Signatures and mechanisms of low-dimensional neural predictive manifolds

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

***For correspondence:** stefanor@uw.edu (FMS)

[†]These authors share senior authorship

[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; [6]IBM Research AI; [7]Allen Institute for Brain Science; Seattle, WA

Abstract Many of the recent advances of neural networks in sequential tasks such as natural language processing applications hinge on the use of representations obtained by predictive models. This success is usually ascribed to the emergence of neural representations that capture the low-dimensional latent structure implicit in the task. Motivated by the recent theoretical proposal that the hippocampus performs its role in sequential planning by organizing semantically related episodes in a relational network, we investigate the hypothesis that this organization results from learning a predictive representation of the world. Using an artificial recurrent neural network model trained with predictive learning on a simulated spatial navigation task, we show that network dynamics exhibit low dimensional but non-linearly transformed representations of sensory input statistics. These neural activations that are strongly reminiscent of the place-related neural activity that is experimentally observed in the hippocampus and in the entorhinal cortex. We quantify these results using measures of intrinsic dimensionality, which indeed confirm that the neural representations obtained with predictive learning reflect the low-dimensional latent structure of the spatial environment underlying the sensory input presented to the network. Moreover, the *dimensionality gain* of the neural representations, a measure of the discrepancy between linear and intrinsic dimensionality, allows us to follow how this process evolves as learning unfolds. Finally, we provide theoretical arguments as to how predictive learning can extract the latent manifold underlying sequential signals, and discuss how our results and methods can aid the analysis of experimental data.

Introduction

The scientific understanding of the role of the hippocampus is traditionally dominated by two distinct theories: the *declarative memory view*, which equates hippocampal function with our ability to recall facts and experiences (***Cohen and Squire, 1980***), and the *spatial navigation view*, which ascribes to the hippocampus a central role in navigation, that of planning routes through physical space (***O'Keefe and Dostrovsky, 1971***). Recently, considerable effort has been devoted to trying to reconcile these apparently contrasting views (***Buzsáki and Moser, 2013***; ***Milivojevic and Doeller, 2013***; ***Eichenbaum and Cohen, 2014***; ***Schiller et al., 2015***). In particular, ***Eichenbaum and***

Cohen (*2014*) proposed that the hippocampus supports a *semantic relational network* that organizes semantically related episodes to subserve sequential planning.

But how does such an organization of semantic information emerge? Two related bodies of work have shown that this can occur thanks to the process of prediction. First, neural networks have been successfully used to extract semantic characteristics from linguistic corpora simply by training them to predict the context (i.e., the adjacent words) in which a given word appears (*Bengio et al., 2003*; *Turian et al., 2010*; *Collobert et al., 2011*; *Mikolov et al., 2013a*). The resulting neural representations of words (known as *word embeddings*) have intriguing geometric properties that reflect the *semantic meaning* of the words they represent, and made them an invaluable component in many applications in machine learning and computational linguistics. Of relevance for our work, this has been explained by postulating that linguistic corpora are being generated by a dynamical process over a latent low-dimensional "discourse space" that predictive learning is able to uncover (*Arora et al., 2015*). Second, following up on classic work by *Dayan* (*1993*), several recent papers have demonstrated that neural models trained to predict future sensory information can give rise to internal representations that encode spatial maps useful for goal-directed behavior (*Stachenfeld et al., 2014*; *Russek et al., 2017*; *Wayne et al., 2018*).

Taking inspiration from these lines of work, we set out to investigate whether predictive learning could serve as a computational mechanism for the synthesis of semantic information that *Eichenbaum and Cohen* (*2014*) attributed to the hippocampus.

Our goal here is to build theoretical and data-analytic tools that explain why a *prediction 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.

The present work starts from a generative model perspective, whereby observations in a task environment are being generated from latent variables embedded in a *low-dimensional manifold*. In the case of spatial navigation the latent variables are for instance the position and orientation of the subject in the spatial environment, which can only be indirectly observed via the observations that they generate, i.e. the first-person sensory inputs (visual, etc.) corresponding to that location and orientation in space. We then set out to verify our hypothesis that a predictive learning process over the sequence of high-dimensional sensory inputs extracts representations that meaningfully represent the underlying low-dimensional latent variables. We do this in the context of an RNN trained to predict future observations in the environment it is navigating.

latent variables

In order to be able to verify our main hypothesis, we first have to develop the right analytical tools to correctly measure the *intrinsic dimensionality* of the vector representations created by predictive learning and expose their low-dimensional structure. Crucial to this development is the distinction between linear (*Rigotti et al., 2013*; *Mazzucato et al., 2016*; *Litwin-Kumar et al., 2017*; *Gao et al., 2017*) and nonlinear dimensionality (*Camastra and Staiano, 2016*; *Campadelli et al., 2015*), which allows us to uncover a phenomenon that we call *latent space signal transfer*, wherein information about latent variables moves into the top principal components of the activity as learning progresses. This signature is tightly linked with a clear trend in the linear and nonlinear dimensionality of the formed manifold, and with the formation of localized neural fields on the manifold itself. We refer to neuron with such localized activations as *manifold cells* (*Low et al., 2018*). Importantly, all of these signatures can be applied to data from biological or machine learning experiments.

The structure of our paper is the following. We start by analyzing the consequences of our hypothesis that predictive learning extracts the low-dimensional latent structure underlying some high-dimensional sensory signals. This is done in Sec. 1 where we study artificially constructed neural representations encoding a low-dimensional set of latent variables. In particular, we examine a population of neurons each tuned to a particular location in space. Importantly, we show that the use of nonlinear dimensionality reduction techniques is crucial to reveal the low-dimensional latent structure in these neural representations, while standard linear measures of dimensionality would actually give the illusory impression of high dimensionality. In particular, it motivates a quantity

measuring the discrepancy between intrinsic and linear dimensionality that we call *dimensionality gain*.

In Sec. 2, we then show how low-dimensional latent coding can arise through learning. In particular, we show that this can emerge in an RNN trained with *predictive learning* to anticipate future observations in a simulated navigation task of a simple 2-D environment. Interestingly, this is not the case for similar networks that are trained to auto-encode (i.e. compress) their inputs, but do not predict them over time (Sec. 5). In Sec. 3 we dive into the analysis of the learned neural representations, and in Sec. 4 we provide general theoretical arguments linking predictive learning with the extraction of the low-dimensional latent space in a task.

1. Latent and neural representation spaces

In this section we build a model displaying a basic phenomenon that we refer to as low-D coding: that there is a small set of environmental or latent variables to which a large number of neurons are strongly and consistently tuned. A well-known example of this is given by place and grid cells in the context of hippocampal navigation (***O'Keefe and Dostrovsky, 1971***; ***Solstad et al., 2008***; ***Stensola et al., 2012***; ***Wills et al., 2010***). This indeed will be our case study in the following, but it is important to stress that our considerations are valid more in general and an analogous analysis can be carried out for other cases such as orientation selective visual neurons or hippocampal time cells.

We consider an ensemble of N place cells with Gaussian tuning curves that are uniformly distributed over the locations of a given environment, such that every location in the environment uniquely corresponds to an evoked neural population response pattern. In other words, we can thing of the Cartesian coordinates of a position in the environment (x and y) as latent variables that fully describe an agent's state in the environment, and give rise to the neural response patterns that are being observed. Accordingly, a navigation path through the environment describes a trajectory as shown in Fig. 1a, where each location of the environment is colored in a unique way for the sake of presentation. Note that, under our assumptions, the place cells give the agent perfect knowledge of its location and do not depend on past experience.

An example of a Gaussian tuned neural field is shown in Fig. 1b. If the agent is located in position $\boldsymbol{x}_0 = (x_0, y_0)$ then the activity r_i of neuron i with preferred location (x_i, y_i) will be given by $\mathcal{G}_\sigma(x_0 - x_i, y_0 - y_i) = \frac{1}{2\pi\sigma} \exp\left(-\frac{(x_0-x_i)^2+(y_0-y_i)^2}{2\sigma^2}\right)$. We refer to the vector of activities $\boldsymbol{r}_0$ of all neurons at that specific point in space as the *neural representation* at location $\boldsymbol{x}_0$.

