout on how our results can aid the analysis and interpretation of experimental data.

Introduction

Neural network representations are often described as encoding latent semantic information from a corpus of data (1–9). 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. How does such an organization of semantic information emerge? 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 semantic characteristics from linguistic corpora when trained to predict the context in which a given word appears (10–13). The resulting neural representations of words (known as word embeddings) have geometric properties that reflect the semantic meaning of the words they represent (14). Second, models learning to encode for future sensory information give rise to internal representations that encode spatial maps useful for goal-directed behavior (9, 15–17).

Characterizing predictive representations can shed light on *where* and *how* the brain exploits predictive mechanisms to semantically organize sensory information. The hippocampus provides a case in point. While traditionally distinct theories of hippocampus involve declarative memory (18)) and spatial navigation (19), considerable effort has been devoted to reconciling these apparently contrasting views (20–23). In particular, Eichenbaum (22) proposed that the hippocampus supports a *semantic relational network* that organizes related episodes to subserve sequential planning (7, 9, 24).

Inspired by this work, our goal here is to build theoretical and data-analytic tools that explain why a predictive learning process in neural networks leads to low-dimensional maps of the latent structure of the underlying tasks – and what the general signatures of such maps in neural recordings might be.

We begin with a generative model: observations are generated from latent variables embedded in a low-dimensional manifold. In the special case of spatial navigation, 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 (25–29) and nonlinear dimensionality (30, 31), which allows us to uncover what we call *latent space signal transfer*, wherein information about nonlinearly encoded latent variables moves into the linearly defined top principal components of the 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 by the formation of neurons with localized

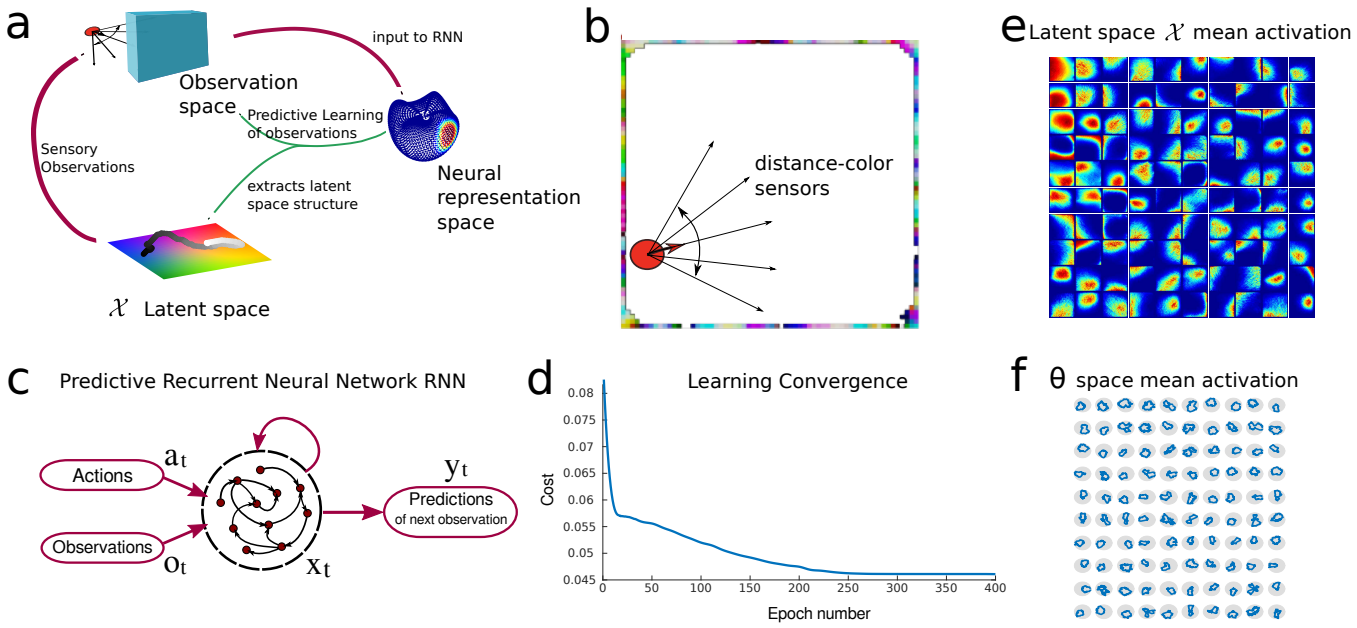

Fig. 1. Predictive network solving a navigation task. a) Logic diagram of task and information: an agent explores a latent space $\mathcal{X}$ through actions and receives observations regarding it. The network's task is to predict the next sensory observation. By learning to do so it recovers information regarding the underlying hidden latent space. b) Illustration of the agent with sensors in square maze where the walls have been colored (cfr. Methods). The 5 sensors span a 90^o degree angle and perceive the color and distance of the wall along their respective directions. The agent moves in a direction θ which is updated continuously according to draws from a gaussian distribution (giving a random walk on a circle). c) Diagram of the predictive recurrent neural network: the network receives actions and observations as inputs and is trained to output the next sensory observation. d) Cost during training for the network (cf. Methods). e) Place cell activities: average activity of 100 neurons (one per small quadrant) against the x, y coordinates of the latent space. f) Head direction activities: average activity of 100 neurons (one per small quadrant) on the latent space against the agent's direction θ.

activations on the nonlinear manifold, *manifold cells* (32). Importantly, all of these phenomena are measurable signatures of predictive learning that can be tested in data from biological or machine learning experiments.

Predictive Learning in a RNN

In predictive learning a neural network is trained to minimize the errors between its output and a stream of future sensory observations. Here we demonstrate our main result: that the network uncovers the low-dimensional latent space structure in the course of optimizing its future predictions (cfr. Fig. 1a). This occurs despite the fact that the network has no direct information regarding the latent variables generating the observations.

We test our hypothesis that predictive learning extracts the underlying low-dimensional latent variables from a high-dimensional sensory stream in the context of a spatial navigation task. In spatial navigation the latent space is the set of spatial coordinates that identify the agent's state, (x, y, θ), where θ identifies its direction. The observation space depends on the agent's ability to sense the environment. The agent we consider is equipped with simple sensors that span a visual cone of 90^o centered on its current direction θ. Each sensor reports the distance and color of the environment's wall along its direction, Fig. 1b. The environment the agent navigates is a discrete grid of locations. Each wall tile, one at each wall location, is colored randomly; a relatively narrow spatial autocorrelation of two tiles induces independent sensory observations across sensors.

For simplicity we consider the case of random exploration, where the agent's actions do not depend on the observations. At each step the agent's direction θ is updated by a small random angle $d\theta$. The agent then moves to the discrete grid location most aligned with the updated direction $\theta + d\theta$ (unless it is not 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 neighbouring locations 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 RNN learns to predict the upcoming sensory observation (see Fig. 2c). This is achieved by minimizing the difference between the RNN output y_t at time t and the upcoming observation o_{t+1} : $C = \sum_t ||y_t - o_{t+1}||^2$, Fig. 2d. We refer to the activations of the units of the trained RNN as its predictive representation.

As the agent traverses the environment, it traces out a trajectory in three spaces: the latent variable space (x, y, θ), the observation space, and the neural activation (representation) space. As the RNN 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 latent variables are a generative model for the observations); *a priori*, it is not obvious which space's influence will be stronger.

