Decoding neural responses with minimal information loss

John A. Berkowitz and Tatyana O. Sharpee

Computational Neurobiology Laboratory, Salk Institute for Biological Studies, La Jolla, CA 92037

Department of Physics, University of California, San Diego, La Jolla, CA 92093

Abstract:

Cortical tissue has a circuit motif termed the cortical column, which is thought to represent its basic computational unit but whose function remains unclear. Here we propose, and show quantitative evidence, that the cortical column performs computations necessary to decode incoming neural activity with minimal information loss. The cortical decoder achieves higher accuracy compared to simpler decoders found in invertebrate and subcortical circuits by incorporating specific recurrent network dynamics. This recurrent dynamics also makes it possible to choose between alternative stimulus categories. The structure of cortical decoder predicts quadratic dependence of cortex size relative to subcortical parts of the brain. We quantitatively verify this relationship using anatomical data across mammalian species. The results offer a new perspective on the evolution and computational function of cortical columns.

Introduction

The mammalian cerebral cortex of mammals is a thin-layered tissue that appears to be assembled from a circuit motif termed the minicolumn, or column for short (Buxhoeveden, 2012). Each column spans the cortical layers and has stereotypic connections between cell types within and across layers. Columns can form 'macro-columns', which are groups of ~100 minicolumns that are bound together by short range connections between columns. Even within macrocolumns, however, one can still discern the vertical structure corresponding to individual columns (Buxhoeveden, 2012). Although there are quantitative variations in column parameters across species, across brain regions, and even within a given macro-column, the main features of this circuit motif are quite universal. Columns are found in both sensory and motor areas of the brain (Harris and Shepherd, 2015), and analogous circuit motifs have also been found in non-mammals, such as birds (Wang et al., 2010). These facts strongly suggest that this circuit motif performs fundamental element(s) of a computation that is needed independent of the stimulus modality. However, the complexity of connections, and the large number of cell types within the column (many of which remain to be strictly defined (Harris and

Shepherd, 2015; Jiang et al., 2015; Luo et al., 2017)), have made it difficult to determine the algorithm implemented by cortical columns.

To gain insights into the computations performed by cortical columns, one can begin by analyzing, from first-principles, possible strategies for representing stimuli with neural responses in ways that allow for their accurate decoding, ideally with minimal loss of information. The colloquial term “information” used here can be quantitatively defined using tools from information theory (Cover and Thomas, 1991). In the context of this work, by information we mean the mutual Shannon information (Cover and Thomas, 1991) between stimuli and neural responses. When considered in small time intervals, neurons respond to stimuli by producing all-or-nothing events in the voltage traces across their membranes termed “spikes”. Unfortunately, unlike the genetic code where it is known how to parse DNA sequence to determine amino acid sequence, it remains unclear how to parse sequences of spikes over time to determine which stimuli they represent (Srivastava et al., 2017; Theunissen and Miller, 1995). Similarly, it is a matter of debate for how to combine spikes from neurons tuned to different stimulus features. A simple weighted average of responses across neurons has been shown to discard substantial amounts of information (Osborne et al., 2008; Reich et al., 2001), indicating that more complex codes are used in the brain. We will discuss how insights into these problems can be gained by searching for the code that allows for decoding with minimal loss of the information contained in neural responses. Further, we present evidence one version of this code is implemented by cortical columns.

Results

A code with information-preserving statistic

We begin by describing how a stimulus can be represented by neural responses in a way that these neural responses can be decoded without loss of information. For ease of exposition, we will first consider the case where, once the stimulus is specified, noise in neural responses is independent across neurons and across different time bins (we will show that the main result holds even when these constraints are removed). Further, we will initially analyze stimulus representations where the responses of any single neuron depend only on one stimulus dimension (this dimension corresponds to neuron’s receptive field (RF), as we mathematically

describe below). Later we will see that expanding representations to allow dependence of neural responses on multiple stimulus dimensions is one of the key distinguishing features of the cortical column decoder, compared to a simpler decoder.

With these assumptions, the neural responses can be described using the so-called linear-nonlinear (LN) model (Schwartz et al., 2006). In this model, the probability of observing a spike depends on the strength of the relevant stimulus component (Figure 1). The nonlinear dependence of the neuronal spike rate on the primary stimulus component can often be well approximated by a saturating (logistic) function that has two parameters: a threshold α and a steepness β. We note that neural responses can often be also equivalently described using tuning curves (Abbott and Dayan, 1999; Georgopoulos et al., 1986; Hohl et al., 2013; Osborne et al., 2008; Shamir, 2014). Tuning curves specify how neural response rates change when stimuli deviate from their optimal settings. There is a one-to-one relationship between the parameters of the tuning curves and those of the saturating nonlinearity in the LN model (Supplementary Text 1). However, the LN formulation makes it possible to mathematically derive the vector quantity $\vec{M}$ that is guaranteed to capture all the information provided by the responses of the neural population (cf. Supplementary Text 2). This quantity is constructed as:

$$\vec{M}\{r_1, \ldots r_N\} = \sum_i \beta_i r_i \vec{w}^{(i)} \quad . \qquad (1)$$

In this expression, r_i denotes the number of spikes produced by the *i*th neuron during the time interval of interest, β_i is the steepness value of the *i*th neuron nonlinearity, and $\vec{w}^{(i)}$ is the preferred stimulus for this neuron, also known as neuron's RF, normalized to have unit contrast (mathematically, $\left|\vec{w}^{(i)}\right| = 1$). We will refer to $\vec{M}$ as the information-preserving population vector. This vector generalizes the standard population vector (Georgopoulos et al., 1986; Hohl et al., 2013; Salinas and Abbott, 1994; Shamir, 2014) by taking into account steepness parameters β_i. In the context of the LN model, steeper nonlinearities indicate more reliable neural responses. Therefore, it is perhaps intuitive that the responses of more reliable neurons should be weighted more strongly within the population average.

Taking into account steepness parameters fully addresses previous concerns regarding the standard population vector, namely that averaging responses of similarly tuned neurons can lead to substantial

information loss (Osborne et al., 2008; Reich et al., 2001). In Figure 2, we show that our information-preserving version of the population vector captures all the information available in neural responses, regardless of whether the neurons in the population are tuned to the same (Fig. 2A) or different (Fig. 2B) features of the stimulus. In contrast, the standard population vector does not capture all of the information when the population contains neurons tuned to the same features of the stimulus (or features with opposite polarity) using nonlinearities with different steepness values.

It is worth noting that some differences in neural tuning curves are irrelevant for capturing all the information contained in neural responses. For example, although differences in thresholds lead to differences in tuning curves, according to the information-preserving expression (1), responses of neurons with different thresholds can still be averaged without losing information. Figure 2C shows that both the standard and information-preserving population vectors capture all the information in model neural population without taking threshold differences into account. These analyses demonstrate that analyses in terms of saturating nonlinearities are more revealing than those based on the tuning curves.

The information-preserving population vector also works in the presence of correlated variability across neurons (the so-called noise correlation reviewed in (Averbeck et al., 2006; Shamir, 2014)). Although noise correlations may affect the overall information provided by the neural responses (Abbott and Dayan, 1999; Ecker et al., 2011; Moreno-Bote et al., 2014; Shamir, 2014; Shamir and Sompolinsky, 2004, 2006; Zohary et al., 1994), the information-preserving population vector continues to capture all of the information that is available in the neural responses (Fig. 2D). This result holds true for noise correlations that differ across pairs of neurons (e.g. according to differences in RFs) as long as noise correlations do not change with the stimulus, as is often observed experimentally(Huang and Lisberger, 2009). In Figure 2 we show the results of model simulations with noise correlations and provide a detailed derivation in the Supplementary Text S3. We also note that in the presence of noise correlations, the nonlinearities of individual neurons may deviate from the logistic function but the information-preserving property still holds as long as the population response can be written in the exponential form (see Eq. S24 in the supplement).

Capturing information in cortical responses

Analyses of the model neural populations show that the information contained in the responses of model neurons conforming to the LN model can be fully captured by a version of the population vector that is modified in a specific way. Although the LN model has been successfully used to describe neural responses in a number of brain circuits (Schwartz et al., 2006), to the extent to which real neural responses deviate from the model assumptions, some information loss will occur. To determine the magnitude of these effects, we tested how the information-preserving population vector performs on the responses of neurons in the primary visual cortex (V1) that were elicited by natural stimuli (Sharpee et al., 2006). For each neuron, we estimated its preferred orientation and nonlinearity. Nonlinearities were fit to the neural responses using logistic regression to find the steepness parameters β. Based on the estimates of β values, we compared the full amount of information provided by the responses of these neurons with the information provided by the standard and information-preserving population vectors. To account for experimental uncertainties in the orientation value, we used a coarse-grained set of orientations that took into account error-bars (see Materials and Methods). We find that just like in the model neural populations, the information-preserving population vector (but not the standard population vector) captured all the information (Figure 3) provided by the responses of simultaneously recorded neurons.

Decoding algorithm

We now show how signals can be decoded from the information-preserving population vector. The value of this vector varies continuously with the stimulus, because different stimuli evoke different responses r_i. The expected value of the information-preserving vector depends on the stimulus $\vec{S}$ as:

$$\vec{M}(\vec{S}) = \sum_i \beta_i r_i \vec{w}^{(i)} = \sum_i \beta_i \vec{w}^{(i)} r\left(\beta_i \vec{w}^{(i)} \cdot \vec{S}\right), \qquad (2)$$

where $r(x)$ is nonlinear response function and parameters $\vec{w}^{(i)}$ and β_i are same as before in Eq. (1) (see Text S3 for derivation). To understand the properties of this mapping, we can build on research in the area of RF estimation (Schwartz et al., 2006), where this mapping uses responses of a single neuron to many different stimuli to estimate that neuron's RF. Here we use RFs of many neurons and their responses to a single stimulus to estimate that stimulus. Based on studies of RF estimation, we can state that the information-

preserving population vector will be aligned with the stimulus multiplied by the covariance matrix *C* of RF components across the neural population under certain statistical conditions (Supplementary Text S4 for a derivation and discussion of deviations when these conditions are not met).

Crucially, using a standard population vector expression $\sum_i \vec{w}^{(i)} r\left(\beta_i \vec{w}^{(i)} \cdot \vec{S}\right)$ (or any expression where RFs are not scaled by the same factor β_i from the spike rate nonlinearity) introduces estimation biases. This is shown in simulations of Figure 4A and mathematically in Text S4. These biases arise as soon as RF components have non-equal variance in different stimulus directions. Such is the case, for example, in V1 where more RFs align with horizontal and vertical orientation than with oblique angles (Dragoi et al., 2001). We have also verified these results using neural data by reconstructing segments of natural movies using either the information-preserving or standard population vector, as well as a population vector with random β_i factors. The information-preserving vector produced significantly more accurate reconstructions than either of the two alternatives ($p< 10^{-39}$ t-test, Fig. 4B). Further, the decoder maintains much of its accuracy even if the estimates of β factors are imprecise. For example, we used a decoder where the β factors were multiplied by a random value between 0 and 1; such a decoder yielded correlation coefficients that, on average, were more than half of the correlation values provided by the true information-preserving population vector. Thus, even partial knowledge of β factors can result in substantial improvements in decoding accuracy.

Feedforward neural network decoder

In terms of biological implementation, stimulus decoding based on the information-preserving vector can be implemented using a three-layer feedforward neural network, as illustrated in Figure 5A. Units within the first layer encode stimulus components; the second layer provides a representation according to Eq. (1); units within the third layer units represent reconstructed stimulus values according to Eq. (2). The key aspect of this decoding scheme is the incoming connections to each second layer unit should be proportional to outgoing connections. A version of this decoder was shown to accurately reflect the synaptic and network mechanisms of the leech nervous system (Lewis and Kristan, 1998). Similar networks operate in subcortical areas in mammals (Joshua and Lisberger, **2015**), and can account for initial stages of olfactory processing (Zhang and

Sharpee, 2016) for odor mixtures. We note that this decoder has compressive nonlinearities (illustrated in Figure S1), making it different from the optimal linear decoder proposed previously (Salinas and Abbott, 1994).

