Random Compressed Coding with Neurons

Simone Blanco Malerba[1], Mirko Pieropan*[1], Yoram Burak[2,3], and Rava da Silveira[1,4,5]

[1]Laboratoire de Physique de l'Ecole Normale Supérieure, ENS, Université PSL, CNRS, Sorbonne Université, Université de Paris, Paris
[2]Racah Institute of Physics, Hebrew University of Jerusalem, Jerusalem
[3]Edmond and Lily Safra Center for Brain Sciences, Hebrew University of Jerusalem, Jerusalem
[4]Institute of Molecular and Clinical Ophthalmology Basel, Basel
[5]Faculty of Science, University of Basel, Basel

August 24, 2021

Abstract

The brain encodes information about the sensory world in the activity of neural populations. In classical population coding models, the responses of single neurons are described by simple, unimodal or monotonic, smooth 'tuning curves'. But various more complex tuning curves have been observed. In the case of grid cells, for example, response periodicity imparts the population code with high accuracy. Here, we ask whether highly accurate codes require a fine response design, as in the case of grid cells, or obtain more generally. To address this question, we consider a simple, shallow network that produces complex but unstructured tuning curves, and we examine 'efficient population coding' in the output layer. Specifically, we optimize the properties of response functions in the input, 'sensory' layer in terms of the information represented in the output layer. Irregularity of the responses in the output layer gives rise to the possibility of catastrophic coding errors. Our optimization approach, unlike most efficient coding procedures which consider peripheral neurons only, yields a non-trivial solution in which the width of sensory tuning curves ensures a balance between 'global' (or catastrophic) errors and 'local' errors. In this regime, information is efficiently compressed from a large sensory layer to a small 'representation' layer, and accuracy is exponential in population size. We revisit data from monkey motor cortex in the light of our approach, where we suggest that a similar compression of information takes place. Efficient (neural) codes do not require a fine design of response properties, but can occur in the presence of randomness, as indeed pointed out by Shannon seventy years ago.

1 Introduction

Neurons convey information about the physical world by modulating their responses as a function of parameters of sensory stimuli. Classically, the mean neural response to a stimulus — referred to as the neuron's 'tuning curve' — is often described as a smooth function of a stimulus parameter with a simple monotonic or unimodal form (Hubel & Wiesel, 1959; Georgopoulos et al., 1982; Taube et al., 1990; Miller et al., 1991; Bremmer et al., 1997; Dayan & Abbott, 2001; Kayaert et al., 2005). The deviation from the mean response — the 'neural noise' — may lead to ambiguity in the identity or strength of the encoded stimulus, and the coding performance of a population of neurons as a whole is dictated by the forms of the tuning curves and the joint neural noise. In the study of population codes, the efficient coding hypothesis has served as a theoretical organizing principle. It posits that tuning curves are arranged in such a way as to achieve the most accurate coding possible given a constraint on the neural resources engaged (Barlow, 1961; Atick & Redlich, 1990; Lewicki, 2002). The latter is often interpreted as a metabolic constraint on the maximum firing rate of a single neuron or on the mean firing rate of the whole population (Zhang & Sejnowski, 1999; Bethge et al., 2002; Wang et al., 2016).

In order to tackle this constrained optimization problem in practice, tuning curves are parametrized, and the corresponding parameters are optimized. Here, the simplicity of the form of tuning curves matters: only a few

*Current affiliation: Department of Applied Science and Technology (DISAT), Politecnico di Torino, Corso Duca degli Abruzzi 24, Torino

parameters need to be optimized. A large body of literature addresses this constrained optimization problem, in particular in the perceptual domain. For example, many studies model tuning curves as Gaussian or other bell-shaped functions, and obtain the values of their **[SIMONE to RAVA: 'means and variances'. Since we are talking about functions and not distributions, I would use centers and widths, also to be coherent with the terms used later.]** centers and widths that minimize the 'perceptual' error committed when information is decoded from the activity of a population of model neurons (Zhang & Sejnowski, 1999; Deneve et al., 1999; Yaeli & Meir, 2010; Ganguli & Simoncelli, 2014; Fiscella et al., 2015). In the resulting optimal populations, and if noise among neurons is independent, the coding error typically scales like $1/\sqrt{N}$, where N is the number of model neurons (Seung & Sompolinsky, 1993). This behavior can be intuited based on the observation that the 'signal' in the neural population grows like N while the noise grows like $\sqrt{N}$, yielding a signal-to-noise ratio that increases in proportion to the square root of the population size. (In some models of population neural coding of a one-dimensional parameter, the width of tuning curves can be further optimized to yield an additional factor of $1/\sqrt{N}$; the error then scales like $1/N$ (Berens et al., 2011; Kim et al., 2020).)

Real neurons, however, can come with much more complex tuning curves than simple Gaussian or bell-shaped ones. Grid cells recorded in the enthorinal cortex offer a salient example (Hafting et al., 2005; Doeller et al., 2010; Yartsev et al., 2011; Killian et al., 2012) ; their tuning curves are multimodal and periodic as a function of spatial coordinates. A number of other examples of neurons with complex tuning curves have also been identified in other cortical regions, in different species (Kadia & Wang, 2003; Sofroniew et al., 2015; Lalazar et al., 2016; Gaucher et al., 2020; Eliav et al., 2021). It was noted early on that such richer tuning curves can give rise to greatly enhanced codes. Given the periodicity of their tuning curves, and provided that the neural population include several modules made up of cells with different periodicities (Fiete et al., 2008; Wei et al., 2015), grid cells can represent spatial location with an accuracy that scales exponentially (rather than algebraically, as above) in the number of neurons (Sreenivasan & Fiete, 2011; Mathis et al., 2012; Burak, 2014). Thus, the richer structure of individual tuning curves can be traded for a strong boost in the efficiency of the population code.

Here, we ask whether highly efficient codes of this sort must rely on finely-tuned properties, such as the tuning curves' periodicity or the arrangement of different modules in the population, or, alternatively, arise generically and robustly in populations of neurons with complex tuning curves, in the absence of any fine-tuning. We approach the question by studying the benchmark case of a random neural code: a population code that relies on irregular tuning curves that emerge from a simple, feedforward, shallow network with random synaptic weights. The input layer in the network is made up of a large array of 'sensory' neurons with classical, bell-shaped tuning curves; these neurons project to a small array of 'representation' neurons with complex tuning curves. We show that, in the resulting population code, the coding error is suppressed exponentially with the number of neurons in the population, even in the presence of high-amplitude noise.

In the context of this highly efficient code, it is not sufficient to consider a 'typical error': efficiency results from the compression of the stimulus space into the activity of a layer of neurons of comparatively small size; the price to pay for this compression is the emergence of two qualitatively distinct types of error—'local errors', in which the encoding of nearby stimuli is ambiguous, and 'global (or catastrophic) errors', in which the identity of the true stimulus is lost altogether. The efficient coding problem then translates into a trade-off between these two types of errors. In turn, this trade-off yields an optimal width of the tuning curves in the 'sensory layer': when stimulus information is compressed into a 'representation layer', tuning curves in the sensory layers have to be sufficiently wide as to prevent a prohibitive rate of global errors.

We first develop the theory for a one-dimensional input (e.g., a spatial location along a line or an angle), then generalize it to higher-dimensional inputs. The latter case is more subtle because the sensory layer itself can be arranged in a number of ways (while still operating with simple, classical tuning curves). This generalization allows us to apply our model to data from monkey motor cortex, where cells display complex tuning curves. We fit our model to the data and discuss the merit of a complex 'representation code'. Overall, our approach can be viewed as an application of the efficient coding principle to downstream ('representation') processing, as opposed to the more common applications to peripheral (sensory) processing. Our study extends earlier theoretical work on grid cells and other 'finely designed' codes by proposing that efficient compression of information can occur robustly even in the case of a random network. We reach our results by considering the geometry of population activity in a compressed, representation layer of neurons.

2 Results

We organize the description of our results as follows. First, we present, in geometric terms, the qualitative difference between a code that uses simple, bell-shaped tuning curves and one that uses more complex forms. Second, we introduce a simple model of a shallow, feedforward network of neurons that can interpolate between

simple and complex tuning curves depending on the values of its parameters. Third, we characterize the accuracy of the neural code in the limiting case of maximally irregular tuning curves. Fourth, we extend the discussion to the more general case in which an optimal code is obtained from a trade-off between local and global errors. All the above is done for the case of a one-dimensional input space. Fifth, we generalize our approach to the case of a multi-dimensional stimulus. This allows us, sixth, to apply our model to recordings of motor neurons in monkey, and to analyze the nature of population coding in that system. **[SIMONE: Added geometry paragraph.]**Seventh, we give a quantitative description of the geometry of the population response induced by our network as a function of its parameters, in particular through a measure of 'dimensionality'. Finally, we extend our model to include an additional source of noise—'input noise' in the sensory layer, in addition to the 'output noise' present in the representation layer; input noise gives rise to correlated noise downstream, and we analyze its impact on the population code.

The geometry of neural coding with simple vs. complex tuning curves

A neural code is a mapping that associates given stimuli to a probability distribution of spiking patterns; in particular, the code maps any given stimulus to a mean population activity. In the case of a continuous, one-dimensional stimulus space, the latter is mapped into a curve in the N-dimensional space of the population activity, whose shape is dictated by the form of the tuning curves of individual neurons. As an illustration, we compare the cases of three neurons with bell-shaped (here, Gaussian) tuning curves and three neurons with periodic (grid-cell-like) tuning curves with three different periods (Fig. 1A). Simple tuning curves generate a smooth population response curve, implying that similar stimuli are mapped to nearby responses; by contrast, more complex tuning curves give rise to a serpentine curve. The latter makes better use of the space of possible population responses than the former, and hence can be expected to yield higher-resolution coding. Indeed, when the population response is corrupted by noise of a given magnitude, it will elicit a smaller *local* error in the case of complex tuning than in the case of simple tuning: by 'stretching' the mean response curve over a longer trajectory within the space of possible population activities, complex tuning affords the code with higher resolution relative to the range of the encoded variable. However, this argument does not capture in full the influence of noise on the nature of coding errors. In the case of a winding and twisting mean response curve, two distant stimuli are sometimes mapped to nearby activity patterns. In the presence of noise, this geometry gives rise to *global* (or catastrophic) errors. The enhanced resolution of the neural code associated with the occurrence of global errors was also noted in the context of grid-cell coding (Welinder et al., 2008; Sreenivasan & Fiete, 2011). Because of this trade-off, whether a simple or complex coding scheme is preferable becomes a quantitative question, which depends upon the details of the structure of the encoding.

Shallow feedforward neural network as a benchmark for efficient coding

In order to address the problem mathematically, we examine the simplest possible model that generates complex tuning curves, namely a two-layer feedforward model. An important aspect of the model is that it does not rely on any finely-tuned architecture or parameter tuning: complex tuning curves emerge solely because of the variability in synaptic weights; thus, the model can be thought of as a benchmark for the analysis of population coding in the presence of complex tuning curves. The architecture of the model network and the symbols associated with its various parts are illustrated in Fig. 1B. In the first layer, a large population of L *sensory* neurons encodes a one-dimensional stimulus, x, into a high-dimensional representation. Throughout, we assume that x takes values between zero and one, without loss of generality. (If the input covered an arbitrary range, say r, then the coding error would be expressed in proportion to r. In other words, one cannot talk independently of the range of the input and of the resolution of the code. We set the range to unity in order to avoid any ambiguity.) Sensory neurons come with classical tuning curves: the mean firing rate of neuron j in response to stimulus x is given by a Gaussian with center c_j (the preferred stimulus of that neurons) and width σ:

$$u_j(x) = A \exp\left(-\frac{(x-c_j)^2}{2\sigma^2}\right). \tag{1}$$

Following a long line of models, we assume that the preferred stimuli in the population are evenly spaced, so that $c_j = j/L$. As a result, the response vector for a stimulus x_0, $\mathbf{u}(x_0)$, can be represented as a Gaussian 'bump' of activity centered at x_0.

Complex tuning curves appear in the second layer containing N *representation* neurons; we shall be interested in instances with $N \ll L$, in which efficient coding results in compression of the stimulus information from a high-dimensional to a low-dimensional representation. Each representation neuron receives random synapses from each of the sensory neurons; specifically, the elements of the all-to-all synaptic matrix, $\mathbf{W}$, are i.i.d.

Gaussian random weights with vanishing mean and variance equal to $1/L$ ($W_{ij} \sim \mathcal{N}(0, 1/L)$). In the simple, linear case that we consider, the mean response of neuron i in the second layer is thus given by

$$v_i(x) = \sum_{j=1}^{L} W_{ij} u_j(x). \tag{2}$$

Since the weights W_{ij} correspond to a given realization of a random process, they generate tuning curves, $v_i(x)$, with irregular profiles. The parameter σ is important in that it controls the smoothness of the tuning curves in the second layer: it defines the width of u_j, which in turn dictates the correlation between the values of the tuning curve v_i for two different stimuli. By the same token, the amplitude of the variations of v_i with x depends upon the value of σ. For a legitimate comparison of population codes in different networks, we set this amplitude to a constant on average,

$$\left\langle \int_0^1 dx \left[v_i(x) - \int_0^1 dx' v_i(x') \right]^2 \right\rangle_W = R, \tag{3}$$

by calibrating the value of the prefactor in Eq. (1), A. Because of the averaging over the synaptic weights, indicated by the brackets $\langle \cdot \rangle_W$, A does not depend upon a specific realization of the synaptic weights. Equation (3) corresponds to the usual constraint of 'resource limitation' in efficient coding models; it amounts to setting a maximum to the variance of the output over the stimulus space, as is commonly assumed in analyses of efficient coding in sensory systems (Atick & Redlich, 1990; Van Hateren & Ruderman, 1998; Doi et al., 2012; Zhaoping, 2014).

Returning to our geometric picture, we observe that, by changing the value of σ, we can interpolate between smooth and irregular tuning curves in the second layer (Fig. 1C). In the limiting case of large σ, representation neurons come with smooth tuning curves akin to classical ones; in the other limiting case of small σ, the mean population response curve becomes infinitely tangled. Thus, as the value of σ is decreased, the mean response curve 'stretches out' and necessarily twists and turns, in such a way as to fit within the allowed space of population responses defined by Eq. (3). A longer population response curve fills the space of population responses more efficiently and represents the stimulus at a higher resolution, but its twists and turns may result in greater susceptibility to noise.

To complete the definition of the model, we specify the nature of the noise in the neural response. We assume that neuron i in the second layer is affected by i.i.d. noise, which we denote by z_i, such that its response at each trial (in which stimulus x is presented) is given by $r_i = v_i(x) + z_i$. For the sake of simplicity, we use Gaussian noise with vanishing mean and variance equal to η^2. In most of our analyses, we suppose that responses in the first layer are noiseless and that the noise in the second layer is uncorrelated among neurons; in the last subsection, however, we relax these assumptions, and discuss the implications of noisy sensory neurons and correlated noise among representation neurons. (Our motivation for considering noiseless sensory neurons is that we are primarily interested in analyzing the compression of the representation of information between the first and the second layer of neurons. By contrast, noise in sensory neurons affects the fidelity of encoding in the *first* layer.) We quantify the performance of the code in the second layer through the mean squared error (MSE) in the stimulus estimate as obtained from an ideal decoder. The use of an ideal decoder is an abstract device that allows us to focus on the uncertainty inherent to *encoding* (rather than to imperfections in *decoding*); it is nevertheless possible to obtain a close approximation to an ideal decoder in a simple neural network with biologically plausible operations (see Methods).

Compressed coding in the limiting case of narrow sensory tuning

It is instructive to study the properties of coding in our model in the limiting case of neurons with narrow tuning curves in the sensory layer ($\sigma \to 0$), because this limit yields the most irregular tuning curves in the representation layer of our network (Fig. 1C). As we shall see, this limiting case also corresponds to that of a completely uncorrelated, random code, for which the mathematical analysis simplifies. When the value of σ is much smaller than $1/L$, neurons in the sensory layers respond only if the stimulus coincides with the preferred stimulus of one of the neurons, and only that neuron is activated by the stimulus presentation; stimulus values that lie in between the preferred stimuli of successive sensory neurons in the first layer do not elicit any activity in the system. We can thus consider that any stimulus of interest is effectively chosen within a discrete set of L stimuli with values $x_j = j/L$, with $j = 1, \ldots, L$.

Each of these stimuli elicits a mean response

$$v_i(x_j) = \tilde{A} W_{ij} \sim \mathcal{N}(0, R) \tag{4}$$

in neuron i of the second layer. Here, the value of $\tilde{A}$ is chosen so as to set the amplitude of the variations of v_i to be equal to the constant R (analogously to Eq. (3) but for the case of discrete stimuli). Geometrically, Eq. (4) represents a mapping from L stimulus values to a set of uncorrelated, random locations in the space of the population activity (as illustrated in Fig. 2A for a two-neuron population). In any given trial, however, the responses in the representation layer are corrupted by noise (Fig. 2A). The ideal decoder ('ideal' in the sense that it minimizes the mean error) interprets a single-trial response as being elicited by the stimulus associated to the nearest possible mean response (Fig. 2A). The outcome of this procedure can be twofold: either the correct or an incorrect stimulus is decoded; in the latter case, because the possible mean responses are arranged randomly in the space of population activity (Fig. 2A and Eq. (4)), errors of all magnitudes are equiprobable. In other words, a model with narrow sensory tuning curves results in a second-layer code that does not preserve distances among inputs, and, consequently, the decoding error is either vanishing or, typically, on the order of the input range (set to unity here). The mean error is then simply proportional to the probability with which the ideal decoder makes a mistake with a constant of proportionality of the order of the stimulus range.**[SIMONE to RAVA: 'The mean error can then simply be equated to'.'Equated' was not strictly mathematically correct, rather proportional with a O(1) constant of proportionality'?]**

In Methods, we provide a derivation of this quantity. In the case of low-error coding, which interests us, we obtain the dependence of the probability of a decoding error as a function of the various model parameters, as

$$P_{\text{error}} \approx \frac{L}{\sqrt{2\pi N}} \exp\left(-\log\left(1+\frac{R}{2\eta^2}\right)\frac{N}{2}\right). \tag{5}$$

The main dependence to note, here, is the exponentially strong suppression as a function of the number of neurons in the second layer (Fig. 2B). By contrast, the probability of error scales merely linearly with the size of the stimulus space, L, as is expected in the low-error limit. This result implies that it is possible to compress information highly efficiently in a comparatively small representation layer ($N \ll L$) *even though* the code is completely random. The price to pay for the use of randomness is that any error is likely 'catastrophic' (on the order of the input range), but these large errors happen prohibitively rarely. It is also worth noting that the rate of exponential suppression depends on the variance of the noise, η^2, or, more precisely, on the single-neuron signal-to-noise ratio, R/η^2 (where R is the variance of the signal, Eq. (3)). In numerical simulations, we set $R = 1$ and we vary η^2 to explore different noise regimes. Interestingly, even when this signal-to-noise ratio becomes small, i.e., when the noise in the activity of individual neurons is comparable to modulations of their mean responses, the exponential suppression of the probability of error remains valid, with a rate approximately equal to $R/4\eta^2$.

Compressed coding with broad tuning curves: trade-off between local and global errors

As we saw in the previous section, in the case of infinitely narrow tuning curves the coding of a stimulus in a given trial is either perfect or indeterminate; that is, any error is typically a global error, on the order of the entire stimulus range. In the more general case of sensory neurons with arbitrary tuning width, the picture is more complicated: in addition to *global* errors which result from the twisting and turning of the mean response curve, the population code is also susceptible to *local* errors (Fig. 3A). This is because broad tuning curves in the sensory layer partly preserve distances: locally, nearby stimuli are associated with nearby points on the mean response curve; as a result, the coding of any given stimulus is susceptible to local errors due to the response noise. As the tuning width in the sensory layer, σ, decreases, two changes occur in the mean response curve: it becomes longer (it 'stretches out') and it becomes more windy (Fig. 1C). Stretching increases the local resolution of the code (because it allows for two nearby stimuli to be mapped to two more distant points in the space of population activity), while windiness increases the probability of global errors. This trade-off is apparent when we plot the histogram of error magnitudes as a function of σ: for larger values of σ, global errors are less frequent, but local errors are boosted (Fig. 3B). Also noticeable, here, is that the large-error tails of the histograms are flat, consistent with the observation that global errors of all sizes are equiprobable. (Strictly speaking, this happens if the stimulus has periodic boundary conditions, such that, picking two random points, the probability that they are at a given distance does not depend on the location of one or the other point.)