At the end of learning, we find that neurons clearly encode the latent space: Fig. 1e shows how the latent variables are encoded in the neural representation space. Moreover, single neurons' receptive fields function as "place" and "border" cells that encode the latent variables x and y, and as "head

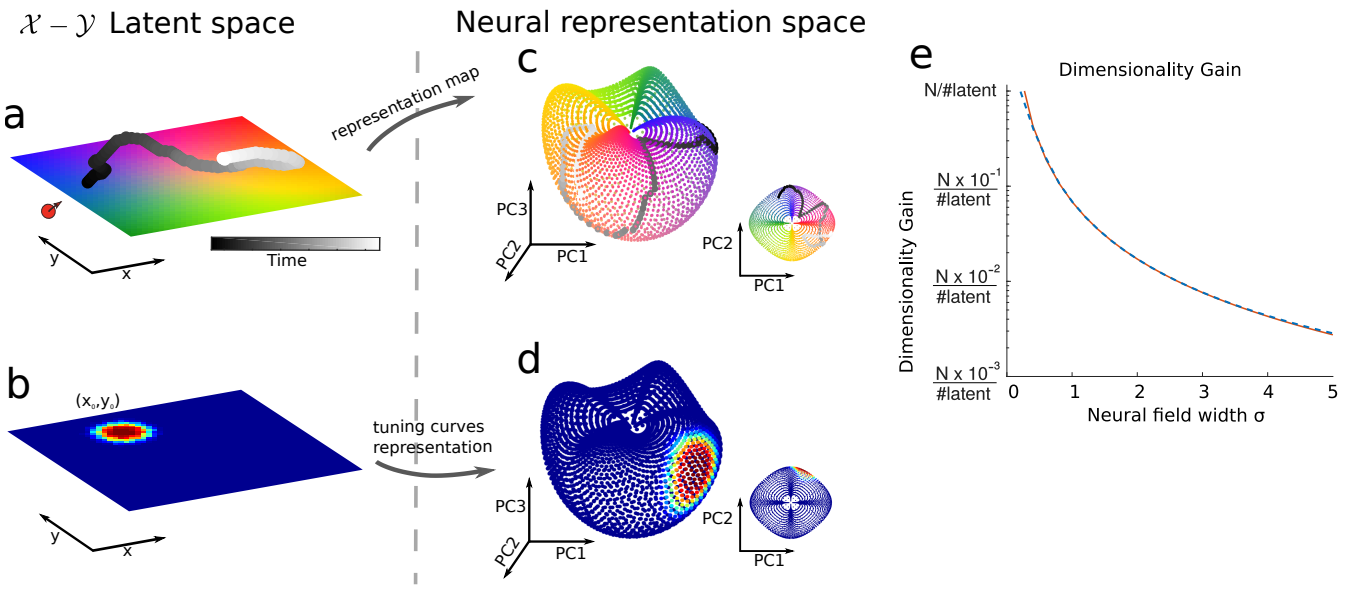

Fig. 2. Manifold analysis example. a) Example of a two dimensional environment in which the agent moves. We assign a unique color to each location of the environment. A segment of the agent's trajectory is represented in gray scale, with shade standing for time. b) Example tuning of a neuron with gaussian 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) PR 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.

direction" cells that encode θ (Fig. 1f) (19, 33, 34). Thus, the neural representation has extracted information about the latent space from the observations, without any explicit prompt to do so. In the last section and more in depth in the Suppl. Mat., we show how this phenomenon is robust to alterations of the sensory observations and network architecture.

Latent and neural representation spaces

So far, we have considered how the latent variables are represented one neuron at a time within our predictive learning RNN. How does the neural population as a whole represent the latent space? To answer this question precisely we develop methods for analyzing neural representation manifolds. We begin with the most basic characteristic of a representation manifold, its dimensionality. We start by analyzing 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 (19, 35–37). Specifically, given two continuous variables x, y that parametrize a latent space, Fig. 2a, 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. 2b, $\mathcal{G}_\sigma(x,y) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{(x-x_0)^2+(y-y_0)^2}{2\sigma^2}\right)$. The responses of an ensemble of N neurons map the latent space manifold (Fig. 2a) 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. 2c. As the agent traverses a trajectory $\boldsymbol{x}_t$ in the 2d latent space (Fig. 2a, grayscale), the representation $\boldsymbol{r}_t$ traces out a trajectory on the response manifold (Fig. 2c, grayscale). We can view the tuning curve of a single neuron (Fig. 2b) on the response manifold to obtain the *manifold tuning curve* of this neuron (Fig. 2d). In the next section we will analyze in more depth the meaning and properties of manifold tuning curves.

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 (30).

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:

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

where $\tilde{\lambda}_i = \lambda_i / \sum_{j=1}^N \lambda_j$, see Suppl.Mat. Fig.S2 (25, 27–29). The two most important aspects of these measures of dimension are:

- ID of the representation manifold is determined by the

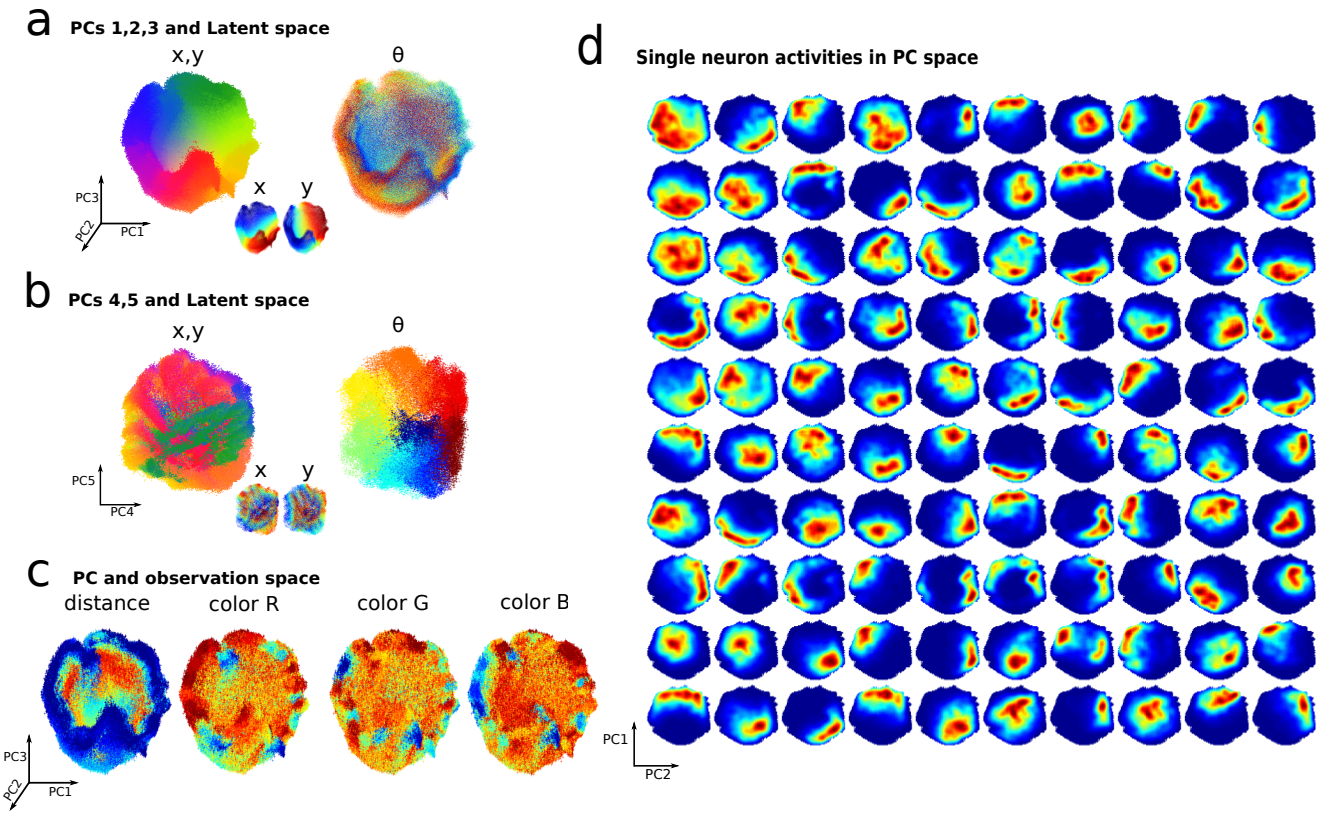

Fig. 3. Signatures of the learned predictive representation. a) 100000 points of the neural network representation, corresponding to an equal number of steps for the agent's exploration, are shown projected into the space spanned by by PCs 1 to 3 of the learned representation, and colored respectively with respect to x, y latent variables (cfr. Fig. 1b for colorcode) 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.

