Encoding and decoding of information through efficient neural representations

Simone Blanco Malerba

PSL
UNIVERSITÉ PARIS

THÈSE DE DOCTORAT

DE L'UNIVERSITÉ PSL

Préparée à l'Ecole Normale Supérieure de Paris

Encoding and decoding of information through efficient neural representations

Soutenue par

Simone BLANCO MALERBA

Le 19 décembre 2022

École doctorale n°564

Physique en Île-de-France

Spécialité

Physique

Composition du jury :

Jean-Pierre NADAL Directeur de Recherche, Ecole Normale Supérieure	*Président du jury*
Julijana GJORGJIEVA Professor, Technische Universität München	*Rapportrice*
Gergő ORBÁN Group Leader, Wigner Research Centre	*Rapporteur*
Nicolas BRUNEL Professor, Duke University	*Examinateur*
Angelika STEGER Professor, ETH Zürich	*Examinatrice*
Rava AZEREDO DA SILVEIRA Directeur de Recherche, Ecole Normale Supérieure	*Directeur de thèse*

ENS | PSL

Contents

Acknowledgments

First, I have to thank my supervisor, Rava da Silveira, for giving me the opportunity to pursue a PhD, and at the same time making me realize, back then, that it was what I really wanted to do. I am thankful for the long afternoons spent in the office, or in a cafè, discussing the more technical mathematical details and the broad scientific questions which made this thesis work possible, but also, and more important, I am grateful for the human bound developed throughout the thesis. Among all things, I really appreciated the efforts spent in encouraging me not to rely solely on my intuitions, despite its genuine ability to really understand them, forcing me to be precise and avoid vagueness and, consequently, to do better science. These four years were a dense experience both from the scientific and human point of view, and I will fondly look at this period.

I acknowledge Yoram Burak, for providing useful feedbacks and helping us with his strong intuitions and dedication; it was an honor to work with such a bright mind. Similarly, I would like to thank Mirko Pieropan, for helping me to deal with senior collaborators and for the wonderful time spent at Cosyne. On the other hand, Aurora Micheli made me realize how difficult the role of a supervisor is; I thank her for working with so much passion on something that was just a very fuzzy idea in my mind and I really look forward to seeing her path evolve, both as a scientist and as an hockey player.

I will always be in debt with Martine Ben-Amar, who recommended me to Rava and, during the years at ENS, became a true friend. My gratitude also go to Vincent Hakim, Remi Monasson, Jean-Francois Allemand and Frederic Chevy for their support during these years. A big thanks goes to Ulisse Ferrari and to my first office mates, especially to Lorenzo Posani, Marco Molari and Kevin Berlemont, who gave me some useful tips that greatly facilitated my workflow and allowed me to pass all the pitfalls of a PhD unscathed. ENS was an unique and lively environment, and I am thankful to all the colleagues that shared the path with me. A special mention must go to my sister-in-PhD, and friend, Trang-Anh Nghiem, for being always there to discuss about science and about life and finding always the time and the will to help me. The pandemic didn't exactly facilitated travels, but I always felt more than welcome in Basel: the merit is largely attributed to the members of the group, Luc Stebens, Luke Ewig, Hoke Wallace, Gilles Kuhorn and Liidia Nadporozhskaia , for all the animated lab meetings, and to Tanyia Channa, for all the logistic support.

When my day at the lab finished, I always knew I would have find a pleasant environment at home, and the merit goes to my past and present roommates, Vanessa, Lara, Alessia, Denitza, Giancarlo and Roger. My year in Turin would not have been feasible without the friends of a life, Andrea, Matteo, Gianluca, Mirko, Federico and Marco. Also, I am immensely grateful to Francesca for our discussions. I thanksmy parents and my brother for always supporting me, even if I repeated failed to explain them what I do. Finally, I thank

Giulia for her unconditional support in all the aspects of my life, with this thesis being no exception.

General Introduction

The brain is a fascinating example of complex system. Several individual cells, the neurons, interact among themselves by mean of electrical signals, giving rise to a rich variety of behaviors, with the ultimate goal of allowing the organism to survive in a likewise complex environment. The interaction with such environment plays a central role: neurons encode and process information about the external world, and they do so with remarkable precision and efficiency, despite the high level of noise which characterize all biological systems. The mechanisms that allow such computations have been subject of interest for scientists since at least one hundred years. In the last decades, with the development of large-scale data recording techniques, we started to get a more systematic understanding of collective behaviors in neural populations, and we began to 'crack' the neural code. The contemporary explosive growth of the field of artificial intelligence gave further rise to a mutual exchange of ideas, pushing both fields towards the current state of the art while also pointing out fundamental differences between artificial and biological intelligence.

Besides the analysis of experimental data, the study of the neural code largely benefited from more theoretical and normative approaches. The hypothesis that neural responses are organized so as to optimize some utility function provides a rationale for the first principles which govern information processing in the brain. Within this framework, simple theoretical models, inspired by empirical observations, allow to isolate and study computational principles which might feature optimal neural representations, at the same time abstracting away specific biological details. These models can be further extended and refined to take into account biological details, and used to interpret and analyze data by identifying signatures of optimality in the specific system.

The neural code can be studied from two, not neatly divided, perspectives. The *encoding* process refers to the way neurons modulate their responses to convey information about external stimuli, starting from the transduction of physical quantities into neurons' electrical activity, followed by the propagation of these signals across different areas of the brain. The *decoding* process, instead, refers to the inverse map, applied in order to recover relevant information about the stimulus from neural activity. This readout process can be performed by an external observer, i.e., the scientist during an experiment, but it must also be implemented by the organism itself, in order to perform an action or a choice in response to external inputs.

Throughout this thesis, we investigate how optimality criteria for encoding and decoding processes concur to shape neural representations of sensory stimuli. We derive coding properties of models of neural systems through analytical calculations and numerical simulations, in order to illustrate general computational principles. Then, we apply theoretical frameworks to the analysis of neural recordings data to validate models and provide concrete

examples of instantiation of such principles. Mathematically, we exploit tools from information theory, statistical physics and machine learning. In particular, information theory is the natural framework to study the problem of communication of signals, and it has been applied to neuroscience, in the pioneering studies of Barlow and Attneave, few years afters its formalization by Shannon. The framework of statistical physics is suited to describe how the interactions among neurons give rise to computations and cognitive functions, similarly to how macroscopic properties of the matter emerge from the complex interactions among individual particles. As for machine learning, we employ artificial neural networks to model the complexity and flexibility of biological systems, studying the patterns which emerge from the artificial learning process.

The encoding process can be characterized through the relationship between features of sensory stimuli and mean neural responses, the so-called 'tuning curve', along with the stochastic deviations from the mean activity, the 'neural noise'. In the first chapter, we investigate the coding properties of neurons which exhibit complex and irregular tuning curves. A salient example is offered by grid cells in enthorinal cortex, whose experimental discovery was awarded by the Nobel prize in 2014. Grid cells are periodically tuned to spatial coordinates, and their joint activity defines a population code conveying information about the position of the animal in the environment. The periodicity of grid cell tuning curves, as well as their functional organization in modules, imparts the population code with an exponentially large dynamic range, defined as the ratio between the range of represented stimuli and resolution. Recently, multiple other examples of neurons with complex, but unstructured tuning curves have been identified. These findings lead us to ask whether highly efficient neural codes require fine organization, as in grid cells, or whether they can be realized with more complex and irregular tuning curves. We approached this question with a benchmark model: a shallow neural network in which irregular tuning curves emerge due to random synaptic weights. The synapses project from a large population of sensory neurons with unimodal tuning curves in response to a one-dimensional stimulus onto a smaller population of 'representation' neurons. A trade-off is observed between two qualitatively different types of readout errors: 'local' errors whereby two nearby stimuli are confused, and 'global' errors, causing complete loss of information about the stimulus. When balancing the two error rates, we obtain an optimal solution in which a population code with irregular tuning curves achieves exponentially large dynamic range. We argue that compression balancing local and global errors takes place in the motor cortex, based on primate cortex recordings. Our results show that highly efficient codes do not require finely tuned response properties, and can emerge even in the presence of random synaptic connectivity.

The results of the first chapter, similarly to previous studies on population codes, are obtained by quantifying the coding performance through an abstract 'ideal' decoder, which has access to all the details of the encoding process and the statistics of the noise. In practice, however, the decoding process requires neural resources, and such ideal decoder might be hard to implement. In the second chapter, we address the problem of decoding the information conveyed by complex and irregular neural responses through a non-ideal neural architecture. We consider a supervised learning framework, in which the decoder learns the correct association between neural responses and stimuli from a limited number of examples. As we assume the decoder to be implemented in a downstream area of the brain, we model it as a flexible architecture, parametrized as a two-layer neural network. We first show that, by training the hidden layer to reproduce the correct posterior distribution over stimuli, we

obtain an approximation of the ideal decoder. We successively relax this strong assumption about the nature of the representation in the hidden layer, by training a decoder on the basis of the error of its final output. The gap between the ideal and non-ideal decoding performance depends on the complexity of the architecture: the number of neurons in the hidden layer, the non-linearity of their transfer function, and the regime in which it is trained. Simple decoders are not able to take advantage of the high (ideal) local accuracy achieved by irregular tuning curves, due to noise in the training examples. This results in a trade-off between the ideal accuracy of a population code and the neural resources necessary to extract the information. A non-ideal decoding process changes, in some cases dramatically, optimality criteria of a neural code.

In the third chapter, we consider the problem of acquiring efficient neural representations in unsupervised frameworks, through a joint optimization of the encoder and the decoder. It has been postulated that the brain maintains internal models of the environment, in which sensory stimuli are sampled from a distribution conditioned on a set of latent, elementary features of interest. This model is then 'inverted' during sensory perception, to infer the most probable features which gave rise to a given observation. We consider the optimization of an encoder and a decoder, under the assumption that the latter is set so as to maintain an internal model of the environment. Mathematically, such internal model is defined as a probabilistic generative model which maps latent features, represented by neural activity patterns and distributed according to some prior, to distributions over stimuli. To be optimal, the generative distribution must match the true distribution of stimuli in the environment. The optimization process is carried out by considering an encoder which performs the reverse operation, mapping observed stimuli to distributions over neural activity patterns: such system has the structure of a variational autoencoder. Formally, the proposed scheme implies that the encoder should be set so as to maximize a lower bound to the information conveyed by neural activity patterns, similarly to what postulated by the efficient coding hypothesis, under a constraint on neural resources. The latter can be interpreted as the metabolic cost of stimulus-evoked variations in the neural activity with respect to the spontaneous activity. We apply study a population coding model of noisy neurons with bell-shaped tuning curves within this framework. As a function of the resources constraint, we obtain different solutions which are characterized by equally satisfying generative models, but different arrangement of tuning curves, coding performance and statistics of spontaneous activity. We predict an optimal arrangement of coding resources as a function of the stimulus distribution. In weakly-constrained systems, such predictions are consistent with the ones obtained in previous studies, while we observe different behaviors in highly-constrained systems, depending on the interaction between the encoding and decoding distribution and the stimulus distribution. We combine two normative assumptions about the relation between stimuli and neural activity patterns to derive a family of efficient neural representations, which also yield a statistically optimal internal model of the environment.

Chapter 1

Random Compressed Coding

The results of this Chapter have been submitted to a peer-review journal and they are available as a preprint [1].

1.1 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 [2, 3, 4, 5, 6, 7]. 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 [8, 9, 10]. 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 [11, 12, 13].

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 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 monotonic [14, 15, 16, 17, 18], or bell-shaped (e.g., Gaussian) [11, 19, 20, 21, 22] functions, and obtain the values of their parameters that minimize the 'perceptual' error committed when information is decoded from the activity of a population of model neurons. 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 [23]. 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 bell-shaped tuning curves can be further optimized to yield an additional factor of $1/\sqrt{N}$; the error then scales like $1/N$ [24, 25].)

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 [26, 27, 28, 29]; their tuning curves in two-dimensional, open field environments, are multimodal and periodic as a function of spatial coordinates. 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 includes several modules made up of cells with different periodicities [30, 31], grid cells can represent spatial location with an accuracy that scales exponentially (rather than algebraically, as above) in the number of neurons [32, 33, 34]. Thus, the richer structure of individual tuning curves can be traded for a strong boost in the efficiency of the population code. Recent observations showed that place cells can also exhibit complex tuning curves in the context of motion in three dimensions, with multiple place fields that are irregular both in location and in size [35]. In addition, [36, 37] found that during motion in three dimensional space, individual grid cells also exhibit irregular firing fields. A number of other examples of neurons with complex, but unstructured, tuning curves has also been identified in other cortical regions and in different species [38, 39, 40, 41].

Here, we ask whether highly efficient codes 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 which 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 onto 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 this population, even in the presence of high-variance 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 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 layer 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 a framework that includes a downstream ('representation') layer of neurons as well as a peripheral ('sensory') layer of neurons. 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.

1.2 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. Seventh, we give a quantitative description of the geometry of the population response induced by our network as a function of its parameters, 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.

1.2.1 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 on neural population activity; 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 Gaussian tuning curves and three neurons with periodic (grid-cell-like) tuning curves with three different periods (Fig. 1.1A). 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 [42, 32].

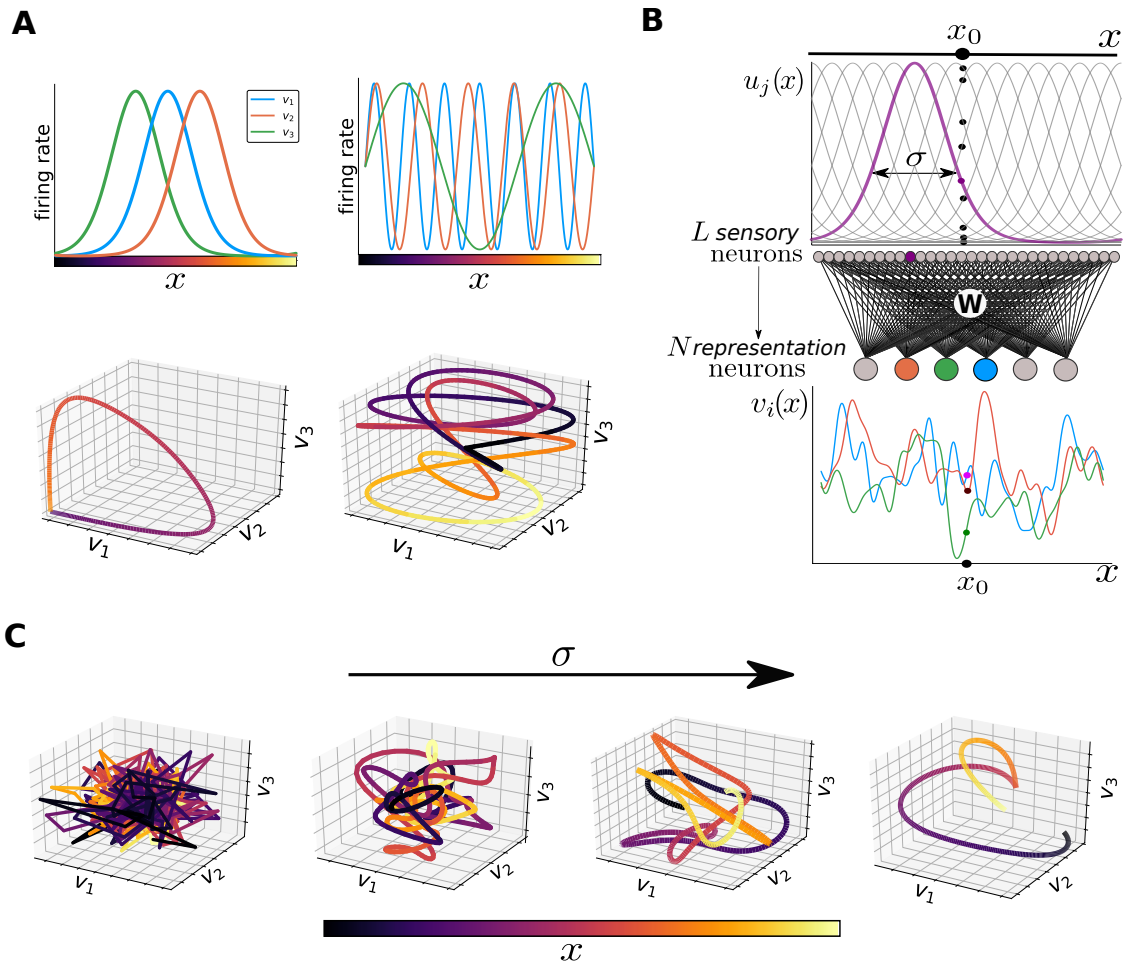

Fig. 1.1 ***Geometrical approach to coding, and the random feedforward neural network architecture.*** continue to next page

Fig. 1.1 *(previous page)* (***A***) *Top: mean responses of three-neuron populations encoding a one-dimensional stimulus. Left: population of neurons with Gaussian tuning curves. Right: population of neurons with periodic tuning curves. Bottom: mean activity in the neural populations, parametrized by the stimulus value colored according to the legend, as one-dimensional curves in a three-dimensional space. Unimodal tuning curves (left) evoke a single-loop curve, which preserves the distances between stimuli in the evoked responses. Periodic tuning curves (right) evoke a more complex curve in which two distant stimuli may be mapped to nearby points in the joint-activity space; the curve is longer, and fills up a larger portion of the 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 a L-dimensional representation. These tuning curves determine the mean 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 plot the tuning curves of three sample neurons, highlighting their response to the stimulus x_0.* (***C***) *Examples of population activity (across the stimulus line, color indicates stimulus value) for three sample representation neurons, for increasing values of σ. When $\sigma \to 0$ (left, $\sigma = 0.001$), neurons produce 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. Ultimately, for large values of σ (right, $\sigma = 0.15$) we recover a scenario similar to that with unimodal tuning curves.*

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.

1.2.2 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. 1.1B. 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 activity 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:

$$u_j(x) = A \exp\left(-\frac{(x-c_j)^2}{2\sigma^2}\right). \tag{1.1}$$

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 activity of neuron i in the second layer is thus given by

$$v_i(x) = \sum_{j=1}^{L} W_{ij} u_j(x). \tag{1.2}$$

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,

$$\left\langle \int_0^1 dx \left[v_i(x) - \int_0^1 dx' v_i(x') \right]^2 \right\rangle_W = R, \tag{1.3}$$

by calibrating the value of the prefactor in Eq. (1.1), 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 (1.3) 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 [9, 43, 44, 45].

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. 1.1C). 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. (1.3). 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 the activity of neuron i in the second layer is affected by 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 already.)

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, 'ideal' in the sense that it minimizes the MSE. (Throughout, in heuristic arguments and analytical calculations, we focus on the MSE. In a number figures, however, we plot its square root, the RMSE, so as to allow for a direct comparison with the stimulus range. The figure captions specify which of the two quantities is illustrated.) 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).

1.2.3 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. 1.1C). 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

$$v_i(x_j) = \tilde{A} W_{ij} \sim \mathcal{N}(0, R) \tag{1.4}$$

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. (1.3) but for the case of discrete stimuli). Geometrically, Eq. (1.4) 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. 1.2A for a two-neuron population). In any given trial, however, the responses in the representation layer are corrupted by noise (Fig. 1.2A). The ideal decoder interprets a single-trial response as being elicited by the stimulus associated to the nearest possible mean response (Fig. 1.2A). 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. 1.2A and Eq. (1.4)), errors

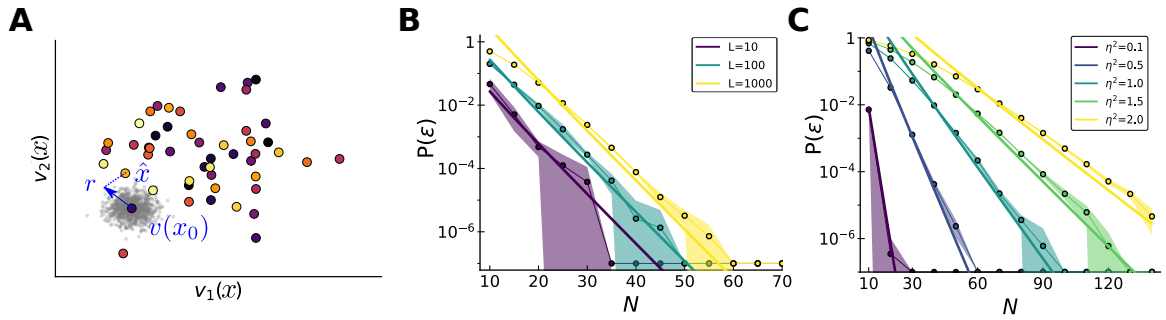

Fig. 1.2 ***Probability of error for narrow tuning curves in the sensory layer.*** *(**A**) Joint mean responses of two neurons to $L = 50$ stimuli, colored according to the legend in Fig. 1.1C. Noise is represented as a cloud of possible responses (in grey) around the mean. An error occurs when the noisy response, $\mathbf{r}$, falls closer to a mean response corresponding to a stimulus, $\hat{x}$, different from the true one, x_0. Since mean responses are uncorrelated, $\hat{x}$ may be distant from x_0. (**B**) Theoretical (solid curves, Eq. (1.5)) and numerical (dots) results for the probability of error as a function of the population size, for different values of L ($\eta^2 = 0.5$). The probability of error scales exponentially with the number of neurons, N, with a multiplicative constant involving the number of stimuli, L. (**C**) Theoretical (solid curves) and numerical (dots) results for the probability of error as a function of the population size for different values of η^2 ($L = 500$).*

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.

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

$$P_{\text{error}} \approx \frac{L}{\sqrt{2\pi N}} \exp\left(-\log\left(1+\frac{R}{2\eta^2}\right)\frac{N}{2}\right). \tag{1.5}$$

The main dependence to note, here, is the exponentially strong suppression as a function of the number of neurons in the second layer (Fig. 1.2B). 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 stimulus 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. (1.3)). 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

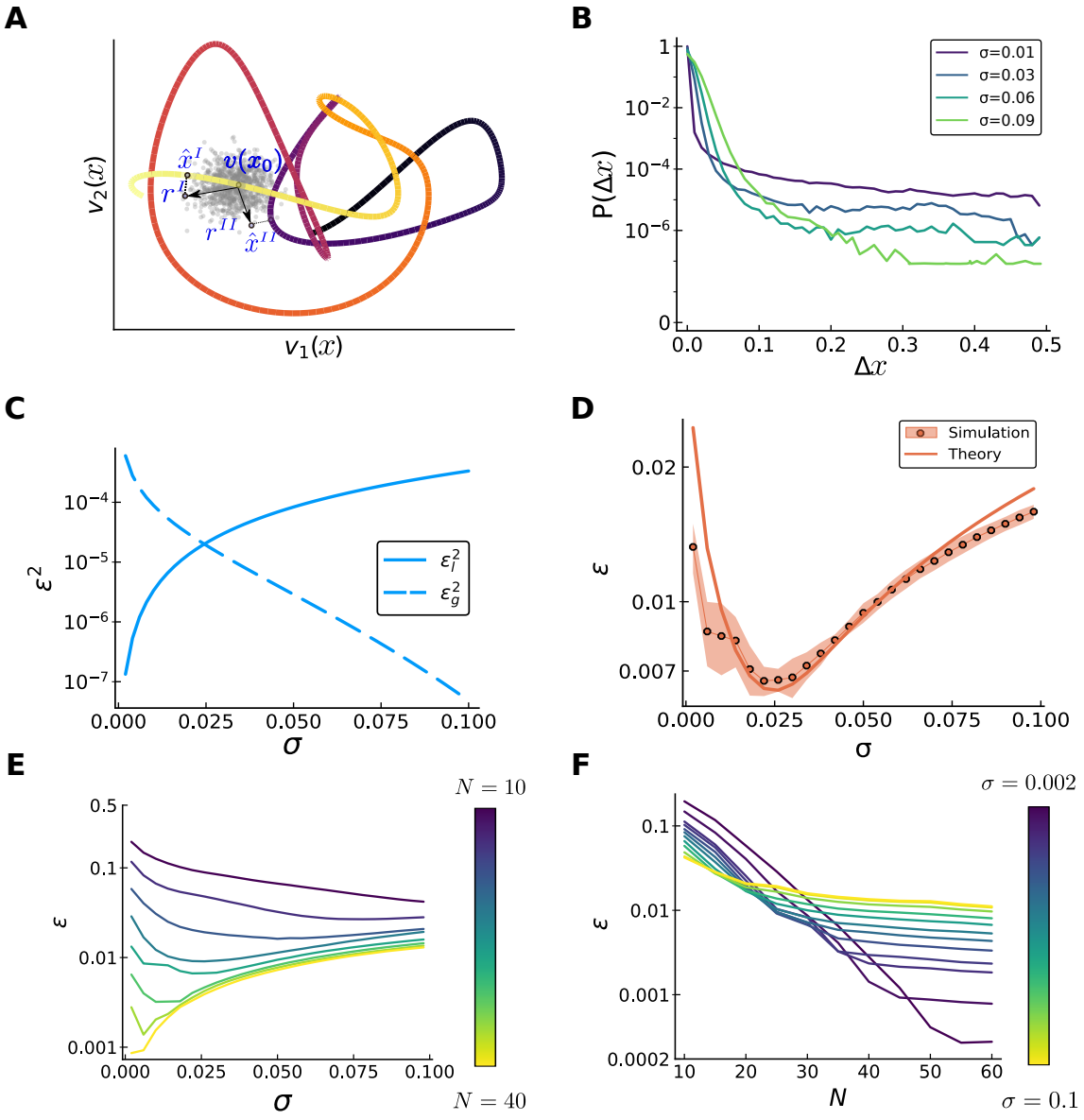

Fig. 1.3 ***Trade-off between local and global errors.*** continue to next page

small, i.e., when the noise in the activity of individual neurons is comparable to modulations of their mean responses, the exponential suppression as a function of N of the probability of error remains valid, with a rate approximately equal to $R/4\eta^2$.

1.2.4 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

Fig. 1.3 *(previous page)* (***A***) *Different types of error in an irregular curve of mean population activity (joint response of two neurons, colored according to the legend in Fig. 1.1C). 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 outputs the stimulus corresponding to the closest point on the curve. In one case,* $\mathbf{r}^I$ *results in a local error, by selecting a point on the curve that represents a nearby stimulus,* $\hat{x}^I$*. In the other case,* $\mathbf{r}^{II}$ *is closer to a point on the curve which represents a stimulus distant from the true one,* $\hat{x}^{II}$*, causing a global error. (****B****) Normalized histogram of absolute error magnitudes,* $\Delta x = |\hat{x} - x|$*, made by an ideal decoder, for different values of* σ *(*$N = 25$*). For better visualization, we consider a stimulus with periodic boundary conditions. The contribution of the two types of error varies with* σ*. For small* σ*, coding is precise locally (fast drop of the purple curve for small errors), but many global errors occur (tail of the distribution is high). For large* σ *(green curves) local accuracy is poorer but global errors are suppressed. (****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* σ *(solid curve), while the number of global errors decreases (dashed curve). (****D****) RMSE as a function of* σ*: comparison between numerical simulations (dots) and theoretical prediction of Eq. (1.6) (solid curve). (****E****) RMSE, as a function of* σ *for different population sizes* N *(increasing from violet to yellow). The smallest RMSE occurs at an optimal value of* σ*,* $\sigma^*(N)$ *, which decreases with increasing* N*. (****F****) Same data, but the error is displayed as a function of* N*, for a fixed value of* σ*. The MSE decreases exponentially rapidly until global errors are suppressed, then the local errors are linearly reduced. A smaller value of* σ *implies a larger value of* N *at which the crossover occurs, as well as a smaller MSE at this crossover value.*

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. 1.3A). 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. 1.1C). 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. 1.3B). 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 calcu-

lation, 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 representing the mean responses to two stimuli distant by more than σ is random and independent of the stimulus values. Using these two assumptions, we obtained the MSE as a sum of two terms (see Methods for mathematical details) corresponding to local and global errors, as

$$\varepsilon^2 = \left\langle E^2 \right\rangle_W \approx \varepsilon_l^2 + \varepsilon_g^2 \approx \frac{2\sigma^2\eta^2}{RN} + \frac{\bar{\varepsilon}_g^2}{\sigma\sqrt{2\pi N}} \exp\left(-\log\left(1+\frac{R}{2\eta^2}\right)\frac{N}{2}\right), \tag{1.6}$$

where $\bar{\varepsilon}_g^2$ 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. (1.6) represents the contribution of local errors, while the second term corresponds to global errors (Fig. 1.3C). Their form can be intuited as follows. The magnitude of local errors is proportional to η^2 and inversely proportional to N, as in classical models of population coding with neurons with bell-shaped tuning curves (see, e.g., [11]). 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. (1.6). (The form of this first term can also be understood as the inverse of the Fisher information [23, 46], which bounds the variance of an unbiased stimulus estimator.) The second term on the right-hand-side of Eq. (1.6) is obtained as an extension of Eq. (1.5): 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. (1.5) is replaced by a factor $1/\sigma$ in Eq. (1.6). Importantly, the two terms in Eq. (1.6) are modulated differently by the two parameters N and σ. Depending upon their values, either local or global errors dominate (Fig. 1.3C).

We tested the validity of Eq. (1.6): 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. 1.3D, see Methods for details). 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. 1.3E). Furthermore, the strong (exponential) reduction in MSE with increasing N occurs only up to a crossover value that depends on σ (Fig. 1.3F); 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. 1.3D 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. (1.6), we obtain the dependence of the optimal width upon the population size, as

$$\sigma^* \approx \left(\frac{\bar{\varepsilon}_g^2}{4\eta^2}\sqrt{\frac{N}{2\pi}}\right)^{1/3} \exp\left(-\log\left(1+\frac{R}{2\eta^2}\right)\frac{N}{6}\right), \tag{1.7}$$

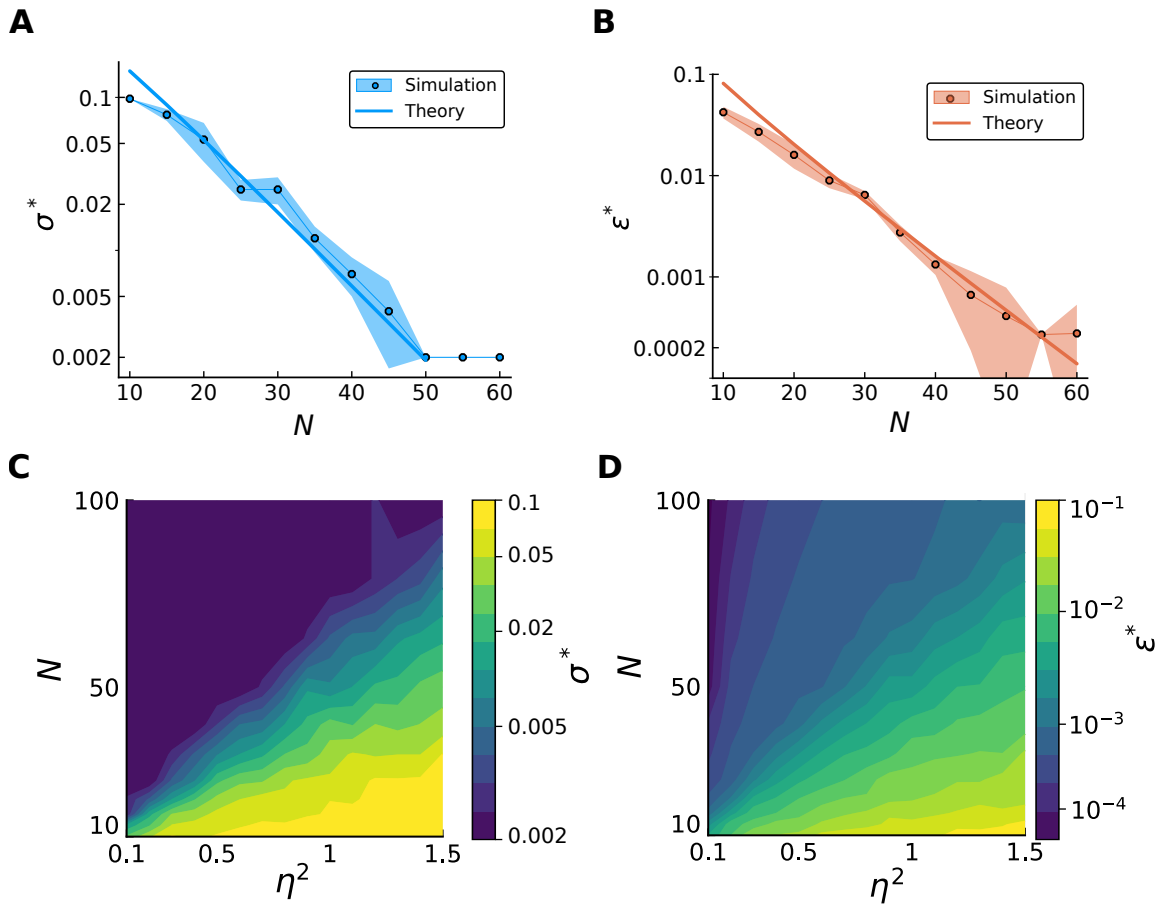

Fig. 1.4 ***Scaling of the optimal width and the optimal MSE as a function of population size and signal-to-noise ratio.*** *(**A**) The optimal σ^* decreases exponentially rapidly with the number of representation neurons, saturating the lower bound imposed by the finite number of neurons of the first layer (corresponding to the spacing of the preferred positions, $1/L$). Simulations (dots) show good agreement with analytical results (solid curve). (**B**) The optimal RMSE is suppressed exponentially rapidly with N. Simulations (dots) agree with analytical results (solid curve). (**C,D**) Optimal width (**C**) and RMSE (**D**) as a function of the parameters N and η^2. The color coding is in log scale, in order to highlight the exponential scaling.*

and the optimal MSE as a function of N, as

$$\varepsilon^{2*} = \langle E^2(\sigma^*)\rangle_W \approx \left(\frac{\eta\bar{\varepsilon}_g^2}{\sqrt{2\pi N}}\right)^{2/3} \exp\left(-\log\left(1+\frac{R}{2\eta^2}\right)\frac{N}{3}\right). \tag{1.8}$$

Both these analytical results agree closely with numerical simulations (Figs. 1.4A and B). Equation (1.8) and Fig. 1.4B 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: furthermore, a degree of smoothness in the tuning of the sensory neurons is advantageous. With the optimal choice of the sensory tuning width, the rate of scaling with N of the argument within the exponential in Eq. (1.8) depends upon the noise variance, η^2; in Figs. 1.4C and D, we illustrate the dependence of σ^* and ε^* upon N and η^2.

1.2.5 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. [47] 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. 1.1 for the case of a one-dimensional stimulus), we can analyze differences in coding properties between pure and conjunctive coding schemes; in Fig. 1.5A, 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. Neurons in the representation layer compute a linear sum of these responses, and therefore their activity can be decomposed as a sum of one-dimensional functions. This implies that the 'response sheet' is maximally curved along each of the stimulus dimensions; geometrically, this results in a 'folded' struc-

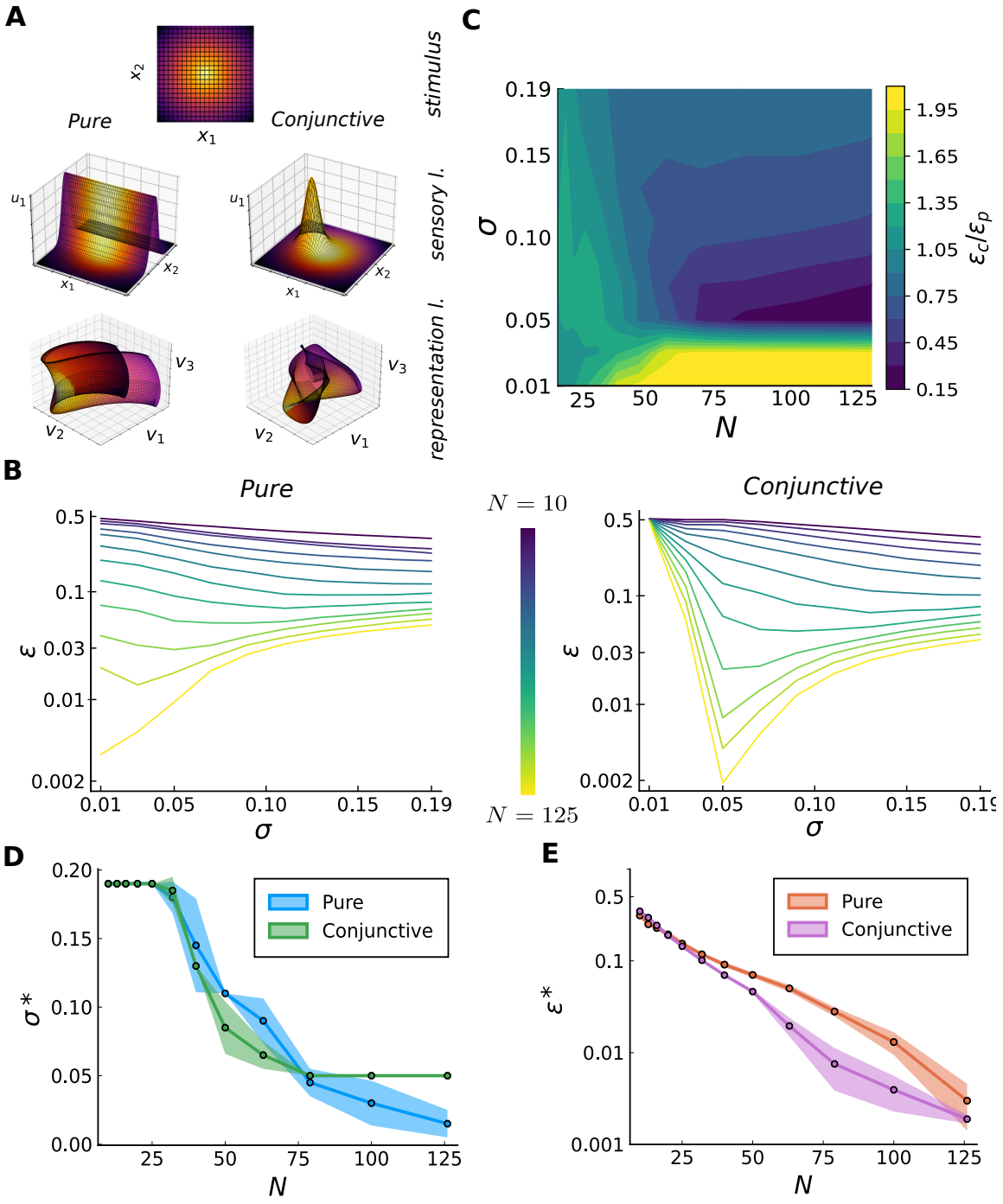

Fig. 1.5 ***Compressed coding with multi-dimensional stimuli.*** *We illustrate the case of a three-dimensional stimulus, with* $L = 3375$, $\eta^2 = 1$, $R = 1$. continue to next page

Fig. 1.5 *(previous page)* (***A***) *Mapping of a multi-dimensional stimulus space into neural population activity as obtained from a two-layer coding scheme. Top: two-dimensional stimulus space; colors serve as stimulus legend for subsequent plots. Middle: mean activity (z-axis) of a sample sensory neuron, for two cases, as a function of the two stimulus coordinates (x- and y- axis). In the pure case (left), a single sensory neuron 'folds' 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. Each of these neurons randomly sum the output 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. The optimal σ, which decreases 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 RMSE below $\sigma = 0.05$ is due to the sensory neurons not tiling the stimulus space, and it is independent of N.* (***C***) *Ratio of the RMSE in the two cases, $\varepsilon_c/\varepsilon_p$, as a function of σ and N. The 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 σ is characterized by a better coverage of the pure population, independently of N. Values greater than one 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 coverage of the stimulus space, the conjunctive case outperforms the pure case. This effect is stronger in the small-σ region, due to the slower scaling of the global errors in the pure case. When σ is large, the ratio saturates at the value given by the ratio of the local errors.* (***D,E***) *Optimal tuning width (**D**) and relative RMSE (**E**), for pure (blue, red) and conjunctive (green, violet) cases. The global error decreases more slowly in the pure case. For $N \gtrsim 75$ the optimal width in the conjunctive case saturates, due to loss of stimulus coverage, while the pure population does not suffer from this limitation. Thus, the RMSE in the conjunctive case stops decreasing exponentially and starts decreasing only linearly with N.*

ture, with creases along the 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 comes with (random) curvature equally along all stimulus dimensions: rather than 'folded', it behaves like a 'crumpled' sheet (Fig. 1.5A).

This geometric picture offers an intuitive explanation of the behavior of the MSE 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 a more 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. (1.69) and (1.74)). The behavior of the global error is also different in the two coding schemes; there are two mechanisms at play, here. First, in the pure scheme, for most realizations of the random tuning curves, global errors occur primarily in a single stimulus dimension (see Methods for mathematical details); this is also apparent in Fig. 1.5A: the 'folded' structure of the response sheet induces global errors in a single stimulus dimension. By contrast, in the conjunctive scheme global errors occur in an arbitrary number of stimulus 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 error suppression with N scales differently as a function of K. Both mechanisms, in a regime in which N is large enough to suppress the contribution of global errors, enhance the probability of global error in the pure scheme as compared to the conjunctive scheme (compare Eq. (1.79) and Eq. (1.81) in Methods). Intuitively, this is because a folded sheet has a larger surface area of contact with itself than a crumpled sheet. Thus, for sufficiently large values of N, the conjunctive scheme is more favorable than the pure one. The corresponding crossover value of N, however, depends on K, and large values of K impose a stringent constraint in the conjunctive case.

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. 1.5B, 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. 1.5C). 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 (not because of a different scaling with N, but instead because of a difference in the prefactors that affect the probability of error in the two cases); 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 relative to global errors (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 ratio between local errors only (Eq. (1.75) for $K = 3$, implies $\varepsilon_{l,c}/\varepsilon_{1,p} \approx 1/\sqrt{3}$). 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. 1.5B, 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. 1.5D), and contributes to the form of the suppression of the RMSE in the two schemes (Fig. 1.5E). 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

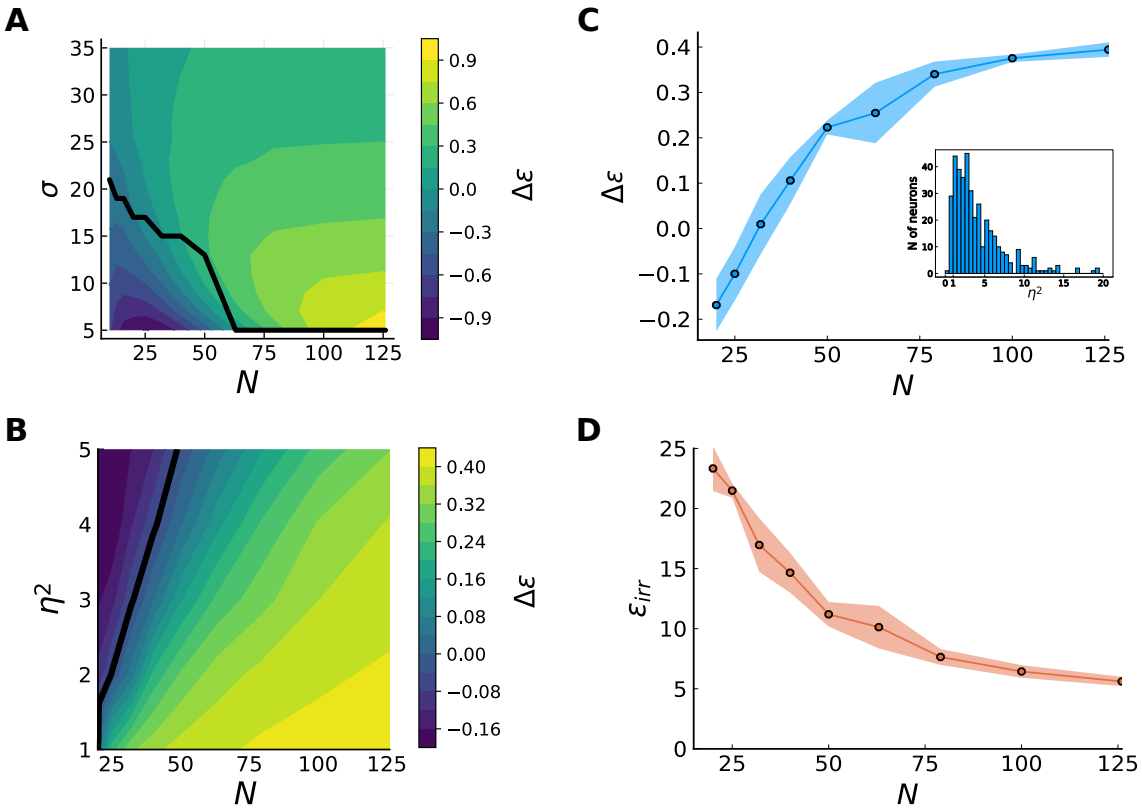

Fig. 1.6 ***Irregular vs. linear tuning.*** continue to next page

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.

1.2.6 Compressed coding in monkey motor cortex

The activity of neurons in the primary motor cortex (M1) of monkey is correlated with the location and movement of the limbs. Here, we consider spatial tuning in the context of a 'static task' [48]. 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 in M1 [48, 49], and it has been customary to model these as a linear projection of the hand position onto a so-called 'preferred vector' or 'positional gradient', $\mathbf{p}$, which thus points in the direction of maximal sensitivity [49]. The tuning curve of neuron i is then written as

$$v_i(\mathbf{x}) = a_i + p_{1,i}x_1 + p_{2,i}x_2 + p_{3,i}x_3 = a_i + \mathbf{p}_i \cdot \mathbf{x}. \tag{1.9}$$

A recent study [40] 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. (1.9). This more elaborate model 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

Fig. 1.6 *(previous page)* (***A***) *Mean fractional improvement for irregular tuning as compared to linear tuning, as a function of population size and tuning-curve width. The black line indicates the critical values of N and σ at which the two coding schemes perform equally well. In the region below (violet), global errors penalize the irregular case, making a smoother code more efficient. With increasing N, global errors become rarer while irregularities improve the local accuracy of the code (yellow region). This advantage increases at smaller values of σ, but so does the value of N required for the irregular case to be advantageous.* (***B***) *Mean fractional improvement in the irregular case, generated with the data-fitted model, compared to the linear one, as a function of N and noise variance, η^2. At small population sizes, irregular tuning curves produce global errors, 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 fractional improvement* (***C***) *and RMSE* (***D***) *in the irregular case as a function of population size, for the noise model extracted from data. A noise variance, η^2, is assigned to each neuron according to the distribution extracted from the data, showed in the inset of panel* (***C***). *For small N, linear tuning yields a better coding performance. 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 $N_{local} \sim 100$, when global errors are fully suppressed; the scaling of the error as a function of the population size is no longer exponential, but only hyperbolic.*

may afford 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 [50]. This population of neurons projects onto a smaller population of M1 neurons which display spatially extended and irregular tuning profiles. In fitting our model to recordings from M1 neurons [40], 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 [40, 51]: 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 statistic of the data referred to as *complexity measure* (a discrete version of the Lipschitz derivative that quantifies the degree of smoothness of a curve, see Methods and [40]). The complexity measure varies from neuron to neuron, and we chose σ so as to minimize the Kolmogorov-Smirnov distance (see Eq. (1.103) in Methods) 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 [40], it yields comparable fit.