Decoding of ambiguous stimuli

We now show how the decoding scheme can be generalized to increase its accuracy and to allow neural circuits to deliberate between alternative, mutually exclusive interpretations of ambiguous stimuli. This is an important problem because sensory perception is in general ill-defined, meaning that different stimuli can give rise to similar patterns of neural responses. The decoder discussed so far (e.g., Figure 4 and 5A) does not solve this problem because for each stimulus it produces a single interpretation. To allow for multiple interpretations, one can expand the stimulus representations quadratically by including all pairwise products $s_i s_j$ between original s_i components. That is, if the original stimulus has *D* components, after expansion, the stimulus will be represented by the original *D* components plus an additional $(D^2+D)/2$ pairwise components. Working in this expanded space, one can construct the information-preserving population vector according to Eq. (1) as before. The information-preserving vector now has a linear part M_i and a quadratic part M_{ij} (see Supplemental Text S5 for details). To decode a single stimulus pattern from these two parts, we need to find a pattern s_i that approximates the matrix M_{ij} as best as possible in the form of $s_i s_j$. Mathematically, this operation corresponds to finding the leading mode of matrix M_{ij}. A key property of this transformation is that multiple modes can potentially provide similar contributions to the matrix. This situation corresponds to ambiguous stimuli, with each mode describing alternative representations. The conflict between modes can be resolved by waiting for additional evidence to favor a specific representation and/or by incorporating evidence from larger scales. Thus, decoding with quadratic stimulus expansion represents a conceptual advance compared to the decoding in the original input space. Of course, to allow for the possibility of multiple modes, the original stimulus should be multidimensional with *D>1.* It also should be noted that the purely quadratic decoder based on M_{ij} does not determine stimulus polarity. The stimulus polarity is determined by comparing the sign of the estimated stimulus with the linear part M_i. Thus, both parts of the information-preserving population vector are needed to ensure complete stimulus reconstruction.

We tested the accuracy of this quadratic decoding algorithm on recoded V1 responses(Sharpee et al., 2006). We found that it produces improved reconstructions compared to those made without quadratic stimulus expansion ($p<10^{-28}$, Fig. 4D). Furthermore, just like in the original stimulus space, a reconstruction based on the information-preserving population vector is significantly better than those based on the standard population vector or population vector computed with randomly selected β_i values ($p<10^{-52}$). Finally, similar to decoding without quadratic stimulus expansion, the quadratic decoder is robust to noise in the estimation of β_i factors. In Fig. 4b we show that decoding using β_i factors that have been multiplied by a random number from 0 to 1 produces accurate stimulus reconstructions. We observe that quadratic stimulus decoding is even more robust to this perturbation than decoding performed in the original stimulus space (Fig. 4B).

Recurrent neural network decoder

Given the computational benefits of decoding with quadratic stimulus expansion, how can it be implemented in neural circuits? We now discuss how each of the steps of this algorithm maps onto computations performed by the cortical column. First, one needs neurons with RFs in the quadratically expanded input space (Fitzgerald et al., 2011). When analyzed in the original input space, these neurons would have quadratic nonlinearities (possibly also with a non-zero linear component). In V1, such neurons are known as complex cells (Movshon et al., 1978). However, they are also found in upper layers of the primary auditory cortex (Atencio et al., 2009). The responses of these neurons can be obtained by adding the responses of simple cells from layer 4 that are selective for stimuli of opposite polarity. This corresponds to the classic model of complex cells responses (Movshon et al., 1978), and which is also consistent with strong projections from layer 4 simple cells to layer2 complex cells (Harris and Mrsic-Flogel, 2013; Harris and Shepherd, 2015).

Next, the information-preserving population vector must be computed both in the original space and in the quadratically expanded space. In the original input space this computation can be done using the same feedforward procedure as before, in this case by pooling the responses of simple cells according to Eq. (2) $M_i = \sum_{i=1}^{N} \beta_i r_i w_k^{(i)}$, where $w_k^{(i)}$ stands for the *k*th component of *i*th neuron RF, and *N* is the number of simple cells. The quadratic part of the information-preserving population vector M_{kn} can be estimated by pooling the

responses of complex cells c_i to form $M_{kn} = \sum_{i=1}^{N} \beta_i^2 c_i w_k^{(i)} w_n^{(i)}$. The weights $w_n^{(i)}$ can be provided by copies of connections from the corresponding simple cells.

The final step is to find the dominant mode of the matrix represented by quantities M_{kn}, choosing its polarity based on M_i values. These two operations can be simultaneously computed by a recurrent network that receives quantities M_i as inputs to its *i*th neuron and connections between neurons *k* and *n* in the network are set to M_{kn}. In the presence of gain normalization (Carandini and Heeger, 2011), the activity of this network will converge to the dominant mode of matrix, implementing the so-called power method.

Based on the circuitry of cortical columns (Harris and Mrsic-Flogel, 2013), one can identify the output recurrent network with that of layer 5 subcerebral projection neurons (SPN) as well as layer 6 cortical thalamic neurons. These neurons receive connections from layer 4 that necessary to compute input values M_i. A key aspect of this recurrent optimization is that connection strengths M_{kn} must vary with the stimulus. The layer 5 contains a population of cells termed intratelecephalic neurons (ITNs) that project to layer 5 SPNs and can modulate connections between SPNs in a stimulus-dependent manner. The ITNs receive signals from layer 2 necessary to compute M_{kn} values. Thus, the cortical column has all of the required components to implement quadratic decoding.

Predicted scaling relationships

If quadratic decoding is indeed the algorithm that is being implemented by cortical columns, then this yields a number of quantitative predictions concerning for the distribution of cell types across layers, and how the size of the cortex should scale with the number of subcortical inputs. Here we review these predictions in turn.

The first prediction is that the number of ITNs in layer 5 should equal to the square of the number of output SPNs in layer 5. There is evidence in the mouse motor cortex that this is the case. Anatomical images from indicate that there are 8-9 corticospinal neurons and 60-80 ITNs in each minicolumn (Oswald et al., 2013) The second prediction describes how the total number of neurons in a column should scale with the dimensionality of the signal. Specifically, a minicolumn that processes *D*-dimensional signals should have $\sim D^2 + \chi D$ neurons. The quadratic term arises from the need to implement stimulus-dependent recurrent

weights between the output neurons. The linear term includes the output neurons as well as neurons from the intermediate representations, such as simple and complex cells, whose number should be proportional to *D*. For a piece of cortical tissue with N_{columns} , the number of neurons will be $N_{\text{cortex}} = N_{\text{columns}}\left(D^2 + \chi D\right)$. The corresponding number of subcortical input neurons $N_{\text{t}} = N_{\text{columns}} D$. Combining these two relations yields the following prediction

$$N_{\text{cortex}} = \frac{1}{N_{\text{columns}}} N_t^{\,2} + \chi N_t \qquad (3)$$

In Figure 6, we show that the predicted quadratic function accounts well for the differences in the number of cortical and subcortical neurons (from the diencephalon, brainstem, and basal ganglia) across 18 mammalian species (Herculano-Houzel, 2009). We fit data across primate species separately because they are known to have scaling exponents that are different from other mammals in a number of ways (Herculano-Houzel, 2009). For rodents/insectivores, the fit yields: $N_{\text{columns}} = \left(2.7 \pm 0.9\right)10^8$, $\chi = 1.75 \pm 0.16$; for primates the fit yields: $N_{\text{columns}} = \left(1.6 \pm 0.4\right)10^8$, $\chi = 11.7 \pm 1.5$. Although the parameter values N_{columns} are obtained merely by fitting the data, without any constraints, they quantitatively match the current estimate of 2×10^8 for the number of minicolumns in the brain (Sporns et al.). The same estimate can also be obtained by dividing the total number of cortical neurons (Herculano-Houzel, 2009) by the estimated number of neurons per minicolumn (Buxhoeveden, 2012). Further, one can the use these parameters to derive estimates for the microscopic parameters of individual columns. Combining the values for the parameter χ with the estimated number of ~100 neurons per minicolumn (Buxhoeveden, 2012) yields estimates for the number of output neurons *D* of ~9 for rodents and ~6 for primates, both of which agree with experimental values for these species (Oswald et al., 2013; Peters and Sethares, 1996). Thus, the scaling rules that follow from the structure of the quadratic decoder are supported by a diverse sets of quantitative predictions across nine orders of magnitude in neuron numbers (from 10 to 10^{10}).

Discussion

In this work we started from first principles of information theory to find a set-up where stimuli can be encoded into neural responses in such a way as to enable the decoding of these responses with minimal loss of information. For this set-up, there a simple vector quantity that captures all of the information contained in neural responses. Based on this quantity, one can build two kinds of decoding algorithms. The first algorithm uses primarily feedforward operations of the kind found in invertebrate (Lewis and Kristan, 1998) and mammalian subcortical circuits (Joshua and Lisberger, 2015). Of course, subcortical circuits compute more sophisticated computations than just stimulus reconstruction. We analyzed stimulus reconstruction just as an example of a computation that neural circuits may perform.

A second more sophisticated way to perform decoding is to quadratically expand the stimulus space and then to use recurrent optimization to invert the transformation. We argue that the second decoding algorithm, which we tern quadratic decoding, is what cortical columns compute to deliver increased accuracy, as well as the ability to disambiguate between alternative stimulus interpretations. Quadratic decoding brings together many disparate properties of cortical processing as parts of a single computation. For example, the decoder requires synaptic weights to be zero on average for some neural populations (see Supplementary Text S4). This corresponds to the so-called balanced regime where inhibition and excitation on average cancel each other (van Vreeswijk and Sompolinsky, 1996). Furthermore, because recurrent networks can become unstable (Sompolinsky et al., 1988), a specific gain control is needed to reduce effective recurrent connections within layer 5 when necessary. The observed inhibitory gain control (Sompolinsky et al., 1988) that layer 5 exerts through layer 6 back to layers 4 and 2 can fulfill this role (Olsen et al., 2012). Some cortical areas, such as the olfactory system as well as the hippocampus are missing layers 2 and 4. These areas, therefore, likely lack this form of gain control, which would explain why they often serve as seizure origination points.

The structure of the quadratic decoder reveals new constraints on mammalian brain evolution. Two separate factors drive brain size expansion within and across mammalian species orders. The first factor controls brain expansion within orders, e.g. within primates. This factor represents the average dimensionality of inputs processed by each column. Among primates, humans have the largest value. The second factor χ

controls brain expansion across orders, such as between rodents and primates. It represents the weighted number of neuronal types per input dimension, weighted by the number of neurons in each type. Because of this weighting and because excitatory neurons comprise a majority of neurons (Jiang et al., 2015), the factor χ mainly reflects diversity of excitatory neuronal types. Taking this into account, the derived value for rodents is consistent with current estimates for the number of cortical excitatory cell types (Jiang et al., 2015; Markram et al., 2015). Further, a large increase in this factor from rodents to primates is consistent with observations that excitatory neuronal types are less conserved between rodents and primates than inhibitory types. A larger number of cell types encoding signals along each dimension increases the accuracy with which each signal can be encoded, as has been demonstrated in the retina (Kastner et al., 2015). At the same time, primate minicolumns process signals of smaller dimensionality than rodent minicolumns. This allows for finer sampling and reduces the number of competing modes for each quadratic decoder. These analyses highlight different axes that evolution can manipulate to acha accurate decoding of complex stimuli.

Author Contribution: JB derived information-preserving property, TS derived decoding algorithms, their implementation in neural circuits, the mapping onto cortical circuitry and the scaling relationships. Both authors analyzed the data and wrote the paper.

Acknowledgments: We thank Vicki Lundblad for comments on the manuscript. This research was supported by the Rose Hill Foundation, the National Science Foundation (NSF) award numbers IIS-1254123, IIS-1724421, and IOS-1556388.

Figures and Figures Legends

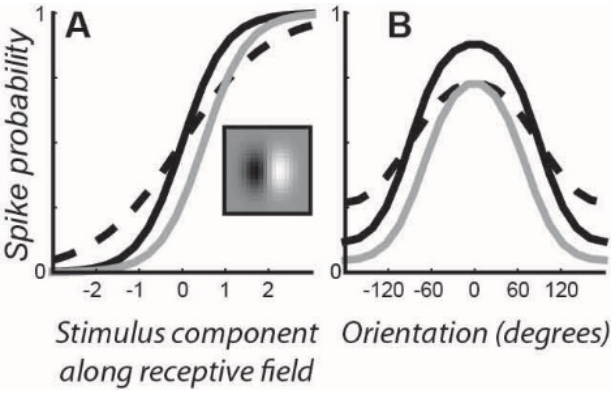

Fig. 1. The relationship between receptive field (RF) and tuning curve descriptions of the neural response. (**A**) Three representative model nonlinearities that describe neural response as a saturating function of stimulus component along RF. Black and dashed lines have the same midpoints α but different steepness values β. Black and gray lines have same β values but different α values. Inset shows an example orientation selective RF. (**B**) Corresponding tuning curves from (**A**) but as a stimulus function of angle.