As the agent navigates the environment, describing a trajectory $\boldsymbol{x}_t$ in the 2d latent space, the representation $\boldsymbol{r}_t$ traces out a trajectory in neural space; that is, the N-dimensional space spanned by the activity of all neurons in the population. A common way of visualizing this is by projecting the trajectory into a lower-dimensional space spanned by the first three Principal Components (PCs). We show this projection in Fig. 1c, together with the *representation manifold*, the full set of neural representations over the entire environment. We color every point on the representation manifold according to its corresponding location in the environment (or latent space variable $\mathcal{X}, \mathcal{Y}$). The two dimensions of this latent space completely parameterize the manifold, meaning that it is a two-dimensional curved surface. The fact that the representation manifold has two dimensions is revealed by a measure that is usually referred to as Intrinsic Dimensionality (ID), whose formal definition relies on concepts in Riemannian Geometry for smooth manifolds or statistics for statistical manifolds (***Camastra and Staiano, 2016***). In Fig. 1d we show the tuning curve of a single neuron on the manifold. In Sec. 3 we will analyze in more depth the meaning of such tuning of individual neurons with respect to manifold parameters. In our analysis we limit ourselves to analyzing neural tuning to manifold variables in the form of localized activations, like in Fig. 1d.

While the ID of the representation manifold is two, due to its curvature many more linear components are necessary to fully describe it in the N-dimensional neural ambient space. This discrepancy between linear dimensionality, vs. nonlinear dimensionality as measured by ID, is an important phenomenon. In general, a curved d-dimensional manifold requires more than d

dimensions to be embedded into a Euclidean space. More than d principal components are needed to capture the variance of such a manifold. Nonlinear dimensionality reduction techniques attempt to account for curvature in attaining a more accurate estimate of d. While many such techniques exist (cfr. ***Van Der Maaten et al. 2009***), we use isomap as an example, which is capable of displaying samples from a manifold in lower dimensions while preserving, as much as possible, the geodesic distance between these samples as computed along the manifold in the original space, i.e. the N-dimensional space of the neural representation (***Tenenbaum et al., 2000***). The representation manifold following this reduction is shown in Figs. 1e and 1f where we see that, by extracting the manifold from the neural representation, the original space has been almost perfectly recovered.

MDS: perserves euclidean distances to map into lower dimension space

isomap preserves geodistance distances
graph based distances.
Captures curvature!

find k nearest neighbors
do this for every point to form a graph
connecting these points with edge weight of euclidean distance

find shortest path between two points using s

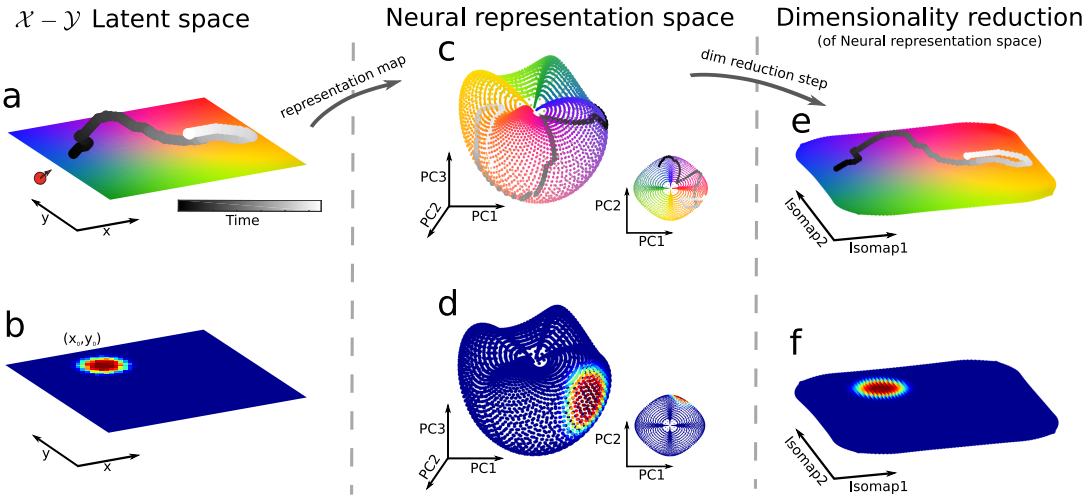

Figure 1. Manifold analysis example. a) Example of a two dimensional environment in which the agent moves. We assign a unique color to each location of the environment. A segment of the agent's trajectory is represented in gray scale, with shade standing for time. b) Example tuning of a neuron with gaussian 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. 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) Example of the manifold recovered from the neural representation by means of the Isomap technique. The manifold embedding dimension is two and the agent's trajectory is shown once again. f) Manifold receptive field: same as panel e but for the neuron receptive field.

Now we focus on characterizing the properties of neural representations when analyzed by means of linear versus nonlinear techniques. The number of PCs needed to capture a given percentage of the variance of a manifold is a measure of the linear dimensionality of the manifold. A closely related measure uses the Participation Ratio (PR) of the eigenvalues $\lambda_{1..N}$ of the covariance matrix $\boldsymbol{C}$ to measure dimensionality:

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

where $\tilde{\lambda}_i = \lambda_i / \sum_{j=1}^{N}\lambda_j$, see Fig. 2a (***Gao et al., 2017***). 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 $\mathrm{PR}(\boldsymbol{C}) = N$. 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 $\mathrm{PR}(\boldsymbol{C}) = M$ (Fig. 2a). For other PCA eigenspectra, this measure interpolates between these two regimes. As a rule of thumb, the PR dimensionality can be thought as the number of dimensions required to explain about 80% of the total population variance in many applications (***Gao et al., 2017***). PR (Participation Ratio) as a linear measure of dimensionality, in contrast with nonlinear ID (Intrinsic Dimensionality).

The PR dimensionality for the representation manifold induced by place cells (Fig. 1) is shown in Fig. 2. The covariance matrix induced by Gaussian receptive fields with standard deviation $\sigma = 2.5$ is shown in the inset of Fig. 2d. This matrix has a diagonal band structure and within this structure each element is a matrix with a diagonal band. It is a matrix of matrices which reflects the 2d structure of the latent space $\mathcal{X}, \mathcal{Y}$.

The PR as a function of the number of neurons or number of points sampled from the manifold is shown in Fig. 2b. This demonstrates the effect of having, as under empirical sampling, fewer neurons or samples (trials). This shows that for the case at hand, a few hundred neurons is sufficient to estimate PR at a value close to its converged limit.

In the Methods we compute the PR as a function of the tuning curve width σ, showing that the PR is inversely proportional to σ. Smaller widths correspond to higher curvature of the response manifold, and hence to higher PR values. This gives a clear illustration of how the linear linear notion of dimensionality via PR depends heavily on the coding properties of single neurons. Later on we will apply methods that estimate the intrinsic dimensionality ID of the manifold from data (***Camastra and Staiano, 2016***; ***Campadelli et al., 2015***); these return values closer to the true dimensionality of the manifold, in terms of the number of its parameters. Thus, while ID is an estimation of the number of variables needed to chart the neural representation manifold, PR appears as a measure of how many coordinates the neural representation is exploiting to represent it.

We suggest the following metric to measure the extent to which a given representation linearly expands the "true" dimensionality of the manifold, which we call *Dimensionality Gain* (DG):

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

In Fig. 2c we show the Dimensionality Gain (DG) as a function of the width σ for the example of Fig. 1. The graph shows how the DG decreases as the width of the fields increases (red line). This is trend is in agreement with the theoretical analysis (blue line). In the following we illustrate how DG be used to assess properties of more complex, learned neural representations.

2. Predictive Learning

In the previous section we illustrated the relationship between latent variable space and neural representation space when neurons function as place cells, so that neurons directly encode the latent space. This led to interesting and readily measurable phenomena: the representation manifold is low-dimensional while appearing higher-dimensional according to linear measures: that is, the representation has a high dimensionality gain (DG). This begs a key question: which kind of learning processes can *generate* representations with such properties – and does this occur when processing naturalistic sensory inputs? In what follows we provide both simulation evidence and theoretical arguments for *predictive learning* in recurrent networks (RNNs) being a basic framework that forms neural representations with the properties at hand. In predictive learning the network is trained to minimize the prediction errors between its output and future sensory observations.

We turn our attention to the representations that are formed by a recurrent neural network (RNN) learning to represent its environment by predicting sensory-like observations. In this case, the RNN agent does not have direct access to its location, but instead has access to "sensory" observations (Fig. 3b) of its environment. The agent performs a random walk in its environment by updating, at each step, its direction θ by an angle $d\theta$. This change in direction $d\theta$ is *i.i.d.* sampled from a wrapped Gaussian distribution with variance σ^2_{theta}, cfr. Fig. 3b inset and Methods for details. The environment is tiled with $64x64 = 4096$ locations, and at every step the agent moves forward to the tile best aligned with the updated direction θ unless its step collides with a border, in which case no movement occurs. An example trajectory is shown in Fig. 3a, where each position in the environment is again identified by a specific color.