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, $\varepsilon^2_{\rm irr}$, and comparing the latter with the corresponding quantity in a coding scheme with the smooth tuning curves defined in Eq. (1.9), $\varepsilon^2_{\rm lin}$. We plot our results in terms of the 'mean fractional improvement', $\Delta\varepsilon \equiv (\varepsilon_{\rm lin} - \varepsilon_{\rm irr})/\varepsilon_{\rm lin}$. $\Delta\varepsilon$ 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. 1.6A). We note the existence of a crossover value of N, N^*, defined as the population size at which $\Delta\varepsilon = 0$ and irregular and linear tuning curves yield the same coding performances. 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 1.6B 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 data (Figs. 1.6C and D). 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. (1.3)). 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. (1.104) in Methods). As in simulations we set the variance of the signal for each neuron to a constant value, we modeled the heterogeneity in the signal-to-noise ratio as a heterogeneous noise variance; the resulting distribution is skewed, with an appreciable fraction of neurons exhibiting low signal-to-noise ratios (Fig. 1.6C, inset). For each value of N, we sampled eight different pools of N neurons from the population, and we averaged the corresponding mean fractional 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. 1.6C), the error magnitude is still appreciable (Fig. 1.6D). This means that even though local and global errors are balanced, both contributions are substantial. $\Delta\varepsilon$ continues to grow with N until global errors are suppressed; beyond this second crossover value, $N_{\rm local}$, $\Delta\varepsilon$ 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 $N_{\rm local}$. When $N < N_{\rm local}$ the MSE decreases exponentially with N, due to the suppression of global errors, while when $N > N_{\rm local}$, the suppression of the MSE is hyperbolic in N, reflecting the behavior of local errors only (Fig. 1.6D). This second crossover occurs at $N_{\rm local} \approx 100$, a figure comparable to the number of neurons that control individual muscles in this specific task, as estimated from decoding EMG signals corresponding to individual muscles from subsets of M1 neurons [40].

1.2.7 Dimensionality of a compressed neural code

We discussed a geometrical interpretation of a neural population code in terms of a map from a set of stimuli to a set of points in the space of (mean) population activity. With smooth tuning curves, a continuous K-dimensional stimulus is represented as a K-dimensional hypersurface embedded in the N-dimensional space of neural activity. This hypersurface is often referred to as a 'neural response manifold' [52, 53] (which implicitly assumes a local homeomorphism to a Euclidean space). In the previous sections, we analyzed the way in which the geometrical properties of the response manifold affect the coding performance. In this section, we relate the picture put forth by our model to recent work that quantified the dimensionality of neural activity as a way to characterize its nature and to infer strategies used by the brain to represent (sensory) information [54, 55].

While a K-dimensional stimulus space may correspond to a K-dimensional neural response manifold, the latter's complicated geometry—as in our model—may make its identification difficult. In practice, one is faced with a data set, namely a noisy sample from a population of tuning curves, and from it one would like to make statements on the geometry of the population activity. Fitting a low-dimensional manifold to a neural population data set is not a trivial task, and is the focus of a large number of studies on dimensional reduction [56]. A simple approach is to consider the eigenvalue spectrum of the covariance matrix of the neural responses across the stimulus range or, equivalently, the variance carried by the the different modes in a principal component analysis (PCA). If we apply this approach to the population response in our model, for different values of σ, we find a spectrum that exhibits a band-pass structure, which plateaus up to a cut-off value before a sharp suppression; the cut-off value is larger for smaller values of σ (Fig. 1.7A). From this analysis one would conclude that the population activity occupies a low-dimensional subspace embedded in the N-dimensional space of neural activity, with dimensionality controlled by σ. As the value of σ falls to zero, the population responses fill an increasingly large fraction of the N available dimensions, until they fill the space entirely for $\sigma \to 0$.

A quantification of the 'intrinsic dimensionality' of the population activity based on this PCA analysis is offered by the participation ration, defined as $d = \left(\sum_{i=1}^{N} \lambda_i\right)^2 / \sum_{i=1}^{N} \lambda_i^2$, where λ_i denotes the ith eigenevalue of the covariance matrix of the neural responses across the stimulus range [57]. Loosely speaking, the participation ratio measures the number of eigenvalues (principal components) which are much larger than the others; for example, if M eigenvalues are of comparable size and much larger than any others, then $d \approx M$.

In our model, while d is close to unity for large values of σ, it becomes larger for smaller values of σ and approaches N when $\sigma \to 0$ (Fig. 1.7B). It is interesting to examine the behavior of this quantity in the vicinity of the optimal value of σ. In Fig. 1.7C, we display the fractional dimensionality (i.e., the participation ratio divided by the number of neurons, d/N) corresponding to the population activity at the optimal value of σ as a function of the population size, for a fixed level of noise. As expected, d increases with N: larger populations allow for more irregular tuning curves which benefit the local accuracy without generating prohibitive global errors. Quantitatively, the value of d hovers around $N/2$. A possible interpretation of this result is that it corresponds to the largest value beyond which a random manifold embedded in N dimensions comes close to intersect itself; thus, this value of d ensures that global errors do not proliferate. While a naive interpretation of the value of the participation ratio would suggest that the neural population encodes an $N/2$-dimensional

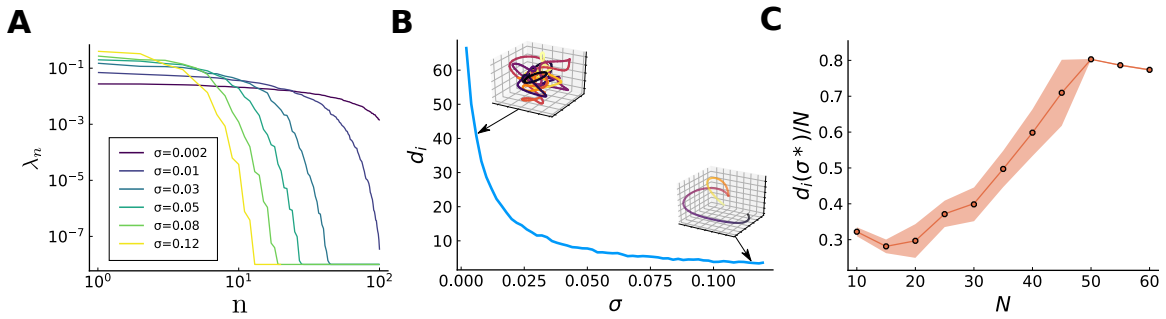

Fig. 1.7 ***Dimensionality of the neural code.*** *(**A**) Spectrum of the eigenvalues of the covariance matrix of neural responses in a population with $N = 100$ representation neurons, for different tuning widths (decreasing from violet to yellow). (**B**) Intrinsic dimensionality, defined as the participation ratio of the eigenvalues of the covariance matrix, in a population with $N = 100$ representation neurons, as a function of σ. Insets exhibit a typical response manifold in a three-dimensional space. (**C**) Ratio of the intrinsic dimensionality at the optimal value of σ and population size, as a function of population size, for the networks illustrated in Fig. 1.3,1.4.*

stimulus, in the context of our model it results from the efficient coding of a one-dimensional stimulus. This points to the difficulty of using a simple criterion to define the dimensionality of a manifold when the latter is highly non-linear.

1.2.8 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 onto 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 noise in representation neurons, 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 has a negligible effect on the coding performance. The presence of sensory noise degrades coding, so a comparison of 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 covariance matrix (i.e., independent noise) with elements chosen as follows. Given the distribution of synaptic weights, the matrix $\mathbf{W}\mathbf{W}^\mathbf{T}$ is sampled from a Wishart distribution with mean given by the identity matrix and fluctuations of order $1/L$ (see Methods); in the limit of $L \to \infty$, the

covariance matrix becomes

$$\mathbf{\Sigma}_{\text{ind}} \equiv (\eta^2 + \xi^2)\mathbf{I} = \tilde{\eta}^2\mathbf{I}, \tag{1.10}$$

i.e., the population noise becomes independent, with single-neuron variance $\tilde{\eta} = \eta^2 + \xi^2$. Hereafter, we compare the two cases of populations with covariance matrices $\mathbf{\Sigma}$ and $\mathbf{\Sigma}_{\text{ind}}$.

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.)

In general, local errors can be either suppressed or enhanced by correlated noise [58]. We can show analytically that in our model, if noise correlations are due to independent noise in the sensory layer, local errors are enhanced. By computing a correction to the diagonal behavior of the covariance matrix in the limit $L \to \infty$ through a perturbative expansion of the inverse covariance matrix to second order in $\xi^2/\tilde{\eta}^2$ (see Methods), we obtain the local contribution to the MSE as

$$\varepsilon_l^2 = \varepsilon_{l,\text{ind}}^2 \left(1 + \frac{N\xi^2}{L\tilde{\eta}^2} - \frac{N\xi^4}{L\tilde{\eta}^4} + \dots\right), \tag{1.11}$$

where $\varepsilon_{l,\text{ind}}^2$ is the corresponding quantity calculated for the matrix $\mathbf{\Sigma}_{\text{ind}}$ rather than the full covariance matrix $\mathbf{\Sigma}$. From Eq. (1.11), it appears that the effect of noise correlations on the MSE is deleterious, but scales proportionally to the ratio between the two population sizes, which we supposed to be small. We checked this behavior numerically (Fig. 1.8A), 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. (1.11) and Fig. 1.8B indicate that there exists a regime in which increasing the relative contribution of input noise, ξ^2, in fact mitigates the deleterious effect of the correlated noise (this is seen in Eq. (1.11) as a partial cancellation of the second- and fourth-order terms).

Finally, we ask whether the impact of the noise correlations 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, $\mathbf{\Sigma}_{\text{rand}} = \eta^2\mathbf{I} + \xi^2\mathbf{X}\mathbf{X}^\mathbf{T}$, where $X_{ij} \sim \mathcal{N}(0, 1/L)$. In this case, noise correlations *suppress* the MSE as compared to the independent case (with $\mathbf{\Sigma}_{\text{ind}}$), because the 'cloud of possible noisy responses' is reoriented randomly with respect to the curve of mean responses. Analytically, the analog of Eq. (1.11) for the case of a covariance matrix $\mathbf{\Sigma}_{\text{rand}}$ (instead of $\mathbf{\Sigma}$) is similar, but skips the lowest-order, deleterious term:

$$\varepsilon_{l,\text{rand}}^2 \approx \varepsilon_{l,\text{ind}}^2 \left(1 - \frac{N\xi^4}{L\tilde{\eta}^4}\right). \tag{1.12}$$

This result, as well as numerical simulations (Fig. 1.8B), demonstrates that generically coding is improved by random noise correlations, and that this improvement increases with

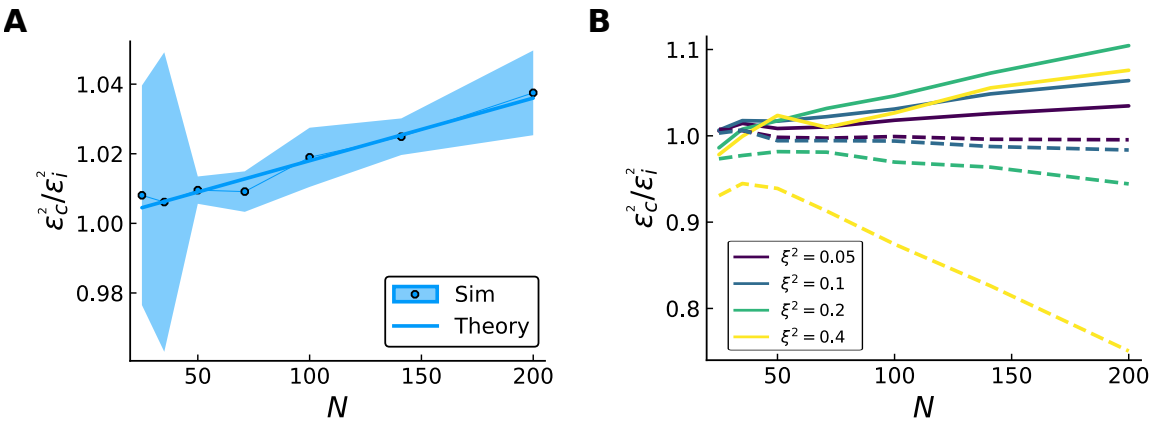

Fig. 1.8 ***Effects of input and correlated noise on compressed coding.*** *(**A**) Ratio of MSE in the presence of correlated noise due to input noise and independent noise of variance $\tilde{\eta}^2$, as a function of N, theoretical prediction (solid curve, Eq. (1.11)) and numerical simulations (dots) with $\sigma = 0.045$, $\tilde{\eta}^2 = 0.5$ and small contribution of input noise, $\xi^2 = 0.05$. (**B**) Ratio of MSE in the presence of correlated noise due to input noise and independent noise of variance $\tilde{\eta}^2$ (solid curves) and ratio of MSE in the presence of correlated noise with random covariance matrix and independent noise of variance $\tilde{\eta}^2$ (dashed curves) Different colors denote different contributions coming from the off-diagonal terms ξ^2, increasing from violet to yellow, and $\tilde{\eta}^2 = 0.5$. When correlations come from input noise, the ratio is positive (detrimental noise correlations). Their effect is non-linear in $\xi^2/\tilde{\eta}^2$, due to the competition between the first-(positive) and second-order (negative) corrections. With a random covariance matrix, correlations enhance coding precision.*

N and also increases with the relative contribution of ξ^2 with respect to η^2. In sum, noise correlations in representation neurons are deleterious if they are inherited from independent noise in sensory neurons—yet, the effect is quantitatively modest.

1.3 Discussion

Summary. We analyzed the properties of a neural population encoding a one-dimensional, continuous stimulus by means of irregular tuning curves, which emerge in a neural circuit with random synaptic weights. This 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. At optimality the mean error is suppressed exponentially with population size; as a result, irregular neural codes allow to compress the representation of a low-dimensional, continuous stimulus from a large, first layer of neurons to a small, second layer. We extended these results to the case of multi-dimensional, continuous 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 [47, 59], 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 [40] 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 relating our picture to analysis of the dimensionality of population activity.

'Exponentially strong' neural population codes. Our results on the exponential scaling of the mean error with population size are similar to results obtained in the context of the representation of position by grid cells [30, 32, 33, 31]. 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.

The notion of an exponentially strong code predates work in computational neuroscience: Shannon introduced it in the context of communication systems and analog signals [60]. 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 transmitted over a noisy channel and 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. [60]), analogous to the mapping illustrated in Fig. 1.1C; 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. 1.2): he demonstrated that it is possible to send a discrete set of messages, with an error suppressed exponentially in the dimensionality of the signal. Our work proposes an extension of this 'fully random' scenario for the representation of a continuous variable based on a smooth, but irregular, mapping in a higher dimensional encoding space. 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 [11], stimulus geometry [61], and the time scale on which coding operates [12, 20]. 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 [62, 21, 63]. A separate direction of

study focused on the effects of heterogeneity in the tuning-curve parameters on the coding performance [64, 65, 22, 66].

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 [67, 68, 69]. 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 [70].

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 [71]. 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 precise identity of the stimulus cannot be recovered in presence of noise, and what matters is the magnitude of the distance between the decoded stimulus and the true one, quantified by an appropriate metric. In other words, 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*), in many 'efficient coding' models, optimality criteria in a neural population are derived under constraints on the activity of the same population. Our results differ in that they are obtained in a downstream ('representation') neural population, subject to constraints on a upstream ('sensory') population.

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 [56, 70]. 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 population responses to a battery of stimuli [54, 53, 72, 73]. 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 we illustrate the dependence of the coding properties of a neural population on the geometry of the representation, which in turns depends on the tuning properties of a presynaptic population [74].

A specific geometrical question is that of the dimensionality of the population response in the representation layer. 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. [72] 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 for 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), as, in this limit, global errors proliferate. The optimal effective dimensionality of the response manifold, as defined by the participation ratio, lies at an intermediate value at which intersections of the manifold with itself are rare and local and global errors are balanced (Fig. 1.7).

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 [75]. 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 N linear, noisy measurements, with $K \ll L$ and $N \ll L$ [76]. 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 layer are not too wide.

A central result in the field of compressed sensing is that random measurements can yield near-optimal reconstructions. Furthermore, for near optimality to be achieved, the required number of measurements scales approximately linearly in K and only logarithmically in the dimensionality of the signal: $N > \mathcal{O}\left(K \log\left(L/K\right)\right)$ [75, 77]. 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. (1.5): 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. We focused in this study exclusively on the properties of encoding in a neural population. For this aim, throughout we assumed an ideal decoder; in principle, this is not a limitation: we show in Methods that an ideal decoder can be implemented by 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 [19, 78, 79]. 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 that approximates the ideal one. A closely related question has been examined by [80], 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 structure of the decomposition of a target function into a set of modes (e.g., Fourier modes). [81] find that learning the high-frequency Fourier components of a target function requires a larger number of training samples, as compared to learning its 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. Noise in the training

samples may also hamper learning severely in the presence of global errors. Furthermore, one can ask how our results might be modified if 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 these questions for future work.

1.4 Methods

Throughout, we denote vectors by bold letters, e.g., $\mathbf{r} = (r_1, r_2, ..., r_N)$, and the L_2 norm as $\|\mathbf{r}\|_2^2 = \sum_i r_i^2$. Capital bold letters, e.g., $\mathbf{W}$, refer to matrices. We denote the derivative of a function as $f'(x) = \partial f/\partial x$.

1.4.1 Network model

Network model for one-dimensional stimuli and constraints on its parameters. We consider a two-layer feedforward network. The first, sensory layer is made up of L neurons, each responding to a continuous scalar stimulus, $x \in [0, 1]$, according to a Gaussian tuning curve. The mean activity of neuron j in response to a stimulus, x, is given by

$$u_j(x) = A \exp\left(-\frac{(x - c_j)^2}{2\sigma^2}\right), \tag{1.13}$$

where c_j is the preferred stimulus of neuron j, σ is the tuning-curve width, and A is a fixed response amplitude. The preferred stimuli are evenly spaced, $c_j = j/L$. Each neuron in the first layer projects onto all N neurons in the second, representation layer. The transfer function is assumed to be linear, and the random synaptic weights are independent realizations of a Gaussian random variable, $W_{ij} \sim \mathcal{N}(0, 1/L)$; hence, the mean activity of representation neuron i can be written as

$$v_i(x) = \sum_{j=1}^{L} W_{ij} u_j(x). \tag{1.14}$$

The value of the amplitude, A, is chosen so as to set the 'dynamic range' of representation neurons to a fixed value; more precisely, we choose the value of A so that the variance of each neuron's response across the stimulus range is invariant under variations in the other parameter of the model, σ, on average over network realizations. This quantity is calculated as

$$\begin{aligned} R &= \left\langle \int_0^1 dx \left[v_i(x) - \int_0^1 dx' v_i(x') \right]^2 \right\rangle_W \\ &= \left\langle \int_0^1 dx v_i(x)^2 - \left(\int_0^1 dx v_i(x) \right)^2 \right\rangle_W \\ &= \left\langle \sum_{j=1,j'=1}^{L} W_{ij} W_{ij'} \left[\left(\int_0^1 dx u_j(x) u_{j'}(x) \right) - \left(\int_0^1 dx u_j(x) \right) \left(\int_0^1 dx u_{j'}(x) \right) \right] \right\rangle_W \\ &= \int_0^1 dx u_j(x)^2 - \left(\int_0^1 dx u_j(x) \right)^2, \end{aligned} \tag{1.15}$$

where $\langle \cdot \rangle_W$ indicates an average over the distribution of synaptic weights. Here (and below), we approximate Gaussian integrals on a bounded domain as

$$\int_0^1 dx u_j(x) \approx \int_{-\infty}^{\infty} dx u_j(x) = A\sqrt{2\pi\sigma^2}; \tag{1.16}$$

this approximation is valid when σ is small with respect to the stimulus range and c_j is separated from the boundaries (0 and 1) by a distance that exceeds σ. As we will consider a

large number of neurons in the sensory layer and relatively small values of σ (up to 1/10th of the stimulus range), errors introduced by this approximation will be negligible. By inserting Eq. (1.16) and a similar approximation for $\int_0^1 dx u_j(x)^2$ into Eq. (1.3), we obtain A as a function of σ, as

$$A^2 = \frac{R}{\sqrt{\pi\sigma^2} - 2\pi\sigma^2}. \tag{1.17}$$

Tuning curves as samples from a Gaussian process. The response of each neuron in the second layer to a stimulus, x, is a sum of realizations of Gaussian random variables; as a result, it is also a realization of a Gaussian random variable, with mean

$$\langle v_i(x)\rangle_W = \sum_{j=1}^{L} \langle W_{ij}\rangle_W u_j(x) = 0, \tag{1.18}$$

and its covariance is calculated as

$$\begin{aligned}
\left\langle v_i(x)v_i(x')\right\rangle_W &= \sum_{j,j'=1}^{L} \left\langle W_{ij}W_{ij'}\right\rangle_W u_j(x)u_{j'}(x') \\
&= \sum_{j=1}^{L} \frac{1}{L} u_j(x)u_j(x') \\
&\approx A^2 \int_0^1 dc_j \exp\left(-\frac{\left((x-c_j)^2 + (x'-c_j)^2\right)}{2\sigma^2}\right) \\
&\approx A^2\sqrt{\pi\sigma^2}\exp\left(-\frac{\Delta x^2}{4\sigma^2}\right),
\end{aligned} \tag{1.19}$$

where $\Delta x = x - x'$. The first approximation is obtained by replacing a sum by an integral $\sum_{j=1}^{L} \frac{1}{L} f(c_j) \approx \int_0^1 f(c_j)dc_j$ and the second approximation consists in extending the integration domain to the entire real line. The first approximation is valid if the spacing between the centers is small relatively to the width of the Gaussian, that is $L\sigma \gg 1$, while the second is valid if the arithmetic mean of x and x' is far from the stimulus boundaries. According to Eqs. (1.18) and (1.19) each neuron's tuning curve can be viewed as a sample from a one-dimensional Gaussian process with vanishing mean and Gaussian kernel with standard deviation equal to $\sqrt{2}\sigma$ [82].

Network model for multi-dimensional stimuli. We denote by K the stimulus dimensionality, such that the stimulus is a K-dimensional vector, $\mathbf{x} = \{x_1, x_2, ..., x_K\}$. Analogously to the one-dimensional case, each stimulus dimension can assume values in a bounded interval, $x_k \in [0, 1]$. We consider the two cases of pure and conjunctive tuning for sensory neurons. In both cases, the sensory layer is made up of L neurons, which project onto all N representation neurons. Similarly to the one-dimensional case, synaptic weights are independent realizations of a Gaussian random variable, $W_{ij} \sim \mathcal{N}(0, 1/L)$.

Sensory neurons with pure tuning. The L neurons are divided in K sub-populations of

$Q = L/K$ neurons. Neurons in the sub-population k are sensitive to the single stimulus dimension x_k. The mean activity of neuron j assigned to sub-population k is given by the one-dimensional Gaussian

$$u^p_{j,k}(\mathbf{x}) = u^p_{j,k}(x_k) = A_p \exp\left(-\frac{\left(x_k - c^k_j\right)^2}{2\sigma^2}\right), \tag{1.20}$$

with preferred stimuli evenly spaced, $c^k_j = j/Q$ for $j = 1, ..., Q$. The mean activity of representation neuron i can be written as a superposition of one-dimensional tuning curves,

$$\begin{aligned} v^p_i(\mathbf{x}) &= \sum_{k=1}^{K}\sum_{j=1}^{Q} W_{ij,k} u_{j,k}(\mathbf{x}) \\ &= \sum_{k=1}^{K} v^p_{i,k}(x_k). \end{aligned} \tag{1.21}$$

Imposing the resource constraint, Eq. (1.3), we obtain $A^2_p = R/\left((\pi\sigma^2)^{1/2} - 2\pi\sigma^2\right)$.

Sensory neurons with conjunctive tuning. Neurons are sensitive to all stimulus dimensions. The mean activity of sensory neuron j is given by the multi-dimensional Gaussian function

$$u^c_j(\mathbf{x}) = A_c \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_j\|^2_2}{2\sigma^2}\right), \tag{1.22}$$

with preferred stimuli, $\mathbf{c}_j$, arranged on a K-dimensional square grid with mesh size $L^{-1/K}$. The mean activity of representation neuron i is obtained as

$$v^c_i(\mathbf{x}) = \sum_{j=1}^{L} W_{ij} u^c_j(\mathbf{x}). \tag{1.23}$$

Imposing the resource constraint, Eq. (1.3), we obtain $A^2_c = R/\left((\pi\sigma^2)^{K/2} - (2\pi\sigma^2)^K\right)$.

1.4.2 Population coding and optimal decoder

Noise Model. We assume that the response of representation neurons is corrupted by noise. The vector of responses to a given stimulus, x, is

$$\mathbf{r} = \mathbf{v}(x) + \mathbf{z}, \tag{1.24}$$

where $\mathbf{z}$ is a noise vector of independent Gaussian entries with vanishing mean and 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 of mean responses of second-layer neurons to a stimulus, x (see Eq. (1.2)). The probability density of a response vector, $\mathbf{r}$, given a stimulus, x, is written as

$$p(\mathbf{r}|x) = \frac{1}{(2\pi\eta^2)^{N/2}} \exp\left(-\frac{\|\mathbf{r} - \mathbf{v}(x)\|^2_2}{2\eta^2}\right). \tag{1.25}$$

Below, we will furthermore consider an extension that takes into account a generic noise covariance matrix, $\mathbf{\Sigma}$, resulting in the more general form

$$p\left(\mathbf{r}|x\right) = \frac{1}{\left(2\pi\right)^{N/2}\left[\det\left(\mathbf{\Sigma}\right)\right]^{1/2}} \exp\left(-\frac{1}{2}\left(\mathbf{r}-\mathbf{v}(x)\right)^{T}\mathbf{\Sigma}^{-1}\left(\mathbf{r}-\mathbf{v}(x)\right)\right). \tag{1.26}$$

Loss function and ideal decoder. We quantify the coding performance of the neural population by the mean squared error (MSE) in the stimulus estimate [6], as obtained from the ideal decoder, or estimator, $\hat{x} = f_{dec}(\mathbf{r})$, expressed as

$$E^2 = \int_0^1 dx \int d\mathbf{r} p(\mathbf{r}|x)\left(\hat{x}-x\right)^2, \tag{1.27}$$

where we have assumed a uniform prior over stimuli, $p(x) \sim \mathcal{U}(0,1)$. We consider the average of this quantity over network realizations, $\varepsilon^2 \equiv \langle E^2 \rangle_W$; in some figures, we plot the square root of this quantity, the RMSE, $\varepsilon \equiv \sqrt{\langle E^2 \rangle_W}$.

For multi-dimensional stimuli, the ideal decoder outputs a vector estimate of the stimulus, $\hat{\mathbf{x}} = f_{dec}(\mathbf{r})$. In this case, we define the MSE as the average squared norm of the difference between the stimulus and the decoder output,

$$E^2 = \int d\mathbf{x} \int d\mathbf{r} p(\mathbf{r}|\mathbf{x}) \left\|\hat{\mathbf{x}}-\mathbf{x}\right\|_2^2, \tag{1.28}$$

where $\left\|\hat{\mathbf{x}}-\mathbf{x}\right\|_2^2 = \sum_{k=1}^{K}(\hat{x}_k - x_k)^2$.

The estimator that minimizes the MSE (Minimum-MSE or MMSE) is given by the mean of the posterior density. We can write the optimal estimator as

$$\hat{x}_{MMSE} = \int_0^1 dx p(x|\mathbf{r}) x = \frac{\int_0^1 dx p(\mathbf{r}|x) x}{\int_0^1 dx p(\mathbf{r}|x)}. \tag{1.29}$$

We note that a simple neural network can output the MMSE estimate. Indeed, if we approximate the integrals in Eq. (1.29) with a discrete sum over M values and we use Eq. (1.25), we obtain

$$\begin{aligned}
\hat{x}_{MMSE} &\approx \frac{\sum_{m=1}^{M} x_m p\left(\mathbf{r}|x_m\right)\Delta x_m}{\sum_{m=1}^{M} p\left(\mathbf{r}|x_m\right)\Delta x_m} \\
&= \frac{\sum_{m=1}^{M} x_m \exp\left(-\frac{1}{2\eta^2}\left(\sum_{i=1}^{N} r_i^2 + \sum_{i=1}^{N} v_i^2(x_m) - 2\sum_{i=1}^{N} v_i(x_m) r_i\right)\right)}{\sum_{m=1}^{M} \exp\left(-\frac{1}{2\eta^2}\left(\sum_{i=1}^{N} r_i^2 + \sum_{i=1}^{N} v_i^2(x_m) - 2\sum_{i=1}^{N} v_i(x_m) r_i\right)\right)} \\
&= \frac{\sum_{m=1}^{M} x_m \exp\left(\frac{1}{2\eta^2}\left(\sum_{i=1}^{N} 2v_i(x_m) r_i - \sum_{i=1}^{N} v_i^2(x_m)\right)\right)}{\sum_{m=1}^{M} \exp\left(\frac{1}{2\eta^2}\left(\sum_{i=1}^{N} 2v_i(x_m) r_i - \sum_{i=1}^{N} v_i^2(x_m)\right)\right)} \\
&= \sum_{m=1}^{M} x_m \tilde{h}_m,
\end{aligned} \tag{1.30}$$

where the terms $\sum_i r_i^2$ in both numerator and denominator cancel and we assumed a constant spacing,

$\Delta x_m = \Delta x_0$. The approximate estimate specified by Eq. (1.30) can be by produced by a two-layer neural network: a first layer of M neurons, whose activities are given by

$$\tilde{h}_m = \frac{\exp\left(\sum_{i=1}^{N} \lambda_{mi} r_i + b_m\right)}{\sum_{m'=1}^{M} \exp\left(\sum_{i=1}^{N} \lambda_{m'i} r_i + b_{m'}\right)}, \tag{1.31}$$

computes a normalized, discrete approximation of the posterior, $\tilde{h}_m \approx p(x_m|\mathbf{r})$, 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 obtained as a linear combination of the activities of the representation neurons plus a bias term, transformed through an exponential non-linearity. The 'synaptic weight' from the ith representation neuron to the mth decoder neuron is a function of the true mean response of neuron i to stimulus x_m and of 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 output neuron weights the activity of these M neurons according to their 'preferred stimulus', x_m.

In what follows, we will also use the maximum a posteriori (MAP) estimator, defined as

$$\hat{x}_{MAP} = \arg\min_{x_m} \|\mathbf{r} - \mathbf{v}(x_m)\|_2^2 . \tag{1.32}$$

It is equal to the maximum likelihood (ML) estimator given the uniformity of the stimulus prior, and it has a simple geometric interpretation: it identifies the stimulus which corresponds to the vector of mean responses closest to the noisy population output. In numerical simulations, the MSEs calculated with the MMSE and the MAP estimators are very similar.

The MMSE estimator can be extended to the case of non-diagonal noise covariance matrix, $\mathbf{\Sigma}$, by combining Eqs. (1.26) and (1.29) . The decoder weights and biases are then correlated, $\lambda_m = \mathbf{v}^T(x_m)\mathbf{\Sigma}^{-1}$ and $b_m = \mathbf{v}^T(x_m)\mathbf{\Sigma}^{-1}\mathbf{v}(x_m)$, where λ_m denotes the vector with elements corresponding to the mth row of λ.

The MMSE estimator can also be extended to the case of multi-dimensional stimuli. In this case, the integrals of Eq. (1.29) are K-dimensional and the output layer is made up by K neurons, which compute a vector estimate of the stimulus, $\hat{\mathbf{x}}$.

Details of numerical simulations. In numerical simulations, we compute the MSE with standard Monte Carlo methods. At each step, we sample a stimulus, we generate a noisy population response and we decode it using the ideal decoder; the squared difference between the stimulus and its estimate is used to update the MSE. This process is repeated and the MSE estimate is updated until convergence, defined as the point for which the variance of the MSE estimates in the last 500 steps, after a burn-in period of 5000 steps, is less than a tolerance threshold, set to 10^{-8}. We set the number of decoder neurons equal to the number of sensory neurons, $M = L$, with uniformly spaced preferred stimuli, $x_m = m/M$. Unless otherwise stated, $L = 500$, $R = 1$ and $\eta^2 = 0.5$. The results are averaged over 8 network realizations and shaded regions corresponds to one s.d.

1.4.3 Analytical derivations

In the calculations that follow, we consider the limit of $L \to \infty$ and we assume $N \ll L$.

Narrow tuning curves. In the limiting case with $\sigma \to 0$, sensory neurons respond only to their preferred stimulus. Therefore, we consider the case of L discrete stimuli corresponding to the neurons' preferred stimuli, $x_j = j/L$. The mean activity of representation neuron i is written as $v_i(x_j) = \tilde{A}W_{ij}$, with $\tilde{A}^2 = LR$, such that $v_i(x_j) \sim \mathcal{N}(0, R)$. The constant of proportionality is computed with the analog of Eq. (1.3) for discrete stimuli, in the limit of large L.

The MSE in the case of narrow tuning curves, ε_n^2, is obtained as

$$\varepsilon_n^2 = \langle E^2 \rangle_W = \left\langle \frac{1}{L} \sum_{j=1}^{L} \int d\mathbf{r} p_e(\mathbf{r}, \mathbf{W}, x_j) \left(\hat{x}_j - x_j\right)^2 \right\rangle_W , \tag{1.33}$$

where $p_e(\mathbf{r}, \mathbf{W}, x_j)$ denotes the probability, given a synaptic matrix $\mathbf{W}$ and noise, of having an incorrect estimate of x_j, i.e., $\hat{x}_j \neq x_j$. For every choice of $\mathbf{W}$ and $\mathbf{r}$, there are $L-1$ equiprobable realizations of the synaptic matrix which correspond to permutations of the identity of the decoded stimulus, such that $\hat{x}_j = x_{j'}$ with $j' \neq j$. Therefore, the MSE can be written as

$$\varepsilon_n^2 = \langle P(E) \rangle_W \frac{1}{L(L-1)} \sum_{j=1}^{L} \sum_{j' \neq j} \left(x_j - x_{j'}\right)^2 , \tag{1.34}$$

where $\langle P(E) \rangle_W = \langle \int d\mathbf{r} p_e\left(\mathbf{r}, \mathbf{W}, x_j\right) \rangle_W$ is the probability of error averaged over the noise and the synaptic weights realizations. The MSE is the product of two terms: the mean probability of error and the average squared magnitude of the error. We now compute these two terms.

Error probability. An error occurs if there exists a j' such that $\mathbf{r}$ is closer to $\mathbf{v}(x_{j'})$ than to $\mathbf{v}(x_j)$, where x_j is the presented stimulus. We calculate the probability of error as a function of the probability of the complementary event, as

$$P(E) = 1 - P\left(\left\| \mathbf{r} - \mathbf{v}(x_{j'}) \right\|_2^2 > \left\| \mathbf{r} - \mathbf{v}(x_j) \right\|_2^2 \quad \forall j \neq j' \right) , \tag{1.35}$$

By averaging over different realizations of $\mathbf{W}$, $\langle P(E) \rangle_W$, the probabilities that an error is not committed on the possible values of j' are independent; thus, we can express the probability of error, as a function of the mean responses, v_i, and the noise, z_i, as

$$\begin{aligned} \langle P(E) \rangle_W &= 1 - \left(1 - \left\langle P\left(\left\| \mathbf{r} - \mathbf{v}(x_{j'}) \right\|_2^2 < \left\| \mathbf{r} - \mathbf{v}(x_j) \right\|_2^2 \right) \right\rangle_W \right)^{L-1} \\ &\approx L \left\langle P\left(\left\| \mathbf{r} - \mathbf{v}(x_{j'}) \right\|_2^2 < \left\| \mathbf{r} - \mathbf{v}(x_j) \right\|_2^2 \right) \right\rangle_W \\ &= L \left\langle P\left(\sum_{i=1}^{N} \left(v_i(x_j) - v_i(x_{j'})\right)^2 - \sum_{i=1}^{N} 2 \left(v_i(x_j) - v_i(x_{j'})\right) z_i < 0 \right) \right\rangle_W . \end{aligned} \tag{1.36}$$

The approximation comes from the assumption that the probability of error is small, and $L - 1 \approx L$ is large, while the last equality is obtained from the definition of noisy responses, Eq. (1.24). The difference between the mean activity of the same neuron to two different stimuli is sampled according to a Gaussian distribution, $\tilde{v}_i \equiv v_i(x_j) - v_i(x_{j'}) = \tilde{A}(W_{ij} - W_{ij'}) \sim \mathcal{N}(0, 2R)$. The mean probability of error is calculated as

$$\langle P(E)\rangle_W \approx L \int \prod_{i=1}^{N} d\tilde{v}_i \prod_{i=1}^{N} dz_i p(\tilde{v}_i) p(z_i) \Theta\left(-\sum_{i=1}^{N} \tilde{v}_i^2 + 2\sum_{i=1}^{N} \tilde{v}_i z_i\right). \tag{1.37}$$

This quantity is the probability that the random variable $\rho = \sum_{i=1}^{N} \tilde{v}_i^2 - \sum_{i=1}^{N} 2\tilde{v}_i z_i$, where $\tilde{v}_i \sim \mathcal{N}(0, 2R)$ and $z_i \sim \mathcal{N}(0, \eta^2)$, is negative. With $\zeta \equiv \sum_{i=1}^{N} \tilde{v}_i^2$, the distribution of ρ conditional on ζ is Gaussian with mean ζ and variance $4\zeta\eta^2$. Thus,

$$\begin{aligned} \langle P(E)\rangle_W &\approx L \int_0^\infty d\zeta p(\zeta) \int_{-\infty}^0 d\rho p(\rho|\zeta) \\ &= \frac{L}{2} \int_0^\infty d\zeta p(\zeta) \operatorname{erfc}\left(\sqrt{\frac{\zeta}{8\eta^2}}\right), \end{aligned} \tag{1.38}$$

where erfc is the complementray error function and

$$p(\zeta) = \frac{\left(\frac{\zeta}{2R}\right)^{N/2-1} \exp\left(-\frac{\zeta}{4R}\right)}{2^{N/2+1}\Gamma(N/2)} \tag{1.39}$$

is the probability density function of a chi-squared distribution. Computing the integral, we obtain

$$\begin{aligned} \langle P(E)\rangle_W &\approx L \frac{(\frac{\eta^2}{2R})^{\frac{N}{2}}\Gamma(N)}{\Gamma(\frac{N}{2})} {}_2\tilde{F}_1\left(\frac{N}{2}, \frac{1+N}{2}, \frac{2+N}{2}, -2\frac{\eta^2}{R}\right) \\ &= L \frac{\left(\frac{\eta^2}{2R}\right)^{\frac{N}{2}} \Gamma(N)}{\Gamma\left(\frac{N}{2}\right)\Gamma\left(\frac{2+N}{2}\right)} \sum_{n=0}^{\infty} \frac{\left(\frac{N}{2}\right)_n \left(\frac{N+1}{2}\right)_n}{\left(\frac{N+2}{2}\right)_n n!} \left(-2\frac{\eta^2}{R}\right)^n, \end{aligned} \tag{1.40}$$

where ${}_2\tilde{F}_1(a, b, c, x)$ is the regularized 2F1 Hypergeometric function; we provide its definition in the last equality. The Pochammer symbol can be defined through Gamma functions, $(x)_n = \frac{\Gamma(x+n)}{\Gamma(x)}$. By using the identity $\sum_{n=0}^{\infty} \frac{(x)_n}{n!} a^n = (1-a)^{-x}$ and the Stirling approximation for Gamma functions, we obtain the expression of the error probability that appears in the main text, Eq. (1.5):

$$\begin{aligned} \langle P(E)\rangle_W &\approx L \left(\frac{\eta^2}{2R}\right)^{\frac{N}{2}} \frac{\Gamma(N)}{\Gamma^2\left(\frac{N}{2}\right) \frac{N}{2} \left(1 + 2\frac{\eta^2}{R}\right)^{\frac{N+1}{2}}} \\ &\approx \frac{L}{\sqrt{2\pi N}} \exp\left(-\log\left(1 + \frac{R}{2\eta^2}\right)\frac{N}{2}\right). \end{aligned} \tag{1.41}$$

Average squared magnitude of error. We denote by $\bar{\varepsilon}_{n,g}^2$ the second factor in Eq. (1.34), which can be written as

$$\begin{aligned} \bar{\varepsilon}_{n,g}^2 &= \frac{1}{L(L-1)} \sum_{j=1}^{L} \sum_{j' \neq j}^{L} \left(\frac{j}{L} - \frac{j'}{L}\right)^2 \\ &= \frac{1}{L(L-1)} \left(\sum_{j=1}^{L} \sum_{j' \neq j} \frac{j^2}{L^2} + \sum_{j=1}^{L} \sum_{j' \neq j}^{L} \frac{j'^2}{L^2} - 2\sum_{j=1}^{L} \sum_{j'=1}^{L} \frac{jj'}{L^2}\right). \end{aligned} \tag{1.42}$$

These sums can be computed through the identities for the sum of the first n squared numbers,
$\sum_{i=1}^{n} i^2 = n(n+1)(2n+1)/6$, and for the sum of the first n numbers, $\sum_{i=1}^{n} i = n(n+1)/2$. The first two sums in Eq. (1.42) are identical, and yield

$$\sum_{j=1}^{L} \sum_{j' \neq j}^{L} \frac{j^2}{L^2} = (L-1) \frac{L(L+1)(2L+1)}{6L^2}, \tag{1.43}$$

while the last term is calculated as

$$\begin{aligned} \frac{1}{L^2} \sum_{j=1}^{L} j \sum_{j' \neq j}^{L} j' &= \frac{1}{L^2} \sum_{j=1}^{L} j \left(\frac{L(L-1)}{2} - j \right) \\ &= \frac{L^2(L-1)^2}{4L^2} - \frac{L(L+1)(2L+1)}{6L^2} \\ &= \frac{3L^3 - 10L^2 - 3L - 2}{12L}. \end{aligned} \tag{1.44}$$

Finally, combining Eqs. (1.43) and (1.44) into Eq. (1.42), we obtain

$$\begin{aligned} \bar{\varepsilon}^2_{n,g} &= \frac{(L+1)(2L+1)}{3L^2} - \frac{3L^3 - 10L^2 - 3L - 2}{6L^2(L-1)} \\ &= \frac{1}{6} \left(1 + \frac{12}{L-1} + \frac{1}{L} \right). \end{aligned} \tag{1.45}$$

This is a term of order 1, the size of the stimulus range, plus corrections of order $1/L$.

Broad tuning curves. In the case of broad tuning curves, we consider the regime of smooth response curves on the scale of the noise amplitude, such that the mean population activity can be approximated locally by a linear function of the stimulus. This regime obtains when the second-order term in the Taylor expansion is negligible with respect to the first-order one:

$$\frac{1}{4} \left\| \mathbf{v}''(x) \right\|_2^2 \Delta x^4 \ll \left\| \mathbf{v}'(x) \right\|_2^2 \Delta x^2, \tag{1.46}$$

where $\Delta x^2 \approx \eta^2$. In order to express this condition in terms of model parameters, we impose it on average over network realizations; we note that this leads to the same result as imposing the condition on average over stimuli, but it requires a simpler calculation. From the identity $\langle W_{ij} W_{ij'} \rangle_W = \frac{1}{L} \delta_{jj'}$, the average of the left-hand-side of Eq. (1.46) is obtained as

$$\begin{aligned} \left\langle \left\| \mathbf{v}''(x) \right\|_2^2 \right\rangle_W &= \left\langle \sum_{i=1}^{N} \sum_{j=1,j'=1}^{L} W_{ij} W_{ij'} u_j''(x) u_{j'}''(x) \right\rangle_W \\ &= \sum_{i=1}^{N} \sum_{j=1}^{L} \frac{1}{L} u_j''(x)^2 \\ &\approx \frac{3\sqrt{\pi} N A^2}{4\sigma^3}, \end{aligned} \tag{1.47}$$

where the approximations consists in replacing the sum with an integral and in extending the integration domain to the real line. A similar calculation can be performed for the right-hand-side of Eq. (1.46):

$$\begin{aligned}\left\langle \left\| \mathbf{v}'(x) \right\|_2^2 \right\rangle_W &= \sum_{i=1}^{N} \sum_{j=1}^{L} \frac{1}{L} u_j'(x)^2 \\ &\approx \frac{\sqrt{\pi} N A^2}{2\sigma}.\end{aligned} \tag{1.48}$$

By combining Eqs. (1.48) and (1.47), and substituting Δx^2 by the variance of the noise, η^2, in Eq. (1.46), we obtain the smoothness condition as

$$\frac{3\eta^2}{8\sigma^2} \ll 1. \tag{1.49}$$

In the case of broad tuning curves the error can be of two qualitatively different types: *local* or *global* (Fig. 1.3A). The width of the Gaussian kernel in Eq. (1.19) gives a measure of the distance in the stimulus space at which population responses are correlated; we refer to a global error when the distance between the stimulus and its estimate is greater than this 'correlation length', σ. We write the MSE as

$$\varepsilon^2 = \varepsilon_l^2 + \varepsilon_g^2, \tag{1.50}$$

and we compute these two terms.

Local error. According to the ML decoder, Eq. (1.32), the stimulus estimate corresponds to the value x' that minimizes the distance between $\mathbf{v}(x')$ and $\mathbf{r}$; if the error is local, this is obtained by projecting the noise vector onto the curve of mean population activity. By expanding the response curve around $\mathbf{v}(x)$, we obtain, to linear order,

$$\left\| \mathbf{z} \cdot \hat{\mathbf{v}}'(x) \right\|_2^2 \approx \left\| \mathbf{v}(x + \Delta x) - \mathbf{v}(x) \right\|_2^2 \approx \left\| \mathbf{v}'(x) \right\|_2^2 \Delta x^2, \tag{1.51}$$

where $\hat{\mathbf{v}}'(x) = \mathbf{v}'(x) / \left\| \mathbf{v}'(x) \right\|_2$. The local error can then be calculated as $\Delta x^2 = (\hat{x} - x)^2 \approx \left\| \mathbf{z} \cdot \hat{\mathbf{v}}'(x) \right\|_2^2 / \left\| \mathbf{v}'(x) \right\|_2^2$. By averaging over the noise and the synaptic weights, we obtain the mean local error as

$$\begin{aligned}\varepsilon_l^2 &= \left\langle \int_0^1 dx \int d\mathbf{z} p(\mathbf{z}) \frac{\left\| \mathbf{z} \cdot \hat{\mathbf{v}}'(x) \right\|_2^2}{\left\| \mathbf{v}'(x) \right\|_2^2} \right\rangle_W \\ &= \left\langle \int_0^1 dx \frac{\eta^2}{\left\| \mathbf{v}'(x) \right\|_2^2} \right\rangle_W .\end{aligned} \tag{1.52}$$

The squared norm of the derivative of the tuning curves is the realization of the random variable

$$\left\| \mathbf{v}'(x) \right\|_2^2 = \sum_{i=1}^{N} \left(\sum_{j=1}^{L} W_{ij} u_j'(x) \right)^2 . \tag{1.53}$$

The terms of the inner sum, $W_{ij} u_j'(x)$, are realizations of independent Gaussian random variables with variable variance; as a result, the outer sum is also the realization of a Gaussian

random variable with mean equal to the sum of the means of its terms and variance equal to the sum of the variances. The sum of the variances can be calculated as

$$\sum_{j=1}^{L} \frac{u_j'(x)^2}{L} \approx \int_{-\infty}^{\infty} dc_j u_j'(x)^2 = \frac{\sqrt{\pi}A^2}{2\sigma}, \tag{1.54}$$

where the approximation consists in replacing the sum with an integral and in extending the integration domain to the real line. Therefore, the inner sum is distributed according to

$$\bar{W}_i \equiv \sum_{j=1}^{L} W_{ij} u_j'(x) \sim \mathcal{N}\left(0, \frac{\sqrt{\pi}A^2}{2\sigma}\right). \tag{1.55}$$

As a result, the quantity $1/\left\|\mathbf{v}'(x)\right\|_2^2 = 1/\sum_{i=1}^{N} \bar{W}_i^2$ is sampled according to a scaled inverse chi-squared distribution, with mean given by

$$\left\langle \frac{1}{\sum_{i=1}^{N} \bar{W}_i^2} \right\rangle_W = \frac{2\sigma}{\sqrt{\pi}(N-2)A^2} \approx \frac{2\sigma}{\sqrt{\pi}NA^2}. \tag{1.56}$$

The local error is then obtained, from Eqs. (1.52) and (1.56), as

$$\varepsilon_l^2 = \frac{2\sigma\eta^2}{\sqrt{\pi}NA^2}. \tag{1.57}$$

Finally, if we approximate the response amplitude for small σ by $A^2 \approx R/\sqrt{\pi\sigma^2}$, we obtain the expression that appears in the main text (first term of Eq. (1.6)). We note that this expression is equal to the inverse of the Fisher information averaged over network realizations; the Fisher information in case of neural responses corrupted by independent Gaussian noise is given by

$$\langle J(x) \rangle_W = \left\langle \frac{\left\|\mathbf{v}'(x)\right\|_2^2}{\eta^2} \right\rangle_W = \frac{\sqrt{\pi}NA^2}{2\sigma\eta^2}. \tag{1.58}$$

Global error. Here, we extend the calculation performed in the case of discrete stimuli. The analog of Eq. (1.34) in the case of broad tuning curves is

$$\varepsilon_g^2 = \langle P_b(E) \rangle_W \, \bar{\varepsilon}_g^2, \tag{1.59}$$

where the two factors are the probability of a global error and the average squared magnitude of a global error, respectively.