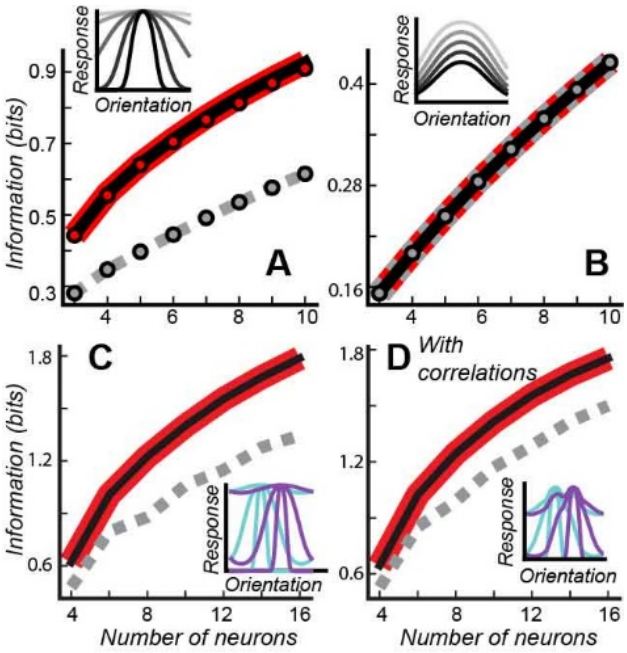

Fig 2. The information-preserving vector captures all information from diverse neural populations and with correlated variability across neurons. Neural populations tuned to the same (**A**, **B**) or different (**C**, **D**) preferred stimuli. In (**A**) differences in neural tuning curves are due to differences in steepness values, whereas in (**B**) they are due to differences in thresholds. Lines formed by dots show the information values obtained by binning response variables. Dotted lines overlap with solid curves. (**D**) same as (**C**) but with noise correlations. Insets show example population tuning curves for $n = 6$. In all panels, we compare information transmitted by a population response (black line) with information transmitted by the information-preserving population vector (red) and the standard population vector (dashed gray).

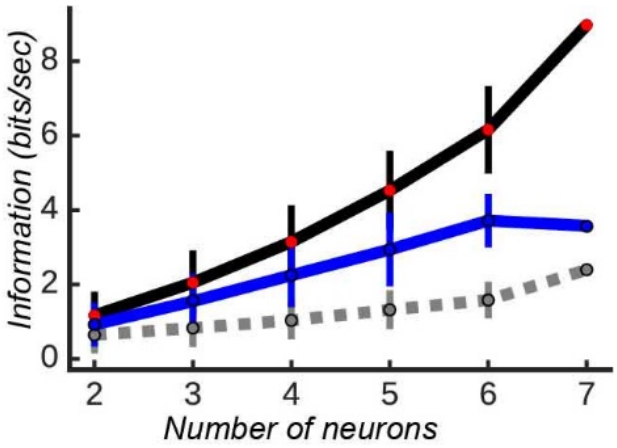

Fig. 3 The information-preserving population vector captures all the information provided by responses of simultaneously recorded V1 neurons. Curves for the information provided by neural responses (black line) and that captured by the information-preserving population vector (red circles) overlap. The standard population vector (blue) and population count (grey) provide smaller amounts of information. Error bars are standard deviations.

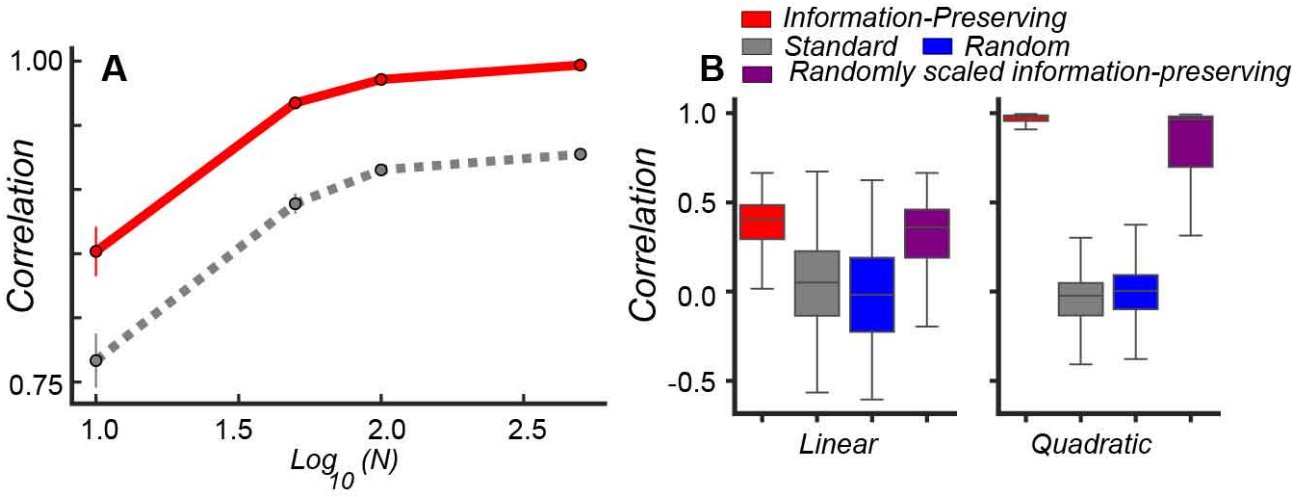

Fig. 4 (A) Average correlation between stimuli and their reconstructions based on either the information-preserving (red) or standard population vectors (grey). The information-preserving decoder converges to an unbiased estimate of stimulus (red line), but not the standard population vector (dashed grey). Error bars are standard error of the mean across realizations of population RFs. (**B**) Test of decoders based on V1 data (86 neurons) from (Sharpee et al., 2006). The reconstructions do not require that all of the neurons are recorded simultaneously. Therefore, we could use an expanded set of 86 V1 neurons for which the responses to the same set of natural stimuli were available. The box and whisker plots show median and interquartile range (IQR); whiskers are 1.5 of IQR. Correlation values here reflect measurements for individual stimuli, not averaged across the set of stimuli.

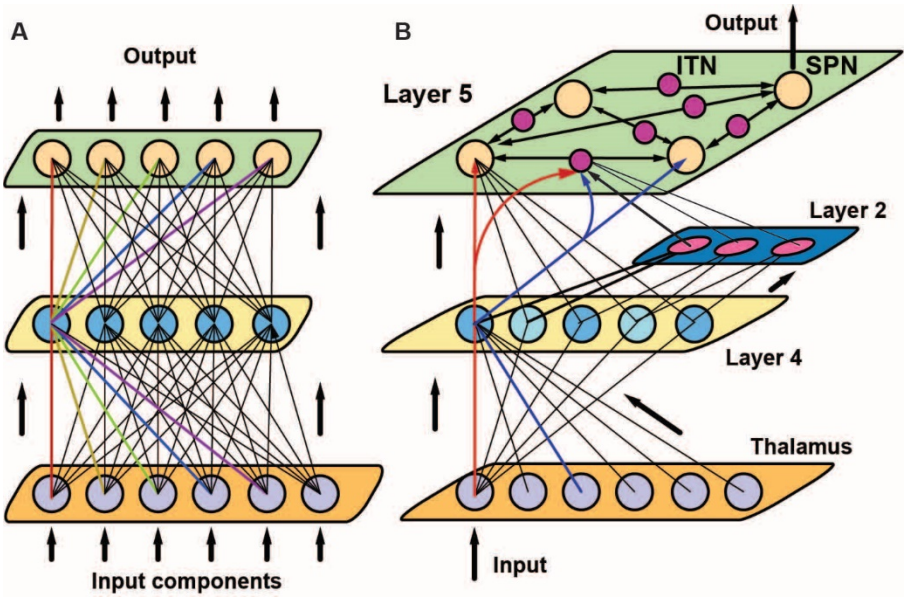

Fig. 5. Schematic of a feedforward (A) and quadratic (B) decoder. (**A**) A feedforward network can implement the decoder in the original input space. Color marks connections of the same strength. (**B**) Quadratic decoding adds recurrent computation in layer 5. Input from layer 2 complex cells to layer 5 intratelecephalic neurons (ITNs) sets the recurrent weights between the output layer 5 subcerebral projection neurons (SPNs). For clarity, only a subset of connections is shown.

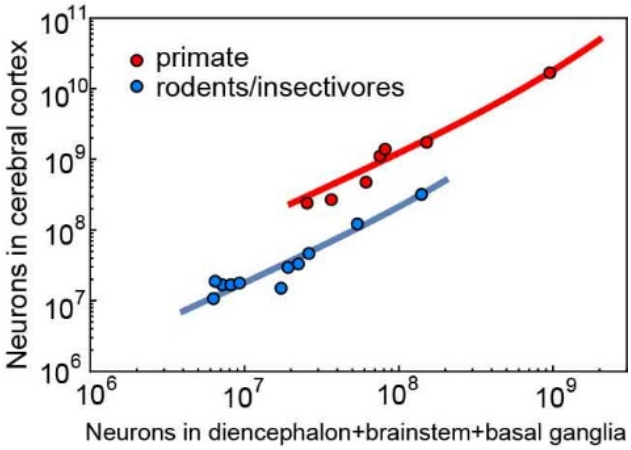

Fig. 6. Quadratic decoding explains quadratic scaling of cortex size across species. The number of cortical neurons as a function of to the total number of neurons in brain stem, basal ganglia and the diencephalon, which includes the thalamus. Data from (Herculano-Houzel, 2009). Solid lines represent fits using Eq. (3).

Materials and Methods

Estimating Mutual Information from V1 Recordings

We represent the responses of a set of N simultaneously recorded V1 cells to a stimulus binned into T segments and repeated K times as a tensor D of shape (T, K, N). T is typically 330, corresponding to time bins of 30 milliseconds. D_{tij} represents the number of times neuron j fired in response to stimulus t on repeat i, and can be any nonnegative integer.

Converting Data to Binary Words

Let ν_{max} be the largest value in D; the maximum number of spikes over all neurons, stimuli segments, and repeats. We form a binary tensor $\tilde{D}$ of shape ($\nu_{max} \times T$,K,N) by resampling each slice $D_{t..}$ into a sub tensor $\tilde{D}^{(t)}$ of shape (ν_{max},K, N) according to the following algorithm:

1. For a given value of i and j we let $n = D_{tij}$. We sample without replacement a set $L = \{\tau_l\}_{l=1}^{n}$ of n indices from the integers $\{1, ..., \nu_{max}\}$
2. We set $\tilde{D}^{(t)}_{\tau ij}$ to 1 if $\tau \in L$, and to 0 if not.
3. Steps 1 and 2 are repeated for all i and j.

After all the time slices of D are resampled, the set $\{\tilde{D}^{(t)}\}$ of binary tensors are concatenated to form a binary data tensor $\tilde{D}$ of shape ($\nu_{max} \cdot T$,K, N). Each row of $\tilde{D}_{ti.}$ corresponds to a sample of $\{r_j\}$. We note that the samples described by $\tilde{D}$ correspond to time bins of length $30/\nu_{max}$ milliseconds.

In Figure 3 we include only sets of neurons recorded simultaneously, and all subsets. Thus, a set of 4 simultaneously recorded neurons yields one set of size 4, 4 sets of size 3, and 6 sets of size 2.