For a more quantitative understanding, we carried out an approximate analytical calculation, in which (*i*) we approximated the mean response curve by a linear function locally and (*ii*) we considered that the distance between two segments of the curve containing the mean response to two stimuli distant by more than σ is sampled randomly. Using these two assumptions, we obtained the MSE as a sum of two terms (see Methods

for mathematical details), as

$$\varepsilon^2 = \left\langle E^2 \right\rangle_W \approx \left\langle E_l^2 \right\rangle_W + \left\langle E_g^2 \right\rangle_W \approx \frac{2\sigma^2\eta^2}{RN} + \frac{1}{\sigma\sqrt{2\pi N}}\bar{\varepsilon}_g \exp\left(-\log\left(1+\frac{R}{2\eta^2}\right)\frac{N}{2}\right), \tag{6}$$

where $\bar{\varepsilon}_g$ is a term of $\mathcal{O}(1)$ that depends upon the choice of stimulus boundary conditions (see Methods). This expression quantifies the MSE for a 'typical' network, obtained by averaging over possible choices of synaptic weights, as indicated by the brackets $\langle\cdot\rangle_W$. The first term on the right-hand-side of Eq. (6) represents the contribution of local errors, while the second term corresponds to global errors (Fig. 3C). Their form can be intuited as follows. The variance of local errors is proportional to σ^2 and inversely proportional to N, as in classical models of population coding with neurons with bell-shaped tuning curves (see, e.g., Zhang & Sejnowski (1999)). Furthermore, decreasing σ stretches out the mean response curve, which increases the local resolution of the code and explains the factor σ^2 in Eq. (6). (The form of this first term can also be understood as the inverse of the Fisher information (Seung & Sompolinsky, 1993; Brunel & Nadal, 1998), which bounds the variance of an unbiased stimulus estimator.) The second term on the right-hand-side of Eq. (6) is obtained as an extension of Eq. (5): instead of considering the probability that two mean response points are placed nearby, we consider the probability that two segments of the mean response curve with size σ each fall nearby. There are $1/\sigma$ such segments (since we have set the stimulus range to unity), and this explains why the factor L in Eq. (5) is replaced by a factor $1/\sigma$ in Eq. (6). Importantly, the two terms in Eq. (6) are modulated differently by the two parameters N and σ. Depending upon their values, either local or global errors dominate (Fig. 3C).

We tested the validity of Eq. (6): it agrees closely with results from numerical simulations, in which we computed the MSE using a Monte Carlo method and a network implementation of the ideal decoder (Fig. 3D see Methods for details). **[SIMONE: Clarification of discrepancy between MSE (text) and RMSE(plots): MSE when talking about error in general, while RMSE only when talking about a specific plot.]**(Henceforth we will refer to the MSE for the analytical computations, but we often plot its square root, the RMSE, such as to allow a direct comparison with the stimulus range. The illustrated quantity is always specified in the figure caption.) The non-trivial dependence is illustrated by the observation that the MSE may decrease or increase as a function of σ, around a given value of σ, depending upon the value of N (Fig. 3E). Furthermore, the strong (exponential) reduction in MSE with increasing N occurs only up to a crossover value that depends on σ (Fig. 3F); beyond this value, global errors disappear, and the error suppression is shallower (hyperbolic in N, due to improved local resolution). For small values of σ, the crossover values of N are larger and occur at lower values of the MSE.

As is apparent from Figs. 3D and E, for any value of N there exists a specific value of $\sigma = \sigma^*(N)$ that balances the two contributions to the MSE such as to minimize it. This optimal width can be thought as the one that stretches out the mean response curve as much as possible to increase local accuracy but that stops short of inducing too many catastrophic errors. The MSE is asymmetric about the optimal width, σ^*: smaller values of σ cause a rapid increase of the error due to an increased probability of global errors, while larger values of σ mainly harm the code's local accuracy, resulting in a milder effect. From Eq. (6), we obtain the dependence of the optimal width upon the population size, as

$$\sigma^* \approx \left(\frac{\bar{\varepsilon}_g}{4\eta^2}\sqrt{\frac{N}{2\pi}}\right)^{1/3} \exp\left(-\log\left(1+\frac{R}{2\eta^2}\right)\frac{N}{6}\right), \tag{7}$$

and the optimal MSE as a function of N, as

$$\varepsilon^{2*} = \langle E^2(\sigma^*)\rangle_W \approx \left(\frac{\eta\bar{\varepsilon}_g}{\sqrt{2\pi N}}\right)^{2/3} \exp\left(-\log\left(1+\frac{R}{2\eta^2}\right)\frac{N}{3}\right). \tag{8}$$

Both these analytical results agree closely with numerical simulations (Figs. 4A and B). Equation (8) and Fig. 4B show that the optimal MSE is suppressed exponentially with the number of representation neurons in the second layer. Thus, highly efficient compression of information and exponentially strong coding also occurs when tuning curves in the sensory layer are *not* infinitely narrow. The rate of this scaling depends upon the noise variance, η^2; in Figs. 4C and D, we illustrate the dependence of σ^* and ε^* upon N and η^2.

Compressed coding of multi-dimensional stimuli

Real-world stimuli are multi-dimensional. Our model can be extended to the case of stimuli of dimensions higher than one, but particular attention should be given to the nature of encoding in the first layer—because sensory neurons can be sensitive to one or several dimensions of the stimulus. In one limiting case, a sensory

neuron is sensitive to all dimensions of the stimulus; for example, place cells respond as a function of the two- or three-dimensional spatial location. Visual cells constitute another example of multi-dimensional sensitivity, as they respond to several features of the visual world; for example, retinal direction-selective cells are sensitive to the direction of motion, but also to speed and contrast. In the other limiting case, sensory neurons are tuned to a single stimulus dimension, and insensitive to others. We will refer to these two coding schemes as *pure* and *conjunctive*, following Ref. Finkelstein et al. (2018) where they are examined in the context of head-direction neurons in bats. The authors conclude that the relative advantage of a pure coding scheme—with neurons that encode a single head-direction angle—with respect to a conjunctive coding scheme—with neurons that encode two head-direction angles—depends on specific contingencies, such as the population size or the decoding time window. Indeed, in a conjunctive coding scheme individual neurons carry more information, but the population as a whole needs to include sufficiently many neurons to cover the (multi-dimensional) stimulus space—a constraint which becomes more restrictive as the number of dimensions increases.

We generalized our model to include the possibility of K-dimensional stimuli. For the sake of simplicity, we consider here only the two limiting cases of *pure* and *conjunctive* coding in the *sensory* layer of our model (i.e., we do not discuss intermediate cases, in which a given sensory neuron is sensitive to several but not all stimulus dimensions, see Methods). In the model, furthermore, neurons in the *representation* layer receive random inputs from *all* sensory neurons; as such, the representation layer always embodies a conjunctive coding scheme.

By extending the geometric picture (illustrated in Fig. 1 for the case of a one-dimensional stimulus), we can analyze differences in coding properties between pure and conjunctive coding schemes; in Fig. 5A, we illustrate the case of a two-dimensional stimulus. In this case, the mean response of representation neurons corresponds to a mapping from a two-dimensional stimulus space to a random 'sheet' (a two-dimensional surface) in the N-dimensional space of the population activity. In the *pure case*, the activity of a given sensory neuron is maximally modulated when the stimulus varies along a particular dimension, the one to which the neuron is sensitive. Variations of the stimulus along orthogonal directions have no effect on the mean neural activity. As a result, neurons in the representation layer, which compute a randomly weighted sum of the responses of sensory neurons, are only mildly responsive to variations of the stimulus along subsets of the stimulus dimensions. The resulting 'response sheet' embedded in N-dimensional space undergoes 'folds' with creases along these directions of mild sensitivity. By contrast, in the *conjunctive case* the activity of a sensory neuron is modulated by variations of the stimulus along any direction. As a result, the 'response sheet' that represents the joint mean activity of neurons in the second layer undergoes (random) curvature along all stimulus dimensions: rather than 'folded', it looks like a 'crumpled' sheet (Fig. 5A).

This geometric picture offers an intuitive explanation of the behavior of the mean decoding error in the two coding schemes. (For the corresponding mathematical treatment, see Methods.) The local error is determined by how much the 'response sheet' is stretched out; in turn, the more the response sheet is stretched out, the more it has to fold (or crumple) to fit in the allowed range of neural activity. Folding allows for more a modest stretching of the sheet than crumpling, and as a result the pure scheme incurs a larger local error than the conjunctive scheme (see Eqs. (53) and (57)). The global error is also different in the two coding schemes; there are two mechanisms at play, here. First, in the pure scheme, the folds of the surface imply that global errors occur between two stimuli that differ in a single dimension, whereas, in the conjunctive scheme, global errors occur between two stimuli that differ in an arbitrary number of dimensions. Second, the *total* variance of the tuning curve across the stimulus space is fixed (and, in particular, set to the same value for the pure and conjunctive schemes), but the signal-to-noise ratio which governs the rate of suppression with N scales differently as a function of K. **[SIMONE to RAVA: 'the *total* variance of the tuning curve across the stimulus space is fixed but it scales differently as a function of K'. Actually it is not the total variance which scales differently as a function of K, but rather the single-neuron signal-to-noise ratio governing the rate of scaling with N, as in one case is the SNR across a single dimension, while in the other case is the SNR across all stimulus space.]** Both mechanisms enhance the probability of global error in the pure scheme as compared to the conjunctive scheme (compare Eq. (54) and Eq. (58) in Methods). Intuitively, this is because a folded sheet has a larger surface area of contact with itself than a crumpled sheet.

We illustrate these conclusions with numerical results in the case of a three-dimensional stimulus ($K = 3$), relevant to the data analysis we present in the next section. In Fig. 5B, we illustrate the behavior of the RMSE as a function of N and σ for the pure and conjunctive coding schemes. In order to quantify the relative advantage of one scheme with respect to the other, we plot the ratio of the RMSE in the two schemes as a function of N and σ (Fig. 5C). The resulting, relatively intricate pattern, can be understood by considering different regimes. If the population size is small, the pure scheme slightly outperforms the conjunctive one; **[SIMONE : R:"Naively, this is counter-intuitive as we expect that the pure case is bad if we have few neurons. Also, if global errors dominate, again the pure case should be worse. So do**

you have a simple explanation for this statement?" S:"For simplicity, we compare just the global errors, as they are dominant. As N is small, the suppression is weak, so the advantage of conj scheme in the rate of scaling is minimal. Rather, it is dominant the influence of the prefactors, i.e., the number of uncorrelated regions. These are more in the conj scheme $(1/\sigma)^K$ rather then K/σ (approximately) of the pure case, which makes the pure case slightly better. I avoided to explain this because 1) One should have the formulas in front 2) The errors are large Propose: add '(due to a difference in the prefactors of the error probability, see Methods) "] in this regime, global errors dominate and coding is poor overall. At larger values of N, the contribution of local errors becomes non-negligible. If local errors dominate (which occurs for large N and sufficiently large σ), then the conjunctive scheme outperforms the pure one, and the ratio of the RMSEs approaches the theoretical prediction $(1/\sqrt{3})$. In the non-trivial regime in which local and global errors are balanced (for large N and intermediate values of σ), the advantage of the conjunctive scheme is further boosted. As explained above, this is due to a stronger suppression of global errors as a function of N in the conjunctive case. Finally, if σ becomes smaller than a crossover value that depends on the number of sensory neurons, the latter no longer cover the stimulus space sufficiently densely, and the conjunctive scheme breaks down; in this regime, thus, the pure scheme is favored.

As illustrated in Fig. 5B, similar to the one-dimensional case there exists in each of the two coding schemes an optimal value of the tuning curve width, σ, which achieves a balance between local and global errors, and it decreases with N. This dependence is somewhat different in the two coding schemes (Fig. 5D), and contributes to the form of the suppression of the RMSE in the two schemes (Fig. 5E). Both quantities, the optimal tuning curve width and the RMSE, decrease more rapidly as a function of N in the conjunctive scheme. This results from the fact that global errors are suppressed more strongly with N in the conjunctive case (as explained above), and therefore a smaller σ, yielding a lower local error, is preferable. At the same time, the requirement that sensory neurons cover the stimulus space yields a more stringent constraint on σ in the conjunctive scheme, yielding a bound on the extent of the regime of exponential error suppression.

[SIMONE to RAVA: On the MSE-RMSE. While talking about analytical computation, we always use MSE, but in plots we show RMSE (allowing for a better comparison with stimulus range). I added a sentence when we first plot the RMSE while talking about the MSE. I would stick to the use of MSE when talking generically about error and about math, while using RMSE while talking explicitly about specific plots, like in this section.]

Compressed coding in monkey motor cortex

The activity of neurons in the primary motor cortex (M1) of monkey is correlated to the location and movement of the limbs. Here, we consider spatial tuning in the context of a 'static task' (Kettner et al., 1988). In this task, the monkey is trained to keep its hand motionless during a given delay after having placed it at one of a set of preselected positions on a three-dimensional grid labeled by the vector $\mathbf{x} = (x_1, x_2, x_3)$. Tuning curves of hand-position selectivity can be extracted from recordings of M1 (Kettner et al., 1988; Wang et al., 2007), and it has been customary to model these as a linear projection of the hand position on a so-called 'preferred vector' or 'positional gradient', varying linearly with a combination of the spatial coordinates of the hand, $\mathbf{p}$, which thus points in the direction of maximal sensitivity(Wang et al., 2007). The tuning curve of neuron i is then written as

$$v_i(\mathbf{x}) = a_i + p_{1,i}x_1 + p_{2,i}x_2 + p_{3,i}x_3 = a_i + \mathbf{p}_i \cdot \mathbf{x}. \tag{9}$$

A recent study (Lalazar et al., 2016) noted, however, that a model of tuning curves that includes a form of irregularity yields an appreciably superior fit to the simple linear behavior of Eq. (9). This more elaborate model (Lalazar et al., 2016) bears similarity with our model of irregular tuning curves, and this naturally led us to ask about potential coding advantages that a complex coding scheme may afford in M1.

To be more specific, one can interpret the first layer in our network featured with neurons with three-dimensional Gaussian tuning curves, as representing neurons in the parietal reach area (or premotor area), which are known to display spatially localized tuning properties (Andersen et al., 1985). This population of neurons projects to a smaller population of M1 neurons which display spatially extended and irregular tuning profiles. In fitting our model to recordings from M1 neurons (Lalazar et al., 2016), we considered the arrangement of stimuli used in the experiment, namely 27 spatial locations arranged in a $3 \times 3 \times 3$ grid fitting in a 40 cm-high cube. We then followed a previous fitting method (Lalazar et al., 2016; Arakaki et al., 2019): given the diversity of the irregular tuning curves in the population we did not aim at fitting individual tuning curves; instead, we allowed for randomly distributed synaptic weights (as in our original model) and we fitted a single parameter, the width of the tuning curves in the first layer, σ. The fit was aimed at reproducing specific summary statistics of the data referred to as *complexity measure* (discrete version of the Lipschitz derivative that quantifies the degree of smoothness of a curve, see Methods and Lalazar et al. (2016)). The complexity measure varies from

neuron to neuron, and we chose σ so as to minimize the Kolmogorov-Smirnov distance between the distribution implied by our model and the one extracted from the data. While our model is somewhat simpler than a model of irregular M1 tuning curves employed previously (Lalazar et al., 2016), it yields comparable fits. In addition to fitting the population of tuning curves, we extracted from the data a quantification of the noise in the response of individual neurons. For each recorded neuron, we computed the variance of the signal as the variance, across different stimuli, of the mean firing rate (left hand side of Eq. (3)). Then, we estimated the variance of the noise by averaging the trial-to-trial variability of responses to the same stimulus. These two quantities allowed us to define a signal-to-noise ratio for each neuron of the population (see Eq. (62) in Methods). As in simulations we fixed the variance of the signal for each neuron to a constant value, we modeled the heterogeneity of the signal-to-noise ratio as a heterogeneous noise variance.

With a neural response model in hand, we can evaluate the coding performance; to do so, we consider a finer, $21 \times 21 \times 21$ grid of spatial locations as our test stimuli. We quantify the merit of a compressed code making use of irregular tuning curves by computing the MSE, $\varepsilon^2_{\text{irr}}$, and comparing the latter with the corresponding quantity in a coding scheme with the smooth tuning curves defined in Eq. (9), $\varepsilon^2_{\text{lin}}$. We plot our results in terms of the 'mean percent improvement', $\Delta\varepsilon \equiv (\varepsilon_{\text{lin}} - \varepsilon_{\text{irr}})/\varepsilon_{\text{lin}}$. $\Delta\varepsilon$ is positive when irregularities favor coding, and is at most equal to unity (in the extreme case in which irregularities allow for error-free coding).

We explore the performance of the two coding schemes for different values of the parameters N and σ, first in an ideal case in which all neurons have the same noise variance (Fig. 6A). We note the existence of a crossover value of N, N^*. When $N < N^*$, small values of σ induce prohibitively frequent global errors in the compressed (irregular) coding scheme, and linear (smooth) tuning curves are more efficient. For $N > N^*$, however, irregularities are always advantageous, and the more so the smaller the value of σ. Because global errors are suppressed exponentially with N, N^* typically takes a moderate value which depends on the magnitude of the noise; the larger the noise, the larger N^*. Figure 6B illustrates this noise-dependent behavior of the crossover population size, for the best-fit value of σ (≈ 23).

Next, for a more realistic modeling of M1 neurons, we analyzed the performance of a model in which each neuron's noise variance is extracted from the data (Figs. 6C and D). Noise variances in the population are heterogeneous, with a fraction of neurons exhibiting low signal-to-noise ratios (Fig. 6C, inset). For each value of N, we sampled eight different pools of N neurons from the population, and we averaged the corresponding mean percent improvement, $\Delta\varepsilon$. We found, again, that the relative merit of compressed coding (with irregular tuning curves) grows with the population size; interestingly, when compressed coding becomes advantageous ($\Delta\varepsilon > 0$ in Fig. 6C), the MSE is still appreciable (Fig. 6D). This means that even though local and global errors are balanced, both occur with non-negligible likelihood. $\Delta\varepsilon$ continues to grow with N until global errors are suppressed; beyond this second crossover value, N_{local}, $\Delta\varepsilon$ saturates because in both coding schemes (with irregular and linear tuning curves) local errors dominate. Correspondingly, the MSE scales differently for N above or below N_{local}. When $N < N_{\text{local}}$ the MSE decreases exponentially with N, due to the suppression of global errors, while when $N > N_{\text{local}}$, the suppression of the MSE is hyperbolic in N, reflecting the behavior of local errors only (Fig. 6D). This second crossover occurs at $N_{\text{local}} \approx 100$, a figure comparable to the number of neurons that control individual muscles in this specific task, as estimated from decoding EMG signals from individual muscles from subsets of M1 neurons (Lalazar et al., 2016).

Dimensionality of a compressed neural code

We introduced the geometrical interpretation of a neural code as a map between a set of stimuli and an ensemble of points in the space of mean population activity. In the case of a continuous K-dimensional stimulus space and smooth tuning curves, the code produces a K-dimensional surface embedded in the N-dimensional space of neural activities, which is often referred to as 'neural response manifold'(Seung & Lee, 2000; Gallego et al., 2017), implicitly assuming a local homeomorphism to a Euclidean space. In the previous sections we analyzed how the geometrical properties of the response manifold affects the coding performance of the population. In this section we aim to enrich this picture by characterizing quantitatively the 'dimensionality' of the manifold we considered, and to give another interpretation, through this measure, of the results of the previous sections. We will focus, for the sake of simplicity, on the one-dimensional case.