We approximate the probability of a global error by considering a division of the curve of mean population activity into σ 'segments'. These segments are roughly uncorrelated and appear in random locations in the space of population activity; as a result, we can replace L by the number of segments in Eq. (1.5) to obtain the probability of a global error, as

$$\langle P_b(E) \rangle_W \approx \frac{1}{\sigma\sqrt{2\pi N}} \exp\left(-\log\left(1 + \frac{R}{2\eta^2}\right)\frac{N}{2}\right). \tag{1.60}$$

Similarly to the discrete case, when a global error occurs, the decoded stimulus is uniformly sampled from all the other stimuli belonging to incorrect segments. We illustrate the

calculation for the case $x - \sigma > 0$ and $x + \sigma < 1$; similar results can be obtained for stimuli close to the boundaries of the stimulus range. In this case, the output of the decoder is distributed uniformly in the interval $\hat{x} \in [0, x - \sigma] \cup [x + \sigma, 1]$. The average magnitude of global errors is therefore

$$\begin{aligned}\bar{\varepsilon}_g^2 &= \left\langle \int_0^1 dx \, (\hat{x} - x)^2 \right\rangle_W \\ &\approx \int_0^1 dx \frac{1}{(1-2\sigma)} \left[\int_0^{x-\sigma} d\hat{x} (\hat{x} - x)^2 + \int_{x+\sigma}^1 d\hat{x} (\hat{x} - x)^2 \right] \\ &= \frac{1\,(1 - 4\sigma^3)}{6\,(1-2\sigma)},\end{aligned} \tag{1.61}$$

which is a term of the order 1, the size of the stimulus range, plus corrections of order σ. We obtain the expression for the global error which appears in the main text (second term of Eq. (1.6)), by combining Eqs. (1.59), (1.60) and (1.61).

Local and global errors in the case of multi-dimensional stimuli. The MSE for multi-dimensional stimuli, Eq. (1.28), averaged over synaptic weights realizations, is defined as the sum of the MSEs along each stimulus dimension,

$$\varepsilon^2 = \sum_{k=1}^K \varepsilon_k^2 = \sum_{k=1}^K \left\langle \int_0^1 dx_k \int d\mathbf{r} p(\mathbf{r}|\mathbf{x}) \, (\hat{x}_k - x_k)^2 \right\rangle_W . \tag{1.62}$$

The local error along stimulus dimension k can be calculated, similarly to Eq. (1.52), as

$$\begin{aligned}\varepsilon_{l,k}^2 &= \left\langle \int d\mathbf{z} p(\mathbf{z}) \, (\hat{x}_k - x_k)^2 \right\rangle_W \\ &\approx \left\langle \int d\mathbf{z} p(\mathbf{z}) \frac{\|\mathbf{z} \cdot \hat{\mathbf{v}}'_k(\mathbf{x})\|_2^2}{\|\mathbf{v}'_k(\mathbf{x})\|_2^2} \right\rangle_W \\ &\approx \left\langle \frac{\eta^2}{\|\mathbf{v}'_k(\mathbf{x})\|_2^2} \right\rangle_W ,\end{aligned} \tag{1.63}$$

where the noise is projected onto the direction parallel to the partial derivative of the mean activity with respect to stimulus dimension k, $\mathbf{v}'_k(\mathbf{x}) = \partial \mathbf{v}(\mathbf{x}) / \partial x_k$

Local error—sensory neurons with pure tuning. The derivative of the tuning function with respect to stimulus dimension k is given by

$$v'_{i,k}(\mathbf{x}) = \frac{\partial v_i(\mathbf{x})}{\partial x_k} = \sum_{j=1}^Q W_{ijk} \frac{\partial u_{j,k}^p(x_k)}{\partial x_k}. \tag{1.64}$$

Similarly to the one dimensional case, this is a sum of realizations of independent Gaussian random variables. Dropping the superscript p for the sake of clarity, the sum of the variances of these terms is calculated as

$$\sum_{j=1}^Q \frac{1}{L} \left(\frac{\partial u_{j,k}(x_k)}{\partial x_k} \right)^2 \approx \frac{1}{K} \int_{-\infty}^{\infty} dc_j^k \left(\frac{\partial u_{j,k}(x_k)}{\partial x_k} \right)^2 = \frac{\sqrt{\pi} A^2}{2K\sigma}, \tag{1.65}$$

where the approximation consists in replacing the sum $\sum_{j=1}^{Q} \frac{K}{L} f(c_j^k)$ with an integral $\int dc_j^k f(c_j^k)$, and in extending the integration domain to the real line. The sum is distibuted as

$$\bar{W}_{i,k}^{p} \equiv \sum_{j=1}^{Q} W_{ijk} \frac{\partial u_{j,k}(x_k)}{\partial x_k} \sim \mathcal{N}\left(0, \frac{\sqrt{\pi} A_p^2}{2K\sigma}\right). \tag{1.66}$$

Finally, by calculating the mean of the scaled inverse chi-squared distribution,

$$\left\langle \frac{1}{\sum_i^N (\bar{W}_{i,k}^{p})^2} \right\rangle_W \approx \frac{2K\sigma}{\sqrt{\pi} N A_p^2}, \tag{1.67}$$

in Eq. (1.63), we obtain the local error along a single stimulus dimension, as

$$\varepsilon_{p,l,k}^2 = \frac{2K\sigma\eta^2}{\sqrt{\pi} N A_p^2}; \tag{1.68}$$

the total local error is then obtained by summing over dimensions,

$$\varepsilon_{p,l}^2 = \sum_{k=1}^{K} \varepsilon_{p,l,k}^2 = \frac{2K^2\sigma\eta^2}{\sqrt{\pi} N A_p^2}. \tag{1.69}$$

Local error—sensory neurons with conjunctive tuning. The derivative of the tuning function with respect to stimulus dimension k is given by

$$\begin{aligned} v'_{i,k}(\mathbf{x}) = \frac{\partial v_i(\mathbf{x})}{\partial x_k} &= \sum_{j=1}^{L} W_{ij} \frac{\partial u_j^c(\mathbf{x})}{\partial x_k} \\ &= -\sum_{j=1}^{L} W_{ij} \frac{(x_k - c_{j,k})}{\sigma^2} u_j^c(\mathbf{x}), \end{aligned} \tag{1.70}$$

where $c_{j,k}$ is the kth component of the preferred stimulus of neuron j, $\mathbf{c}_j$. Similarly to the previous calculations, this is a sum of realizations of independent Gaussian random variables of different variances. Dropping the superscript c for the sake of clarity, the sum of the variances of these terms is calculated as

$$\begin{aligned} \sum_{j=1}^{L} \frac{(x_k - c_{j,k})^2}{L\sigma^4} u_j(\mathbf{x})^2 &\approx \int d\mathbf{c}_j \frac{(x_k - c_{j,k})^2}{\sigma^4} u_j(\mathbf{x})^2 \\ &\approx \frac{\pi^{K/2} A_c^2}{2\sigma^{(2-K)}}, \end{aligned} \tag{1.71}$$

where the approximation consists in replacing the sum $\sum_{j=1}^{L} \frac{1}{L} f(\mathbf{c}_j)$ with a K-dimensional integral $\int d\mathbf{c}_j f(\mathbf{c}_j)$, and in extending the integration domain. The sum is therefore distributed as

$$\bar{W}_{i,k}^{c} \equiv \sum_{j=1}^{L} W_{ij} \frac{\partial u_j(\mathbf{x})}{\partial x_k} \sim \mathcal{N}\left(0, \frac{\pi^{(K/2)} A_c^2}{2\sigma^{(2-K)}}\right). \tag{1.72}$$

Finally, by calculating the mean of the scaled inverse chi-squared distribution,

$$\left\langle \frac{1}{\sum_i^N (\bar{W}_{i,k}^c)^2} \right\rangle_W \approx \frac{2\sigma^{2-K}}{\pi^{(K-2)} N A_c^2}, \tag{1.73}$$

in Eq. (1.63), and by summing over dimensions, we obtain the total local error as

$$\varepsilon_{c,l}^2 = \sum_{k=1}^{K} \varepsilon_{c,l,k}^2 = \frac{2\sigma^{(2-K)}\eta^2}{\pi^{K/2} N A_c^2}. \tag{1.74}$$

If we approximates A_c^2 and A_p^2 for small values of σ, we obtain that the ratio of the local errors in case of sensory neurons with pure and conjunctive tuning is

$$\frac{\epsilon_{c,l}^2}{\varepsilon_{p,l}^2} \approx \frac{1}{K}. \tag{1.75}$$

Global error—sensory neurons with pure tuning. In the case of sensory neurons with pure tuning, the tuning function of a representation neuron is obtained as the superposition of one-dimensional tuning curves (Eq. (1.21)). According to the ML decoder, Eq .(1.32), the decoder output can be written as

$$\begin{aligned} \hat{\mathbf{x}} &= \arg\min_{\mathbf{x}'} \left\| \mathbf{v}(\mathbf{x}) + \mathbf{z} - \mathbf{v}(\mathbf{x}') \right\|_2^2 \\ &= \arg\min_{\mathbf{x}'} \left\| \sum_{k=1}^{K} \left(\mathbf{v}_k(x_k) - \mathbf{v}_k(x_k') + \mathbf{z}_k \right) \right\|_2^2, \end{aligned} \tag{1.76}$$

where $\mathbf{z}_k$ is the projection of the noise vector onto the direction parallel to the partial derivative of the mean activity with respect to stimulus dimension k. For most realizations of the random tuning curves, if $K \ll N$, the K vectors summed in Eq. (1.76) are likely orthogonal. Thus, minimizing the squared norm of the sum is equivalent to minimizing the sum of the squared norms of each of the vectors. This, in turn, the stimulus estimate can be obtained independently for each stimulus dimension, as

$$\hat{x}_k = \arg\min_{x_k'} \left\| \mathbf{v}_k(x_k) - \mathbf{v}_k(x_k') + \mathbf{z}_k \right\|. \tag{1.77}$$

Therefore, a global error can occur in one or several stimulus dimensions; it requires that $|\hat{x}_k - x_k| > \sigma$ for some k. If the probability of a global error on more than one stimulus dimension is negligible, the total probability of a global error can be approximated as the sum of probabilities over dimensions, $\langle P(E_{p,g}) \rangle_W \approx \sum_{k=1}^{K} \langle P(E_{k,g}) \rangle_W$. We calculated the probability of a global error in the one-dimensional case in the previous section. In order to extend the formula to this case, we have to take into account that the variance of the tuning function along one stimulus dimension is

$$\begin{aligned} \left\langle \int_0^1 dx_k \left[v_i(\mathbf{x}) - \left(\int_0^1 dx_k v_i(\mathbf{x}) \right) \right]^2 \right\rangle_W &= \sum_{j=1}^{Q} \frac{1}{L} \left(\int_0^1 dx_k u_{j,k}(x_k)^2 - \left(\int_0^1 dx_k u_{j,k}(x_k) \right)^2 \right) \\ &\approx \frac{R}{K}. \end{aligned} \tag{1.78}$$

This quantity is the signal variance which governs the rate of exponential suppression of the probability of global error; replacing R by R/K in Eq. (1.60), multiplying by the average squared magnitude of global errors and summing over dimensions, we obtain the global error as

$$\varepsilon_{p,g}^2 \approx \frac{K\bar{\varepsilon}_g^2}{\sigma\sqrt{2\pi N}} \exp\left(-\log\left(1+\frac{R}{2K\eta^2}\right)\frac{N}{2}\right). \tag{1.79}$$

Global error - sensory neurons with conjunctive tuning. The correlation of the responses of neuron i to two stimuli, $\mathbf{x}$ and $\mathbf{x}'$, reads

$$\left\langle v_i(\mathbf{x})v_i(\mathbf{x}')\right\rangle_W \approx A_c^2\left(\pi\sigma^2\right)^{K/2}\exp\left(-\frac{\Delta\mathbf{x}^2}{4\sigma^2}\right), \tag{1.80}$$

where $\Delta\mathbf{x}^2 = \|\mathbf{x}-\mathbf{x}'\|_2^2$; it is exponentially suppressed if $\|\mathbf{x}-\mathbf{x}'\|_2 > \sigma$. By analogy to the one-dimensional case, we divide the surface described by the population activity as a function of the stimulus, $\mathbf{v}(\mathbf{x}) = \{v_1(\mathbf{x}), ..., v_N(\mathbf{x})\}$, into $1/\sigma^K$ uncorrelated regions. We calculate the global error by replacing L with the number of uncorrelated regions in Eq. (1.5), obtaining

$$\varepsilon_{c,g}^2 \approx \frac{\bar{\varepsilon}_{c,g}^2}{\sigma^K\sqrt{2\pi N}} \exp\left(-\log\left(1+\frac{R}{2\eta^2}\right)\frac{N}{2}\right), \tag{1.81}$$

where $\bar{\varepsilon}_{c,g}^2$ is the average squared magnitude of a global error, a term of the order of the stimulus range.

Influence of correlated output noise on population coding.

Correlated output noise due to independent noise in sensory neurons. We consider the case in which the activity of sensory neurons is affected by independent 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 noise in the responses of representation neurons, with covariance matrix $\mathbf{\Sigma} = \eta^2\mathbf{I} + \xi^2\mathbf{W}\mathbf{W}^T$. The matrix $\mathbf{W}\mathbf{W}^T$ is sampled according to a Wishart distribution, with mean $\mathbf{I}$ and variance of the matrix elements of order $1/L$ [83]. We write the covariance matrix as the identity plus a perturbation, as

$$\begin{aligned}\mathbf{\Sigma} &= \tilde{\eta}^2\mathbf{I} + \xi^2(\mathbf{W}\mathbf{W}^T - \mathbf{I}) \\ &= \tilde{\eta}^2\left(\mathbf{I} + \frac{\xi^2}{\tilde{\eta}^2}\left(\mathbf{W}\mathbf{W}^T - \mathbf{I}\right)\right),\end{aligned} \tag{1.82}$$

where $\tilde{\eta}^2 = \eta^2 + \xi^2$. In order to quantify the effect of input noise on the coding performance, we calculate the inverse of the Fisher information (FI) as a lower bound to the MSE. The FI is written as

$$\begin{aligned}J(x) &= \mathbf{v}'(x)^T\mathbf{\Sigma}^{-1}\mathbf{v}'(x) \\ &= \mathbf{u}'(x)^T\mathbf{W}^T\mathbf{\Sigma}^{-1}\mathbf{W}\mathbf{u}'(x).\end{aligned} \tag{1.83}$$

We expand the inverse of the noise covariance matrix to second order in $\xi^2/\tilde{\eta}^2$, as

$$\mathbf{\Sigma}^{-1} \approx \frac{1}{\tilde{\eta}^2}\left(\mathbf{I} - \frac{\xi^2}{\tilde{\eta}^2}\left(\mathbf{W}\mathbf{W}^T - \mathbf{I}\right) + \frac{\xi^2}{\tilde{\eta}^4}\left(\mathbf{W}\mathbf{W}^T - \mathbf{I}\right)^2\right). \tag{1.84}$$

In this approximation, the FI becomes $J(x) \approx J_{\text{ind}}(x) + \delta J(x)$, with

$$J_{\text{ind}}(x) = \frac{1}{\tilde{\eta}^2}\mathbf{u}'(x)^T\mathbf{B}\mathbf{u}'(x), \tag{1.85}$$

and

$$\delta J(x) = \frac{1}{\tilde{\eta}^2}\left(-\frac{\xi^2}{\tilde{\eta}^2}\mathbf{u}'(x)^T\left(\mathbf{B^2}-\mathbf{B}\right)\mathbf{u}'(x) + \frac{\xi^4}{\tilde{\eta}^4}\mathbf{u}'(x)^T\left(\mathbf{B}^3-2\mathbf{B}^2+\mathbf{B}\right)\mathbf{u}'(x)\right), \tag{1.86}$$

where $\mathbf{B} = \mathbf{W}^T\mathbf{W}$. The first term, $J_{\text{ind}}(x)$, is the FI for independent Gaussian output noise with variance $\tilde{\eta}^2$; by averaging over synaptic weights realizations, we obtain the expression in Eq. (1.58),

$$\langle J_{\text{ind}}(x)\rangle_W = \frac{\sqrt{\pi}NA^2}{2\sigma\tilde{\eta}^2}. \tag{1.87}$$

The average of the second term, $\delta J(x)$, over network realizations depends on the moments of the matrix $\mathbf{B}$, which can be computed using Wick's theorem: from the identity $\langle W_{ij}W_{mn}\rangle_W = \frac{1}{L}\delta_{im}\delta_{jn}$, we obtain

$$\langle B_{mn}\rangle_W = \left\langle \sum_{j=1}^{N} W_{jm}W_{jn}\right\rangle_W = \frac{N}{L}\delta_{mn}, \tag{1.88}$$

$$\left\langle B_{mn}^2\right\rangle_W = \left\langle \sum_{i=1}^{L}\sum_{j=1,j'=1}^{N} W_{jm}W_{ji}W_{j'i}W_{j'n}\right\rangle_W = \left(\frac{N}{L}+\frac{N^2}{L^2}+\frac{N}{L^2}\right)\delta_{mn}, \tag{1.89}$$

$$\begin{aligned}\left\langle B_{mn}^3\right\rangle_W &= \left\langle \sum_{i=1,i'=1}^{L}\sum_{j=1,j'=1,j''=1}^{N} W_{jm}W_{ji}W_{j'i}W_{j'i'}W_{j''i'}W_{j''n}\right\rangle_W \\ &= \left(\frac{N}{L}+3\frac{N^2}{L^2}+3\frac{N}{L^2}+\frac{N^3}{L^3}+3\frac{N^2}{L^3}+4\frac{N}{L^3}\right)\delta_{mn}.\end{aligned} \tag{1.90}$$

From now on, we consider the terms up to $\mathcal{O}(N^2/L^2)$; the mean of the perturbation term in the FI becomes

$$\langle \delta J(x)\rangle_W \approx \frac{1}{\tilde{\eta}^2}\mathbf{u}'(x)^T\mathbf{I}\mathbf{u}'(x)\left(-\frac{N^2\xi^2}{L^2\tilde{\eta}^2}+\frac{N^2\xi^4}{L^2\tilde{\eta}^4}\right). \tag{1.91}$$

Finally, we compute the first factor by approximating the discrete sum with the integral, similarly to previous calculations, obtaining

$$\begin{aligned}\frac{1}{\tilde{\eta}^2}\mathbf{u}'(x)^T\mathbf{I}\mathbf{u}'(x) &= \frac{1}{\tilde{\eta}^2}\sum_{j=1}^{L}\frac{(x-c_j)^2}{\sigma^4}u_j(x)^2 \\ &\approx \frac{L}{\sigma^4\tilde{\eta}^2}\int_{-\infty}^{\infty} dc_j\,(x-c_j)^2\,u_j(x)^2 \\ &= \frac{\sqrt{\pi}LA^2}{2\sigma\tilde{\eta}^2}.\end{aligned} \tag{1.92}$$

This quantity is proportional to the mean of the FI in the case of independent noise, Eq. (1.87), by a factor N/L. Combining Eqs. (1.87), (1.91) and (1.92), we obtain

$$\langle J(x)\rangle_W \approx \frac{\sqrt{\pi}NA^2}{2\sigma\tilde{\eta}^2}\left(1-\frac{N\xi^2}{L\tilde{\eta}^2}+\frac{N\xi^4}{L\tilde{\eta}^4}\right). \tag{1.93}$$

We approximate the local error as the inverse of the FI; including only corrections up to $\mathcal{O}\left(N\xi^4/L\tilde{\eta}^4\right)$, we obtain the expression that appears in the main text (Eq. (1.11)),

$$\varepsilon_l^2 \approx \frac{1}{\langle J(x)\rangle_W} \approx \varepsilon_{l,\text{ind}}^2\left(1+\frac{N\xi^2}{L\tilde{\eta}^2}-\frac{N\xi^4}{L\tilde{\eta}^4}\right). \tag{1.94}$$

Correlated output noise with random covariance structure. Similar calculations can be carried out for a noise covariance matrix that obeys the same statistics as those of $\mathbf{W}\mathbf{W}^T$, but that does not derive from the structure of synaptic weights. We consider

$$\mathbf{\Sigma}_{\text{rand}} = \eta^2 I + \xi^2 \mathbf{X}\mathbf{X}^T, \tag{1.95}$$

with $X_{ij} \sim \mathcal{N}(0, \frac{1}{L})$, such that $\langle X_{ij}W_{mn}\rangle_{W,X} = 0$ and $\langle X_{ij}X_{mn}\rangle_X = \frac{1}{L}\delta_{im}\delta_{jn}$. In this case, by expanding the inverse of the covariance matrix to second order in $\xi^2/\tilde{\eta}^2$ in Eq. (1.83), we obtain the perturbation term in the FI as

$$\begin{aligned}\delta J(x) = \frac{1}{\tilde{\eta}^2}\Bigg(&-\frac{\xi^2}{\tilde{\eta}^2}\mathbf{u}'(x)^T\left(\mathbf{W}^T\mathbf{X}\mathbf{X}^T\mathbf{W}-\mathbf{B}\right)\mathbf{u}'(x)\\ &+\frac{\xi^4}{\tilde{\eta}^4}\mathbf{u}'(x)^T\left(\mathbf{W}^T\left(\mathbf{X}\mathbf{X}^T\right)^2\mathbf{W}-2\mathbf{W}^T\mathbf{X}\mathbf{X}^T\mathbf{W}+\mathbf{B}\right)\mathbf{u}'(x)\Bigg).\end{aligned} \tag{1.96}$$

We compute the mean of these matrices over realizations of the noise covariance matrix and of the synaptic matrix using Wick's theorem. We obtain

$$\left\langle\left(W^T X X^T W\right)_{mn}\right\rangle_{W,X} = \left\langle\sum_{i=1}^{L}\sum_{j=1,j'=1}^{N} X_{ji}W_{jm}X_{j'i}W_{j'n}\right\rangle_{W,X} = \frac{N}{L}\delta_{mn}, \tag{1.97}$$

$$\begin{aligned}\left\langle\left(W^T\left(XX^T\right)^2 W\right)_{mn}\right\rangle_{W,X} &= \left\langle\sum_{i=1,i'=1}^{L}\sum_{j=1,j'=1,j''=1}^{N} W_{jm}X_{ji}X_{j'i}X_{j'i'}X_{j''i'}W_{j''n}\right\rangle_{W,X}\\ &= \left(\frac{N}{L}+\frac{N^2}{L^2}\right)\delta_{mn}.\end{aligned} \tag{1.98}$$

Therefore, the first order correction vanishes, and the FI is increased,

$$\langle J(x)\rangle_{W,X} \approx \frac{\sqrt{\pi}NA^2}{2\sigma\tilde{\eta}^2}\left(1+\frac{N\xi^4}{L\tilde{\eta}^4}\right), \tag{1.99}$$

yielding a negative correction to the MSE (Eq. (1.12)).

1.4.4 Data analysis and model fitting

Description of the data and summary statistics. The data consist of the responses (firing rates) of $N \sim 500$ neurons, recorded during an arm posture 'hold' task including 27 different positions, with 2 hand orientations each, arranged in a virtual cube of size 40x40x40 cm. The response of each neuron for each hand position is recorded in several trials (~ 10 trials per hand position). Tuning curves are computed by averaging over trials. In order to quantify the degree of irregularity of a tuning curve in a non-parametric form, the authors used a 'complexity measure': for neuron i, it is defined as the standard deviation of a discretized derivative of the mean response:

$$c(D_{\min})_i = \text{std}\left(\frac{\|v_i(\mathbf{x}) - v_i(\mathbf{x}+\Delta\mathbf{x})\|}{\sqrt{\|\Delta\mathbf{x}\|^2}} \quad \text{s.t.} \quad \|\Delta\mathbf{x}\|_2^2 < D_{\min}\right), \tag{1.100}$$

where $v_i(\mathbf{x})$ is the mean response, D_{min} is the distance between two neighboring hand positions, and in the experiment is equal to 35. [40] evaluated also another summary statistics, the distribution of R^2 values resulting from a fit of the tuning curves with a linear model (see Eq. (1.9), originally proposed by [48]):

$$R_i^2 = 1 - \frac{\sum_{\mathbf{x}} \left(\mathbf{v}_l(\mathbf{x}) - \mathbf{v}(\mathbf{x})\right)^2}{\sum_{\mathbf{x}} \mathbf{v}(\mathbf{x})^2}, \tag{1.101}$$

where $\mathbf{v}_l(\mathbf{x})$ is the response predicted by a linear regression of the data, and the sum is over hand positions used in the experiment. The distribution of these two quantities across neurons is a measure of the irregularity of the neural population response; if the population were perfectly described by a linear model, the R^2-distribution would be a constant for all neurons and equal to 1, while the complexity measure would exhibit low values.

Model fitting and comparison between irregular and linear tuning curves. We consider neurons responding with at least 5 spikes/s at more than two target positions and we compute their tuning curves by averaging the firing rates over trials. Then, we shift and normalize the tuning curves to cancel their means and set their variances across hand positions to unity. We use a version of our shallow network model to produce three-dimensional mean-response profiles. The sensory layer is made up of $L = 100^3$ neurons; the preferred stimuli (here, hand positions) are arranged so as to cover a space of 100x100x100 cm, in such a way that hand positions used in the experiment are placed far from the boundaries of the stimulus space. To limit computation load, we choose $\mathbf{W}$ as a sparse random matrix, with sparsity equal to 0.1, with Gaussian-distributed elements, similarly to the model of [40]. The sparsity of the matrix does not affect our results, as long a proper normalization of the synaptic weights is taken into account and the representation neurons receive a sufficient number of inputs from the sensory layer, i.e., as long as the matrix is not too sparse and the tuning width is not too narrow. The tuning curves are normalized to have zero mean and unit variance across hand positions. With respect to the model of [40], there are two main differences: in their case the random weights were distributed according to a uniform distribution, and a rectifying non-linear function was applied to the random sum of the activity of first-layer neurons to enforce a positive activity of the representation neurons. Their model thus had two tunable parameters: the tuning width of first-layer neurons, σ, and the the

threshold of the non-linear transfer function in the second layer. The only tunable parameter in our model is σ.

In order to fit our model, we generate neural responses of a number of representation neurons equal to the number of recorded neurons, using the same set of hand positions to as used in the experiment. We then computed the distribution of the complexity measure for different values of σ; we denote by σ_f the tuning-curve width which minimizes the Kolmogorov-Smirnov (KS) distance between the distribution produced by the model and that extracted from the data (Fig. S1.1A). The KS distance is a measure of discrepancy between two probability distributions. We denote by $F_{\text{data/model}}(c)$ the empirical cumulative distribution function of the complexity measure across data/model, that is, the empirical probability of finding a neuron with complexity measure less than c,

$$F_{\text{data/model}}(c) = \frac{\#\text{ neurons in the data/model with complexity measure} < \text{c}}{N}, \tag{1.102}$$

where N is the total number of neurons. The KS distance is defined as the maximum absolute difference between the F_{data} and F_{model}:

$$KS \equiv \max_c |F_{\text{data}}(c) - F_{\text{model}}(c)|. \tag{1.103}$$

Figure S1.1C compares the distribution of the complexity measure across neurons for our model with $\sigma = \sigma_f$ with the one found in data and the one calculated for a population with linear tuning curves. For the sake of completeness, we also computed the KS distance between the distributions of R^2 corresponding to model and data (Fig. S1.1A, red line). We mention that the model of [40] with two tunable parameters did not reproduce the distributions of complexity measure and of R^2, and only the complexity measure was taken into account in the fitting procedure. A better fit can be obtained in a heterogeneous model, at the cost of tracking many more parameters (two per neuron): see [51] for a more detailed discussion of the fitting procedure in such a model.

We also extract a noise model from the data, as follows. We define the variance of the mean response of neuron i across hand positions as the variance of the average responses across hand positions, $\hat{R}_i = \langle(\tilde{v}_i(\mathbf{x}) - \langle\tilde{v}_i(\mathbf{x})\rangle_x)^2\rangle_{\mathbf{x}}$, where $\tilde{v}_i(\mathbf{x})$ is the unnormalized tuning curve. Similarly, we average the trial-to-trial variability across different stimuli to obtain the variance of the noise, $\hat{\eta}_i^2 = \left\langle\left\langle r_i^t - \tilde{v}_i(\mathbf{x})\right\rangle_t\right\rangle_{\mathbf{x}}$, where r^t is the response at trial t. In the model, we set the variance of the signal to unity and we rescaled the noise variance correspondingly, as

$$\eta_i^2 = \frac{\hat{\eta}_i^2}{\hat{R}_i}. \tag{1.104}$$

In principle, the noise may depend on the stimulus. To control for this effect, we pre-process the data with a variance stabilizing transformation. We substitute $r_i(\mathbf{x})$ by $\sqrt{r_i(\mathbf{x})}$, [84]), and we recalculated the variance of the noise accordingly. In this way, if the noise were proportional to the mean, one would obtain a constant estimate of the variance of the responses for different hand positions. The distribution of noise variances across neurons calculated in this way does not differ substantially from the one obtained without this data transformation.

For numerical simulations (Fig. 1.6), the tuning curves are computed at a finer scale than in the data (cubic grid of 21x21x21 points instead of 3x3x3). We illustrate three examples

of tuning curves obtained with $\sigma = \sigma_f$, measured at these hand positions in Fig. S1.1D-F, together with the prediction obtained from a linear regression (Eq. (1.9)). We note that there are some neurons which are well described by the linear model while others are not compatible with it. We generated the tuning curves for a number of neurons equal to the number of neurons analyzed in the fitting procedure ($N_{\text{tot}} = 400$). Results for a given population size, N, are obtained by averaging over 8 different pools of size N sampled with replacement from N_{tot}. In Fig. 1.6A-C, we compare the MSE as obtained in a population in which neurons respond according to the irregular tuning curves generated by our model and a population in which the tuning curves are linear, Eq. (1.9). The latter are generated according to Eq. (1.9), by sampling the preferred directions, $\mathbf{p}_i$, uniformly on the unit sphere; the tuning curves are shifted and normalized to have zero mean and unit variance across hand positions. The comparison is quantified through the mean fractional improvement, defined as

$$\Delta\varepsilon \equiv \frac{\varepsilon_{\text{lin}} - \varepsilon_{\text{irr}}}{\varepsilon_{\text{lin}}}, \tag{1.105}$$

where $\varepsilon_{\text{lin/irr}}$ is the RMSE as obtained in the population with linear/irregular tuning curves.

S1.5 Supplementary Information

S1.5.1 Supplementary figures

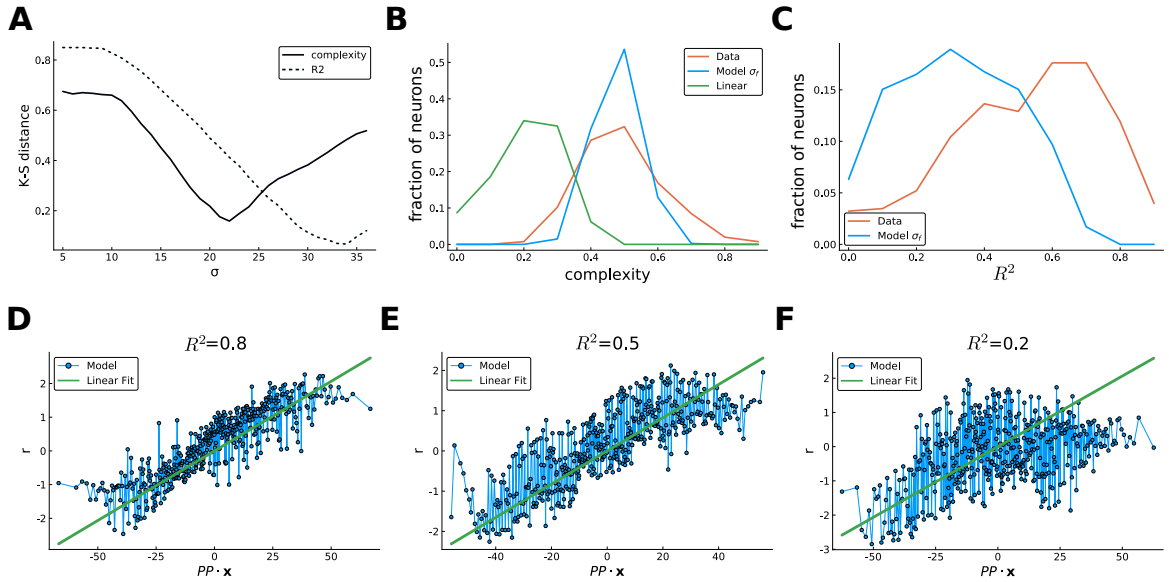

Fig. S1.1 ***Model fitting and tuning curves.*** *(**A**) Kolmogorov-Smirnov distances between the distributions of complexity measure (solid line) and R^2 of fitting across neurons (dashed line) for data and model, for different values of σ: σ_f is chosen to be the value at which the minimum of the distance between complexity distributions is attained, $\sigma_f \sim 22$. (**B**) Normalized-histograms of the distribution of complexity measure (arbitrary units) across the neurons in the data (red), with irregular tuning with $\sigma = \sigma_f$ (blue) and a linear tuning curves (green). The irregular model captures the bulk of the distribution for the data better than a linear model. Nevertheless, the data present a broader distribution across the population. (**C**) Normalized-histograms of the distribution of the R^2 of linear fits across neurons of the data and irregular tuning curves with $\sigma = \sigma_f$ (red). (**D-F**) Three examples of irregular tuning curves with $\sigma = \sigma_f$, showing a broad range of behaviors with respect to a linear fit. The tuning curves are plotted as a function of the projection of the hand position onto a preferred direction, obtained by the fit with Eq.(1.9) (green line). Some neurons are well described by the parametric function (**D**), while others show consistent deviations (**E**); in a subset of neurons, a linear fit fails altogether (**F**)*

Chapter 2

Decoding Complex Neural Responses

2.1 Introduction

The majority of neural populations receive information about the sensory world only through the neural activity of other neural populations. In the search for optimality principles underlying organization of population codes, the efficient coding hypothesis [8] represents one of the most influential theories. It posits that neural responses are arranged so as to maximize the information about the sensory stimuli, given a constraint on the neural resources available. This normative approach has been successful in predicting response patterns evoked by sensory features in different brain areas [9, 85, 86, 87]. In the majority of these studies, however, few assumptions are made about how the information encoded in the neural activity is used in downstream brain areas.

The efficiency of a neural code is often measured through task-agnostic quantities, such as the mutual information between the stimulus and neural activity patterns [88, 46, 89]. When a decoding stage is considered, often within a stimulus-reconstruction task, the point of view of a an *ideal observer*, which has access to the details of the encoding process and the statistics of the noise, is adopted [19, 21]. Similarly, the Fisher information, whose inverse bounds the variance of any unbiased estimator, is often used as a proxy for the minimum decoding error, especially in fine-discrimination tasks where a small difference between stimuli has to be detected from neural responses [90, 91, 22, 92].

In some cases, the assumption of unlimited decoding capacity may lead to paradoxical predictions about the optimal arrangement of neural responses. As an example, we take the model analyzed in Chapter 1: a population of neurons which exhibit complex tuning curves. The optimal level of smoothness is dictated by a balance between two types of errors, and depends on the size of the population and on the variance of the noise. If the population size is large enough, the mean squared error of an ideal decoder is minimized when the correlations between responses vanish, yielding non-smooth tuning curves (Fig. 1.2A). On the other hand, experimental evidence shows that neurons are broadly tuned to parameters of sensory stimuli, implying smooth neural representations [72].

Here, we address this apparent discrepancy by quantifying the coding performance of a neural population belonging to the family described in Chapter 1, a fairly general model for

neurons with smooth tuning curves, through the error in the stimulus estimate as obtained from a non-ideal decoder. If the decoder is defined as a parametric function which maps neural activity patterns to stimulus estimates, our task is to set these parameters optimally. We assume a supervised learning framework, where the parameters are learned so as to minimize a loss function defined on a set of training examples consisting in neural activity patterns and the corresponding stimuli. The performance is then evaluated by measuring the error on the whole distribution of stimuli neural responses, thereby testing the ability of the decoder to generalize. More specifically, we parametrize our decoding function as a deep neural network [93], as, theoretically, neural networks have the property of universal function approximators [94]. Deep artificial neural networks are among state-of-the-art methods to perform regression tasks and, despite their 'artificiality', in the last decade they have acquired an important role as models in computational neuroscience [95].

By restricting ourselves to two-layer neural networks, we first analyze a specific architecture that can approximate an ideal decoder. We show that, by training the decoder to reproduce an approximation of the posterior distribution in the hidden layer, the decoding capacity of this non-ideal decoder approaches the ideal one. The training procedure of such a network requires an assumption about the nature of the hidden-layer representation, and, correspondingly, a particular choice for the loss function. We next relax these strong assumptions by considering a generic two-layer neural network trained to minimize an error-based loss function. In a regime where the parameters of the neural networks vary negligibly during the minimization of the loss, also called 'lazy' regime [96, 97], the decoding function exhibits poor performance when the tuning curves are irregular, yielding a large gap between ideal and non-ideal error. Instead, when the network learns rich 'features' from the data, it is able to take advantage of the higher accuracy achieved by irregular tuning curves. By varying the number of neurons in the hidden layer, we measure how many of these features are necessary to efficiently decode the information contained in the input. We find this number to be inversely proportional to the width of correlation between neural responses. This results in a trade-off between the ideal accuracy of a population code, maximized when neurons possess irregular tuning curves, and the ease of the decoding process, which is facilitated by neural responses which vary smoothly.

Our results complement a growing literature that considers the efficiency of a neural code from the point of view of a downstream area [98, 81, 99]. Here, the decoder's performance is limited by its access to a finite set of noisy examples; such limitation affects different architectures in different ways. We discuss how these limitations and constraints on the neural resources allocated to the decoding process can modify, and in some cases completely reverse, optimality criteria of a neural code.

2.2 Results

2.2.1 Problem setting

We consider a neural population model with neurons with complex, possibly multimodal, responses as a function of a continuous, one-dimensional, stimulus. In particular, we consider a population of N 'representation' neurons in which the mean response function of neuron i as a function of the stimulus, x, i.e., the tuning curve, is sampled from a Gaussian process

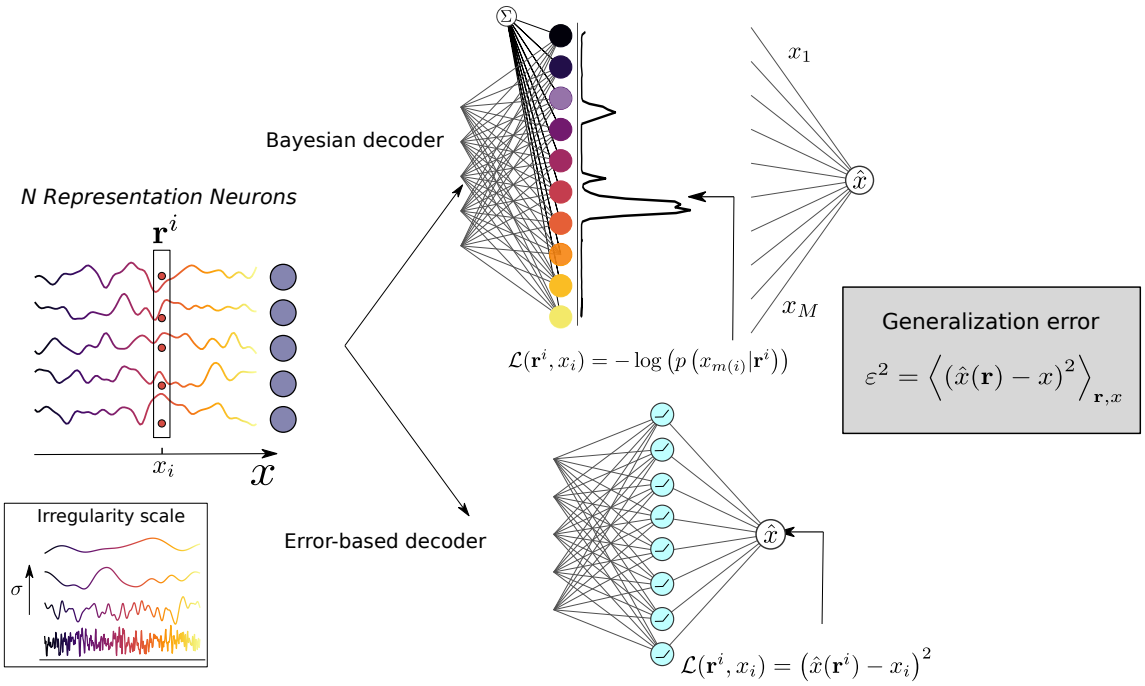

Fig. 2.1 ***Non-ideal decoding architectures.*** *From left to right: N representation neurons respond to a scalar stimulus, x, according to random tuning curves; color indicates stimulus value. The scale of irregularity of the tuning curves is controlled by the parameter σ (bottom inset). At each trial a noisy neural activity pattern, $\mathbf{r}^i$ (red dots), is sampled, and the decoding network returns an estimate of the stimulus. Top: in a Bayesian decoder, the hidden layer reproduces the posterior distribution over stimuli. Each node has a 'preferred stimulus' (color indicates stimulus value), and the parameters are trained so as to maximize the activity of the neuron corresponding to the correct stimulus. A final readout neuron output stimulus estimate by weighting the activity of hidden-layer neurons according to their preferred stimulus value. Bottom: in an error-based decoder, weights and biases of the neural network are trained so as to minimize the error of the stimulus estimate. The decoding performance is evaluated by calculating the MSE of the optimized decoder over the joint distribution of stimuli and neural activity patterns.*

with vanishing mean and Gaussian kernel of width equal to σ,

$$v_i(\cdot) \sim \mathcal{GP}\left(0, \bar{k}\left(\cdot,\cdot\right)\right), \tag{2.1}$$

where $\bar{k}(x,x') = \exp\left(-\frac{(x-x')^2}{4\sigma^2}\right)$[1] . This distribution of tuning curves might be generated by a random linear combination of the activity of neurons with classical, bell-shaped tuning curves (see Chapter 1). We will refer to the parameter σ as the tuning width, as it measures the width of correlation of the Gaussian kernel and it controls the smoothness of the tuning curves. By changing it, we can interpolate between the extreme case of neurons exhibiting

[1] In order to be consistent with the previous chapter, we keep the factor of 4 at the denominator.

random and uncorrelated responses as function of x, when $\sigma \to 0$, and broad, monomodal tuning curves, when σ is large (Fig. 1.1,2.1). At each trial, when stimulus x is shown, the activity of the representation neurons deviates from the mean due to random noise. We assume independent Gaussian noise with variance η^2, such that the noisy activity pattern is obtained as

$$\mathbf{r}(x) = \mathbf{v}(x) + \mathbf{z}, \tag{2.2}$$

where $\mathbf{v}(x) = (v_1(x), ..., v_N(x))$, and $\mathbf{z} \sim \mathcal{N}(o, \eta^2 I)$; the dependence of $\mathbf{r}$ on x will be sometimes implicit in what follows. In Chapter 1, we quantified the mean squared error (MSE) in the stimulus estimate as obtained from an ideal decoder (ideal error) as a function of the population size, N, the noise variance, η^2, and the smoothness of the tuning curves, σ. In particular, we showed that in non-trivial regimes of population size and noise variance, an optimal level of irregularity, controlled by σ, balances two qualitatively different contributions to the error: local and global. When considered as a function of the population size, the optimal width, σ^*, and the optimal error, $\varepsilon^2(\sigma^*)$, decrease exponentially, yielding a code which 'compresses' information from a high-dimensional to a low-dimensional representation. Thus, if the population size is sufficiently large, the minimum-MSE is achieved with extremely irregular tuning curves (σ small). Does this property hold when the stimulus estimate is obtained from a non-ideal decoder, which does not have access to the details of the encoding process?

We define a non-ideal decoder as a parametric function, $\hat{x} = f_\theta(\mathbf{r})$, with θ denoting the set of parameters. We consider a supervised setting, where we are given a training dataset which consists in P pairs of stimuli and the corresponding evoked noisy activity patterns, $\mathcal{D}_v = \{\mathbf{r}^i, x_i\}_{i=1}^P$; the subscript v denotes the dependence of the dataset from the set of tuning curves, $\{v_i\}$. The parameters of the decoder are set so as to minimize a loss function defined on the dataset,

$$\hat{\theta} = \arg\min_\theta \mathcal{L}(f_\theta, \mathcal{D}_v). \tag{2.3}$$

Once the parameters are learned, the decoding performance is defined as the MSE averaged over the distribution of possible stimulus-response pairs, $p_v(\mathbf{r}, x)$,

$$\varepsilon^2(\mathcal{D}_v) = \int d\mathbf{r} dx p_v(\mathbf{r}, x) \left(f_{\hat{\theta}}(\mathbf{r}) - x\right)^2; \tag{2.4}$$

this quantity is known as generalization error, or test error. This quantity must then be averaged over possible datasets and over the distribution of tuning curves, $p(\{v_i\})$, to obtain the mean decoding performance, $\varepsilon^2 = \langle \varepsilon^2(\mathcal{D}_v) \rangle_{\mathcal{D}_v, \{v_i\}}$. We compare this quantity with the ideal error, ε^2_{id}, as obtained by an ideal decoder. In this work, we are primarily interested in understanding how the noise in the training data affects the optimized decoding function. In order to distinguish these effects from limitations imposed purely by the limited amount of data, which is the subject of many studies [81, 99], we will consider the limit of large values of P, such that the quantity in Eq. (2.4) does not depend on the specific realization of the dataset.

We parametrize the decoder as a two-layer neural network. In the hidden layer, M neurons compute a possibly non-linear combination of the activity of the N representation neurons, while an output neuron produces an estimate of the stimulus. We distinguish between two major classes of architectures on the basis of the loss function we use in the training procedure, as follows (Fig. 2.1).