Estimating parameters of the linear-nonlinear models

To compute population vectors and the information captured by them we need estimates of the linear-nonlinear (LN) model parameters $\vec{w}_k$ and β_k for every neuron. In order to estimate β_k, we fit the response rate of the kth neuron evoked by stimulus $\vec{s}$ (averaged across the repeated presentation of this stimulus) using a logistic function:

$$r_k(\vec{s}) = \frac{R_{\max}}{1 + e^{-2(\beta_k(x-\alpha_k))}}, \quad x = \vec{v}_k \cdot \vec{s}. \tag{1}$$

In this expression $\vec{v}_k$ is estimated as the maximally informative dimension for the neuron [Sharpee et al., 2006]. The parameters $R_{\max}$, α_k, and β_k are fit by minimizing the mean square error between $r_k(\vec{s})$ and experimentally measured firing rates.

To account for non-monotonic response functions, we also fit neural response functions using logistic function applied to a quadratic function of the stimulus. Specifically:

$$r(\vec{s}) = \frac{R_{\max}}{1 + e^{-2(\beta_1 x_1 + \beta_2 x_2 + \beta_{11} x_1^2 + 2\beta_{12} x_1 x_2 + \beta_{22} x_2^2 - \alpha)}}, \quad x_1 = \vec{s} \cdot \vec{v}^{(1)}, \quad x_2 = \vec{s} \cdot \vec{v}^{(2)}, \tag{2}$$

where $\vec{v}^{(1)}$ and $\vec{v}^{(2)}$ are two RF components estimated as the first and second maximally informative dimensions for the neuron [Sharpee et al., 2006]. These fits were used with quadratic decoding described in Text S5.

Information captured by population vectors

For the analysis in Figure 3 we used $\vec{w}_k = (\cos(\varphi_k), \sin(\varphi_k))^T$, where φ_k are preferred orientation values estimated from the MID vector $\vec{v}_k$ for each neuron computed in [Sharpee et al., 2006, Sharpee et al., 2008], along with their standard deviations $\Delta\varphi_k$. Additionally, in this figure we plot the information computed under a coarse grained realization of orientation values φ_k to take into account experimental errorbars $\Delta\varphi_k$ accociated with them. The coarse graining is based on the following measure of distinguishability between orientation values for neurons i and j:

$$d_{ij} = \frac{\angle(\varphi_i, \varphi_j)}{\frac{1}{2}(\Delta\varphi_i + \Delta\varphi_j)} \tag{3}$$

using the following procedure:

1. Find the pair of neurons (or subpopulations if multiple neurons have the exact same φ) with the smallest value of d_{ij}.
2. Compute $\bar{\varphi}$ as the weighted angular average of all φ for the set of neurons in step 1, with weights given by $\Delta\varphi_j^{-1}$. Similary compute the average value of $\Delta\varphi$.
3. For all neurons in the set found in step 1, replace φ with $\bar{\varphi}$ and $\Delta\varphi$ with its average.
4. Repeat steps 1-3 until no pair of neurons with distinct φ have $d_{ij} < 1$.

Adjusting for finite sample effects

We now describe how we estimated the information transmitted by a set of neural responses $\{r_k\}$ about a set of stimulus samples $\tilde{D}$. The process is the same for estimating the information transmitted by the standard and information-preserving population vectors, as they are also discrete random variables of known cardinality. Our information estimate is the finite sample approximation of Shannon's Mutual Information

$$\hat{I}(\tilde{D}) = -\sum_{\{r_k\}} \hat{P}(\{r_k\}) \log_2 \hat{P}(\{r_k\}) - \frac{1}{T}\sum_t \left(-\sum_{\{r_k\}} \hat{P}_t(\{r_k\}) \log_2 \hat{P}_t(\{r_k\}) \right), \tag{4}$$

where $\hat{P}_t(\{r_k\})$ is the empirical probability of the population response equalling $\{r_k\}$ at time bin t, computed across repeats. The marginal distribution $\hat{P}(\{r_k\})$ is simply the average of $\hat{P}_t(\{r_k\})$ across time bins:

$$\hat{P}(\{r_k\}) = \frac{1}{T}\sum_t \hat{P}_t(\{r_k\}). \tag{5}$$

Because $\hat{I}$ is a biased estimate of the true mutual information for finite samples [Treves and Panzeri, 1995, Strong et al., 1998], we corrected for finite sample effects by subsampling $\tilde{D}$ using the approach [Treves and Panzeri, 1995]. Specifically, we computed $\hat{I}$ using a fraction f of the repeats, for $f \in \{1.0, 0.95, 0.90, 0.85\}$, sampling repeats without replacement. We

performed this subsampling ten times for each value of f. We perform linear regression on the values of $\hat{I}$ vs f^{-1} and extrapolate to $f^{-1} = 0$, the limit of infinite sample size [Strong et al., 1998]. We report the extrapolated value as our final estimate. The resulting information value $\hat{I}$ has the units of bits. To convert $\hat{I}$ to bits per second, we multiply by it by $\nu_{max}/0.03$.

Computing Mutual Information from Simulations

For Figure 2 we plotted the Shannon Mutual Information (I) under various settings of the population parameters, for various values of the population size N. To compute $I(\vec{r}, \vec{s})$ we use the following formulation of I:

$$I(\vec{r}, \vec{s}) = -\sum_{\{r_k\}} P(\{r_k\}) \log P(\{r_k\}) - \int d\vec{s} P(\vec{s}) \sum_{\{r_k\}} P(\{r_k\}|\vec{s}) \log P(\{r_k\}|\vec{s}), \tag{6}$$

where $P(\{r_k\}) = \int d\vec{s} P(\vec{s}) P(\{r_k\}|\vec{s})$. Expectation with respect to $P(\vec{s})$ was approximated by averaging over $N_s = 5000$ samples drawn from $P(\vec{s})$, which was the uniform distribution on the two dimensional unit circle. Because both the information-preserving population vector $\vec{M}$ and the standard population vector $\vec{U}$ are discrete random variables like $\{r_k\}$, $P(\vec{M}|\vec{s})$ and $P(\vec{U}|\vec{s})$ are computed by pooling $P(\{r_k\}|\vec{s})$ across all $\{r_k\}$ that map to the same value of $\vec{M}$ or $\vec{u}$. These discrete mappings are precomputed. Once we have computed $P(\vec{M}|\vec{s})$ and $P(\vec{U}|\vec{s})$, $I(\vec{M}, \vec{s})$ and $I(\vec{U}, \vec{s})$ are computed in the same way as $I(\vec{r}, \vec{s})$.

The simulated neural populations had the following parameters in Figure 2. In panels A and B $\varphi_i = 0$ for all i. In panel A the β values are uniformly distributed on a $\log_{10}$ scale between 0.1 and 10, and the α values are set so that the peak firing rate equals 0.8 for all neurons (see Eq. (S5) below). In panel B all β are set equal to 1 and the α values are set so that the peak firing rate varies uniformly between 0.4 and 0.8. In panels C and D the population is divided into subpopulations of equal size with preferred orientations at ± 45 degrees. In panel C, for both subpopulations, the β and α values are set as in panel A. In panel D the same α and β values are used as in panel C though the peak firing rate may differ from 0.8 due to the presence of interneuronal coupling induced by noise correlations. The noise correlations parameters were set according to Eq. (S25) below.

In the case of neural populations tuned to the same stimulus feature, we also computed information while binning the response statistics. The standard population vector reduces in this case to the population count variable U_{count}. The information-preserving population vector also becomes a scalar variable $M_{\text{count}} = \sum r_i \beta_i$. To compute the binned versions of these quantities U_{bin} and M_{bin}, we divided the support of either M_{count} or U_{count} variables into 15 equal sized bins. We then computed the mappings assigning values of $\{r_K\}$ to values of M_{bin} and U_{bin}, using the mappings from $\{r_K\}$ to M_{count} and U_{count} as an intermediate step. The results are included in Figure 2 as dotted lines. They indicate that even a small number of bins is sufficient to capture essentially all the information provided by the responses of a neural population.

Text S1 Linear-Nonlinear Model and Orientation Tuning

We begin by modeling neural responses from individual neurons as a binary variable r taking a value 1 when the neuron produces a spike and 0 otherwise. To account for response saturation and rectification, we model the probability of a spike ($r = 1$) as a saturating function of the stimulus projection onto the neuron's receptive field $\vec{w}_k$. Specifically, we choose the logistic function in order to take advantage of the properties of exponential families described below.

$$P(r_k = 1|\vec{s}) = \frac{1}{1 + \exp(-2\beta_k(x - \alpha_k))}, \quad x = \vec{w}_k \cdot \vec{s} \tag{S1}$$

Here, vector $\vec{s}$ represents the current stimulus, $\vec{w}_k$ represents the preferred stimulus or receptive field (RF) of the kth neuron, and x is the component of the stimulus along the receptive field. The parameters α_k and β_k describe, respectively, the midpoint and slope of the logistic function (Figure 1a). As a matter of notation neurons will be indexed by the letters i, j, and k and dimensions of the stimulus or neural receptive fields will be indexed by a, b, c, and d. The RF can be thought of as a pattern of unit contrast that, if presented, would elicit the strongest response from the neuron. Both $\vec{s}$ and $\vec{w}_k$ are D-dimensional vectors and $\vec{w}_k$ is assumed to be normalized. We note that if a neuron with an orientation sensitive receptive field, such as the one shown in the inset of Figure 1a, were probed by stimuli of oriented gratings with fixed contrast level then the nonlinearity described in (S1) would yield a typical tuning curve around the preferred orientation (Fig. 1B). Instead of considering such a high dimensional receptive field we work with a simplified model of orientation tuning in order to provide a clearer link between the parameters α_k, β_k and the shape of the orientation tuning curve. Specifically, for a neuron with preferred orientation φ_k we define RF as $\vec{w}_k = (\cos(\varphi_k), \sin(\varphi_k))^T$, while stimuli are described by $\vec{s}(\theta) = (\cos(\theta), \sin(\theta))^T$. Thus, in the framework of the linear-nonlinear model, the probability to observe a spike is given by:

$$P(r_k = 1|\theta) = \frac{1}{1 + e^{-2\beta_k(\cos(\varphi_k - \theta) - \alpha_k)}}. \tag{S2}$$

The maximal spike rate is achieved for $\theta = \phi_k$, which is given by:

$$P_{0,k} = \frac{1}{1 + e^{-2\beta_k(1 - \alpha_k)}}. \tag{S3}$$

The width of the orientation tuning curve, which we define as the inverse of the second derivative of the logarithm of the tuning curve is:

$$\delta_k \equiv \left(-\frac{d^2}{d\theta^2} \ln \left(P(r_k = 1|\theta)\Big|_{\theta=\varphi_k} \right) \right)^{-1} = \frac{1}{2\beta_k(1 - P_{0,k})}. \tag{S4}$$

We can invert the above equations to express α_k and β_k in terms of $P_{0,k}$ and δ_k:

$$\begin{aligned} \alpha_k &= 1 - \delta_k(1 - P_{0,k}) \ln \left(\frac{P_{0,k}}{1 - P_{0,k}} \right), \\ \beta_k &= \frac{1}{2\delta_k(1 - P_{0,k})}. \end{aligned} \tag{S5}$$

Thus, there is a one-to-one correspondence between parameters of tuning curves and nonlineary of the LN model.

Text S2 Information-Preserving Population Vector

Population response probability is an Exponential Family

We start by describing the case where neural responses are conditionally independent given the stimulus; the generalization to the case of correlated neural variability will be discussed in Text S3. Further, we will begin our arguments by considering the responses of a population of neurons at one time point, and in a time window small enough for the responses of individual neurons to be binary. At the end of Text S2 we will discuss how the results can be generalized to tackle the case of longer time windows where neurons produce multiple spikes.