The complex tuning curves of the representation neurons, serving as coordinates of the resulting manifold, are samples from a Gaussian process (see Methods); therefore, the manifolds we obtain belong to the class of Gaussian manifolds (Lahiri et al., 2016). As the stimulus is one-dimensional, the manifold of mean population activity occupies a one-dimensional subspace (i.e., it is a curve); nevertheless, we showed in Fig. 1C that its 'complexity' changes as a function of σ. In order to quantify this complexity, we start by measuring how the neural responses are 'spread' in the N-dimensional space through the spectrum of the eigenvalues of the covariance matrix of the neural responses, corresponding to the variance carried by the principal components

in PCA (Fig. 7A). The resulting spectrum exhibits a band-pass structure, as it is flat up to a cut-off value, which increases by decreasing σ, and then falls quickly to 0. This implies that neural responses are equally spread across a number of principal axes, occupying a lower dimensional subspace of the N-dimensional space of neural activities; by decreasing σ, we increase the number of these principal axes, up to the limit of $\sigma \to 0$, when the responses spread across all the N dimensions, and the spectrum is completely flat.

Following Gao et al. (2017), we can extract a measure of the 'complexity' of these manifolds based on the spectrum of the covariance matrix, by defining the *intrinsic dimensionality* as the eigenvalues' participation ratio: $d_i = (\sum_{i=1}^{N} \mu_i)^2 / \sum_{i=1}^{N} \mu_i^2$. This measure yields indeed a value close to N in the case of $\sigma \to 0$, denoting an infinitely tangled manifold, and it is inversely proportional to σ, leading to values close to 1 in the case of broad tuning curves (Fig. 7B). With this definition in our hands, it is natural to investigate the geometrical properties of the manifold produced by our coding scheme at the optimal value of σ. In Fig. 7C we computed the fraction of dimensions (i.e., the intrinsic dimensionality divided by the number of neurons) occupied by the optimal neural manifold, for a fixed noise level, as a function of the population size. When N is low, the manifold is forced to lie on a low-dimensional subspace in order to avoid global errors. As N increases, the manifold can occupy a larger fraction of the available dimensions, increasing its length and consequently the local accuracy, but still yielding a low number of global errors. This analysis offers an example, through a simple mechanism, of how the optimal dimensionality of a neural code is a quantitative question, which depends upon the population size and the variance of the noise.

Compressed coding with noisy sensory neurons

Until now, we have considered the presence of response noise only in second-layer neurons. In this case, as long as sensory neurons are tiling the stimulus space (i.e., unless there are regions in stimulus space in which sensory neurons are unresponsive), stimuli are encoded with perfect accuracy in the activity of the first layer, and the MSE inferred from activity in the second layer can be made arbitrarily small for sufficiently large N. If sensory neurons are also noisy, then they represent stimuli only up to some degree of precision. Furthermore, because of the (dense) projections from the first to the second layer of neurons, independent noise in sensory neurons induces correlated noise in representation neurons. If the independent noise in sensory neurons is Gaussian with variance equal to ξ^2, then the covariance of the noise in the second layer becomes $\mathbf{\Sigma} = \eta^2 \mathbf{I} + \xi^2 \mathbf{W}\mathbf{W}^\mathbf{T}$. Thus, sensory noise affects the nature of the 'representation noise', and it is natural to ask how this changes the population coding properties.

As we shall show, in the compression regime ($N \ll L$) on which we focus, the kind of correlations generated by noise in the sensory layer have a negligible effect on the coding performance. Obviously, the introduction of sensory noise degrades coding, so the comparison of the noisy and noiseless systems is not very telling. Instead, we compare population coding in the presence of the full noise covariance matrix, $\mathbf{\Sigma}$, and in the presence of a diagonal noise covariance matrix with matched diagonal elements. Since synaptic weights are realizations of a Gaussian random variable, the matrix $\mathbf{W}\mathbf{W}^\mathbf{T}$ follows a Wishart distribution with mean the identity matrix (see Methods); therefore, the average variance-matched diagonal covariance matrix is written as $\mathbf{\Sigma}_{\text{ind}} = \left(\eta^2 + \xi^2\right) \mathbf{I}$. **[SIMONE: Simulations with W-dependent diagonal and average diagonal noise leads to the same results. The variance of the diagonal elements is 1/L, multiplied by the noise we considered (less than 1), leads to a very weakly anisotropic Gaussian noise, which can be fairly assumed to be isotropic. Also, I would use this comparison because it facilitates the analytic treatment and leads to homogeneous noise variance, which is the regime we treated previously.]** (The correct comparison is between the full covariance matrix and its diagonal counterpart, leading to anisotropic variance which depends on the specific realization of the random weights. As fluctuations of $\mathbf{W}\mathbf{W}^\mathbf{T}$ around the identity are of order $1/L$, simulations with the average diagonal matrix leads to the same results in the regime of noise we considered.) The latter is equivalent to a network with noiseless sensory neurons, but enhanced independent noise in representation neurons, with variance $\tilde{\eta}^2 \equiv \eta^2 + \xi^2$. In numerical studies, we observe, first, that the MSE depends only weakly on the noise correlations, as a function of σ. This behavior obtains because noise correlations primarily affect local errors, not global errors. (As noise correlations reduce the noise entropy—they 'shrink the cloud of possible noisy responses'—with respect to the independent case, one expects that correlations reduce the probability of occurrence of global errors. Numerical simulations however indicate that this effect is quantitatively negligible.)

The picture is different in the case of local errors, which can be either suppressed or enhanced by correlated noise (da Silveira & Rieke, 2021). We can show analytically that, here, local errors are enhanced; from a perturbative expansion of the inverse covariance matrix (see Methods for details), we obtained the local contributions to the MSE as

$$\varepsilon_l^2 = \varepsilon_{l,\text{ind}}^2 \left(1 + \frac{N\xi^2}{L\tilde{\eta}^2} - \frac{N\xi^4}{L\tilde{\eta}^4} + \dots\right) \tag{10}$$

in orders of $\xi^2/\tilde{\eta}^2$, where $\varepsilon^2_{l,\mathrm{ind}}$ is the corresponding quantity calculated for the variance-matched covariance matrix $\mathbf{\Sigma}_{\mathrm{ind}}$ rather than the full covariance matrix $\mathbf{\Sigma}$.

From Eq. (10), it appears that the effect of noise correlations on the MSE is deleterious but scales only weakly with $N/L \ll 1$. We checked this behavior numerically (Fig. 8A), and found a good match with the analytical result. We also compared the impact of different values of ξ^2, while keeping the effective noise variance, $\tilde{\eta}^2$, fixed (i.e., varying the relative contribution of input noise and output noise). Both Eq. (10) and Fig. 8B indicate that there exists a regime in which increasing ξ^2 in fact mitigates the deleterious effect of the correlated noise (this is seen in Eq. (10) as a partial cancellation of the second- and fourth-order terms).

Finally, we ask whether the impact of the noise correlation results specifically from the form with which sensory noise invests it. To answer this question, we examine a network with noiseless sensory neurons, but in which representation neurons exhibit correlated Gaussian noise, with a covariance matrix that has the same statistics as those of $\mathbf{\Sigma}$, but in which the form of correlations is not inherited from the network structure through the synaptic matrix $\mathbf{W}$; specifically, we consider a random covariance matrix, $\mathbf{\Sigma}_{\mathrm{rand}} = \eta^2\mathbf{I} + \xi^2\mathbf{X}\mathbf{X}^{\mathbf{T}}$, where $X_{ij} \sim \mathcal{N}(0, 1/L)$. In this case, noise correlations *suppress* the MSE as compared to the independent case (with $\mathbf{\Sigma}_{\mathrm{ind}}$), because the 'cloud of possible noisy responses' is reoriented randomly with respect to the curve of mean responses. Analytically, the analog of Eq. (10) for the case of a covariance matrix σ_{rand} (instead of σ) is similar, but skips the lowest-order, deleterious term:

$$\varepsilon^2_{l,\mathrm{rand}} \approx \varepsilon^2_{l,\mathrm{ind}} \left(1 - \frac{N\xi^4}{L\tilde{\eta}^4}\right). \tag{11}$$

This result, as well as numerical simulations (Fig. 8B), demonstrate that generically coding is improved by random noise correlations, and that this improvement increases with N and with the relative contribution of ξ^2. **[SIMONE: 'increases with ξ^2' .Stated like this, it seems that increasing input noise improves coding, but it also increases $\tilde{\eta}$. I would add ' with the relative contribution of'.]** In sum, noise correlations in representation neurons are deleterious if they are inherited from noise in sensory neurons—yet, the effect is quantitatively modest.

3 Discussion

Summary. We analyzed the coding properties of neural populations beyond classic models of tuning curves, by considering irregular response profiles resulting from random feedforward connectivity. Our model can interpolate between an irregular coding scheme, highly efficient but prone to catastrophic errors, and a smooth one, more robust in the face of noise. Optimality is achieved at an intermediate level of irregularity, which depends on the population size and on the variance of the noise. In the optimal code, the mean error is suppressed exponentially with population size. As a result, irregular neural codes allow for a strong compression of stimulus information from a large, first layer of neurons to a small, second layer. We extended these results to the case of multi-dimensional stimuli, more intricate because sensory neurons can exhibit various degrees of mixed selectivity; we considered in particular a pure coding scheme, in which sensory neurons are sensitive to a single stimulus dimension, and a conjunctive coding scheme, in which sensory neurons are sensitive to all stimulus dimensions. We examined the relative advantage of one scheme with respect to the other, a question explored recently elsewhere also (Finkelstein et al., 2018; Harel & Meir, 2020), and elucidated its dependence on the number of representation neurons and on the tuning parameters. These analyses enabled us to revisit data from M1 neurons in monkey (Lalazar et al., 2016) and to discuss the benefits of an irregular code in the context of the representation of hand position. Finally, we broadened the picture of compressed coding by considering input noise, in addition to output noise, and by analyzing the dimensionality of population activity in the case of an optimal code. This dimensionality is larger than the dimensionality of the stimulus, but smaller than the population size, so that it allows for an efficient use of the space of population activity while avoiding the proliferation of global errors.

'Exponentially strong' neural population codes. Our results on the exponential scaling of the mean error with the population size are similar to results obtained in the context of the representation of position by grid cells (Fiete et al., 2008; Sreenivasan & Fiete, 2011; Mathis et al., 2012; Wei et al., 2015). According to the terminology adopted in this literature the random compressed coding presented here is an 'exponentially strong' population code. Grid cell-tuning is a particular instance of exponentially strong codes making use of periodicity; the model presented here offers another example, in which tuning curves are random. In fact, the notion of an exponentially strong code predates work in computational neuroscience, as Shannon, already in 1949 introduced it in the context of a communication system for a continuous quantity (Shannon, 1949). In his framework, a sender maps a 'message' (a continuously varying quantity analogous to our stimulus) into a

'signal' (a higher-dimensional continuous quantity analogous to the output of our representation layer) which is then decoded by a receiver. The specific illustration he provides is that of a one-dimensional message mapped into a higher-dimensional signal (Fig. 4 in Ref. Shannon (1949)), analogous to the mapping illustrated in Fig. 1C; this mapping corresponds to a curve that wraps around in a higher-dimensional space. Shannon argues that an efficient code is obtained by stretching this curve to make it as long as possible up to the point at which the winding and twisting causes the curve to pass too close to itself, thereby generating catastrophic errors.

Yet Shannon went further, and showed that such a code need not to be carefully designed. His calculation corresponds, in our framework, to the case of infinitely narrow tuning curves in the sensory layer (Fig. 2): he demonstrated that, in this scenario, it is possible to send a set of discrete messages, with an error that is suppressed exponentially in the dimensionality of the signal. Our work proposes an extension of this 'fully random' scenario: by varying the width of tuning curves in sensory neurons, σ, one can modulate the smoothness of the mapping and trade off global errors with local errors. In this more general, 'correlated random' scenario, it is optimal to choose a non-vanishing value of σ which depends on the population size and other model parameters.

Coding with complex tuning curves. A large body of literature has addressed the problem of coding low-dimensional stimuli in populations of neurons with simple tuning curves. The most common assumption is that of bell-shaped tuning curves; these are often chosen to model sensory coding in peripheral neurons. Various studies set in this context discussed the shape of optimal tuning curves as a function of population size and stimulus dimensionality (Zhang & Sejnowski, 1999), stimulus geometry (Montemurro & Panzeri, 2006), and the time scale on which coding operates (Bethge et al., 2002; Yaeli & Meir, 2010). More recent work analyzed the influence of a (non-uniform) prior distribution of stimuli on the optimal arrangement and shapes of tuning curves across a population of neurons; a particular prediction is that the tuning-curve width is narrower for neurons with a preferred stimulus over-represented in the prior (Wei & Stocker, 2012; Ganguli & Simoncelli, 2014; Yerxa et al., 2020) . A separate direction of study focused on the effects of heterogeneity in the tuning-curve parameters on the coding performance (Wilke & Eurich, 2002; Shamir & Sompolinsky, 2006; Fiscella et al., 2015; Berry et al., 2019)

Our study falls in this line of work, but it presents two important differences: (i) we consider a family of irregular tuning curves (to be contrasted with simpler tuning curves, such as bell shaped or monotonic) and (ii) we consider downstream neurons rather than peripheral ones. To be more specific about point (i), we consider tuning curves resulting from a feedforward network with random synaptic weights. The assumption of random connectivity yields a 'benchmark model'; similar comparisons with benchmark random models have been used previously in examining information processing among layers of neural networks (Barak et al., 2013; Babadi & Sompolinsky, 2014; Litwin-Kumar et al., 2017). In our case, the irregularity of tuning curves makes the response of any single neuron highly ambiguous; the resulting code is thus distributed, and the neural population as a whole is viewed as the relevant unit of computation (Saxena & Cunningham, 2019).

Distributed codes have been argued to come with high capacity. An early example was developed in the context of face coding in the superior temporal sulcus of monkey (Abbott et al., 1996). Data analysis indicated that single-neuron sensitivity was heterogeneous and uninformative, but the number of distinguishable face stimuli grew exponentially with the population size. Our work provides an example of a random distributed code for continuous stimuli, which exhibits similar scaling properties. The main difference is that, in the case of continuous stimuli, the identity of a stimulus is ill-defined, what matters are the distances between pairs of stimuli. In other word, both the probability of occurrence of an error and its magnitude matter. The requirement of minimizing the mean squared error then yields a particular coding scheme that balances small (local) and large (global) errors.

Regarding point (ii), we recall that, to date, 'efficient coding' models of neural coding have addressed peripheral or 'receptor' neurons, i.e., neurons activated directly by a physical stimulus. Our work departs from this framework, in that it focuses on coding in a population of neurons downstream from receptor neurons. In this case, it is not possible to optimize the properties of one population without considering those of the other population. In particular, in our approach we optimize the coding scheme in receptor (sensory) neurons subject to constraints on the activity of downstream (representation) neurons. This way of thinking provides a different angle on the rationale for an optimal code.

Geometry and dimensionality of population responses. In the past decade, the progress in experimental methods has allowed for the recording of neural populations on a large scale (Cunningham & Yu, 2014; Saxena & Cunningham, 2019). In an effort to interpret the way in which information is represented in population activity, various approaches have been focusing on the geometric properties of ensembles of population responses to a battery of stimuli (Fusi et al., 2016; Gallego et al., 2017; Stringer et al., 2019; Kobak et al., 2019) . Points in a high-dimensional space, each corresponding to the neural population response to a stimulus, are often interpreted as being located on a manifold which describes the space of possible population activity.

Quantifying the geometry, and more specifically the dimensionality of this manifold, offers a characterization of neural population activity. This geometric element is eminently relevant in our work, too, where the distribution of coding errors depends directly on the geometry of the population activity in a representation layer, that results from the properties of neural responses in a sensory layer.

A specific geometrical question is that of the dimensionality of the population response in the representation layer. In Sec. 2, we showed that the spectrum of the covariance of the population activity in the representation layer, across the stimulus space, comes with a band-pass structure; by decreasing the width of tuning curves in the sensory layer, the band-pass profile acquires additional modes. Stringer et al. (2019) discussed a similar picture in analyzing recordings from a large population of visual neurons responding to a large, but discrete, set of images. In their case, the spectrum of the covariance matrix of population responses exhibits an algebraic (power-law) tail, and the authors argue that this property allows a high-dimensional population activity while retaining smoothness of the code. Our work presents a different, and more elementary, mechanism by which a large number of modes can be accommodated by the population activity (while retaining smoothness). The non-trivial point, in our case, is that it is not beneficial for coding to be poised in the limiting case in which the number of modes is maximal but the code becomes singular (non-smooth). The reason is that, in this limit, global errors proliferate. The optimal dimensionality of the response manifold lies at an intermediate value at which intersections of the manifold with itself are rare and local and global errors are balanced (Fig. 7).

Compressed sensing. We studied a network in which the information encoded in a high-dimensional activity pattern is compressed into the activity of a comparatively small number of neurons, a setting which exhibits analogies with the one of compressed sensing (Candes & Tao, 2006). Compressed Sensing is a signal-processing approach for reconstructing L-dimensional signals, which are K-sparse in some basis (i.e., they can be expressed as vectors with only K non-vanishing elements) from a N linear and noisy measurements of the original signals, with $K \ll L$ and $N \ll L$ (Donoho, 2006). In our study, the low dimensionality of the stimulus, x, implies sparsity of the L-dimensional activity of the sensory layer, as long as the tuning curves in the sensory are not too wide. **[**(but see Baraniuk & Wakin (2009) for more details) **SIMONE TO RAVA: What do you mean by 'Why does 'this' appear here'? The paper extends the result of CS to signals which are not explicitly sparse (at least, not with a linear change of basis), but rather lie on a low-dimensional manifold, analog to our case where there is a low-dimensional variable which generates the signal through a non linear process. Therefore some of the results of CS are still valid, in particular the fact that a logartihmic number of Random Projections preserve distances between points. That's why I would avoid the sentence 'as long as the tuning curves in the sensory are not too wide.', because yes, in that case the signal is not strictly sparse, but still this paper says that some results of CS are valid.(Actually, their result invoke a notion of 'curvature', so the comprssibility of the data increases as the width increases.)**

A central result in the field of compressed sensing is that random measurements achieve near-optimal results. Furthermore, for this to obtain, the number of measurements scales with K and only logarithmically with the dimensionality of the signal $N > \mathcal{O}\left(K \log\left(L/K\right)\right)$ (Candes & Tao, 2006; Baraniuk et al., 2008). In effect, in our network the representation layer operates a limited number of random measurements from the sensory layer. And we obtain an analog scaling form by inverting Eq. (5): the number of random projections, N, necessary to decode L different stimuli with negligible error scales logarithmically with the number of stimuli. We note, however, that our framework differs from that of compressed sensing as the objective is to decode the identity of the stimulus rather than a high-dimensional signal vector (in our case, the activity pattern of the sensory layer).

Encoding vs. decoding. In our study, we focused exclusively on the properties of encoding in a neural population. For this aim, throughout we assume an ideal decoder; in principle, this is not a limitation: we show in Methods that an ideal decoder can be implemented in a simple, two-layer neural network. The first layer computes a discretized approximation of the posterior distribution over stimuli, and the second layer computes the mean of this distribution, in such a way as to minimize the MSE. Furthermore, all the operations carried out by this two-layer network—linear filtering, non-linear transfer, and normalization—are plausible biological operations (Deneve et al., 1999; Kouh & Poggio, 2008; Carandini & Heeger, 2012). The parameters involved, however, have to be chosen with the knowledge of the tuning curves and noise model.