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. 2e 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. In the Suppl.Mat. we give an analytical formula for this relationship as well as a more thorough explanation of relationships among ID, PR, and DG (Fig.S1).

The learned neural representation manifold

In the previous section we illustrated signatures of low-D representation manifolds of the latent variable space in the case where neurons function exactly as place cells directly encoding the latent space. This led to interesting and readily measurable phenomena. First, neurons show clear response fields on the response manifold, which we named manifold tuning curves. Moreover, the representation manifold is low-dimensional while appearing higher-dimensional according to linear measures: that is, the representation has a high dimensionality gain (DG). These observations beg the question of whether representation manifolds learned via the predictive learning framework introduced in the first section will have the same properties.

We begin by showing in Fig. 3a 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 that 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. 3b.

As the agent's input is the observations rather than the latent variables, it is natural to ask whether the observation

variables are similarly encoded in the RNN representation. Fig. 3c 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 channels. Intriguingly, this is a consequence of learning: average color information *is* encoded in the first PCs in the beginning of learning (see more details below). Figs. 3a and 3b, taken together, suggest that the network allocates most of its internal variability to the encoding of latent variables.

We next explore the relationship between the responses of single cells and the population activity along the manifold. In the simplest case of Fig. 2, in which the latent space directly parameterized the responses of individual cells, we showed that the receptive fields of single cells tiled the representation manifold in the same way that they tiled the latent space. Does the same phenomenon occur for learned representations in the RNN? Fig. 3d demonstrates that this is indeed the case, by showing the activity of the same 100 neurons in Fig. 1e averaged over "locations" in the space spanned by the first two PCs.

This reveals that single neurons have activities that resemble receptive fields on the neural representation manifold. We refer to these as *neural manifold cells*. If the neural manifold clearly represents the latent space (Fig. 3a) and neural receptive fields tile the latent space (Fig. 1e), then neural activities are also localized on the manifold. Intriguingly the reverse is also true: localized activities in the latent space (e.g. place cells, cfr. Fig. 1e) follow from tiling of the manifold by single neuron receptive fields.

The preceding analysis shows how the neural representation manifold and single neuron coding are tied to one another, via the latent space. We proceed to study how the manifold and its connection to the latent space emerges over the course of predictive learning.

In Fig. 2 we highlighted two different ways to assess the dimensionality of the representation: a linear measure (Participation Ratio, PR) and a nonlinear one (Intrinsic Dimensionality, ID). Here, we find that the PR of predictive representations, computed at every training epoch, keeps increasing through learning (Fig. 4a). The increase corresponds to the formation of place cells with respect to the latent space (Fig. 1e) or, equivalently, manifold cells with respect to the representation manifold (Fig. 2d). While the PR increases, ID decreases until it reaches a value of approximately 5 (Fig. 4b; see also Methods). Recall from our analysis in the previous section that the value of ID is independent of single neuron fields. Although we cannot explain the number 5 precisely, we note that if the latent variables are encoded then it cannot be less than the number of latent components (x, y, θ). Furthermore the encoding of the actions could explain the fact that it is higher than 3. ID is considerably smaller than PR, pointing to a dimensionality gain DG of roughly $\mathrm{DG} = \frac{\mathrm{PR}}{\mathrm{ID}} \approx 3$ toward the end of learning. This is consistent with our previous analysis where we showed that local manifold fields tend to increase the DG, (cf. Fig. 2e and Suppl. Mat.).

In Figs. 3a and 3b we showed that the first five PCs of the learned representation are highly correlated with latent space variables. This *latent space signal transfer* is another signature of predictive learning that we can exploit and track through training. Specifically, 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. 4c shows the average CC between the representation in PCs 1 to 3 and the position x, y of the agent in latent space. When the average CC is 1, this means that all the signal regarding x, y has been transferred onto PCs 1 to 3. Similar interpretations hold for the other curves we show, which track the transfer of signal relative to the latent space x, y, θ. Fig. 4c 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 activities. The same analysis can be carried out with respect to observation space variables. This is shown in Fig. 4d, and indicates that the observation space signal flows out of the first few PC components as learning progresses. Together Figs. 4c and 4d show that the representation, as interpreted through PC components, encodes more latent space information vs. observation space information as learning progresses (blue and red lines).

The transfer of latent variable information to the first PCs of the representation is tightly connected to the linear and non-linear dimensionality of the representation, as discussed in more depth in the Suppl. Mat. Altogether Fig. 4 suggests that predictive learning forms a low-D representation (Fig. 4a), with specific signatures that can be quantified via latent signal transfer and dimensionality (Fig. 4b).

A neural network mechanism for low-D representation manifolds through predictive learning

Why does predictive learning lead to the discovery, and low-D representation, of the latent space? In this section we provide theoretical arguments suggesting why the predictive step, in particular, can be such an important ingredient in extracting latent manifolds.

For simplicity, we consider the case where the movement of the agent in the latent space $\mathcal{X}$ is governed by a discrete-time dynamical system:

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + F(\boldsymbol{x}_t) \tag{3}$$

where $\boldsymbol{x} = (x, y, \theta)$ and $F(\boldsymbol{x})$ is a vector field on $\mathcal{X}$; for the arguments below, this vector field may be deterministic or stochastic (as for the "off policy" actions taken by the agent in our simulations). We note that F may depend on a learned policy but, without loss of generality, we omit this detail. The agent's observation at time t is then defined as a differentiable function of the latent variable: $\boldsymbol{o}_t = \varphi(\boldsymbol{x}_t)$. Such a mapping induces a nonlinear dynamical system in the space of the observations $\boldsymbol{o}$ which can be written in terms of the dynamics of $\boldsymbol{x}_t$: $\boldsymbol{o}_{t+1} = \varphi(\boldsymbol{x}_t + F(\boldsymbol{x}_t))$. Assuming that the trajectory $\boldsymbol{x}_t$ stays close to a reference point $\boldsymbol{x}^* \in \mathcal{X}$ we can expand the