It is helpful to re-write the response function of an individual neuron (S1) in an exponential form:

$$P(r_k|\vec{s}) = \frac{e^{(2r_k-1)\beta_k(\vec{w}_k\cdot\vec{s}-\alpha_k)}}{2\cosh(\beta_k(\vec{w}_k\cdot\vec{s}-\alpha_k))} = e^{h_k(r_k)+2\vec{s}\cdot\vec{M}_k(r_k)-A_k(\vec{s})}, \tag{S6}$$

where we have defined the functions $h_k(r_k)$, $\vec{M}_k(r_k)$, and $A_k(\vec{s})$ for notational convenience:

$$h_k(r_k) = -\beta_k\alpha_k(2r_k-1), \quad A_k(\vec{s}) = \ln(2\cosh(\beta_k(\vec{w}_k\cdot\vec{s}-\alpha_k))) - \beta_k\vec{w}_k\cdot\vec{s}, \quad \vec{M}_k(r_k) = \beta_k r_k\vec{w}_k. \tag{S7}$$

For conditionally independent responses, the probability to observe a response pattern $r_1, r_2, \ldots, r_N$ across N neurons to stimulus $\vec{s}$ is:

$$P(\{r_1,\ldots,r_N\}|\vec{s}) = \prod_i^N P(r_i|\vec{s}). \tag{S8}$$

Using Eq. (S6), this probability distribution can also be written in the exponential form:

$$P(\{r_k\}|\vec{s}) = \exp(h(\{r_k\}) + 2\vec{s}\cdot\vec{M}(\{r_k\}) - A(\vec{s})), \tag{S9}$$

where the functions $h(\{r_k\})$, $\vec{M}(\{r_k\})$ and $A(\vec{s})$ are the summations of the corresponding individual neuron functions:

$$h(\{r_k\}) = \sum_k h_k(r_k), \quad A(\vec{s}) = \sum_k A_k(\vec{s}), \quad \vec{M}(\{r_k\}) = \sum_k \vec{M}_k(r_k). \tag{S10}$$

The vector $\vec{M}(\{r_k\})$ is the information-preserving population vector whose expression we explicitly write out because of its importance:

$$\vec{M}(\{r_k\}) = \sum_k \beta_k r_k\vec{w}_k. \tag{S11}$$

It provides a mapping from a set of neural responses $\{0,1\}^N$ to a finite subset of $\mathbb{R}^d$. The important conclusion is that the population response model (S9) forms an exponential family with natural parameter $\vec{s}$ and sufficient statistic [Wainwright and Jordan, 2008] $\vec{M}(\{r_k\})$ that corresponds to the information-preserving population vector. We will see in Sec. Text S3 that the population response model with correlated variability across neurons also forms an exponential family As a matter of terminology, a single exponential family is considered to have fixed values of $\{\alpha_k\}$, $\{\beta_k\}$, and $\{\vec{w}_k\}$ with different members of the family indexed by different stimulus values of $\vec{s}$.

Sufficient statistics preserve information

An important result of $P(\{r_k\}|\vec{s})$ being an exponential family is that the mutual information is preserved by the sufficient statistic [Cover and Thomas, 2012], which in our case is the information-preserving population vector M from the main text. That is, our goal is to show that:

$$I(\{r_k\},\vec{s}) = I(\vec{M},\vec{s}) \tag{S12}$$

To show this directly in our case we first define the "density-of-states" function $C(\vec{M})$ as the sum of $e^{h(\{r_k\})}$ over all $\{r_k\}$ that map to the same value of $\vec{M}$:

$$C(\vec{M}) = \sum_{\{r_k\}} e^{h(\{r_k\})}\delta(\vec{M} - \vec{M}(\{r_k\})) \tag{S13}$$

The conditional and marginal distribution of $\vec{M}$ can be expressed in terms of $C(\vec{M})$, without direct reference to $\{r_k\}$:

$$\begin{aligned} P(\vec{M}|\vec{s}) &= C(\vec{M})\exp(2\vec{s}\cdot\vec{M} - A(\vec{s})), \\ P(\vec{M}) &= C(\vec{M})\int P(\vec{s})\,\exp(2\vec{s}\cdot\vec{M} - A(\vec{s}))\,d\vec{s} \end{aligned}$$

We note the relationships between $P(\{r_k\}|\vec{s})$, $P(\{r_k\})$ and $P(\vec{M}|\vec{s})$, $P(\vec{M})$ respectively:

$$\begin{aligned} P(\vec{M}|\vec{s}) &= \sum_{\{r_k\}} P(\{r_k\}|\vec{s})\delta(\vec{M} - \vec{M}(\{r_k\})) \\ P(\vec{M}) &= \sum_{\{r_k\}} P(\{r_k\})\delta(\vec{M} - \vec{M}(\{r_k\})). \end{aligned} \tag{S14}$$

We now have the following important identity:

$$\begin{aligned} \frac{P(\{r_k\}|\vec{s})}{P(\{r_k\})} &= \frac{\exp(h(\{r_k\}) + 2\vec{s}\cdot\vec{M}(\{r_k\}) - A(\vec{s}))}{\int P(\vec{s}')\,\exp(h(\{r_k\}) + 2\vec{s}'\cdot\vec{M}(\{r_k\}) - A(\vec{s}'))\,d\vec{s}'} \\ &= \frac{\exp(2\vec{s}\cdot\vec{M}(\{r_k\}) - A(\vec{s}))}{\int P(\vec{s}')\,\exp(2\vec{s}'\cdot\vec{M}(\{r_k\}) - A(\vec{s}'))\,d\vec{s}'} \\ &= \frac{C(\vec{M})\exp(2\vec{s}\cdot\vec{M}(\{r_k\}) - A(\vec{s}))}{\int P(\vec{s}')\,C(2\vec{M})\exp(\vec{s}'\cdot\vec{M}(\{r_k\}) - A(\vec{s}'))\,d\vec{s}'} \\ &= \frac{P(\vec{M}|\vec{s})}{P(\vec{M})}. \end{aligned} \tag{S15}$$

This last equality applied to Eq. (S12) for the mutual information yields:

$$I(\{r_k\},\vec{s}) = \int P(\vec{s})\sum_{\{r_k\}} P(\{r_k\}|\vec{s})\,\ln\left(\frac{P(\{r_k\}|\vec{s})}{P(\{r_k\})}\right) = \int P(\vec{s})\sum_{\{r_k\}} P(\{r_k\}|\vec{s})\,\ln\left(\frac{P(\vec{M}(\{r_k\})|\vec{s})}{P(\vec{M}(\{r_k\}))}\right)$$

This in turn yields that

$$I(\{r_k\},\vec{s}) = \int P(\vec{s})\sum_{\vec{M}} P(\vec{M}|\vec{s})\,\ln\left(\frac{P(\vec{M}|\vec{s})}{P(\vec{M})}\right) = I(\vec{M},\vec{s}). \tag{S16}$$

Another corollary of (S15) is that the posterior distribution of $\vec{s}$ given $\{r_k\}$ depends only on $\vec{M}(\{r_k\})$:

$$P(\vec{s}|\{r_k\}) = P(\vec{s}|\vec{M}(\{r_k\})). \tag{S17}$$

Therefore, a Bayes optimal decoder needs only to carry out the weighted summation rather than keep track of which response (out of 2^N possible) was observed. Similar sufficiency properties are known for Gaussian r_k as well as binary population models with independent and identically distributed neurons [Wainwright and Jordan, 2008, Ma et al., 2006]. The derivation provides the first demonstration, to our knowledge, for a sufficient statistics for a population model for binary neurons that are neither independent (see also Text S3 below) nor identically distributed and which has dimension D independent of population size (D is the stimulus dimension).

Cumulants of the information-preserving population vector $\vec{M}$

The mean value of the information-preserving population vector varies smoothly with the stimulus as we illustrate in Figure S1. To show this analytically, we provide in this sub-section analytic expressions for the first two cumulants of $\vec{M}$ as a function of $\vec{s}$. Since $\vec{s}$ is the natural parameter of the class of models we consider, the cumulants of $\vec{M}$ can be computed by taking derivatives of different orders of the log-partition function $A(\vec{s})$ with respect to $\vec{s}$. In particular the mean and covariance are the gradient and Hessian respectively:

$$\vec{T}(\vec{s}) = \langle \vec{M} \rangle_{\vec{M}|\vec{s}} = \frac{1}{2}\vec{\nabla}_{\vec{s}} A(\vec{s}), \tag{S18}$$

$$\mathbf{V}_{a,b}(\vec{s}) = \langle (M_a - T_a(\vec{s}))(M_b - T_b(\vec{s})) \rangle_{\vec{M}|\vec{s}} = \frac{1}{4}\frac{\partial^2 A(\vec{s})}{\partial s_b \partial s_a}. \tag{S19}$$

Since covariance matrices are always positive semi-definite we see here that $A(\vec{s})$ is a convex function. We also note that $\mathbf{V}_{a,b}(\vec{s})$ is the Jacobian of $\vec{T}(\vec{s})$, as well as the Fisher information matrix of $\vec{M}$ with regards to $\vec{s}$. When the neurons are conditionally independent, $\vec{T}(\vec{s})$ and $\mathbf{V}_{a,b}(\vec{s})$ take simple forms:

$$\vec{T}(\vec{s}) = \sum_k \beta_k \vec{w}_k \, \sigma(2\beta_k(\vec{w}_k \cdot \vec{s} - \alpha_k)), \quad \sigma(x) = \frac{1}{1+e^{-x}}, \tag{S20}$$

$$\mathbf{V}_{a,b}(\vec{s}) = \sum_k (\beta_k)^2 w_{a,k} w_{b,k} \sigma(2\beta_k(\vec{w}_k \cdot \vec{s} - \alpha_k)) \left(1 - \sigma(2\beta_k(\vec{w}_k \cdot \vec{s} - \alpha_k))\right) \tag{S21}$$

Because $\mathbf{V}_{a,b}(\vec{s})$ is a continuous mapping we see here that $\vec{T}(\vec{s})$ is a smooth function of $\vec{s}$. It is sometimes the case that stimulus parameter is embedded in a higher dimensional space as the natural parameter (e.g. $\vec{s}(\theta)$ has higher dimensionality then θ, a single parameter). In this case the family is said to be *curved* with respect to θ. We note that the information-preserving property also holds for curved exponential families. However, when calculating the cumulants, it is necessary to take gradients with respect to $\vec{s}$ and not θ.

Analysis of non-binary neural responses

We now discuss how these analyses can be generalized to non-binary neural responses that may appear over longer time windows. The time window size T still needs to be constrained by the stimulus dynamics to ensure that the stimulus does not change appreciably during the response time window. Nevertheless, for natural visual stimuli it would not be uncommon for stimuli to be approximately constant over the time period of $\sim$ 30 msec, given the predominance of low temporal frequencies in natural stimuli [Simoncelli and Olshausen, 2001]. Over this time window the responses of visual cortical neurons, for example, would commonly produce more than one spike. How should we treat such multiple responses? Formally, we can split the time window of interest T into smaller bins of width ΔT of such duration that the neural response can only be binary (e.g. $\sim$ 1 msec) when considered in these ΔT time intervals. The maximal number of spikes that a neuron can produce is then $N_t = T/\Delta T$. We can model responses of this neuron that as a set of N_t binary neurons with identical LN parameters. Therefore, applying the same mathematical arguments as above, one observes that the responses of these auxiliary neurons can be simply averaged without incurring information loss. (These summed responses will follow a binomial distribution.) Returning to the population of N neurons with different RFs and non-binary responses, we can expand this population to size $N_t \cdot N$ where each neuron from the original population is represented by a subpopulation of N_t neurons, with the same RF and β factors as the original neuron, and whose responses can therefore be averaged. This analysis indicates that the responses of non-binary neurons can be analyzed by fitting the neural responses to stimuli with a logistic function scaled by a constant $R_{\max}$. The inverse of the scaling constant $R_{\max}$ yields the time window duration over which the responses of this neuron can be considered binary. The expression (S11) for the information preserving population vector remains unchanged, $\vec{M}\{r_1, \dots r_N\} = \sum_i \beta_i \vec{w}^{(i)} r_i$, except now r_i are no longer binary variables and instead represent the number of spikes produced by ith neuron during time interval T. Furthermore, because the response averaging holds in the presence of noise correlations, as long as they are stimulus-independent (see Text S3 below), the averaging across time will be valid even when responses across different time bins are not independent, again as long as these correlations across time bins do not depend on the stimulus.