One can ask whether biologically plausible learning rules can result in a decoder than approximates closely the ideal one. A closely related questions has been examined by Bordelon et al. (2020), who analyzed how the generalization error in a deep neural network trained with gradient descent depends on the number of training samples and on the Fourier modes of the target function **[SIMONE to RAVA:Strictly speaking, it is not restricted to Fourier modes, as their theory generalize to any eigenfunction decomposition of a function belonging to a RKHS. But for the sake of simplicity, maybe talking about Fourier modes (which is a special case) is more immediate?]**. Bordelon et al. (2021) find that learning the high-frequency

Fourier components of the target function requires a larger number of training samples, as compared to learning the low-frequency components. Similarly, in the context of our network one expects that learning a decoder in the case of narrow tuning curves in the sensory layer is more laborious than in the case of broad tuning curves. Furthermore, noise in the training samples may hamper learning severely in the presence of global errors. Broadly, one can ask to what extent our results may be modified if the decoding is carried out by a decoder different from the ideal one, for example by a decoder obtained through adequately chosen learning rules. We leave the study of this question for future work.

4 Methods

Throughout the discussion, bold letters denote vectors $\mathbf{r} = \{r_1, r_2, ..., r_N\}$. $\|\mathbf{r}\|_2^2 = \sum_i r_i^2$ represents the L_2 norm. Capital bold letters $\mathbf{W}$ denote matrices. Numerical simulations and data analysis were done using a custom code written in Julia (Bezanson et al., 2017).

Model description: one-dimensional stimulus

Network definition and constraints. The first, sensory layer is made of L neurons, encoding a continuous scalar stimulus $x \in [0, 1]$, with Gaussian tuning curves. The firing rate of neuron j as a function of x is given by

$$u_j(x) = A \exp\left(-\frac{(x - c_j)^2}{2\sigma^2}\right), \qquad \text{(1 restated)}$$

where c_j is the preferred stimulus of neuron j, σ is the tuning width, and A is a gain which will be chosen accordingly. The preferred stimuli are evenly spaced, $c_j = j/L$. These neurons are all-to-all connected to the N representation neurons, with i.i.d Gaussian weights, $W_{ij} \sim \mathcal{N}(0, 1/L)$. The response of representation neuron i is therefore obtained as

$$v_i(x) = \sum_{j=1}^{L} W_{ij} u_j(x). \qquad \text{(2 restated)}$$

The gain A is chosen such to keep constant the variance of the responses across stimuli, averaged over different network realizations. This constraint reads

$$\begin{aligned} R &= \left\langle \int_0^1 dx \left[v_i(x) - \int_0^1 dx' v_i(x') \right]^2 \right\rangle_W \\ &= \left\langle \int_0^1 dx v_i(x)^2 - \left(\int_0^1 dx v_i(x) \right)^2 \right\rangle_W \\ &= \left\langle \sum_{j,j'} W_{ij} W_{ij'} \left(\int_0^1 dx u_j(x) u_{j'}(x) \right) - \sum_{j,j'} W_{ij} W_{ij'} \left(\int_0^1 dx u_j(x) \right) \left(\int_0^1 dx u_{j'}(x) \right) \right\rangle_W \\ &= \int_0^1 dx u_j(x)^2 - \left(\int_0^1 dx u_j(x) \right)^2, \end{aligned} \qquad (12)$$

where $\langle \cdot \rangle_W$ indicates the average over the distribution of synaptic weights. Since the synaptic weights are i.i.d. Gaussian, $\langle W_{ij} W_{ij'} \rangle_W = \frac{1}{L} \delta_{jj'}$. Here and in following calculations, we use the approximation for the Gaussian integral **[MIRKO: Express the result in terms of error functions and then use the approximation? S: I don't think is needed, as we will never use the exact expression.]**

$$\int_0^1 dx u_j(x) \approx \int_{-\infty}^{\infty} u_j(x) = A\sqrt{2\pi\sigma^2}, \qquad (13)$$

where the approximation is valid when c_j is far from stimulus boundaries and σ is small with respect to the stimulus range. Since we consider a large number of neurons in the first layer and relatively small σ (up to 1/10 of the stimulus range), the effects of this approximation in our results are negligible. By inserting Eq. (13) and a similar approximation for $\int_0^1 dx u_j(x)^2$ into Eq. (12), we obtain

$$A^2 = \frac{R}{\sqrt{\pi\sigma^2} - 2\pi\sigma^2}. \qquad (14)$$

Gaussian process interpretation. The response of each neuron of the second layer to a fixed stimulus x is a sum of Gaussian random variables. As a result, it is also a Gaussian random variable with mean

$$\langle v_i(x)\rangle_W = \sum_{j=1}^{L} \langle W_{ij}\rangle_W u_j(x) = 0. \tag{15}$$

The covariance between the response of the same neuron to two different stimuli, x and x', is given by

$$\langle v_i(x)v_i(x')\rangle_W = \sum_{j,j'} \langle W_{ij}W_{ij'}\rangle_W \, u_j(x)u_{j'}(x') = \sum_{j=1}^{L} \frac{1}{L} u_j(x)u_j(x'). \tag{16}$$

We can approximate the discrete sum with the integral $\sum_{j=1}^{L} f(c_j)\Delta c_j \approx \int_0^1 f(c_j)dc_j$, and the error of this approximation will be of order $1/L$. Finally, we obtain the covariance function

$$\begin{aligned}
\langle v_i(x)v_i(x')\rangle_W &\approx \int_0^1 dc_j u_j(x)u_j(x') \\
&= A^2 \int_0^1 dc_j \exp\left(-\frac{\left((x-c_j)^2 + (x'-c_j)^2\right)}{2\sigma^2}\right) \\
&\approx A^2\sqrt{\pi\sigma^2}\exp\left(-\frac{\Delta x^2}{4\sigma^2}\right),
\end{aligned} \tag{17}$$

where $\Delta x = x - x'$. In the last line we took the limit of integration going to infinity, similarly to Eq. (13); this approximation is valid if the arithmetic mean of x and x' is far from the stimulus boundaries. Equation (15) and Equation (17) show that each neuron tuning curve is a sample from a one-dimensional Gaussian process with 0 mean and squared exponential kernel with correlation length equal to $\sqrt{2}\sigma$ Rasmussen (2004).

Encoding - decoding

Noise Model. Representation neurons are affected by additive isotropic Gaussian noise. At each trial, the vector of responses to a given stimulus x is obtained as

$$\mathbf{r} = \mathbf{v}(x) + \mathbf{z}, \tag{18}$$

where $\mathbf{z}$ is a noise vector of independent Gaussian entries with a fixed variance, $z_i \sim \mathcal{N}(0, \eta^2)$. Here, $\mathbf{v}(x) = \{v_1(x), v_2(x), ..., v_N(x)\}$ is the vector containing the mean responses of second layer neurons to the same stimulus x, Eq. (2). As a result, we can write the likelihood of a response given a stimulus as

$$p\left(\mathbf{r}|x\right) = \frac{1}{\left(2\pi\eta^2\right)^{N/2}} \exp\left(-\frac{\|\mathbf{r}-\mathbf{v}(x)\|_2^2}{2\eta^2}\right). \tag{19}$$

This expression can be extended to keep into account a generic noise covariance matrix, $\mathbf{\Sigma}$, leading to

$$p\left(\mathbf{r}|x\right) = \frac{1}{\left(2\pi\right)^{N/2}\left(\det\left(\mathbf{\Sigma}\right)^{1/2}\right.} \exp\left(-\left(\mathbf{r}-\mathbf{v}(x)\right)^T \mathbf{\Sigma}^{-1}\left(\mathbf{r}-\mathbf{v}(x)\right)\right). \tag{20}$$

Loss function and decoder. [MIRKO: SHould we add the picture of the decoder? S: I have it, we have enough figures but it can be added in Methods figures.] We measured the coding performance of the neural population through the Mean Squared Error (MSE) in stimulus estimate Dayan & Abbott (2001). For a generic decoder, or estimator, $\hat{x} = f_{dec}(\mathbf{r})$, the MSE is defined as

$$E^2 = \int dx \int d\mathbf{r} p(\mathbf{r}|x)\left(\hat{x}-x\right)^2. \tag{21}$$

We considered this quantity averaged over network realizations, $\varepsilon^2 \equiv \langle E^2 \rangle_W$; we often plot the square root of this quantity, $\varepsilon \equiv \sqrt{\langle E^2 \rangle_W}$, since it has the same unit of measurement of the stimulus. For multidimensional stimuli, we average the squared norm, $\|\hat{\mathbf{x}} - \mathbf{x}\|_2^2$.

The estimator which minimizes the MSE (Minimum-MSE or MMSE) is given by the average of the posterior distribution. As we assume an uniform prior over stimuli, $p(x) \sim \mathcal{U}(0,1)$, we can write the estimator as a function of the likelihood, as

$$\hat{x}_{MMSE} = \int_0^1 dx p(x|\mathbf{r}) x = \frac{\int_0^1 dx p(\mathbf{r}|x) x}{\int_0^1 dx p(\mathbf{r}|x)}. \tag{22}$$

In order to numerically implement this estimator, we approximate the integrals with a discrete sum over M values and, by inserting Eq. (19), we obtain

$$\begin{aligned}\hat{x}_{MMSE} &\approx \frac{\sum_{m=1}^M x_m p\left(\mathbf{r}|x_m\right) \Delta x_m}{\sum_{m=1}^M p\left(\mathbf{r}|x_m\right) \Delta x_m} = \frac{\sum_{m=1}^M x_m \exp\left(-\frac{1}{2\eta^2}\left(\sum_{i=1}^N r_i^2 + \sum_{i=1}^N v_i^2(x_m) - 2\sum_{i=1}^N v_i(x_m) r_i\right)\right)}{\sum_{m=1}^M \exp\left(-\frac{1}{2\eta^2}\left(\sum_{i=1}^N r_i^2 + \sum_{i=1}^N v_i^2(x_m) - 2\sum_{i=1}^N v_i(x_m) r_i\right)\right)} \\ &= \frac{\sum_{m=1}^M x_m \exp\left(\frac{1}{2\eta^2}\left(\sum_{i=1}^N 2 v_i(x_m) r_i - \sum_{i=1}^N v_i^2(x_m)\right)\right)}{\sum_{m=1}^M \exp\left(\frac{1}{2\eta^2}\left(\sum_{i=1}^N 2 v_i(x_m) r_i - \sum_{i=1}^N v_i^2(x_m)\right)\right)} = \sum_{m=1}^M x_m \tilde{h}_m,\end{aligned} \tag{23}$$

where in the last step the terms $\sum_i r_i^2$ cancels as it is common to both numerator and denominator and we assumed a constant Δx_m.

This series of operations can be implemented in a two-layer neural network. A first layer of M neurons, whose activity is given by

$$\tilde{h}_m = \frac{\exp\left(\sum_{i=1}^N \lambda_{mi} r_i + b_m\right)}{\sum_{m'=1}^M \exp\left(\sum_{i=1}^N \lambda_{m'i} r_i + b_{m'}\right)}, \tag{24}$$

computes a normalized discrete approximation of the likelihood, $\tilde{h}_m \propto p(\mathbf{r}|x_m)$, such that $\sum_{m=1}^M \tilde{h}_m = 1$. The unnormalized activity of neuron m, $h_m = \exp\left(\sum_{i=1}^N \lambda_{mi} r_i + b_m\right)$, is a linear combination of the activity of the representation neurons plus a bias term, passed through an exponential non-linearity. The synaptic weights between the m-th decoder neuron and the i-th representation neuron, are a function of the true mean response of neuron i to stimulus x_m and the variance of the noise, $\lambda_{mi} = v_i(x_m)/\eta^2$. Similarly, the bias term is obtained as $b_m = -\sum_i v_i(x_m)^2/2\eta^2$. Finally, to obtain the MMSE estimate, a single neuron weights the activity of these M neurons according to their 'preferred stimulus', x_m.

In the following discussion, we will often use the Maximum a Posterior (MAP) estimator, which in this case corresponds to a Maximum Likelihood (ML) estimator, defined as

$$\hat{x}_{MAP} = \arg\max_{x_m} \tilde{h}_m = \arg\min_{x_m} \|\mathbf{r} - \mathbf{v}(x_m)\|_2^2, \tag{25}$$

since it has a simpler geometrical interpretation: it finds the stimulus which corresponds to the closest vector of mean responses to the noisy output. In numerical simulations, the MSE for these two estimators are very similar.

The same decoder can be extended to deal with the case of non-diagonal noise covariance matrix σ, plugging Eq. (20) into Eq. (22) . The decoding weights and biases are now correlated, $\lambda_m = \mathbf{v}^T(x_m)\sigma^{-1}$ and $b_m = \mathbf{v}^T(x_m)\sigma^{-1}\mathbf{v}(x_m)$, where λ_m denotes the m-th row of λ.

In numerical simulations, we computed the MSE with standard Monte Carlo method. We generated the noisy responses to sampled stimuli and we decoded them using the ideal decoder, updating the estimated MSE until convergence (i.e., until the estimate was within a tolerance of 10^{-8} in the last 500 steps, after a burn-in period of 4000 steps). For the sake of simplicity and to avoid any limitations to the decoder, we set the number of decoder neurons equal to the number of sensory neurons, $M = L$, with uniformly space preferred stimuli, $x_m = m/M$.

Errors computation

Narrow tuning curves. If $\sigma \to 0$, the first layer neurons respond only to their preferred stimulus. For this limit case, we consider that the stimulus can take only L discrete values, $x_j = j/L$. The responses of the second layer neurons are given by $v_i(x_j) = \tilde{A} W_{ij}$, with $\tilde{A}^2 = LR$ such to have $v_i(x_j) \sim \mathcal{N}(0, R)$. The constant of proportionality is computed with the analog of Eq. (3) for discrete stimuli, in the limit of large L.

We denote with $p_e(\mathbf{r}|x_j) = p(\mathbf{r}|x_j)\Theta\left(|\hat{x} - x_j|\right)$ the probability density function that the noise will produce an error in decoding the response associated to stimulus x_j. With a small abuse of notation, we define the Heaviside function as $\Theta(z) = 1$ if $z > 0$, and 0 otherwise. When we take the average over synaptic weights, the probability of having an error on a stimulus x_j is independent on the decoded stimulus $\hat{x}$. The average MSE, Eq. (21), can be therefore approximated as **[MIRKO (AND SIMONE): This passage is not easy to justify, although the final formula is easy to understand. My reasoning is: the probability of having an error on a given stimulus, and the relative error, are dependent, and linked by the synaptic matrix W. We make the approximation that the probability of having an error on a stimulus and the MSE are two independent r.v., given the realization of W. As a result, in computing the average of the product, we can simply compute the product of the average.]**

$$\begin{aligned} \langle E^2 \rangle_W = & \frac{1}{L} \sum_{j=1}^{L} \left\langle \int d\mathbf{r} p_e(\mathbf{r}|x_j) \left(\hat{x} - x_j\right)^2 \right\rangle_W \\ & \approx \langle P(E) \rangle_W \left\langle \frac{1}{L} \sum_{j=1}^{L} (\hat{x} - x_j)^2 \right\rangle_W , \end{aligned} \tag{26}$$

where $\langle P(E) \rangle_W = \left\langle \int d\mathbf{r} p_e(\mathbf{r}|x_j) \right\rangle_W$ is the average probability that, given a stimulus, the noise will cause an error in its estimate. Despite the notation, it does not depend on the specific value of x_j. This formula has an intuitive interpretation: the MSE is the mean probability of having an error on a stimulus multiplied by the average squared error. By noticing that, if there is an error, the decoder can output any of the others $L - 1$ stimuli, we obtain

$$\left\langle \frac{1}{L} \sum_{j=1}^{L} (\hat{x} - x_j)^2 \right\rangle_W = \frac{1}{L^2} \sum_{j=1}^{L} \sum_{j'=1, j' \neq j}^{L} \left(\frac{j'}{L} - \frac{j}{L} \right)^2 \approx \frac{1}{6}, \tag{27}$$

where the last approximation holds for large L. The thing to notice here is that this quantity is of order 1, the size of the stimulus range. The average probability of error is the probability that it exists one j' such that $\mathbf{r}$ is closer to $\mathbf{v}(x_{j'})$ than to $\mathbf{v}(x_j)$. We can express this probability as a function of the probability of the complementary event,

$$\langle P(E) \rangle_W = 1 - \left\langle P\left(\|\mathbf{r} - \mathbf{v}(x_{j'})\|_2^2 > \|\mathbf{r} - \mathbf{v}(x_j)\|_2^2 \quad \forall j \neq j' \right) \right\rangle_W . \tag{28}$$

Averaging over different realizations of the synaptic matrix, the probability of not having an error on $x'$are i.i.d for different j', and we can write

$$\begin{aligned} \langle P(E) \rangle_W &= 1 - \left(1 - \left\langle P\left(\|\mathbf{r} - \mathbf{v}(x_{j'})\|_2^2 < \|\mathbf{r} - \mathbf{v}(x_j)\|_2^2 \right) \right\rangle_W \right)^{L-1} \\ &\approx L \left\langle P\left(\|\mathbf{r} - \mathbf{v}(x_{j'})\|_2^2 < \|\mathbf{r} - \mathbf{v}(x_j)\|_2^2 \right) \right\rangle_W \\ &= L \left\langle P\left(\sum_{i=1}^{N} \left(v_i(x_j) - v_i(x_{j'})\right)^2 - \sum_{i=1}^{N} 2 \left(v_i(x_j) - v_i(x_{j'})\right) z_i < 0 \right) \right\rangle_W . \end{aligned} \tag{29}$$

In the first step, we assumed that the probability of having an error on a stimulus x_j is small, and $L - 1 \approx L$ is large, such that we can approximate $(1 - z)^L \approx 1 - Lz$. In the last step we simply inserted Eq. (18). The difference between the response of the same neuron to two different stimuli, averaged over different synaptic weights realizations, has a Gaussian distribution, $\tilde{v_i} \equiv v_i(x_j) - v_i(x_{j'}) = \tilde{A}(W_{ij} - W_{ij'}) \sim \mathcal{N}(0, 2R)$. By averaging over the noise distribution too, the probability of error reads

$$\langle P(E) \rangle_W \approx L \int \prod_{i=1}^{N} d\tilde{v}_i \prod_{i=1}^{N} dz_i p(\tilde{v}_i) p(z_i) \Theta\left(-\sum_{i=1}^{N} \tilde{v}_i^2 + 2 \sum_{i=1}^{N} \tilde{v}_i z_i \right) . \tag{30}$$

This is the probability that the random variable $\rho = \sum_{i=1}^{N} \tilde{v}_i^2 - \sum_{i=1}^{N} 2\tilde{v}_i z_i$ is less than 0, where $\tilde{v}_i \sim \mathcal{N}(0, 2R)$ and $z_i \sim \mathcal{N}(0, \eta^2)$. We can compute this quantity by noticing that, if we fix $\zeta = \sum_{i=}^{N} \tilde{v}_i^2$, the conditional distribution of ρ is Gaussian, $\rho|\{\tilde{v}_i^2\} \sim \mathcal{N}(\zeta, 4\zeta\eta^2)$. By using the definition of error function we can rewrite the error probability as

$$\begin{aligned}\langle P(E)\rangle_W &\approx L\int_0^\infty d\zeta p(\zeta)\int_{-\infty}^0 d\rho p(\rho|\zeta)\\ &= \frac{L}{2}\int_0^\infty d\zeta p(\zeta)\,\mathrm{erfc}\left(\sqrt{\frac{\zeta}{8\eta^2}}\right),\end{aligned} \tag{31}$$

where $p(\zeta) = \frac{(\zeta/2R)^{N/2-1}\exp(-\zeta/4R)}{2^{N/2+1}\Gamma(N/2)}$ is the probability density function of a Chi-squared distribution. Computing this integral, we obtain

$$\begin{aligned}\langle P(E)\rangle_W &\approx L\frac{(\frac{\eta^2}{2R})^{\frac{N}{2}}\Gamma(N)}{\Gamma(\frac{N}{2})}{}_2\tilde{F}_1\left(\frac{N}{2},\frac{1+N}{2},\frac{2+N}{2},-2\frac{\eta^2}{R}\right)\\ &= L\frac{\left(\frac{\eta^2}{2R}\right)^{\frac{N}{2}}\Gamma(N)}{\Gamma\left(\frac{N}{2}\right)\Gamma\left(\frac{2+N}{2}\right)}\sum_{n=0}^{\infty}\frac{\left(\frac{N}{2}\right)_n\left(\frac{N+1}{2}\right)_n}{\left(\frac{N+2}{2}\right)_n n!}\left(-2\frac{\eta^2}{R}\right)^n,\end{aligned} \tag{32}$$

where ${}_2\tilde{F}_1(a,b,c,x)$ is the regularized 2F1 Hypergeometric function and in the second line we substituted its definition. The Pochammer symbol is also defined through Gamma functions, $(x)_n = \frac{\Gamma(x+n)}{\Gamma(x)}$. Simplifying and using the identity $\sum_{n=0}^{\infty}\frac{(x)_n}{n!}a^n = (1-a)^{-x}$, we obtain the expression for the error probability which appears in the main text

$$\begin{aligned}\langle P(E)\rangle_W &\approx L(\frac{\eta^2}{2R})^{\frac{N}{2}}\frac{\Gamma(N)}{\Gamma^2(\frac{N}{2})\frac{N}{2}(1+2\eta^2/R)^{\frac{N+1}{2}}}\\ &\approx \frac{L}{\sqrt{2\pi N}}\exp\left(-\log\left(1+\frac{R}{2\eta^2}\right)\frac{N}{2}\right),\end{aligned} \tag{5 restated}$$

where in the last step we used the Stirling approximation for the Gamma functions.