The agent is equipped with sensors oriented in the direction θ (see Fig. 3b). The task of the RNN is to predict the sensory observation of the agent on the next time step, given the current sensory

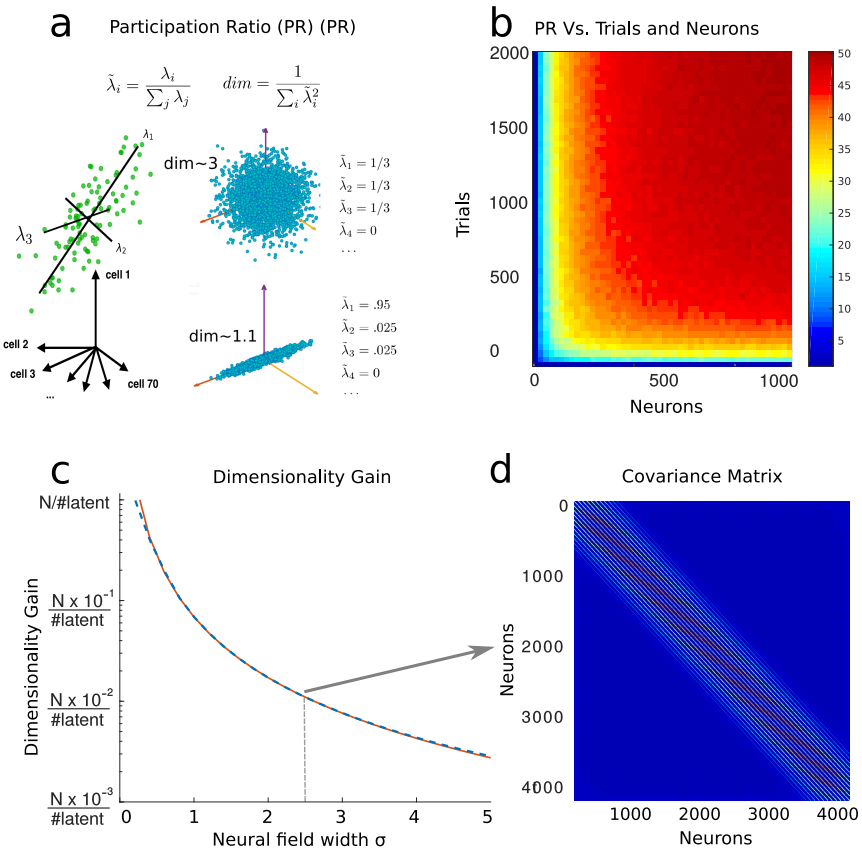

Figure 2. Linear dimensionality analysis. a) Illustration of nthe Participation Ratio (PR) dimensionality measure. The mathematical expression in terms of the eigenvalues of the covariance is given and illustrated for a few distributions in PC space. The left part shows an example of point cloud distribution 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. 1 with $\sigma = 2.5$. c) PR dependence on the size of the gaussian field σ for the example of Fig. 1. The red line represents the DG as computed for 4096 neurons tiling the latent space shown in Fig. 1. The blue dotted line represents the theoretical analysis (cfr. Methods). d) Example of the covariance matrix for $\sigma = 2.5$.

observation (see Fig. 3c).

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 representation space. As the RNN learns to predict the next observation, the neural representation will change to better perform the task. This 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 low-dimensional generative model for the observations); *a priori*, it is not obvious which space's influence will be stronger.

The neural representation at the end of learning (see Fig. 3d) represents latent information. This is shown in Fig. 3e, which illustrates that the latent variables are strongly represented in the neural representation space after learning. Further, single neurons' receptive fields function as place and border cells encoding the latent variables x, y, and as head direction cells encoding θ (Fig. 3f). This shows that the internal representation of the network has naturally extracted information about the latent space from the observations, without being explicitly prompted to do so. As we will show below this phenomenon relies on the underlying task being predictive. We first highlight important properties of the learned representation manifold.

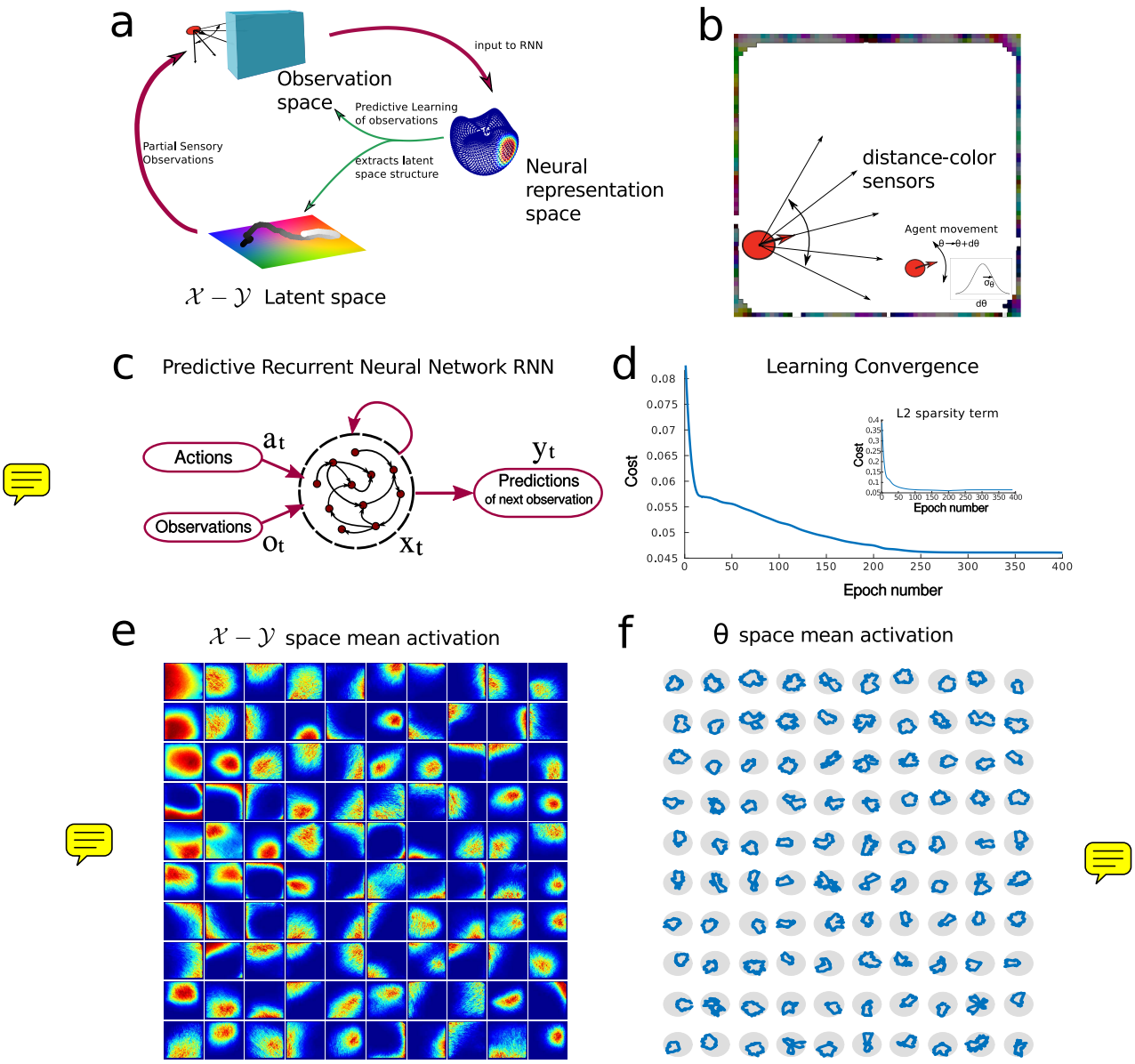

Figure 3. Predictive network solving a navigation task. a) Logic diagram of task and information: an agent explores a latent space through actions and receives partial observations regarding it. The network's task is to predict the next sensory observation. By learning to do so it recovers information 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 inset illustrates the agent navigation driven by θ. θ is updated continuously and updates are drawn from a gaussian distribution (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 (cfr. Sec. 4 and Methods). The inset shows the L_2 norm of the activations computed during training on the representation (although this is not used as a regularizer). e) Place cell activities: average activity of 100 neurons (one per small quadrant) against the $\mathcal{X}, \mathcal{Y}$ 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.