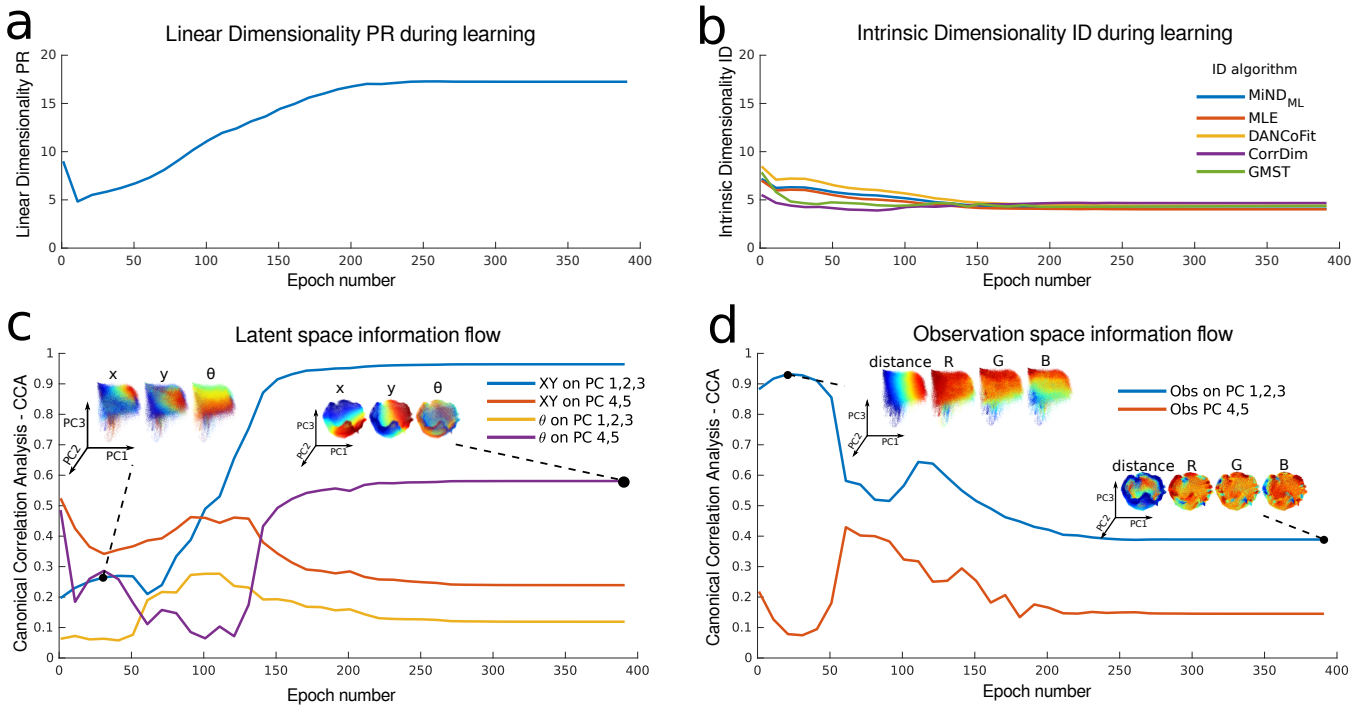

Fig. 4. Learning the predictive representation. a) Participation Ratio of the representation during learning. b) Intrinsic Dimensionality (ID) of the representation during learning. Five different intrinsic dimensionality estimators are used (cfr. Methods). c) Signal transfer analysis: Canonical Covariance Analysis between PCs of the neural representation and the latent space. d) Same as panel c) but for the observation space.

observations in terms of the latent variables around $\boldsymbol{x}^*$:

$$\begin{aligned}\boldsymbol{o}_{t+1} &= \varphi(\boldsymbol{x}^*) + D\varphi(\boldsymbol{x}^*)(\boldsymbol{x}_t + F(\boldsymbol{x}_t) - \boldsymbol{x}^*) + \mathcal{O}(2) \\ &= \varphi(\boldsymbol{x}^*) + D\varphi(\boldsymbol{x}^*)(\boldsymbol{x}_t - \boldsymbol{x}^*) + D\varphi(\boldsymbol{x}^*)F(\boldsymbol{x}_t) + \mathcal{O}(2) \\ &\simeq \boldsymbol{o}_t + D\varphi(\boldsymbol{x}^*)F(\boldsymbol{x}_t)\end{aligned} \tag{4}$$

where higher order terms can be neglected when the linear regime dominates and $D\varphi(\boldsymbol{x}^*)$ is the Jacobian matrix of φ evaluated at $\boldsymbol{x}^*$.

We now turn to the update rules of the artificial recurrent network, also defined as a discrete-time dynamical system:

$$\begin{aligned}\boldsymbol{r}_t &= g(\boldsymbol{W}\boldsymbol{r}_{t-1} + \boldsymbol{W}_{in}\boldsymbol{o}_t) \\ \boldsymbol{y}_t &= g(\boldsymbol{W}_{out}\boldsymbol{r}_t)\end{aligned} \tag{5}$$

where g is a nonlinear function and $\boldsymbol{W}, \boldsymbol{W}_{in}, \boldsymbol{W}_{out}$ are respectively recurrent, input and output weights (the agent's actions are not considered here, cfr. Suppl. Mat. for further details).

We compare the effect of two cost functions on learning in the network, given an agent's trajectory $\{\boldsymbol{x}_t |\ 0 \leq t \leq T\}$ in latent space: one predictive and another non-predictive, respectively represented by

$$\begin{aligned}\mathcal{C}_{pred} &= \frac{1}{T}\sum_{t=0}^{T-1} ||\boldsymbol{o}_{t+1} - \boldsymbol{y}_t||^2, \\ \mathcal{C}_{non-pred} &= \frac{1}{T}\sum_{t=0}^{T-1} ||\boldsymbol{o}_t - \boldsymbol{y}_t||^2.\end{aligned} \tag{6}$$

For the predictive coding objective $\mathcal{C}_{pred}$, we use Eq. (4) and Eq. (5) to obtain

$$\begin{aligned}||\boldsymbol{o}_{t+1} - \boldsymbol{y}_t||^2 = ||\boldsymbol{o}_t + D\varphi(\boldsymbol{x}_t)F(\boldsymbol{x}_t) - \\ g(\boldsymbol{W}_{out}g(\boldsymbol{W}\boldsymbol{r}_{t-1} + \boldsymbol{W}_{in}\boldsymbol{o}_t))||^2.\end{aligned} \tag{7}$$

Assuming that the activity of the network remains in a regime where g is approximately linear (for convenience, with slope 1), we can further simplify Eq. (7) into

$$\begin{aligned}||\boldsymbol{o}_{t+1} - \boldsymbol{y}_t||^2 &= ||\boldsymbol{o}_t + D\varphi(\boldsymbol{x}^*)F(\boldsymbol{x}_t) - \boldsymbol{W}_{out}\boldsymbol{W}\boldsymbol{r}_{t-1} \\ &\quad - \boldsymbol{W}_{out}\boldsymbol{W}_{in}\boldsymbol{o}_t||^2 \\ &\leq ||\boldsymbol{o}_t - \boldsymbol{W}_{out}\boldsymbol{W}_{in}\boldsymbol{o}_t||^2 \\ &\quad + ||D\varphi(\boldsymbol{x}^*)F(\boldsymbol{x}_t) - \boldsymbol{W}_{out}\boldsymbol{W}\boldsymbol{r}_{t-1}||^2.\end{aligned} \tag{8}$$

The two terms in this inequality suggest a possible solution to minimizing $\mathcal{C}_{pred}$: to "auto-encode" the observation at the current time $\boldsymbol{o}_t$ while learning a linear representation of the observed dynamics. The latter necessarily implies a low dimensional representation, the same as latent space. To see this, consider a sample trajectory of length T in a neighborhood of $\boldsymbol{x}^*$: $\{\boldsymbol{x}_t | 1 < t < T\}$ and the corresponding network activations $\{\boldsymbol{r}_t | 1 < t < T\}$. Let X and R be the following $N_{latent} \times T$ and $N \times T$ matrices, respectively:

$$X = \begin{pmatrix} | & & | \\ \boldsymbol{x}_1 & \dots & \boldsymbol{x}_T \\ | & & | \end{pmatrix}, \quad R = \begin{pmatrix} | & & | \\ \boldsymbol{r}_1 & \dots & \boldsymbol{r}_T \\ | & & | \end{pmatrix}$$

It follows that minimizing the contribution of each term in Eq. (8) to minimize $\mathcal{C}_{pred}$ is equivalent to solving the

ordinary least squares problem:

$$\begin{aligned} \varphi(X) &\simeq \boldsymbol{W}_{out}\boldsymbol{W}_{in}\varphi(X) \\ D\varphi(\boldsymbol{x}^*)F(X) &\simeq \boldsymbol{W}_{out}\boldsymbol{W}R \end{aligned} \quad \textbf{(9)}$$

where φ and F are applied column-wise to X. This suggests that $\boldsymbol{W}_{out}\boldsymbol{W}_{in} \approx \mathcal{I}$ while the activation vector $\boldsymbol{r}$ mainly encodes a representation of the latent variable's dynamic update rule $F(\boldsymbol{x})$ (akin to the dynamics' derivative). Furthermore, as X is rank N_{latent} and, assuming $\boldsymbol{W}_{out}$ and $\boldsymbol{W}$ are of higher rank, a natural way to satisfy this is by R also being rank N_{latent}. This is consistent with low-dimensional network dynamics.