Broad tuning curves. In this case we consider continuous stimuli, and the noise can also produce small scale errors; we therefore consider two contributions to the error: local and global. Since our system has a natural correlation length, Eq. (17), we define as global an error when the difference between the stimulus and its estimate is greater than σ, $|\hat{x}-x| > \sigma$. Despite this definition may seem bit tricky, as for very large σ all the errors will be local, the maximum value we considered for σ is still quite small, about 1/10 of the stimulus range. We rewrite the MSE as

$$\varepsilon^2 = \langle E^2\rangle_W = \langle E_l^2 + E_g^2\rangle_W = \left\langle\int dx d\mathbf{r} p_l(\mathbf{r}|x)(\hat{x}-x)^2\right\rangle_W + \left\langle\int dx d\mathbf{r} p_g(\mathbf{r}|x)(\hat{x}-x)^2\right\rangle_W, \tag{33}$$

where $p_{l/g}(\mathbf{r}|x) = p(\mathbf{r}|x)\Theta\big(\pm(\sigma - |\hat{x}-x|)\big)$ denotes the probability density function that, given x, the noise will cause a local/global error. It holds the following normalization $\int d\mathbf{r}\,(p_l(\mathbf{r}|x) + p_g(\mathbf{r}|x)) = 1$.

Local error. According to the ML decoder, Eq. (25), the stimulus estimate will correspond to the x' such that $\mathbf{v}(x')$ has the minimal distance from $\mathbf{r}$. If the error is local, this point corresponds to the projection of the noise vector onto the response curve. By expanding linearly the response curve around $\mathbf{v}(x)$, we obtain

$$\left\|\mathbf{z}\cdot\hat{\mathbf{v}}'(x)\right\|_2^2 \approx \|\mathbf{v}(x+\Delta x) - \mathbf{v}(x)\|_2^2 \approx \|\mathbf{v}'(x)\|_2^2\,\Delta x^2, \tag{34}$$

where $\mathbf{v}'(x) = \partial\mathbf{v}(x)/\partial x$ and $\hat{\mathbf{v}}'(x)$ is the normalized vector with the same direction. The error can be estimated as $\Delta x^2 = (\hat{x}-x)^2 = \frac{\|\mathbf{z}\cdot\hat{\mathbf{v}}'(x)\|_2^2}{\|\mathbf{v}'(x)\|_2^2}$. We will show that the probability of global error will be exponentially small in N, therefore we can approximate $p_l(\mathbf{r}|x)$ with the whole Gaussian, Eq. (19). When integrating over the isotropic Gaussian noise, the magnitude of the projection onto a fixed unit vector will be simply equal to the

the variance. We obtain therefore

$$\begin{aligned}\left\langle E_l^2\right\rangle_W &= \left\langle \int_0^1 dx \int d\mathbf{z} \frac{\left\| \mathbf{z}\cdot\hat{\mathbf{v}}'(x)\right\|_2^2}{\|\mathbf{v}'(x)\|_2^2}\right\rangle_W \\ &= \left\langle \int_0^1 dx \frac{\eta^2}{\|\mathbf{v}'(x)\|_2^2}\right\rangle_W \end{aligned} \tag{35}$$

We now approximate the average of the inverse with the inverse of the average of the derivative of the tuning curves, $\langle 1/\cdot\rangle_W \approx 1/\langle\cdot\rangle$, an approximation which is valid if L is large and σ is small with respect to stimulus range. For the average derivative of the tuning curves, we obtain

$$\begin{aligned}\left\langle \|\mathbf{v}'(x)\|_2^2\right\rangle_W &= \left\langle \sum_{i=1}^N \left(\sum_{j=1}^L W_{ij}\frac{\partial u_j(x)}{\partial x}\right)^2\right\rangle_W = \left\langle \sum_{i=1}^N \left(\sum_{j=1}^L W_{ij}\frac{(x-c_j)}{\sigma^2}u_j(x)\right)^2\right\rangle_W \\ &= \sum_{i=1}^N\sum_{jj'}\langle W_{ij}W_{ij'}\rangle_W \frac{(x-c_j)(x-c_{j'})}{\sigma^4}u_j(x)u_{j'}(x) \\ &= \frac{N}{\sigma^4}\sum_{j=1}^L \frac{1}{L}(x-c_j)^2 u_j^2(x) \approx \frac{N}{\sigma^4}\int_0^1 dc_j\,(x-c_j)^2 u_j^2(x) \\ &\approx \frac{NA^2\sqrt{\pi\sigma^2}}{2\sigma^2}, \end{aligned} \tag{36}$$

where in the last step we approximated the sum with the integral and we took the limit of integration going to infinity, similarly to previous calculations. Finally, by using the approximation for small σ, $A^2 \approx R/\sqrt{\pi\sigma^2}$, we get the expression which appears in the main text

$$\varepsilon_l^2 = \langle E_l^2\rangle_W \approx \frac{2\sigma^2\eta^2}{RN}. \tag{37}$$

This expression is equivalent to the inverse of the average Fisher Information, which, in case of neurons affected by i.i.d Gaussian noise, is given by $J(x) = \|\mathbf{v}'(x)\|_2^2/\eta^2$.

Global error. In computing an approximation for the scaling of global errors, we extend the reasoning we have done for discrete stimuli. By assuming that the magnitude of a global error is independent on its probability, we write an expression similar to Eq. (26),

$$\left\langle E_g^2\right\rangle_W = \langle P(E)\rangle_W \left\langle \int_0^1 dx(\hat{x}-x)^2\right\rangle_W. \tag{38}$$

We use the approximation that, in case of global error, the decoded stimulus, averaging over different distributions of synaptic weights, is uniformly distributed in the interval $\hat{x} \notin [x-\sigma, x+\sigma]$. The average magnitude of global errors is therefore

$$\begin{aligned}\bar{\varepsilon}_g &= \left\langle \int_0^1 dx\,(\hat{x}-x)^2\right\rangle_W \approx \int_0^1 dx \frac{1}{(1-2\sigma)}\left[\int_0^{x-\sigma} d\hat{x}(\hat{x}-x)^2 + \int_{x+\sigma}^1 d\hat{x}(\hat{x}-x)^2\right] \\ &= \frac{1}{6}\frac{\left(1-4\sigma^3\right)}{(1-2\sigma)} \approx \frac{1}{6}, \end{aligned} \tag{39}$$

where, for simplicity, we considered only the case where $x-\sigma > 0$ and $x+\sigma < 1$. The important thing to notice here is that it is a term of order 1, the size of the stimulus range.

The probability of an error being global, averaged over different realizations of W, does not depend on the specific value of the stimulus. Computing this probability rigorously is hard, due to the correlations between nearby responses. Nevertheless, we know that for stimuli at a distance greater than σ the two responses are basically uncorrelated, Eq. (17). We divide the curve into $1/\sigma$ uncorrelated segments of responses, and we consider the distance between these segments sampled randomly; therefore, it is sufficient to substitute to L in Eq. (5) the number of segments, to obtain the expression of the global error which appears in the main text,

$$\varepsilon_g^2 = \left\langle E_g^2\right\rangle_W \approx \frac{1}{\sigma\sqrt{2\pi N}}\bar{\varepsilon}_g \exp\left(-\log\left(1+\frac{R}{2\eta^2}\right)\frac{N}{2}\right). \tag{40}$$

Input noise. We consider the case in which the first layer responses are affected by i.i.d Gaussian noise, $\tilde{\mathbf{u}}(x) = \mathbf{u}(x) + \mathbf{z}^{\mathbf{u}}$, with $z_i^u \sim \mathcal{N}(0, \xi^2)$. This results in a multivariate Gaussian distribution for the responses of the second layer, Eq. (20), with covariance matrix $\mathbf{\Sigma} = \eta^2\mathbf{I} + \xi^2\mathbf{W}\mathbf{W}^T$. The matrix $\mathbf{W}\mathbf{W}^T$ follows the well known Wishart distribution, with mean $\mathbf{I}$ and fluctuations of order $1/L$. We rewrite the covariance matrix as the sum of the identity plus a perturbation

$$\mathbf{\Sigma} = \tilde{\eta}^2\mathbf{I} + \xi^2(\mathbf{W}\mathbf{W}^{\mathbf{T}} - \mathbf{I}), \tag{41}$$

where $\tilde{\eta}^2 = \eta^2 + \xi^2$. In order to obtain an estimate of the effects of input noise on the local error, we consider the inverse of the Fisher Information (FI) as a lower bound to the MSE. For correlated populations, the FI is given by Shamir & Sompolinsky (2006)

$$J(x) = \mathbf{v}'(x)^T\mathbf{\Sigma}^{-1}\mathbf{v}'(x). \tag{42}$$

If the perturbation matrix is small, we can approximate the inverse of the correlation matrix at the second order $\mathbf{\Sigma}^{-1} \approx \frac{1}{\tilde{\eta}^2}\mathbf{I} - \frac{\xi^2}{\tilde{\eta}^4}\left(\mathbf{W}\mathbf{W}^{\mathbf{T}} - \mathbf{I}\right) + \frac{\xi^4}{\tilde{\eta}^6}\left(\mathbf{W}\mathbf{W}^{\mathbf{T}} - \mathbf{I}\right)^2$, and write the FI as

$$\begin{aligned} J(x) &\approx J^{ind}(x) - \delta J(x) \\ &= \frac{\|\mathbf{v}'(x)\|_2^2}{\tilde{\eta}^2} - \frac{\xi^2}{\tilde{\eta}^4}\mathbf{u}'^T(x)\left(\mathbf{B}^2 - \mathbf{B}\right)\mathbf{u}'(x) + \frac{\xi^4}{\tilde{\eta}^6}\mathbf{u}'^T(x)\left(\mathbf{B}^3 - 2\mathbf{B}^2 + \mathbf{B}\right)\mathbf{u}'(x), \end{aligned} \tag{43}$$

where $\mathbf{B} = \mathbf{W}^{\mathbf{T}}\mathbf{W}$. We recognize in the first term, $J^{ind}(x)$, the FI in the case of i.i.d Gaussian output noise with variance $\tilde{\eta}^2$. All the correction terms to the FI are related to the moments of the matrix $\mathbf{B}$. Since all the entries are Gaussian, it is possible to compute their mean through the Isserlis' theorem. Using the identity $\langle W_{ij}W_{mn}\rangle_W = \frac{1}{L}\delta_{im}\delta_{jn}$, we obtain:

$$\langle B_{mn}\rangle_W = \left\langle \sum_{j=1}^{N} W_{jm}W_{jn} \right\rangle_W = \frac{N}{L}\delta_{mn}, \tag{44}$$

$$\langle B_{mn}^2\rangle_W = \left\langle \sum_{i=1}^{L}\sum_{j=1,j'=1}^{N} W_{jm}W_{ji}W_{j'i}W_{j'n} \right\rangle_W = \left\langle \frac{N}{L} + \frac{N^2}{L^2} + \frac{N}{L^2} \right\rangle \delta_{mn}, \tag{45}$$

$$\langle B_{mn}^3\rangle_W = \left\langle \sum_{i=1,i'=1}^{L}\sum_{j=1,j'=1,j''=1}^{N} W_{jm}W_{ji}W_{j'i}W_{j'i'}W_{j''i'}W_{j''n} \right\rangle_W = \left(\frac{N^3}{L^3} + 3\frac{N^2}{L^3} + 3\frac{N^2}{L^2} + 4\frac{N}{L^3} + 3\frac{N}{L^2} + \frac{N}{L}\right)\delta_{mn}. \tag{46}$$

To leading order in N/L, the mean of the perturbation term read

$$\begin{aligned} \langle \delta J(x)\rangle_W &\approx \frac{N^2\xi^2}{L^2\tilde{\eta}^4}\mathbf{u}'(x)^T\mathbf{I}\mathbf{u}'(x) - \frac{N^2\xi^4}{L^2\tilde{\eta}^6}\mathbf{u}'(x)^T\mathbf{I}\mathbf{u}'(x) \\ &= \frac{N^2\xi^2A^2\sqrt{\pi\sigma^2}}{2L\tilde{\eta}^4\sigma^2}\left(1 - \frac{\xi^2}{\tilde{\eta}^2}\right), \end{aligned} \tag{47}$$

where we computed $\mathbf{u}'(x)^T\mathbf{I}\mathbf{u}'(x) = \sum_{j=1}^{L} u_j'(x)^2$ in the same way of Eq. (36). By inserting Eq. (36) and Eq. (47) in Eq. (43), we obtain the expression for the FI in case of input noise

$$\langle J(x)\rangle_W \approx \frac{A^2N\sqrt{\pi\sigma^2}}{2\sigma^2\tilde{\eta}^2}\left(1 - \frac{N\xi^2}{L\tilde{\eta}^2} + \frac{N\xi^4}{L\tilde{\eta}^4}\right), \tag{48}$$

and the approximation for the MSE which appears in the main text,

$$\varepsilon_l^2 = \langle E^2\rangle_W \approx \frac{1}{\langle J(x)\rangle_W} \approx \varepsilon_{l,\text{ind}}^2\left(1 + \frac{N\xi^2}{L\tilde{\eta}^2} - \frac{N\xi^4}{L\tilde{\eta}^4}\right), \tag{10 restated}$$

where $\varepsilon_{l,\text{ind}}^2 \approx 2\sigma^2\tilde{\eta}^2/RN$. Similar calculations can be done assuming a covariance matrix with the same statistic, but uncorrelated with the synaptic weights. As an example, we considered $\mathbf{\Sigma}_{\boldsymbol{rand}} = \eta^2 I + \xi^2\mathbf{X}\mathbf{X}^T$ with $X_{ij} \sim \mathcal{N}(0, \frac{1}{L})$ such that $\langle X_{ij}W_{mn}\rangle_{W,X} = 0$. In this case we have no first order corrections, and the FI is increased

$$\langle J(x)\rangle_{W,X} \approx \frac{A^2N\sqrt{\pi\sigma^2}}{2\sigma^2\tilde{\eta}^2}\left(1 + \frac{N\xi^4}{L\tilde{\eta}^4}\right), \tag{49}$$

yielding a lower MSE, Eq. (11).

Extension to multidimensional stimuli

We consider stimuli in the hypercube $\mathbf{x} \in [0,1]^K$and the two extreme cases of pure and conjunctive sensory neurons.

Pure case. Each sensory neuron is sensitive to a single stimulus dimension, x_k. The L neurons are equally assigned to stimulus dimensions, such that each dimension is monitored by $Q = L/K$ neurons. The activity of neuron $u_{j,k}$ is given by

$$u^p_{j,k}(\mathbf{x}) = u^p_{j,k}(x_k) = A_p \exp\left(-\frac{\left(x_k - c^k_j\right)^2}{2\sigma^2}\right), \tag{50}$$

with preferred stimuli evenly spaced, $c^k_j = j/Q$ for $j = 1, ..., Q$. The responses of second layer neurons are given by a random sum of all sensory neurons, and can be written as a superposition of one-dimensional tuning curves, independent for each dimension,

$$\begin{aligned} v^p_i(\mathbf{x}) &= \sum_{k=1}^{K}\sum_{j=1}^{Q} W_{ijk} u_{j,k}(\mathbf{x}) \\ &= \sum_{k=1}^{K} v^p_{i,k}(x_k). \end{aligned} \tag{51}$$

Imposing the resource constraint, Eq. (12), with similar calculations we obtain $A^2_p = R/\left(\left(\pi\sigma^2\right)^{1/2} - 2\pi\sigma^2\right)$.

The local error along each dimension is computed, similarly to Eq. (34), expanding linearly the surface, and obtaining

$$\Delta x^2_k \approx \frac{\left\|\mathbf{z}\cdot\hat{\mathbf{v}'}_k(\mathbf{x})\right\|^2_2}{\left\|\mathbf{v}'_k(\mathbf{x})\right\|^2_2}, \tag{52}$$

where $\mathbf{v}'_k = \partial\mathbf{v}(\mathbf{x})/\partial x_k$. The calculation is analogous to the one-dimensional case, but the derivative along each dimension acts only on $1/K$ terms. As a consequence, the local error along each dimension is

$$\varepsilon^2_{l,p,k} = \frac{2K\sigma^2\eta^2}{NA^2_p\sqrt{\pi\sigma^2}} \approx \frac{2K\sigma^2\eta^2}{RN}, \tag{53}$$

and the total one is $\varepsilon^2_{l,p} = K\varepsilon^2_{l,p,k}$.

As for global errors, since the multi-dimensional tuning curves are superposition of one-dimensional ones, we can obtain a global error on each dimension independently. By assuming that the probability of having a global error on more than one dimension is negligible, we can approximate the total probability of having a global error as the sum of probabilities along each dimension, $P(E_g) = \sum_{k=1}^{K} P(E_{g,k})$. In computing the probability along each dimension, we have to keep into account that, as the total variance across all stimulus space is equal to R, the variance across each dimension is reduced by a factor of K. By inserting R/K instead of R in Eq. (5) and summing over the dimensions, we obtain that the global error for the pure case scales approximately as

$$\varepsilon^2_{g,p} \approx \frac{K\bar{\varepsilon}_g}{\sigma\sqrt{2\pi N}} \exp\left(-\log\left(1 + \frac{R}{2K\eta^2}\right)\frac{N}{2}\right), \tag{54}$$

where the average magnitude of global error, $\bar{\varepsilon}_g$, is again a term of order 1.

Conjunctive case. In this case the response of sensory neurons are multi-dimensional Gaussian,

$$u^c_j(\mathbf{x}) = A_c \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_j\|^2_2}{2\sigma^2}\right), \tag{55}$$

with preferred stimuli arranged on a K-dimensional square grid of side $L^{-1/K}$. The response of the second layer neurons are given by

$$v^c_i(\mathbf{x}) = \sum_{j=1}^{L} W_{ij} u^c_j(\mathbf{x}). \tag{56}$$

In this case, the tuning curves of the representation layer are multi-dimensional Gaussian processes with covariance function $\langle v_i(\mathbf{x}) v_i(\mathbf{x} + \Delta\mathbf{x})\rangle_W = A^2_c \left(\pi\sigma^2\right)^{K/2} \exp\left(-\|\Delta\mathbf{x}\|^2_2/4\sigma^2\right)$. By imposing the resource constraint,

we obtain $A_c^2 = R/\left(\left(\pi\sigma^2\right)^{K/2} - (2\pi\sigma^2)^K\right)$. We notice that, as K becomes large, the edge effects in the approximations such Eq. (13) become more relevant, and the denominator may become negative. For high number of dimensions, we may need broader widths to cover the stimulus space, and the difference between Gaussian integrals $\left(\int d\mathbf{x} u_j(\mathbf{x})\right)^2$ and $\int d\mathbf{x} u_j^2(\mathbf{x})$ change sign for large values of σ. We didn't encounter this problem for $K = 3$ (results of the main text) and in the range of values for σ we explored.