3. The learned neural representation manifold and its signatures

As the network learns to predict future observations it may be expected that most of the network activity is dedicated to encoding features of the observation space. The natural consequence is that the leading PC components of the RNN representation carry information about observation space variables. On the other hand, the network develops place cells (Fig. 3e) which suggests that the

latent spaces is also strongly encoded. As we will see next, it is indeed the latent space variables that are most strongly encoded in the first PCs of RNN activity. The latent space for the navigation task is parametrized by x, y, θ. In Fig. 4a we show the RNN representation projected into the space of the first three PC components of the RNN neural activity, colored according to each of these three latent variables. That is, each point in these plots corresponds to the RNN 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 visualization clearly 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.

As the agent's input are the observations rather than the latent variables, it is natural to ask whether the observation variables are similarly encoded in the RNN representation. Fig. 4c shows that this is not the case. The first three PCs don't appear to be encoding for the average, across sensors, of color sensory information for the three color channels RGB. They do encode for the distance (as they also encode for the position) but not for color. Later, in Fig. 5d, we will further justify and quantify this observation. We will also show that average color sensory information is encoded in the first PCs in the beginning of learning while it is not clearly encoded in the final learned representation as shown in Fig. 4c.

Figs. 4a and 4b, taken together, suggest that the RNN allocates most of its internal variability to the encoding of latent variables. In this specific example the first five PC components explain respectively $13.7\%, 11.4\%, 10.2\%, 5.5\%, 5.4\%$ of the total variance in the activity of the RNN population. We next explore the relationship between the responses of single cells and the population activity along the manifold. In the simplest case of Fig. 1, 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. 4d demonstrates that this is indeed the case, by showing the activity of the same 100 neurons shown in Fig. 3e 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 name these units *neural manifold-cells*. If the neural manifold clearly represents the latent space (Fig. 4a) and neural receptive fields tile the latent space (Fig. 3e), then neural activities are also localized on the manifold. We observe that the reverse is also true: localized activities in the latent space (e.g. place cells, cfr.Fig. 3e) can be interpreted as a result of single neural receptive fields tiling the manifold. In our analysis single neurons appear to have localized activations and do not develop other patterns of activity such as grid-like activations. The extention of our analysis to grid-like representations is beyond the scope of our present contribution although the tools here introduced would directly applied.

The preceding analysis suggests that neural representation manifold and single neuron coding are tied to one another, as they are both linked to the latent space. We proceed to study how the manifold and its connection to the latent space emerge over the course of predictive learning.

In Fig. 3 we highlighted two different ways to access the dimensionality of the representation: a linear measure (Participation Ratio, PR) and a nonlinear one (Intrinsic Dimensionality, ID). The PR of the representation is shown in Fig. 5a. This measure is sensitive to the neural activation on the manifold as described in Sec. 1. The PR increases as the receptive fields become more local. The PR, computed at every training epoch for $5 \cdot 10^5$ navigation steps, keeps increasing epoch after epoch, and the slow increase corresponds to the formation of place cells with respect to the latent space (Fig. 3e) and manifold cells with respect to the representation manifold (Fig. 4d). While the PR increases across epochs, all estimators of the manifold's ID decrease until they reach a value of approximately 5 (Fig. 5b; see also Methods). Recall from our analysis in Sec. 1 that the value of ID is independent of single neuron fields. Although we cannot explain this number precisely, we note that if the latent variables are encoded then it cannot be less than 3, the number of latent components (x, y, θ). Moreover, ID is considerably smaller than PR, pointing to a dimensionality

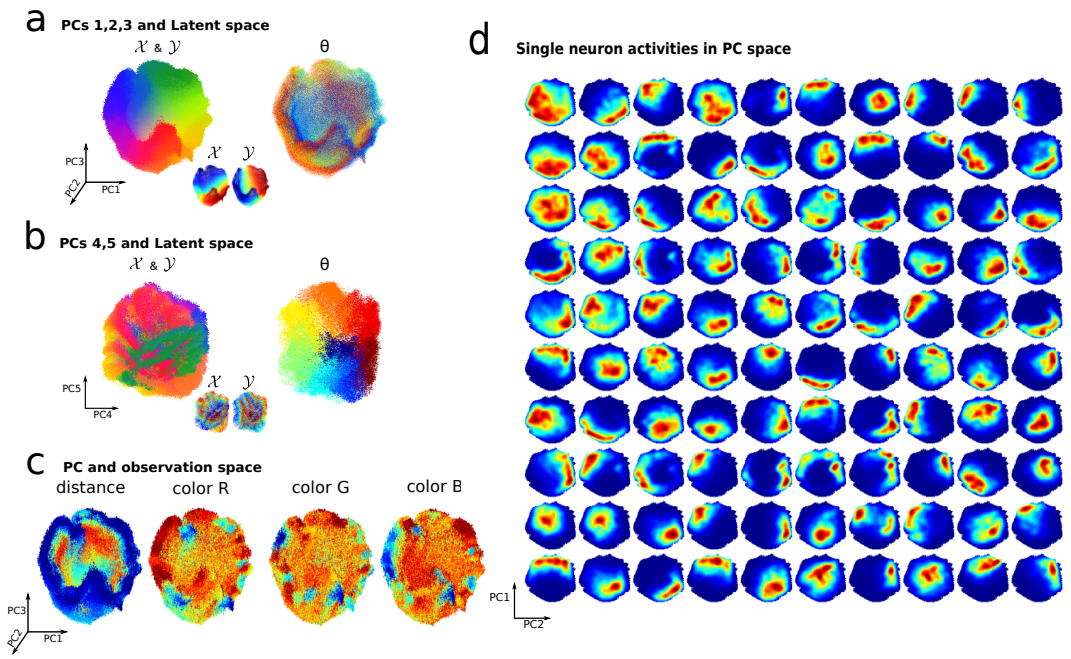

Figure 4. Signatures of the learned predictive representation. a) 100000 points of the neural network representation, corresponding to an equal number of steps for the agent's exploration, are shown projected into the space spanned by by PCs 1 to 3 of the learned representation, and colored respectively with respect to $\mathcal{X}, \mathcal{Y}$ latent variables (cfr. Fig. 1a 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. 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.

gain DG of roughly $\mathrm{DG} = \frac{\mathrm{PR}}{\mathrm{ID}} \approx 3$ toward the end of learning. This is consistent with our analysis of Sec. 1 where we showed that local manifold fields tend to increase the DG.

In Figs. 4a and 4b we showed that the first five PCs of the learned representation are highly correlated with latent space variables. This 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. 5c 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 CCA 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 $\mathcal{X}, \mathcal{Y}, \theta$. Fig. 5c 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 very same analysis can be carried out with respect to observation space variables. This is shown in Fig. 5d. The observation space signal *flows out* of the first few PC components as learning progresses. Together Figs. 5c and 5d show that the total variability of the representation, as interpreted through PC components, encodes more latent space information vs. observation space information as learning progresses (blue and red lines). Altogether Fig. 5 suggests that predictive learning, throughout training, forms a low-dimensional representation (Fig. 5a), with properties (the high linear dimensionality) that facilitate its linear readout (Fig. 5b).