We emphasize that the analysis above is approximate, and is local, involving linearization around a given point $\boldsymbol{x}^*$ in the latent space. However, by allowing $\boldsymbol{x}^*$ to change in time so that the linear approximation holds for trajectories on a longer scale, the network would then learn a collection of local linear dynamics. We observe clues in our numerical experiments that these approximate relationships are indeed respected. Fig. S2b shows that the matrix $\boldsymbol{W}_{out}\boldsymbol{W}_{in}$ has a clear diagonal structure suggesting that input observations are fed forward to the outputs. The role of recurrent dynamics is then to approximate the local map $D\varphi(\boldsymbol{x}^*)F(\boldsymbol{x})$. In this sense the representation $\boldsymbol{r}$ doesn't directly encode for $\boldsymbol{x}$ but rather represents a collection of local linear maps indexed by the position of the agent in the latent space, and coding for its dynamics in this space.

By contrast, for the non-predictive objective $\mathcal{C}_{non-pred}$ the terms $||\boldsymbol{o}_{t+1} - \boldsymbol{y}_{t+1}||^2 = ||\boldsymbol{o}_t - \boldsymbol{W}_{out}\boldsymbol{W}\boldsymbol{r}_{t-1} - \boldsymbol{W}_{out}\boldsymbol{W}_{in}\boldsymbol{o}_t||^2$ are missing the dynamic update and cannot be decomposed as in Eq. (7). The absence of the low-dimensional latent space dynamics in this non-predictive settings suggests that the representation shouldn't discover the latent manifold through learning. We will demonstrate this explicitly in the next section.

The arguments above imply that predictive representations will have low ID (i.e., low nonlinear dimensionality). We next give reasoning for why such predictive representation develop localized receptive fields. As shown in Fig. 2e, 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:

$$d_{\boldsymbol{r},\boldsymbol{r}'} \leq \kappa d_{\boldsymbol{x},\boldsymbol{x}'} \quad \textbf{(10)}$$

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 (38, 39) (cf. Suppl.Mat). Interestingly, in our framework this result appears to be true for both positive representations (when the activation function is a sigmoid) and more general ones (e.g. when the activation function is tanh, data not shown).

The arguments above indicate that predictive learning leads to increases in linear dimensionality, as observed in our learning simulations (Fig. 4a). 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 in Eq. (6) 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. (40–42)). 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. 4a.

Non-predictive learning fails to extract low-D latent manifold

A central idea in this article is that learning is predictive, so that the underlying RNN is learning to anticipate the observation on the next timestep. But is the predictive aspect really necessary for the network to extract the low-D latent manifold? Here we address this question by directly contrasting predictive learning with the corresponding non-predictive case.

We train 100 RNNs, which differ only in the initialization of their weights and the agent's generated trajectory, in two different scenarios: predictive learning vs. non-predictive auto-encoding; that is, predicting the next observation o^{t+1} as described earlier, and vs. returning the current observation o^t (43, 44). We find that all networks trained through predictive learning show the characteristics outlined above (low Intrinsic Dimensionality with high Dimensionality Gain and latent signal transfer), while the same networks trained with the auto-encoding loss develop qualitatively different representations. Most importantly, with the auto-encoding loss the learned representations do not reflect the latent state variables as they do for the predictive coding loss, i.e. latent variables do not dominate the linear factors.

In Figs. 5a and 5b we show CCA between the first three PCs of the representation and the latent space or the observations yields completely different trends in the predictive vs non-predictive case. In Fig. 5a the average CCA coefficient between the representation and the latent space grows throughout learning while the average coefficient between the representation and the observations decreases (cfr. Figs. 4c and 4d). In contrast, by this metric the networks trained to auto-encode the observations do not develop representations that encode the latent space, but rather only the observations. Consequently, as shown in Fig. 5c the non-predictive representation fails to develop place fields; in particular, the activities of neurons are not

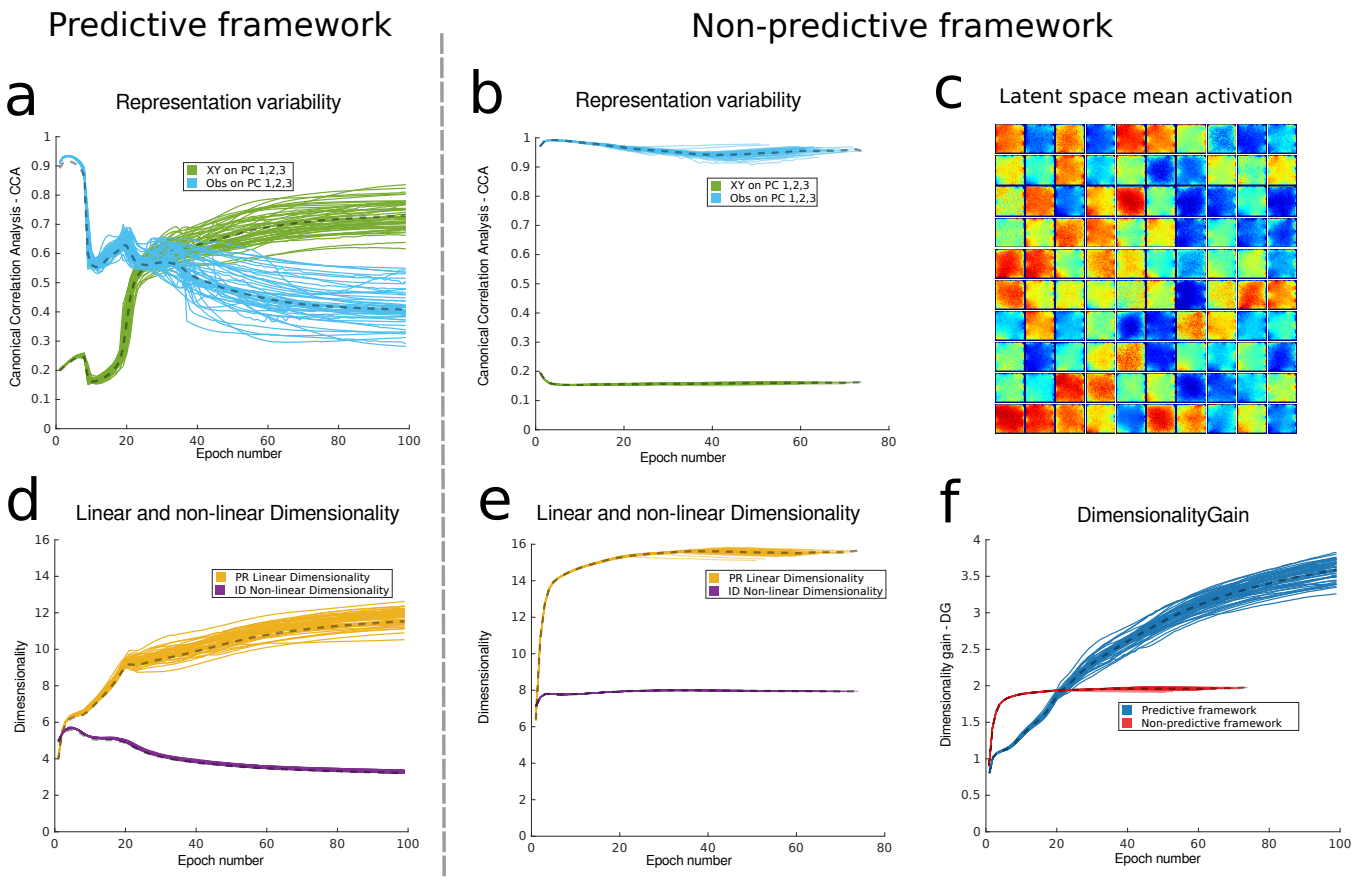

Fig. 5. Comparing signatures of learned representations in the predictive vs non-predictive framework. a) Canonical Correlation Analysis (CCA) between PCs 1 to 3 of the neural representation and the latent (green) or observation (blue) spaces during learning for 100 network instances. Same as panel b but for the non-predictive case. c) Neural spatial tuning: average activations for 100 cells in the non-predictive case. Same as for Fig. 1d but for non-predictive learning. d) PR and ID dimensionality for networks trained on predictive learning. e) PR (yellow) and ID (purple) dimensionality for the non-predictive networks. f) Dimensionality gain for predictive (blue) and non-predictive (red) networks throughout learning.

localized in the latent space. This is in striking contrast with the same plots for the predictive case Fig. 1e.