2.2.2 Bayesian decoder

The first architecture we consider is inspired by the ideal decoder. We train the hidden layer of a two-layer neural network to reproduce the posterior distribution over stimuli, and the output neuron computes the mean of this approximated posterior. If the approximation of the posterior is exact, the stimulus estimate corresponds to the ideal one (see Methods in Chapter 1, and Fig. 2.1, top). More specifically, we assume a discretization of the stimulus space into M bins, and we assign to each of the M hidden-layer neurons a preferred stimulus, x_m, the midpoint of the m-th bin. The output of the m-th neuron is obtained as

$$h_m(\mathbf{r}) = \mathcal{S}(\mathbf{u})_m = \frac{\exp(u_m)}{\sum_{m'=1}^{M} \exp u_{m'}}, \tag{2.5}$$

where $\mathcal{S}(\cdot)$ is the softmax function [100]. We intepret the vector $\mathbf{u}$ as presynaptic currents, which are obtained as a linear combination of the neural activity patterns, $\mathbf{u} = \lambda\mathbf{r} + \mathbf{b}$. The softmax function ensures that the vector output of the hidden-layer neurons, $\mathbf{h}$, sum to 1, allowing us to interpret it as a discrete approximation of a probability distribution over stimuli, $h_m(\mathbf{r}^i) \approx q(x_m|\mathbf{r}^i)$.

The parameters of the network, $\theta = \{\lambda, \mathbf{b}\}$, are optimized with stochastic gradient descent so as to minimize the following loss function (which can be interpreted as the cross entropy between the output of the hidden layer and the correct stimulus distribution, in the discretized space, represented as a one-hot vector),

$$\hat{\theta} = \arg\min_{\theta} \left\{ -\frac{1}{P} \sum_{i=1}^{P} \log \left(h_{m(i)}(\mathbf{r}^i) \right) \right\}, \tag{2.6}$$

where the value of the index $m(i)$ is chosen such that $x_i \in (x_m - 1/2M, x_m + 1/2M)$. The loss function described in Eq. (2.6) is optimized when the output of the hidden-layer neurons approximates the correct posterior distribution over stimuli, i.e., $h_m(\mathbf{r^i}) \approx p(x_m|\mathbf{r}^i)$ (see Methods). Indeed, for this functional form of the decoder, there exists a set of weights and biases such the decoder becomes ideal in the large M limit (see Chapter 1). An approximation of the mean of the posterior, corresponding to the minimum MSE estimator, is obtained by weighting the ouptut of the hidden-layer neurons according to their preferred stimulus, $\hat{x} = \sum_m x_m h_m$. We evaluate the decoding performance by measuring the MSE of the stimulus estimate, Eq. (2.4). In order to allow a comparison between different population sizes, we keep fixed the samples-to-parameters ratio, $\gamma = P/N_P$, where $N_p = (M+1)N$, and we consider the regime with $\gamma > 1$ (also called underparametrized regime).

The decoding error for this architecture exhibits the same behavior as the ideal one, with a non-trivial optimal value of σ that balances local and global errors and decreases as a function of the population size (Fig. 2.2A, compare with Fig. 1.3). The non-ideal error is between 2- and 5-fold larger than the ideal one, with this ratio increasing as the tuning curves become more irregular (σ small, Fig. 2.2B). As a result, although the non-ideal optimal σ decreases exponentially fast with the population size, just slightly slower than in the ideal case (Fig. 2.2C, inset), the optimal error is one order of magnitude larger than the ideal one (Fig. 2.2C). Finally, we compare the weights obtained by minimizing the empirical loss function and the ideal ones, by measuring the Pearson correlation coefficient between the column of the connectivity matrix, $\lambda_{:,i}$ and the ideal ones, which correspond to

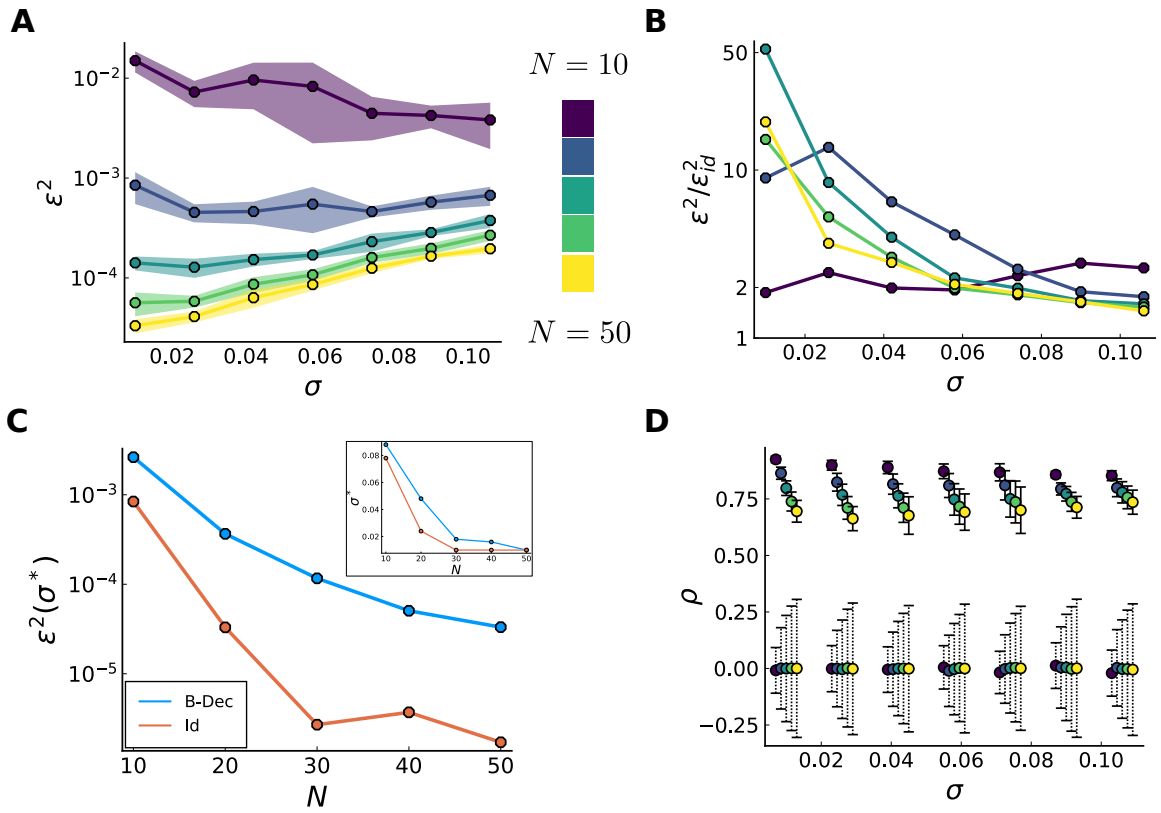

Fig. 2.2 ***Decoding performance of the Bayesian decoder.*** *(**A**) Generalization error, ε^2, of Bayesian decoder as a function of σ for different population sizes, N, colored according to the legend ($M = 500, \gamma = 3$)). (**B**) Ratio between generalization error of Bayesian decoder and ideal error, for different values of N, colored according to the legend in panel A. (**C**) Optimal width (inset) and optimal error as a function of the population size, as obtained with the Bayesian (blue) and ideal (red) decoder. (**D**) Average Pearson correlation coefficient between the columns of the learned synaptic matrix and the ones of the ideal synaptic matrix, for different values of N, colored according to the legend in panel A (solid error bars). As comparison, the mean correlation between two randomly chosen columns from the two matrices vanishes (dashed error bars).*

a discretization of the tuning curves $\lambda^{id}_{:,i} = \{\frac{v_i(x_m)}{\eta^2}\}_{m=1}^{M}$. The two set of weights exhibit a high correlation, which decreases with N, as learning become more difficult (Fig. 2.2D).

2.2.3 Error-based decoder

Learning an approximation of the optimal decoder requires to define the loss function on the hidden-layer output, and a specific teaching signal which targets the neuron whose preferred stimulus corresponds to the correct one. Such setting differs from the classical machine learning regression setting, as well as from the more biological reinforcement learning setting, in which typically an output error signal is provided. We consider a more plausible setting by

studying the decoding properties of a generic two-layer neural network (Fig. 2.1, bottom). The stimulus estimate, $\hat{x}$, given a neural activity pattern, $\mathbf{r}$, is represented by the activity of the output neuron,

$$\hat{x} = f_\theta(\mathbf{r}) = \mathbf{w}_{(1)}^T f_{(0)} \left(W_{(0)}\mathbf{r} + \mathbf{b}_{(0)} \right) + b_{(1)}, \tag{2.7}$$

with $f_{(0)}$ a generic (non-linear) function applied pointwise. For this decoder, the learnable weights, θ, are the input-to-hidden layer weights and biases, $W_{(0)}$ and $\mathbf{b}_{(0)}$, and the final readout weights and biases, $\mathbf{w}_{(1)}$ and $b_{(1)}$. The parameters of the network are optimized with stochastic gradient descent so as to minimize the MSE on the dataset, Eq. (2.39). We illustrate the relevance of the non-linearity of the transfer function, $f_{(0)}$, by examining the trivial, but instructive, linear case, for which an approximate analytical analysis can be carried out. In this case, the decoder output can be written as a linear combination of the output of the N representation neurons, as

$$f_\theta(\mathbf{r}) = \mathbf{w}_{lin}\mathbf{r} + b_{lin}, \tag{2.8}$$

with $\mathbf{w}_{lin} = \mathbf{w}_{(1)}^T W_{(0)}$ and $b_{lin} = \mathbf{w}_{(1)}^T \mathbf{b}_{(0)} + b_{(1)}$. We note, however, that solutions found by optimizing the parameters in Eq. (2.7) through stochastic gradient descent might differ from the optimal linear weights of Eq. (2.8) (which are a solution of a convex problem), as a result of the non-linear training dynamics of deep neural networks [101, 102].

Linear decoder

We consider a linear decoder with a vanishing bias (see Methods),

$$f_l(\mathbf{r}) = \frac{1}{\sqrt{N}} \mathbf{w}^T \mathbf{r}. \tag{2.9}$$

The weights which minimize the MSE, $\hat{\mathbf{w}}$, are obtained as

$$\hat{\mathbf{w}} = (RR^T)^{-1} R\bar{x}, \tag{2.10}$$

where R is the $N \times P$ matrix of neural responses in the training dataset, $R_{ij} = r_i(x_j)/\sqrt{N}$, and $(\bar{x})_i = x_i$ is the column vector of training stimuli.

In a recent study, Jacot et al. [103] considered a regression setting in which assumptions on the statistics of the dataset also apply to our model (Sec. 2.2.1), and they obtained analytical approximations and bounds for the generalization error. Here, we extend their results, obtained in a noise-free setting, to obtain an approximation for the MSE in the noisy case. The MSE can be written as a sum of three terms,

$$\varepsilon^2 = \langle B(x) + V_1(x) + V_2(x) \rangle_x. \tag{2.11}$$

Here, $B(x)$ is a bias term, which measures the systematic deviation of the decoder output averaged over different neural responses to stimulus x. The terms $V_1(x)$ and $V_2(x)$ are variance terms. In particular, $V_1(x)$ measures the variance across different datasets of the decoder output averaged over different neural responses to stimulus x, while $V_2(x)$ measures the variance of the decoder output across different neural responses to stimulus x. In the case of noisy neural responses, $\mathbf{r}$, and large number of samples, P, we argue that $V_2(x)$ yields the dominant contribution to the MSE, as in the other terms the noise in the neural responses

is averaged out (see Methods). An analytical approximation of this term can be calculated (see Methods for the derivation; the calculations are based on the results presented in Ref. [103]). In order to illustrate the limitations introduced by the linear decoder, we write the generalization error as

$$\varepsilon^2_{lin} = \varepsilon^2_{loc,id}\varepsilon^2_{corr}, \tag{2.12}$$

where $\varepsilon^2_{loc,id} = 2\eta^2\sigma^2/N$ is the ideal local error, as calculated in the previous chapter, first term in Eq. (1.6), and

$$\varepsilon^2_{corr} \approx \frac{1}{2\sigma^2\left(1 - \frac{1}{N}\sum_{i=1}^{P}\left(\frac{P\bar{d}^c_i+\eta^2}{P\bar{d}^c_i+\eta^2+\tilde{\lambda}}\right)^2\right)} \sum_{i=1}^{P} \frac{\left(P\bar{d}^c_i+\eta^2\right)\bar{w}_i^2}{\left(P\bar{d}^c_i+\eta^2+\tilde{\lambda}\right)^2}, \tag{2.13}$$

is a multiplicative correction term introduced by the linear decoder. Here, $\bar{d}^c_i$ are the eigenvalues of the kernel integral operator, defined by

$$\int dx' p(x')\bar{k}(x,x')\phi_i(x') = \bar{d}^c_i\phi_i(x), \tag{2.14}$$

$\bar{w}_i$ are the set of weights in the decomposition of the target function, $x = f_l(\mathbf{r})$, as a superposition of kernel eigenfunctions[2], $x = \frac{1}{\sqrt{P}}\sum_{i=1}^{P}\bar{w}_i\phi_i(x)$, and $\tilde{\lambda}$ is an 'effective regularizer,' defined by the identity

$$\sum_{i=1}^{P}\frac{P\bar{d}^c_i+\eta^2}{P\bar{d}^c_i+\eta^2+\tilde{\lambda}} = N. \tag{2.15}$$

We emphasize that the expression for the generalization error depends only on the spectral properties (i.e., the eigenvalues) of the noise-free kernel, $\bar{k}$, and on the variance of the noise, which are deterministic quantities.

The noise affects in a non trivial way the terms in Eq. (2.13), contributing explicitly both to the numerator and the denominator, as well as implicitly, through the definition of the effective regularizer. The tuning width, instead, determines the spectral properties of the kernel, and the weights of the target function decomposition, $\{\bar{w}_i\}$ (Fig. S2.1). Numerically, we observe that, for a fixed noise variance, the effect of the correction term is to revert the increasing behavior as a function of σ of the ideal error.

We check the validity of the analytical results with numerical simulations, by computing the average MSE of the optimal linear decoder and by calculating numerically Eq. (2.13). The linear decoder exhibits a poor performance when compared to the ideal one. In order to illustrate the fundamental limitations imposed by the architecture, which are independent on the size of the population and are due to the noise in the training set, we plot results for large values of N and small values of the noise variance ($\eta = 0.1$). We compare different population sizes by keeping fixed the samples-to-parameters ratio, $\gamma = P/N$, and we explore the regime with $\gamma > 1$. As anticipated above, the MSE decreases as a function of σ, for all values of N (Fig. 2.3A). Thus, the optimal value of σ is constant as a function of N, with optimality achieved with broad tuning curves: the linear decoder is unable to exploit the high

[2]The decoder is a function of $\mathbf{r}$, but its properties are defined as a function of the kernel in the space of stimuli, x; in this space, the target function is simply the identity function. The eigenfunctions of the kernel operator form an orthonormal basis for the space of L^2 functions with respect to the measure $p(x)$, thus such decomposition is always achievable.

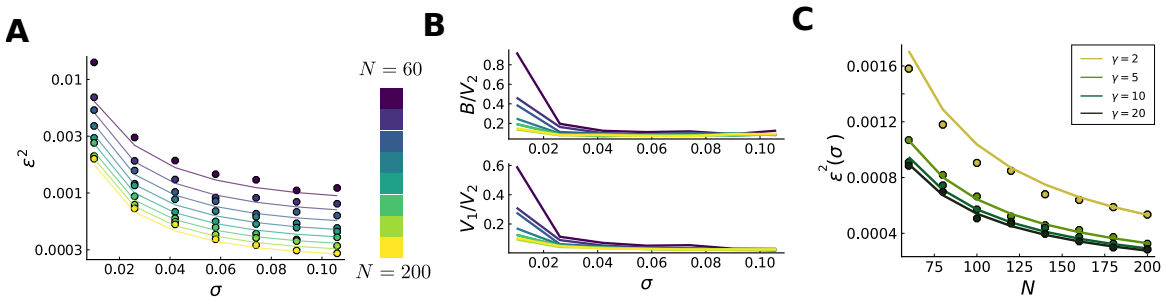

Fig. 2.3 ***Decoding performance of the linear decoder.*** *(**A**) Generalization error of the linear decoder (dots), and comparison with the analytical expression in Eq. (2.13) (solid curves) as a function of σ, for different population sizes, N, colored according to the legend ($\gamma = 5$). The ideal error is not shown, as even the best non-ideal decoding error for large population size ($N = 200$, $\varepsilon^2 \approx 10^{-3.5}$) is more than one order of magnitude larger than the worse ideal error for small population size ($N = 60$, $\varepsilon^2_{id} \approx 10^{-5}$) (**B**) Ratio between the contributions of the first two terms in Eq. (2.11) and V_2, as a function of σ for different values of N, colored according to the legend in panel A. The two terms are relevant only in the low N - low σ regime. (**C**) Generalization error of the linear decoder (dots), and comparison with the analytical expression in Eq. (2.13) (solid curves), for a fixed value of $\sigma = 0.11$ (corresponding to the minimum value of the error), as a function of the population size, for different values of the samples-to-parameters ratio, γ.*

local (ideal) accuracy achieved with irregular tuning curves (σ small). By plotting the inverse ratio between $V_2(x)$ and the two other terms in Eq. (2.11), averaged over the distirbution of stimuli, we show that indeed V_2 dominates the MSE, especially for large values of σ and N (Fig. 2.3B). For a fixed σ, the error scales as an inverse function of the population size. As we increase the size of the dataset, by increasing γ, we observe a saturation in the error curves, suggesting that the limited performance is not due to a limited amount of data (Fig. 2.3C).

Non-linear neural networks

In order to overcome the limitations imposed by a linear decoder, we consider non-linear transfer function in the hidden layer. Although non-linear neural networks represent the state of the art for many tasks, a comprehensive theoretical understanding of their complex learning dynamics is still lacking (despite the huge progresses coming from a broad range of approaches [104, 105, 106]). A recent line of research has shown that, in some limits, the output of deep neural networks trained with gradient descent can be well approximated by a linear combination of a set of non-linear 'features' of the inputs, which are purely determined by the value of the parameters at initialization [96, 97, 107]. As a result, the network perform analogously to a kernel machine [108]. In this regime, roughly speaking (see Methods for an informal overview), the parameters of the network change only negligibly during the minimization of the loss, and therefore the regime is referred to as '*lazy*'. The function learned by the network can then be written as the solution to a kernel regression

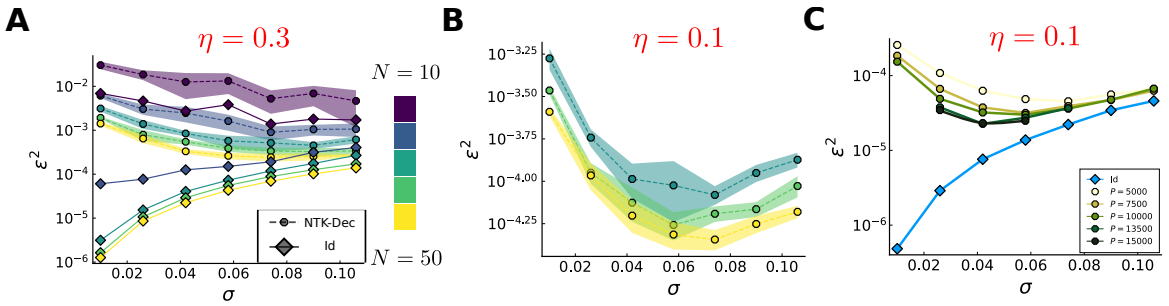

Fig. 2.4 ***Decoding performance of the NTK decoder.*** *(**A**), (**B**) Generalization error of the NTK decoder (dots and dashed curves) and ideal error (diamonds and solid curves) as a function of σ, for different population sizes, N, colored according to the legend ($P = 100N$). Panel A and B illustrate results obtained in the case of high and low noise variance, respectively. For the sake of clarity, in panel B only results for large population sizes are shown. (**C**) Generalization error of the NTK decoder (dots) and ideal error (diamonds) as a function of σ, for different values of P ($N = 50$), in the regime of low noise variance. For the sake of clarity, for large values of P only the intermediate values of σ, where the minimum is achieved and becomes deeper, are shown; the rest of the curves coincides with the ones obtained for low values of P.*

problem (see Methods, Eq. (2.47)),

$$f_{NTK}(\mathbf{r}) = \bar{x}^T K_{NTK}^{-1} \mathbf{k}_{NTK}(\mathbf{r}), \tag{2.16}$$

where $(K_{NTK})_{ij} = k_{NTK}(\mathbf{r}^i, \mathbf{r}^j)$ is a kernel Gram matrix calculated for the response patterns in the dataset, and $(\mathbf{k}_{NTK}(\mathbf{r}))_i = k_{NTK}(\mathbf{r}^i, \mathbf{r})$. Here, $k_{NTK}(\cdot, \cdot)$ is the so-called neural tangent kernel (NTK), see Eq. (2.87), which depends only on the value of the parameters at initialization and on the choice of the non-linear function, $f_{(0)}$, and is fixed during the training dynamic. The initialization of the weights is typically random, yet, for large networks, and if the sampling distribution satisfies certain properties, it has been shown that the NTK converges to a deterministic kernel. For some choices of the non-linear function, it is possible to compute an analytical expression of the NTK [109]. Kernel methods are amenable to analytic treatment, and therefore a series of results and generalization properties have been obtained for neural networks trained in this regime [110, 111, 112, 80].

We test the decoding performance of the function defined in Eq. (2.16), with the NTK obtained in the deterministic limit (see Methods). From the theory referred above, such a function corresponds to the output of an infinitely wide ($M \to \infty$) neural network trained in the lazy regime. Empirically, we found the NTK corresponding to a network with Erf (error function) non-linearity to perform slightly better for this problem, than the more usual NTK with rectifiying linear units (ReLu); therefore, we will illustrate results for the former choice. For a large value of the noise variance, we find a similar monotonic decrease of the MSE as a function of σ, as obtained in the linear case (Fig. 2.4A). Quantitatively, the ideal and non-ideal error are comparable in the regime of large σ, as opposed to a worse performance in the linear case. Intuitively, this is because training a neural network in the lazy regime is equivalent to perform a linear regression in a high-dimensional (with

dimension equal to the number of parameters) space of non-linear functions of the input data (see Methods, Eq. (2.82)). When the variance of the noise is small, a minimum is observed in the generalization error curves at a non-trivial value of the tuning width (albeit shallow, Fig. 2.4B). In order to see if the non-trivial behavior of the generalization error as a function of σ is not merely a result of a limited amount of data, we plot the error curves for increasing values of P; we observe a saturation beyond a given value of P (Fig. 2.4C). (Kernel methods are typically computationally expensive, as they require the inversion of a matrix which grows proportionally to the size of the dataset; for this reason, we were not able to see if a minimum in the error curve was achieved also in the high signal-to-noise regime for very large values of P. Instead, we show below that a neural network trained in the rich regime outperforms a NTK decoder trained on the same amount of data.)

The results in Fig. 2.4A,B show that neural networks trained in the lazy regime take only partial advantage of the higher local precision afforded by irregular tuning curves, and they do so only in the regime of low noise variance. In order to check that this is indeed a limitation of the training regime, we compare the performance of the NTK decoder with that of a wide neural network trained in the rich regime (in what follows, network decoder) on the same dataset (Fig. 2.5A). (As pointed out in [97, 113], the transition between the two regimes is controlled by the variance of the values of the parameters at initialization; see Methods for details on the training procedure employed here.) The network decoder achieves a lower error for intermediate values of σ, while the two performances are comparable for larger values of σ. Thus, a minimum in the generalization error curve is observed for non-trivial values of the tuning width, suggesting that the rich training regime is more efficient in extracting information when it is encoded in complex neural activity patterns. The decoding performance can be further increased by increasing the size of the dataset, allowing for a more accurate characterization of the statistics of the neural noise; the non-trivial optimal value of σ, which decreases as a function of the population size, becomes more evident (Fig. 2.5B). The ratio between non-ideal and ideal generalization errors is large for irregular tuning curves, and it becomes of order 1 as the tuning width increases (Fig. 2.5B, inset).

These results are obtained for neural networks with a wide hidden layer ($M = 1000$). Given the remarkable ability of neural networks to extract useful features from inputs, we ask how many of these features are necessary to achieve a good decoding performance. By varying the number of hidden-layer neurons, we observe that, depending on the value of σ, smaller networks can achieve similar performances as a wide one (Fig. 2.5C).

Figure 2.5D illustrates, as a function of σ, the crossover number of hidden-layer neurons, M_{sat}, such that a further increase in the size of the network causes a negligible increase in the decoding performance. This number is inversely proportional to σ: it's behavior can be decribed well by fitting an hyperbolic function, $M_{sat}(\sigma) = a/\sigma + b$ (the exact coefficients depend on the number of input neurons, N, on the noise variance and on the criterium chosen for the performance saturation). The number of hidden-layer neurons corresponds to the number of non-linear projections of the neural responses. By assuming that, in the rich training regime, the network learns 'meaningful' projections, capturing the structure of the underlying curve of population activity, the minimum number of them which is necessary for an accurate readout is inversely proportional to the scale of irregularity of tuning curves (see Fig. 2.6B and Discussion for a geometrical interpretation). More generally, this result reveals a trade-off between the theoretical *encoding* capacity of the population code, which increases with the complexity of neural responses, and the number of neural resources needed

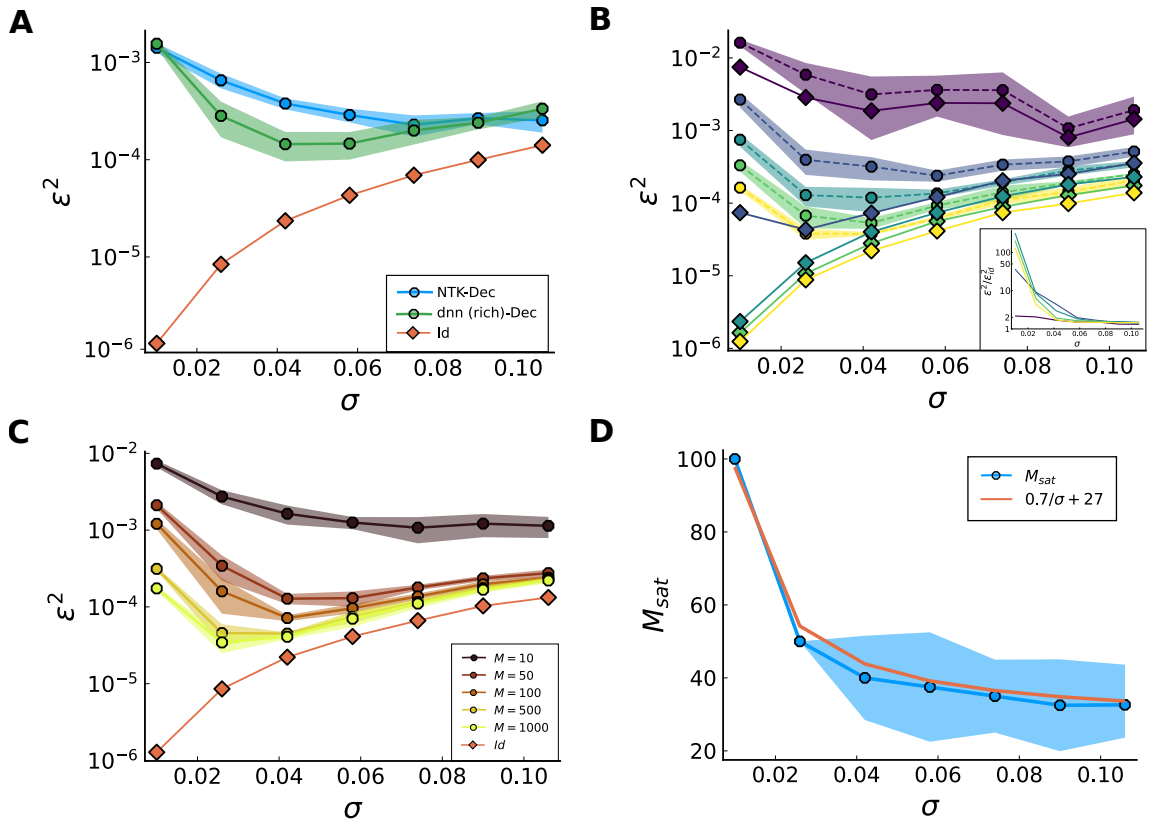

Fig. 2.5 ***Decoding performance of the network decoder trained in the rich regime.*** *In all simulations $\eta = 0.3$. (**A**) Generalization error of the NTK decoder (blue curve), network decoder trained in the rich regime (green curve) and ideal error (red curve) as a function of σ ($N = 50$). The decoders are trained on the same dataset of size $P = 10000$. (**B**) Generalization error of the network decoder (dots and dashed curves) and ideal error (diamonds and solid curves) as a function of σ, for different population sizes, N, colored according to the legend in Fig. 2.4 ($M = 1000$, $\gamma = 3$). Inset: ratio between non-ideal and ideal error as a function of σ. (**C**) Generalization error of the network decoder and ideal error with different number of hidden-layer neurons, M, as a function of σ ($N = 50$). (**D**) Same data of panel C. Number of hidden-layer neurons necessary to saturate the decoding performance of the network decoder as a function of σ (blue curve) and numerical fit with an hyperbolic function (red curve).*

to *decode* the information conveyed by neural activity patterns.

2.3 Discussion

Efficiency as balance between coding and decoding. We quantified the coding properties of a neural population from the point of view of a non-ideal decoder. We considered a fairly general coding scheme with neurons with complex neural responses, in which the only constraint was the level of smoothness (i.e., the two points correlation) of the tuning

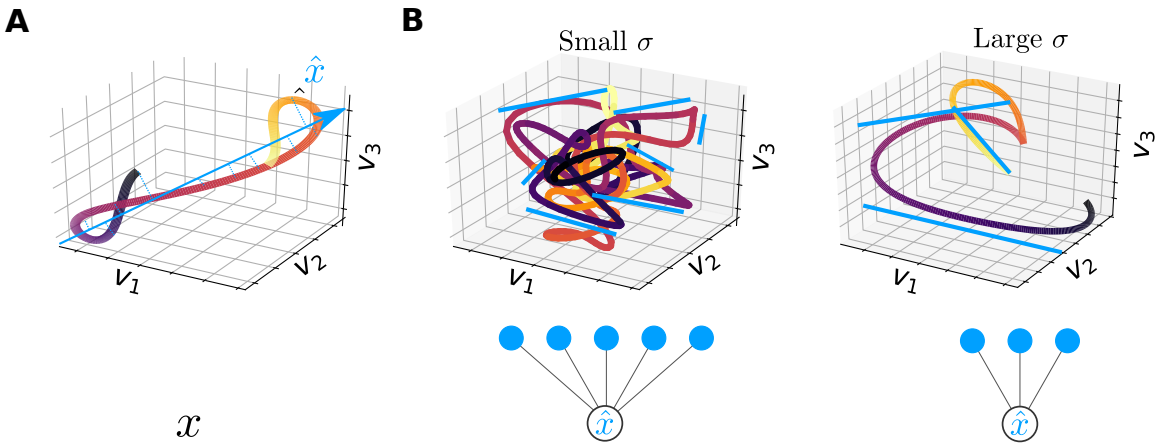

Fig. 2.6 ***Geometrical perspective on different decoders.*** *The axes represent the neural activity of $N = 3$ samples neurons, As a function of the stimulus (colored according to the legend), the mean population activity describes a curve in the 3-dimensional space. (**A**) A linear decoder is the optimal vector in the space of neural population activity (blue) such that the stimulus estimate can be read by projecting the neural activity onto it. When σ is large, as in the figure, it is simple to align the decoding vector to curve of mean population activity. (**B**) The curve of population activity can be decomposed in $\sim 1/\sigma$ uncorrelated segments. As a neural network computes M non-linear 'features' of the input, a possible decoding strategy could be to learn a different 'feature' for each segment (blue segments) and then combine these features to obtain a stimulus estimate. A curve with small σ (left) requires a higher number of hidden neurons than a curve with large σ (right), as confirmed by the results in Fig. 2.5.*

curves. In the limit of large population sizes, the highest ideal accuracy is achieved when neurons exhibit irregular and non-smooth tuning curves. Indeed, in a code which associates stimuli to uncorrelated random points in a high-dimensional space, each stimulus can be decoded unambiguously if the noise variance is smaller than a threshold [60]. More generally, in the high signal-to-noise ratio, maximizing the mutual information between stimuli and neural responses corresponds to minimizing the redundancy and decorrelating neural outputs [8, 114, 43, 115]. (Depending on the the level of the noise and on the constraints on the neural resources, however, a certain degree of redundancy might be optimal [44, 116].) Similarly, in a population of neurons with bell-shaped tuning curves, in the asymptotic limit of large population sizes, the minimum coding error is achieved with infinitely narrow tuning curves (and, thus, uncorrelated neural responses) [11, 12, 61, 25].

However, these theoretical predictions partially contrast with experimental evidence. Neural activity vary smoothly as a function of features of external stimuli, and it has been suggested that neural responses are confined to low-dimensional subspaces, or 'manifolds', in the high-dimensional space of neural activity [117, 53, 118]. Theoretically, it is unclear the role that such 'smoothness' plays in determining the coding performance of the population [72, 119].

Recently, a line of theoretical studies showed that smooth and correlated neural responses,

achieved with broad tuning curves, confer a high 'sample efficiency' to a downstream linear decoder [120, 81, 99]. Here, 'sample efficiency' is defined as the ability to learn a target function in response to neural activity patterns from a limited set of input-output examples. This can be explained through the properties of the kernel associated to the neural code, i.e., the covariance between neural responses to different stimuli, which in our case is determined by $\bar{k}$ in Eq. (2.1). As we showed, the generalization error of a linear decoder is determined by the properties of this kernel. When tuning curves are broad, the kernel spectrum possess a high amount of power in the first 'modes', i.e., the first eigenfunctions in Eq. (2.67) are associated with larger eigenvalues (Fig. S2.1). In a regression task, the components of the target function associated with these first modes (i.e., the coefficient $\bar{w}_i$ in Eq. (2.73)) require a smaller number of examples to be learned, as compared to the components associated to higher modes [80]. Since to similar stimuli are often associated similar behavioral outputs, behaviorally relevant target functions are supposed to be smooth [98], Thus, the spectral properties of the kernel associated to the neural code and the characteristics of behaviorally relevant target functions determine the capacity of the readout to learn quickly and generalize well with a limited number of training samples [121, 81] .

These results were obtained in a noise-free setting. In this work, by studying a special case of target function–the reconstruction of the input–we argue that correlated and smooth neural codes facilitate learning in the presence of noise. Adding noise to training data is a popular technique in machine learning to avoid overfitting and improve generalization [122]. Indeed, it can be shown that adding noise to inputs is equivalent to adding a regularization which penalizes the Jacobian of the decoding function, effectively enforcing smoothness [123]. Our task, reconstructing the input from an intermediate representation, is analog to the one of autoencoders, in which an encoder and a decoder are trained jointly to extract data 'features' in the intermediate representation [124]. In Refs. [125, 126] it has been suggested that adding noise to the input results smoother features, and thus more robust to small variations in the data, in the intermediate layer. Conversely, in this work we show that smooth intermediate representations are more robust to noise when the decoding process is non-ideal.

Despite these similarities, our framework differs from the classical machine learning setting, in that we also test the network on noisy corrupted activity patterns (while, typically, the test set is noise-free). This results in a balance between two instances. On one hand, irregular tuning curves minimize the minimum error which is achieved through an *ideal* stimulus estimate. On the other hand, noise in the training examples biases algorithms towards learning smooth decoding functions, which do not take advantage of small-scale irregularities. Whether a smooth or an irregular neural code is preferable becomes thus a quantitative question, which depends on the decoding architecture.

The role of different architectures. The structure of the Bayesian decoder have been previously employed in Ref.[21] to obtain an approximation of the posterior distribution and, consequently, of the minimum-MSE estimator. It has been named 'Bayesian population vector' due to the similarity of Eqs .(2.28)-(2.30) with the equation of the population vector decoder defined in Ref. [127] as a readout for neurons with monomodal tuning curves with preferred stimuli $\{x_n\}_{n=1}^N$,

$$\hat{x} = \frac{\sum_{n=1}^N x_n r_n}{\sum_{n=1}^N r_n}. \tag{2.17}$$

Our results suggest a possible way to learn such decoder. Interesting, learning is successful even in the presence of global errors, when the identity of the stimulus is ambiguous and the true posterior distribution is multimodal (as in Fig. 2.1). The full posterior distribution represented in the hidden layer might be useful in many other computations which require to manipulate probabilities [128, 129] (see also next Chapter 3). However, given the particular assumption about the training loss, it is worth considering also architectures which are optimized by minimizing an error-based loss function, as postulated by classical theories on the mechanisms of supervised learning in the brain [130, 131].

Linear decoders, computing a weighted combination of neural activities, are diffused tools in the analyses of neural codes, and a number of theoretical results on their performance have been obtained [127, 23, 65, 132, 81]. In the case of neural responses affected by correlated noise, a 'local' linear decoder saturates the lower bound to the MSE imposed by the Fisher information [133, 92]. Here, 'local' means that it is trained to distinguish between two nearby stimuli from the elicited neural responses, implying that, in general, the optimal weights might be different for different regions of the stimulus space. Here, in turn, we investigated the properties of a *global* linear estimator, which operates on the whole range of stimuli. We showed that such decoder achieves the worse performance when tuning curves are highly irregular and it is unable to exploit the high ideal accuracy of complex tuning curves. It is possible to get an intuitive understanding of this behavior from a geometric picture, by representing the joint mean activity of the N neurons as a function of the stimulus as a curve in a N-dimensional space. The weights of the decoder define the optimal N-dimensional vector such that it is possible to read the stimulus estimate from the linear projection of the neural activity onto it. In the case of broad tuning curves, the curve of population activity is smoother and exhibits a lower intrinsic dimensionality (see Chapter 1, Fig. 1.7), facilitating its alignment with the vector (Fig. 2.6A).

Linear decoders are often employed to show that sensory information is represented in the neural activity in an easily decodable way, and sometimes optimality principles are derived under the assumption that the readout is linear [54, 69, 134]. At the end of information processing hierarchy, right before behavioral outputs, such as a choice or a motor command, information must surely be represented in an explicit and easily accessible way. In other regions, however, neural representations can be arranged according to other optimality principles, such as the sparsity of neural activity or the compression of information, relying on subsequent stages of the hierarchy to perform the decoding process. As an example, to read the spatial information contained in the activity of grid cells one necessitates of a decoding algorithm which hierarchically combines the information at different scales coming from different modules [135]. It is therefore important to investigate the properties of more complex architectures. Deep neural networks, due to their (theoretical) capacity of approximate any function [94], represent a good candidate to model flexible and powerful decoders [95].

In many cases, the performance of neural networks in the more tractable lazy regime exhibits a large gap with the state-of-the-art, which depends on the data distribution and on the target function [106]. In the task considered here, estimating the stimulus from a noisy high-dimensional representation, although the input data are N-dimensional, there is clearly a one-dimensional structure in the curve of population activity. The complexity of this low-dimensional structure, defined by the tuning width, affects the complexity of the target function, as irregular tuning curves also imply an irregular decoding function. A neural network trained in the lazy regime learns a function which is a linear combination of a

set of non-linear functions (features) of the input data which are defined by the parameters at initialization, Eq. (2.82), and thus it does not learn new 'features' (see Ref. [111] for a formal discussion). As these features are random and not adapted to the specific problem, the decoding capacity is impaired when the input data possess a complex structure.

Recently, Damian et al. [136] showed that neural networks trained in the rich regime outperform their associated kernel machine when the target function depends only on variations of the input data along a few number of relevant directions. This result gives an intuitive explanation of the gap between lazy and rich regime in our context. Indeed, in the latter we are able to discover the low-dimensional structure of the population activity even in the presence of noise, as also observed empirically in Ref. [137]. An intuition about the fact that the network decoder learns useful features of the data is given by the saturation of its performance beyond a given number of hidden neurons, M_{sat}, which scales empirically as $1/\sigma$; such scaling has a simple geometrical interpretation (Fig. 2.6,B). Indeed, we can imagine to divide the curve of population activity in $\sim 1/\sigma$ correlated 'segments', as also done in the calculations of Chapter 1 to compute the global error. A possible decoding strategy, then, is to learn a local decoder for each of these segments in the hidden layer, and combine their outputs to obtain a final estimate. The nature of the representation in the hidden layer, corresponding to the features extracted by the network, is an interesting object of investigation, which we leave for future research.

2.4 Methods

Throughout, we denote vectors by bold letters, e.g., $\mathbf{r} = (r_1, ..., r_N)$, and matrices by capital letters, e.g., W. We denote as $\langle f(z) \rangle_z$ the expectation of a function f of a random variable z, distributed according to $p(z)$, $\langle f(z) \rangle_z = \int dz p(z) f(z)$.

2.4.1 Data distribution

We consider a population of N neurons each responding to a continuous scalar stimulus, x, sampled from a uniform distribution in the interval $[-0.5, 0.5]$ [3]. The tuning curve which describes the mean response of neuron i as a function of the stimulus is sampled from a Gaussian process,

$$v_i(\cdot) \sim \mathcal{GP}\left(0, \bar{k}\left(\cdot, \cdot\right)\right), \tag{2.18}$$

with vanishing mean and Gaussian covariance function, or kernel,

$$\bar{k}(x, x') = a \exp\left(-\frac{(x - x')^2}{4\sigma^2}\right). \tag{2.19}$$

Here, σ is the tuning width, which controls the length scale of correlation in the process and a is an amplitude coefficient. The notation in Eq. (2.18) indicates the following: the mean responses of neuron i to a set of stimuli, $\{x_1, ..., x_n\}$, are distributed according to a multivariate Gaussian,

$$\begin{bmatrix} v_i(x_1) \\ \vdots \\ v_i(x_n) \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} \bar{k}(x_1, x_1) \dots \bar{k}(x_n, x_1) \\ \vdots \ddots \\ \bar{k}(x_1, x_n) \dots \bar{k}(x_n, x_n) \end{bmatrix}\right). \tag{2.20}$$

In each trial, the responses of neurons are corrupted by independent Gaussian noise of variance equal to η^2; the neural activity pattern, given a stimulus, x, is obtained as

$$\mathbf{r}(x) = \mathbf{v}(x) + \mathbf{z}, \tag{2.21}$$

where $z_i \sim \mathcal{N}(0, \eta^2)$ and $\mathbf{v}(x) = (v_1(x), ..v_N(x))$. In what follows, we will often make implicit the dependence of $\mathbf{r}$ on the stimulus, x. The value of the amplitude a is fixed to 1, such that the signal-to-noise ratio, defined as the ratio between the variance of the responses (the diagonal elements of the covariance matrix) and the noise variance, is constant as a function of the tuning width and equal to $1/\eta^2$ [4].

Given the statistics of the noise, the joint distribution of noisy neural responses as a function of the stimulus can be viewed as a Gaussian process as well,

$$r_i(\cdot) \sim \mathcal{GP}\left(0, k(\cdot, \cdot)\right), \tag{2.22}$$

with an effective covariance function,

[3] This choice, which differs from the 0-1 interval used in Chapter 1, simplifies some analyses, still preserving the generality of the results.

[4] In the limit of $\sigma \ll 1$, this constraint and the constraint adopted in Chapter 1, by fixing the variance of responses across the stimulus space, yield equal values of the gain and Eq. (1.19)-(2.19) become equivalent.

$$k(x, x') = \bar{k}(x, x') + \eta^2 \tilde{\delta}(x, x'), \tag{2.23}$$

where, in this notation, $\tilde{\delta}(x, x')$ is a white noise kernel, which is equal to 1 when x and x' are the same stimulus at the same trial. The noise therefore adds a constant term proportional to η^2, also called a 'ridge', to the diagonal elements of the covariance matrix, leading to the joint distribution of noisy neural responses to a set of stimuli

$$\begin{bmatrix} r_i(x_1) \\ \vdots \\ r_i(x_n) \end{bmatrix} \sim \mathcal{N}\left(0, \bar{K} + \eta^2 I\right), \tag{2.24}$$

where $\bar{K}_{ij} = \bar{k}(x_i, x_j)$ as in Eq. (2.20).

Note that, although Eq. (2.22) describes correctly the statistics of noisy neural responses, it does not distinguish between the stochasticity due to random tuning curves and the trial-to-trial variability. In Chapter 1, the stochasticity of the tuning curves is the result of a set of random synaptic weights, which we assume to be fixed for a given neural circuit. This variability should be treated as a 'quenched' disorder, as opposed to the 'annealed' disorder of the trial-to-trial variability. We make this difference explicit by denoting as $p_v(\mathbf{r}, x)$ the joint distribution of stimuli and neural responses given a realization of a set of random tuning curves, $\{v_i\}$. In practice, this probability distribution is obtained by first sampling N tuning curves from Eq. (2.18) (i.e., sampling the mean neural responses to a set of stimuli corresponding to a fine discretization of the stimulus space, $\{\mathbf{v}(x_1), ..., \mathbf{v}(x_n)\}$), and then considering the conditional trial-to-trial variability, $p_v(\mathbf{r}|x_i) = \mathcal{N}(\mathbf{v}(x_i), \eta^2 I)$ for $i = 1, ..., n$.