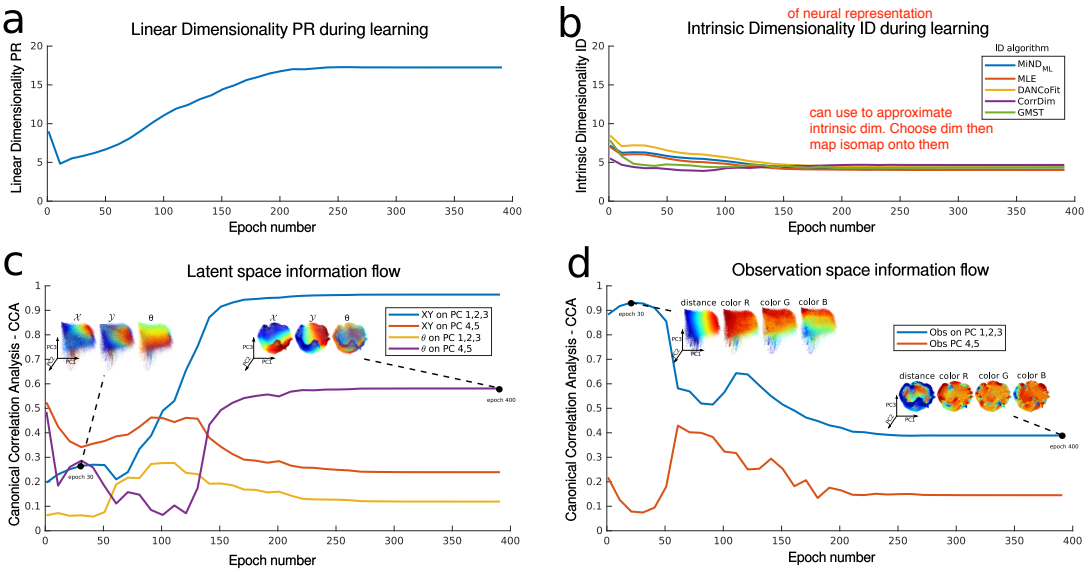

Figure 5. Learning the predictive representation. a) Participation Ratio of the representation during learning. b) Intrinsic Dimensionality (ID) of the representation during learning. Five different intrinsic dimensionality estimators are used (cfr. Methods). c) Signal transfer analysis: Canonical Covariance Analysis between PCs of the neural representation and the latent space. d) Same as panel c) but for the observation space.

4. A neural network mechanism for learning low-D latent manifolds

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

For simplicity, let us suppose that the movement of our agent in the latent space $\mathcal{X}$ is governed by a deterministic, 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 $\mathcal{X}$. 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 $\boldsymbol{x}_t$: $\boldsymbol{o}_{t+1} = \varphi(\boldsymbol{x}_t + F(\boldsymbol{x}_t))$. We choose a point $\boldsymbol{x}^* \in \mathcal{X}$ around which to expand φ to get:

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

where $D\varphi(\boldsymbol{x}^*)$ is the Jacobian matrix of φ evaluated at $\boldsymbol{x}^*$. In the above, we assume that the trajectory $\boldsymbol{x}_t$ stays close to $\boldsymbol{x}^*$ so that the linear regime dominates and higher order terms can be neglected. This may only hold momentarily so that this linearization remains a local approximation (more on this below).

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

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

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. Methods for further details). The local

neural

dynamics in latent space induces a dynamics in representation space that is sketched in Fig. 6a.

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 (4) and (5) to obtain

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

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 (7) into

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

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 $3 \times T$ 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 (8) to minimize $\mathcal{C}_{pred}$ is equivalent to solving the ordinary least squares problem:

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

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, it is easy to see that X is rank 3 and, assuming $\boldsymbol{W}_{out}$ and $\boldsymbol{W}$ are of higher rank, a natural way to satisfy this is by R also being rank 3. This is consistent with low-dimensional network dynamics. The latent space dynamics induces a dynamics on the representation space which is locally in direct

Although these relations do not hold in the general nonlinear case it is reasonable to think that they may hold in an approximate way. For instance, 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. Indeed, Fig. 6b shows that the matrix $\boldsymbol{W}_{out}\boldsymbol{W}_{in}$ has a clear diagonal structure. This suggests that the 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

353 $||\boldsymbol{o}_{t+1} - \boldsymbol{y}_{t+1}||^2 = ||\boldsymbol{o}_t - \boldsymbol{W}_{out}\boldsymbol{W}\boldsymbol{r}_{t-1} - \boldsymbol{W}_{out}\boldsymbol{W}_{in}\boldsymbol{o}_t||^2$ are missing the dynamic update and cannot be
354 decomposed as in (7). The absence of the low-dimensional latent space dynamics in this non-
355 predictive settings suggests that the representation shouldn't "discover" the latent manifold through
356 learning. We demonstrate this explicitly in the next section.

357 The series of arguments presented above is meant to provide intuition about how predictive
358 learning may extract a representation of the latent space. We stress that this is not a formal
359 derivation and its limitations should be kept in mind. The extracted manifold can be pictured, cfr.
360 Fig. 6b, as a low dimensional curved manifold in the high dimensional neural space.

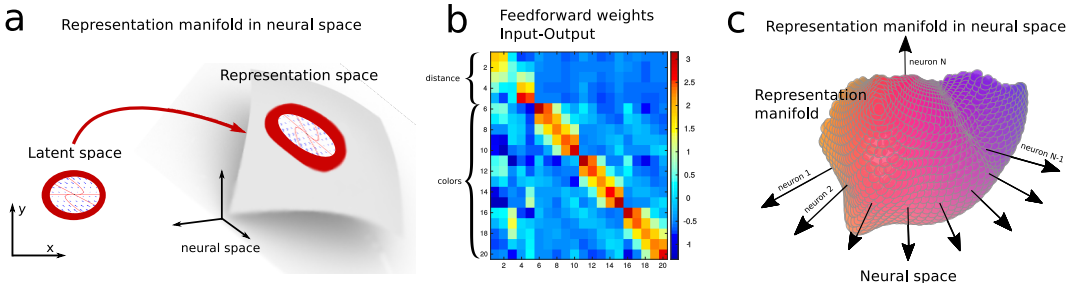

Figure 6. Theoretical arguments. a Neighborhood projection of the local dynamical system between latent and neural representation space. b Feedforward connections that pass input observations to outputs: matrix of weights $W_{out}W_{int}$ from predictive learning. c) Representation manifold in neural space: example where the low dimensional manifold spans many neural directions despite being low dimensional.

361 ## 5. Non-predictive learning fails to extract low-D latent manifold

362 A central idea in this article is the importance of the learning being *predictive*, so that the underlying
363 RNN is learning to anticipate the observation on the next timestep into the future. is the predictive
364 aspect itself necessary to produce the phenomena studied above? Here we address this question
365 by directly contrasting predictive learning with the non-predictive case.

366 We train each of 100 RNNs, which differ only in the initialization of their weights and the
367 agent's trajectory, in two different scenarios: predictive learning and recurrent auto-encoding; that
368 is, predicting the next step observation o^{t+1} as described earlier and auto-encoding the current
369 observation o^t (***Hinton and Salakhutdinov, 2006***; ***Vincent et al., 2008***). We find that all networks
370 trained through predictive learning show the same characteristics as outlined above, while the same
371 networks trained with the auto-encoding loss develop different representations. Most importantly,
372 with the auto-encoding loss the learned representations do not reflect the latent state variables
373 and statistics in the same way as for the predictive coding loss.

374

375 In Figs. 7a and 7b we show that the Canonical Correlation Analysis (CCA) between the first three
376 PCs of the representation and the latent space or the observations have completely different trends
377 in the predictive vs non-predictive case. In Fig. 7a the CCA coefficients between the representation
378 and the latent space grows throughout learning (each line corresponds to a different network and
379 the dashed line to the mean) while the coefficients corresponding to observations decrease (cfr.
380 Figs. 5c and 5d). In contrast, by this metric the networks trained to auto-encode the observations
381 did not develop representations that encode the latent space, but rather only the observations.
382 Specifically, throughout training there is little information regarding the latent space encoded
383 in the first PCs of the representation, even though they account for most of the variability of
384 sensory observations. Meanwhile, Fig. 7b also shows that the average CCA coefficients between
385 the representation and the observations are high throughout learning. Consequently, as shown in
386 Fig. 7c the non-predictive representation fails to develop place fields; in particular, the activities of
387 neurons are not localized in the latent space. This is in striking contrast with the same plots for the
388 predictive case.

The dimensionality of the learned representations also differs strongly between the predictive and non-predictive settings. We show this by displaying the PR and ID for networks trained through predictive learning in Fig. 7d, and on the auto-encoding task in Fig. 7e. In the first scenario PR grows and ID decreases throughout training. In the second PR grows but ID does not decrease, as the representation doesn't "extract" the latent manifold. We can summarize these properties by analyzing the Dimensionality Gain (DG) as above; recall that this is the ratio between the PR and the ID (see Methods). Fig. 7f shows that the DG in the predictive case (blue line) progressively increases through learning, while this does not occur for the non-predictive case. Thus, a key signature of encoding of a low-D (latent) space appears for predictive, but not for non-predictive, learing. As shown in Figs. 1 and 3, having a low-dimensional nonlinear structure with a linear high-dimensional representation facilitates both generalization, by means of the representation manifold being low-dimensional, and the reading out (by means of a linear decoder) of the encoded information: this is what the DG expresses, cfr. Sec. 1.