The PR and ID dimensions of the learned representations also differ significantly between the predictive and non-predictive settings. For the predictive learning network (Fig. 5d), PR grows and ID decreases throughout training. For the auto-encoding network (Fig. 5e) PR grows but ID does not decrease, as the representation does not extrac the latent manifold. We can summarize these properties by analyzing the Dimensionality Gain. Fig. 5f shows that in the predictive case (blue line) DG progressively increases through learning, while this does not occur for the non-predictive case.

Control simulations that test role of recurrence and robustness to task setup

We conducted a series of additional simulations to control that our main findings are robust and rely on predictive learning, as our theoretical arguments predict. The Suppl. Mat. gives a fuller description of these (Fig. S3, Fig. S4); we give a brief listing here. First, we checked the importance of a trained RNN architecture, showing that both freezing the output weights and using a non-recurrent network hinder the development of predictive representations with the key properties described above. We also checked the importance of the predictive nature of the training objective: training the network to reproduce the observations on the last time step as opposed to predicting those on the next step hinders learning (as does autoencoding the input, as pointed out above). Finally, our results are robust to the details of input statistics, specifically to adding noise in the input and to the degradation of the information contained in the input to be without color space, action space or distance-related information. Altogether these findings corroborate our theoretical arguments, cfr. Suppl.Mat. and Fig. S3, Fig. S4.

Discussion

How the brain extracts information about the latent structures of the external world given only indirect sensory observations is a long-standing question. We find that predictive learning in recurrent neural networks (RNNs) leads to an intriguing answer, as it automatically constructs a low dimensional neural representation of the latent space. We explore this phenomenon both in simulations of an egocentric spatial navigation task – a situation that is naturally described by latent variables corresponding to the spatial coordinates, and providing intuitive mathematical arguments that indicate the generality of the phenomenon.

Signatures of predictive learning in neural data What features characterize predictive learning in neural data? When the observations to be predicted arise from an environment with an underlying low-dimensional latent structure, our work suggests several distinct signatures. First, the dimensionality of the set of neural responses will likely appear high when assessed with standard linear measures, such as the participation ratio. However, when assessed through nonlinear metrics sensitive to the dimensionality of curved manifolds, the dimensionality will be lower, tending to the number of independent latent variables. These two signatures taken together imply a high dimensionality gain (DG), or ratio of linear to nonlinear dimension.

The presence of a low-D *neural representation manifold* suggests another signature of predictive learning: *neural manifold cells*, with responses strongly tuned to the variables which parameterize the neural representation manifold (cfr. Fig. 3d). While locality in latent space is an established aspect of neural receptive fields, locality in the manifold is an allied feature that will be exciting to check in experimental data. This builds on recent work on understanding neuronal representations through the lens of representation dimensionality (26–28, 38, 45).

Discovering latent structure in data and sensory observations Our techniques require no advance knowledge of what the latent variables are, or even how many of them there are. 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 by (43, 46–48). We introduce *latent signal transfer* as a viable way to uncover the relevant variables fig. 3d: as the response manifold is learned, the position of population responses along the manifold can be increasingly well predicted by the true low-dimensional latent variables, but increasingly poorly predicted by irrelevant variables. Thus, the problem of discovering the low-dimensional, latent structure in complex, high-dimensional dynamic signals becomes that of discovering the variables that parameterize a low-dimensional neural response manifold. We suggest that such *parametrization* of learning via dimensionality and latent signal transfer – two related phenomena as discussed in the Suppl. Mat. – may contribute to the understanding of how both biological brains and neural network algorithms solve difficult tasks such as navigating an environment based on complex, high-dimensional cues.

Related frameworks and findings 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 the semantic relationships between the data samples in the tasks. On one hand, information retrieval and computational linguistics have benefited enormously from the geometric properties of word embeddings learned by predictive models (10–12, 46). On the other hand, prediction over observations has been used as an auxiliary task in reinforcement learning to acquire representations favoring goal-directed learning (9, 15–17). Finally we note that the responses are reminiscent of the types of place-related activity observed in the hippocampus and entorhinal cortex, lending in particular mechanistic grounding to the recent proposal by (22) that the hippocampus builds a *semantic relational network*. We argue that relevant semantic relations are encoded by neural representation of low intrinsic dimensionality, and in turn these are being 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 (14, 15, 49).

Open questions Distinctive to our work is the use of nonlinear dimensionality analysis to characterize the relationship between the neural representation manifold and the latent space. In order to reveal this low-dimensional structure, we rely on nonlinear techniques, as more common linear measures would give the illusion of high-dimensional representations. Nonetheless, more work is needed to harness and theoretically formalize the role of nonlinearities in neural population codes. Furthermore, predictive learning is a general framework that goes beyond the example of navigation analyzed here, and future work will expand in other directions (text, visual processing, behavioral tasks, etc.) that may open new theoretical frameworks and new implications for learning and generalization.

Finally, it will be crucial 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, dimensionality gain, and latent signal transfer. A very exciting possibility is that this might uncover the presence of latent variables in tasks where they were previously unsuspected or unidentified.

Acknowledgments

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. The authors would like to acknowledge the numerous colleagues who have helped to crystallise 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).

Bibliography

1. Yoshua Bengio. Deep Learning of Representations: Looking Forward. In Adrian-Horia Dediu, Carlos Martín-Vide, Ruslan Mitkov, and Bianca Truthe, editors, *Statistical Language and Speech Processing*, number 7978 in Lecture Notes in Computer Science, pages 1–37. Springer Berlin Heidelberg, July 2013. ISBN 978-3-642-39592-5 978-3-642-39593-2. doi: 10.1007/978-3-642-39593-2_1.
2. Rodrigo Laje and Dean V. Buonomano. Robust timing and motor patterns by taming chaos in recurrent neural networks. *Nature Neuroscience*, 16(7):925–933, July 2013. ISSN 1097-6256. doi: 10.1038/nn.3405.
3. Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, and

others. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529–533, 2015. 00269.

