Title

A Study of Different Aspects of Neural Networks: Neural Representations, Connectivity and Computation

Author

De, Anandita

Publication Date

2023

Peer reviewed|Thesis/dissertation

A Study of Different Aspects of Neural Networks: Neural Representations, Connectivity and Computation

By

ANANDITA DE

DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Physics

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

Daniel Cox, Chair

Rishidev Chaudhuri

Mark Goldman

Timothy Lewis
Committee in Charge

2023

Abstract

This dissertation is split into 3 parts. In the first part (Chapter 2) we look at shapes or manifolds on which neural activity over time lies. In a neural state space, where each axis represents a neuron, neural activity over time forms a point cloud. This point cloud often occupies a small region in the space of all possible activity patterns thus revealing structure in data. We consider point clouds from neural activities from common population codes known as "tuning curve models". In these models, the firing rate of each neuron is a function of a latent variable which might be a stimulus variable or a variable related to an internal state and a tuning curve parameter which labels each neuron. We address the question: How close are point clouds formed by such models to a linear subspace? To answer this, we define the linear dimension of the data to be the number of dimensions which captures a very high fraction of variance, for example 95% variance in this data. We show that the linear dimension grows exponentially with the number of latent variables encoded by the population. Thus the manifolds formed by the neural activities from these models are extremely non-linear. Linear dimension is not a good measure for the intrinsic dimension of the manifold on which this point cloud lies.

In the second part (Chapter 3), we model connections between distant brain regions by sparse random connections. We start by observing that such a network has a special property known as the expander property. Using this property it can be shown that information can be transmitted efficiently from a source region to a target region even if the target region has fewer neurons than the source region. We also consider if the compressed patterns in the target region can be re-coded or expanded to perform some computation. We show that the compressed patterns can be re-expanded by algorithms known as Locally Competitive Algorithms (LCA) and the re-expanded patterns can be separated by a downstream neuron into arbitrarily defined classes. We next consider whether long range reciprocal connections between two regions can be used to maintain persistent activity in both the regions. Such activity is thought to be a substrate for working memory, the ability to hold things in mind. We show that the network can indeed maintain sparse patterns of activity through simple network dynamics. We conclude that sparse random connections can be used to transmit information effectively and improve the performance of certain computations compared to dense random connections.

In the last part (Chapter 4), we built a computational rate model for the pre-cortex biological neural circuit responsible for the localisation of sound in the vertical plane. Interaction of incoming sound waves with the outer ear filters out energy from specific frequency bands in the spectrum of the incoming sound. The frequency bands with zero or reduced power in them are known as notches. The position of the notches is a function of the angle of elevation of the sound source. There is a dedicated set of neurons in the auditory pathway which are sensitive to the position of these notches and hence thought to be responsible for the localization of sound in the vertical plane. These neurons show different levels of excitation or inhibition above or below their spontaneous rates for

different combinations of frequencies and intensities of sound. We built a computational model to probe how this complex set of responses arise from the interaction between the various populations of neurons in the auditory pathway.

Acknowledgments

I would like to begin by thanking my Master's thesis advisor Prof. Shiraz Minwalla. His way of thinking about science and research made a lasting impact on my outlook towards research. I am grateful to my co-advisor Prof. Daniel Cox for providing me a stepping stone to smoothly transition into neuroscience. His guidance and encouragement over the years helped me to keep moving towards my goals. I am deeply indebted to my advisor Prof. Rishidev Chaudhuri, for giving me the opportunity to work on a diverse set of projects because of which I got to learn different areas of math and neuroscience. I have learnt a lot from him over the years, not just about science but the ways of doing research, keeping things organized, writing and giving talks. I am also grateful for the freedom he gave me to explore different things I wanted and always being an useful resource for the topics I was interested in. I have really enjoyed working with him. I would like to thank Prof. Mark Goldman for allowing me to present my work in his lab and all the discussions with him and his group members that has informed by work. I thank my labmates Jiacheng Xu, Tanner Stevenson and Kayleigh Adams for useful discussions at different times.