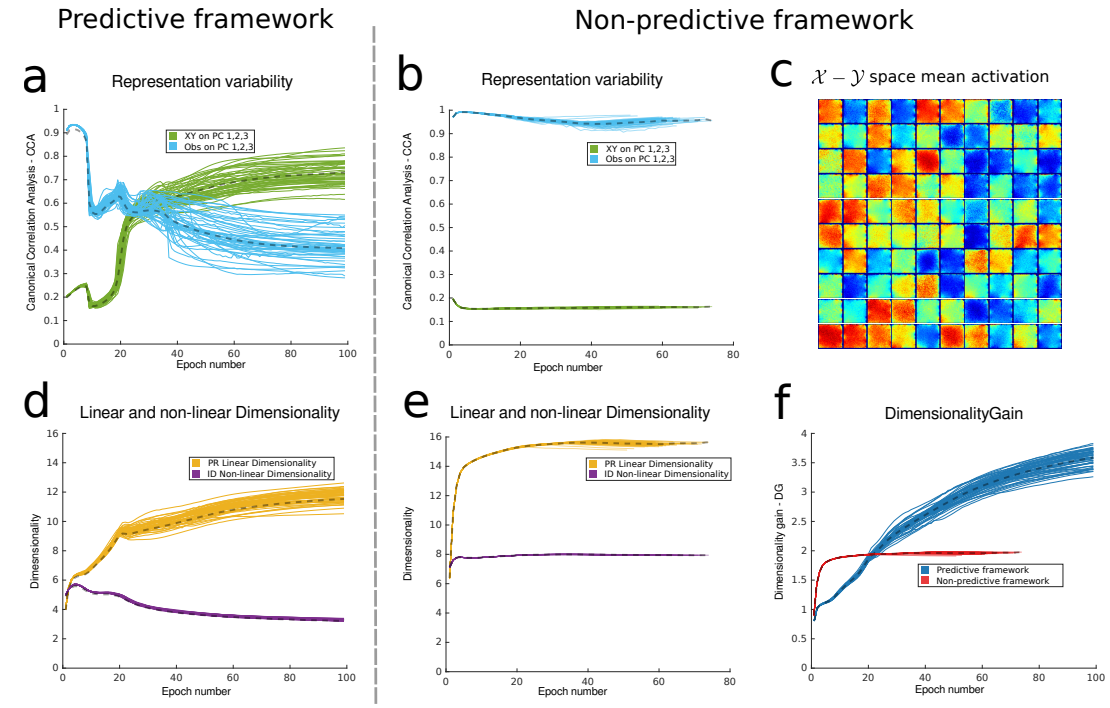

Figure 7. Comparing signatures of learned representations in the predictive vs non-predictive framework. a) Signal transfer analysis: Canonical Correlation Analysis (CCA) between PCs 1 to 3 of the neural representation and the latent or observation spaces during learning. This is displayed for an ensemble of 100 networks (only first 100 epochs shown, cfr Methods). Same as panel b but for the non-predictive case. c) Place cell activations: average activations for 100 cells in the non-predictive case. This is the same plot as for Fig. 4d but in the non-predictive case; note that the neurons do not display localized activations. d) Linear and nonlinear dimensionality for networks trained on predictive learning . e) Linear and nonlinear dimensionality for the non-predictive networks. f) Dimensionality gain for predictive and non-predictive networks throughout learning.

Conclusion and discussion

How the brain extracts information about the external world given only indirect sensory observations is been a long-standing question in neuroscience. Here we propose predictive learning over observations as a computational mechanism to construct neural representations that encode the latent variables underlying the observations and their semantic relation.

We validate our proposal by examining predictive learning in a simulated egocentric spatial navigation task, a situation that is naturally described by latent variables corresponding to the spatial coordinates in the task. Indeed, we verify that the resulting neural representations reflect

the low-dimensional structure of the task and contain responses that are tantalizingly reminiscent of the types of place-related activity famously observed in the hippocampus and entorhinal cortex.

Crucially, in order to reveal this low-dimensional structure, we have to rely on nonlinear techniques that can expose the intrinsic dimensionality of the neural representation manifold, as more common linear measures would give the illusory impression of high-dimensional representations.

In summary, our work gives concrete algorithmic grounding to the recent proposal by ***Eichenbaum and Cohen*** (***2014***) that the hippocampus builds a *semantic relational network* of related episodes at the service of sequential planning. In particular, 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.

Signatures of predictive learning in neural data

What features would one expect to find in biological data from a neural network that is performing predictive learning? As long as the signals that the network is trained to predict arise from an environment with an underlying low-dimensional latent structure, we suggest looking for several distinct signatures. The first signature is the dimensionality of the set of neural responses collected simultaneously across multiple cells, and over multiple task conditions. This dimensionality will likely appear high when assessed with standard linear measures, such as the participation ratio. However, a signature of predictive learning is that it is accompanied by low-dimensional representations, with a dimensionality equaling the number of independent latent encoding variables, when assessed through nonlinear metrics sensitive to the dimensionality of curved manifolds. These two signatures taken together imply a high dimensionality gain (DG), or ratio of linear to nonlinear dimension. The presence of such a low-dimensional *neural representation manifold* opens the door to another signature of predictive learning. Individual cells produce responses which appear strongly tuned when plotted against the (curved) variables lying on the neural representation manifold; we refer to this as the appearance of neural manifold cells (cfr. Fig. 4d). While locality in latent space is an established aspect of neural hippocampal representation in the navigation problem, locality in the manifold is an allied hypothesis that will be exciting to check in experimental data. This builds on recent work on understanding neuronal representations through the lens of representation dimensionality (***Rigotti et al., 2013***; ***Mazzucato et al., 2016***; ***Litwin-Kumar et al., 2017***; ***Cayco-Gajic et al., 2017***). Importantly, manifold-localized activations have also been shown to be optimal for similarity-preserving networks (***Sengupta et al., 2018***; ***Pehlevan et al., 2018***). This points to such signature in the activations as a critical feature of the representation and to similarity-preserving as a possible condition for its emergence. We look forward to further examining how predictive learning could implement this condition.

Discovering latent structure in data and sensory observations

Our results demonstrate that predictive learning can lead to responses lying on a low-dimensional neural representation manifold, with the same dimension as that of the latent space that parametrize the underlying signals that the network has learned to predict. This requires 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 ***Mikolov et al.*** (***2013b***); ***Hinton and Salakhutdinov*** (***2006***); ***Hastie et al.*** (***2009***); ***Weinberger and Saul*** (***2006***). Here, we show that what we call *latent signal transfer* is one way to uncover the relevant variables fig. 4d: 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. Overall, we suggest that such *parametrization* of learning via dimensionality and latent signal transfer may contribute to the understanding of how both biological brains and

neural network algorithms solve difficult tasks such as navigating an environment.

Open questions

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 enormously benefited from the geometric properties of word embeddings learned by predictive models (***Bengio et al., 2003***; ***Turian et al., 2010***; ***Collobert et al., 2011***; ***Mikolov et al., 2013a***). On the other hand, prediction over observations has been used as an auxiliary task in reinforcement learning to acquire representations favoring goal-directed learning (***Dayan, 1993***; ***Stachenfeld et al., 2014***; ***Russek et al., 2017***; ***Wayne et al., 2018***).

Distinctive to our work, is the use of nonlinear dimensionality analysis of the learned representations to characterize the relationship between the neural representation manifold and the latent space, and the use of the measure of dimensionality gain to follow the evolution of this relationship as learning progresses. Nevertheless, more work is needed to theoretically formalize the phenomena that we have demonstrated in simulation.

Perhaps foremost, the way the properties of the representations that are extracted by predictive learning depend on the neural architecture and the implementation of the training algorithm needs to be systematically studied. Moreover, predictive learning is a general framework that goes beyond the example of navigation analyzed here and can be expanded to many different scenarios and behavioral tasks.

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 over long timescales such that the evolution of the neural representation induced by learning can be followed over time with metrics such as the dimensionality gain, and latent signal transfer. A very exciting possibility is that this exercise might uncover the presence of relevant latent variables in a task that were previously unsuspected.

References

Arora S, Li Y, Liang Y, Ma T, Risteski A. Rand-walk: A latent variable model approach to word embeddings. arXiv preprint arXiv:150203520. 2015; .