4. David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, and others. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016. 00043.
5. Alex Graves, Greg Wayne, Malcolm R. Eynolds, Tim Harley, Ivo Danihelka, Agnieszka Grabska-Barwinska, Sergio Gomez Colmenarejo, Edward Grefenstette, Tiago R. Amalho, John Agapiou, Adria Puigdomenech Badia, Karl Moritz Hermann, Yori Zwols, Georg O. Strovski, Adam C. Ain, Helen King, Christopher Summerfield, Phil B. Lunsom, Koray Kavukcuoglu, and Demis Hassabis. Hybrid computing using a neural network with dynamic external memory. *Nature*, 538(7626):471–+, October 2016. doi: 10.1038/nature20101. WOS:000386654400049.
6. Tejas D. Kulkarni, Ardavan Saeedi, Simanta Gautam, and Samuel J. Gershman. Deep Successor Reinforcement Learning. *arXiv:1606.02396 [cs, stat]*, June 2016. arXiv: 1606.02396.
7. Andrea Banino, Caswell Barry, Benigno Uria, Charles Blundell, Timothy Lillicrap, Piotr Mirowski, Alexander Pritzel, Martin J. Chadwick, Thomas Degris, Joseph Modayil, Greg Wayne, Hubert Soyer, Fabio Viola, Brian Zhang, Ross Goroshin, Neil Rabinowitz, Razvan Pascanu, Charlie Beattie, Stig Petersen, Amir Sadik, Stephen Gaffney, Helen King, Koray Kavukcuoglu, Demis Hassabis, Raia Hadsell, and Dharshan Kumaran. Vector-based navigation using grid-like representations in artificial agents. *Nature*, 557(7705):429–433, May 2018. ISSN 1476-4687. doi: 10.1038/s41586-018-0102-6.
8. Arkady Konovalov and Ian Krajbich. Neurocomputational Dynamics of Sequence Learning. *Neuron*, 98(6):1282–+, June 2018. ISSN 0896-6273. doi: 10.1016/j.neuron.2018.05.013. WOS:000436587600024.
9. Greg Wayne, Chia-Chun Hung, David Amos, Mehdi Mirza, Arun Ahuja, Agnieszka Grabska-Barwinska, Jack Rae, Piotr Mirowski, Joel Z. Leibo, Adam Santoro, Mevlana Gemici, Malcolm Reynolds, Tim Harley, Josh Abramson, Shakir Mohamed, Danilo Rezende, David Saxton, Adam Cain, Chloe Hillier, David Silver, Koray Kavukcuoglu, Matt Botvinick, Demis Hassabis, and Timothy Lillicrap. Unsupervised Predictive Memory in a Goal-Directed Agent. *arXiv:1803.10760 [cs, stat]*, March 2018. arXiv: 1803.10760.
10. Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.
11. Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394. Association for Computational Linguistics, 2010.
12. Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537, 2011.
13. Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
14. Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. Rand-walk: A latent variable model approach to word embeddings. *arXiv preprint arXiv:1502.03520*, 2015.
15. Peter Dayan. Improving generalization for temporal difference learning: The successor representation. *Neural Computation*, 5(4):613–624, 1993. 00086.
16. Kimberly L Stachenfeld, Matthew Botvinick, and Samuel J Gershman. Design Principles of the Hippocampal Cognitive Map. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2528–2536. Curran Associates, Inc., 2014.
17. Evan M Russek, Ida Momennejad, Matthew M Botvinick, Samuel J Gershman, and Nathaniel D Daw. Predictive representations can link model-based reinforcement learning to model-free mechanisms. *PLoS computational biology*, 13(9):e1005768, 2017.
18. N. J. Cohen and L. R. Squire. Preserved learning and retention of pattern-analyzing skill in amnesia: dissociation of knowing how and knowing that. *Science*, 210(4466):207–210, October 1980.
19. 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(1):171–175, November 1971.
20. György Buzsáki and Edvard I. Moser. Memory, navigation and theta rhythm in the hippocampal-entorhinal system. *Nat Neurosci*, 16(2):130–138, February 2013. doi: 10.1038/nn.3304.
21. Branka Milivojevic and Christian F Doeller. Mnemonic networks in the hippocampal formation: From spatial maps to temporal and conceptual codes. *Journal of Experimental Psychology: General*, 142(4):1231, 2013.
22. Howard Eichenbaum and Neal J. Cohen. Can we reconcile the declarative memory and spatial navigation views on hippocampal function? *Neuron*, 83(4):764–770, August 2014. doi: 10.1016/j.neuron.2014.07.032.
23. Daniela Schiller, Howard Eichenbaum, Elizabeth A. Buffalo, Lila Davachi, David J. Foster, Stefan Leutgeb, and Charan Ranganath. Memory and Space: Towards an Understanding of the Cognitive Map. *J Neurosci*, 35(41):13904–13911, October 2015. doi: 10.1523/JNEUROSCI.2618-15.2015.
24. Ingmar Kanitscheider and Ila Fiete. Training recurrent networks to generate hypotheses about how the brain solves hard navigation problems. In *Advances in Neural Information Processing Systems*, pages 4529–4538, 2017.
25. Larry F Abbott, Kanaka Rajan, and Haim Sompolinsky. Interactions between intrinsic and stimulus-evoked activity in recurrent neural networks. *The dynamic brain: an exploration of neuronal variability and its functional significance*, pages 1–16, 2011.
26. Mattia Rigotti, Omri Barak, Melissa R Warden, Xiao-Jing Wang, Nathaniel D Daw, Earl K Miller, and Stefano Fusi. The importance of mixed selectivity in complex cognitive tasks. *Nature*, 497(7451):585, 2013.
27. Luca Mazzucato, Alfredo Fontanini, and Giancarlo La Camera. Stimuli Reduce the Dimensionality of Cortical Activity. *Frontiers in Systems Neuroscience*, 10, February 2016. ISSN 1662-5137. doi: 10.3389/fnsys.2016.00011.
28. Ashok Litwin-Kumar, Kameron Decker Harris, Richard Axel, Haim Sompolinsky, and L. F. Abbott. Optimal Degrees of Synaptic Connectivity. *Neuron*, 93(5):1153–1164.e7, March 2017. ISSN 0896-6273. doi: 10.1016/j.neuron.2017.01.030.
29. Peiran Gao, Eric Trautmann, Byron M. Yu, Gopal Santhanam, Stephen Ryu, Krishna Shenoy, and Surya Ganguli. A theory of multineuronal dimensionality, dynamics and measurement. *bioRxiv*, page 214262, November 2017. doi: 10.1101/214262.
30. Francesco Camastra and Antonino Staiano. Intrinsic dimension estimation: Advances and open problems. *Information Sciences*, 328:26–41, January 2016. ISSN 0020-0255. doi: 10.1016/j.ins.2015.08.029.
31. P. Campadelli, E. Casiraghi, C. Ceruti, and A. Rozza. Intrinsic Dimension Estimation: Relevant Techniques and a Benchmark Framework. *Mathematical Problems in Engineering*, 2015. doi: 10.1155/2015/759567.
32. Ryan J Low, Sam Lewallen, Dmitriy Aronov, Rhino Nevers, and David W Tank. Probing variability in a cognitive map using manifold inference from neural dynamics. *bioRxiv*, 2018. doi: 10.1101/418939.
33. Jeffrey S Taube, Robert U Muller, and James B Ranck. Head-direction cells recorded from the postsubiculum in freely moving rats. i. description and quantitative analysis. *Journal of Neuroscience*, 10(2):420–435, 1990.
34. Maya Geva-Sagiv, Liora Las, Yossi Yovel, and Nachum Ulanovsky. Spatial cognition in bats and rats: from sensory acquisition to multiscale maps and navigation. *Nature Reviews Neuroscience*, 16(2):94, 2015.
35. Trygve Solstad, Charlotte N. Boccara, Emilio Kropff, May-Britt Moser, and Edvard I. Moser. Representation of Geometric Borders in the Entorhinal Cortex. *Science*, 322(5909): 1865–1868, December 2008. doi: 10.1126/science.1166466. WOS:000261799400061.
36. Hanne Stensola, Tor Stensola, Trygve Solstad, Kristian Froland, May-Britt Moser, and Edvard I. Moser. The entorhinal grid map is discretized. *Nature*, 492(7427):72–78, December 2012. doi: 10.1038/nature11649. WOS:000311893400047.
37. Tom J. Wills, Francesca Cacucci, Neil Burgess, and John O'Keefe. Development of the Hippocampal Cognitive Map in Preweanling Rats. *Science*, 328(5985):1573–1576, June 2010. doi: 10.1126/science.1188224. WOS:000278859200051.
38. Anirvan Sengupta, Mariano Tepper, Cengiz Pehlevan, Alexander Genkin, and Dmitri Chklovskii. Manifold-tiling localized receptive fields are optimal in similarity-preserving neural networks. *bioRxiv*, 2018. doi: 10.1101/338947.
39. Cengiz Pehlevan, Anirvan M Sengupta, and Dmitri B Chklovskii. Why do similarity matching objectives lead to hebbian/anti-hebbian networks? *Neural computation*, 30(1):84–124, 2018.
40. Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th annual international conference on machine learning*, pages 689–696. ACM, 2009.
41. Alekh Agarwal, Animashree Anandkumar, Prateek Jain, Praneeth Netrapalli, and Rashish Tandon. Learning sparsely used overcomplete dictionaries. In *Conference on Learning Theory*, pages 123–137, 2014.
42. Subhaneil Lahiri, Peiran Gao, and Surya Ganguli. Random projections of random manifolds. *arXiv preprint arXiv:1607.04331*, 2016.
43. G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006. ISSN 0036-8075. doi: 10.1126/science.1127647.
44. Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and Composing Robust Features with Denoising Autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 1096–1103, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-205-4. doi: 10.1145/1390156.1390294.
45. N Alex Cayco-Gajic, Claudia Clopath, and R Angus Silver. Sparse synaptic connectivity is required for decorrelation and pattern separation in feedforward networks. *Nature Communications*, 8(1):1116, 2017.
46. Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.
47. Trevor Hastie, Robert Tibshirani, and Jerome Friedman. Unsupervised learning. In *The elements of statistical learning*, pages 485–585. Springer, 2009.
48. Kilian Q. Weinberger and Lawrence K. Saul. Unsupervised learning of image manifolds by semidefinite programming. *International journal of computer vision*, 70(1):77–90, 2006.
49. Kimberly Lauren Stachenfeld, Matthew M Botvinick, and Samuel J Gershman. The hippocampus as a predictive map. *bioRxiv*, 2016. doi: 10.1101/097170.
50. Gabriele Lombardi, Alessandro Rozza, Claudio Ceruti, Elena Casiraghi, and Paola Campadelli. 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, pages 374–389, Berlin, Heidelberg, 2011. Springer-Verlag. ISBN 978-3-642-23782-9.
51. Elizaveta Levina and Peter J. Bickel. Maximum Likelihood Estimation of Intrinsic Dimension. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 777–784. MIT Press, 2005.
52. Claudio Ceruti, Simone Bassis, Alessandro Rozza, Gabriele Lombardi, Elena Casiraghi, and Paola Campadelli. DANCo: Dimensionality from Angle and Norm Concentration. *arXiv:1206.3881 [cs, stat]*, June 2012. arXiv: 1206.3881.
53. Peter Grassberger and Itamar Procaccia. Measuring the strangeness of strange attractors. *Physica D: Nonlinear Phenomena*, 9(1):189–208, October 1983. ISSN 0167-2789. doi: 10.1016/0167-2789(83)90298-1.
54. Joshua B. Tenenbaum, Vin De Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.
55. Jose Costa and Alfred Hero. Manifold Learning with Geodesic Minimal Spanning Trees. *arXiv:cs/0307038*, July 2003. arXiv: cs/0307038.
56. Laurens Van Der Maaten, Eric Postma, and Jaap Van den Herik. Dimensionality reduction: a comparative. *J Mach Learn Res*, 10:66–71, 2009.
57. Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.
58. Mattia Rigotti, Daniel Ben Dayan Rubin, Xiao-Jing Wang, and Stefano Fusi. Internal representation of task rules by recurrent dynamics: the importance of the diversity of neural