My friends who cheered me through various low points in this journey has made it possible to complete this journey and added color and much joy to it. Thanks to Arnab for being a mentor and a friend in the first year of gradschool. Thanks to Guga and Miranda for weekly cooking hangouts and crazy jokes, you helped me maintain some sanity during an insane time. Thanks to Sharvaree for being a friend through all these years and using all possible remote methods to spend time with me during the pandemic. Thanks to Adarsh for the weekly online lunches, all the friendly advise and checking up on me when things were really dark. Thanks to Pranav for sharing all my happiness and sadness and being a voice of reason. Thanks to Pershang for being a wonderful roommate and friend and for all the fun we had doing various things together. Thanks to Harsha for trusting me and my abilities unconditionally and being a pillar of love and support in my life. You all have been my lights, when things were very dark.

I would like to thank my mother who inculcated a love for maths and science in me at a very young age. I believe I am here because of that. I would like to thank my father and brother for cheering me on to be the best that I can be. Finally thanks to Sourav, for being beside me with love and patience through crazy ups and downs. I am grateful for the part of journey I got to share with you, I am grateful for the person I have become because of you. I would not have been on this path if it were not for you. You continue to inspire me by being your amazing self.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Neurons encode information about the external world and internal states through patterns of firing rate activity. These patterns of activity arise through the interaction between the neurons at the synapses. Different patterns of activities produced by such interactions in different groups of neurons are used by the brain for different computations. Understanding how neural activity patterns, connectivity between neurons, dynamics of the network and the computation they perform are linked is one of the central problems of neuroscience. In this thesis, I have studied the different aspects of neural networks, namely the patterns of activity in populations of neurons, sparse connectivity between groups of neurons and consequences of sparse connectivity and the link between computation and connectivity in a biological neural circuit responsible for the localization of sound in the vertical plane. This thesis is divided into three chapters which follow this, and each of the chapters are dedicated to a particular topic.

In Chapter 2 of the thesis I have described my work on geometric structure of population codes. A common hypothesis about information encoding by neurons is that populations of neuron collectively encode information about stimulus variables or internal states. Under this hypothesis, activities of individual neurons are highly correlated and the population as a whole encodes information. Therefore one needs to study the time varying pattern of activity of the population to understand how information is encoded by this population. One example of population coding is the coding of different

memories in the hippocampus. Each memory is thought to be represented as a unique pattern of activity across a population of hippocampal neurons, which can be reactivated later to retrieve the memory. One of the central questions of neuroscience is what information is encoded in the population activity patterns of neurons in different areas in the brain?

A geometric approach to ask questions and test hypotheses about population codes has emerged in the recent years. In this approach, the activities of neurons over time are viewed as trajectories in a neural state space, where each axes represents a neuron and each point represents the activity of the neurons at a given time. Any kind of neural recordings over time such as data from fMRI, calcium imaging or electrophysiology forms a point cloud in this space. These points lie in an ambient space of N dimensions where N is the number of neurons but it is often found that the point cloud can be confined to a low dimensional manifold. This low dimensional manifold can give insights into the information encoded by the population, and the way it is embedded depends on the neural computation performed by this population [1]. For example, Chaudhuri et al. [2] analyzed neural activity associated with a population of head direction cells in the anterodorsal thalamic nucleus of mice during active foraging as well as rapid eye movement (REM) sleep. They used a novel nonlinear decoding strategy to discover the ring structure around which the population activity was organized and the position along the ring parametrically encoded the animals' head direction. However, this ring was embedded in the ambient space in a highly nonlinear manner like a twisted elastic band. This approach has led to insights into the neural encoding of other brain areas as well. For example, neurons in the visual cortex respond selectively to the orientation of a rotating grating [3], responses in the olfactory cortex reveal the latent organization of the chemical odour space [4], and the dynamics of population activity in the motor cortex provide a potential basis set for outgoing muscle-like commands [5]. We found the linear dimension of manifolds formed by a common class of population codes. A modified manuscript has been submitted to PNAS and received positive reviews. We hope that it will be published soon.The background, methods and results of this work are described in Chapter 2 of

this thesis.