Bengio Y, Ducharme R, Vincent P, Jauvin C. A neural probabilistic language model. Journal of machine learning research. 2003; 3(Feb):1137–1155.

Buzsáki G, Moser EI. Memory, navigation and theta rhythm in the hippocampal-entorhinal system. Nat Neurosci. 2013 Feb; 16(2):130–138. http://dx.doi.org/10.1038/nn.3304, doi: 10.1038/nn.3304.

Camastra F, Staiano A. Intrinsic dimension estimation: Advances and open problems. Information Sciences. 2016 Jan; 328:26–41. http://www.sciencedirect.com/science/article/pii/S0020025515006179, doi: 10.1016/j.ins.2015.08.029.

Campadelli P, Casiraghi E, Ceruti C, Rozza A. Intrinsic Dimension Estimation: Relevant Techniques and a Benchmark Framework. Mathematical Problems in Engineering. 2015; https://www.hindawi.com/journals/mpe/2015/759567/, doi: 10.1155/2015/759567.

Cayco-Gajic NA, Clopath C, Silver RA. Sparse synaptic connectivity is required for decorrelation and pattern separation in feedforward networks. Nature Communications. 2017; 8(1):1116.

Ceruti C, Bassis S, Rozza A, Lombardi G, Casiraghi E, Campadelli P. DANCo: Dimensionality from Angle and Norm Concentration. arXiv:12063881 [cs, stat]. 2012 Jun; http://arxiv.org/abs/1206.3881, arXiv: 1206.3881.

Cohen NJ, Squire LR. Preserved learning and retention of pattern-analyzing skill in amnesia: dissociation of knowing how and knowing that. Science. 1980 Oct; 210(4466):207–210.

Collins J, Sohl-Dickstein J, Sussillo D. Capacity and Trainability in Recurrent Neural Networks. ArXiv e-prints. 2016 Nov; .

Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Kuksa P. Natural language processing (almost) from scratch. Journal of Machine Learning Research. 2011; 12(Aug):2493–2537.

Costa J, Hero A. Manifold Learning with Geodesic Minimal Spanning Trees. arXiv:cs/0307038. 2003 Jul; http://arxiv.org/abs/cs/0307038, arXiv: cs/0307038.

Dayan P. Improving generalization for temporal difference learning: The successor representation. Neural Computation. 1993; 5(4):613–624. http://www.mitpressjournals.org/doi/abs/10.1162/neco.1993.5.4.613, 00086.

Eichenbaum H, Cohen NJ. Can we reconcile the declarative memory and spatial navigation views on hippocampal function? Neuron. 2014 Aug; 83(4):764–770. http://dx.doi.org/10.1016/j.neuron.2014.07.032, doi: 10.1016/j.neuron.2014.07.032.

Gao P, Trautmann E, Yu BM, Santhanam G, Ryu S, Shenoy K, Ganguli S. A theory of multineuronal dimensionality, dynamics and measurement. bioRxiv. 2017 Nov; p. 214262. https://www.biorxiv.org/content/early/2017/11/05/214262, doi: 10.1101/214262.

Grassberger P, Procaccia I. Measuring the strangeness of strange attractors. Physica D: Nonlinear Phenomena. 1983 Oct; 9(1):189–208. http://www.sciencedirect.com/science/article/pii/0167278983902981, doi: 10.1016/0167-2789(83)90298-1.

Hastie T, Tibshirani R, Friedman J. Unsupervised learning. In: *The elements of statistical learning* Springer; 2009.p. 485–585.

Hinton GE, Salakhutdinov RR. Reducing the Dimensionality of Data with Neural Networks. Science. 2006; 313(5786):504–507. http://science.sciencemag.org/content/313/5786/504, doi: 10.1126/science.1127647.

LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015 May; 521(7553):436–444. doi: 10.1038/nature14539, wOS:000355286600030.

Levina E, Bickel PJ. Maximum Likelihood Estimation of Intrinsic Dimension. In: Saul LK, Weiss Y, Bottou L, editors. *Advances in Neural Information Processing Systems 17* MIT Press; 2005.p. 777–784. http://papers.nips.cc/paper/2577-maximum-likelihood-estimation-of-intrinsic-dimension.pdf.

Lipton ZC. A Critical Review of Recurrent Neural Networks for Sequence Learning. CoRR. 2015; abs/1506.00019. http://arxiv.org/abs/1506.00019.

Litwin-Kumar A, Harris KD, Axel R, Sompolinsky H, Abbott LF. Optimal Degrees of Synaptic Connectivity. Neuron. 2017 Mar; 93(5):1153–1164.e7. https://www.cell.com/neuron/abstract/S0896-6273(17)30054-5, doi: 10.1016/j.neuron.2017.01.030.

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, Berlin, Heidelberg: Springer-Verlag; 2011. p. 374–389. http://dl.acm.org/citation.cfm?id=2034117.2034142.

Low RJ, Lewallen S, Aronov D, Nevers R, Tank DW. Probing variability in a cognitive map using manifold inference from neural dynamics. bioRxiv. 2018; https://www.biorxiv.org/content/early/2018/09/16/418939, doi: 10.1101/418939.

Mazzucato L, Fontanini A, La Camera G. Stimuli Reduce the Dimensionality of Cortical Activity. Frontiers in Systems Neuroscience. 2016 Feb; 10. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4756130/, doi: 10.3389/fnsys.2016.00011.

Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. arXiv preprint arXiv:13013781. 2013; .

Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J. Distributed representations of words and phrases and their compositionality. In: *Advances in neural information processing systems*; 2013. p. 3111–3119.

Milivojevic B, Doeller CF. Mnemonic networks in the hippocampal formation: From spatial maps to temporal and conceptual codes. Journal of Experimental Psychology: General. 2013; 142(4):1231.

O'Keefe J, Dostrovsky J. The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat. Brain Res. 1971 Nov; 34(1):171–175.

Pascanu R, Mikolov T, Bengio Y. On the difficulty of training Recurrent Neural Networks. ArXiv e-prints. 2012 Nov; .

555 **Pehlevan C**, Sengupta AM, Chklovskii DB. Why do similarity matching objectives lead to Hebbian/anti-Hebbian
556 networks? Neural computation. 2018; 30(1):84–124.

557 **Rigotti M**, Barak O, Warden MR, Wang XJ, Daw ND, Miller EK, Fusi S. The importance of mixed selectivity in
558 complex cognitive tasks. Nature. 2013; 497(7451):585.

559 **Rigotti M**, Ben Dayan Rubin D, Morrison SE, Salzman CD, Fusi S. Attractor concretion as a mechanism for the
560 formation of context representations. Neuroimage. 2010 Sep; 52(3):833–847. http://dx.doi.org/10.1016/j.
561 neuroimage.2010.01.047, doi: 10.1016/j.neuroimage.2010.01.047.

562 **Rigotti M**, Ben Dayan Rubin D, Wang XJ, Fusi S. Internal representation of task rules by recurrent dynam-
563 ics: the importance of the diversity of neural responses. Frontiers in Computational Neuroscience. 2010;
564 4(24):29. http://frontiersin.org/Journal/Abstract.aspx?s=237&name=Computational_Neuroscience&ART_DOI=
565 10.3389/fncom.2010.00024, doi: 10.3389/fncom.2010.00024.

566 **Russek EM**, Momennejad I, Botvinick MM, Gershman SJ, Daw ND. Predictive representations can link model-
567 based reinforcement learning to model-free mechanisms. PLoS computational biology. 2017; 13(9):e1005768.

568 **Schiller D**, Eichenbaum H, Buffalo EA, Davachi L, Foster DJ, Leutgeb S, Ranganath C. Memory and Space:
569 Towards an Understanding of the Cognitive Map. J Neurosci. 2015 Oct; 35(41):13904–13911. http://dx.doi.org/
570 10.1523/JNEUROSCI.2618-15.2015, doi: 10.1523/JNEUROSCI.2618-15.2015.

571 **Sengupta A**, Tepper M, Pehlevan C, Genkin A, Chklovskii D. Manifold-tiling Localized Receptive Fields are Optimal
572 in Similarity-preserving Neural Networks. bioRxiv. 2018; https://www.biorxiv.org/content/early/2018/10/29/
573 338947, doi: 10.1101/338947.

574 **Solstad T**, Boccara CN, Kropff E, Moser MB, Moser EI. Representation of Geometric Borders in the Entorhinal
575 Cortex. Science. 2008 Dec; 322(5909):1865–1868. doi: 10.1126/science.1166466, wOS:000261799400061.