When we compute the local error, Eq. (52), the derivative acts on all the terms of the sum, as all sensory neurons are sensitive to stimulus variations. As a result, we obtain, for the local error along each dimension in the conjunctive case,

$$\varepsilon_{l,c,k}^2 = \frac{2\sigma^2\eta^2}{NA_c^2(\pi\sigma)^{K/2}} \approx \frac{2\sigma^2\eta^2}{RN}, \tag{57}$$

Simarly, the total one will be $\varepsilon_{l,c}^2 = K\varepsilon_{l,c,k}^2$. The ratio between local errors in the two cases is therefore $\varepsilon_{l,c}^2/\varepsilon_{l,p}^2 = 1/K$.

As for global errors, we extend the reasoning of the one-dimensional case, dividing the K -dimensional surface of responses into $1/\sigma^K$ regions of correlated stimuli. By substituting the number of uncorrelated regions to L in Eq. (5), we obtain that the global errors scales approximately as

$$\varepsilon_{g,c}^2 \approx \frac{1}{\sigma^K\sqrt{2\pi N}}\bar{\varepsilon}_g \exp\left(-\log\left(1+\frac{R}{2\eta^2}\right)\frac{N}{2}\right). \tag{58}$$

Data analysis and model fitting

Data description and summary statistics The detailed data description is reported in Lalazar et al. (2016), and data are publicly available at https://osf.io/u57df/. They consist of the responses (firing rates) of $N \sim 500$ neurons, recorded during an arm posture 'hold' task at 27 different positions, with 2 hand orientations, up and down, arranged on a virtual cube of size 40x40x40 cm. The response of each neuron for each position is recorded for several trials ($\sim$ 10 trials per position). Tuning curves are computed by averaging over trials. In order to measure the level of irregularity of a single tuning curve in a non parametric form, the authors introduced a complexity measure : for each neuron, it is defined as the standard deviation of the discrete derivative between the response at one target position, $\mathbf{x}$ and its response at the closest target, $\mathbf{x} + D_{min}$

$$c(D_{min})_i = std\left(\frac{\|\mathbf{v}(\mathbf{x}) - \mathbf{v}(\mathbf{x}+\Delta\mathbf{x})\|}{\sqrt{\|\Delta\mathbf{x}\|^2}} s.t.\ \|\Delta\mathbf{x}\|_2^2 < D_{min}\right), \tag{59}$$

where $\mathbf{v}(\mathbf{x})$ is the mean response at stimulus $\mathbf{x}$. In the data, the D_{min} is imposed by the experiment and is equal to 35. This limitation, inherent to the data themselves, prevent us from capturing high frequency components due to aliasing phenomena. The author measured also another summary statistic, the distribution of R^2 values resulting from the fit of the tuning curves with a linear model, Eq. (9),

$$R_i^2 = 1 - \frac{\sum_{\mathbf{x}}\left(\mathbf{v}_l(\mathbf{x}) - \mathbf{v}(\mathbf{x})\right)^2}{\sum_{\mathbf{x}}\mathbf{v}(\mathbf{x})^2}, \tag{60}$$

where $\mathbf{v}_l(\mathbf{x})$ is the response predicted by a fitted linear model, and the sum is over the stimuli used in the experiment. The distribution of these quantities across different neurons is a measure of the irregularity of the neural population; if the population were perfectly described by Eq. (9), the R^2 distribution would be a delta function peaked at 1, while the complexity measure would be biased towards low values.

Model fitting. We considered the tuning curves as a function of the position only, ignoring the difference in hand orientation. We analyzed neurons responding with at least 5 spikes/s at more than two positions. We shifted and normalized the data such that tuning curves have zero mean and unit variance across different stimuli. We generated an irregular population with our model, featured by a sensory layer of conjunctive neurons responding to a three-dimensional stimulus. We used $L = 100^3$ neurons, tiling a 200 by 200 by 200 cube, such that the stimulus space is covered without boundary effects, with preferred stimuli arranged on a square grid of side 2. For computational reasons, due to the slowness of multiplying large full matrices, $\mathbf{W}$ is taken as a sparse random matrix (sparsity equal to 0.1) with Gaussian entries, similarly to the model of Lalazar et al. (2016). **[MIRKO: Add discussion about sparsity here? (SIMONE: I performed some simulations with sparse matrices. Sparsity does not affect our results, as long as the representation neurons receive enough inputs from the first layer, therefore as long as sparsity $> \sigma$. Should we mention it?)]**The tuning curves in the second layer were normalized one by one to have variance equal to 1. With respect to the

model of Lalazar et al. (2016), there are two main differences: in their case the random weights were distributed according to a uniform distribution, and the random sum was passed through a threshold-linear function. With this formulation, the model had two tunable parameters: the tuning width of first layer neurons, σ, and the the threshold of the non linear function of the second layer. The only tunable parameter of our model is σ.

In order to fit the model, we generated the neural responses of a number of representation neurons equal to the number of recorded neurons at the same stimuli (27) of the data. We then computed the distribution of the complexity measure for different values of σ and we chose σ_f such as to minimize the Kolmogorov-Smirnov (KS) distance between the distribution of the model and the one of the data (Fig. 9A). This is a measure of distance between two probability distributions, and it is obtained as the superior of the difference between the two empirical cumulative distribution functions, $F_{model}(c)$ and $F_{data}(c)$,

$$KS(\sigma) = \sup_c |F_{model}(c) - F_{data}(c)|. \tag{61}$$

At the minimum value σ_f, the two distributions are very similar, even if real data show a broader distribution of values in both directions; for comparison, the distribution implied by a linear model is biased towards lower values (Fig. 9B). For the sake of completeness, we computed the KS distance between the model and the data also for the R^2 measure (Fig. 9A, red line). This quantity simply decreases with σ, and the model at σ_f underestimate the linear component of the tuning curves (Fig. 9C). Nevertheless, this is expected, since our model has no non linearity, which potentially may increase the linear component of the tuning curves. It is worth noticing that in the original work, a model with two parameters still underestimates the distribution of R^2 values and only the complexity measure was considered in the fitting procedure. The authors obtained a good agreement only considering a model with more parameters (namely, different threshold for each neuron and different widths in the sensory layer).

We also performed simulations with a heterogeneous noise variance across the population extracted from the data. We assigned to each recorded neuron a signal-to-noise ratio in the following way. We estimated the variance of the signal as the variance of the mean responses across different stimuli, $\hat{R}_i = \langle (v_i(x) - \langle v_i(x) \rangle_x)^2 \rangle_x$. Then, we averaged the trial to trial variability, across different stimuli, $\hat{\eta}_i^2 = \left\langle \langle r_i^t - v_i(x) \rangle_t \right\rangle_x$, where r^t is the response at each trial. As in our model we kept the variance of the signal equal to 1, the noise variance in the i-th neuron in simulations was set equal to

$$\eta_i^2 = \frac{\hat{\eta}_i^2}{\hat{R}_i}. \tag{62}$$

In principle, the noise may be dependent from the mean. To control for this effect, we also preprocessed the data with a variance stabilizing transformation (substituting $r(\mathbf{x})$ with $\sqrt{r(\mathbf{x})}$, SRJ & Everitt (1999)). The distribution of the noise variance across neurons obtained in this way does not vary substantially.

For numerical simulations in Fig. 6, the tuning curves are computed at a finer scale than the data (cubic grid of 21 by 21 by 21 points). As expected, the tuning curves show a broad range of behavior with respect to the linear fit, that goes from very linear to very irregular (Fig. 9 D-F). The linear population for the comparison is constructed by sampling the preferred vectors ,$\mathbf{P}_i = (b_1, c_2, d_3)$, uniformly on the unit sphere and using Eq. (9) to generate the responses. Similarly to the irregular ones, these tuning curves are shifted and normalized to have zero mean and unit variance.

5 Acknowledgements

[SIMONE to RAVA: Plotting the error bars in a log-scale often results in bad lookign shaded regions, asymmetric around the mean. Having a look to the literature, maybe is better to plot, instead of the s.d., the relative error, that is instead of plotting $x + \delta x$, we plot $x + \delta x/x$, which indeed look better and is symmetric. What do you think about it?]

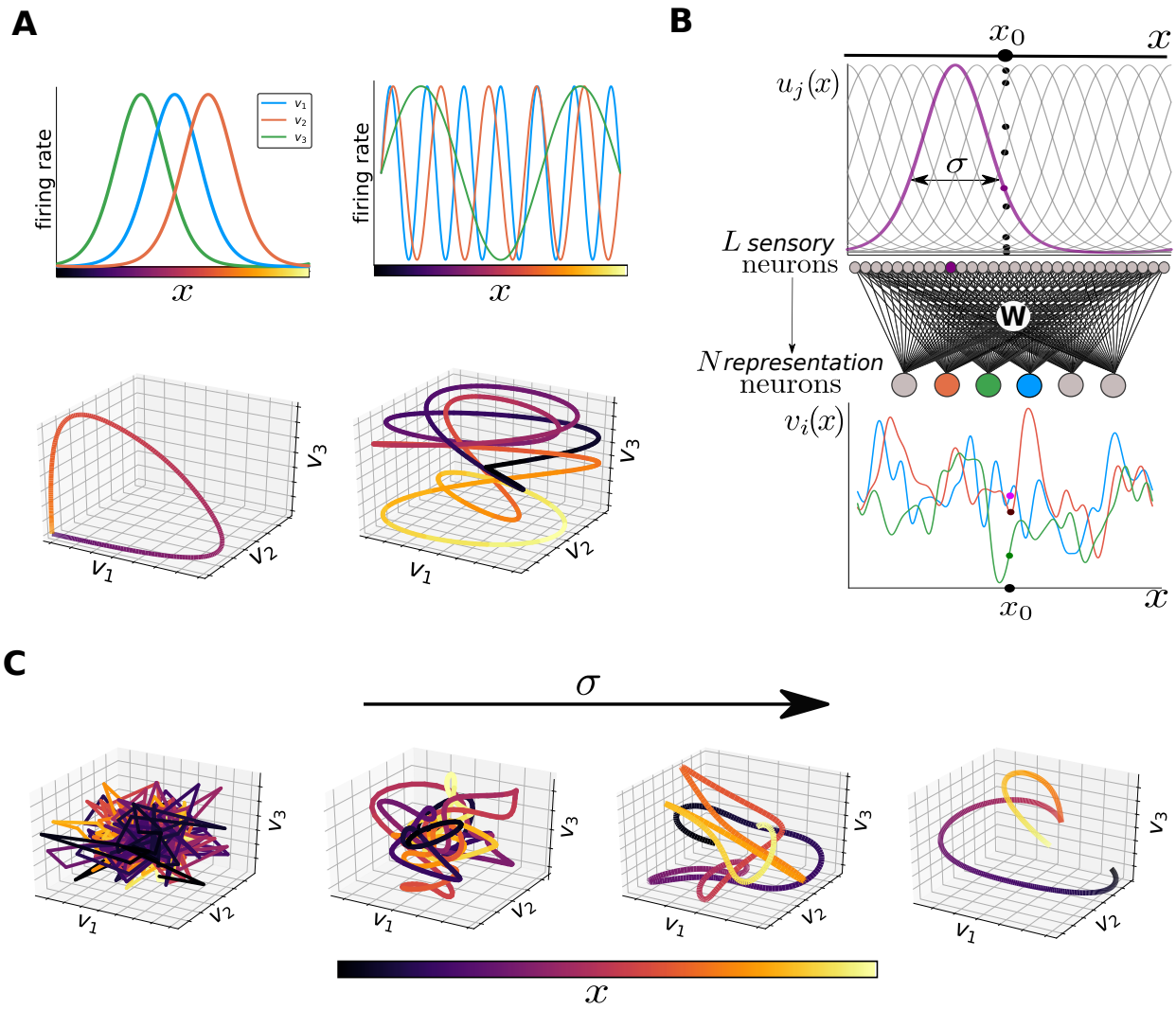

Figure 1: **Geometrical approach to coding, and the random feedforward neural network architecture.**(**A**) Top: mean responses of neural populations encoding a one-dimensional stimulus. Left: population of neurons with Gaussian, translationally invariant tuning curves. Right: population of neurons with periodic tuning curves. We note that grid cells are composed of periodic Gaussian activity bumps, and are thus not sinusoidal. For the sake of illustration, we plotted three sinusoids with three different periods. Bottom: joint activity of the neural population, as a function of the stimulus value, colored according to the legend, corresponds to a one-dimensional curve in a N-dimensional space. We show a three-dimensional subspace, corresponding to the responses of the highlighted neurons. Unimodal tuning curves (left) evoke a single-loop curve, which preserves the distances between stimuli in the evoked responses. Instead, periodic tuning curves (right) evoke a more complex curve, and it can happen that two distant stimuli are mapped to nearby points in the activity space. At the same time, the curve is longer, and fills up a larger portion of the possible activity space. (**B**) Feedforward neural network. An array of L sensory neurons with Gaussian tuning curves (one highlighted in purple) encodes a one-dimensional stimulus into an high dimensional representation. These tuning curves determine the response of the population for a given stimulus, x_0 (dots). This layer projects onto a smaller layer of N representation neurons with an all-to-all random connectivity matrix $\mathbf{W}$, generating irregular responses. We plotted the tuning curves of three sample neurons, highlighting their response to the stimulus x_0. (**C**) Example of joint activity as a function of the stimulus, colored according to the previous legend, shown for three sample neurons of the second layer, for increasing σ. When $\sigma \to 0$ (left, $\sigma = 0.001$), neurons generates uncorrelated random responses to different stimuli, generating a spiky curve made up by broken segments. As σ grows ($\sigma = 0.015$, $\sigma = 0.03$) irregularities are smoothed out, and nearby stimuli evoke increasingly correlated responses. By decreasing the complexity of the curve, we ultimately recover the scenario of unimodal tuning curves (right, $\sigma = 0.15$).

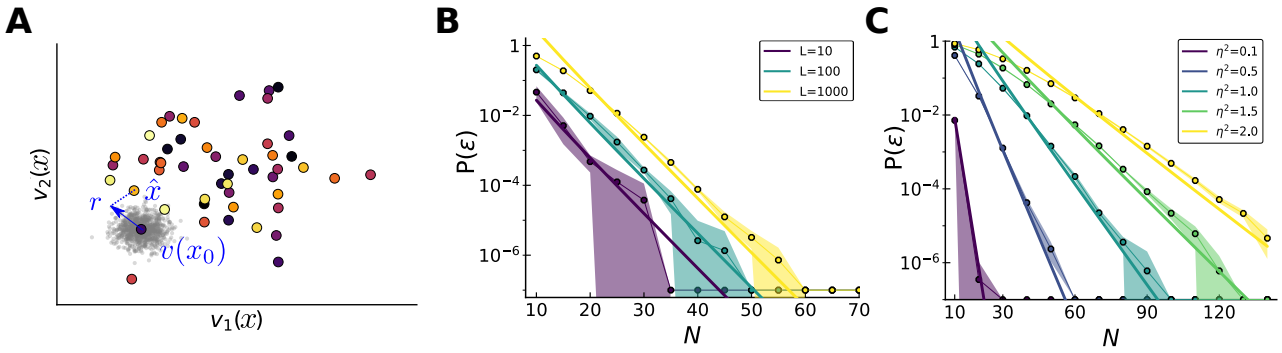

Figure 2: **Error probability for discontinuous random responses.** (**A**) Joint responses of two neurons to $L = 50$ stimuli, colored according to the previous legend. Noise is represented as a cloud of possible responses (in grey) around the mean. An error occurs when the noisy response $\mathbf{r}$ happens to be closer to a point representing another stimulus $\hat{x}$ than the true one x_0. Since responses are uncorrelated, that point may represent a distant stimulus. (**B**) Theoretical (solid curves) and numerical results (circles) for the probability of error as a function of the population size, for different numbers of discrete stimuli encoded with uncorrelated random responses ($\eta^2 = 0.5$, averaged over 8 network realizations, shaded region corresponds to 1 s.d.). The error probability scales exponentially with the number of neurons, with a multiplicative constant given by the number of stimuli. The high variance is due to the difficulty in estimating probabilities when they are very low. (**C**) Theoretical (solid curves) and numerical results (circles) for the probability of error as a function of the population size for $L = 500$ discrete stimuli, for different noise magnitudes. Results are averaged over 8 network realizations, shaded region corresponds to 1 s.d.

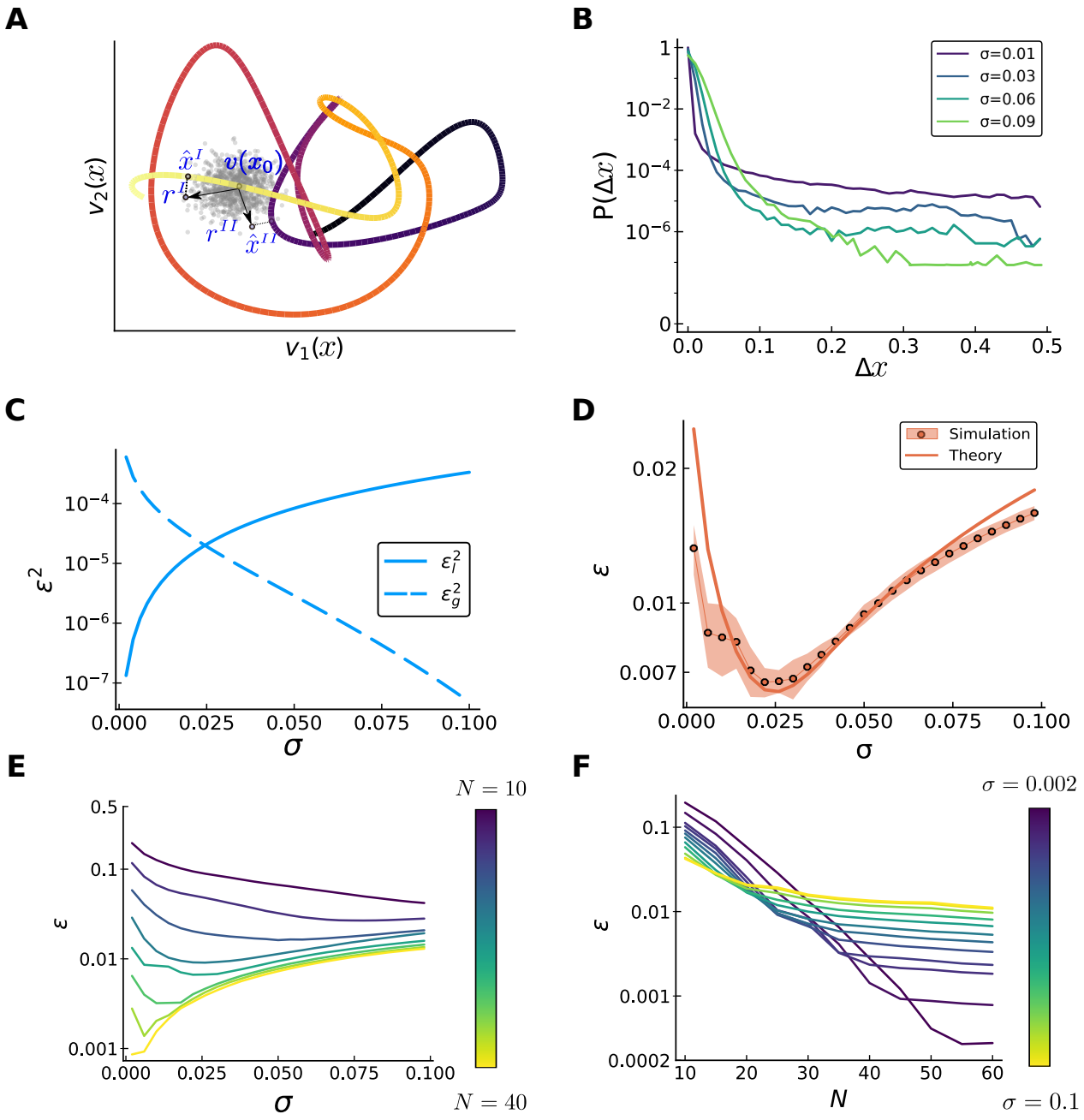

Figure 3: **Trade-off between local and global errors.** In all simulations, $L = 500$ and $\eta^2 = 0.5$, $R = 1$.
continue to next page

Figure 3 *(previous page)*: (**A**) Different types of error in a complex curve of mean population activities (joint response of two neurons, colored according to previous legend). Here, $\mathbf{r}^I$ and $\mathbf{r}^{II}$ are two possible noisy responses to the same stimulus, extracted from the Gaussian cloud surrounding the mean response, $\mathbf{v}(x_0)$. An ideal decoder will output the stimulus corresponding to the closest point of the curve. In one case, $\mathbf{r}^I$ will cause a local error, falling on a point of the curve that represents a similar stimulus, $\hat{x}^I$. Instead, $\mathbf{r}^{II}$ happens to be closer to a point of the curve which represents a stimulus quite far from the true one, $\hat{x}^{II}$, causing a global error. (**B**) Normalized histogram of absolute errors, $\Delta x = |\hat{x} - x|$, made by an ideal decoder, for different values of σ ($N = 25$). We tested the response to 10^7 stimuli, uniformly spaced between $[0, 1]$, and we averaged over 8 realizations of the connectivity matrix. For better visualization, we considered a stimulus with periodic boundary conditions, such that all global error magnitudes have the same probability. Contributions of the two types of error varies with σ. For small σ, coding is very precise locally (fast drop of the purple curve for small errors), but we have a great number of global errors (tail of the distribution is high). Vice versa, smoother codes (green curves) yield poor local accuracy (larger local errors), but high noise robustness (very few large scale errors). (**C**) Theoretical prediction for the two contributions to the MSE as a function of σ ($N = 30$). The magnitude of local errors increases with larger widths (solid curve), while the number of global errors decreases (dashed curve). (**D**) RMSE as a function of σ: comparison between numerical simulations (solid curve) and theoretical prediction of Eq.(6) (dots). Results are averaged over 8 network realizations, shaded region corresponds to 1 s.d. (**E**) RMSE, as a function of σ for different population sizes N (increasing from violet to yellow). The optimal error is attained at an optimal $\sigma^*(N)$, which decreases with increasing N. (**F**) Same data, but the error is showed as a function of N, for a fixed value of σ. The error at first decreases exponentially fast until global errors are suppressed, then the local errors are linearly reduced. Decreasing σ, we increase the N at which the transition occurs, but also the error at this critical value.