In Chapter 3 of the thesis, I have explored the interaction and exchange of information between distant brain regions using sparse connections. Neurons interact with each other at synapses. As already discussed, individual neurons encode information through patterns of electrical potential across it's membrane and firing action potentials. The action potential is a rapid depolarisation that travels down the axon of a neuron and triggers the release of neurotransmitter molecules in the synapse when it reaches the end. These neurotransmitters diffuse across the synapse and bind to receptors on the surface of the target neuron. The binding of neurotransmitters to receptors on the target neuron causes a change in the electrical potential across the target neuron's membrane, which can either excite or inhibit the firing of an action potential in the target neuron. Moreover, many neurons have complex dendritic branches which form multiple synapses with other neurons. These synapses allow neurons to integrate information in a highly specific manner. The precise location of synapses on the dendrites and soma of a neuron can influence the strength and directionality of the signal transmission. Thus, another key challenge in neuroscience is to understand how the neural representations or population codes discussed above arise from synaptic interactions within a population as well as signals external to the population. This question can also be recast to ask: how are synaptic connectivity and the computation a network performs linked? In this context, neural computation can be defined as the way neurons process information about external world or internal states and produces an output which directly leads to the activity patterns (neural representations) formed by a population.

Achieving this objective for a cortical network where there are thousands of neurons and millions of synapses is daunting even if we had data on neural activity of every neuron and strength of all synapses. Computational models of neural networks provide a natural test-bed for finding the relationship between the connectivity, dynamics and computation in a network. However this relationship is well understood only in a few cases [6–8]. The Hopfield model [6] is a recurrent neural network that can store binary patterns of activity as fixed points in its dynamics. Hopfield showed that the weights of the network could be

learned such that starting from a pattern which is slightly distorted from a pattern the network has learnt to store, the dynamics converge to that of the stored pattern. The dynamical update rule of the network moves it to the minima of an energy landscape, the minima corresponding to the stored pattern. In [8], the authors have shown that when the connectivity of a recurrent network has a low rank structure,i.e, it can be expressed as the sum of outer products of independent vectors, the activity of the network organizes in low dimensional subspace. The organization of the activity in a low dimensional subspace can be used to solve a number of cognitive tasks. (1) A go-no-go task where the network has to produce a go signal and output zero for a no-go signal. (2) A temporal integration task, where the network has to output one after integrating a signal and the integrated value crosses a threshold and a zero output if the integrated value is below the threshold. (3)-(4) Both these tasks in context dependent cases. These are a few examples in which the authors have linked the connectivity of a network to the computation it performs. We explored the consequences of modeling long distance connections with sparse random connections on the interacting between distant brain regions.

It has been observed that probability of connection between two neurons fall-off exponentially [9]. Motivated by this observation, we considered brain areas interacting through sparse random connections. We observed that these connections have a special property known as the expander property which is defined in Chapter 3. We explored the consequences of the connections having this property on some computations that these brain regions might perform. The background, methods and the result of this work is described in Chapter 3.

In the last chapter of this thesis, I investigated the relationship between neural activity, connectivity and computation for a particular biological network responsible for the localization of sound in the vertical plane. The interaction of sound with the outer ear alters the spectrum of sound hitting the eardrums filtering out energy from particular frequency bands. These regions of frequency in the spectrum where energy is filtered is known as notches. The position of these notches depend upon the vertical angle of the source of the sound. It has been found experimentally that there are dedicated neurons in

the auditory pathway whose responses are sensitive to the position of the notches in the spectrum. In addition to that, these neurons show complex responses to different auditory stimuli such as pure tones, broad band noise and notched noise. We built a computational model of the neurons in the auditory pathway starting from auditory nerve fibers, to dorsal cochlear nucleus (DCN), to inferior colliculus (IC) and investigated how the connectivity between the various populations of neurons led to their complex responses. The background, methods and results of this chapter are described in Chapter 4.

Thus each of the subsequent chapters are on three different topics and correspond to the different projects in my PhD. Each chapter explores different aspects of a neural network, starting from dimensionality of neural representations in Chapter 2, connectivity between distant brain regions and consequences on dynamics of this connectivity in Chapter 3 and the relationship between representation, connectivity and computation for the biological network responsible for localization of sound in the vertical plane in Chapter 4.