Text S3 Taking into account correlated variability across neurons

We now expand the population response model to allow for the presence of correlated variability among neurons observed under repeated stimulus presentations. A standard way to include such pairwise "noise" correlations between neurons is with the following probability response distribution [Granot-Atedgi et al., 2013]:

$$P(\{r_1, \dots, r_N\}|\vec{s}) = \exp\left(h(\{r_i\}) + 2\vec{s}\cdot\vec{M} - A(\vec{s}, \mathbf{J}) + \sum_{ij} r_i r_j J_{ij}\right) \tag{S22}$$

The new term $\sum_{ij} r_i r_j J_{ij}$ that describes noise couplings between neurons does not depend on the stimulus, and so can be incorporated into $h(\{r_k\})$:

$$h(\{r_k\}, \mathbf{J}) = \sum_{i,j} \mathbf{J}_{ij} r_i r_j - \sum_k \beta_k \alpha_k (2r_k - 1) \tag{S23}$$

The joint distribution on population responses is again an exponential family:

$$P(\{r_k\}|\vec{s},\mathbf{J}) = \exp\left(h(\{r_k\},\mathbf{J}) + 2\vec{s}\cdot\vec{M}(\{r_k\}) - A(\vec{s},\mathbf{J})\right) \tag{S24}$$

Importantly, the vector of sufficient statistics $\vec{M}(\{r_k\})$ is the same as before and still preserves information in neural responses. Thus, the strategy for reading out the activity does not need to be modified in the presence of correlations. The normalization factor $A(\vec{s},\mathbf{J})$ is now defined as a stimulus dependent normalizing term but in general lacks a closed-form expression similar to $A(\vec{s})$ from Eq. (S10).

In the simulations in Figure 2D, we used coupling coefficients parameterized by $\{\varphi\}$ as follows:

$$\mathbf{J}_{ij} = \frac{1+\cos(\varphi^{(i)}-\varphi^{(j)})}{10\sqrt{N}}. \tag{S25}$$

It is worth nothing that the framework remains valid in the presence of noise correlations that depend on differences in the stimulus selectivity between neuronal pairs. We note that for non-zero $\mathbf{J}$, computing $P(r_k|\vec{s})$ requires marginalizing over the states of all other neurons in the population and will in general differ from the logistic response function Eq. (S1). This observation demonstrates that it is not the logistic form Eq. (S1) that guarantees $\vec{M}$ preserves information but rather that the population's response distribution is coupled to $\vec{s}$ only through $\vec{M}(\{r_k\})$. Furthermore, since the expressions relating the cumulants of $\vec{M}(\{r_k\})$ to gradients of $A(\vec{s})$ are still valid in this case and $V_{a,b}(\vec{s})$ remains positive semi-definite by construction, the mapping described by the expected value $\vec{T}(\vec{s})$ of the information-preserving population vector remains a smooth function of stimuli $\vec{s}$.

Text S4 Stimulus Decoding from Population Vectors

We now describe convergence properties of the information-preserving population vector in the limit of large neural populations. Here, the response function of individual neurons does not have be to a logistic function for most of the important properties to hold. Therefore, we write this response function from Eq. (S1) more generally in terms of a smooth monotonic function g:

$$P(r_k = 1|\vec{s}) = g(\tilde{w}_k\cdot\vec{s} - \tilde{\alpha}_k), \tag{S26}$$

where we introduced a new notation $\tilde{w}_k = \beta_k\vec{w}_k$ for the RF scaled by its amplitude β_k. The thresholds have also been correspondingly scaled $\tilde{\alpha}_k = \beta_k\alpha_k$.

We will normalize both the information-preserving and the standard population vectors by the number of neurons N in the population

$$\begin{aligned}\bar{M} &= \frac{1}{N}\sum_k \tilde{w}_k r_k),\\ \bar{u} &= \frac{1}{N}\sum_k \frac{\tilde{w}_k}{|\tilde{w}_k|} r_k.\end{aligned}$$

By the weak law of large numbers both $\bar{M}$ and $\bar{u}$ converge in probability to their expected value as N grows large:

$$\begin{aligned} \bar{M} &\xrightarrow{p} \bar{M}(\vec{s}) \equiv \int P(\tilde{w})\, \tilde{w} g(\tilde{w}\cdot\vec{s})\, d\tilde{w} \\ \bar{u} &\xrightarrow{p} \bar{u}(\vec{s}) \equiv \int P(\tilde{w})\, \frac{\tilde{w}}{|\tilde{w}|} g(\tilde{w}\cdot\vec{s})\, d\tilde{w}, \end{aligned}$$

where $P(\tilde{w})$ describes the distribution of (scaled) RF components. We now show that when $P(\tilde{w})$ is described by a multivariate Gaussian distribution, the information-preserving vector will produce unbiased stimulus estimates, whereas biases will persist in reconstructions based on the standard population vector. For clarity of the presentation, we first consider the case where RF components have zero mean and that all neurons have the same scaled thresholds $\tilde{\alpha}_k$. Later we will show how solutions generalize to the case of unequal thresholds and nonzero RF components.

Just like in the analysis of STA convergence properties [Paninski, 2003, Sharpee et al., 2004], we can use Stein's lemma for Gaussian $P(\tilde{w})$ [Stein, 1981] that expresses averages of RF components weighted by the nonlinear function g of these components as the product of correlations between components and the average of the gradient of g:

$$\langle \tilde{w} g(\tilde{w})\rangle = C\langle \vec{\nabla} g(\tilde{w})\rangle. \tag{S27}$$

Here $\langle\cdot\rangle$ denotes expectation with respect to $P(\tilde{w})$ and $C_{ij} = \langle \tilde{w}_i \tilde{w}_j\rangle$ is the covariance of RF components across the population. Stein's lemma applies as long as g is a smooth function. [Technically, $\frac{\partial g}{\partial \tilde{w}_i}$ should exist almost everywhere and $\langle \frac{\partial g}{\partial \tilde{w}_i}\rangle < \infty$]. Applying the Stein's lemma to Eq. (S27) for the information-preserving population vector one finds that:

$$\bar{M}(\vec{s}) = C\vec{s}\langle g'(\tilde{w}\cdot\vec{s})\rangle \tag{S28}$$

In these equations, the average $\langle g'(\tilde{w}\cdot\vec{s})\rangle$ describes the compressive nonlinearity of the kind shown in Fig. S1. The important conclusion from Eq. (S28) is that the information-preserving population vector is aligned with $C\vec{S}$. The stimulus direction can therefore be determined by multiplying the information-preserving population vector by the inverse covariance matrix C^{-1}. This procedure is completely analogous to the one used to reconstruct the RF from the STA in the presence of stimulus correlations [Sharpee et al., 2004, Ringach et al., 2002].

One finds a very different answer when applying Stein's lemma to the standard population vector. In this case:

$$\bar{u}(\vec{s}) = C\vec{s}\left\langle \frac{1}{|\tilde{w}|} g'(\tilde{w}\cdot\vec{s})\right\rangle - C\left\langle \frac{\tilde{w}}{|\tilde{w}|^3} g(\tilde{w}\cdot\vec{s})\right\rangle. \tag{S29}$$

Here, the average in the second term will not be aligned with the stimulus $\vec{s}$ unless the input distribution has spherical symmetry [Chichilnisky, 2001, Paninski, 2003]. Thus, the right hand side is not aligned with $C\vec{s}$. This indicates that that decoding based on the standard population vector does not readily produce an estimate of the stimulus, even in the limit of large neural populations.

Standard numerical issues might arise when computing the inverse of the covariance matrix C if RFs primarily sample one portion of the stimulus space. These issues have been discussed in detail in the context of RF estimation [Sharpee et al., 2008,

Sharpee, 2013, Ringach et al., 2002, Theunissen et al., 2000]. In Figure 4 we plot, the correlation between stimuli $\vec{s}$ and $C^{-1}M$. However, one can also follow [Simmons et al., 2013] to bypass these issues by analyzing the correlation between $\vec{M}$ and $C\vec{s}$:

$$Corr(\vec{M}, C\vec{s}) = \frac{\vec{M} \cdot C\vec{s}}{|\vec{M}|\,|C\vec{s}|}. \tag{S30}$$

We now return to consider generalizations to the case where RF components have non-zero mean and thresholds vary among neurons. To compensate for the non-zero mean, one can subtract from the information-preserving population vector a vector of $\langle\tilde{w}\rangle\langle g(\tilde{w}\cdot\vec{s})\rangle$, the average RF scaled by the population firing rate in response to $\vec{s}$. This linear procedure can be achieved in the brain in the balanced regime [van Vreeswijk and Sompolinsky, 1996] where excitatory and inhibitory inputs are balanced on average together with homeostatic scaling of synaptic inputs.

The case of variable thresholds is treated by considering the average nonlinear function

$$g_{\text{eff}}(\tilde{w}\cdot\vec{s}) = \int P(\vec{\alpha})\, g(\tilde{w}\cdot\vec{s} - \alpha)\, d\alpha \tag{S31}$$

The above results remain valid as long as α are distributed independently of $\tilde{w}$.

Finally, the distribution of RF components might not be purely Gaussian. Such deviations will cause systematic distortions in the mapping (Fig. S1). As long as these distortions are indeed systematic, they can be learned and compensated for. Such learning also needs to happen every time the RF changes following adaptation to changes in the stimulus statistics. One possible approach for computing the expected deviations of the estimator Eq. (S27) for weakly non-Gaussian stimuli can be found in Appendix A of [Sharpee et al., 2004].

The question of what sets of receptive fields might provide maximal information about natural scenes and allow for their accurate reconstruction represents an active area of research [Olshausen and Field, 1996, Olshausen and Field, 1997, Henniges et al., 2010, Bornschein et al., 2013]. Here we pursued a separate question of how to decode neural responses based on a fixed set of receptive field while minimizing information loss. Finding optimal receptive field parameters is an important task for future research, with results that will likely differ for linear-nonlinear or quadratic decoding.

Receptive field decorrelation by recurrent networks with divisive normalization

According to Eq. (S28), the information-preserving population vector yields an estimate of the stimulus $\vec{s}$ multiplied by the covariance matrix C of neural RFs in the population. This systematic shift can be compensated for by "decorrelating" RFs to applying such transformations that the resultant covariance matrix becomes proportional to a unit matrix. We now discuss how divisive normalization in a recurrent network can approximate this operation.

In the divisive normalization model [Carandini and Heeger, 2012] neural responses depend on the activation of other neurons in the network as follows:

$$r_k(\vec{s}) = \frac{g(\tilde{w}^{(k)}\cdot\vec{s})}{1 + \sum_j \epsilon_{jk} g(\tilde{w}^{(j)}\cdot\vec{s})}, \tag{S32}$$

where $g(\tilde{w}^{(k)}\cdot\vec{s})$ describes the activation function of the kth neuron without taking into account recurrent connections. The activation function $g(x)$ is again a smooth, monotonically increasing function. The parameters ϵ_{jk} describe the strength of the

recurrent connection from neuron j to neuron k. In general they are not symmetric, $\epsilon_{jk} \neq \epsilon_{kj}$. We let $\vec{\varepsilon}^{(k)} \equiv (\epsilon_{1k}, ..., \epsilon_{Nk})^T$ denote the vector of incoming connections to neuron k.

We consider effective receptive fields $\tilde{w}_{eff}^{(k)} = C^{-\frac{1}{2}}\tilde{w}^{(k)}$ (the Cholesky decomposition of C^{-1}), so that the covariance matrix of $\tilde{w}_{eff}$ across the population is the D-dimensional identity matrix $\mathbf{I}$. We seek to find a setting of $\vec{\varepsilon}^{(k)}$ so that $r_k(\vec{s}) \approx g(\tilde{w}_{eff}^{(k)} \cdot \vec{s})$ as close as possible:

$$\vec{\varepsilon}_*^{(k)} = \arg\min_{\vec{\varepsilon}^{(k)}} \frac{1}{2} \left\langle \left(r_k(\vec{s}) - g(\tilde{w}_{eff}^{(k)} \cdot \vec{s}) \right)^2 \right\rangle_{\vec{s}}, \tag{S33}$$

where the stimuli are drawn from a distribution $P(\vec{s})$. We note that the optimization in (S33) can be carried out independently for each k. We performed an example computation with the following parameters: $D = 2$, $N = 500$, $P(\vec{s})$ was a white noise gaussian that we drew 1,000 samples from. Rfs $\tilde{w}$ were drawn from a zero mean Gaussian distribution with the following covariance matrix:

$$C = \begin{pmatrix} 3.0 & 1.05 \\ 1.05 & 0.5 \end{pmatrix} \tag{S34}$$

The optimization in (S33) was computed using L-BFGS algorithm, and ϵ_{jk} was initialized as $\cos(\theta_{jk})/N$ where θ_{jk} is the angle between $\tilde{w}^j$ and $\tilde{w}^k$. This initialization was chosen to ensure that no numerical overflow occurred in the evalation of $r_k(\vec{s})$. After finding $\vec{\varepsilon}_*^{(k)}$ for all k, we computed the pearson correlation between ϵ_{jk} and $\cos(\theta_{jk})$. For this example we found a correlation of 0.02 with p-value $p = 4.2 \times 10^{-23}$, indicating a weak but statistically significant correlation between the strength of recurrent connections and receptive field overlaps, as is observed experimentally [Yoshimura and Callaway, 2005, Yoshimura et al., 2005].