2.4.2 Learning from examples

We consider a non-ideal decoder as a function learned in a supervised setting. We generate a dataset, $\mathcal{D}_v = \{\mathbf{r}^i, x_i\}_{i=1}^P$, which consists in P pairs of stimulus-activity pattern sampled independently from the joint distribution $p_v(\mathbf{r}, x) = p_v(\mathbf{r}|x)p(x)$. The task consists in learning a parametric function which takes as input an activity pattern and outputs an estimate of the stimulus, $\hat{x} = f_\theta(\mathbf{r})$. The parameters of the decoder, θ, are set so as to minimize a loss function defined on the dataset (a principle which is sometimes called Empirical Risk Minimization [138]),

$$\hat{\theta} = \arg\min_\theta \mathcal{L}\left(f_\theta, \mathcal{D}_v\right). \tag{2.25}$$

The decoding performance is measured by the squared error of the stimulus estimate averaged over the joint distribution of stimuli-activity patterns, also called generalization error (or MSE),

$$\varepsilon^2(\mathcal{D}_v) = \int d\mathbf{r} dx p_v(\mathbf{r}, x) \left(f_{\hat{\theta}}\left(\mathbf{r}; \mathcal{D}_v\right) - x\right)^2, \tag{2.26}$$

where $\{v_i\}$ denotes the dependence of this quantity on the specific realization of the set of tuning curves. The coding performance of a decoder are measured by averaging this quantity over the distribution of datasets and the possible realizations of tuning curves,

$$\varepsilon^2 = \left\langle \left\langle \varepsilon^2(\mathcal{D}_v) \right\rangle_{\mathcal{D}_v} \right\rangle_{\{v_i\}}. \tag{2.27}$$

2.4.3 Bayesian decoder

Neural architecture. We consider a two-layer neural network with a hidden layer of size M and a readout neuron which outputs the stimulus estimate. We mimic the structure of the ideal decoder, see Methods of Chapter 1, by treating the hidden layer as a multi-class classifier, returning the probability that the input pattern, $\mathbf{r}$, belongs to one-out-of-M classes. The output of the m-th hidden layer neuron is modeled as

$$h_m(\mathbf{r}) = \mathcal{S}(\mathbf{u}) = \frac{\exp(u_m)}{\sum_{m'=1}^{M} \exp(u_m)}, \tag{2.28}$$

where $\mathcal{S}(\cdot)$ is the softmax function, commonly used in machine learning [100] in the final layer of classifiers. The vector $\mathbf{u}$, which can be interpreted as a vector of pre-synaptic currents, is obtained as a biased linear combination of the activity patterns,

$$\mathbf{u} = \lambda \mathbf{r} + \mathbf{b}, \tag{2.29}$$

with λ a $M \times N$ matrix and $\mathbf{b}$ a vector of biases. We assume the M classes to represent a discretization of the stimulus space into M bins, and we assign to each neuron of the hidden layer the midpoint of the m-th bin, x_m, as its 'preferred stimulus.' We then interpret the output of the classifier, h_m, as the probability with which the input pattern $\mathbf{r}^i$ is elicited by a stimulus belonging to the m-th bin, $x_i \in (x_m - 1/2M, x_m + 1/2M)$; the softmax function ensures that these probabilities sum up to 1. We denote by $q(x_m|\mathbf{r}) = h_m(\mathbf{r})$ the resulting discrete posterior distribution. By assuming that $q(x_m|\mathbf{r})$ is a good approximation of the true (discretized) posterior distribution, $p(x_m|\mathbf{r})$, we can obtain the mean of the posterior, which corresponds to the minimum-MSE estimate, by weighting the output of the M neurons according to their preferred positions,

$$\hat{x} = \sum_{m=1}^{M} x_m h_m(\mathbf{r}). \tag{2.30}$$

Loss function. We assume the set of preferred stimuli, x_m, to be fixed. The training parameters of the network are the weights and biases in Eq. (2.29), $\theta = \{\lambda, \mathbf{b}\}$. Given the interpretation of the hidden layer activity as a posterior distribution over stimuli, we train the parameters to reproduce the correct posterior. We start by considering the approximation of the posterior distribution, $q(x|\mathbf{r})$, obtained as $M \to \infty$, and we quantify its distance from the true posterior, $p(x|\mathbf{r})$ through the Kullback-Leibler divergence (KL),

$$\mathrm{KL}\left(p(x|\mathbf{r})||q(x|\mathbf{r})\right) = \int dx p_v(x|\mathbf{r}) \log\left(\frac{p(x|\mathbf{r})}{q(x|\mathbf{r})}\right); \tag{2.31}$$

by definition, this quantity is non-negative and vanishes only when the two distributions are identical. We consider this quantity averaged over the distribution of neural activity patterns,

$$\left\langle \mathrm{KL}\left(p\left(x|\mathbf{r}\right)||q\left(x|\mathbf{r}\right)\right)\right\rangle_{p_v(\mathbf{r})} = -\int d\mathbf{r} dx p_v(\mathbf{r}, x) \log q(x|\mathbf{r}) - H(x|\mathbf{r}); \tag{2.32}$$

the second term is the conditional entropy of the stimulus given the neural response, a quantity which does not depend on the decoder parameters. By using the dataset, $\mathcal{D}_v = \{\mathbf{r}^i, x_i\}_{i=1}^P$, to approximate the integral, we obtain

$$\begin{aligned} -\int d\mathbf{r}dx p_v(\mathbf{r},x)\log q(x|\mathbf{r}) &\approx -\frac{1}{P}\sum_{i=1}^{P}\log q(x_i|\mathbf{r}^i) \\ &\approx -\frac{1}{P}\sum_{i=1}^{P}\log q(x_{m(i)}|\mathbf{r}^i) \\ &= -\frac{1}{P}\sum_{i=1}^{P}\log\left(h_{m(i)}(\mathbf{r}^i)\right), \end{aligned} \tag{2.33}$$

where the second approximation comes from the discretization of the posterior into m bins, by defining $m(i) = m : x_i \in [x_m - 1/2M, x_m + 1/2M]$. The quantity which appears in the last line is the loss function we wrote in the main text Eq. (2.6); thus, a neural network classifier estimates the posterior probability [139]. Once the parameters are learned by minimizing the loss function,

$$\hat{\theta} = \arg\min_{\theta}\left\{-\frac{1}{P}\sum_{i=1}^{P}\log\left(h_{m(i)}(\mathbf{r}^i)\right)\right\}, \tag{2.34}$$

we quantify the decoding performance by measuring the MSE, Eq. (2.26), given the stimulus estimate as obtained in Eq. (2.30). The loss function yields a simple update rule for the synaptic weights. The gradient of the loss calculated at a single input pattern, $\mathcal{L}(\mathbf{r}^i) = -\log\left(h_{m(i)}(\mathbf{r}^i)\right)$, with respect to the synaptic weight between the n-th input neuron and the m-th hidden-layer neuron, is obtained as

$$\frac{\partial\mathcal{L}(\mathbf{r}^i)}{\partial\lambda_{mn}} = \begin{cases} r_n(h_m - 1) & \text{if } m \equiv m(i) \\ r_n h_m & \text{otherwise} \end{cases}, \tag{2.35}$$

leading to the gradient-based update rule (one-sample Stochastic Gradient Descent)

$$\lambda_{mn}^{t+1} = \lambda_{mn} - \eta_l\left(r_n(h_m - \delta_{m,m(i)})\right), \tag{2.36}$$

where η_l is the learning rate.

2.4.4 Error-based decoder: general setting

We consider a two-layer neural network performing a regression task. Given an activity pattern, $\mathbf{r}$, as input, the stimulus estimate is obtained as

$$\hat{x} = f_\theta(\mathbf{r}) = \mathbf{w}_{(1)}^T\mathbf{h}(\mathbf{r}) + b_{(1)}, \tag{2.37}$$

where $\mathbf{h}$ is the vector output of the hidden-layer neurons,

$$\mathbf{h}(\mathbf{r}) = f_{(0)}\left(W_{(0)}\mathbf{r} + \mathbf{b}_{(0)}\right). \tag{2.38}$$

Here, the training parameters are $\theta = \{\mathbf{w}_{(0)}, W_{(1)}, \mathbf{b}_{(0)}, b_{(1)}\}$, where $W_{(0)}$ is the $M \times N$ matrix of synaptic weights from the N representation neurons to the M hidden-layer neurons, $\mathbf{w}_{(1)}$

the $1 \times M$ vector of weights from the M neurons to the readout neuron, $b_{(1)}$, $\mathbf{b}_{(0)}$ are biases and $f_{(0)}$ is a (non-linear) function applied pointwise. We set the parameters of the network so as to minimize the MSE of the stimulus estimate on the dataset,

$$\hat{\theta} = \arg\min_{\theta} \left\{ \frac{1}{P} \sum_{i=1}^{P} \left(f_\theta(\mathbf{r}^i) - x_i \right)^2 \right\}, \tag{2.39}$$

and we quantify the decoding performance by measuring the generalization error, Eq. (2.26).

2.4.5 Linear decoder

The general case reduces to the case of a linear decoder,

$$\hat{x} = f_l(\mathbf{r}) = \frac{1}{\sqrt{N}} \mathbf{w}^T \mathbf{r} + b, \tag{2.40}$$

when $f_{(0)}$ is the identity function, $\mathbf{w}^T = \mathbf{w}_{(1)}^T W_{(0)}$ and $b = \mathbf{w}_{(1)}^T \mathbf{b}_{(0)} + b_{(1)}$. In order to make the problem analytically tractable, we consider the regularized minimization problem (ridge regression),

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \left\{ \frac{1}{P} \sum_{i=1}^{P} (\mathbf{w}^\mathbf{T} \mathbf{r}^i + b - x_i)^2 + \lambda \left\| \mathbf{w} \right\|_2^2 \right\}, \tag{2.41}$$

where a penalty on the magnitude of the weights is added; ultimately, we will consider the limit $\lambda \to 0$. Equation (2.41) admits the solution

$$\hat{\mathbf{w}} = R \left(R^T R + \lambda I \right)^{-1} \bar{x}, \tag{2.42}$$

and

$$\hat{b} = \hat{\mathbf{w}}^T \langle \mathbf{r}_i \rangle - \langle x_i \rangle \tag{2.43}$$

in terms of the $N \times P$ data matrix, R, with elements $R_{ij} = \frac{1}{\sqrt{N}} r_i^j = \frac{1}{\sqrt{N}} r_i(x_j)$, and the P-dimensional vector of training stimuli, $(\bar{x})_i = x_i$. From now on, we ignore the bias term, as the empirical averages of neural responses, $\langle \mathbf{r}_i \rangle$, and stimuli, $\langle x_i \rangle$, vanish in the limit of many samples. Thus, the optimal linear decoder is obtained as

$$f_l(\mathbf{r}) = \frac{1}{\sqrt{N}} \bar{x}^T \left(R^T R + \lambda I \right)^{-1} R^T \mathbf{r}. \tag{2.44}$$

Based on this form, it is possible to study the generalization error of the linear decoder by exploiting a connection with a machine learning approach for learning functions: kernel methods [108].

Overview on kernel ridge regression. We provide a brief overview of kernel ridge regression; we refer the reader to Ref. [108] for more details. The task of kernel regression is to select a function which belongs to a specific space of functions, called Reproducing Kernel Hilbert Space (RKHS), defined as follows. Given a Hilbert space of functions, $\mathcal{H}$, equipped with an inner product, $\langle \cdot, \cdot \rangle_{\mathcal{H}}$, this space is a RKHS if there exists a function which maps elements of a set, $y \in \mathcal{Y}$, to functions in $\mathcal{H}$, denoted as $\kappa(\cdot, y)$, such that their inner product

with a function f evaluates the function at y, $\langle \kappa(\cdot, y), f \rangle_{\mathcal{H}} = f(y)$. The reproducing kernel for the space $\mathcal{H}$ is defined as the function $\mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ obtained as

$$\kappa(y, y') = \langle \kappa(\cdot, y), \kappa(\cdot, y') \rangle_{\mathcal{H}}. \tag{2.45}$$

Conversely, a kernel $\kappa(\cdot, \cdot)$, if it is symmetric and positive definite, can be used to define a unique RKHS (Moore–Aronszajn theorem).

These spaces of functions find numerous applications in machine learning and statistics: here, we focus on the problem of regression. In kernel ridge regression, given a dataset, $\mathcal{D} = \{\mathbf{z}^i, y_i\}_{i=1}^P$, where $\mathbf{z} \in \mathbb{R}^I$ and $y \in \mathbb{R}$, and given a RKHS, $\mathcal{H}$, with kernel, κ, the task is to select the function $f \in \mathcal{H} : \mathbb{R}^I \to \mathbb{R}$ that minimizes the regularized loss,

$$\arg\min_{f \in \mathcal{H}} \left\{ \frac{1}{P} \sum_{i=1}^{P} \left(f(\mathbf{z}^i) - x_i \right)^2 + \lambda \left\| f \right\|_{\mathcal{H}}^2 \right\}, \tag{2.46}$$

where $\|f\|_{\mathcal{H}}^2 = \langle f, f \rangle_{\mathcal{H}}$. This minimization problem is convex and the optimal function is obtained in closed form as

$$f(\mathbf{z}) = \bar{y}^T \left(K + \lambda I \right)^{-1} \mathbf{k}(\mathbf{z}), \tag{2.47}$$

where $K_{ij} = \kappa(\mathbf{z}^i, \mathbf{z}^j)$ is the kernel Gram matrix and $\mathbf{k}(\mathbf{z})_i = \kappa(\mathbf{z}^i, \mathbf{z})$.

The $N \to \infty$ limit. We now show that, in the limit $N \to \infty$, the solution to the linear regression problem, in which inputs are N-dimensional vectors, corresponds to the solution to a kernel regression problem with the input defined in the one-dimensional stimulus space. In this limit, the elements of the empirical covariance matrix in Eq. (2.42), $(R^T R)_{ij} = \frac{1}{N} \sum_{l=1}^{N} r_l(x_i) r_l(x_j)$, converge, by definition of the Gaussian process, to the elements of the kernel Gram matrix, $K_{ij} = k(x_i, x_j)$, with k defined in Eq. (2.23). Analogously, the product between the data matrix and a neural activity pattern, $\mathbf{r}$, evoked by a stimulus, x, $\frac{1}{\sqrt{N}} (R^T \mathbf{r})_i = \frac{1}{N} \sum_{l=1}^{N} r_l(x_i) r_l(x)$, converges to $\mathbf{k}(x)_i = k(x_i, x)$. Thus, the decoding function, Eq. (2.44), can be written as

$$\begin{aligned} f_l(\mathbf{r}) = f_l(\mathbf{r}) &= \frac{1}{\sqrt{N}} \bar{x}^T \left(R^T R + \lambda I \right)^{-1} R^T \mathbf{r}(x) \\ &\approx \bar{x}^T (K + \lambda I)^{-1} \mathbf{k}(x). \end{aligned} \tag{2.48}$$

The expression in Eq. (2.48) is equivalent to the solution of a kernel regression problem, Eq. (2.47), with $\mathbf{z} = x$, $y = x$ and $\kappa = k$ [5].

From the definition of k, Eq. (2.23), we can see that the noise results in adding a constant term proportional to η^2 to the regularization coefficient, λ, therefore making it non-vanishing even when $\lambda \to 0$. Interestingly, the noise does not contribute to the term $\mathbf{k}(x)$, which represents the overlap between the activity patterns in the dataset and $\mathbf{r}$. An intuitive reason for this is that, in the limit $N \to \infty$, since the decoder is linear and the noise is independent across neurons, the noise is averaged out. An analysis of how the properties of the kernel (here, determined by the value of σ) and the regularization coefficient affect the the generalization error as a function of the number of samples, P, is reported in Ref. [80]. Here,

[5] Linear regression can be seen as a particular case of kernel ridge regression when the kernel is a dot-product kernel; here, due to our assumptions on $\mathbf{r}$, the dot-product kernel defined on the space of neural responses converges to the Gaussian kernel defined in the stimulus space, $k_{lin}(\mathbf{r}^i, \mathbf{r}^j) \propto (\mathbf{r}^i)^T \mathbf{r}^j = k(x_i, x_j)$.

we focus on the non-trivial case in which N is finite, and the noise affects the generalization error even in the limit of large P.

Implicit regularization due to neural noise and finite population size. The previous calculations show that we can obtain an approximation of the kernel regression function in the following way. First, we evaluate N independent Gaussian processes ('features') with covariance function equal to the kernel evaluated at the training points (which, in the previous case, correspond to stimuli $\{x_1, ..., x_P\}$). Then, we compute the optimal linear coefficients, $\hat{\mathbf{w}}$, on the set of N-dimensional features, $\mathbf{r}(x_1), ..., \mathbf{r}(x_P)$. The linear function, $f(\mathbf{r}(x)) = \hat{w}^T\mathbf{r}(x)$, approximates the kernel regression solution in the limit $N \to \infty$. This example is a particular instance of a deeper connection between random functions and kernel methods. Indeed, by relying on asymptotic behaviors of the type described above, random functions can be used to approximate kernel methods and reduce their computational burden (kernel methods require the inversion of a $P \times P$ matrix, while the computation of the optimal regression coefficient requires the inversion of a $N \times N$ matrix) [140, 141] . When the number of random functions, N, is finite, differences from the asymptotic case emerge. Jacot et al. [103] studied a model with assumptions on the distribution of the data which bear similarity to our setting. Here, in order to obtain an analytical expression for the generalization error, we first summarize their results and calculations; next, we adapt them to the case of noisy neurons in the limit of many training samples.

Expression for the generalization error. In Ref. [103] the following setting is considered. The set of stimulus samples, $\bar{x}$, is assumed to be fixed, and the data matrix, R, is obtained by sampling the rows from N realizations of a Gaussian process, Eq. (2.22), evaluated at P stimulus values. This yields the same statistics of the data matrix as in our problem; as the stimuli are kept fixed, the only source of stochasticity in the dataset, $\mathcal{D} = \{R, \bar{x}\}$, is in the data matrix, R. In the asymptotic limit of large P, we can make a similar assumption also in our case (the dimensionality of the stimulus space is much smaller than the dimensionality of the neural activity space). During the test phase, to compute the generalization error, neural responses, $\mathbf{r}$ to a stimulus, x, are sampled from the conditional Gaussian distributions, obtained under the Gaussian process assumption, Eq.(2.24),

$$\begin{aligned} p(\mathbf{r}|x) &= \prod_{i=1}^{N} p(r_i|x) \\ &= \prod_{i=1}^{N} \mathcal{N}(\mu_n(x), \sigma_n^2(x)), \end{aligned} \tag{2.49}$$

where

$$\mu_n(x) = R_{i,:}K^{-1}\mathbf{k}(x), \tag{2.50}$$

$$\sigma_n^2(x) = k(x,x) - \mathbf{k}(x)^T K^{-1}\mathbf{k}(x), \tag{2.51}$$

with $R_{i,:}$ denoting the i-th row of the matrix R, $K_{ij} = k(x_i, x_j)$ and $(\mathbf{k}(x))_i = k(x_i, x)$ as before. The generalization error is then computed by averaging the MSE over the distribution of stimuli and neural responses, and over the stochasticity in the dataset,

$$\varepsilon_g^2 = \left\langle \left\langle (f_l(\mathbf{r}|R) - x)^2 \right\rangle_{\mathbf{r},x} \right\rangle_R, \tag{2.52}$$

with f_l defined as in Eq. (2.44), and we made explicit its dependence on the data matrix, R. We note that, with respect to the definition of ε^2 in Eqs. (2.26)-(2.27), in this case, in the dataset, we make no difference between the trial-to-trial variability due to noise and the stochasticity due to random tuning curves. Moreover, in this formulation, the probability of an activity pattern, $\mathbf{r}$, given a stimulus, x, (i.e., the inner average in Eq. (2.52)) depends on the whole dataset (which consists in noisy samples) and not only on the distribution of tuning curves. However, as we show in Sec. S3.5, in the asymptotic limit of large P, the average over activity patterns and stimuli calculated in this way becomes equivalent to the average over their joint distribution for a given set of tuning curves, $p_v(\mathbf{r}, x)$, as the noise in the training set can be averaged out. Thus, by decomposing the average over the data matrix as the average over different realizations of tuning curves and of the noisy responses in the dataset, $\langle\cdot\rangle_R \approx \langle\cdot\rangle_{\mathcal{D}_v,\{v\}}$, Equation (2.52) and Equations (2.26)-(2.27) become equivalent.

Decomposition of the generalization error. The generalization error, Eq. (2.52), can be decomposed into the sum of three terms,

$$\varepsilon^2 = \langle B(x) + V_1(x) + V_2(x)\rangle_x \,. \tag{2.53}$$

Here, the first term,

$$B(x) = \Big(\langle f_l\left(\mathbf{r}|R\right)\rangle_{\mathbf{r},R} - x\Big)^2 \tag{2.54}$$

is the bias of the mean output of the decoding function (i.e., averaged over different neural responses to stimulus x, $\mathbf{r} \equiv \mathbf{r}(x)$, and over different realizations of the data matrix, R). The second term,

$$V_1(x) = \Big\langle \langle f_l(\mathbf{r}|R)\rangle_{\mathbf{r}}^2 \Big\rangle_R - \langle \langle f_l(\mathbf{r}|R)\rangle_{\mathbf{r}}\rangle_R^2 \,, \tag{2.55}$$

is the variance, over different realizations of R, of output of the decoding function averaged over different neural responses to stimulus x. The last term,

$$\begin{aligned} V_2(x) &= \Big\langle \Big\langle f_l(\mathbf{r}|R)^2 \Big\rangle_{\mathbf{r}} - \langle f_l(\mathbf{r}|R)\rangle_{\mathbf{r}}^2 \Big\rangle_R \\ &= \langle \mathrm{Var}(f_l\left(\mathbf{r}|R\right)\rangle_R \,, \end{aligned} \tag{2.56}$$

is the mean variance (i.e., over different realizations of R) of the output of the decoding function evaluated at responses to the stimulus, x. We argue that, among the three terms in Eq. (2.53), in the case of noisy neural responses and if the number of samples is large, the dominant one is $V_2(x)$. Indeed, in the asymptotic limit of large P we expect the mean output of the decoder, averaged over different noisy responses to stimulus x, to be unbiased and independent on the specific dataset. As a consequence, we expect $B(x)$ and $V_1(x)$ to be negligible as compared to $V_2(x)$, which quantifies the variability of the decoder output driven by the noise in the responses, $\mathbf{r}$, to stimulus x. We checked the validity of this assumption in numerical simulations (Fig. 2.3B). We also note that, in general, $V_2(x)$ constitutes a lower bound to the generalization error.

Calculation of $V_2(x)$. The decoder output, conditioned on the data matrix, R, is a weighted sum of independent Gaussian-distributed random variables, Eqs. (2.40) and (2.42); as a

result, the variance is obtained as

$$\begin{aligned}\mathrm{Var}\left(f_l\left(\mathbf{r}|R\right)\right) &= \frac{1}{N}\sum_{i=1}^{N}\hat{w}_i^2\mathrm{Var}\left(r_i(x)|R\right)\\ &= \frac{\|\hat{\mathbf{w}}\|^2}{N}\sigma_n^2(x),\end{aligned} \tag{2.57}$$

where $\sigma_n^2(x)$ is the variance of a neural response, Eq. (2.51). To compute $V_2(x)$, we have to average this quantity over different realizations of R,

$$\begin{aligned}\langle\|\hat{\mathbf{w}}\|^2\rangle_R &= \left\langle \bar{x}^T(R^TR+\lambda I)^{-1}R^TR(R^TR+\lambda I)^{-1}\bar{x}\right\rangle_R\\ &= \bar{x}^T\left\langle R^T(RR^T+\lambda I)^{-2}R\right\rangle_R\bar{x},\end{aligned} \tag{2.58}$$

where we used the push-through identity, $(I+UV)^{-1}U = U(I+VU)^{-1}$ [142]. Central to the calculations developed in [103] are the properties of the so-called general Wishart matrices, $W\Sigma W^T$, with Σ a $N\times N$ symmetric covariance matrix (e.g., a kernel Gram matrix), and W a $P\times N$ matrix with independent realizations of standard Gaussian random variables. The calculations are quite involved; hereafter, we provide a sketch of the reasoning which leads to the final result.

First, we diagonalize the kernel Gram matrix as $K = UDU^T$, and we denote by $\{d_i\}_{i=1}^P$, the eigenvalues of K which constitute the diagonal elements of D. According to the Kosambi-Karhunen-Loeve theorem [143], the data matrix R, which contains independent samples from Gaussian processes with kernel k, can be written as $R = \frac{1}{\sqrt{N}}WK^{1/2}$, where $K^{1/2} = UD^{1/2}U^T$ and $W_{ij}\sim\mathcal{N}(0,1)$, such that $\langle R^TR\rangle_W = K$. This allows us to substitute the average over the data matrix, R, with the average over the matrix W. We rewrite Eq. (2.58) as

$$\langle\|\hat{\mathbf{w}}\|^2\rangle_R = \bar{x}^T\left\langle \frac{d}{dz}A(z)\Big|_{z=-\lambda}\right\rangle_W\bar{x}, \tag{2.59}$$

with $A(z)$ defined as

$$A(z) = \frac{1}{N}K^{1/2}W^T\left(WKW^T - zI\right)^{-1}WK^{1/2}. \tag{2.60}$$

We now work, without loss of generality, in the diagonal basis, such that $K = D$. A key result of Ref. [103] is that, in this basis, the expected value of the matrix $A(z)$ is a diagonal matrix with elements

$$\left\langle A(z)_{ii}|_{z=-\lambda}\right\rangle_W \approx \frac{d_i}{d_i+\tilde{\lambda}}, \tag{2.61}$$

with $\tilde{\lambda} = \tilde{\lambda}(z)|_{z=-\lambda}$ defined through the implicit equation

$$\tilde{\lambda} = \lambda + \frac{\tilde{\lambda}}{N}\sum_{i=1}^{P}\frac{d_i}{d_i+\tilde{\lambda}}. \tag{2.62}$$

Therefore, $\left\langle A(z)|_{z=-\lambda}\right\rangle_W = K(K+\tilde{\lambda})^{-1}$. A similar result holds for the derivative of the

matrix, for which we obtain that

$$\begin{aligned}\left\langle \frac{d}{dz}A(z)\right\rangle_W &\approx \frac{d}{dz}\left\langle A(z)\right\rangle_W \\ &\approx -\frac{d\tilde{\lambda}(z)}{dz}K\left(K+\tilde{\lambda}(z)\right)^{-2}\end{aligned} \tag{2.63}$$

Finally, by inserting Eq. (2.63) into Eq. (2.59), we have that

$$\langle\|\hat{\mathbf{w}}\|^2\rangle_R = \frac{d\tilde{\lambda}}{d\lambda}\bar{x}^T K\left(K+\tilde{\lambda}\right)^{-2}\bar{x}, \tag{2.64}$$

and, by combining the result with Eqs. (2.57)-(2.56), we obtain

$$V_2(x) \approx \frac{d\tilde{\lambda}}{d\lambda}\frac{\bar{x}^T M_{\tilde{\lambda}}\bar{x}}{N}\sigma_n^2(x), \tag{2.65}$$

where $M_{\tilde{\lambda}} = K(K+\tilde{\lambda})^{-2}$.

Approximations for large P. The expression above still depends on the specific choice of the dataset, through the eigenvalues of the Gram matrix, K, the stimuli in the dataset, x, and the posterior variance of neural responses, $\sigma_n^2(x)$. Here, we evaluate these quantities in the the large-P limit, in which they depend exclusively upon the properties of the kernel function, k.

First, we write the kernel Gram matrix as $K = \bar{K} + \eta^2 I$, where $\bar{K}_{ij} = \bar{k}(x_i, x_j)$ is the noise-free kernel Gram matrix with $\bar{k}$ as in Eq. (2.19). As a result, the eigenvalues of the matrix K are obtained as $d_i = \bar{d}_i + \eta^2$, with $\bar{d}_i$ being an eigenvalue of $\bar{K}$; the eigenvectors, the columns of the matrix U, are the same for the two matrices. The matrix eigenvalue problem,

$$\bar{K}\mathbf{u}_i = \bar{d}_i\mathbf{u}_i, \tag{2.66}$$

with $\mathbf{u}_i$ the i-th column of U, for $P \to \infty$ approaches an integral eigenvalue problem

$$\begin{aligned}\bar{d}_i^c\phi_i(x) &= \int d'x p(x')\bar{k}(x,x')\phi_i(x') \\ &\approx \frac{1}{P}\sum_{l=1}^{P}\bar{k}(x,x_l)\phi_i(x_l),\end{aligned} \tag{2.67}$$

where $p(x) = \mathcal{U}[-\frac{1}{2},\frac{1}{2}]$ is the distribution of stimuli [82]. The eigenvectors (eigenfunctions) and eigenvalues, respectively, in the two formulations are related through

$$\sqrt{P}(\mathbf{u}_i)_j \approx \phi_i(x_j) \tag{2.68}$$

and

$$P\bar{d}_i^c \approx \bar{d}_i; \tag{2.69}$$

the P-dependent factors arise due to the discretization and the different normalizations of eigenfunctions and eigenvectors. Equation (2.67) is the integral eigenvalue problem for the kernel operator associated with $\bar{k}$, with respect to the probability density $p(x)$; the eigenfunctions $\phi_i(x)$ are orthonormal, i.e., $\int dx p(x)\phi_i(x)\phi_j(x') = \delta_{ij}$. Eigenfunctions and eigenvalues

play a central role in analyses of the properties of a RKHS; indeed, from Mercer's theorem [144]) it is possible to represent the elements of the kernel as a sum of its eigenfunctions, as

$$\bar{k}(x,x') = \sum_{i=1}^{P'} \bar{d}_i^c \phi_i(x)\phi_i(x'), \tag{2.70}$$

with P' the number of solutions to integral problem in Eq. (2.67), potentially infinite. We note the matrix problem yields P eigenvalues, while P' can be infinite, in general. This is indeed the case for a Gaussian kernel, but the eigenvalues are exponentially suppressed after a threshold which depends on the value of σ. By ignoring boundary conditions and assuming translational invariance, we have that $\bar{d}_i^c \sim \exp(-\sigma^2 i^2)$, where i is the rank of the eigenvalue [108, 72] (Fig. S2.1A). Thus, it is sufficient to choose $P \gg 1/\sigma$ to ensure that the non negligible eigenvalues are well approximated.

By applying Mercer's theorem, we can approximate the value of the kernel function at at a stimulus x, $(\bar{\mathbf{k}}(x))_j = \bar{k}(x_j, x)$, as

$$\begin{aligned} \bar{k}(x_j,x) &= \sum_{i=1}^{P'} \bar{d}_i^c \phi_j(x_j)\phi_j(x) \\ &\approx \frac{1}{\sqrt{P}} \sum_{j=1}^{P} \bar{d}_i (\mathbf{u_i})_j \phi_j(x), \end{aligned} \tag{2.71}$$

where we used the approximations in Eqs. (2.68)-(2.69). Equation (2.71) can be written, in vector form, as

$$\bar{\mathbf{k}}(x) = \frac{1}{\sqrt{P}} U \bar{D} \vec{\phi}(x), \tag{2.72}$$

with $\vec{\phi}(x) = \{\phi_i(x)\}_{i=1}^P$ and $\bar{D} = \mathrm{diag}(\bar{d}_1, ..., \bar{d}_P)$; the approximation above goes by the name of Nystrom method [145]. Finally, we note that, by definitions of k and $\bar{k}$, we have that $\mathbf{k}(x) = \bar{\mathbf{k}}(x)$.

We use the approximations derived above to obtain a closed-form expression of Eq. (2.65). The kernel eigenfunctions form an orthonormal basis of functions which are L_2 in the stimulus space [108]. As a result, there exists a set of weights, $\{\bar{w}_i\}_{i=1}^P$, such that we can decompose the target function in this basis, $x = \frac{1}{\sqrt{P}} \sum_{i=1}^P \bar{w}_i \phi_i(x)$, with $x \in [0.5, 0.5]$. In particular, for the vector of stimuli in the training set we have that we have that $\bar{x}^T = \bar{\mathbf{w}}^T U^T$. These weights can be calculated as

$$\bar{w}_i = \sqrt{P} \int dx p(x) x \phi_i(x). \tag{2.73}$$

Figure S2.1 illustrates the behavior of the kernel eigenvalues and eigenvectors, as well as of the optimal weights, as a function of the value of σ. The decomposition of the target function allows us to evaluate the numerator on the r.h.s. of Eq. (2.65), as

$$\begin{aligned} \bar{x}^T M_{\tilde{\lambda}} \bar{x} &\approx \bar{\mathbf{w}}^T U^T K (K + \tilde{\lambda} I)^{-2} U \bar{\mathbf{w}} \\ &= \bar{\mathbf{w}}^T D (D + \tilde{\lambda} I)^{-2} \bar{\mathbf{w}} \\ &= \sum_{i=1}^{P} \frac{(P\bar{d}_i^c + \eta^2)\bar{w}_i^2}{\left(P\bar{d}_i^c + \eta^2 + \tilde{\lambda}\right)^2}, \end{aligned} \tag{2.74}$$

where the first equality is obtained by substituting $K = UDU^T$ and exploiting the fact that $UU^T = I$, while in last equality we substituted $d_i = \bar{d}_i + \eta^2 = P\bar{d}_i^c + \eta^2$. In this limit, we also have that the posterior variance is dominated by the noise variance; indeed, we have that

$$\begin{aligned}\sigma_n^2(x) &= k(x,x) - \mathbf{k}(x)^T K^{-1}\mathbf{k}(x) \\ &\approx k(x,x) - \frac{1}{P}\vec{\phi}(x)\bar{D}U^T(UDU^T)^{-1}U\bar{D}\vec{\phi}(x) \\ &\approx k(x,x) - \frac{1}{P}\sum_{i=1}^{P}\frac{P^2\bar{d}_i^{c^2}}{P\bar{d}_i^c + \eta^2}\phi_i(x)^2,\end{aligned} \tag{2.75}$$

where we exploited the fact that $\mathbf{k}(x) = \bar{\mathbf{k}}(x)$ and we substituted the Nystrom approximation, Eq. (2.72). We now consider the terms up to an index $i' \ll \log(P)/\sigma$, such that we have $P\bar{d}_i^c \gg \eta^2\ P$. For these indices, we have that

$$\frac{P^2\bar{d}_i^{c^2}}{P\bar{d}_i^c + \eta^2} \approx P\bar{d}_i^c, \tag{2.76}$$

while the other indices correspond to terms which are exponentially small, as they scale as $P^2\bar{d}_i^{c^2} \sim P^2\exp(-2i^2\sigma^2)$. Thus, by truncating the sum in Eq. (2.70) at index i', we obtain

$$\frac{1}{P}\sum_{i=1}^{P}\frac{P^2\bar{d}_i^{c^2}}{P\bar{d}_i^c + \eta^2}\phi_i(x)^2 \approx \frac{1}{P}\sum_{i=1}^{i'}P\bar{d}_i^c\phi_i(x)^2 \approx \bar{k}(x,x), \tag{2.77}$$

which results in the posterior variance equal to the noise variance,

$$\sigma_n^2(x) \approx \bar{k}(x,x) + \eta^2 - \bar{k}(x,x) = \eta^2. \tag{2.78}$$

Finally, by evaluating $\frac{d\tilde{\lambda}}{d\lambda}$ as

$$\begin{aligned}\frac{d\tilde{\lambda}}{d\lambda} &= 1 + \frac{1}{N}\frac{d\tilde{\lambda}}{d\lambda}\left(\sum_{i=1}^{P}\frac{d_i}{d_i+\tilde{\lambda}} - \sum_{i=1}^{P}\frac{\tilde{\lambda}}{(d_i+\tilde{\lambda})^2}\right) \\ &= \left(1 - \frac{1}{N}\sum_{i=1}^{P}\frac{d_i^2}{(d_i+\tilde{\lambda})^2}\right)^{-1}\end{aligned} \tag{2.79}$$

and combining Eqs.(2.65), (2.74) and (2.78), we obtain the expression which appears in the main text, Eq. (2.13):

$$\varepsilon^2 \approx \frac{\eta^2}{N - \sum_{i=1}^{P}\left(\frac{d_i}{d_i+\tilde{\lambda}}\right)^2}\sum_{i=1}^{P}\frac{d_i\bar{w}_i^2}{\left(d_i+\tilde{\lambda}\right)^2}, \tag{2.80}$$

with $d_i = P\bar{d}_i^c + \eta^2$. To complete the calculation, $\tilde{\lambda}$ must be evaluated in the limit $\lambda \to 0$, to obtain the implicit Eq. (2.15),

$$\sum_{i=1}^{P}\frac{d_i}{d_i+\tilde{\lambda}} = N. \tag{2.81}$$

2.4.6 Lazy regime

We provide a brief background on the so called 'lazy regime,' in which neural networks 'behave like kernel methods'. By this we mean that the function learned by the network can be expressed as the solution to a kernel regression problem, Eq. (2.47) with $\lambda \to 0$, with k a kernel that depends exclusively on the parameters of the network at initialization. We refer the reader to Refs. [96, 97] for a more rigorous treatment.

We start by considering the Taylor expansion of the output of the network around the initialization parameters,

$$f_\theta(\mathbf{r}) = f_{\theta_0}(\mathbf{r}) + (\theta - \theta_0)^T \nabla_\theta f_\theta(\mathbf{r})|_{\theta_0}, \tag{2.82}$$

where θ is the vector of all the parameters of the neural network, initialized at θ_0, and $\nabla_\theta f_\theta(\mathbf{r})$ is the gradient column vector of the network output with respect to the parameters (Jacobian); as an example, in a two-layer neural network with M hidden neurons, the vector θ has dimension $N_p = (N+1)M + M + 1$.

Equation (2.82) can be regarded as a linear combination of a set of non-linear 'features' of the input defined through the gradient of the output function with respect to the parameters of the network, $\bar{\psi}(\mathbf{r};\theta) = \nabla_\theta f(\mathbf{r})$. Empirically, it has been observed that, in large neural networks (with many parameters) and weights initialized according to a distribution with sufficiently large variance, the network fits the training data remarkably well with minimal variations in the parameters values. More precisely, the criterion for this 'lazy' training to occur is formulated by imposing that the amount of change in the parameters, $\|\theta - \theta_0\|$, required for a substantial decrease in the loss function, causes a negligible change in the Jacobian of the network, $\nabla_\theta f_\theta(\mathbf{r})$ (see [97]). As a result, the feature map defined above, $\bar{\psi}(\mathbf{r};\theta)$, is fixed during training, and equal to its form at initialization. We now consider the loss function, Eq. (2.46), rewritten as

$$\mathcal{L} = \frac{1}{2} \|\mathbf{f} - \bar{x}\|^2, \tag{2.83}$$

where $\mathbf{f} = \{f(\mathbf{r}^i)\}_{i=1}^P$ and $(\bar{x})_i = x_i$; we added a factor of $1/2$ for later convenience and we dropped the divisive constant, P, for the sake of simplicity. When the weights are updated according to the gradient of the loss function with respect to its parameters, and if we consider the limiting case of a vanishing small learning rate, the parameters evolve according to the gradient flow equation, as

$$\begin{aligned} \frac{d\theta}{dt} &= -\nabla_\theta \mathcal{L} \\ &= -\nabla_\theta \mathbf{f}(\mathbf{f} - \bar{x}), \end{aligned} \tag{2.84}$$

where $\nabla_\theta \mathbf{f}$ is the $N_p \times P$ matrix of derivatives

$$\nabla_\theta \mathbf{f} = \begin{pmatrix} \frac{\partial f(\mathbf{r}^1)}{\partial \theta_1} & \cdots & \frac{\partial f(\mathbf{r}^P)}{\partial \theta_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial f(\mathbf{r}^1)}{\partial \theta_{N_p}} & \cdots & \frac{\partial f(\mathbf{r}^P)}{\partial \theta_{N_p}} \end{pmatrix}. \tag{2.85}$$

The vector output of the network then evolves according to the equation

$$\begin{aligned}\frac{d\mathbf{f}}{dt} &= \nabla_\theta \mathbf{f}^T \frac{d\theta}{dt} \\ &= -K_{NTK}(\mathbf{f} - \bar{x}),\end{aligned} \tag{2.86}$$

where

$$(K_{NTK})_{ij} = k_{NTK}(\mathbf{r}^i, \mathbf{r}^j) = \sum_{l=1}^{N_p} \frac{\partial f(\mathbf{r}^i)}{\partial \theta_l} \frac{\partial f(\mathbf{r}^j)}{\partial \theta_l} \tag{2.87}$$

is the Gram matrix of the so-called neural tangent kernel (NTK), k_{NTK}. In the lazy training regime described above, the neural tangent kernel remains constant during training; moreover, when the hidden layer size $M \to \infty$ and weights are initialized appropriately, the kernel converges to a (deterministic) function which is independent of the values of the parameters at initialization [96, 146, 97]. The gradient flow dynamic converges to the kernel regression solution [96], and the final decoding function converges to

$$f_{NTK}(\mathbf{r}) = \bar{x}^T (K_{NTK})^{-1} \mathbf{k}_{NTK}(\mathbf{r}), \tag{2.88}$$

where $(\mathbf{k}_{NTK}(\mathbf{r}))_i = k_{NTK}(\mathbf{r}^i, \mathbf{r})$.

For a two-layer neural network with rectifying linear activation function (ReLu) in the hidden layer, vanishing bias, and weights initialized as

$$(W_{(0)})_{ij} \sim \mathcal{N}(0, 1), \tag{2.89}$$

$$(W_{(1)})_i \sim \mathcal{N}\left(0, \frac{2}{M}\right), \tag{2.90}$$

the deterministic value of the neural tangent kernel in the infinite-M case can be calculated explicitly [97, 110], as

$$K_{NTK,relu}(\mathbf{r}, \mathbf{r}') = \|\mathbf{r}\| \|\mathbf{r}\| \left(2u\left(1 - \frac{\arccos(u)}{\pi}\right) + \frac{\sqrt{1-u^2}}{\pi}\right), \tag{2.91}$$

where $u = \mathbf{r}^T\mathbf{r}' / \|\mathbf{r}\| \|\mathbf{r}\|$ is the cosine of the angle between the two vectors, $\mathbf{r}$ and $\mathbf{r}'$.

We found that the NTK associated with a similar architecture, but with an Erf (error) non-linear function performs slightly better empirically. In this case, the NTK can be calculated [147] as

$$K_{NTK,Erf}(\mathbf{r}, \mathbf{r}') = \arcsin\left(\frac{\mathbf{r}^T\mathbf{r}'}{\sqrt{\left(1 + \|\mathbf{r}\|^2\right)\left(1 + \|\mathbf{r}'\|^2\right)}}\right). \tag{2.92}$$

2.4.7 Details of numerical simulations

Numerical simulations are carried out with custom codes written in Julia [148].

Bayesian decoder. We train a Bayesian decoder with $M = 500$ hidden-layer neurons with preferred stimuli, x_m, equally spaced in the stimulus space. For a given set of tuning

curves, $\{v_i\}$, we generate a dataset composed by $P = \gamma N_P$, where $N_p = MN$, samples from the joint distribution $p_v(\mathbf{r}, x)$. The results are obtained by averaging over 8 realizations of the set of tuning curves. The parameters of the network, $\{\lambda, \mathbf{b}\}$, are learned through stochastic gradient descent on the loss on mini-batches of size 128 with Adam algorithm [149], with learning rate equal to 10^{-3} and otherwise standard hyperparameters. The training is iterated over multiple passes over the data (epochs) with a maximum of 2000 epochs and stopped when the training loss running average stays constant (with a tolerance of 10^{-5}) for 10 consecutive epochs. The decoder is then tested by calculating the generalization error on a set of 10^6 samples from the joint distribution $p_v(\mathbf{r}, x)$.

Linear decoder. For a given set of tuning curves, we generate a dataset composed by $P = \gamma N$ samples from the joint distribution $p_v(\mathbf{r}, x)$. We compute the optimal regression coefficients as a function of the data matrix R, Eq. (2.42). We then test the decoder by calculating the generalization error on a set of 10^6 samples; in order to calculate the different terms in Eq. (2.53), we divide the test points into $10^6/P$ noisy responses to each stimulus used in the training set. The results are averaged over 8 realizations of the set of tuning curves. We calculate the analytical prediction by solving the Gram matrix eigenvalue problem and by solving numerically Eq. (2.15) to find the effective regularizer.

Lazy regime. We analyze the properties of the kernel regression solution obtained with the deterministic limit of the neural tangent kernel. The decoding function is obtained as in Eq. (2.47) with the NTK defined as in Eq. (2.92). Kernel methods are non-parametric methods, thus we kept constant the ratio between the number of samples and the dimensionality of the data, N. For a given set of tuning curves, we generate a dataset composed by $P = \gamma_n N$ samples from the joint distribution $p_v(\mathbf{r}, x)$. We compute the $P \times P$ kernel Gram matrix to obtain the decoding vector $\alpha = \bar{x}^T (K_{NTK})^{-1}$. We test the decoding function by computing $\mathbf{k}_{NTK}(\mathbf{r})$ on 10^6 samples and by calculating the generalization error. The results are averaged over 8 realizations of the set of tuning curves.

Rich regime. As pointed out in [97], the transition between lazy and rich regime is controlled by the variance of the weights at initialization. An intuition for this, is that, by rescaling the weights, we can violate the condition necessary for the lazy training to occur, namely that an appreciable change in the loss can be obtained by a negligible change in the parameters. We trained a neural network with a hidden layer of size M in the rich regime by initializing the biases to 0 and the weights as

$$(W_{(0)})_{ij} \sim \mathcal{N}(0, \frac{\alpha_s}{M}), \tag{2.93}$$

$$(W_{(1)})_i \sim \mathcal{N}\left(0, \frac{2}{M}\right), \tag{2.94}$$

where $\alpha_s = 10^{-3}$. For a given set of tuning curves, $\{v_i\}$, we generate a dataset composed by $P = \gamma N_P$, where $N_p = MN + 2M + 1$, samples from the joint distribution $p_v(\mathbf{r}, x)$. The results are obtained by averaging over 8 realizations of the set of tuning curves. The parameters of the network are learned through stochastic gradient descent on the MSE loss on mini-batches of size 128 with Adam algorithm [149], with learning rate equal to 10^{-4} and otherwise standard hyperparameters. The training is iterated over multiple passes over

the data (epochs) with a maximum of 2000 epochs and stopped when the training loss running average stays constant (with a tolerance of 10^{-7}) for 10 consecutive epochs. The decoder is then tested by calculating the generalization error on a set of 10^6 samples from the joint distribution $p_v(\mathbf{r}, x)$. The crossover value M_{sat} is the value of M at which the difference with the decoding performance of the widest network is smaller that the 10% of the difference in decoding performances between the widest and the smallest network, $\varepsilon^2(M_{sat}) - \varepsilon^2(M_{max}) < \frac{\varepsilon^2(M_{min}) - \varepsilon^2(M_{max})}{10}$.

S2.5 Supplementary Information

S2.5.1 Differences in the calculation of the generalization error

Here, we show that, in the asymptotic limit of large P, the expression for the generalization error from Ref. [103] becomes equivalent to the generalization error that we want to compute, Eqs. (2.26)-(2.27). The generalization error in Ref. [103], Eq. (2.95) is written explicitly as

$$\varepsilon_g^2 = \int dR p(R) \left[\int d\mathbf{r} dx p(x) p(\mathbf{r}|x;R) \left(f_l(\mathbf{r};R) - x \right)^2 \right], \tag{2.95}$$

where we made explicit the dependence of the conditional distribution of neural responses given the stimulus on the data matrix, R. First, we note that, by sending $P \to \infty$, we can consider the set of training stimuli, $\bar{x} = \{x_i\}_{i=1}^P$, as an infinite discretization of the stimulus space, and we can decompose the average over the data matrix as

$$p(R) = p(R|\{v_i\}) p(\{v_i\}). \tag{2.96}$$

Here, $\{v_i\} = \{v_i(x_1), ..., v_i(x_P)\}$ is the set of tuning curves, distributed according to the noise-free Gaussian process distribution, Eq. (2.20),

$$\begin{bmatrix} v_i(x_1) \\ \vdots \\ v_i(x_P) \end{bmatrix} \sim \mathcal{N} \left(0, \begin{bmatrix} \bar{k}(x_1,x_1) \dots \bar{k}(x_P,x_1) \\ \vdots \ddots \\ \bar{k}(x_1,x_P) \dots \bar{k}(x_P,x_P) \end{bmatrix} \right), \tag{2.97}$$

and, as the noise is independent across neurons and across trials, we have that

$$p(R|\{v_i\}) = \prod_{i=1}^P p_v(\mathbf{r}^i|x) = \prod_{i=1}^P \mathcal{N}\left(\mathbf{v}(x_i), \eta^2 \right), \tag{2.98}$$

where $\mathbf{v}(x) = (v_1(x), ..v_N(x))$ ad $\mathbf{r}^i$ are the columns of the matrix R. As we assumed P large, R becomes independent on the specific set of training stimuli, and the stochasticity in the dataset, $\mathcal{D}_v = \{R, \bar{x}\}$, depends only on the stochasticity in the neural responses, R, as $p(\mathcal{D}_v) \approx p(R|\{v_i\})$.

However, we $p(\mathbf{r}|x;R)$ depends on the noisy dataset R and not only on $\{v_i\}$, as in Eq. (2.26). We recall that the new activity patterns are distributed according to $p(\mathbf{r}|x) = \prod_{i=1}^N p(r_i|x) = \prod_{i=1}^N \mathcal{N}(\mu_n(x), \sigma_n^2(x))$, with $\mu_n(x)$ and $\sigma_n^2(x)$ as in Eqs. (2.51)-(2.50) . In Eq. (2.78) we showed that, in the asymptotic limit $P \to \infty$, $\sigma_n^2(x) \approx \eta^2$, independently on the specific realization of the dataset, i.e., the variance of a neural response is only dictated by the variance of the noise. As for the mean, by diagonalizing the kernel Gram matrix and writing $R = WUK^{1/2}U^T$, we obtain

$$\begin{aligned} \mu_n(x) &= R_{i,:} K^{-1} \mathbf{k}(x) \\ &\approx \frac{1}{\sqrt{PN}} W_{i,:} U D^{1/2} U^T (U D U^T)^{-1} U \bar{D} \vec{\phi}(x) \\ &= \frac{1}{\sqrt{PN}} W_{i,:} U \sum_{i=1}^P \frac{P\bar{d}_i^c \sqrt{P\bar{d}_i^c + \eta^2}}{P\bar{d}_i^c + \eta^2} \phi_i(x), \end{aligned} \tag{2.99}$$

where the approximation comes from Eq. (2.72) and $W_{i,:}$ denotes the i-th row of W. By considering the limit of large P, similarly to the calculation of the posterior variance, we have that $\frac{P\bar{d}_i^c\sqrt{P\bar{d}_i^c+\eta^2}}{P\bar{d}_i^c+\eta^2} \approx \sqrt{P\bar{d}_i^c}$, and we can write Eq. (2.99) in matrix form as

$$\mu_n(x) = \frac{1}{\sqrt{N}} W_{i,:} U \bar{D}^{1/2} \phi_i(x); \tag{2.100}$$

such expression does not depend on the variance of the noise, implying that the mean of the posterior distribution is only determined by the distribution of the noise-free tuning curves. As a result, the probability of a neural activity pattern, $\mathbf{r}$, becomes independent on the specific realization of the noise in the activity patterns in the dataset, and depends only on the tuning curves, $p(\mathbf{r}|x;R) \approx p_v(\mathbf{r}|x) = \prod_{i=1}^N \mathcal{N}(\mathbf{v}(x), \eta^2)$. An intuitive explanation for this fact is that, if we have enough data samples, we can average out the noise in the responses and obtain an estimate of the noise-free tuning curve, $\mathbf{v}(x)$, at any value of the stimulus.