Chapter 2

Neural manifolds from tuning curve models are extremely nonlinear

2.1 Introduction

The brain encodes information about the external world and internal variables through population activity. For example, the direction of arm movement is controlled by a population of neurons in the primate motor cortex [10]. A weighted sum of the individual activities of the neurons in the motor cortex could accurately predict the direction of the arm movement [10]. Distributed coding of different features of objects such as shapes, contours, boundaries, colors is ubiquitous in the ventral visual pathway [11–15]. These studies show that multiple neurons respond to the same stimulus features and each neuron responds to multiple features. Thus, it would not be possible to infer information encoded in correlated neural activity from single cell recordings. Several studies of population coding have propelled the advancement of technologies which enable recording from tens off thousands of neurons simultaneously.

Such technologies has resulted in a boom in the amount of various types of neural data. New tools and methods are needed to form and test hypotheses based on such data. A powerful method to understand neural population dynamics is to represent neural activity data in a neural state space. Each neuron in a population corresponds to an axis in the

neural state space. Thus the neural state space of a population of N neurons is $\mathbb{R}^N$. Firing rates of neurons or other measures of neural activity at each moment is represented as a point in this space. Neural activity over time thus forms a point cloud in this space. Structure in this point cloud reveals information about encoding and computation [1]. This framework allows the application of geometric tools to understand neural data.

The ambient space in which this point cloud lies has dimensions equal to the number of total neurons (N) which is very high. It is often found that this point cloud lies on a low dimensional manifold with dimension $M << N$. Such low dimensional structure reveals structure in data and emphasises the importance of population coding. Single neuron trajectories may seem chaotic or irregular, but population activity shows structure and allows us to test hypotheses about population coding [5, 16]. For example, in [16], the authors found that the activity of $\sim$ 800 neurons in the prefrontal cortex of monkey during a working memory task lay on low dimensional manifold of dimension $3 - 6$, which explained 95% of variance in data. Viewing neural activity as points on a manifold has been useful in other parts of the brain as well. For example, references [17, 18] posit that the responses of the neurons in visual pathway for an object under different lighting, orientation, etc, form low dimensional manifolds in the high dimensional visual space. They hypothesize that the role of the visual cortical hierarchy is to disentangle the object manifolds to facilitate object recognition. Activity of neurons in the motor cortex during reaching activity lie on a low dimensional manifold and show oscillatory activity [5]. These are a few examples of studies that have used population activity and dimensionality reduction methods to uncover structure and computational principles from neural data.

A manifold is a space which resembles an Euclidean space locally but globally it might have a different structure. A very common example is a D dimensional sphere which can be locally mapped to $\mathbb{R}^D$ but such a mapping cannot be extended to all of the sphere. Suppose a manifold M can be expressed as a union of local patches U^α such that $M = \bigcup_\alpha U^\alpha$. Formally, smooth functions $\psi^\alpha : U^\alpha \to \mathbb{R}^D$ can be defined from local patches of a manifold U^α. In the case of neural data, low dimensional structure might

arise when the population of N neurons encode D latent variables. Therefore the firing rate of a neuron m in a population of N neurons can be expressed $r_m(t) = \boldsymbol{F}_m(\boldsymbol{x}(t))$ where $F : \mathbb{R}^D \to \mathbb{R}^N$ and $\boldsymbol{x}(t) \in \mathbb{R}^D$. $\boldsymbol{x}(t)$ is known as the latent variable since it cannot be observed directly but gives rise to the shared structure in the population activity. The population activity lies on a low dimensional manifold if the range of F can be confined to a manifold M with dimensions less than N.

Assuming that the data points lie on a low dimensional manifold M, a fundamental question asks how close this manifold is to a linear subspace. In many cases, neural data can be expressed as a linear combination of a few latent variables, which can be found using techniques like Principal Component Analysis (PCA) and Factor Analysis(FA) [16]. PCA finds a set of orthogonal linear axes to explain the variance in the data. The first axis corresponds to the direction of maximum variance in the data and the subsequent axes have decreasing amount of variance along them. It is a commonly used method for dimensionality reduction where only a set of independent dimensions to explain a large fraction of variance in the data (80-95 %) is retained. PCA assumes that the variance in the data is common or shared between all components. FA seeks to find fewer latent variables or factors on which the data depends. The data can be explained as a linear combination of these factors plus some error. FA assumes that each factor might have an unique variance due to error in addition to the shared common variance between. When this unique variance is equal for all latent variables, FA is equivalent to PCA.