Text S5 Quadratic Expansion of the Input Space

To account for non-monotonic spike probabilities as well as the dependence on multiple stimulus components, one can follow the framework of minimal models [Fitzgerald et al., 2011] to expand the stimulus by including all pairwise products between stimulus components. If the original stimulus has D components, the expanded stimulus will have $D + D^2$ components of the following form:

$$\vec{\zeta}(\vec{s}) \equiv \{\vec{s},\ \vec{s}\,\vec{s}^T\}. \tag{S35}$$

The neural response probability

$$P(r_k = 1|\vec{s}) = \frac{1}{1 + \exp(-2(\tilde{w}_k \cdot \vec{s} - \tilde{\alpha}_k + \vec{s}^T \boldsymbol{\gamma}_k \vec{s}))} \tag{S36}$$

can be compactly written as

$$P(r_k = 1|\vec{s}) = \frac{1}{1 + \exp(-2(\tilde{W}_k \cdot \vec{\zeta} - \tilde{\alpha}_k)} \tag{S37}$$

in terms of the RF $\tilde{W}$ of $D + D^2$ dimensions in the expanded input space. Thus, the information-preserving properties

analyzed in Text S2 and decoding properties discussed in Text S4 will hold. We note that the quadratic matrix γ can have both positive and negative eigenvalues to describe both excitatory and suppressive stimulus dimensions [Rust et al., 2005].

The information-preserving population vector will now also have $D + D^2$ components:

$$\vec{M}_{\text{expanded}}(\{r_k\}) \quad = \quad \{\vec{M}^{\text{lin}}(\{r_k\}),\, \vec{M}^{\text{quad}}(\{r_k\}),$$

where

$$\vec{M}^{\text{lin}}(\{r_k\}) = \sum_k \tilde{w}_k(2r_k - 1), \quad \vec{M}^{\text{quad}}(\{r_k\}) = \sum_k \boldsymbol{\gamma}_k(2r_k - 1). \tag{S38}$$

If both $\tilde{w}$ and $\boldsymbol{\gamma}$ are normally distributed, then, following the arguments from Text S4, the information-preserving population vector in the expanded space will be aligned with $C_{\text{expanded}}\zeta$. Here, C_{expanded} is the covariance matrix of the $D + D^2$ dimensional vector $\{\tilde{w}, \boldsymbol{\gamma}\}$. On a particular trial, $\vec{M}_{\text{expanded}}$ may produce a direction ζ that does not satisfy the constraint of Eq. (S35) associated with quadratic expansion of the stimulus. However, we can search for a value of ζ_{est} that satisfies this constraint and is most consistent with $\vec{M}_{\text{expanded}}$. This will in turn produce a stimulus estimate in the original input space. To find such a pattern, we note that $\vec{M}^{\text{quad}}$ is a symmetric matrix. The best rank one approximation of this matrix, according to the Eckart–Young–Mirsky theorem, is $\lambda_1 \vec{e}_1 \vec{e}_1^T$, where λ_1 is the largest (in absolute value) eigenvalue of $\vec{M}^{\text{quad}}$ and $\vec{e}_1$ its associated eigenvector. We set $\hat{q} = \pm\vec{e}_1$ and resolve the ambiguity in sign by requiring consistency with the estimate derived from $\vec{M}^{\text{lin}}$.

The power iteration algorithm can be used to find $\vec{e}_1$. For our purposes, we initialize $\hat{q}_0$ randomly and apply the following iteration:

$$\hat{q}_{t+1} = \frac{\vec{M}^{\text{quad}}\,\hat{q}_t}{|\vec{M}^{\text{quad}}\,\hat{q}_t|} \tag{S39}$$

$\hat{q}_t$ will eventually converge to $\vec{e}_1$. We note that the iteration in (S39) involves a sequential application of matrix multiplication by $\vec{M}^{\text{quad}}$ (recurrent processing), followed by normalization of the resultant vector (gain normalization). The vector $\hat{q}$ provides the best approximation of M_{expanded} that can be constructed from D dimensional vectors as $\{\hat{q},\, \hat{q}\,\hat{q}^T\}$.

For the decoding of V1 data, we took $\{\tilde{w}_k^{(1)}\}$ and $\{\tilde{w}_k^{(2)}\}$ to be the first and second receptive field components of [Sharpee et al., 2006]. The quadratic kernel γ_k is computed as $\beta_{11,k} \cdot \tilde{w}_k^{(1)}\tilde{w}_k^{(1)T} + \beta_{22,k} \cdot \tilde{w}_k^{(2)}\tilde{w}_k^{(2)T} + \beta_{12,k} \cdot \tilde{w}_k^{(1)}\tilde{w}_k^{(2)T} + \beta_{12,k} \cdot \tilde{w}_k^{(2)}\tilde{w}_k^{(2)T}$, where coefficients β_{11}, β_{12} and β_{22} are defined in Eq. (2). In this case it can be shown that under the assumption that RF components $\tilde{w}$ are described by a Gaussian distribution with zero mean, the covariance matrix in the expanded space allows a decomposition such that

$$C_{\text{expanded}}\zeta = \{C\vec{s},\, C_2\vec{s}\,(C_2\vec{s})^T\}, \tag{S40}$$

where $C_2 = \frac{1}{N}\sum_k \lambda_{1,k}\tilde{e}_k\tilde{e}_k^T + \lambda_{2,k}\tilde{u}_k\tilde{u}_k^T$, $\lambda_{1,k}$ and $\lambda_{2,k}$ are the first and second eigenvalues of the quadratic kernel for kth neuron, and $\tilde{e}_k$ and $\tilde{u}_k$ are their corresponding eigenvectors. Specifically, $\lambda_{1,2} = \frac{1}{2}\left(\beta_{11} + \beta_{22} \pm (\beta_{11} - \beta_{22})/\cos(2\theta)\right)$ and $\theta = \arctan(2\beta_{12}/(\beta_{22} - \beta_{11}))/2$. Thus, we did not need to compute the full covariance matrix in the expanded space. Comparing Eq. (S40) and the definition of $\hat{q}$, we can find the stimulus direction based on quadratic decoder by applying C_2^{-1} to $\hat{q}$. Alternatively, to avoid inversion of a poorly conditioned C matrix, we compute, just as in Text S4, the vector

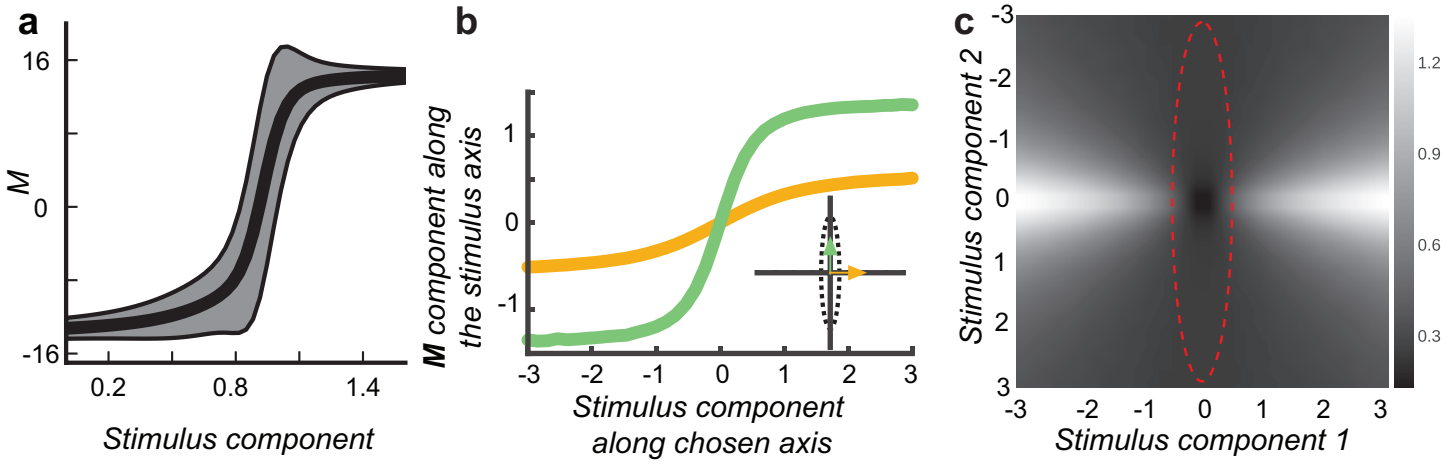

Figure S1: (**A**) The expected value of the information-preserving spike count varies smoothly as a function of the stimulus component along the RF. Gray shading represents one standard deviation around the mean (black line). (**B**) Illustration of the compressive nonlinearity in a population tuned to different input features. The curves relate the magnitude of stimuli along one of two axes in the input space (green or yellow) to the magnitude of the information-preserving population vector along these axes. Green/yellow curves are for directions of maximal/minimal variance of RF components. (**C**) Map of the compressive nonlinearity showing the magnitude of the information-preserving population vector multiplied by C^{-1} (to correct for differences in RF distribution) for different stimuli, see also Text S4. The red dotted line shows the standard deviation of the RF distribution.

correlation between $\hat{q}$ and $C_2\vec{s}$.

References

[Abbott and Dayan, 1999] Abbott, L. F. and Dayan, P. (1999). The effect of correlated variability on the accuracy of a population code. Neural Comput. *11*, 91–101.

[Atencio et al., 2009] Atencio, C. A., Sharpee, T. O. and Schreiner, C. E. (2009). Hierarchical Computation in the Canonical Auditory Cortical Circuit. PNAS *106*, 21894–21899.

[Averbeck et al., 2006] Averbeck, B. B., Latham, P. E. and Pouget, A. (2006). Neural correlations, population coding and computation. Nat Rev Neurosci *7*, 358–366.

[Bornschein et al., 2013] Bornschein, J., Henniges, M. and ücke, J. (2013). Are V1 simple cells optimized for visual occlusions? A comparative study. PLoS Comput Biol *9*, e1003062.

[Carandini and Heeger, 2012] Carandini, M. and Heeger, D. J. (2012). Normalization as a canonical neural computation. Nature Reviews Neuroscience *13*, 51–62.

[Chichilnisky, 2001] Chichilnisky, E. J. (2001). A simple white noise analysis of neuronal light responses. Network: Comput. Neural Syst *12*, 199–213.

[Cover and Thomas, 2012] Cover, T. M. and Thomas, J. A. (2012). Elements of information theory. John Wiley & Sons.

[Dragoi et al., 2001] Dragoi, V., Turcu, C. M. and Sur, M. (2001). Stability of cortical responses and the statistics of natural scenes. Neuron *32*, 1181–1192.

[Ecker et al., 2011] Ecker, A. S., Berens, P., Tolias, A. S. and Bethge, M. (2011). The effect of noise correlations in populations of diversely tuned neurons. J. Neurosci. *31*, 14272–14283.

[Fitzgerald et al., 2011] Fitzgerald, J. D., Rowekamp, R. J., Sincich, L. C. and Sharpee, T. O. (2011). Second order dimensionality reduction using minimum and maximum mutual information models. PLoS Comput. Biol. *7*, e1002249.

[Georgopoulos et al., 1986] Georgopoulos, A. P., Schwartz, A. B. and Kettner, R. E. (1986). Neuronal population coding of movement direction. Science *233*, 1416–1419.

[Granot-Atedgi et al., 2013] Granot-Atedgi, E., Tkacik, G., Segev, R. and Schneidman, E. (2013). Stimulus-dependent maximum entropy models of neural population activity. Plos Comp Biol *9*, e1002922.

[Harris and Mrsic-Flogel, 2013] Harris, K. D. and Mrsic-Flogel, T. D. (2013). Cortical connectivity and sensory coding. Nature *503*, 51.