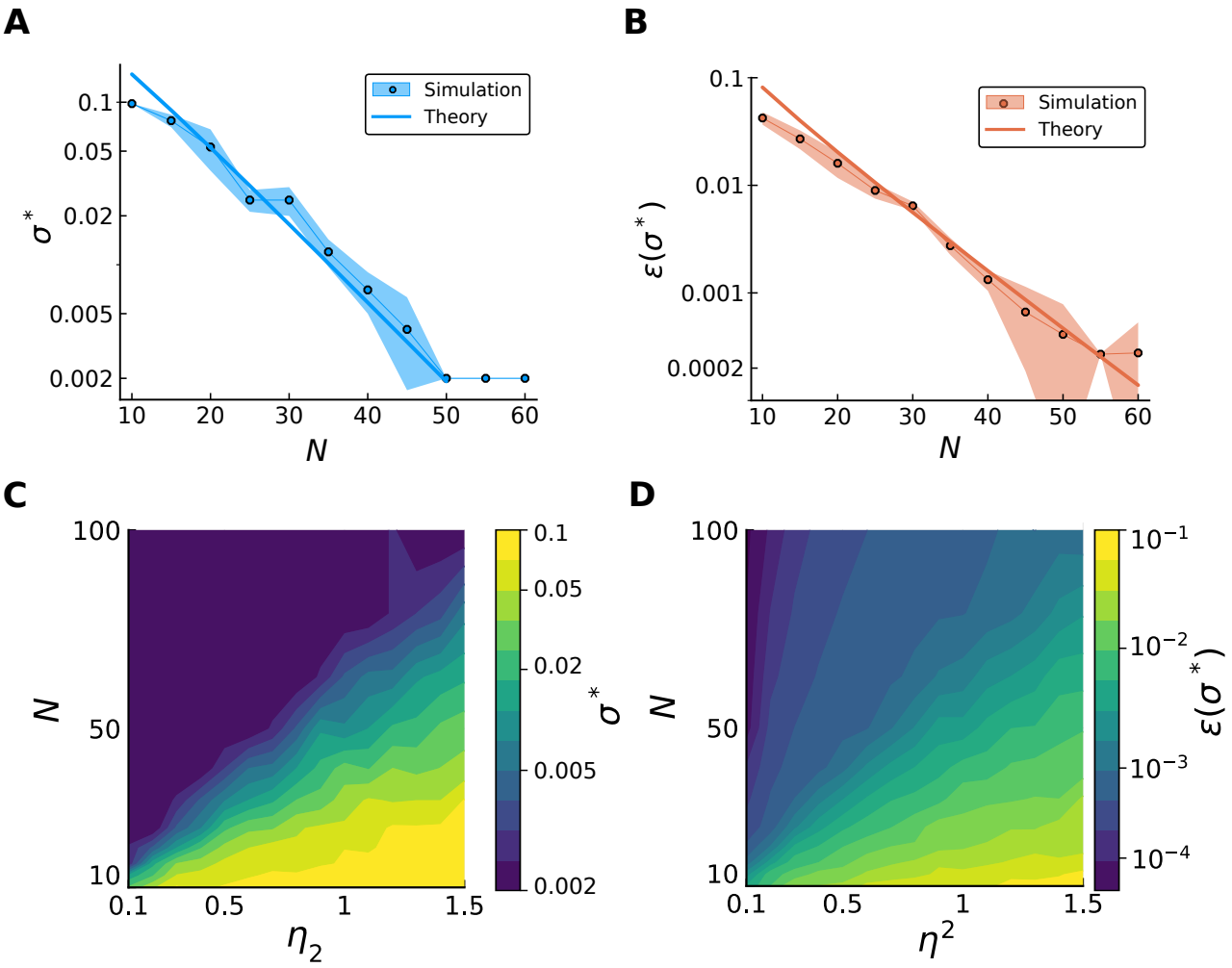

Figure 4: **Scaling of the optimal width and optimal error as a function of population size and signal-to-noise ration.** In all simulations $L = 500$, and results are averaged over 8 network realizations. In (**A-B**) $\eta^2 = 0.5$. (**A**) The mean optimal σ^* decreases exponentially fast with the number of neurons, saturating the lower bound imposed by the finite number of neurons of the first layer (corresponding to the spacing of the preferred positions, $1/L$). Simulations (circles) show good agreement with the theory (solid line). Shaded region corresponds to 1 s.d. (**B**) As a consequence, the optimal error, $\varepsilon(\sigma^*)$, which is linear in σ, is also suppressed exponentially fast in N. As before, simulations (circles) are well predicted by the theory (solid curve). (**C,D**) Optimal width (**C**) and error (**D**) as a function of the parameters $N - \eta^2$. The color code is in log scale, used in order to highlight the exponential scaling.

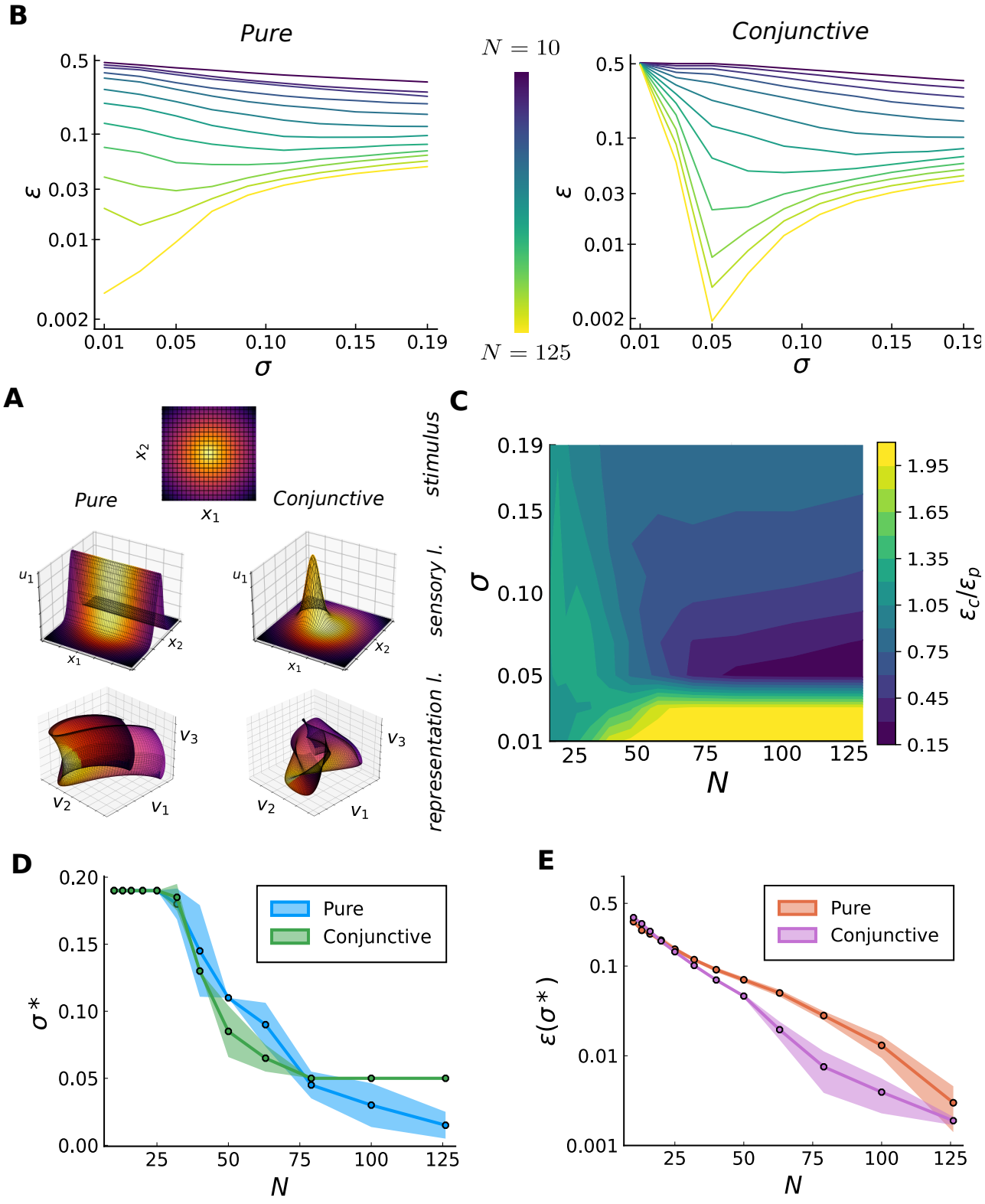

Figure 5: **Compressed coding for high-dimensional stimuli.** Here, for numerical results, we illustrated the case of a three-dimensional stimulus, $L = 3375$, $\eta^2 = 1$, $R = 1$; results are averaged over 8 network realizations.
continue to next page

Figure 5 *(previous page)*: (**A**) Subsequent mapping of high-dimensional stimulus space into neural activity operated by the two layer coding scheme. Top: two-dimensional stimulus space, colors serving as legend for following plots. Middle: firing rate (z-axis) of a sample sensory neuron, for the two cases, as a function of the two stimulus coordinates (x- and y- axis), colored according to the previous legend. In the pure case (left), a single sensory neuron 'fold' the two-dimensional sheet across a direction, specified by its preferred position and dimension (here, x_2). In the conjunctive case (right), a sensory neuron creates a 'bump' in the sheet. Bottom: joint activity of three representation neurons as a function of the stimulus, colored according to the previous legend. Each of these neurons will randomly sum the transformations of sensory neurons, producing a randomly 'folded' sheet in the pure case (left) and a 'crumpled' sheet in the conjunctive case (right). (**B**) RMSE as a function of σ for different population sizes N (increasing from violet to yellow), when the first layer consists of pure (left) or conjunctive (right) cells. An optimal σ, decreasing with N, optimizes the balance between local and global errors, similarly to the one-dimensional case. In the conjunctive case the rapid increase of the error below $\sigma = 0.05$ is due to the sensory neurons not tiling the space, and it is independent of N. (**C**) Mean ratio between the error in the two cases, $\varepsilon_c/\varepsilon_p$, as a function of σ and population size. Yellow (violet) region indicates an outperformance of the pure (conjunctive) population. To aid visualization, the yellow region indicates all the values greater than 2. This regime of small sigma values is characterized by a better coverage of the pure population, independently of the output layer population size. Values greater than 1 occur also when N is small, due to the prefactor of the global error being lower in the pure case. As soon as N is sufficiently large and σ is sufficiently large to allow for good coverage of the stimulus space, the conjunctive case outperforms the pure case. This effect is stronger in the low σ region, due to the slower scaling of the global errors in the pure case. On the other hand, when sigma is sufficiently large the ratio saturates at the value given by the ratio of the local errors. (**D,E**) Optimal tuning width (**D**) and relative error (**E**), for pure (blue) and conjunctive (red) cases. Shaded region corresponds to 1 s.d. The global error decreases more slowly in the pure population, as one can see from both the optimal width and the total error being larger and with a smaller slope. At very low population sizes, the difference in the prefactor of global errors leads to slightly better performance of the pure code. For $N \gtrsim 75$ the optimal width in the conjunctive case saturates, due to the loss of coverage. The relative error stops decreasing exponentially and starts decreasing only linearly, while the pure population does not suffer from this limitation.

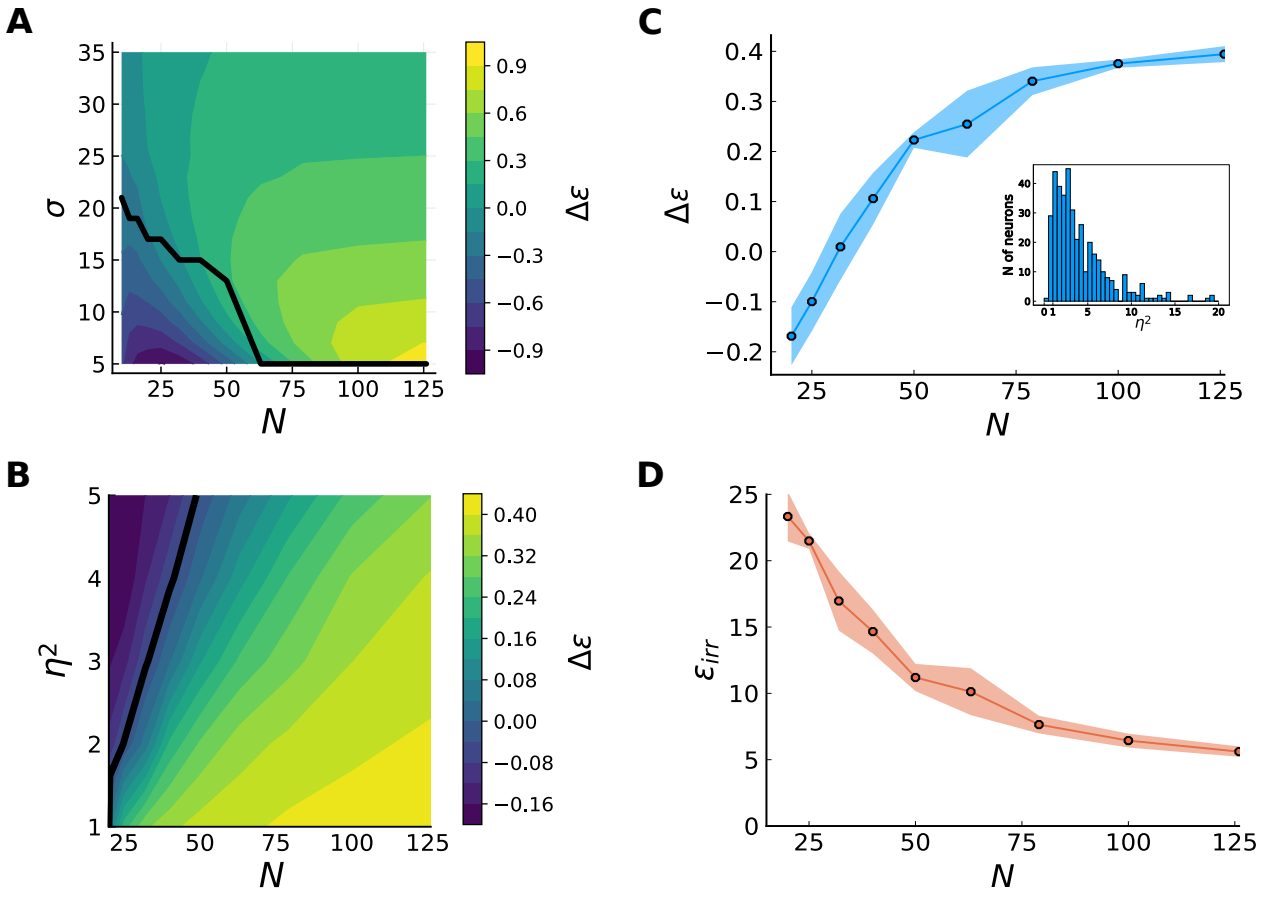

Figure 6: **Linear vs irregular tuning.** (**A**) Mean percent improvement of the irregular population (averaged over 8 different pools of a given size) compared to the linear one, as a function of population size and tuning width. The black line indicates the critical values of $N - \sigma$ at which they perform equally. In the region below (violet) global errors heavily affect the irregular population, making a smoother code more efficient. With increasing N, global errors become more rare while irregularities improve the local accuracy of the code (yellow region). The advantage increases at smaller values of σ, but so does the required value of N for the irregular population to be advantageous. (**B**) Mean percent improvement (averaged over 8 different pools of a given size) of an irregular population, generated with the data-fitted model, compared to the linear one, as a function of $N - \eta^2$. At small population sizes, the irregular tuning produces global error and smoother tuning curves perform better (violet region, $\Delta\varepsilon < 0$). By increasing N, global errors are suppressed and irregularities improve the local accuracy (yellow region,$\Delta\epsilon > 0$). The black line marks the transition values. (**C**,**D**) Mean percent improvement (**C**) and RMSE (**D**) of the irregular population as a function of population size (averaged over 8 different pools of neurons), for the noise model extracted from data. Shaded region corresponds to 1 s.d. A noise variance is assigned to each neuron according to the distribution extracted from the data, showed in the inset of panel (**C**). For low levels of N, linear tuning produces better results. At $N \sim 40$, the higher local accuracy compensates for global errors, and the irregular code starts to perform better, although the error is still substantial. The improvement saturates to a finite value of ~ 0.4 at a value of $N \sim 100$, when global errors are fully suppressed; the scaling of the error as a function of the population size is no more exponential, but only hyperbolic.

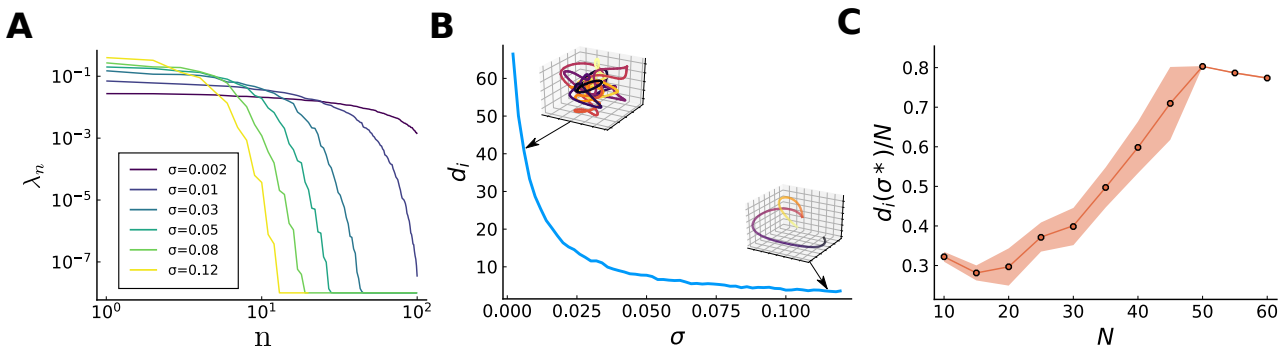

Figure 7: **Dimensionality of the neural code.** (**A**) Spectrum of the eigenvalues of the covariance matrix of neural responses of a population of $N = 100$ representation neurons, for different tuning widths (decreasing from violet to yellow). The spectrum is flat, denoting responses equally spread across different principal axes, up to a cut-off value, which decreases increasing σ, then it falls quickly to 0. (**B**) Intrinsic dimensionality, defined as the participation ratio of the eigenvalues of the covariance matrix, for a population of $N = 100$ representation neurons, as a function of σ. Insets showing a typical manifold in a three dimensional space. (**C**) Ratio of the intrinsic dimensionality at the optimal σ and population size (maximal value of intrinsic dimensionality) as a function of population size, for the networks of Fig. 3,4. Shaded region corresponds to 1 s.d.