2.2 Tuning curve model

In a tuning curve model each neuron is “”tuned”” to, or maximally responsive to a particular value of a latent variable. Tuning curve models are commonly used to describe responses of neurons in early sensory areas. For example, auditory nerve fibers are most responsive to a particular frequency [19]. Neurons in V1 are selective to the orientation of a rotating bar or, more generally orientation of boundaries [7, 20]. Neurons in higher cortical areas in the visual pathway like the inferior temporal (IT) cortex are tuned to

certain objects [13]. We generalise this model to a population code where each neuron is labelled by a tuning curve parameter $\boldsymbol{\alpha}$. This might be the value of the latent variable a neuron is most sensitive to, or it might represent other parameters of the tuning curve function, like the slope of a sigmoid, for neurons having a sigmoidal tuning curve. The latent variables $\boldsymbol{x}(t)$ on which the firing rates depend might be direct features of stimulus like frequency or intensity of sound [21], orientation of rotating bars [7], spatial frequency of visual stimulus [22] or more abstract things like shape, colour, [23] or behavorial context [24]. The firing rate of the population can be expressed as

$$\boldsymbol{r}(t) = F(\boldsymbol{x}(t), \boldsymbol{\alpha}) \tag{2.1}$$

where $\boldsymbol{r}$ is the N dimensional vector of the activities of the neurons, $F : \mathbb{R}^D \times \mathbb{R}^P \to \mathbb{R}^N$, expresses the firing rates as a function of the D dimensional latent variables $\boldsymbol{x}(t)$ and the P dimensional tuning curve parameter $\boldsymbol{\alpha}$. Note that P and D may not be equal.

Population with shared tuning curves In the model described above, the response of the n_{th} neuron is given by the n_{th} component of the function F, $r_n(t) = F_n(\boldsymbol{x}(t), \boldsymbol{\alpha})$. A population of neurons often shares the same shape of the tuning, $r_n(t) = f(\boldsymbol{x}(t), \boldsymbol{\alpha_n})$ with different parameters α_n, where α could be the width of the tuning curve, or the slope of a saturating curve. In other words all the functions $F_n(\boldsymbol{x}(t), \boldsymbol{\alpha})$ have the same form $f(\boldsymbol{x}(t), \boldsymbol{\alpha_n})$. Such shared tuning curves are common for populations described above such as V1 neurons, auditory neurons and even neurons tuned to numbers [25, 26].

Linear structure in population response A common structure to seek in data is linearity or how close the data lies to a linear subspace. In the presence of noise the points may not exactly lie on a linear subspace. Even in the absence of noise, the point cloud might only be approximately linear. In such cases, we try to find a subspace that can explain a large fraction of the variance in the data. In other words we seek L basis patterns $\boldsymbol{v_l}$ that can approximate the data points ie $\boldsymbol{r}(t) \approx \sum_l a_l(t)\boldsymbol{v}_l$.

More precisely we define the $L_{1-\epsilon}$ dimension of the data matrix A to be the smallest R such that a rank R approximation A_R of A satisfies $||A - A_R||_F^2 < \epsilon||A||_F^2$. $||A||_F$ is

the Frobenious norm of a matrix A which is defined as

$$||A||_F = \sqrt{\sum_i \sum_j a_{ij}^2} \tag{2.2}$$

This definition corresponds to the notion of dimensionality commonly used in neural data analysis [27], where dimension is defined by the number of Principal Components required to explain a large fraction of variance in the data, i.e. $(1-\epsilon)$ in our definition.