[Harris and Shepherd, 2015] Harris, K. D. and Shepherd, G. M. (2015). The neocortical circuit: themes and variations. Nature Neuroscience *18*, 170–181.

[Henniges et al., 2010] Henniges, M., Puertas, G., Bornschein, J., Eggert, J. and Lücke, J. (2010). Binary sparse coding. Latent Variable Analysis and Signal Separation *6365*, 450–457.

[Herculano-Houzel, 2009] Herculano-Houzel, S. (2009). The human brain in numbers: a linearly scaled-up primate brain. Frontiers in human neuroscience *3*.

[Hohl et al., 2013] Hohl, S. S., Chaisanguanthum, K. S. and Lisberger, S. G. (2013). Sensory population decoding for visually guided movements. Neuron *79*, 167–179.

[Huang and Lisberger, 2009] Huang, X. and Lisberger, S. G. (2009). Noise correlations in cortical area MT and their potential impact on trial-by-trial variation in the direction and speed of smooth-pursuit eye movements. Neuron *101*, 3012–3030.

[Jiang et al., 2015] Jiang, X., Shen, S., Cadwell, C. R., Berens, P., Sinz, F., Ecker, A. S., Patel, S. and Tolias, A. S. (2015). Principles of connectivity among morphologically defined cell types in adult neocortex. Science *350*, aac9462.

[Joshua and Lisberger, 2015] Joshua, M. and Lisberger, S. G. (2015). A tale of two species: Neural integration in zebrafish and monkeys. Neurocience *296*, 80–91.

[Kastner et al., 2015] Kastner, D. B., Baccus, S. A. and Sharpee, T. O. (2015). Critical and maximally informative encoding between neural populations in the retina. Proceedings of the National Academy of Sciences *112*, 2533–2538.

[Lewis and Kristan, 1998] Lewis, J. E. and Kristan, W. B. (1998). A neuronal network for computing population vectors in the leech. Nature *391*, 76–79.

[Luo et al., 2017] Luo, C., Keown, C. L., Kurihara, L., Zhou, J., He, Y., Li, J., Castanon, R., Lucero, J., Nery, J. R., Sandoval, J. P., Bui, B., Sejnowski, T. J., Harkins, T. T., Mukamel, E. A., Behrens, M. M. and Ecker, J. R. (2017). Single-cell methylomes identify neuronal subtypes and regulatory elements in mammalian cortex. Science *357*, 600–604.

[Ma et al., 2006] Ma, W. J., Beck, J., Latham, P. E. and Pouget, A. (2006). Bayesian inference with probabilistic population codes. Nature Neurosci. *9*, 1432–1438.

[Markram et al., 2015] Markram, H., Muller, E., Ramaswamy, S., Reimann, M. W., Abdellah, M., Sanchez, C. A., Ailamaki, A., Alonso-Nanclares, L., Antille, N., Arsever, S., Kahou, G. A., Berger, T. K., Bilgili, A., Buncic, N., Chalimourda, A., Chindemi, G., Courcol, J. D., Delalondre, F., Delattre, V., Druckmann, S., Dumusc, R., Dynes, J., Eilemann, S., Gal, E., Gevaert, M. E., Ghobril, J. P., Gidon, A., Graham, J. W., Gupta, A., Haenel, V., Hay, E., Heinis, T., Hernando, J. B., Hines, M., Kanari, L., Keller, D., Kenyon, J., Khazen, G., Kim, Y., King, J. G., Kisvarday, Z., Kumbhar, P., Lasserre, S., Le Be, J. V., Magalhaes, B. R., Merchan-Perez, A., Meystre, J., Morrice, B. R., Muller, J., Munoz-Cespedes, A., Muralidhar, S., Muthurasa, K., Nachbaur, D., Newton, T. H., Nolte, M., Ovcharenko, A., Palacios, J., Pastor, L., Perin, R., Ranjan, R., Riachi, I., Rodriguez, J. R., Riquelme, J. L., Rossert, C., Sfyrakis, K., Shi, Y., Shillcock, J. C., Silberberg, G., Silva, R., Tauheed, F., Telefont, M., Toledo-Rodriguez, M., Trankler, T., Van Geit, W., Diaz, J. V., Walker, R., Wang, Y., Zaninetta, S. M., DeFelipe, J., Hill, S. L., Segev, I. and Schurmann, F. (2015). Reconstruction and simulation of neocortical microcircuitry. Cell *163*, 456–492.

[Moreno-Bote et al., 2014] Moreno-Bote, R., Beck, J., Kanitscheider, I., Pitkow, X., Latham, P. and Pouget, A. (2014). Information-limiting correlations. Nature Neuroscience *17*, 1410–1417.

[Movshon et al., 1978] Movshon, J. A., Thompson, I. D. and Tolhurst, D. J. (1978). Receptive field organization of complex cells in the cat's striate cortex. The Journal of Physiology *283*, 79–99.

[Olsen et al., 2012] Olsen, S. R., Bortone, D. S., Adesnik, H. and Scanziani, M. (2012). Gain control by layer six in cortical circuits of vision. Nature *483*, 47–52.

[Olshausen and Field, 1996] Olshausen, B. A. and Field, D. J. (1996). Natural image statistics and efficient coding. Network: computation in neural systems *7*, 333–339.

[Olshausen and Field, 1997] Olshausen, B. A. and Field, D. J. (1997). Sparse coding with an overcomplete basis set: A strategy employed by V1? Vision Research *37*, 3311–3325.

[Osborne et al., 2008] Osborne, L. C., Palmer, S. E., G., L. S. and Bialek, W. (2008). The neural basis for combinatorial coding in a cortical population response. J Neurosci *28*, 13522–13531.

[Oswald et al., 2013] Oswald, M. J., Tantirigama, M. L., Sonntag, I., Hughes, S. M. and Empson, R. M. (2013). Diversity of layer 5 projection neurons in the mouse motor cortex. Frontiers in cellular neuroscience *7*.

[P Buxhoeveden, 2012] P Buxhoeveden, D. (2012). Minicolumn size and human cortex. Progress in brain research *195*, 219–35.

[Paninski, 2003] Paninski, L. (2003). Convergence properties of three spike-triggered average techniques. Network: Comput. Neural Syst. *14*, 437–464.

[Peters and Sethares, 1996] Peters, A. and Sethares, C. (1996). Myelinated axons and the pyramidal cell modules in monkey primary visual cortex. Journal of Comparative Neurology *365*, 232–255.

[Reich et al., 2001] Reich, D. S., Mechler, F. and Victor, J. (2001). Independent and redundant information in nearby cortical neurons. Science *294*, 2566–2568.

[Ringach et al., 2002] Ringach, D. L., Hawken, M. J. and Shapley, R. (2002). Receptive field structure of neurons in monkey visual cortex revealed by stimulation with natural image sequences. Journal of Vision *2*, 12–24.

[Rust et al., 2005] Rust, N. C., Schwartz, O., Movshon, J. A. and Simoncelli, E. P. (2005). Spatiotemporal elements of macaque V1 receptive fields. Neuron *46*, 945–956.

[Salinas and Abbott, 1994] Salinas, E. and Abbott, L. F. (1994). Vector reconstruction from firing rates. J Comp Neurosci *1*, 89–107.

[Schwartz et al., 2006] Schwartz, O., Pillow, J., Rust, N. and Simoncelli, E. P. (2006). Spike-triggered neural characterization. Journal of Vision *6*, 484–507.

[Shamir, 2014] Shamir, M. (2014). Emerging principles of population coding: in search for the neural code. Curr. Opin. Neurob. *25*, 140–148.

[Shamir and Sompolinsky, 2004] Shamir, M. and Sompolinsky, H. (2004). Nonlinear population codes. Neural Comput *16*, 1105–1136.

[Shamir and Sompolinsky, 2006] Shamir, M. and Sompolinsky, H. (2006). Implications of neuronal diversity on population coding. Neural Comput *18*, 1951–1986.

[Sharpee, 2013] Sharpee, T. (2013). Computational identification of receptive fields. Annu Rev. Neurosci. *36*, 103–120.

[Sharpee et al., 2004] Sharpee, T., Rust, N. and Bialek, W. (2004). Analyzing neural responses to natural signals: Maximally informative dimensions. Neural Computation *16*, 223–250.

[Sharpee et al., 2008] Sharpee, T. O., Miller, K. D. and Stryker, M. P. (2008). On the importance of static nonlinearity in estimating spatiotemporal neural filters with natural stimuli. J. Neurophysiol. *99*, 2496–2509.

[Sharpee et al., 2006] Sharpee, T. O., Sugihara, H., Kurgansky, A. V., Rebrik, S. P., Stryker, M. P. and Miller, K. D. (2006). Adaptive filtering enhances information transmission in visual cortex. Nature *439*, 936–942.

[Simmons et al., 2013] Simmons, K. D., Prentice, J. S., Tkačik, G., Homann, J., Yee, H. K., Palmer, S. E., Nelson, P. C. and Balasubramanian, V. (2013). Transformation of stimulus correlations by the retina. PLoS Comput Biol *9*, e1003344.

[Simoncelli and Olshausen, 2001] Simoncelli, E. P. and Olshausen, B. A. (2001). Natural image statistics and neural representation. Annu. Rev. Neurosci. *24*, 1193–1216.

[Sompolinsky et al., 1988] Sompolinsky, H., Crisanti, A. and Sommers, H. J. (1988). Chaos in random neural networks. Phys. Rev. Lett. *61*, 259–262.

[Sporns et al., 2005] Sporns, O., Tononi, G. and Kötter, R. (2005). The human connectome: a structural description of the human brain. PLoS computational biology *1*, e42.

[Srivastava et al., 2017] Srivastava, K. H., Holmes, C. M., Vellema, M., Pack, A. R., Elemans, C. P., Nemenman, I. and Sober, S. J. (2017). Motor control by precisely timed spike patterns. Proceedings of the National Academy of Sciences *114*, 201611734.

[Stein, 1981] Stein, C. M. (1981). Estimation of the mean of a multivariate normal distribution. The annals of Statistics *9*, 1135–1151.

[Strong et al., 1998] Strong, S. P., Koberle, R., van Steveninck, R. R. d. R. and Bialek, W. (1998). Entropy and information in neural spike trains. Physical review letters *80*, 197.

[Theunissen and Miller, 1995] Theunissen, F. and Miller, J. P. (1995). Temporal encoding in nervous systems: a rigorous definition. Journal of computational neuroscience *2*, 149–162.

[Theunissen et al., 2000] Theunissen, F. E., Sen, K. and Doupe, A. J. (2000). Spectral-temporal receptive fields of nonlinear auditory neurons obtained using natural sounds. Journal of Neuroscience *20*, 2315–2331.

[Treves and Panzeri, 1995] Treves, A. and Panzeri, S. (1995). The upward bias in measures of information derived from limited data samples. Neural Comp. *7*, 399–407.

[van Vreeswijk and Sompolinsky, 1996] van Vreeswijk, C. and Sompolinsky, H. (1996). Chaos in neuronal networks with balanced excitatory and inhibitory activity. Science *274*, 1724–1726.

[Wainwright and Jordan, 2008] Wainwright, M. J. and Jordan, M. I. (2008). Graphical models, exponential families, and variational inference. Foundations and Trends® in Machine Learning *1*, 1–305.

[Wang et al., 2010] Wang, Y., Brzozowsska-Prechtl, A. and Karten, H. J. (2010). Laminar and columnar auditory cortex in avian brain. Proc Natl Acad Sci U S A *107*, 12676–12681.

[Yoshimura and Callaway, 2005] Yoshimura, Y. and Callaway, E. M. (2005). Fine-scale specificity of cortical networks depends on inhibitory cell type and connectivity. Nature Neurosci. *8*, 1552–1559.

[Yoshimura et al., 2005] Yoshimura, Y., Dantzker, J. L. and Callaway, E. M. (2005). Excitatory cortical neurons form fine-scale functional networks. Nature *433*, 868–873.

[Zhang and Sharpee, 2016] Zhang, Y. and Sharpee, T. O. (2016). A Robust Feedforward Model of the Olfactory System. PLoS Comput. Biol. *12*, e1004850.

[Zohary et al., 1994] Zohary, E., Shadlen, M. N. and Newsome, W. T. (1994). Correlated neuronal discharge rate and its implications for psychophysical performance. Nature *86*, 140–143.