576 **Stachenfeld KL**, Botvinick M, Gershman SJ. Design Principles of the Hippocampal Cognitive Map. In:
577 Ghahramani Z, Welling M, Cortes C, Lawrence ND, Weinberger KQ, editors. *Advances in Neural Infor-*
578 *mation Processing Systems 27* Curran Associates, Inc.; 2014.p. 2528–2536. http://papers.nips.cc/paper/
579 5340-design-principles-of-the-hippocampal-cognitive-map.pdf.

580 **Stensola H**, Stensola T, Solstad T, Froland K, Moser MB, Moser EI. The entorhinal grid map is discretized. Nature.
581 2012 Dec; 492(7427):72–78. doi: 10.1038/nature11649, wOS:000311893400047.

582 **Sutskever I**, Vinyals O, Le QV. Sequence to sequence learning with neural networks. In: *Advances in neural*
583 *information processing systems*; 2014. p. 3104–3112.

584 **Tenenbaum JB**, De Silva V, Langford JC. A global geometric framework for nonlinear dimensionality reduction.
585 science. 2000; 290(5500):2319–2323.

586 **Turian J**, Ratinov L, Bengio Y. Word representations: a simple and general method for semi-supervised learning.
587 In: *Proceedings of the 48th annual meeting of the association for computational linguistics* Association for
588 Computational Linguistics; 2010. p. 384–394.

589 **Van Der Maaten L**, Postma E, Van den Herik J. Dimensionality reduction: a comparative. J Mach Learn Res.
590 2009; 10:66–71.

591 **Vincent P**, Larochelle H, Bengio Y, Manzagol PA. Extracting and Composing Robust Features with Denoising
592 Autoencoders. In: *Proceedings of the 25th International Conference on Machine Learning* ICML '08, New York, NY,
593 USA: ACM; 2008. p. 1096–1103. http://doi.acm.org/10.1145/1390156.1390294, doi: 10.1145/1390156.1390294.

594 **Wayne G**, Hung CC, Amos D, Mirza M, Ahuja A, Grabska-Barwinska A, Rae J, Mirowski P, Leibo JZ, Santoro A,
595 Gemici M, Reynolds M, Harley T, Abramson J, Mohamed S, Rezende D, Saxton D, Cain A, Hillier C, Silver D,
596 et al. Unsupervised Predictive Memory in a Goal-Directed Agent. arXiv:180310760 [cs, stat]. 2018 Mar;
597 http://arxiv.org/abs/1803.10760, arXiv: 1803.10760.

598 **Weinberger KQ**, Saul LK. Unsupervised learning of image manifolds by semidefinite programming. International
599 journal of computer vision. 2006; 70(1):77–90.

600 **Werbos PJ**. Backpropagation through time: what it does and how to do it. Proceedings of the IEEE. 1990;
601 78(10):1550–1560.

602 **Wills TJ**, Cacucci F, Burgess N, O'Keefe J. Development of the Hippocampal Cognitive Map in Preweanling Rats.
603 Science. 2010 Jun; 328(5985):1573–1576. doi: 10.1126/science.1188224, wOS:000278859200051.

Methods

Linear Dimensionality: Participation Ratio

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

$$PR = \frac{(\mathrm{Tr}\boldsymbol{C})^2}{\mathrm{Tr}(\boldsymbol{C}^2)} = \frac{(\sum_{i=1}^N \lambda_i)^2}{\sum_{i=1}^N \lambda_i^2} = \frac{1}{\sum_{i=1}^N \tilde{\lambda}_i^2} \tag{10}$$

where $\tilde{\lambda}_i = \lambda_i / \sum_{j=1}^N \lambda_j$. In the case of the example of Fig. 1, if we assume that all the locations of the latent space $\mathcal{X}, \mathcal{Y}$ 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 $\boldsymbol{\sigma} \equiv (\sigma_x, \sigma_y) = (\sigma, \sigma)$ is given by:

$$\begin{aligned}
\boldsymbol{C}_{ij} &= \frac{1}{T}\int_0^T dt\, (\mathcal{G}_\sigma(\boldsymbol{x}_i - \boldsymbol{x}_t) - \frac{1}{T}\int_0^T \mathcal{G}_\sigma(\boldsymbol{x}_i - \boldsymbol{x}_s)ds)(\mathcal{G}_\sigma(\boldsymbol{x}_j - \boldsymbol{x}_t) - \frac{1}{T}\int_0^T \mathcal{G}_\sigma(\boldsymbol{x}_j - \boldsymbol{x}_s)ds) = \\
&= \frac{1}{T}\int_0^T dt(\mathcal{G}_\sigma(\boldsymbol{x}_i - \boldsymbol{x}_t) - 1)(\mathcal{G}_\sigma(\boldsymbol{x}_j - \boldsymbol{x}_t) - 1) = \frac{1}{T}\int_0^T dt\, \mathcal{G}_\sigma(\boldsymbol{x}_i - \boldsymbol{x}_t)\mathcal{G}_\sigma(\boldsymbol{x}_j - \boldsymbol{x}_t) - 1 = \\
&= \frac{1}{\sqrt{2\pi}\sigma}\frac{1}{T}e^{-\frac{\Delta^2}{2\sigma^2}}\int_0^T dt\, \mathcal{G}_{\sigma/\sqrt{2}}((\boldsymbol{x}_i + \boldsymbol{x}_j)/2 - \boldsymbol{x}_t) - 1 = \\
&= \frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{\Delta^2}{2\sigma^2}} - 1\,.
\end{aligned} \tag{11}$$

where $\mathcal{G}_\sigma$ is a Gaussian with variance σ normalized to 1 as described in the main text. Eq. 11 shows that C_{ij} has a band structure; in particular it is in Toeplitz form, with entries that decay with the distance between neurons in latent space (***Gao et al., 2017***). We can now compute the terms in Eq. 10 that determine the PR. Specifically we obtain:

$$\begin{aligned}
(\boldsymbol{C}^2)_{ij} &= \sum_{k=1}^N C_{ik}C_{jk} \approx \int_{-\infty}^{\infty} \mathcal{G}_\sigma(i-k)\mathcal{G}_\sigma(k-j)dk = \\
&= \frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(i-j)^2}{2\sigma^2}}\,.
\end{aligned} \tag{12}$$

Thus the PR in the limit of large N is:

$$PR = \frac{(Tr\boldsymbol{C})^2}{Tr(\boldsymbol{C}^2)} = \frac{1}{\sqrt{2\pi}\sigma}\,. \tag{13}$$

This shows that the PR dimensionality grows with the inverse of the width of the Gaussian kernel.

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 (***Camastra and Staiano, 2016***; ***Campadelli et al., 2015***). The set of techniques we use includes: $\mathrm{MiND}_{\mathrm{ML}}$ (***Lombardi et al., 2011***), MLE (***Levina and Bickel, 2005***), DancoFit (***Ceruti et al., 2012***), CorrDim (***Grassberger and Procaccia, 1983***) and GMST (***Tenenbaum et al., 2000***; ***Costa and Hero, 2003***). These techniques follow the selection criteria illustrated in ***Camastra and Staiano*** (***2016***), emphasizing ability to handle high-dimensional data (in our case hundreds of dimensions) and being robust, efficient and reliable; we refer the reader to ***Van Der Maaten et al.*** (***2009***) as a useful comparison. We implement these techniques using the code from the the authors available online ***Levina and Bickel*** (***2005***); ***Ceruti et al.*** (***2012***); ***Camastra and Staiano*** (***2016***), "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 (***Sutskever et al., 2014***), 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 (***Rigotti et al., 2010b***,a; ***Lipton, 2015***).

Figure 3c 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 $\vec{r}$ 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 tiles). 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, Fig.3b 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. 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.3D. Such a vector, together with the action encoded through a $1-8$ 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 (number of tiles in the environment) is then used, for each color channel, to make the color representations smooth. Fig. 3b 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 (***Werbos, 1990***). While RNNs are known to be difficult to train in many cases (***Pascanu et al., 2012***), a simple vanilla RNN model with hyperbolic tangent activation function is able to learn our benchmark task.

The connectivity matrix of the recurrent network is initialized to the identity (***LeCun et al., 2015***; ***Collins et al., 2016***), 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.

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. 7 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 $1 \cdot 10^{-7}$.