The low rank approximation of A_R of the data matrix A can be obtained by singular value decomposition (SVD) of the matrix A to give

$$A_R = \sum_{k=1}^{R} \sigma_k \boldsymbol{u_k} \boldsymbol{v_k}^T \tag{2.3}$$

where σ_k's are the singular values of A in descending order of magnitude and $\boldsymbol{u_k}, \boldsymbol{v_k}$ are the left singular vectors and right singular vectors respectively. In this case the remaining variance $||A - A_R||_F^2 = \sum_{k=R+1}^{N} \sigma_k^2$. In other words, the matrix A has $(1-\epsilon)$-linear-dimension $L_{1-\epsilon}$ if

$$\sum_{k=1}^{L_\epsilon} \sigma_k^2 / \sum_{k=1}^{N} \sigma_k^2 \geq 1-\epsilon \quad \text{but} \quad \sum_{k=1}^{L_\epsilon - 1} \sigma_k^2 / \sum_{k=1}^{N} \sigma_k^2 < 1-\epsilon \tag{2.4}$$

The singular values of A can also be calculated from the eigenvalues of the (non-mean-subtracted) covariance matrix AA^T, which is the matrix of covariances between neurons averaged over time, or A^TA, which is the matrix of covariances between data points averaged over neurons. The k-th eigenvalue of each of these matrices is $\lambda_k = \sigma_k^2$ (for $k \leq N$, assuming more time points than neurons).

Constructing such a low rank approximation to the data matrix (or, equivalently, fitting a linear subspace to the data point cloud) is the foundation of commonly used dimensionality reduction methods such as PCA and Factor Analysis. Moreover, a number of nonlinear dimensionality reduction techniques rely on approximating the data point cloud or manifold by a family of linear subspaces [28–30]. Such methods will be expected to perform well when the data point cloud or manifold is near-linear and poorly when

the data manifold is highly non-linear.

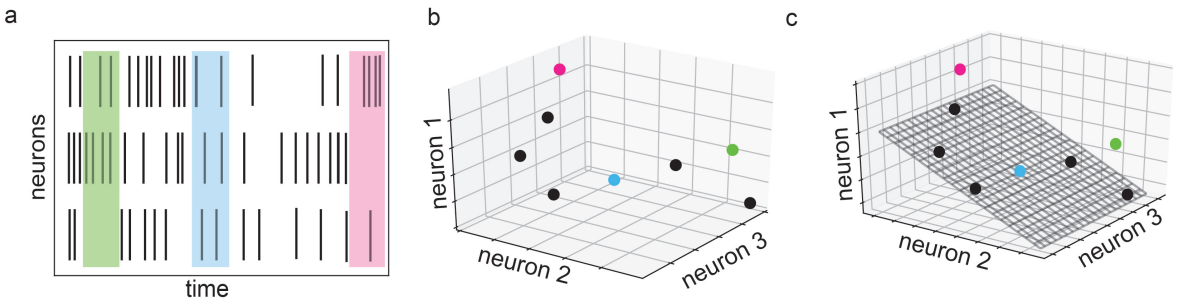

Figure 2.1: **Schematic of low-dimensional structure in neural population data.** (a) Spiking activity of 3 neurons over time. Shaded regions show three sample time bins, each used to compute an activity vector. (b) Activity represented as a collection of points in 3-dimensional space. Colored points correspond to shaded regions in panel (a). (c) Lower-dimensional linear structure in data, shown as a 2-dimensional plane chosen to capture as much variance in the data as possible. Scatter of points (e.g., pink and green points) off of plane reflects variance that is not captured.

2.3 Translation invariant tuning curves

A commonly found population code in the brain is one in which each neuron in the population share the same tuning curve centered around different values in the latent space. For example, orientation selective neurons in the V1 have the same shape of tuning curves centered around angles of orientation. Place cells and grid cells also share the same tuning curves centered around different points in physical space. Neurons in the parietal cortex show translation invariant tuning to numbers on a logarithmic scale [25, 26]. Similarly, auditory nerve fibers are tuned to frequencies on a logarithmic scale and have the same structure. In such a code, the tuning curves of the neurons are shifted versions of each other.

$$f(\boldsymbol{x} + \boldsymbol{\delta}, \boldsymbol{\alpha} + \boldsymbol{\delta}) = f(\boldsymbol{x}, \boldsymbol{\alpha}) = f(\boldsymbol{x} - \boldsymbol{\alpha}, 0) \tag{2.5}$$

This implies that the tuning curves have to be a function of the difference between the latent variable $\boldsymbol{x}$ and the tuning curve parameter $\boldsymbol{\alpha}$.