Thus, we can rewrite Eq. (2.95) as

$$\begin{aligned} \varepsilon_g^2 &= \int \prod_{i=1}^N dv_i dR p(R|\{v_i\}) p(\{v_i\}) \left[\int d\mathbf{r} dx p(x) p(\mathbf{r}|x;R) \left(f_l(\mathbf{r};R) - x \right)^2 \right], \\ &\approx \int \prod_{i=1}^N dv_i p(\{v_i\}) \left[\int d\mathcal{D}_v p(\mathcal{D}_v) \left[\int d\mathbf{r} dx p(x) p_v(\mathbf{r}|x) \left(f_l(\mathbf{r};\mathcal{D}_v) - x \right)^2 \right] \right], \end{aligned} \tag{2.101}$$

where $\prod_{i=1}^N dv_i = \prod_{i=1}^N \prod_{j=1}^P dv_i(x_j)$, and the generalization error becomes equivalent to our expression, Eqs. (2.26)-(2.27).

S2.5.2 Supplementary figures

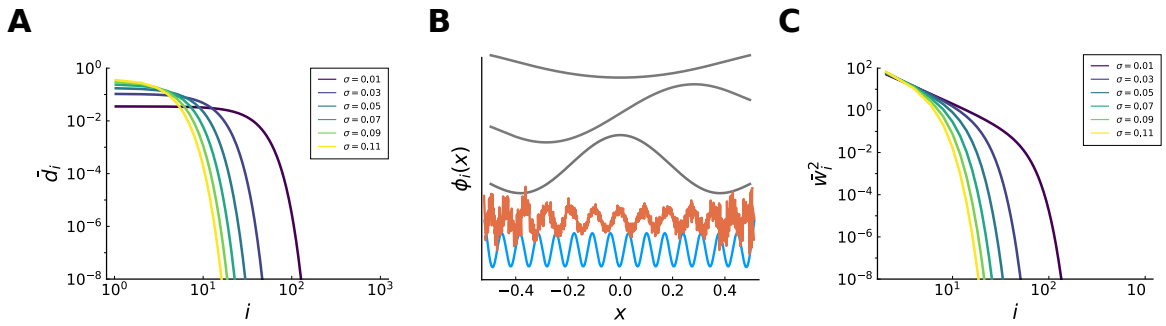

Fig. S2.1 ***Kernel spectral properties.*** *(**A**) Kernel eigenvalues, $\bar{d}_i^c$, as obtained by solving numerically Eq. (2.67), for different values of σ. (**B**) First three kernel eigenfunctions, $\phi_i(x)$ (grey, shifted to aid visualization) and high-index eigenfunction ($i = 30$) for a kernel with small (blue, $\sigma = 0.02$) and large tuning width (red, $\sigma = 0.9$). The low frequency eigenfunctions do not depend on the value of σ, and are similar to Fourier modes, as expected from a translational invariant kernel. However, at higher frequencies, the boundary effects become relevant and a difference as a function of σ emerge. (**C**) Optimal readout weights, $\bar{w}_i$, in the basis of kernel eigenfunctions, as obtained by calculating Eq. (2.73), for different values of σ. Since the target function is odd and kernel eigenfunctions are sine and cosine functions, the weights corresponding to an even index vanish; for simplicity, only non-vanishing weights are shown.*

Chapter 3

Efficient Coding and Decoding: a Variational Autoencoder Framework

3.1 Introduction

Normative models in neuroscience offer a theoretical framework to understand the optimality principles underlying information transmission and stimulus representation in the brain. Among these, the efficient coding hypothesis [8] posits the neural responses are set so as to maximize the information about external stimuli, under biological resource constraints. Despite this minimal assumption, this hypothesis has been successful in predicting neural responses to natural stimuli in various sensory areas [9, 150, 10]. The typical approach consists in specifying an *encoding* model, as a stochastic map between stimuli and neural responses. The parameters of this model are then chosen so as to optimize a function that quantifies the coding performance, e.g., the mutual information between stimuli and neural responses. This optimization is carried out under some metabolic cost proportional, e.g., to the energy needed to emit a spike [151]. The decoding process is ideally carried out in a Bayesian framework. Prior knowledge about the environment is combined with the sensory evidence, the likelihood of observing neural response given the encoding model, to form a posterior belief about the stimulus [21, 152].

The idea that the brain is capable of manipulating probabilities and uncertainty dates back to Helmoltz's view of perception as an inference process, in which the brain learns an internal statistical model of sensory inputs [153]. Mathematically, such an internal model can be formalized as a generative model which describes how external stimuli are generated by sampling from a conditional distribution given a set of 'latent,' elementary features [154, 6]. These features might be chosen so as to allow for a semantic interpretation, such as oriented edges or textures in generative models for natural images [155, 156], but this does not have to be the case in general. It is then assumed that the role of sensory areas is to perform statistical inference by computing the posterior distribution over the latent features which are most likely to have generated the sensory observation, thereby 'inverting' the internal model. Such posterior distribution is represented in the neural activity, and different representation mechanisms have been proposed [157, 158, 159]. As opposed to the efficient coding approach,

which prescribes a stochastic mapping from stimulus to neural activity, the generative model approach prescribes a stochastic mapping from neural activity to stimulus. This mapping implies a posterior distribution on neural activity, which can be read off from neural data.

Here, we consider an extended efficient coding approach: while, typically, only the sensory encoding process is optimized, we consider jointly the decoding process. In addition to a class of encoding transformations from stimuli to neural responses in a sensory area, we assume a class of generative models implemented in the downstream area. These define maps from neural activity patterns, corresponding to latent variables, to distributions over stimuli. Optimality is achieved when the generative distribution matches the true distribution of stimuli in the environment. If one assumes that the encoder and the decoder are jointly optimized in this framework, the system has the structure of a variational autoencoder (VAE) [160].

Similarly to the classical efficient coding framework, here the encoder is set so as to maximize a variational approximation to the mutual information between stimuli and neural responses under a constraint on the neural resources. However, an important aspect of this formulation is that the constraint, rather than being imposed by hand, is a direct consequence of the assumption of an optimal internal model. This constraint is obtained as the statistical distance between the stimulus-evoked distribution and the prior distribution over neural activity assumed by the generative model. The latter, in turn, can be interpreted as the statistics of spontaneous neural activity [161]; the statistical constraint can thus be viewed as the metabolic cost of stimulus-induced deviations from spontaneous neural activity.

We apply the theoretical framework to the study of a population coding model with neurons with bell-shaped tuning curves. By capitalizing on recent advances in the VAE literature, we solve the optimization problem as a function of the constraint on neural resources: we obtain a family of solutions which yield equally satisfying generative models [162]. However, these solutions make different predictions about the corresponding neural representations, which correspond to different arrangements of tuning curves, statistics of spontaneous neural activity, and coding performances. Related approaches have been explored in the literature, and predictions about the optimal allocation of coding resources, i.e., the tuning curves, as a function of the stimulus distribution have been derived [46, 21]. We further illustrate how, in our framework, the optimal allocation of coding resources as a function of the stimulus distribution varies as a function of the constraint. Despite the differences in the objective function, our results are consistent with previous predictions in a weakly constrained regime, while more complex behaviors are observed in a highly-constrained regime. Our results illustrate how the interactions between the encoder and the internal model shape neural representations of sensory stimuli.

3.2 Methods

In what follows, we denote vectors in bold font and scalars in regular font. We denote as $\langle f(z)\rangle_{p(z)}$ the expectation of a function f of a random variable z distributed according to $p(z)$, $\langle f(z)\rangle_{p(z)} = \int dz p(z) f(z)$.

Encoder (sensory representation). We consider a population of N neurons responding to a continuous scalar stimulus, x, distributed according to a prior distribution, $p(x)$

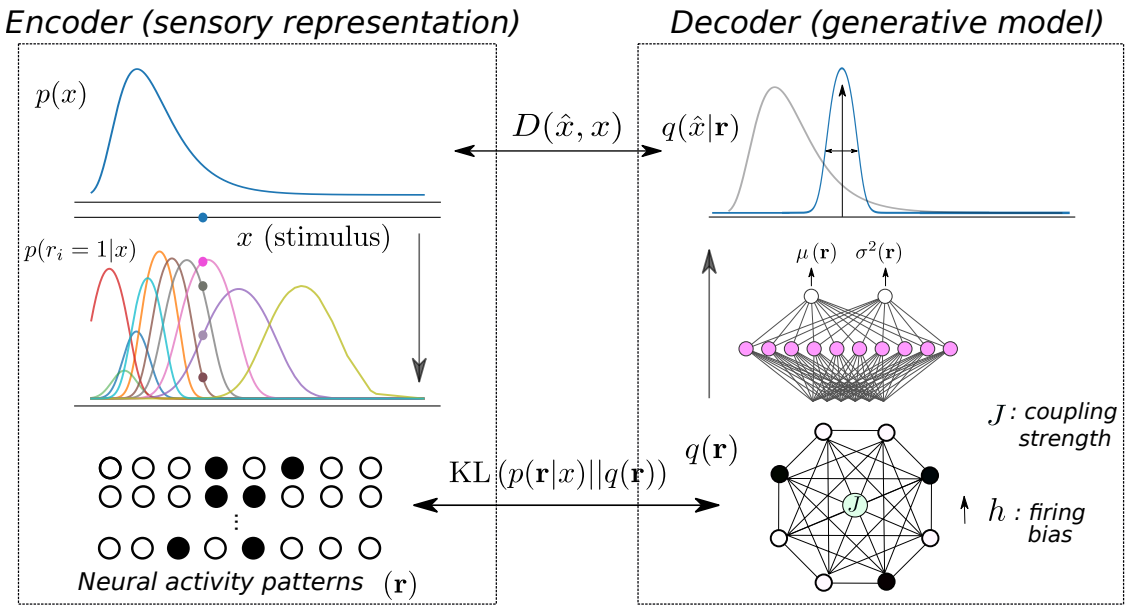

Fig. 3.1 ***Model architecture.*** *Left: encoder, or sensory representation. Neurons emits spikes according to bell-shaped tuning curves in response to a stimulus, x, drawn from the distribution $p(x)$, and the population response consists in a neural activity pattern, $\mathbf{r}$. Right: decoder, or generative model. The generative model maps neural activity patterns, sampled from the prior distribution (a Boltzmann machine), $q_\psi(\mathbf{r})$, to parameters, μ and σ, of a Gaussian distribution over stimuli, $q_\psi(x|\mathbf{r})$. When an activity pattern is observed, $q_\psi(x|\mathbf{r})$ is used to obtain an estimate of the stimulus which evoked it, as well as the associated uncertainty. The distortion term drives the system to maximize the likelihood of the observed stimulus given the generative distribution, while the rate term pushes the conditional encoding distribution to match the distribution of spontaneous activity of the network.*

(Fig. 3.1, left). In order to avoid confusion with the prior distribution over neural activity patterns, $q(\mathbf{r})$, defined below, we will refer to $p(x)$ as the data, or stimulus, distribution. We consider neural activity in the limit of short time intervals, such that each neuron either emits a spike or is silent. The set of possible activity patterns is then the set of binary vectors, $\mathbf{r} = (r_1, r_2, ..., r_N)$ where $r_i \in \{0, 1\}$; in what follows, the sum $\sum_{\mathbf{r}} \cdot$ denotes the sum over these 2^N binary patterns. The encoding distribution is the conditional probability distribution over neural activity patterns given the stimulus, $p_\theta(\mathbf{r}|x)$, where θ denotes the set of parameters. We assume neurons to spike independently, such that $p_\theta(\mathbf{r}|x) = \prod_{i=1}^N p_\theta(r_i|x)$.

We consider the limit of small time bins of the Poisson model for spiking neurons [163, 164], by taking into account only the first two terms of the Poisson distribution. With proper normalization, the probability of spiking of a neurons is obtained as

$$p_\theta(r_i = 1|x) = \frac{f_i(x)}{1 + f_i(x)}, \tag{3.1}$$

where $f_i(x)$ is the neuron's tuning curve. We parametrize tuning curves as Gaussian func-

tions, a shape widely observed in early sensory areas, as

$$f_i(x) = A_i \exp\left(-\frac{(x-c_i)^2}{2w_i^2}\right), \tag{3.2}$$

with c_i the preferred stimulus of neuron i, w_i the tuning width, and A_i the amplitude. Thus, the probability of spiking of a neuron can be written as $p_\theta(r_i = 1|x) = \mathcal{S}(\eta_i(x))$, with $\eta_i(x) = \frac{(x-c_i)^2}{2w_i^2} - \log A_i$ and $\mathcal{S}(y) = 1/(1+\exp(-y))$, the logistic function. In the canonical form of exponential families, the resulting multivariate Bernoulli distribution can be written as

$$p_\theta(\mathbf{r}|x) = \exp\left(\boldsymbol{\eta}(x)^T\mathbf{r} - \sum_{i=1}^{N}\log\left(1+e^{\eta_i(x)}\right)\right), \tag{3.3}$$

with $\boldsymbol{\eta}(x) = (\eta_1(x), ..., \eta_N(x))$ the vector of natural parameters and $\theta = \{A_i, c_i, w_i\}_{i=1}^N$ the set of parameters of the encoder.

Decoder (generative model). We define an internal model of the environment as a generative model, by specifying a parametric joint probability of neural activity patterns and sensory stimuli, $q_\psi(\mathbf{r}, x)$, where ψ denotes the set of parameters (Fig. 3.1, right). The neural activity patterns are treated as latent variables, sampled from a prior distribution, $q_\psi(\mathbf{r})$, and mapped to a distribution over stimuli, $q_\psi(x|\mathbf{r})$. As the prior distribution does not depend on the stimulus, we interpret $q_\psi(\mathbf{r})$ as describing the statistics of the spontaneous neural activity. We model this distribution through a maximum-entropy distribution constrained by the first- and second-order statistics of neural activity, a model which has been proposed as a model of the distribution of activity in neural systems, e.g., in retina and in cortex [165]. In the case of binary patterns, this maximum-entropy distribution takes the form of an Ising model, or Boltzmann machine,

$$q_\psi(\mathbf{r}) = \exp\left(\mathbf{h^T r} + \mathbf{r^T} J\mathbf{r} - \log Z\right), \tag{3.4}$$

where $\mathbf{h}$ is the vector of biases, J is the matrix of couplings (with our choice of parametrization, the diagonal elements of J vanish), and $Z = \sum_{\mathbf{r}} \exp(\mathbf{h^T r} + \mathbf{r^T} J\mathbf{r})$ is a normalization constant (also called partition function).

On the basis of experimental findings, it has been suggested that the brain encodes both a stimulus estimate and the associated uncertainty [128, 166]. Thus, we model the generative distribution as a Gaussian, whose mean (stimulus estimate) and variance (uncertainty) are generic functions of neural activity patterns,

$$q_\psi(x|\mathbf{r}) = \mathcal{N}\left(\mu_\phi(\mathbf{r}), \sigma_\phi(\mathbf{r})\right); \tag{3.5}$$

we parametrize these functions as two-layer neural networks, and we denote by ϕ the set of weights and biases. The parameters of the generative distribution and of the prior, $\psi = \{\phi, \mathbf{h}, J\}$, constitute the set of parameters of the generative model. In this framework, while the encoding distribution and the prior of the generative model are defined on the space neural activity patterns, the generative distribution is defined on the space of stimuli, which can be related to behavioral outputs (stimulus estimate). The neural network, thus, is not intended to be interpreted as a biological neural circuit, but just as a flexible

model of the map between neural activity and behavioral output.

Training objective. The internal model is optimal when the output probability distribution, $q_\psi(x) = \sum_{\mathbf{r}} q_\psi(x|\mathbf{r}) q_\psi(\mathbf{r})$, matches the true distribution of stimuli, $p(x)$. We achieve this by setting the parameters of the generative model so as to minimize the Kullback-Leibler (KL) divergence between the data and the generative distribution,

$$\min_{\psi} \left\{ \mathrm{KL}\left(p(x)||q_\psi(x)\right) = H(p) - \langle \log q_\psi(x) \rangle_{p(x)} \right\}, \tag{3.6}$$

where $H(p)$, the stimulus entropy, does not depend on the parameters. In order to learn the optimal parameters on the basis of a set of data points, we assume a two-stages encoding-decoding process. The encoder maps a stimulus sample, x, to a neural activity pattern, $\mathbf{r}$, according to $p_\theta(\mathbf{r}|x)$. The activity pattern corresponds to a configuration of latent variables in the generative model, and is mapped back ('decoded') to a distribution over stimuli according to $q_\psi(x|\mathbf{r})$. By including also the encoder, we can rewrite the second term on the right-hand-side of Eq. (3.6) as the sum of three terms,

$$\langle \log q_\psi(x) \rangle_{p(x)} = \left\langle \mathrm{KL}\left(p_\theta(\mathbf{r}|x)||q_\psi(\mathbf{r}|x)\right) + \sum_{\mathbf{r}} p_\theta(\mathbf{r}|x) \log q_\psi(x|\mathbf{r}) - \mathrm{KL}\left(p_\theta(\mathbf{r}|x)||q_\psi(\mathbf{r})\right) \right\rangle_{p(x)}. \tag{3.7}$$

The first term in the sum involves the posterior distribution over neural activity patterns, $q_\psi(\mathbf{r}|x) = q_\psi(x|\mathbf{r}) q_\psi(\mathbf{r}) / q_\psi(x)$; calculating $q_\psi(x)$ requires summing over all patterns of activity, $\mathbf{r}$, which is computationally prohibitive. Instead, we use the fact that the KL divergence is positive, and vanishes only when the two distributions are identical, to convert Eq. (3.7) into an inequality,

$$\langle \log q_\psi(x) \rangle_{p(x)} \geq \left\langle \sum_{\mathbf{r}} p_\theta(\mathbf{r}|x) \log q_\psi(x|\mathbf{r}) - \mathrm{KL}\left(p_\theta(\mathbf{r}|x)||q_\psi(\mathbf{r})\right) \right\rangle_{p(x)}. \tag{3.8}$$

Since the generative distribution, $\log q_\psi(x)$, is often referred to as the 'evidence' for a data point, x, the quantity on the right hand side of Eq. (3.8) goes by the name of 'evidence lower bound' (ELBO).

We can then address a variational approximation to the problem in Eq. (3.6) by maximizing the lower bound (ELBO). Equivalently, we optimize the encoder and decoder parameters so as to the minimize the negative ELBO, written as the sum of two terms,

$$\min_{\{\psi,\theta\}} \left\{ -\mathrm{ELBO} = D + R \right\}; \tag{3.9}$$

borrowing the nomenclature from rate-distortion theory, we define as *distortion* the quantity

$$D = \left\langle -\sum_{\mathbf{r}} p_\theta(\mathbf{r}|x) \log q_\psi(x|\mathbf{r}) \right\rangle_{p(x)}, \tag{3.10}$$

equal to to the opposite on the right-hand-side of Eq. (3.8), which measures the average log-probability of a stimulus, x, after the encoding-decoding process, and as *rate* the quantity

$$R = \left\langle \mathrm{KL}\left(p_\theta(\mathbf{r}|x)||q_\psi(\mathbf{r})\right) \right\rangle_{p(x)} = \left\langle \sum_{\mathbf{r}} p_\theta(\mathbf{r}|x) \log \left(\frac{p_\theta(\mathbf{r}|x)}{q_\psi(\mathbf{r})} \right) \right\rangle_{p(x)}, \tag{3.11}$$

equal to the opposite of the second term, which measures the statistical distance between the encoding distribution and the prior assumed by the generative model. This framework goes by the name of variational autoencoder (VAE) [160]. As one typically does not have access to the true data distribution, but only to a set of samples, the average over $p(x)$ is approximated by an empirical average over a set of P samples, $\langle f(x)\rangle_{p(x)} \approx \sum_{i=1}^{P} f(x_i)/P$.

We note that, due to the fact that the variance of the generative distribution depends on the neural responses, the distortion differs from the more usual mean squared error (MSE) loss function of classical autoencoders, also commonly employed to measure the performance of neural codes. Indeed, here the distortion function is written as

$$D = \left\langle \sum_{\mathbf{r}} p(\mathbf{r}|x) \left(\frac{\left(\mu_\phi\left(\mathbf{r}\right) - x\right)^2}{2\sigma_\phi^2(\mathbf{r})} + \frac{1}{2}\log\left(2\pi\sigma_\phi^2(\mathbf{r})\right)\right)\right\rangle_{p(x)}, \tag{3.12}$$

while the MSE is obtained as

$$\varepsilon^2 = \left\langle \sum_{\mathbf{r}} p(\mathbf{r}|x) \left(\mu_\phi\left(\mathbf{r}\right) - x\right)^2 \right\rangle_{p(x)}, \tag{3.13}$$

where we have used the fact that the optimal estimator is given by the mean of the posterior.

Constrained optimization and connection with efficient coding. It is a known issue in the VAE literature that, when the generative distribution is flexible enough as compared to the data distribution (meaning that $q_\psi(x|\mathbf{r})$ has enough degrees of freedom to approximate complex distributions), the ELBO optimization problem exhibits multiple solutions. Optimization algorithms based on stochastic gradient descent are biased towards solutions with low rate and high distortion, a phenomenon which goes by the name of posterior collapse [167, 162]. In the extreme case, the model relies entirely on the power of the decoder and ignores the latent variables altogether: all realizations of the latent variables are mapped to the data distribution, $q_\psi(x|\mathbf{r}) \approx p(x)$, and, consequently, all stimuli are mapped to the same representation, $p_\theta(\mathbf{r}|x) \approx q_\psi(\mathbf{r})$.

We overcome this issue by addressing a related constrained optimization problem. We minimize the distortion subject to a maximum value, or 'target', of the rate, $\bar{R}$:

$$\begin{aligned} \min_{\{\theta,\psi\}} \quad & D \\ \text{subject to} \quad & R \leq \bar{R}. \end{aligned} \tag{3.14}$$

The set of parameters $\{\theta, \psi\}$ satisfying the constraint $R \leq \bar{R}$ is called feasible set. By writing the associated Lagrangian function with multiplier $\beta \geq 0$, we have that

$$\max_{\beta\geq 0} \left\{ L(\theta,\psi,\beta) = D + \beta(R - \bar{R}) \right\} = \begin{cases} D & \text{if } \{\theta,\psi\} \text{ is feasible} \\ \infty & \text{otherwise.} \end{cases} \tag{3.15}$$

Solutions of Eq. (3.14) can thus be found as solutions to the 'minmax' problem, $\min_{\{\theta,\psi\}} \max_\beta L(\theta,\psi,\beta)$, but this problem is numerically intractable[1]. Instead, we solve the

[1]Numerically, the optimization can be carried out by applying stochastic gradient descent on the loss function. In order to respect the constraint, the gradient with respect to the parameters, $\{\theta, \psi\}$, should be projected on the feasible set. This implies finding the closest vector to the gradient which belongs to the feasible set, and update the parameters according to such 'projected gradient'. Typically, due to the high-dimensionality of the parameters space and the non-linearity of the constraint, the projected gradient is hard to calculate.

dual 'maxmin' problem, defined as

$$\max_{\beta \geq 0} \min_{\{\theta,\psi\}} D + \beta(R - \bar{R}). \tag{3.16}$$

The solution of the dual problem yields a lower bound for the original (primal) problem, as $\max_\beta \min_{\{\theta,\psi\}} L \leq \min_{\{\theta,\psi\}} \max_\beta L$. If D and R are convex and the solution satisfies certain conditions (Karush–Kuhn–Tucker conditions [168]) we have the so-called strong duality, and the solutions to dual and primal problems are identical. In the dual problem, the constraint $\beta \geq 0$ is handled straightforwardly, and we can optimize the loss function with alternate gradient descent/ascent (Arrow-Hurwicz algorithm). This framework was presented as an extension to the classical VAE, with the objective of obtaining disentangled latent representations, in Refs. [169, 162]. By interpreting β as a global modulatory signal [170, 171], we assume that $\{\theta, \psi\}$ and β evolve according to two different time scales, Alg.1. We denote the optimal parameters by $\{\theta^*, \psi^*, \beta^*\}$.

The two terms contributing to the ELBO are related to the mutual information between stimuli and neural responses,

$$I_p(\mathbf{r}, x) = \left\langle \log \frac{p_\theta(\mathbf{r}, x)}{p(x)p_\theta(\mathbf{r})} \right\rangle_{p_\theta(\mathbf{r},x)}, \tag{3.17}$$

through the bounds

$$H(p) - D \leq I_p(\mathbf{r}, x) \leq R, \tag{3.18}$$

where $H(p)$ is the entropy of the stimulus distribution[2][162]. The two inequalities arise, respectively, because in the variational approximation the posterior over stimuli, $q_\psi(x|\mathbf{r})$, replaces $p_\theta(x|\mathbf{r})$, and the prior over activity patterns, $q_\psi(\mathbf{r})$, replaces $p_\theta(\mathbf{r})$. Thus, constraining the maximum value of the rate is equivalent to constraining an upper bound to the mutual information between stimuli and neural responses.

Equation (3.18) has two important consequences. First, it allows us to interpret the problem in Eq. (3.14) as an efficient coding problem, where the objective is to maximize a lower bound to the mutual information, $H - D$, subject to a bound on the neural resources, $\bar{R}$. Contrary to the classical efficient coding literature, in which a metabolic constraint is imposed by hand, here it results from the original formulation of the problem as optimization of the ELBO, and it is affected by the assumptions made on the generative model (more specifically, on the prior distribution).

Second, we note that the Lagrangian has a form similar to the negative ELBO (up to an additive constant which does not depend on the parameters), with an additional β factor multiplying the rate. For all $\beta^* \neq 0$, the constraint on the rate is satisfied as an equality, $R|_{\theta^*,\psi^*} = \bar{R}$. If the variational distributions, $q_\psi(\mathbf{r})$ and $q_\psi(x|\mathbf{r})$, are flexible enough to approximate $p_\theta(\mathbf{r})$ and $p_\theta(x|\mathbf{r})$, the optimal solution achieves both equalities and we have $D|_{\theta^*,\psi^*} = H(p) - R|_{\theta^*,\psi^*}$. Since we have $\frac{dD}{dR}\Big|_{\theta^*,\psi^*} = -\beta^*$, the optimization problem in Eq. (3.16) yields $\beta^* = 1$ and the loss function coincides with the negative ELBO.

Numerical optimization and related computations. Numerical simulations are carried out using PyTorch. We solve the optimization problem of Eq. (3.16) through stochastic

[2]We note that, since we are considering continuous stimuli, H is a differential entropy.

Algorithm 1 Two time-scales optimization algorithm.

Inputs: target rate $\bar{R}$, dataset $\mathcal{D}$
Initialize: $\beta = 1$, encoder/decoder parameters= $\{\theta_i, \psi_i\}$
while convergence **do**
    Define β-ELBO: $L_\beta = D + \beta R$
    for `batch in` $\mathcal{D}$ **do**
        Update parameters: $(\theta, \psi) \leftarrow \text{Adam}\left(\nabla_\theta L_\beta(\texttt{batch}), \nabla_\psi L_\beta(\texttt{batch})\right)$
    end for
    $\beta \to \max\{\beta + \eta_\beta(R - \bar{R}), 0\}$
end while
return

gradient descent on the loss on a dataset of $P = 5000$ samples from $p(x)$, divided in mini-batches of size 128, with the Adam optimizer [149] with learning rate equal to 10^{-4} and otherwise standard hyperparameters. The learning rate for β, η_β, is set to 0.1. The training is iterated over multiple passes over the data (epochs) with a maximum of 5000 epochs and it is stopped when the training loss running average stays constant (with a tolerance of 10^{-5}) for 100 consecutive epochs. The parameters are initialized as follows. The preferred positions, c_i, are initialized as the centroids obtained by applying a k-means clustering algorithm (with $k = N$) to the set of stimuli in the dataset. Tuning widths are initialized by setting $w_i = |c_i - c_j|$, with c_j being the closest preferred position to c_i, and the amplitude is set equal to 1, corresponding to a maximum probability of spiking of 0.5. Random noise of small variance is then applied to the initial value of the parameters. We illustrate results obtained by averaging over different random initializations. An example of the evolution of D, R, and β during training is illustrated in Fig. S3.1.

Here, we illustrate results in the case of N small enough so that it be possible to compute explicitly the sums over activity patterns appearing in the loss function. This also allows us to explore regimes in which the information is compressed in the activity of a small number of neurons. To extend the model to larger populations, there are two numerical issues to consider. The first one concerns the distortion term and the gradient with respect to the parameters of the encoder. In order to obtain a low-variance estimate of the gradient, a solution is to use the so-called reparametrization trick together with a continuous relaxation of the discrete random variable (or Gumbel-softmax trick [172, 173]), and calculate the gradient as

$$\begin{aligned} \nabla_\theta D(x) &= \nabla_\theta \left\langle \log q_\psi(x|\mathbf{r}) \right\rangle_{p_\theta(\mathbf{r}|x)} \\ &\approx \left\langle \nabla_\theta \log q_\psi(x|f_\theta(\xi, x)) \right\rangle_{p(\xi)} , \end{aligned} \tag{3.19}$$

with $p(\xi) = \mathcal{U}(0, 1)$ and

$$f_\theta(\xi, x)) = \mathcal{S}\left(\frac{\boldsymbol{\eta}_\theta(x) + \mathcal{S}^{-1}(\xi)}{\tau} \right) \tag{3.20}$$

depends deterministically on the parameters θ through the natural parameters of the encoder; the hyperparameter τ controls the steepness of the logistic function, and consequently the bias-variance trade-off for the gradient; simulations with values of $\tau = 10^{-2}$ yield results

comparable to the ones presented here. As for the rate, the expression can be simplified as

$$\begin{aligned}\mathrm{KL}\left(p_\theta(\mathbf{r}|x)||q_\psi(\mathbf{r})\right) &= \left\langle (\boldsymbol{\eta}(x)-\mathbf{h})\,\mathbf{r}-\mathbf{r}^T J\mathbf{r}\right\rangle_{p_\theta(\mathbf{r}|x)} - \sum_{i=1}^{N}\log\left(1+e^{\eta_i(x)}\right)+\log Z \\ &= (\boldsymbol{\eta}(x)-\mathbf{h})\,\mathbf{p}(x) - \mathbf{p}^T(x)J\mathbf{p}(x) - \sum_{i=1}^{N}\log\left(1+e^{\eta_i(x)}\right)+\log Z,\end{aligned} \tag{3.21}$$

where $\mathbf{p}(x)=\mathcal{S}\left(\boldsymbol{\eta}(x)\right)$ is the vector of mean parameters of the encoding distribution (i.e., the probability of spiking of neurons). In the expectation of the quadratic form, $\langle \mathbf{r}^T J\mathbf{r}\rangle_{p_\theta(\mathbf{r}|x)} = \mathrm{tr}(K_{\mathbf{rr}}J)+\mathbf{p}^T(x)J\mathbf{p}(x)$, we have that $\mathrm{tr}(K_{\mathbf{rr}}J)=0$, as the covariance matrix of the activity patterns, $K_{\mathbf{rr}}$, is proportional to the identity, and the diagonal elements of J vanish. Here, the bottleneck is in computing the gradient of the log-partition function, $\log Z$, which can be done by Monte Carlo methods [174].

3.3 Results

We optimize jointly an encoder, a population of neurons with simple tuning curves which stochastically maps stimuli to neural activity patterns, and a decoder, a neural network which maps activity patterns, interpreted as latent variables, to distributions over stimuli. The system is set so as to minimize a bound to the Kullback-Leibler (KL) divergence between the generative distribution and the true distribution of stimuli (Fig. 3.1). By formulating the training objective as a constrained optimization problem, we characterize the space of optimal solutions as a function of the value of the constraint; we discuss the properties of the encoder and of the decoder in the family of solutions.

3.3.1 Nature of the optimal representation

We begin by illustrating two alternative solutions of the ELBO optimization problem, Eq. (3.9), characterized by different contributions of the two terms, D and R. We first consider the simple, but instructive, case of a Gaussian distribution over stimuli, $p(x)=\mathcal{N}\left(\mu_p,\sigma_p^2\right)$. In order to minimize the rate, a possible solution is to set the parameters of the encoder so as to map all stimuli to the same distribution over neural activity patterns, which takes a similar form as the prior distribution, $p_\theta(\mathbf{r}|x)\approx q_\psi(\mathbf{r})$. This is achieved through neurons with low selectivity, i.e., with broad and overlapping tuning curves (Fig. 3.2A, top). Despite the non-informative neural representation, a perfect generative model is obtained (in this special Gaussian case) by mapping all activity patterns to the parameters of the data distribution, $\mu_\psi(\mathbf{r})=\mu_p$ and $\sigma_\psi^2(\mathbf{r})=\sigma_p^2$ for all $\mathbf{r}$; this way, the generative distribution becomes independent from the neural activity, $q_\psi(x|\mathbf{r})\approx p(x)$ (Fig. 3.2A, bottom). The rate term is then negligible and the distortion achieves its minimum possible value (given the inequality in Eq. (3.18)) which corresponds to the stimulus entropy. The sum of the two terms therefore yields the optimal value of the ELBO; the neural representation, however, retains no information about the stimulus.

At the opposite extreme, it is possible to minimize the distortion by learning an injective encoding map which associates distinct stimuli to distinct activity patterns. The decoder can then map each activity pattern to a narrow Gaussian distribution over stimuli. In our

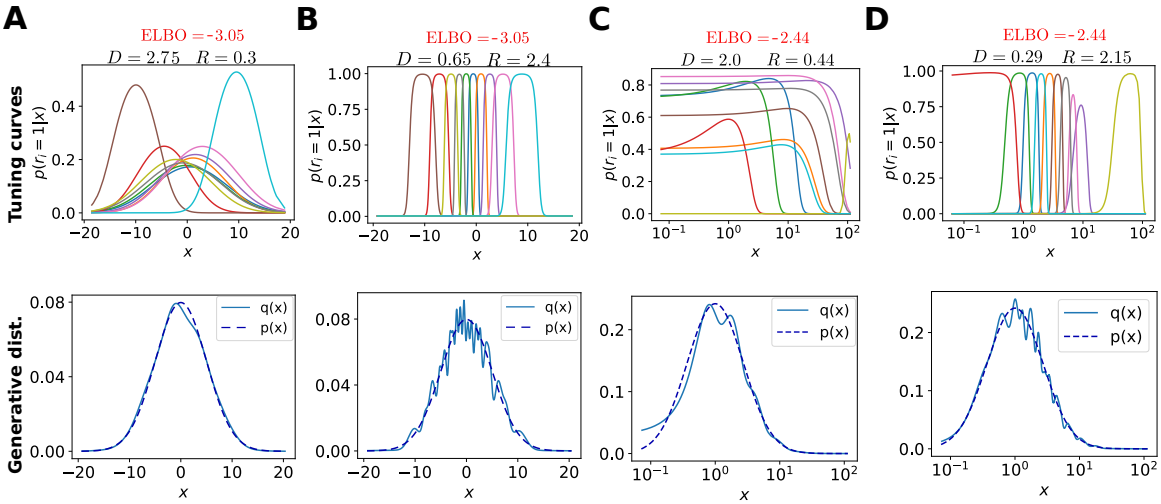

Fig. 3.2 ***Qualitatively different optimal configurations.*** *In all simulations,* $N = 10$. *Top row: probability of spiking of neurons. Bottom row: generative distributions,* $q(x) = \sum_{\mathbf{r}} q(x|\mathbf{r})q(\mathbf{r})$ *(solid curve),, compared to the stimulus distributions (dashed curve). (**A**) High-distortion, low-rate solution in the case of a Gaussian distribution of stimuli,* $p(x) = \mathcal{N}(0, 5)$. *(**B**) Low-distortion, high-rate solution for the same Gaussian distribution over stimuli. (**C**),(**D**) Same as panels (**A**),(**B**), for a log-normal distribution of stimuli,* $p(x) = \text{Lognormal}(1, 1)$.

framework, this is achieved through narrow and non-overlapping tuning curves, tiling the stimulus space such that stimuli in a small interval activate a single neuron (Fig. 3.2B, top). For a given encoding distribution, the optimal prior distribution which minimizes the rate, Eq. (3.11), is equivalent to the marginal encoding distribution [175],

$$q_{\psi^*}(\mathbf{r}) = \langle p_\theta(\mathbf{r}|x) \rangle_{p(x)} \,. \tag{3.22}$$

If the encoding distribution is different for each stimulus, the rate term does not vanish, but the parameters of the prior can still be set so as to approximate Eq. (3.22), achieving the rightward equality in Eq. (3.18). As a consequence, the ELBO is optimized as well, and it is possible to obtain a generative model that approximates closely the stimulus distribution, though less smoothly (Fig. 3.2B, bottom). Indeed, the marginal distribution, $q_\psi(x) = \sum_{\mathbf{r}} q_\psi(x|\mathbf{r})q_\psi(\mathbf{r})$, is a Gaussian mixture, which is a universal approximator of densities (i.e., a well-chosen Gaussian mixture can be used to approximate any smooth density function [176, 177]).

Thus, although these two solutions yield comparable values of the ELBO and equally satisfying generative models, the corresponding neural representations are utterly different. This case is special and contrived, as the conditional generative distribution has the same functional form as the stimulus distribution, and thus a perfect generative model is obtained even when it ignores the latent variables. However, the reasoning extends to more complex cases, and the choice of the forms of the decoding distribution and the prior determines the ability of the system to optimize the ELBO in different ways [162]. In order to achieve an optimal distortion at low rates, the generative distribution must be complex enough to

approximate the data distribution even when the latent variables carry no information about the stimulus. Conversely, prior distributions which can fit marginal encoding distributions in which each data point is mapped precisely to a realization of the latent variables, achieve the optimal values of the rate at low values of the distortion (Eq. (3.22)). Indeed, we observe the existence of multiple solutions to the ELBO optimization problem also for more complex stimulus distributions (Fig. 3.2C,D, Fig. S3.2).

3.3.2 Analysis of the family of optimal solutions

We explore systematically the space of solutions which optimize the ELBO by minimizing the distortion subject to a constraint on the maximum ('target') value of the rate, $\bar{R}$, a formulation which yields a generalized objective function (Eq. (3.16)) with a factor of β that weighs the rate term (see Methods). The value of $\bar{R}$ is an upper bound to the mutual information between stimulus and neural response; it thereby imposes a degree of 'compression' of the information in the encoding process. We illustrate results for the simple, yet non-trivial, choice of a log-normal stimulus distribution, but similar observations are valid for other distributions as well (in Fig. S3.3 we illustrate the case of a more complex, multimodal distribution).

Each solution is associated with a point $(\bar{R}, D)$ in the rate-distortion plane. By varying the value of $\bar{R}$, we trace the curve of the optimal distortion as a function of the target rate (Fig. 3.3A). We focus on the range of values of $\bar{R}$ resulting in $\beta^* = 1$, in which $R = \bar{R}$ and the corresponding solutions also yield an optimal value of the ELBO. These solutions belong to the line $D = H(p) - R$, with $H(p)$ the stimulus entropy, and both inequalities in Eq. (3.18) are achieved. (As the stimulus and the generative distribution do not belong to the same parametric family, it is not possible to achieve an optimal distortion with $R = 0$.) As a result, the mutual information is equal to $\bar{R}$ (Fig. 3.3A, inset). Eventually, for sufficiently large $\bar{R}$, the distortion stops decreasing and saturates; this occurs when the tuning curves are as narrow as possible while still tiling the stimulus space (Fig. 3.2B,D). The distortion can be further decreased by increasing the number of available activity patterns, which depends on the population size (Fig. S3.2).

Different values of $\bar{R}$ result in different arrangements of the tuning curves (Fig. 3.3B). For small values of $\bar{R}$, tuning curves are broad and the spacing between preferred positions is small, causing large overlaps; different stimuli are mapped to similar distributions over neural activity patterns. Moreover, they are characterized by low amplitudes and, thus, higher stochasticity; indeed stochastic neurons yield compressed representations [178]. Increasing $\bar{R}$ causes noise to be suppressed through an increase in the amplitude, and narrower and more distributed tuning curves.

We also illustrate coding properties in terms of a common quantity used in perceptual experiments and theoretical analyses: the mean squared error (MSE) in the stimulus estimate, as obtained from the mean of the approximate decoding distribution, $q_\psi(x|\mathbf{r})$. As expected from the higher mutual information between stimuli and neural responses, the coding performance of the encoder-decoder system increases as a function of $\bar{R}$ (Fig. 3.3C). Although this qualitative statement is obvious, it is worth examinining quantitatively. The MSE has a slightly different functional form, which does not depend on the variance of the decoding distribution (see Methods, Eq. (3.13)); it does not decrease linearly with $\bar{R}$, but rather it exhibits a rapid decrease followed by a slower one.

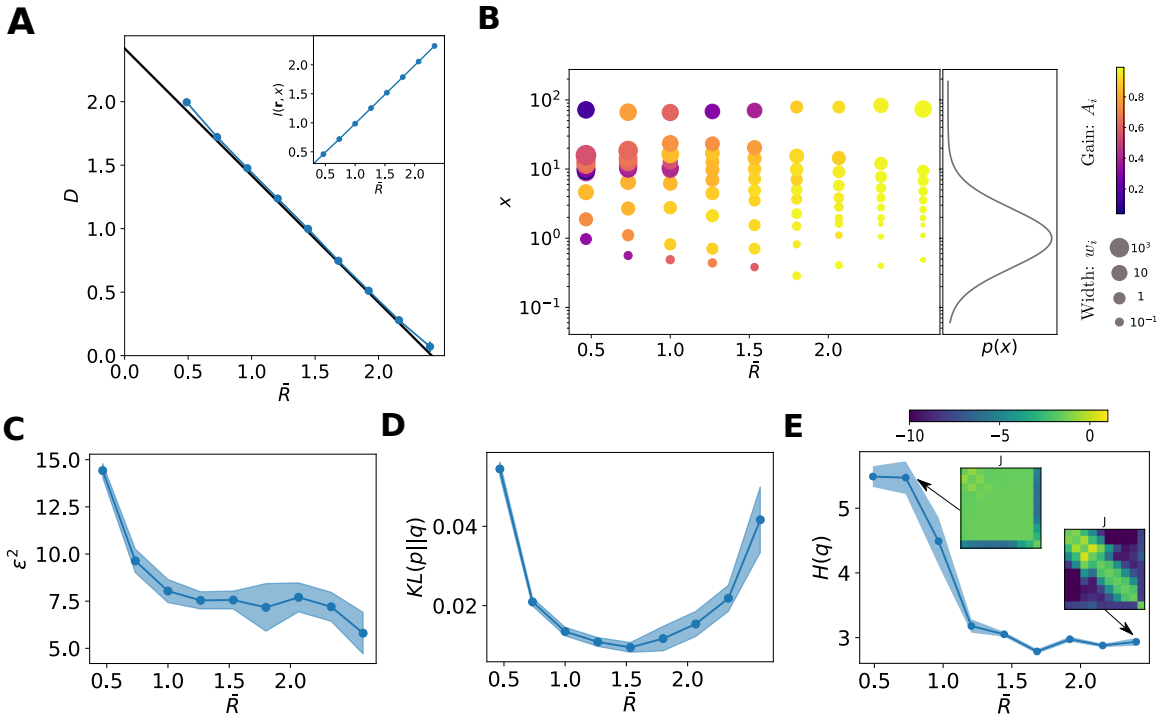

Fig. 3.3 ***Characterization of the optimal solutions as a function of the target rate.*** *In all simulations,* $N = 10$, $p(x) = \text{Lognormal}(1, 1)$ *and results are averaged over 8 initializations of the parameters. (**A**) Solutions to the* ELBO *optimization problem as a function of target rate,* $D(\bar{R})$ *(blue curve), and theoretical optimum,* $D = H(p) - R$ *(black curve), in the rate-distortion plane. For all the solutions, we have* $R = \bar{R}$*. Solutions depart from the optimal line when the rate is very low (poor generative model) or very high (saturated distortion). Inset: mutual information between stimuli and neural responses as a function of* $\bar{R}$*. (**B**) Optimal tuning curves for different values of* $\bar{R}$*. Each dot represents a neuron: the position on the y-axis corresponds to its preferred stimulus, the size of the dot is proportional to the tuning width, and the color refers to the amplitude (see legend). The curve illustrates the data distribution,* $p(x)$*. (**C**) MSE of the stimulus estimate as a function of* $\bar{R}$*. (**D**)* KL *divergence between the stimulus and the generative distribution, as a function of* $\bar{R}$*. (**E**) Entropy of the prior distribution over neural activity as a function of* $\bar{R}$*. Insets show two configurations of the coupling matrices, with rows ordered according to the neurons' preferred stimuli, and coupling strengths colored according to the legend.*

We now turn to the effect of the constraint on the generative model. As expected from the value of the ELBO, the difference between the generative,$q_\psi(x)$, and the stimulus distribution, $p(x)$, measured by the KL divergence, is negligible for all values of $\bar{R}$ (Fig. 3.3D). (We recall that the ELBO, up to a constant, is a lower bound to this quantity, and the gap is quantified by the KL divergence between the true and the approximate posterior distribution, Eq. (3.7)). The U-shape is due to the jaggedness of the generative model at high values of $\bar{R}$. This suggests that intermediate representations, yielding a smooth approximation of the stimulus distribution, yet achieving a low coding error, are preferred to representations with extremely narrow tuning curves (Fig. 3.2B,D).

The solutions also differ in the amount of information about the stimulus embedded in the structure of the prior over neural activity, $q_\psi(\mathbf{r})$ (Fig. 3.3E, insets). In the regime in which the decoder ignores the latent variables, i.e., $q_\psi(x|\mathbf{r}) \approx q_\psi(x)$, the prior, $q_\psi(\mathbf{r})$ is left unstructured and the couplings, J, are weak. By contrast, when $\bar{R}$ is large, the structure of the stimulus distribution affects the coupling matrix in the prior, inducing coupling strengths that depend on the distances between the neurons' preferred positions. As the coupling strengths increase, the entropy of the prior distribution decreases (Fig. 3.3E). In more complex distributions where, even when the rate is low, a structure to the prior is imposed through the biases, $\mathbf{h}$, in order to obtain a satisfying generative model, the entropy can exhibit a non monotonic behavior (Fig. S3.3E).