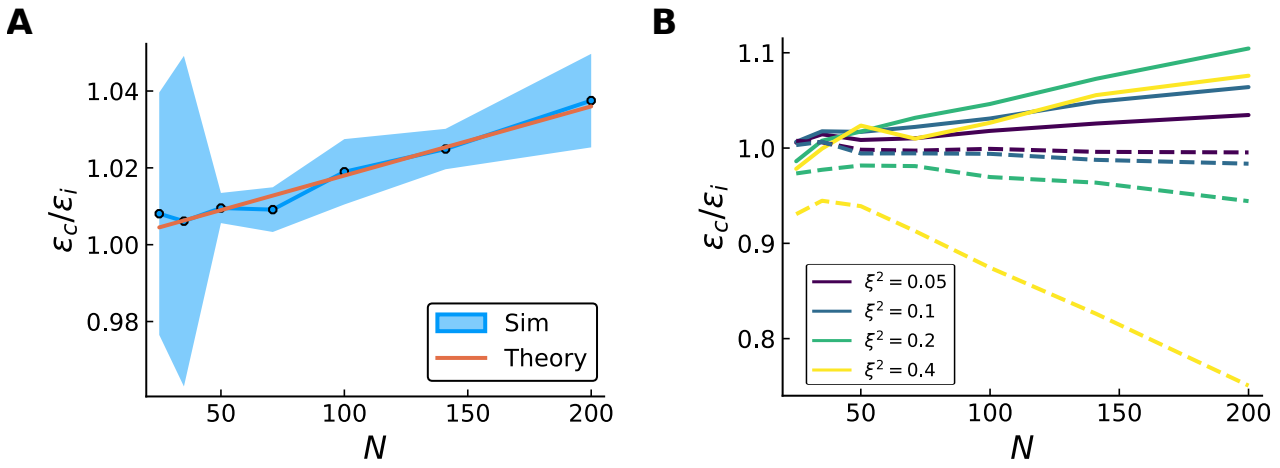

Figure 8: **Effects of correlated noise on compressed coding.** (**A**) Error ratio (MSE) between correlated noise due to input noise and variance-matched diagonal noise, as a function of N, and theoretical prediction, Eq.(10). $\sigma = 0.045$, $\tilde{\eta}^2 = 0.5$ and the contribution of input noise is small, $\xi^2 = 0.05$. Results are averaged over 8 network realizations, shaded region corresponds to 1 s.d. High variability for low values of N is expected, as global errors are more dependent on the specific realization of the weights. (**B**) Error ratio (MSE) between correlated noise due to input noise and diagonal noise (solid lines), and error ratio (MSE) between correlated noise with random covariance matrix and diagonal noise (dashed). Different colors denote different contributions coming from the off-diagonal terms ξ^2, increasing from violet to yellow, when the variance-matched noise is kept fixed, $\tilde{\eta}^2 = 0.5$. When correlations come from shared connections, the ratio is positive since we have information-limiting correlations. Their effect are a non-linear function of $\xi^2/\tilde{\eta}^2$, due to the competition between the first order (positive) and second order (negative) corrections. With a random covariance matrix, correlations decrease the error and enhance coding precision.

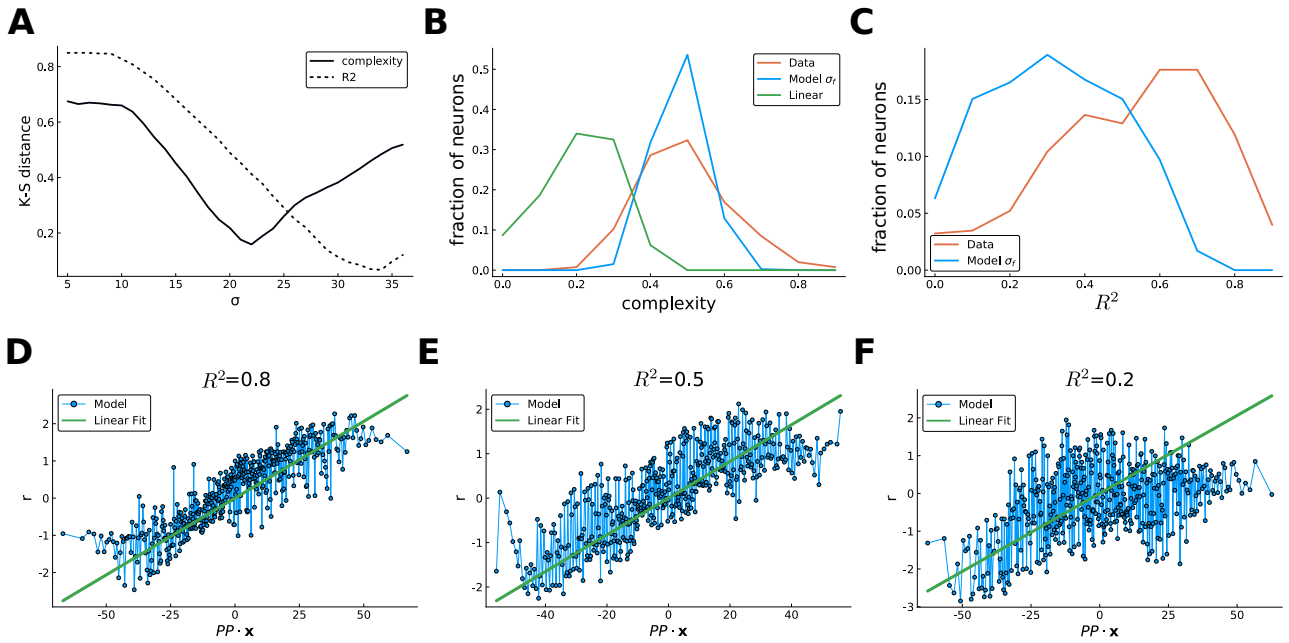

Figure 9: **Model fitting and tuning curves.** (**A**) Kolmogorov-Smirnov distance between the distributions of complexity measure (full line) and R^2 of fitting (dashed) across neurons from the data and the model for different σ. σ_f is chosen to be the value at which the minimum of the distance between complexity distributions is attained, $\sigma_f \sim 22$. (**B**) Normalized histogram of the distribution of complexity measure (arbitrary units) across the neurons of the data (red), the irregular population at σ_f (blue) and a linear population (green). The model is able to capture the bulk of the distribution of the real data much better than a linear model. Nevertheless, the data show a much broader distribution across the population. (**C**) Normalized histogram of the distribution of the R^2 of linear fit across neurons of the data and the irregular population at σ_f (red). Both distributions are broad, but the data show a more consistent linear part. (**D-F**) Three examples of tuning curves of the irregular population at σ_f, showing a broad range of behavior with respect to the linear fit. The tuning curves are plotted as a function of the projection of the stimulus (target position) onto a preferred position, obtained by the fit with Eq.(9) (green line). Some neurons are well described by the parametric function (d), some others show consistent deviations (e), while in others the linear behavior is absent (f). This is reflected in the broadness of the distribution of the R^2.

References

Abbott, L. F., Rolls, E. T., & Tovee, M. J. (1996). Representational capacity of face coding in monkeys. Cerebral Cortex, 6, 498–505.

Andersen, R. A., Essick, G. K., & Siegel, R. M. (1985). Encoding of spatial location by posterior parietal neurons. Science, 230, 456–458.

Arakaki, T., Barello, G., & Ahmadian, Y. (2019). Inferring neural circuit structure from datasets of heterogeneous tuning curves. PLoS Computational Biology, 15, e1006816–.

Atick, J. J. & Redlich, A. N. (1990). Towards a Theory of Early Visual Processing. Neural Computation, 2, 308–320.

Babadi, B. & Sompolinsky, H. (2014). Sparseness and Expansion in Sensory Representations. Neuron, 83, 1213–1226.

Barak, O., Rigotti, M., & Fusi, S. (2013). The sparseness of mixed selectivity neurons controls the generalization-discrimination trade-off. Journal of Neuroscience, 33, 3844–3856.

Baraniuk, R., Davenport, M., DeVore, R., & Wakin, M. (2008). A simple proof of the restricted isometry property for random matrices. Constructive Approximation, 28, 253–263.

Baraniuk, R. G. & Wakin, M. B. (2009). Random projections of smooth manifolds. Foundations of Computational Mathematics, 9, 51–77.

Barlow, H. B. (1961). Possible Principles Underlying the Transformations of Sensory Messages. Sensory Communication, pp. 216–234.

Berens, P., Ecker, A. S., Gerwinn, S., Tolias, A. S., & Bethge, M. (2011). Reassessing optimal neural population codes with neurometric functions. Proceedings of the National Academy of Sciences of the United States of America, 108, 4423–4428.

Berry, M. J., Lebois, F., Ziskind, A., & da Silveira, R. A. (2019). Functional diversity in the retina improves the population code. Neural Computation, 31.

Bethge, M., Rotermund, D., & Pawelzik, K. (2002). Optimal short-term population coding: When Fisher information fails. Neural Computation, 14, 2317–2351.

Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2017). Julia: A fresh approach to numerical computing. SIAM Review.

Bordelon, B., Canatar, A., & Pehlevan, C. (2020). Spectrum Dependent Learning Curves in Kernel Regression and Wide Neural Networks. In International Conference of Machine Learning (ICML).

Bordelon, B., Paulson, J. A., & Pehlevan, C. (2021). Population Codes Enable Learning from Few Examples By Shaping Inductive Bias. bioRxiv.

Bremmer, F., Ilg, U. J., Thiele, A., Distler, C., & Hoffmann, K. P. (1997). Eye position effects in monkey cortex. I. Visual and pursuit-related activity in extrastriate areas MT and MST. Journal of Neurophysiology, 77.

Brunel, N. & Nadal, J. P. (1998). Mutual Information, Fisher Information, and Population Coding. Neural Computation, 10, 1731–1757.

Burak, Y. (2014). Spatial coding and attractor dynamics of grid cells in the entorhinal cortex. Current Opinion in Neurobiology, 25, 169–175.

Candes, E. J. & Tao, T. (2006). Near-optimal signal recovery from random projections: Universal encoding strategies? IEEE Transactions on Information Theory.

Carandini, M. & Heeger, D. J. (2012). Normalization as a canonical neural computation. Nature Reviews Neuroscience.

Cunningham, J. P. & Yu, B. M. (2014). Dimensionality reduction for large-scale neural recordings. Nature Neuroscience, 17, 1500–1509.

da Silveira, R. A. & Rieke, F. (2021). The Geometry of Information Coding in Correlated Neural Populations. Annu. Rev. Neurosci., pp. 1–30.

Dayan, P. & Abbott, L. F. (2001). Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems. (MIT Press).

Deneve, S., Latham, P. E., & Pouget, A. (1999). Reading population codes: A neural implementation of ideal observers. Nature Neuroscience, 2, 740–745.

Doeller, C. F., Barry, C., & Burgess, N. (2010). Evidence for grid cells in a human memory network. Nature, 463.

Doi, E., Gauthier, J. L., Field, G. D., Shlens, J., Sher, A., Greschner, M., Machado, T. A., Jepson, L. H., Mathieson, K., Gunning, D. E., Litke, A. M., Paninski, L., Chichilnisky, E. J., & Simoncelli, E. P. (2012). Efficient coding of spatial information in the primate retina. Journal of Neuroscience, 32, 16256–16264.

Donoho, D. L. (2006). Compressed sensing. IEEE Transactions on Information Theory.

Eliav, T., Maimon, S. R., Aljadeff, J., Tsodyks, M., Ginosar, G., Las, L., & Ulanovsky, N. (2021). Multiscale representation of very large environments in the hippocampus of flying bats. Science, 372.

Fiete, I. R., Burak, Y., & Brookings, T. (2008). What grid cells convey about rat location. Journal of Neuroscience, 28, 6858–6871.

Finkelstein, A., Ulanovsky, N., Tsodyks, M., & Aljadeff, J. (2018). Optimal dynamic coding by mixed-dimensionality neurons in the head-direction system of bats. Nature Communications, 9.

Fiscella, M., Franke, F., Farrow, K., Müller, J., Roska, B., da Silveira, R. A., & Hierlemann, A. (2015). Visual coding with a population of direction-selective neurons. Journal of Neurophysiology, 114, 2485–2499.

Fusi, S., Miller, E. K., & Rigotti, M. (2016). Why neurons mix: High dimensionality for higher cognition. Current Opinion in Neurobiology, 37, 66–74.

Gallego, J. A., Perich, M. G., Miller, L. E., & Solla, S. A. (2017). Neural Manifolds for the Control of Movement. Neuron, 94, 978–984.

Ganguli, D. & Simoncelli, E. P. (2014). Efficient sensory encoding and Bayesian inference with heterogeneous neural populations. Neural Computation.

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

Gaucher, Q., Panniello, M., Ivanov, A. Z., Dahmen, J. C., King, A. J., & Walker, K. M. (2020). Complexity of frequency receptive fields predicts tonotopic variability across species. eLife, 9.

Georgopoulos, A. P., Kalaska, J. F., Caminiti, R., & Massey, J. T. (1982). On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex. Journal of Neuroscience, 2, 1527–1537.

Hafting, T., Fyhn, M., Molden, S., Moser, M. B., & Moser, E. I. (2005). Microstructure of a spatial map in the entorhinal cortex. Nature.

Harel, Y. & Meir, R. (2020). Optimal multivariate tuning with neuron-level and population-level energy constraints.

Hubel, D. H. & Wiesel, T. N. (1959). Receptive fields of single neurones in the cat's striate cortex. The Journal of Physiology.

Kadia, S. C. & Wang, X. (2003). Spectral integration in A1 of awake primates: Neurons with single- and multipeaked tuning characteristics. Journal of Neurophysiology, 89.

Kayaert, G., Biederman, I., Op De Beeck, H. P., & Vogels, R. (2005). Tuning for shape dimensions in macaque inferior temporal cortex. European Journal of Neuroscience, 22.

Kettner, R. E., Schwartz, A. B., & Georgopoulos, A. P. (1988). Primate motor cortex and free arm movements to visual targets in three-dimensional space. III. Positional gradients and population coding of movement direction from various movement origins. Journal of Neuroscience, 8.

Killian, N. J., Jutras, M. J., & Buffalo, E. A. (2012). A map of visual space in the primate entorhinal cortex. Nature.

Kim, J. H. J., Fiete, I., & Schwab, D. J. (2020). Superlinear Precision and Memory in Simple Population Codes. pp. 1–5.

Kobak, D., Pardo-Vazquez, J. L., Valente, M., Machens, C. K., & Renart, A. (2019). State-dependent geometry of population activity in rat auditory cortex. eLife, 8, 1–27.

Kouh, M. & Poggio, T. (2008). A canonical neural circuit for cortical nonlinear operations. Neural Computation.

Lahiri, S., Gao, P., & Ganguli, S. (2016). Random projections of random manifolds. pp. 1–45.

Lalazar, H., Abbott, L. F., & Vaadia, E. (2016). Tuning Curves for Arm Posture Control in Motor Cortex Are Consistent with Random Connectivity. PLoS Computational Biology, 12, 1–27.

Lewicki, M. S. (2002). Efficient coding of natural sounds. Nature Neuroscience, 5.

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

Mathis, A., Herz, A. V., & Stemmler, M. B. (2012). Resolution of nested neuronal representations can be exponential in the number of neurons. Physical Review Letters, 109, 1–5.

Miller, J. P., Jacobs, G. A., & Theunissen, F. E. (1991). Representation of sensory information in the cricket cercal sensory system. I. Response properties of the primary interneurons. Journal of Neurophysiology, 66.

Montemurro, M. A. & Panzeri, S. (2006). Optimal tuning widths in population coding of periodic variables. Neural Computation, 18, 1555–1576.

Rasmussen, C. E. (2004). Gaussian Processes in machine learning. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics).

Saxena, S. & Cunningham, J. P. (2019). Towards the neural population doctrine. Current Opinion in Neurobiology, 55, 103–111.

Seung, H. S. & Lee, D. D. (2000). The manifold ways of perception.

Seung, H. S. & Sompolinsky, H. (1993). Simple models for reading neuronal population codes. Proceedings of the National Academy of Sciences of the United States of America, 90, 10749–10753.

Shamir, M. & Sompolinsky, H. (2006). Implications of neuronal diversity on population coding. Neural Computation, 18, 1951–1986.

Shannon, C. E. (1949). Communication in the Presence of Noise. Proceedings of the IRE, 37, 10–21.

Sofroniew, N. J., Vlasov, Y. A., Hires, S. A., Freeman, J., & Svoboda, K. (2015). Neural coding in barrel cortex during whisker-guided locomotion. eLife, 4.

Sreenivasan, S. & Fiete, I. (2011). Grid cells generate an analog error-correcting code for singularly precise neural computation. Nature Neuroscience, 14, 1330–1337.

SRJ & Everitt, B. S. (1999). The Cambridge Dictionary of Statistics. Journal of the American Statistical Association.

Stringer, C., Michaelos, M., & Pachitariu, M. (2019). High precision coding in visual cortex. High precision coding in mouse visual cortex, p. 679324.

Taube, J. S., Muller, R. U., & Ranck, J. B. (1990). Head-direction cells recorded from the postsubiculum in freely moving rats. I. Description and quantitative analysis. Journal of Neuroscience.

Van Hateren, J. H. & Ruderman, D. L. (1998). Independent component analysis of natural image sequences yields spatio-temporal filters similar to simple cells in primary visual cortex. Proceedings of the Royal Society B: Biological Sciences, 265.

Wang, W., Chan, S. S., Heldman, D. A., & Moran, D. W. (2007). Motor cortical representation of position and velocity during reaching. Journal of Neurophysiology, 97, 4258–4270.

Wang, Z., Stocker, A., & Lee, D. (2016). Efficient neural codes that minimize Lp reconstruction error. Neural Computation, 28.

Wei, X. X., Prentice, J., & Balasubramanian, V. (2015). A principle of economy predicts the functional architecture of grid cells. eLife, 4, 1–29.

Wei, X. X. & Stocker, A. A. (2012). Efficient coding provides a direct link between prior and likelihood in perceptual Bayesian inference. Advances in Neural Information Processing Systems, 2, 1304–1312.

Welinder, P. E., Burak, Y., & Fiete, I. R. (2008). Grid cells: The position code, neural network models of activity, and the problem of learning.

Wilke, S. D. & Eurich, C. W. (2002). Representational accuracy of stochastic neural populations. Neural Computation, 14, 155–189.

Yaeli, S. & Meir, R. (2010). Error-based analysis of optimal tuning functions explains phenomena observed in sensory neurons. Frontiers in Computational Neuroscience, 4, 1–16.

Yartsev, M. M., Witter, M. P., & Ulanovsky, N. (2011). Grid cells without theta oscillations in the entorhinal cortex of bats. Nature, 479.

Yerxa, T. E., Kee, E., DeWeese, M. R., & Cooper, E. A. (2020). Efficient sensory coding of multidimensional stimuli. PLoS computational biology, 16, e1008146.

Zhang, K. & Sejnowski, T. J. (1999). Neuronal tuning: To sharpen or broaden? Neural Computation, 11, 75–84.

Zhaoping, L. (2014). Understanding Vision: Theory, Models, and Data. Perception, 17.