responses. *Frontiers in Computational Neuroscience*, 4(24):29, 2010. ISSN 1662-5188. doi: 10.3389/fncom.2010.00024.
59. Mattia Rigotti, Daniel Ben Dayan Rubin, Sara E. Morrison, C. Daniel Salzman, and Stefano Fusi. Attractor concretion as a mechanism for the formation of context representations. *Neuroimage*, 52(3):833–847, September 2010. doi: 10.1016/j.neuroimage.2010.01.047.
60. Zachary Chase Lipton. A critical review of recurrent neural networks for sequence learning. *CoRR*, abs/1506.00019, 2015.
61. Paul J Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.
62. R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training Recurrent Neural Networks. *ArXiv e-prints*, November 2012.
63. Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553): 436–444, May 2015. doi: 10.1038/nature14539. WOS:000355286600030.
64. J. Collins, J. Sohl-Dickstein, and D. Sussillo. Capacity and Trainability in Recurrent Neural Networks. *ArXiv e-prints*, November 2016.

Methods

Nonlinear dimensionality: Intrinsic Dimensionality While research on estimating intrinsic dimensionality ID is advancing, there is still no single 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 (30, 31). The set of techniques we use include: $\mathrm{MiND_{ML}}$ (50), MLE (51), DancoFit (52), CorrDim (53) and GMST (54, 55). These techniques follow the selection criteria illustrated in (30), 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 (56) for a useful comparison. We implement these techniques using the code from the the authors available online (30, 51, 52), "out of the box" without modifying hyperparameters.

Neural network model We study a Recurrent Neural Network (RNN) that generates predictive neural representations of hidden states during the exploration of partially observable environments. RNNs are suited to processing sequence-to-sequence tasks (57), i.e. to generating sequences of outputs (here, the sequence of future observations) upon receiving sequences of inputs (here, the sequences of observations and actions). This is achieved by exploiting internal recurrent units in the network whose activity is updated as a function of their state at the previous time step, together with the current input. The state of a recurrent network is thus a function of the history of previous observations, and can be exploited by the readout to learn contextually appropriate responses to a new given input (58–60).

Figure 2c illustrates our RNN model. In more detail: At a given time t the RNN receives as input an observation vector $\vec{o}$ and a vector representation of the action $\vec{a}$. The internal state $\vec{r}^t$ of the network is updated and used to generate the network's output through Eq. (5). The RNN is trained to predict the observation at the next time step by minimizing the first cost function in Eq. (6).

Description of the environment We consider a navigation task in two dimensions. We simulate the navigation of the agent in a square maze tessellated by a grid of evenly spaced cells (64x64=4096 locations). At every time t the agent is in a given location in the maze and heads in a direction $\varphi \in [0, 2\pi)$. The agent executes a random walk in the maze which is simulated as follows. At every step in the simulation an action is selected by updating the direction variable θ stochastically with $d\theta$ (*i.i.d.* sampled from a Gaussian distribution with variance $\sigma^2_{theta} = 0.5$ rad), Fig.2b inset. The agent then attempts a move to the cell, among the 8 adjacent ones, that is best aligned to θ. The move occurs unless the target cell is occupied by a wall, in which case the agent remains in the current position.

The chosen action is encoded in a one-hot vector that indexes the movement. Note that the actions are discrete choices $a_t \in [0..8]$ a set related to but distinct from the direction, which is a continuous variable $\theta_t \in [0, 2\pi)$. Moreover, knowledge of the action doesn't provide direct information about the agent's direction, as for each location and action there are many possible directions the agent may point towards and consequently as many possible observations.

As the agent explores the environment it collects, through a set of 5 sensors, the distance and color of the walls along 5 different directions equally spaced in a 90 degree visual cone centered at φ. Thus it records, for each sensor, four variables at every time step: the distance from the wall and the RGB components of the color of the wall. This information is represented by a vector o_t of size 5x4=20 as shown in Fig.2D. Such a vector, together with the action encoded through a $0-8$ one-hot representation, is fed as input into the network and used for the training procedure. The walls are initially colored so that each tile corresponding to a wall carries a random color (i.e. three uniformly randomly generated numbers in the interval [0,1]). A Gaussian filter of variance 2 is then used, for each color channel, to make the color representations smooth. Fig. 2b shows an example of such an environment.

Description of the network training We train the connections in our RNN by minimizing the cost function in Eq. (6) via backpropagation through time (61). While RNNs are known to be difficult to train in many cases (62), a simple vanilla RNN model with hyperbolic tangent activation function is able to learn our task, Fig. 1d

The connectivity matrix of the recurrent network is initialized to the identity (63, 64), while input and output connectivity matrices are initialized to be normally distributed random matrices. The network has 500 recurrent units (with the exception noted below), while the input and output size depend on the task as described in the description of the environment. Each epoch of training corresponds to $T = 10^6$ time steps.

We train the network through the optimizer RMSprop (though we checked that this specific choice does not influence our main results). Learning proceeds through successive epochs until the cost function fails to diminish in value for 25 consecutive epochs. For the simulations of Fig. 5 we trained 100 networks of 100 neurons: 50 networks in the predictive case (cost function $\mathcal{C} = \frac{1}{T}\sum_{t=0}^{T-1} ||\vec{o}^{t+1} - \vec{y}^t||^2$, cfr. Eq. (6)) and 50 networks in the non-predictive case ($\mathcal{C} = \frac{1}{T}\sum_{t=0}^{T-1} ||\vec{o}^t - \vec{y}^t||^2$).

The specific parameters adopted for the training of the

recurrent network are: input weights $\sim \mathcal{N}(0, 0.02)$, output weights $\sim \mathcal{N}(0, 0.02)$, RMSprop learning constant 0.0001, RMSprop $\alpha = 0.95$, RMSprop ϵ regularizer $1 \cdot 10^{-7}$.