3.3.3 Optimal allocation of neural resources and coding performance

The classical efficient coding hypothesis prescribes an allocation of neural resources as a function of the stimulus distribution: more frequently stimuli are represented with higher precision, which, in turn, can explain perceptual accuracy and biases [62, 21, 179]. We investigate, in our model, the relations between stimulus distribution, the use of neural resources (tuning curves), and coding performance, and how each vary with $\bar{R}$. We emphasize that the functional form of the stimulus distribution affects these relations, through its interplay with the functional form not only of the encoder (as in the classical efficient coding framework), but also of the generative distribution. We illustrate this difference with results on a stimulus distribution which belongs to the same parametric family as the generative distribution (Gaussian), and one which has a different form (log-normal). In order to make statements about the typical behavior of the system, we average our results over different random initializations of the parameters; single solutions might deviate from the average behavior due to the small number of neurons and the high dimensionality of the parameters space. Our conclusions can be compared with results from previous studies. In particular, we make use of the analytical results derived in Ref. [21] for a similar population coding model; in Sec. S3.5, we provide an alternative derivation of these results and we comment on the main differences with our model. Here, we note that our results are obtained by considering regime of strong compression of the information (small population sizes), while previous studies focused on the asymptotic regime $N \to \infty$.

As illustrated in Fig. 3.3B, the target rate affects the neural density, i.e., the number of neurons with preferred stimuli within a given stimulus window. In previous work, maximizing the mutual information required that the neural density be proportional to the stimulus density, $d(x) \propto p(x)$ [46, 21]. In our case, the range of possible behaviors is richer, especially when the stimulus distribution is non-trivial (i.e., it does not have the same functional form

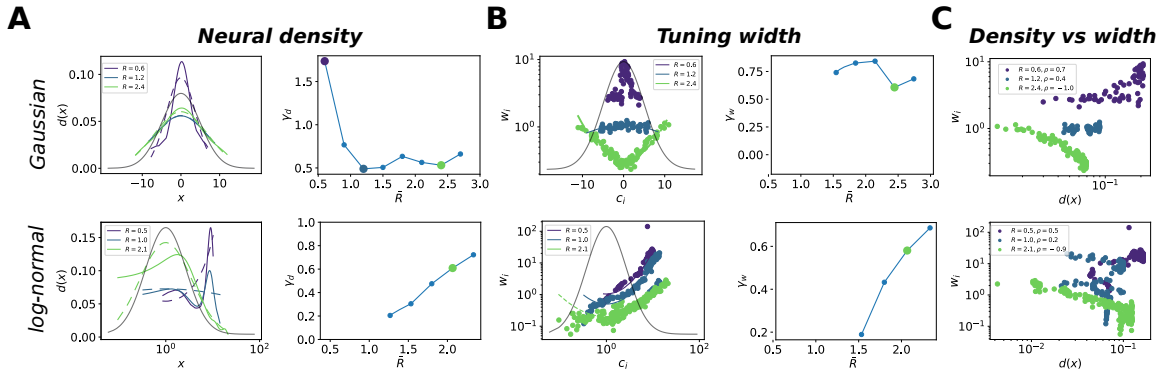

Fig. 3.4 ***Optimal allocation of neural resources.*** *In all simulations,* $N = 12$ *and results are averaged over 16 initializations of the parameters. Top row: Gaussian distribution over stimuli,* $p(x) = \mathcal{N}(0, 5)$*. Bottom row: log-normal distribution over stimuli,* $p(x) = \text{Lognormal}(1, 1)$*. Results are illustrated for regions of the stimulus space where the coding performance is sufficiently high, defined as the regions where the MSE is lower than the variance of the stimulus distribution. (**A**) Left: neural density as a function* x *(solid curves) and power-law fits (dashed curves), for three values of* $\bar{R}$ *(low, intermediate, and high); the grey curve illustrates the stimulus distribution. The density is computed by applying kernel density estimation to the set of the preferred positions of the neurons. Right: optimal exponent,* γ_d*, as a function of* $\bar{R}$*; large dots correspond to the examples shown in the left panels and are colored accordingly. Only the points for which the fit accounts for more than* 70% *of variance in data are shown. (**B**) Left: tuning width,* w_i *as a function of preferred stimuli,* c_i*, and power-law fits (dashed curves) for three values of* $\bar{R}$*; the grey curve illustrates the stimulus distribution. Right: optimal exponent,* γ_d*, as a function of* $\bar{R}$*; large dots correspond to the examples shown in the left panels and are colored accordingly. Only the points for which the fit accounts for more than* 70% *of variance in data are shown. (**C**) Tuning width,* w_i*, as a function of the neural density,* $d(x)$*, for three values of* $\bar{R}$*;* ρ *indicates the corresponding Pearson correlation coefficient.*

as that of the generative distribution). At low rates, the location of maximum density might be different from the mode of the stimulus distribution, depending on the interplay between the generative and the stimulus distributions (Fig. 3.4A, left). The neural density becomes more similar to the stimulus distribution for large values of $\bar{R}$: a power law functional form, $d(x) = A_d p(x)^{\gamma_d}$, yields a good agreement with our numerical results, with an optimal exponent, γ_d, close to $1/2$ (Fig. 3.4A, right).

In Ref. [21, 180], analytical results were obtained by constraining the neural density and the tuning width relative to each other. This is equivalent to fixing the overlap between tuning curves, by imposing $w(x) \propto d^{-1}(x) \propto p(x)^{-1}$ (see Sec. S3.5). In our case, the tuning width and neural density vary independently of each other, and the distribution of widths exhibits an intricate behavior at small values of $\bar{R}$ (Fig. 3.3B, left). However, at large values of $\bar{R}$, the tuning width decreases for large values of the stimulus distribution, and its behavior is well described by a power law, $w_i = A_w / p(c_i)^{\gamma_w}$ (Fig. 3.3B, right). As a result, we also

obtain an inverse relation between the neural density and the tuning width (Fig. 3.3C).

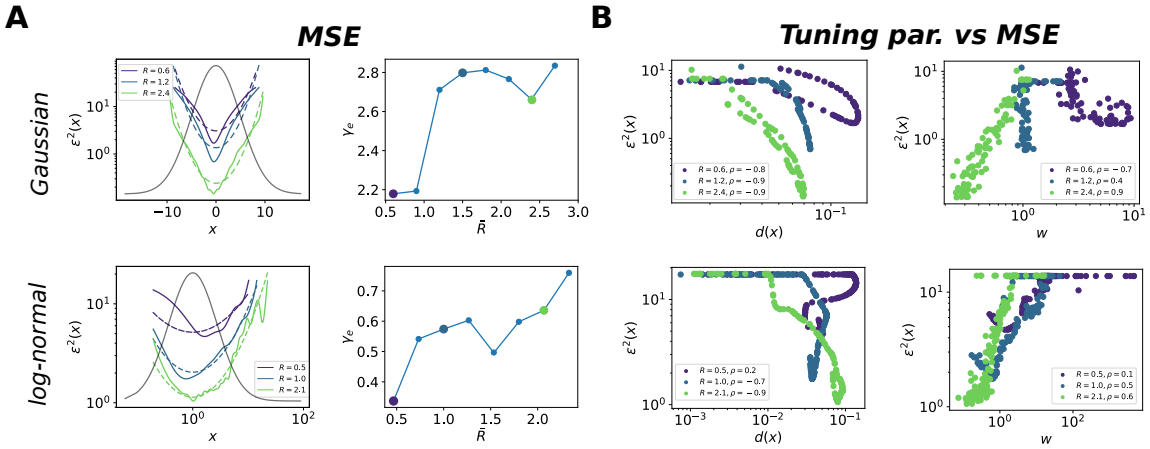

Fig. 3.5 ***Optimal allocation of coding performance.*** *Same numerical simulations as in Fig. 3.4. Top row: Gaussian distribution over stimuli,* $p(x) = \mathcal{N}(0, 5)$. *Bottom row: log-normal distribution over stimuli,* $p(x) = \text{Lognormal}(1, 1)$. ***(A)*** *Left: MSE as a function of* x *(solid curves) and power-law fits (dashed curves), for three values of* $\bar{R}$. *Right: optimal exponent,* γ_e, *as a function of* $\bar{R}$; *the large dots correspond to the examples shown in the left panels and are colored accordingly. The fits account for more than* 90% *of variance in data.* ***(B)*** *MSE as a function of the neural density (left) and tuning width (right), for three values of* $\bar{R}$; ρ *indicates the corresponding Pearson correlation coefficient.*

A consequence of the heterogeneous allocation of neural resources is a non-uniform coding performance across stimuli. Figure 3.5A shows that the MSE exhibits an inverse relation as a function of the stimulus distribution, with more frequent stimuli encoded more precisely. This is broadly consistent with previous studies [21, 89], which maximized the mutual information to obtain the expression

$$\varepsilon^2(x) \propto \frac{1}{p^2(x)}. \tag{3.23}$$

More precisely, this expression was derived using the Fisher information, whose inverse is a lower bound to the variance of any unbiased estimator and which can be related to the mutual information in some limits. Here, for all values of $\bar{R}$, the error is well described by a power law, $\varepsilon^2(x) = A_e/p(x)^{\gamma_e}$, with an exponent which tends to increase with the rate, but whose precise behavior and numerical value are affected by the form of the stimulus distribution (Fig. 3.5 A, right). Finally, we illustrate how the configuration of the tuning curves affects the coding performance, by plotting the MSE as a function of the neural density and tuning width. We observe a positive correlation between high coding performance and regions of high neural density as well as with narrow tuning widths, for large values of $\bar{R}$ (3.5B).

To summarize, given our choice of the loss function, which constrains the encoding stage as a function of the decoding stage, we obtain a range of possible optimal neural representations. In weakly constrained systems (large values of $\bar{R}$), we qualitatively recover previously derived relationships between tuning curves, stimulus distribution, and coding performance. (The difference in the numerical values of the exponents of the power laws can be explained

by the differences between the two models, see Sec. S3.5. We note that, in Ref. [21] the numerical value of the exponents also change as a function of the form of the loss function.) In systems with severe information compression (small values of $\bar{R}$), the optimal resource allocation exhibits a more intricate behavior, that depends on the interaction between the stimulus distribution and the properties of the generative model.

3.4 Discussion

Summary. We studied neural representations that emerge in a framework in which neural populations encode information about a continuous stimulus with simple tuning curves, but with the additional assumption that the task of the decoder is to maintain a generative model of the stimulus distribution. The consequence of this specific task imposed on the *decoder* is that the *encoder* is set so as to maximize a bound to the mutual information between stimulus and neural activity, as postulated by the efficient coding hypothesis, subject to a constraint on the relative entropy between evoked and spontaneous activity. As a function of this constraint, different optimal solutions are obtained, corresponding to equally efficient generative models but qualitatively different neural representations of the stimulus (Fig. 3.3). These representations differ in the degree of compression of information in the neural responses, reflected in encoding (neural) properties (Figs. 3.3 and 3.4), in the generative model prior (the statistics of spontaneous neural activity, Fig. 3.3E) and in the coding performance (Figs. 3.3 and 3.5).

Internal models and perception as inference. Our choice on the form of the decoder stems from the assumption, motivated by behavioral evidence, that organisms interact with the environment by constructing internal models. Internal models allow the individual to make predictions and perform inference, but the neural basis underlying them is debated. In previous studies [158, 159, 161, 181], internal models were defined by conditioning the probability of stimuli, x, on configurations of latent variables, z, through their joint distribution, $q(z, x) = q(x|z)q(z)$. In particular, no assumptions are made on how the latent variables are related to a specific neural representation. Then, the task that sensory areas were assumed to implement was the computation of the posterior distribution over the latent variables, $q(z|x)$. Only later the neural activity is invoked as a way to represent this posterior distribution, either approximately through samples [159, 181], or through the use of a specified parametric form and the assumption that the values of the parameters are encoded in neural activity [157, 182].

Instead, we define the generative model directly as a joint distribution of two random variables, $q(\mathbf{r}, x)$; $\mathbf{r}$ is assumed to represent the neural activity, while x is defined on the space of stimuli. The neural activity plays the role of a latent representation of the stimulus, but it is not set, a priori, to some interpretable feature, such as the presence or the intensity of a Gabor filter in models involving natural images (as in Refs. [159, 181]). In order to constrain sensory areas, we assume the generative model to be implemented in downstream areas and we model its output with a flexible function, a neural network, which outputs a point estimate and an uncertainty about the value of the stimulus [128, 166]. We don't make assumptions about the biological neural circuit implementing the generative model, but the variable x, which corresponds to a perceptual representation of the stimulus in the brain, can be related to behavioral outputs. Mathematically, the encoding distribution, $p_\theta(\mathbf{r}|x)$,

is obtained as a variational approximation of the posterior distribution of the generative model, $q_\psi(\mathbf{r}|x)$, as in previous work. This distribution, however, is defined on the space of neural activity patterns, and not on a set of abstract features. This choice has the drawback of the absence of a simple semantic interpretation of the latent features, but presents the advantage of a natural connection with an encoder based on properties of a neural system, e.g., a set of tuning curves and a model of neural noise. In the case of flexible generative models, different statistics of the latent variables turn out to be optimal. In this sense, the choice of the encoder, as well as the prior of the generative model, is useful to impose a structure on the characteristics of the neural representations.

Optimal tuning width. Our choice of encoding model allows us to compare our results with those of earlier studies that considered the optimal arrangement of neurons with bell-shaped tuning curves in the presence of non-uniform stimulus distributions [46, 21]. While for higher values of the target rate we recover the previously derived allocation of neural resources as a function of the stimulus distribution, the behavior for lower values of the target rate is more varied, and depends on the form of the stimulus distribution. Thus, in our case, the constraint on neural resources has a stronger impact on their optimal allocation than, for example, in Ref. [21], where the bound on mean activity of the population merely acts as a scaling factor, and the behavior of the tuning curves is more constrained. In particular, in Ref. [21] the tuning width was fixed a priori to be inversely proportional to the neural density, to enforce a fixed amount of overlap between tuning curves: it was not optimized. This choice was made to avoid a common issue in this type of calculation: in the case of a one-dimensional stimulus and in the asymptotic limit of infinitely many neurons, the maximization of the mutual information yields the unbiological regime of infinitely narrow tuning curves [23, 11]. Metabolic constraints on the neural activity do not solve the issue, as narrow tuning curves exhibit a low mean activity (as long as the amplitude does not diverge). In our framework, instead, the optimal tuning width, as well the amount of overlap between tuning curves, varies as a function of $\bar{R}$. Moreover, a regime with intermediate values of the constraint, in which tuning curves are broad, exhibits both a smooth generative model (low KL divergence) *and* a low MSE (Fig. 3.3D). This suggests that broad tuning curves are beneficial to obtain smooth generative models, while still allowing high for coding performance.

Interpretation of the resource constraint. The constraint in Eq. (3.14) consists in the divergence between the evoked neural activity and its prior distribution according to the generative model. This formulation is different from usual metabolic constraints which take account for the energetic cost of neural activity [151], and one may ask whether such a constraint, statistical in nature, also comes with a biological interpretation. Since the prior distribution is defined independently from the value of the stimulus, we interpret it as describing statistics of the spontaneous neural activity. The ELBO is optimized when the prior distribution is set to be equal to the so-called 'aggregated posterior', Eq. (3.22) [175]. This is a direct consequence of the assumption of an optimal internal model, since in a well-calibrated internal model the prior equals the average posterior [6]. This basic observation was exploited by Berkes et al. [161] to argue for the evidence of an optimal internal model in V1 for natural images, acquired progressively during development. By comparing the average evoked activity to the spontaneous activity according to the KL divergence, the authors showed that the two quantities become closer during development, and that this phenomenology is specific to naturalistic stimuli.

As we showed, there are multiple ways to achieve a statistically optimal internal model

and to minimize the KL divergence between the two sides of Eq. (3.22), which differ in the value of the rate. The latter instead measures the average KL divergence between the evoked and the spontaneous activity. At low rates, Eq. (3.22) is approximated by relying on the optimization of the encoder parameters which are set so as to make $p_\theta(\mathbf{r}|x)$ similar to the prior for all stimuli; this then results in an unstructured coupling matrix in the prior distribution (Fig. 3.3E, top). Conversely, at high rates, the encoder has a well defined structure which achieves a low distortion, and Eq. (3.22) is approximated by optimizing the parameters of the prior and embedding the structure of the average posterior distribution in the connectivity matrix (Fig. 3.3E, bottom). The value of the target rate can therefore be thought of as cost of imposing structure in spontaneous activity, presumably through circuit properties. In principle, a direct comparison with couplings extracted from experimental data on spontaneous activity is possible, and offers an alternative normative view (as compared, e.g., to pure information maximization, as proposed in Ref. [183]).

Finally, we mention a recent result in statistical mechanics of out-of-equilibrium systems which established a connection between the response of a system to an external perturbation and concepts of information theory, and allows a concrete interpretation of the constraint in terms of metabolic cost. The response of a system to a perturbation, e.g., the presentation of a stimulus, is measured by the difference in the mean value of an observable, e.g., the mean spike count, between the perturbed and unperturbed state. In physics, the fluctuation-dissipation theorem relates the response of a system to the fluctuations (the variance) of the same observable in the unperturbed state; in Ref. [184] the authors derived a generalized version of this relationship. The latter takes the form of a bound to the response of a system which involves the KL divergence between the probability distributions describing the perturbed and unperturbed systems, in our case $p_\theta(\mathbf{r}|x)$ and $q_\psi(\mathbf{r})$ respectively. Several studies pointed out that neural circuits are endowed with homeostatic plasticity mechanisms that set the average activity of the network at rest around a set point [185, 186, 187], and it has also been argued that such point represents an energetic equilibrium [188]. If so, the constraint on the rate represents the metabolic cost of changing the value of the firing rate from its equilibrium set point in response to a sensory perturbation.

VAEs in neuroscience: related studies. VAEs are among the state of the art approaches to unsupervised learning, and in recent years they have been applied in different contexts in neuroscience as models to characterize the neural representation of sensory stimuli. A series of papers have considered neuroscience-inspired VAEs, in which the generative model is based on a decomposition of natural images into a sparse combination of linear features [85]. The latter is the paired with a powerful encoder, which models the sensory encoding process, and a specific prior distribution of the latent variables, to obtain representations similar to the ones observed in the early visual pathway (in V1 and V2) [189, 190, 156]. In these models, the simplicity of the generative distribution prevents posterior collapse. We note that, in our case, we reverse this approach, by imposing a specific and simple form to the encoder (a set of tuning curves), while we assume a flexible decoder. A more complex generative model, instead, is needed to explain neural representations in higher visual areas [191]; to overcome the issue of posterior collapse, the authors used a loss function akin to the one in Eq. (3.16), but the value of β was chosen by hand. In doing so, the authors obtain an empirical advantage in the semantic interpretability of the latent variables, at the cost of abandoning the requirement that the loss function be a bound to the log-likelihood. This, so-called, β-VAE approach was also employed in Ref. [192] to study optimal tuning curves

in a population coding model of spiking neurons similar to ours. In this study, however, the population was constrained to emit one spike only, limiting the number of available activity patterns to N (the number of neurons). Moreover, the encoder and the decoder shared the same parameters; this choice prevented the emergence of multiple alternative neural representations in the $\beta = 1$ case. By varying β, the authors obtained neural representations which differ in the shape of the optimal tuning curves, but, as for $\beta \neq 1$ the ELBO is not optimized, they can result in a poorer fit of the generative model.

S3.5 Supplementary Information

S3.5.1 Optimal heterogeneous allocation of neural resources

We provide an alternative derivation, based on scaling arguments, of the results in Ref. [21]. We consider a population of N neurons, in which neuron i responds to a continuous scalar stimulus, x, according to a bell-shaped tuning curve, $f_i(x)$. We consider a discretization of the stimulus space, $x = \{x_i\}_{i=1}^{L}$, and we denote by d_i the number of neurons whose preferred stimulus is x_i and by w_i their tuning width (S3.4A). The number of neurons encoding information about stimulus x_i scales as

$$\#\text{neurons} = M_i \sim d_i w_i \tag{3.24}$$

as increasing the number of neurons and the tuning width (both of which, we assume, vary sufficiently smoothly with position) each increases the 'coverage' of the stimulus. We assume that neural responses, r, are corrupted by a noise of size η. Through a simple geometric argument (Fig. S3.4B), we estimate the square of the difference between the stimulus estimate based on the activity of neuron j and the true stimulus, i.e., the error, as

$$(\hat{x}_i - x_i)^2 \equiv \Delta x_i^2 \sim \left(\frac{\eta}{f_j'(x_i)}\right)^2, \tag{3.25}$$

where $f_j'(x)$ denotes the slope of the tuning curves at x_i. The derivative of a bell-shaped tuning curve scales as $f'(x) \sim f(x)/w_i$; if noise has a Poisson distribution, the variance of the response is equal to the mean, and we have

$$\Delta x_i^2 \sim \left(\frac{\text{const}}{w_i}\right)^{-2} \sim w_i^2. \tag{3.26}$$

As M independent neurons encode stimulus x_i, we can average the single estimates of the neurons to obtain a more faithful estimate. The variance of this population estimate, i.e., the MSE, for stimulus i, scales as

$$\begin{aligned} \varepsilon_i^2 = \text{Var}\left(\frac{1}{M_i}\sum_{j=1}^{M}(\Delta x_i)_j\right) &= \frac{1}{M^2}\sum_{j=1}^{M_i}(\Delta x_i^2)_j \\ &\sim \frac{w_i^2}{M_i} \\ &\sim \frac{w_i}{d_i}, \end{aligned} \tag{3.27}$$

where in the last line we used Eq. (3.24). By taking the limit of an infinitely fine discretization, $L \to \infty$, and assuming that the population size is large enough such that the quantities d_i and w_i vary smoothly, we can consider a continuum limit with

$$d_i \to d(x), \tag{3.28}$$

the neural density,

$$w_i \to w(x), \tag{3.29}$$

the tuning width, and

$$M_i \to M(x) \tag{3.30}$$

Furthermore, we assume that neurons are arranged so as to ensure a uniform coverage across stimuli, i.e., $M(x) = \text{constant}$, or

$$w(x) \sim \frac{1}{d(x)}. \tag{3.31}$$

The efficient coding hypothesis posits that neurons are arranged so as to maximize the mutual information between stimuli and neural responses. An approximation of the mutual information in terms of the Fisher information, $J(x)$, in the asymptotic limit, can be obtained as [46]

$$I(r, x) = \int dx p(x) \log\left(J(x)\right) + \text{const}, \tag{3.32}$$

where $p(x)$ is the distribution of stimuli and const denotes terms that don't depend on the neural responses. The Fisher information is a lower bound to the variance of any unbiased estimator; if we assume that such bound is tight, we have that

$$J(x) \approx \frac{1}{\varepsilon^2(x)}, \tag{3.33}$$

where $\varepsilon^2(x)$ corresponds to the continuum limit of Eq. (3.27).

We now maximize the mutual information subject to a constraint on the neural resources–here, merely, the number of neurons–by optimizing the sum of the two terms

$$\max_{d(x),w(x)} \left\{ \int dx p(x) \log\left(\frac{d(x)}{w(x)}\right) + \beta \int dx d(x) \right\} = \max_{d(x)} \left\{ \int dx p(x) \log\left(d(x)^2\right) + \beta \int dx d(x) \right\}. \tag{3.34}$$

By taking a functional derivative with respect to $d(x)$ and setting it to zero, we obtain

$$d(x) \sim p(x), \tag{3.35}$$

and, consequently,, the scaling of the MSE as

$$\varepsilon^2(x) \sim \frac{1}{p^2(x)}. \tag{3.36}$$

S3.5.2 Main differences with our model

Our model is similar to the one presented above, but it exhibits some differences which complicate analytical calculations and give rise to more complex behaviors.

- The first difference is in the noise model: we assume binary neurons, while these calculations are carried out for Poisson neurons, an assumption which allows the simplification in Eq. (3.26). When neurons are affected by Poisson noise, increasing the tuning amplitude increases the variance of the noise, while, in our model, at large amplitudes neurons become deterministic ($p(r_i = 1|x) \approx 1$, Fig. 3.2).
- The second difference is that, in our formulation, the tuning width and neural density are free to vary independently, and we can achieve a non-uniform coverage across stimuli.

- The third difference is that we assume a finite population size, rather than working in the asymptotic $N \to \infty$ limit.
- Finally, our loss function is similar to that in Eq. (3.34) for what concerns the first term, which represents the mutual information between stimuli and neural responses (although in our case we have a lower bound, which depends also on the decoder), but the constraint is more intricate due to its dependence on the generative model.

S3.5.3 Supplementary figures

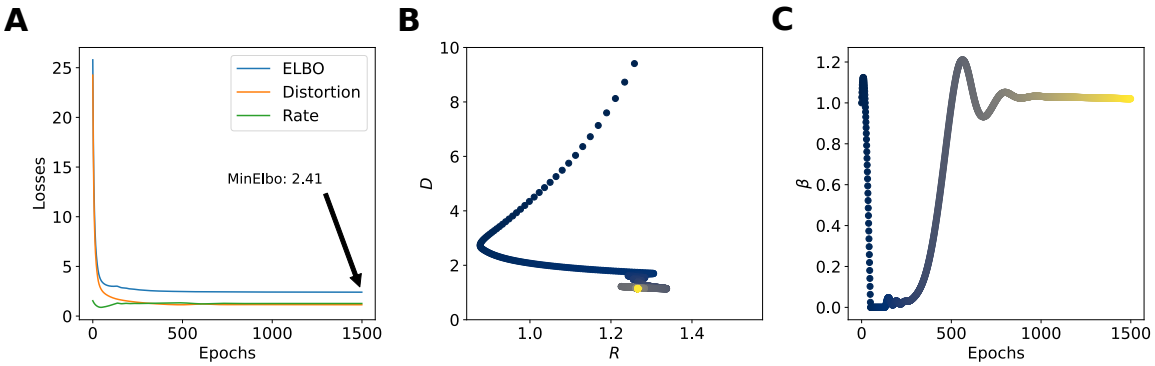

Fig. S3.1 ***Example of training.*** *(**A**) Evolution of the* ELBO, *D, and R with training epochs. (**B**) Joint evolution of R and D in the rate-distortion plane, colored according to the epoch (increasing from blue to yellow). (**C**) Evolution of β during training.*

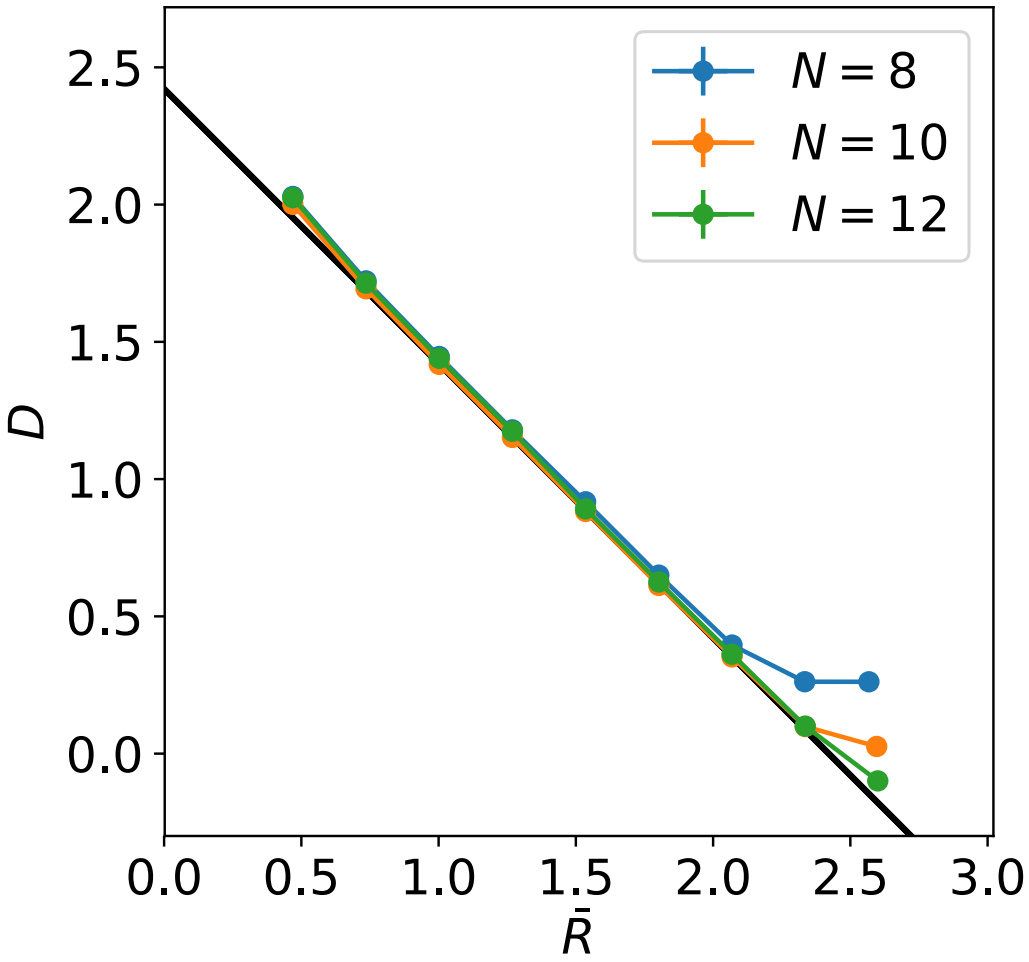

Fig. S3.2 ***Dependence of the rate-distortion curve on the population size.*** *Curves of optimal solutions, $D(\bar{R})$, for different population sizes, N, and theoretical optimum (black curve), $D = H(p) - R$, in the rate-distortion plane.*

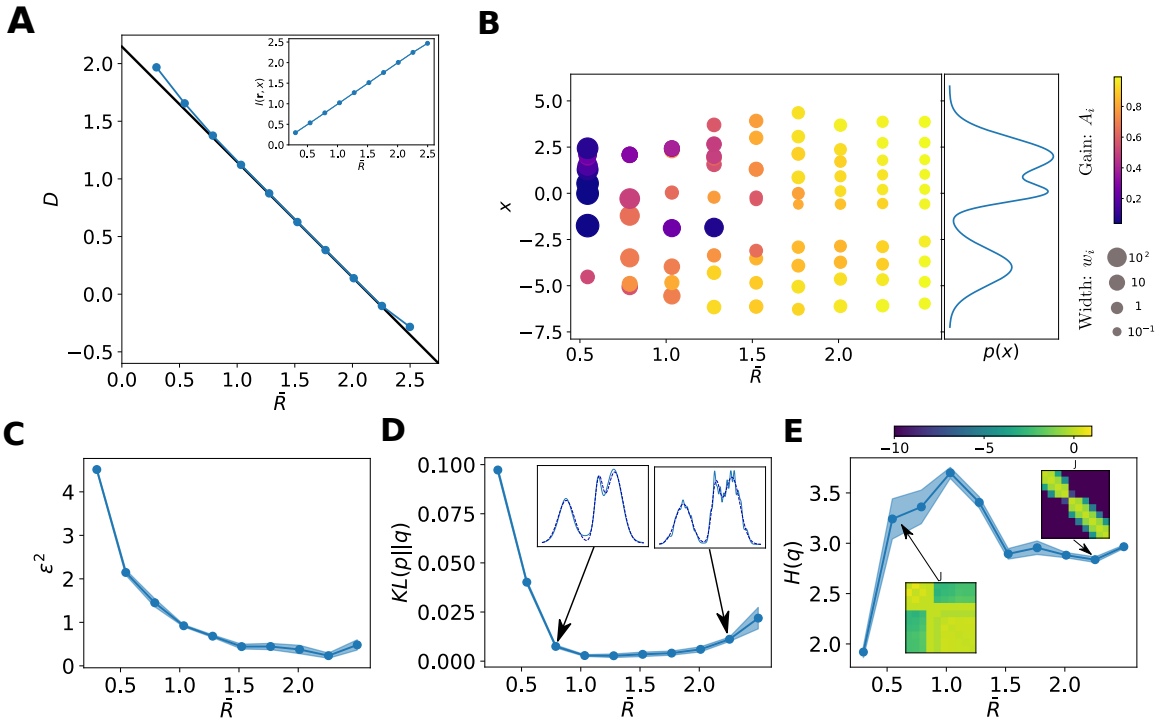

Fig. S3.3 ***Characterization of the optimal solutions as a function of the target rate.*** *Same as Fig. 3.3, but with $p(x)$ a multimodal distribution: a mixture of three Gaussians with means $-4, 0, 2$, variances $1, 0.5, 1$ and mixture coefficients $0.3, 0.2, 0.5$. Panel D illustrates, in the insets, an example of comparison between the generative (solid curve) and the data (dashed curve) distribution.*

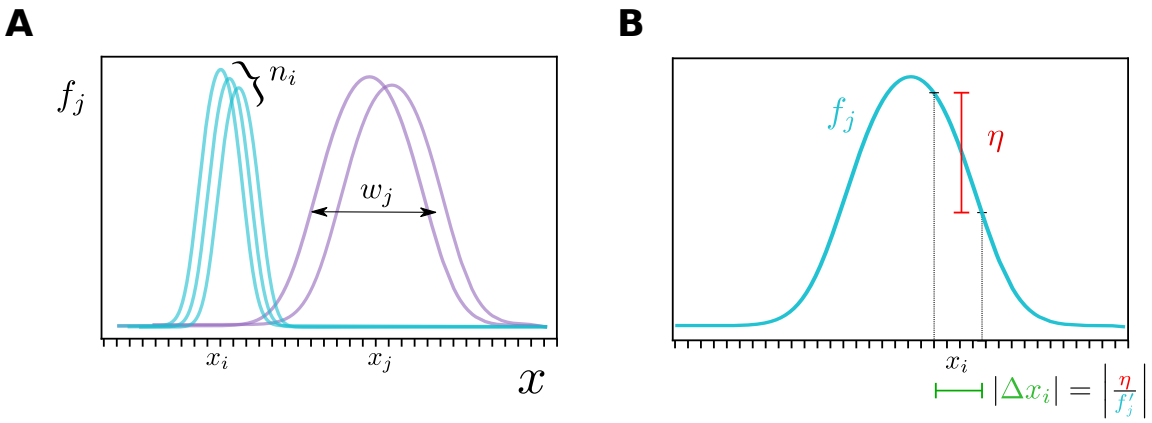

Fig. S3.4 ***Population coding model with bell-shaped tuning curves.*** *(**A**) A one-dimensional stimulus is encoded through bell-shaped tuning curves. The number of neurons whose preferred positions are a given stimulus, x_i, is denoted by n_i, while w_i denotes the tuning width. (**B**) Approximate scaling of the error in a stimulus estimate, Δx_i, when the response of a neuron, f_j, is affected by a noise of size η.*

Broad Discussion

Evolution is a long and selective processes, and it is natural to expect that animals developed strategies to interact with the surrounding environment in a way that optimizes their survival. Among these strategies, we find a system dedicated to encode and process information by means of noisy electrical signals. In this thesis, we discussed different optimality principles which might underlie such information processing and shape neural representations of the external world.

Highly structured neural codes typically require a finely tuned synaptic connectivity, which must either be encoded in the genes or acquired through experience. Our results of the first chapter suggest that, in some cases, irregular neural codes relying on randomness might represent a valid alternative to finely structured ones, yet achieving a high coding performance. There is currently experimental evidence supporting the idea that the brain exploits this randomness, which is inherent to all biological processes, to forge efficient neural representations [193, 40, 35]. Further, modeling techniques which incorporate irregularities have been showed to yield better fits of data from neural recordings, discovering complex selectivity patterns in neurons tuned to sensory stimuli [194, 195]. With an increasing ability to measure and characterize the statistics of large neural populations, we expect a further growth of observations of complex and irregular tuning curves: our work provides a theoretical framework to explain implications of these observations for coding.

The capacity of conveying a high amount of information is not, however, the only characteristic which a 'good' neural code must possess. As we show in the second chapter, it exists a trade-off, in terms of irregularity, between the ideal accuracy of a code and the ease of the decoding process, which depends upon the structure of the decoder itself. It is hard to access to the exact readout schemes used by the brain, but a natural way to test the decoding performance is to measure behavioral output. Recent studies showed that, in a stimulus discrimination task, the accuracy afforded by sensory codes outperforms of orders of magnitudes the behavioral accuracy, implying that the readout of information is suboptimal [196]. Suboptimal readouts might benefit of stimulus and noise correlations which, in principle, decrease the information conveyed, suggesting that information might be encoded suboptimally to facilitate the subsequent decoding process [197]. By recording neural activity in different areas at the same time, we can understand how these subsequent encoding-decoding processes are implemented and what are the biological origins of readout suboptimality and information loss across the hierarchy [198, 199]. Also in this case, large scale recording techniques, as well as the capacity to manipulate artificially neural activity through optogenetic, will improve our understanding of the intersection between information encoding and decoding in neural populations. New theoretical tools will be needed to analyze data and assess optimality criteria, combining normative approaches with empirical

observations and, given the complexity of the systems involved, a possible approach consists in the use of deep artificial neural networks as computational models [95].

Throughout the thesis, we mainly focused on using relatively simple artificial neural networks for such computational models, and one might ask if this work also brings new contributions to the field of artificial intelligence. We mention an interesting connection between the balance between local and global errors, analyzed in the first chapter, and the field of adversarial machine learning. Global errors can be thought of as the analog of the so-called adversarial perturbations, small perturbations of the input data, e.g., small noise added to an image, which cause a large error on the output of a machine learning algorithm, e.g., a neural network classifier labeling a picture of a 'dog' as a 'cat'. Biological neural networks are much less sensible to this kind of errors than their artificial counterpart. Recently, it has been showed that, by adding a regularization inspired by the structure of the neural code in V1 [72], the adversarial robustness of artificial neural networks increases [200]. It might be interesting to explore how other strategies employed by the brain to form 'compressed' neural codes, still avoiding catastrophic errors, could be used to improve artificial neural architectures.

More broadly, improving our understanding of the neural encoding-decoding processes has relevant implications in engineering, where efficient decoding schemes are needed to build brain-machine interfaces and prosthetics. On the clinical side, neural disorders strongly impact the typical information processing system. As an example, in autism, the difficulty in reading other people intentions from movement can be explained by a difference in the internal model of behavior of autistic subjects with respect to typical subjects, but also by a deficit in the readout mechanisms [201]. Understanding which specific stage of the encoding-decoding process manifests anomalies is critical for a global understanding of the disorder and to design target interventions. These applications are outside of the scope of this thesis, which is mostly theoretical and abstracts from many biological details; however, the advantage of abstraction is that "it allows the construction of a semantic level in which one can be absolutely precise." [3]

[3] Edsger Dijkstra, ACM Turing Lecture, 1972.

Résumé long en français

Le cerveau est un exemple fascinant de système complexe. Plusieurs cellules individuelles, les neurones, interagissent entre elles au moyen de signaux électriques, donnant lieu à une riche variété de comportements, dans le but ultime de permettre à l'organisme de survivre dans un environnement tout aussi complexe. L'interaction avec cet environnement joue un rôle central : les neurones codent et traitent des informations sur le monde extérieur, et ils le font avec une précision et une efficacité remarquables, malgré le niveau élevé de bruit qui caractérise tous les systèmes biologiques. Outre l'analyse des données expérimentales, l'étude du code neuronal a largement bénéficié d'approches plus théoriques et normatives. L'hypothèse selon laquelle les réponses neuronales sont organisées de manière à optimiser une certaine fonction d'utilité permet de justifier les premiers principes qui régissent le traitement de l'information dans le cerveau.

Le code neuronal peut être étudié sous deux angles différents. Le processus de *encodage* se réfère à la manière dont les neurones modulent leurs réponses pour transmettre des informations sur les stimuli externes, à partir de la transduction de quantités physiques en activité électrique des neurones, suivie de la propagation de ces signaux à travers différentes zones du cerveau. Le processus de *décodage*, au contraire, est appliquée afin de récupérer des informations pertinentes sur le stimulus à partir de l'activité neuronale. Ce processus de décodage peut être réalisé par un observateur externe, c'est-à-dire le scientifique lors d'une expérience, mais il doit également être mis en œuvre par l'organisme lui-même, afin d'effectuer une action ou un choix en réponse à des stimuli externes.

Au cours de cette thèse, nous étudions comment les critères d'optimalité des processus d'encodage et de décodage concourent à façonner les représentations neuronales des stimuli sensoriels. Nous dérivons les propriétés de codage de modèles de systèmes neuronaux par des calculs analytiques et des simulations numériques, afin d'illustrer les principes généraux de calcul. Ensuite, nous appliquons les cadres théoriques à l'analyse des données d'enregistrements neuronaux pour valider les modèles et fournir des exemples concrets d'instanciation de ces principes. Sur le plan mathématique, nous exploitons des outils issus de la théorie de l'information, de la physique statistique et de l'apprentissage automatique. En particulier, la théorie de l'information est le cadre naturel pour étudier le problème de la communication des signaux, et elle a été appliquée aux neurosciences, dans les études pionnières de Barlow et Attneave, quelques années après sa formalisation par Shannon. Le cadre de la physique statistique est adapté pour décrire comment les interactions entre les neurones donnent lieu à des calculs et à des fonctions cognitives, de la même manière que les propriétés macroscopiques de la matière émergent des interactions complexes entre les particules individuelles. En ce qui concerne l'apprentissage automatique, nous utilisons des réseaux neuronaux artificiels pour modéliser la complexité et la flexibilité des systèmes bi-

ologiques, en étudiant les modèles qui émergent du processus d'apprentissage artificiel.

A1.1 Codage compressive aléatoire

Les neurones transmettent des informations sur le monde extérieur en modulant leurs réponses en fonction des stimuli sensoriels. La fonction décrivant la réponse moyenne des neurones à un stimulus est appelée "courbe de réponse " du neurone [6]. Les écarts par rapport à la réponse moyenne - le "bruit neuronal" - entraînent une ambiguïté quant à l'identité du stimulus codé. Dans un modèle classique avec des neurones caractérisés par des courbes de réponse simples, souvent paramétrées comme des fonctions gaussiennes ou en forme de cloche, l'erreur dans l'estimation du stimulus est généralement inversement proportionnelle à la taille de la population, N.

Cependant, les neurones peuvent présenter des courbes de réponse plus complexes. Dans le cortex enthorinal, la périodicité des courbes de réponse des cellules de la grille, ainsi que leur organisation fonctionnelle en modules, leur permet de représenter les coordonnées spatiales avec une résolution exponentielle, plutôt qu'algébrique comme ci-dessus„ du nombre de neurones [32]. Récemment, de nombreux autres exemples de neurones présentant des courbes de réponse complexes et non structurées ont été identifiés. Ces résultats nous amènent à nous demander si les codes neuronaux hautement efficaces nécessitent une organisation fine, comme dans les cellules de la grille, ou s'ils peuvent être réalisés avec des courbes de réponse complexes et irrégulières.

Nous abordons cette question en étudiant le cas d'un code neuronal "aléatoire": un code de population qui compte sur des courbes de réponse échantillonnées à partir d'un processus stochastique. La seule contrainte sur la forme des courbes de réponse est leur 'lissage', paramétrée par l'échelle de longueur des irrégularités du processus. Nous montrons comment telles courbes d'accord irrégulières pourraient provenir de simples courbes en forme de cloche, d'un réseau neuronal à deux couches avec des poids aléatoires. La largeur d'accord des neurones de la première couche contrôle l'échelle de longueur des irrégularités de ceux de la deuxième couche.

Nous considérons une interprétation géométrique d'un code neuronal, comme une application entre l'espace des stimuli et une courbe dans l'espace à N dimensions de l'activité de la population. Des courbes de réponse simples génèrent une courbe de réponse de la population lisse, ce qui implique que des stimuli similaires correspondent à des réponses proches; en revanche, des courbes de réponse plus complexes donnent lieu à une courbe en serpentin. Cette dernière utilise mieux l'espace des réponses possibles de la population que la première, et on peut donc s'attendre à un codage à plus haute résolution. En effet, lorsque la réponse de la population est corrompue par un bruit d'une magnitude donnée, l'erreur locale sera plus faible dans le cas d'un accord complexe que dans le cas d'un accord simple: en "étirant" la courbe de réponse moyenne sur une trajectoire plus longue dans l'espace des activités possibles de la population, l'accord complexe confère au code une résolution plus élevée par rapport à la gamme de la variable codée. Cependant, dans le cas d'une courbe de réponse moyenne sinueuse et tordue, deux stimuli éloignés sont parfois mis en correspondance avec des modèles d'activité proches. En présence de bruit, cette géométrie donne lieu à des erreurs globales (ou catastrophiques).

Nous avons effectué un calcul analytique approximatif de la contribution de ces deux

types d'erreurs en fonction du niveau d'irrégularité des courbes de réponse, de la taille de la population et de la variance du bruit. Pour une irrégularité optimale des courbes de réponse, lorsque les erreurs locales et globales s'équilibrent, la population neuronale comprime les informations relatives à un stimulus continu dans une représentation à faible dimension, et le code distribué qui en résulte atteint une précision exponentielle en fonction de la taille de la population. Nous généralisons ensuite notre approche au cas d'un stimulus multidimensionnel. Cela nous permet d'appliquer notre modèle à des enregistrements de motoneurones chez le singe, et d'analyser la nature du codage de la population dans ce système [40]. Dans ce contexte, nous quantifions l'avantage relatif des courbes de réponse complexes et irrégulières par rapport aux courbes simples et régulières. En adaptant notre modèle aux données expérimentales, nous discutons des mérites d'un "code compressive" complexe. Dans l'ensemble, notre étude prolonge les travaux théoriques antérieurs sur les cellules de grille et d'autres codes "finement conçus" en proposant qu'une compression efficace de l'information peut se produire de manière robuste même dans le cas de courbes de réponse complexes, mais irrégulières.

A1.2 Décodage de réponses neuronales complexes

Les résultats du premier chapitre, à l'instar des études précédentes sur les codes de population [19, 21], sont obtenus en quantifiant la performance du codage par un décodeur abstrait "idéal", qui a accès à tous les détails du processus d'encodage et aux statistiques du bruit. En pratique, cependant, le processus de décodage nécessite des ressources neuronales, et un tel décodeur idéal peut être difficile à mettre en œuvre. Dans le deuxième chapitre, nous abordons le problème du décodage de l'information véhiculée par des réponses neuronales complexes et irrégulières à travers une architecture neuronale non idéale.

Nous paramétrons un décodeur comme un réseau neuronal à deux couches, une architecture flexible qui présente des propriétés remarquables d'approximation de fonctions [94]. Dans un cadre d'apprentissage supervisé, nous fixons ses paramètres de manière à minimiser l'erreur empirique sur un ensemble d'exemples d'entraînement, constitué de réponses neuronales et des stimuli correspondants qui l'ont évoquée. Les performances sont ensuite évaluées en mesurant l'erreur sur l'ensemble de la distribution des stimuli et des réponses neuronales, ce qui permet de tester la capacité du décodeur à généraliser.

En nous limitant aux réseaux de neurones à deux couches, nous analysons d'abord une architecture spécifique qui peut se rapprocher d'un décodeur idéal. Nous montrons qu'en entraînant le décodeur à reproduire une approximation de la distribution postérieure dans la couche cachée, la capacité de décodage de ce décodeur non-idéal se rapproche de l'idéal. La procédure d'apprentissage d'un tel réseau nécessite une hypothèse sur la nature de la représentation de la couche cachée et, par conséquent, un choix particulier pour la fonction de perte. Nous relâchons ensuite ces hypothèses fortes en considérant un réseau neuronal générique à deux couches formé pour minimiser une fonction de perte basée sur l'erreur. Dans un régime où les paramètres des réseaux neuronaux varient de façon négligeable pendant la minimisation de la perte, également appelé régime "paresseux" [96], la fonction de décodage présente de mauvaises performances lorsque les courbes de réglage sont irrégulières, ce qui entraîne un écart important entre l'erreur idéale et l'erreur non idéale. Au contraire, lorsque le réseau apprend des "caractéristiques" riches à partir des données, il est capable de tirer

parti de la précision supérieure obtenue par des courbes d'accord irrégulières. En faisant varier le nombre de neurones dans la couche cachée, nous mesurons combien de ces caractéristiques sont nécessaires pour décoder efficacement l'information contenue dans l'entrée. Nous constatons que ce nombre est inversement proportionnel à la largeur de la corrélation entre les réponses neuronales. Il en résulte un compromis entre la précision idéale d'un code de population, maximisée lorsque les neurones possèdent des courbes d'accord irrégulières, et la facilité du processus de décodage, qui est facilitée par des réponses neuronales qui varient de façon régulière.

Nous montrons comment la performance du décodeur est limitée par son accès à un ensemble fini d'exemples bruyants ; cette limitation affecte différentes architectures de différentes manières. Nous discutons comment ces limitations et contraintes sur les ressources neuronales allouées au processus de décodage peuvent modifier, et dans certains cas inverser complètement, les critères d'optimalité d'un code neuronal.

A1.3 Codage et décodage efficaces : un cadre de autoencodeur variationnel

L'hypothèse du codage efficace [8] postule que les réponses neuronales sont établies de manière à maximiser l'information sur les stimuli externes, sous des contraintes de ressources biologiques. Cette hypothèse a permis de prédire les réponses neuronales aux stimuli naturels dans diverses zones sensorielles. L'approche typique consiste à spécifier un modèle d'*encodage*, comme une application stochastique entre les stimuli et les réponses neuronales. Les paramètres de ce modèle sont ensuite choisis de manière à optimiser une fonction qui quantifie la performance du codage, par exemple, l'information mutuelle entre les stimuli et les réponses neuronales.

D'autre part, il a été postulé que le cerveau apprend et maintient un modèle interne du monde extérieur, et que la perception sensorielle correspond à un processus d'inférence [153]. Mathématiquement, un tel modèle interne peut être formalisé sous la forme d'un modèle génératif qui décrit comment les stimuli externes sont générés par échantillonnage à partir d'une distribution conditionnelle, compte tenu d'un ensemble de caractéristiques élémentaires "latentes" [154].

Dans la troisième partie, nous considérons une approche de codage efficace étendue : alors que, typiquement, seul le processus d'encodage sensoriel est optimisé, nous considérons conjointement le processus de décodage. En plus d'une classe de transformations d'encodage des stimuli en réponses neuronales dans une zone sensorielle, nous supposons une classe de modèles génératifs mis en œuvre dans la zone en aval. Ceux-ci définissent des applications à partir de modèles d'activité neuronale, correspondant à des variables latentes, vers des distributions sur les stimuli. L'optimalité est atteinte lorsque la distribution générative correspond à la distribution réelle des stimuli dans l'environnement. Si l'on suppose que l'encodeur et le décodeur sont optimisés conjointement dans ce cadre, le système a la structure d'un autoencodeur variationnel (VAE) [160].

Comme dans le cadre classique du codage efficace, le codeur est ici réglé de manière à maximiser une approximation variationnelle de l'information mutuelle entre les stimuli et les réponses neuronales sous une contrainte sur les ressources neuronales. Cependant, un aspect important de cette formulation est que la contrainte, plutôt que d'être imposée

manuellement, est une conséquence directe de l'hypothèse d'un modèle interne optimal. Cette contrainte est obtenue comme la distance statistique entre la distribution provoquée par le stimulus et la distribution antérieure de l'activité neuronale supposée par le modèle génératif. Cette dernière, à son tour, peut être interprétée comme la statistique de l'activité neuronale spontanée ; la contrainte statistique peut donc être considérée comme le coût métabolique des déviations induites par le stimulus par rapport à l'activité neuronale spontanée.

Nous appliquons le cadre théorique à l'étude d'un modèle de codage de population avec des neurones présentant des courbes d'accord en forme de cloche. En capitalisant sur les avancées récentes de la littérature VAE, nous résolvons le problème d'optimisation en fonction de la contrainte sur les ressources neuronales : nous obtenons une famille de solutions qui donnent des modèles génératifs également satisfaisants [162]. Cependant, ces solutions font des prédictions différentes sur les représentations neuronales correspondantes, qui correspondent à des arrangements différents des courbes d'accord, des statistiques d'activité neuronale spontanée et des performances de codage. Des approches connexes ont été explorées dans la littérature, et des prédictions sur l'allocation optimale des ressources de codage, c'est-à-dire les courbes d'accord, en fonction de la distribution du stimulus ont été dérivées [46, 21]. Dans notre cadre, l'allocation optimale des ressources de codage en fonction de la distribution des stimuli varie en fonction de la contrainte. Malgré les différences dans la fonction objective, nos résultats sont conformes aux prédictions précédentes dans un régime faiblement contraint, tandis que des comportements plus complexes sont observés dans un régime fortement contraint. Nos résultats illustrent comment les interactions entre l'encodeur et le modèle interne façonnent les représentations neuronales des stimuli sensoriels.

Plus largement, l'amélioration de notre compréhension des processus d'encodage-décodage neuronaux a des implications pertinentes en ingénierie, où des schémas de décodage efficaces sont nécessaires pour construire des interfaces cerveau-machine et des prothèses. Sur le plan clinique, les troubles neuronaux ont un impact important sur le système de traitement de l'information typique. Comprendre quelle étape spécifique du processus d'encodage-décodage présente des anomalies est essentiel pour une compréhension globale du trouble et pour concevoir des interventions ciblées.

Bibliography

[1] Simone Blanco Malerba, Mirko Pieropan, Yoram Burak, and Rava Azeredo da Silveira. Random Compressed Coding with Neurons. *bioRxiv*, page 2022.01.06.475186, 1 2022.

[2] A. P. Georgopoulos, J. F. Kalaska, R. Caminiti, and J. T. Massey. On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex. *Journal of Neuroscience*, 2(11):1527–1537, 1982.

[3] J. S. Taube, R. U. Muller, and J. B. Ranck. Head-direction cells recorded from the postsubiculum in freely moving rats. I. Description and quantitative analysis. *Journal of Neuroscience*, 10(2):420–435, 1990.

[4] J. P. Miller, G. A. Jacobs, and F. E. Theunissen. Representation of sensory information in the cricket cercal sensory system. I. Response properties of the primary interneurons. *Journal of Neurophysiology*, 66(5):1680–1689, 1991.

[5] F. Bremmer, U. J. Ilg, A. Thiele, C. Distler, and K. P. Hoffmann. Eye position effects in monkey cortex. I. Visual and pursuit-related activity in extrastriate areas MT and MST. *Journal of Neurophysiology*, 77(2):944–961, 1997.

[6] Peter Dayan and L F Abbott. *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. MIT Press, 2001.

[7] Greet Kayaert, Irving Biederman, Hans P. Op De Beeck, and Rufin Vogels. Tuning for shape dimensions in macaque inferior temporal cortex. *European Journal of Neuroscience*, 22(1):212–224, 2005.

[8] H. B. Barlow. Possible Principles Underlying the Transformations of Sensory Messages. *Sensory Communication*, 1(01):216–234, 1961.

[9] Joseph J. Atick and A. Norman Redlich. Towards a Theory of Early Visual Processing. *Neural Computation*, 2(3):308–320, 1990.

[10] Michael S. Lewicki. Efficient coding of natural sounds. *Nature Neuroscience*, 5(4):356–363, 2002.

[11] Kechen Zhang and Terrence J. Sejnowski. Neuronal tuning: To sharpen or broaden? *Neural Computation*, 11(1):75–84, 1999.

[12] M. Bethge, D. Rotermund, and K. Pawelzik. Optimal short-term population coding: When Fisher information fails. *Neural Computation*, 14(10):2317–2351, 2002.

[13] Z. Wang, A. Stocker, and D. Lee. Efficient neural codes that minimize Lp reconstruction error. *Neural Computation*, 28(12):2656–2686, 2016.

[14] Mark D McDonnell and Nigel G Stocks. Maximally informative stimuli and tuning curves for sigmoidal rate-coding neurons and populations. *Physical review letters*, 101(5):058103, 2008.

[15] Julijana Gjorgjieva, Haim Sompolinsky, and Markus Meister. Benefits of pathway splitting in sensory coding. *Journal of Neuroscience*, 34(36):12127–12144, 2014.

[16] David B. Kastner, Stephen A. Baccus, and Tatyana O. Sharpee. Critical and maximally informative encoding between neural populations in the retina. *Proceedings of the National Academy of Sciences of the United States of America*, 112(8):2533–2538, 2015.

[17] Fang Han, Zhijie Wang, and Hong Fan. Determine neuronal tuning curves by exploring optimum firing rate distribution for information efficiency. *Frontiers in Computational Neuroscience*, 11(10):1–11, 2017.

[18] Julijana Gjorgjieva, Markus Meister, and Haim Sompolinsky. Functional diversity among sensory neurons from efficient coding principles. *PLoS computational biology*, 15(11):e1007476, 2019.

[19] Sophie Deneve, Peter E. Latham, and Alexandre Pouget. Reading population codes: A neural implementation of ideal observers. *Nature Neuroscience*, 2(8):740–745, 1999.

[20] Steve Yaeli and Ron Meir. Error-based analysis of optimal tuning functions explains phenomena observed in sensory neurons. *Frontiers in Computational Neuroscience*, 4(130):1–16, 2010.

[21] Deep Ganguli and Eero P. Simoncelli. Efficient sensory encoding and Bayesian inference with heterogeneous neural populations. *Neural Computation*, 26(10):2103–2134, 10 2014.

[22] Michele Fiscella, Felix Franke, Karl Farrow, Jan Müller, Botond Roska, Rava Azeredo da Silveira, and Andreas Hierlemann. Visual coding with a population of direction-selective neurons. *Journal of Neurophysiology*, 114(4):2485–2499, 2015.

[23] H. S. Seung and H. Sompolinsky. Simple models for reading neuronal population codes. *Proceedings of the National Academy of Sciences of the United States of America*, 90(22):10749–10753, 1993.

[24] Philipp Berens, Alexander S. Ecker, Sebastian Gerwinn, Andreas S. Tolias, and Matthias Bethge. Reassessing optimal neural population codes with neurometric functions. *Proceedings of the National Academy of Sciences of the United States of America*, 108(11):4423–4428, 2011.

[25] Jimmy H. J. Kim, Ila Fiete, and David J. Schwab. Superlinear Precision and Memory in Simple Population Codes. *arXiv preprint arXiv:2008.00629*, 2020.

[26] Torkel Hafting, Marianne Fyhn, Sturla Molden, May Britt Moser, and Edvard I. Moser. Microstructure of a spatial map in the entorhinal cortex. *Nature*, 436(7052):801–806, 2005.

[27] Christian F. Doeller, Caswell Barry, and Neil Burgess. Evidence for grid cells in a human memory network. *Nature*, 463(7281):657–661, 2010.

[28] Michael M. Yartsev, Menno P. Witter, and Nachum Ulanovsky. Grid cells without theta oscillations in the entorhinal cortex of bats. *Nature*, 479(7371):103–107, 2011.

[29] Nathaniel J. Killian, Michael J. Jutras, and Elizabeth A. Buffalo. A map of visual space in the primate entorhinal cortex. *Nature*, 7426(2012):761–764, 2012.

[30] Ila R. Fiete, Yoram Burak, and Ted Brookings. What grid cells convey about rat location. *Journal of Neuroscience*, 28(27):6858–6871, 2008.

[31] Xue Xin Wei, Jason Prentice, and Vijay Balasubramanian. A principle of economy predicts the functional architecture of grid cells. *eLife*, 4:e08362, 2015.

[32] Sameet Sreenivasan and Ila Fiete. Grid cells generate an analog error-correcting code for singularly precise neural computation. *Nature Neuroscience*, 14(10):1330–1337, 2011.

[33] Alexander Mathis, Andreas V.M. Herz, and Martin B. Stemmler. Resolution of nested neuronal representations can be exponential in the number of neurons. *Physical Review Letters*, 109(1):1–5, 2012.

[34] Yoram Burak. Spatial coding and attractor dynamics of grid cells in the entorhinal cortex. *Current Opinion in Neurobiology*, 25:169–175, 2014.

[35] Tamir Eliav, Shir R. Maimon, Johnatan Aljadeff, Misha Tsodyks, Gily Ginosar, Liora Las, and Nachum Ulanovsky. Multiscale representation of very large environments in the hippocampus of flying bats. *Science*, 372(6545):eabg4020, 2021.

[36] Gily Ginosar, Johnatan Aljadeff, Yoram Burak, Haim Sompolinsky, Liora Las, and Nachum Ulanovsky. Locally ordered representation of 3D space in the entorhinal cortex. *Nature*, 596(7872):404–409, 2021.

[37] Roddy M. Grieves, Selim Jedidi-Ayoub, Karyna Mishchanchuk, Anyi Liu, Sophie Renaudineau, Eléonore Duvelle, and Kate J. Jeffery. Irregular distribution of grid cell firing fields in rats exploring a 3D volumetric space. *Nature Neuroscience*, 24(11):1567–1573, 2021.

[38] Siddhartha C. Kadia and Xiaoqin Wang. Spectral integration in A1 of awake primates: Neurons with single- and multipeaked tuning characteristics. *Journal of Neurophysiology*, 89(3):1603–1622, 2003.

[39] Nicholas James Sofroniew, Yurii A. Vlasov, Samuel Andrew Hires, Jeremy Freeman, and Karel Svoboda. Neural coding in barrel cortex during whisker-guided locomotion. *eLife*, 4:e12559, 2015.

[40] Hagai Lalazar, L. F. Abbott, and Eilon Vaadia. Tuning Curves for Arm Posture Control in Motor Cortex Are Consistent with Random Connectivity. *PLoS Computational Biology*, 12(5):1–27, 2016.

[41] Quentin Gaucher, Mariangela Panniello, Aleksandar Z. Ivanov, Johannes C. Dahmen, Andrew J. King, and Kerry M.M. Walker. Complexity of frequency receptive fields predicts tonotopic variability across species. *eLife*, 9:e53462, 2020.

[42] Peter E. Welinder, Yoram Burak, and Ila R. Fiete. Grid cells: The position code, neural network models of activity, and the problem of learning. *Hippocampus*, 18(12):1283–1300, 2008.

[43] J. H. Van Hateren and D. L. Ruderman. 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(1412):2315–2320, 1998.

[44] Eizaburo Doi, Jeffrey L. Gauthier, Greg D. Field, Jonathon Shlens, Alexander Sher, Martin Greschner, Timothy A. Machado, Lauren H. Jepson, Keith Mathieson, Deborah E. Gunning, Alan M. Litke, Liam Paninski, E. J. Chichilnisky, and Eero P. Simoncelli. Efficient coding of spatial information in the primate retina. *Journal of Neuroscience*, 32(46):16256–16264, 2012.

[45] Li Zhaoping. *Understanding Vision: Theory, Models, and Data*. Number 4. OUP Oxford, 2014.

[46] Nicolas Brunel and Jean Pierre Nadal. Mutual Information, Fisher Information, and Population Coding. *Neural Computation*, 10(7):1731–1757, 1998.

[47] Arseny Finkelstein, Nachum Ulanovsky, Misha Tsodyks, and Johnatan Aljadeff. Optimal dynamic coding by mixed-dimensionality neurons in the head-direction system of bats. *Nature Communications*, 9(1):1–17, 2018.

[48] R. E. Kettner, A. B. Schwartz, and A. P. Georgopoulos. 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(8):2938–2947, 1988.

[49] Wei Wang, Sherwin S. Chan, Dustin A. Heldman, and Daniel W. Moran. Motor cortical representation of position and velocity during reaching. *Journal of Neurophysiology*, 97(6):4258–4270, 2007.

[50] Richard A Andersen, Greg K Essick, and Ralph M Siegel. Encoding of spatial location by posterior parietal neurons. *Science*, 230(4724):456–458, 1985.

[51] Takafumi Arakaki, G. Barello, and Yashar Ahmadian. Inferring neural circuit structure from datasets of heterogeneous tuning curves. *PLoS Computational Biology*, 15(4):e1006816–, 2019.

[52] H. S. Seung and D. D. Lee. The manifold ways of perception. *Science*, 290(5500):2268–2269, 2000.

[53] Juan A. Gallego, Matthew G. Perich, Lee E. Miller, and Sara A. Solla. Neural Manifolds for the Control of Movement. *Neuron*, 94(5):978–984, 2017.

[54] Stefano Fusi, Earl K. Miller, and Mattia Rigotti. Why neurons mix: High dimensionality for higher cognition. *Current Opinion in Neurobiology*, 37:66–74, 2016.

[55] Stefano Recanatesi, Serena Bradde, Vijay Balasubramanian, Nicholas A Steinmetz, and Eric Shea-Brown. A scale-dependent measure of system dimensionality. *Patterns*, 3(8):100555, 2022.

[56] John P. Cunningham and Byron M. Yu. Dimensionality reduction for large-scale neural recordings. *Nature Neuroscience*, 17(11):1500–1509, 2014.

[57] Peiran Gao, Eric Trautmann, Byron Yu, Gopal Santhanam, Stephen Ryu, Krishna Shenoy, and Surya Ganguli. A theory of multineuronal dimensionality, dynamics and measurement. *bioRxiv*, page 214262, 1 2017.

[58] Rava Azeredo da Silveira and Fred Rieke. The Geometry of Information Coding in Correlated Neural Populations. *Annu. Rev. Neurosci.*, pages 1–30, 2021.

[59] Yuval Harel and Ron Meir. Optimal multivariate tuning with neuron-level and population-level energy constraints. *Neural Computation*, 32(4):794–828, 2020.

[60] Claude E. Shannon. Communication in the Presence of Noise. *Proceedings of the IRE*, 37(1):10–21, 1949.

[61] Marcelo A. Montemurro and Stefano Panzeri. Optimal tuning widths in population coding of periodic variables. *Neural Computation*, 18(7):1555–1576, 2006.

[62] Xue-xin Wei and Alan A Stocker. Efficient coding provides a direct link between prior and likelihood in perceptual Bayesian inference. *Advances in Neural Information Processing Systems*, 25, 2012.

[63] Thomas E. Yerxa, Eric Kee, Michael R. DeWeese, and Emily A. Cooper. Efficient sensory coding of multidimensional stimuli. *PLoS computational biology*, 16(9):e1008146, 2020.

[64] Stefan D. Wilke and Christian W. Eurich. Representational accuracy of stochastic neural populations. *Neural Computation*, 14(1):155–189, 2002.

[65] Maoz Shamir and Haim Sompolinsky. Implications of neuronal diversity on population coding. *Neural Computation*, 18(8):1951–1986, 2006.

[66] Michael J. Berry, Felix Lebois, Avi Ziskind, and Rava Azeredo da Silveira. Functional diversity in the retina improves the population code. *Neural Computation*, 31(2):270–311, 2 2019.

[67] Omri Barak, Mattia Rigotti, and Stefano Fusi. The sparseness of mixed selectivity neurons controls the generalization-discrimination trade-off. *Journal of Neuroscience*, 33(9):3844–3856, 2013.

[68] Baktash Babadi and Haim Sompolinsky. Sparseness and Expansion in Sensory Representations. *Neuron*, 83(5):1213–1226, 2014.

[69] Ashok Litwin-Kumar, Kameron Decker Harris, Richard Axel, Haim Sompolinsky, and L. F. Abbott. Optimal Degrees of Synaptic Connectivity. *Neuron*, 93(5):1153–1164, 2017.

[70] Shreya Saxena and John P. Cunningham. Towards the neural population doctrine. *Current Opinion in Neurobiology*, 55:103–111, 2019.

[71] L. F. Abbott, Edmund T. Rolls, and Martin J. Tovee. Representational capacity of face coding in monkeys. *Cerebral Cortex*, 6(3):498–505, 1996.

[72] Carsen Stringer, Marius Pachitariu, Nicholas Steinmetz, Matteo Carandini, and Kenneth D. Harris. High-dimensional geometry of population responses in visual cortex. *Nature*, 571(7765):361–365, 2019.

[73] Dmitry Kobak, Jose L. Pardo-Vazquez, Mafalda Valente, Christian K. Machens, and Alfonso Renart. State-dependent geometry of population activity in rat auditory cortex. *eLife*, 8:1–27, 2019.

[74] Nikolaus Kriegeskorte and Xue Xin Wei. Neural tuning and representational geometry. *Nature Reviews Neuroscience*, 22(11):703–718, 2021.

[75] Emmanuel J. Candes and Terence Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Transactions on Information Theory*, 52(12):5406–5425, 2006.

[76] David L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.

[77] Richard Baraniuk, Mark Davenport, Ronald DeVore, and Michael Wakin. A simple proof of the restricted isometry property for random matrices. *Constructive Approximation*, 28(3):253–263, 2008.

[78] Minjoon Kouh and Tomaso Poggio. A canonical neural circuit for cortical nonlinear operations. *Neural Computation*, 20(6):1427–1451, 2008.

[79] Matteo Carandini and David J. Heeger. Normalization as a canonical neural computation. *Nature Reviews Neuroscience*, 13(1):51–62, 2012.

[80] Blake Bordelon, Abdulkadir Canatar, and Cengiz Pehlevan. Spectrum dependent learning curves in kernel regression and wide neural networks. *International Conference on Machine Learning*, pages 1024–1034, 2020.

[81] Blake Bordelon, John A Paulson, and Cengiz Pehlevan. Population Codes Enable Learning from Few Examples By Shaping Inductive Bias. *bioRxiv*, page 2021.03.30.437743, 2021.

[82] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT press, 2005.

[83] Giacomo Livan, Marcel Novaes, and Pierpaolo Vivo. *Introduction to Random Matrices - Theory and Practice*. Springer Cham, 2018.

[84] Brian S Everitt and Anders Skrondal. *The Cambridge dictionary of statistics*. Cambridge University Press, 4 edition, 2018.

[85] Bruno A. Olshausen and David J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996.

[86] Emanuel Todorov. Cosine tuning minimizes motor errors. *Neural Computation*, 14(6):1233–1260, 2002.

[87] Evan C. Smith and Michael S. Lewicki. Efficient auditory coding. *Nature*, 439(7079):978–982, 2006.

[88] Alessandro Campa, Paolo Del Giudice, Nestor Parga, and Jean Pierre Nadal. Maximization of mutual information in a linear noisy network: A detailed study. *Network: Computation in Neural Systems*, 6(3):449–468, 1995.

[89] Xue Xin Wei and Alan A. Stocker. Mutual information, fisher information, and efficient coding. *Neural Computation*, 28(2):305–326, 2 2016.

[90] Peggy Seriès, Peter E. Latham, and Alexandre Pouget. Tuning curve sharpening for orientation selectivity: Coding efficiency and the impact of correlations. *Nature Neuroscience*, 7(10):1129–1135, 2004.

[91] Stuart Yarrow, Edward Challis, and Peggy Seriès. Fisher and Shannon information in finite neural populations. *Neural Computation*, 24(7):1740–1780, 2012.

[92] Ingmar Kanitscheider, Ruben Coen-Cagli, Adam Kohn, and Alexandre Pouget. Measuring Fisher Information Accurately in Correlated Neural Populations. *PLoS Computational Biology*, 11(6):1–27, 2015.

[93] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[94] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[95] Guangyu Robert Yang and Xiao Jing Wang. Artificial Neural Networks for Neuroscientists: A Primer. *Neuron*, 107(6):1048–1070, 2020.

[96] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in Neural Information Processing Systems*, 31, 2018.

[97] Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. *Advances in Neural Information Processing Systems*, 32, 2019.

[98] Emilio Salinas. How behavioral constraints may determine optimal sensory representations. *PLoS Biology*, 4(12):2383–2392, 2006.

[99] Biraj Pandey, Marius Pachitariu, Bingni W Brunton, and Kameron Decker Harris. Structured random receptive fields enable informative sensory encodings. *PLOS Computational Biology*, 18(10):e1010484, 2022.

[100] Christopher M Bishop and M. Nasrabadi Nasser. *Pattern Recognition and Machine Learning*, volume 4. Springer, New York, 2006.

[101] Andrew M. Saxe, James L. McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *2nd International Conference on Learning Representations, ICLR*, volume 2, 2014.

[102] Haozhe Shan and Haim Sompolinsky. A Minimum Perturbation Theory of Deep Perceptual Learning. *bioRxiv*, page 2021.10.05.463260, 1 2022.

[103] Arthur Jacot, Berfin Simsek, Francesco Spadaro, Clément Hongler, and Franck Gabriel. Implicit regularization of random feature models. In *37th International Conference on Machine Learning, ICML*, pages 4631–4640. PMLR, 2020.

[104] Chiyuan Zhang, Benjamin Recht, Samy Bengio, Moritz Hardt, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *5th International Conference on Learning Representations, ICLR*, 2017.

[105] Lenka Zdeborová. Understanding deep learning is also a job for physicists. *Nature Physics*, 16(6):602–604, 2020.

[106] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

[107] Jaehoon Lee, Lechao Xiao, Samuel S. Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide Neural Networks of Any Depth Evolve as Linear Models Under Gradient Descent. *Advances in neural information processing systems*, 32, 2019.

[108] B Scholkopf and Alexander J. Smola. Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. *MIT Press*, 2001.

[109] Amit Daniely, Roy Frostig, and Yoram Singer. Toward deeper understanding of neural networks: The power of initialization and a dual view on expressivity. *Advances in neural information processing systems*, 29, 2016.

[110] Alberto Bietti and Julien Mairal. On the Inductive Bias of Neural Tangent Kernels. *Advances in Neural Information Processing Systems*, 32, 2019.

[111] Behrooz Ghorbani, Song Mei, Theodor Misiakiewicz, and Andrea Montanari. Linearized two-layers neural networks in high dimension. *The Annals of Statistics*, 49(2):1029–1054, 4 2019.

[112] Alberto Bietti and Francis Bach. Deep Equals Shallow for ReLU Networks in Kernel Regimes. *arXiv preprint arXiv:2009.14397*, 9 2020.

[113] Blake Woodworth, Suriya Gunasekar, Jason D Lee, Edward Moroshko, Pedro Savarese, Itay Golan, Daniel Soudry, and Nathan Srebro. Kernel and rich regimes in overparametrized models. In *Conference on Learning Theory*, pages 3635–3673. PMLR, 2020.

[114] Anthony J. Bell and Terrence J. Sejnowski. The 'independent components' of natural scenes are edge filters. *Vision Research*, 37(23):3327–3338, 1997.

[115] Xaq Pitkow and Markus Meister. Decorrelation and efficient coding by retinal ganglion cells. *Nature Neuroscience*, 15(4):628–635, 2012.

[116] Eizaburo Doi and Michael S. Lewicki. A Simple Model of Optimal Population Coding for Sensory Systems. *PLoS Computational Biology*, 10(8):e1003761, 2014.

[117] Valerio Mante, David Sussillo, Krishna V. Shenoy, and William T. Newsome. Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature*, 503(7474):78–84, 2013.

[118] Rishidev Chaudhuri, Berk Gerçek, Biraj Pandey, Adrien Peyrache, and Ila Fiete. The intrinsic attractor manifold and population dynamics of a canonical cognitive circuit across waking and sleep. *Nature Neuroscience*, 22(9):1512–1520, 2019.

[119] Juan A. Gallego, Matthew G. Perich, Raeed H. Chowdhury, Sara A. Solla, and Lee E. Miller. Long-term stability of cortical population dynamics underlying consistent behavior. *Nature Neuroscience*, 23(2):260–270, 2020.

[120] Florian Meier, Raphaël Dang-Nhu, and Angelika Steger. Adaptive Tuning Curve Widths Improve Sample Efficient Learning. *Frontiers in Computational Neuroscience*, 14:1–13, 2020.

[121] Abdulkadir Canatar, Blake Bordelon, and Cengiz Pehlevan. Spectral bias and task-model alignment explain generalization in kernel regression and infinitely wide neural networks. *Nature Communications*, 12(1):1–12, 2021.

[122] Nicholas Lange, C. M. Bishop, and B. D. Ripley. Neural Networks for Pattern Recognition. *Journal of the American Statistical Association*, 92(440), 1997.

[123] Christopher M Bishop. Training with Noise is Equivalent to Tikhonov Regularization. *Neural Computation*, 7(1):108–116, 1995.

[124] Mark A. Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal*, 37(2):233–243, 1991.

[125] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre Antoine Manzagol. Stacked denoising autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. *Journal of Machine Learning Research*, 11(12), 2010.

[126] Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio. Contractive auto-encoders: Explicit invariance during feature extraction. *Proceedings of the 28th International Conference on Machine Learning, ICML 2011*, pages 833–840, 2011.

[127] Apostolos P. Georgopoulos, Andrew B. Schwartz, and Ronald E. Kettner. Neuronal population coding of movement direction. *Science*, 233(4771):1416–1419, 1986.

[128] Wei Ji Ma, Jeffrey M. Beck, Peter E. Latham, and Alexandre Pouget. Bayesian inference with probabilistic population codes. *Nature Neuroscience*, 9(11):1432–1438, 2006.

[129] Mehrdad Jazayeri and J. Anthony Movshon. Optimal representation of sensory information by neural populations. *Nature Neuroscience*, 9(5):690–696, 2006.

[130] David Marr and W Thomas Thach. A theory of cerebellar cortex. In *From the Retina to the Neocortex*, pages 11–50. Springer, 1991.

[131] James S Albus. A theory of cerebellar function. *Mathematical biosciences*, 10(1-2):25–61, 1971.

[132] Omri Barak, David Sussillo, Ranulfo Romo, Misha Tsodyks, and L. F. Abbott. From fixed points to chaos: Three models of delayed discrimination. *Progress in Neurobiology*, 103:214–222, 2013.

[133] Jeffrey Beck, Vikranth R Bejjanki, and Alexandre Pouget. Insights from a simple expression for linear fisher information in a recurrently connected population of spiking neurons. *Neural computation*, 23(6):1484–1502, 2011.

[134] Nikolaus Kriegeskorte and Pamela K. Douglas. Interpreting encoding and decoding models. *Current Opinion in Neurobiology*, 55:167–179, 2019.

[135] Martin Stemmler, Alexander Mathis, and Andreas V.M. Herz. Connecting multiple spatial scales to decode the population activity of grid cells. *Science Advances*, 1(11):e1500816, 2015.

[136] Alex Damian, Jason D Lee, Mahdi Soltanolkotabi, Po-Ling Loh, and Maxim Raginsky. Neural Networks can Learn Representations with Gradient Descent. In *Proceedings of Machine Learning Research*, volume 178, pages 5413–5452, 2022.

[137] Timo Flesch, Keno Juechems, Tsvetomira Dumbalska, Andrew Saxe, and Christopher Summerfield. Orthogonal representations for robust context-dependent task performance in brains and neural networks. *Neuron*, 110(7):1258–1270, 4 2022.

[138] Vladimir Vapnik. Principles of risk minimization for learning theory. *Advances in neural information processing systems*, 4, 1992.

[139] Michael D. Richard and Richard P. Lippmann. Neural Network Classifiers Estimate Bayesian a posteriori Probabilities. *Neural Computation*, 3(4):461–483, 1991.

[140] Ali Rahimi and Benjamin Recht. Uniform approximation of functions with random bases. In *46th Annual Allerton Conference on Communication, Control, and Computing*, 2008.

[141] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. *Advances in Neural Information Processing Systems*, 20, 2009.

[142] H. V. Henderson and S. R. Searle. On Deriving the Inverse of a Sum of Matrices. *SIAM Review*, 23(1), 1981.

[143] D D Kosambi. *Statistics in function space*. Springer, New Delhi, 2016.

[144] Hermann König. *Eigenvalue distribution of compact operators*, volume 16. Birkhäuser, 2013.

[145] Christopher T H Baker. *The numerical treatment of integral equations*. Oxford University Press, 1977.

[146] Sanjeev Arora, Simon S. Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-Grained Analysis of Optimization and Generalization for Overparameterized Two-Layer Neural Networks. *International Conference on Machine Learning*, pages 322–332, 1 2019.

[147] Christopher K.I. Williams. Computation with Infinite Neural Networks. *Advances in neural information processing systems*, 9, 1998.

[148] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B. Shah. Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1):65–98, 2017.

[149] Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR*, 2015.

[150] E. P. Simoncelli and B. A. Olshausen. Natural image statistics and neural representation. *Annual Review of Neuroscience*, 24(1):1193–1216, 2001.

[151] Simon B Laughlin, Rob R de Ruyter van Steveninck, and John C Anderson. The metabolic cost of neural information. *Nature neuroscience*, 1(1):36–41, 1998.

[152] Il Memming Park and Jonathan Pillow. Bayesian Efficient Coding. *bioRxiv*, page 178418, 2017.

[153] Hermann Von Helmholtz. Helmholtz's treatise on physiological optics. *Optometry and Vision Science*, 4(11), 1927.

[154] Peter Dayan, Geoffrey E Hinton, Radford M Neal, and Richard S Zemel. The helmholtz machine. *Neural computation*, 7(5):889–904, 1995.

[155] Martin J. Wainwright and Eero P. Simoncelli. Scale mixtures of Graussians and the statistics of natural images. In *Advances in Neural Information Processing Systems*, 2000.

[156] Ferenc Csikor, Balázs Meszéna, Bence Szabó, and Gergő Orbán. Top-down inference in an early visual cortex inspired hierarchical Variational Autoencoder. *arXiv preprint arXiv:2206.00436*, 6 2022.

[157] Richard S Zemel, Peter Dayan, and Alexandre Pouget. Probabilistic Interpretation of Population Codes. *Neural computation*, 10(2):403–430, 1998.

[158] Tai Sing Lee and David Mumford. Hierarchical Bayesian inference in the visual cortex. *JOSA A*, 20(7):1434–1448, 2003.

[159] Patrik O. Hoyer and Aapo Hyvärinen. Interpreting neural response variability as Monte Carlo sampling of the posterior. *Advances in Neural Information Processing Systems*, 15, 2003.

[160] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, 2014.

[161] P Berkes, G Orbán, M Lengyel, and J Fiser. Spontaneous cortical activity reveals hallmarks of an optimal internal model of the environment. *Science*, 331(6013):83–88, 2011.

[162] Alexander A. Alemi, Ben Poole, Ian Fische, Joshua V. Dillon, Rif A. Saurous, and Kevin Murphy. Fixing a broken elbo. *35th International Conference on Machine Learning, ICML*, 1:245–265, 2018.

[163] Emilio Salinas and L. F. Abbott. Vector reconstruction from firing rates. *Journal of Computational Neuroscience*, 1(1):89–107, 1994.

[164] Tatyana O. Sharpee and John A. Berkowitz. Linking neural responses to behavior with information-preserving population vectors. *Current Opinion in Behavioral Sciences*, 29:37–44, 10 2019.

[165] Elad Schneidman, Michael J. Berry, Ronen Segev, and William Bialek. Weak pairwise correlations imply strongly correlated network states in a neural population. *Nature*, 440(7087):1007–1012, 2006.

[166] Edgar Y. Walker, R. James Cotton, Wei Ji Ma, and Andreas S. Tolias. A neural basis of probabilistic computation in visual cortex. *Nature Neuroscience*, 23(1):122–129, 2020.

[167] Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. Ladder variational autoencoders. *Advances in Neural Information Processing Systems*, 29, 2016.

[168] Kenneth Joseph Arrow, Hirofumi Azawa, Leonid Hurwicz, Hirofumi Uzawa, Hollis Burnley Chenery, Selmer M Johnson, and Samuel Karlin. *Studies in linear and non-linear programming*, volume 2. Stanford University Press, 1958.

[169] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2016.

[170] Peng Yi and Shinung Ching. Multiple timescale online learning rules for information maximization with energetic constraints. *Neural Computation*, 31(5):943–979, 5 2019.

[171] Anthony M V Jakob and Samuel J Gershman. Rate-distortion theory of neural coding and its implications for working memory. *bioRxiv*, page 2022.02.28.482269, 1 2022.

[172] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables. *arXiv preprint arXiv:1611.00712*, 11 2016.

[173] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[174] Jason Tyler Rolfe. Discrete Variational Autoencoders. *arXiv preprint arXiv:1609.02200*, 9 2016.

[175] Jakub M. Tomczak and Max Welling. VAE with a VampPrior. *International Conference on Artificial Intelligence and Statistics*, pages 1214–1223, 5 2017.

[176] S. R. Dalal and W. J. Hall. Approximating Priors by Mixtures of Natural Conjugate Priors. *Journal of the Royal Statistical Society: Series B (Methodological)*, 45(2), 1983.

[177] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[178] Alessandro Achille and Stefano Soatto. Information Dropout: Learning Optimal Representations Through Noisy Computation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2897–2905, 2018.

[179] Xue Xin Wei and Alan A. Stocker. A Bayesian observer model constrained by efficient coding can explain 'anti-Bayesian' percepts. *Nature Neuroscience*, 18(10):1509–1517, 2015.

[180] Deep Ganguli and Eero P. Simoncelli. Neural and perceptual signatures of efficient sensory coding. *arXiv preprint arXiv:1603.00058*, 2 2016.

[181] Gergő Orbán, Pietro Berkes, József Fiser, and Máté Lengyel. Neural Variability and Sampling-Based Probabilistic Representations in the Visual Cortex. *Neuron*, 92(2):530–543, 2016.

[182] Eszter Vertes and Maneesh Sahani. Flexible and accurate inference and learning for deep generative models. *Advances in Neural Information Processing Systems*, 31, 2018.

[183] Gašper Tkačik, Jason S Prentice, Vijay Balasubramanian, and Elad Schneidman. Optimal population coding by noisy spiking neurons. *PNAS*, 107(32):14419–14424, 2010.

[184] Andreas Dechant and Shin-Ichi Sasa. Fluctuation-response inequality out of equilibrium. *Proceedings of the National Academy of Sciences*, 117(12):6430–6436, 2020.

[185] Gina G. Turrigiano and Sacha B. Nelson. Homeostatic plasticity in the developing nervous system. *Nature Reviews Neuroscience*, 5(2):97–107, 2004.

[186] Keith B. Hengen, Mary E. Lambo, Stephen D. VanHooser, Donald B. Katz, and Gina G. Turrigiano. Firing rate homeostasis in visual cortex of freely behaving rodents. *Neuron*, 80(2):335–342, 2013.

[187] Keith B. Hengen, Alejandro Torrado Pacheco, James N. McGregor, Stephen D. Van Hooser, and Gina G. Turrigiano. Neuronal Firing Rate Homeostasis Is Inhibited by Sleep and Promoted by Wake. *Cell*, 165(1):180–191, 3 2016.

[188] Chaitanya Chintaluri and Tim P Vogels. Metabolically driven action potentials serve neuronal energy homeostasis and protect from reactive oxygen species. *bioRxiv*, page 2022.10.16.512428, 1 2022.

[189] Gabriel Barello, Adam S Charles, and Jonathan W Pillow. Sparse-Coding Variational Auto-Encoders. *bioRxiv*, page 399246, 1 2018.

[190] Laurence Aitchison, Guillaume Hennequin, and Mate Lengyel. Sampling-based probabilistic inference emerges from learning in neural circuits with a cost on reliability. *arXiv preprint arXiv:1807.08952*, 7 2018.

[191] Irina Higgins, Le Chang, Victoria Langston, Demis Hassabis, Christopher Summerfield, Doris Tsao, and Matthew Botvinick. Unsupervised deep learning identifies semantic disentanglement in single inferotemporal face patch neurons. *Nature communications*, 12(1):1–14, 2021.

[192] Guy Aridor, Francesco Grechi, and Michael Woodford. Adaptive Efficient Coding: A Variational Auto-encoder Approach. *bioRxiv*, page 2020.05.29.124453, 1 2020.

[193] Sophie J.C. Caron, Vanessa Ruta, L. F. Abbott, and Richard Axel. Random convergence of olfactory inputs in the Drosophila mushroom body. *Nature*, 497(7447):113–117, 2013.

[194] Anqi Wu, Nicholas A. Roy, Stephen Keeley, and Jonathan W. Pillow. Gaussian process based nonlinear latent structure discovery in multivariate spike train data. *Advances in Neural Information Processing Systems*, 30, 2017.

[195] Anqi Wu, Stan L. Pashkovski, Sandeep Robert Datta, and Jonathan W. Pillow. Learning a latent manifold of odor representations from neural responses in piriform cortex. *Advances in Neural Information Processing Systems*, 2018-Decem(NeurIPS):5378–5388, 2018.

[196] Carsen Stringer, Michalis Michaelos, Dmitri Tsyboulski, Sarah E Lindo, and Marius Pachitariu. High-precision coding in visual cortex. *Cell*, 184(10):2767–2778, 2021.

[197] Martina Valente, Giuseppe Pica, Giulio Bondanelli, Monica Moroni, Caroline A Runyan, Ari S Morcos, Christopher D Harvey, and Stefano Panzeri. Correlations enhance the behavioral readout of neural population activity in association cortex. *Nature neuroscience*, 24(7):975–986, 2021.

[198] Stefano Panzeri, Christopher D. Harvey, Eugenio Piasini, Peter E. Latham, and Tommaso Fellin. Cracking the Neural Code for Sensory Perception by Combining Statistics, Intervention, and Behavior, 2017.

[199] Stefano Panzeri, Monica Moroni, Houman Safaai, and Christopher D Harvey. The structures and functions of correlations in neural population codes. *Nature Reviews Neuroscience*, 23(9):551–567, 2022.

[200] Josue Nassar, Piotr Sokol, SueYeon Chung, Kenneth D Harris, and Il Memming Park. On 1/n neural representation and robustness. *Advances in Neural Information Processing Systems*, 33, 2020.

[201] Noemi Montobbio, Andrea Cavallo, Dalila Albergo, Caterina Ansuini, Francesca Battaglia, Jessica Podda, Lino Nobili, Stefano Panzeri, and Cristina Becchio. Intersecting kinematic encoding and readout of intention in autism. *Proceedings of the National Academy of Sciences of the United States of America*, 119(5):e2114648119, 2022.

RÉSUMÉ

Au cours de cette thèse, nous étudions les principes qui sous-tendent le codage optimal de l'information dans les systèmes neuronaux, en combinant des modèles de la issus de la théorie de l'information et de l'apprentissage machine et l'analyse de données expérimentales. Une grande partie des travaux théoriques sur le codage efficace s'est concentrée sur les neurones dont la réponse moyenne en fonction du stimulus–la courbe de réponse– peut être décrite par une fonction simple. Néanmoins, les neurones présentent souvent des courbes de réponse plus complexes: dans les cellules de grille, par exemple, la périodicité des réponses confère au codage de la population une grande précision. Il n'est pas clair si la haute précision résulte de la structure périodique des réponses ou s'obtient plus généralement dans les neurones avec des courbes de réponse complexes.
Dans un premier projet, nous abordons cette question avec l'utilisation d'un modèle théorique: un réseau neuronal dans lequel des courbes de réponse complexes et irrégulières émergent dans les neurones de la deuxième couche en raison de poids synaptiques aléatoires. L'irrégularité améliore la résolution locale du code mais donne lieu à des erreurs globales catastrophiques. Lors de l'équilibrage de ces deux erreurs, le code résultant atteint une précision exponentielle en fonction de le nombre des neurones et le réseau comprime l'information d'une répresentation de haute dimension en basse dimension. En analysant les enregistrements du cortex moteur du singe, nous fournissons un exemple d'un tel code 'compressé'. Nos résultats montrent que les codes efficaces n'ont peut-être pas besoin d'une structure finement réglée, mais ils émergent de manière robuste du caractère aléatoire des réseaux de neurones.
Dans le premier chapitre, les propriétés de codage de la population sont calculées sous l'hypothèse d'un décodeur idéal. Dans le deuxième chapitre, nous nous demandons comment les critères d'optimalité d'un tel code neuronal sont affectés lorsque le circuit effectuant le dècodage n'est pas idéal. Nous considérons des décodeurs paramétrés comme des réseaux de neurones, entraînés de manière supervisée sur un jeu de données limité d'exemples de paires stimulus-réponse. En raison du bruit dans les données d'entraînement, le réseau est biaisé vers l'apprentissage de fonctions lisses et régulières. Cela donne un écart de performances par rapport au décodeur idéal, qui obtient une erreur plus faible en exploitant les irrégularités des courbes de réponse. Cet écart est réduit lorsque la complexité de l'architecture de décodage est augmentée, révelant un compromis entre entre l'éfficacité de l'encodage et la facilité du décodage.
Dans un troisième projet, nous considérons les représentations neuronales qui émergent au cours de l'apprentisssage non supervisé. Un encodeur, qui associe les stimuli aux réponses neuronales, et un décodeur, dont la tâche est de maintenir un modèle génératif interne de l'environment, sont optimisés conjointement, dans un cadre d'auto-encodeur variationnel. L'optimalité est atteinte lorsque l'encodeur est réglé de manière à maximiser une limite à l'information mutuelle entre les stimuli et les réponses neuronales, comme postulé par l'hypothèse de codage efficace, sous réserve d'une contrainte métabolique qui pénalise la différence entre l'activité neuronale induite par un stimulus et l'activité neuronale spontanée. Nous calculons des réponses neuronales optimales dans un modèle conventionnel de codage de population avec des courbes de réponse simples selon ce cadre. En faisant varier la contrainte, on obtient une famille de solutions qui donnent des modèles génératifs tout aussi satisfaisants, mais des représentations neuronales qualitativement différentes. Nos travaux illustrent comment l'interaction entre le processus d'encodage et de décodage façonne la représentation neuronale du monde extérieur.

MOTS CLÉS

Codage neural, codage efficace, reseaux de neurones, courbes de réponse, apprentissage supervisé, autoencodeur variationnel, modèle génératif

ABSTRACT

In this thesis, we investigate the principles which underlie optimal information coding in neural systems, by combining models from information theory and machine learning with experimental data analysis. Much of the theoretical work on efficient coding has focused on neurons whose mean response as a function of stimulus features–the neuron's tuning curve–can be described by a simple function. Real neurons, however, often exhibit more complex tuning curves: in grid cells, for example, the periodicity of the responses imparts the population code with high accuracy. It is unclear if the high accuracy results from the fine periodic structure of the responses or obtains more generally in neurons with complex tuning curves.
In a first project, we address this question with the use of a benchmark model: a shallow neural network in which complex and irregular tuning curves emerge in the second layer neurons due to random synaptic weights. Irregularity enhances the local resolution of the code but gives rise to catastrophic, global errors. When balancing these two errors, the resulting code achieves exponential accuracy as a function of the population size, and the network compresses information from a high-dimensional to a low-dimensional representation. By analyzing recordings from monkey motor cortex, we provide an example of such 'compressed' code. Our results show that efficient codes might not need a finely tuned design, but they emerge robustly from randomness and irregularity.
In the first chapter, the population coding properties are derived under the assumption of an 'ideal' decoder, which has access to details of the encoding process. In the second chapter we ask how optimality criteria of such a neural code are affected when the system performing the decoding operation is non-ideal. We consider decoders parametrized as neural networks, trained in a supervised setting on a dataset of pairs of stimuli and noisy responses. Due to the noise in the training set, the decoder is biased towards learning smooth and regular functions. This yields a gap in the performance as compared to the ideal decoder, which achieves a lower error by exploiting the irregularities of the tuning curves. This gap is reduced when the complexity of the decoding architecture is increased, revealing a trade-off between the ideal performance of a coding scheme and the ease of the decoding process.
In a third project, we consider the neural representations which emerge in an unsupervised learning setting. An encoder, which maps stimuli to neural responses, and a decoder, whose task is to maintain an internal generative model of the environment, are optimized jointly, in a variational autoencoder framework. Optimality is achieved when the encoder is set so as to maximize a bound to the mutual information between stimuli and neural responses, as postulated by the efficient coding hypothesis, subject to a metabolic constraint which penalizes the difference between stimulus-evoked and spontaneous neural activity. We derive optimal neural responses in a conventional model of population coding with simple tuning curves according to this framework. By varying the constraint, we obtain a family of solutions which yield equally satisfying generative models, but qualitatively different neural representations. Our work illustrate how the interaction between the encoding and the decoding process shape neural representation of the external world.

KEYWORDS

Neural coding, efficient coding, neural networks, tuning curves, supervised learning, Variational Autoencoder, generative models