$$f(\boldsymbol{x}, \boldsymbol{\alpha}) = g(\boldsymbol{x} - \boldsymbol{\alpha}) \tag{2.6}$$

For tuning curves to be truly translation invariant, the latent space has to be periodic or infinite. In this work, we assume that $\boldsymbol{x}, \boldsymbol{\alpha} \in [0,1]^D$ and f is periodic along each dimension with period 1. In the case where the latent variables are uniformly sampled from this space, the neuron-neuron covariance matrix $C(\boldsymbol{\alpha_m}, \boldsymbol{\alpha_n}) = c(\boldsymbol{\alpha_m} - \boldsymbol{\alpha_n})$, where c is a periodic function with period 1 along each dimension, which has been derived in Section 2.6.2. Thus translation invariant tuning curves give rise to translation invariant covariance matrices. It is a well known result that the eigenvalues of a translation invariant kernel is given by its D-dimensional Fourier transform. This can be shown to be true for the discrete translation invariant covariance matrix too (Section 2.6.3).

In 1D, if the tuning curve centers are uniformly spaced in the interval, [0,1) the covariance matrix is a circulant matrix where each row is obtained by cyclically shifting the row above to the right. Eigenvalues of circulant matrices are given by the discrete Fourier transform of the first row of a circulant matrix. When the boundary conditions are not periodic, the covariance matrix is Toeplitz, and the eigenvalue relation holds approximately [31].

2.3.1 1D Gaussian tuning curves

Population responses can often be fit by a Gaussian tuning curves [32, 33]. We first consider a population with 1D translation invariant Gaussian tuning curves. We assume that the latent variables x and neuron labels α lie on a circle with circumference, $x, \alpha \in [0, 1)$. Consider, for example orientation selective neurons in the V1, visualised in Fig. 2a. The data matrix for this model is given by

$$A(x_p, \alpha_q) = K_1 \exp\left(\frac{-(x_p - \alpha_q)^2}{2\sigma^2}\right) \tag{2.7}$$

where K_1 is the maximal firing rate when the latent variable x is equal to the neuron label α and σ is the width of the tuning curves. In Fig 2a. the x_p's are points along the circle. The distance $|x_p - \alpha_q|$ is the shortest distance between x_p and α_q along the circle.

The covariance matrix can be obtained by averaging over the latent variables. We

assume that the latent variables are sampled uniformly from $[0, 1)$. The covariance is given by a Gaussian with width $\sqrt{2}\sigma$

$$\begin{aligned} c(\alpha_m - \alpha_n) = c(\delta) =& K_2 \exp\left(-\frac{\delta^2}{4\sigma^2}\right), \quad \delta \leq \frac{1}{2} \\ =& K_2 \exp\left(-\frac{(1-\delta)^2}{4\sigma^2}\right), \quad \delta > \frac{1}{2} \end{aligned} \tag{2.8}$$

which is periodic generalisation of a Gaussian in the range [0,1). The constant $K_2 = K_1^2\sqrt{\pi\sigma^2}$ If tuning curve centers are evenly spaced, this covariance profile is sampled at $\delta_n = n/N$, where n takes values in $[0, \ldots, N-1]$.

As described in Section 2.3, the eigenvalues of the covariance matrix are given by the Fourier transform of this profile. For convenience we index the eigenvalues by p ranging from $\lfloor(-N+1)/2\rfloor$ to $\lfloor N/2\rfloor$. For each such p we have an eigenvalue

$$\begin{aligned} \lambda_p &= K_2 \sum_{n=0}^{N-1} c(n/N) e^{-2\pi i p\, n/N} = K_2 \left(\sum_{n=0}^{\lfloor N/2\rfloor} c(n/N) e^{-2\pi i p\, n/N} + \sum_{l=1}^{\lfloor (N+1)/2\rfloor} c(1-l/N) e^{2\pi i p\, l/N} \right) \\ &= K_2 \left(\sum_{n=0}^{\lfloor N/2\rfloor} \exp\left(\frac{-n^2}{4\sigma^2 N^2}\right) \exp\left(\frac{-2\pi i n p}{N}\right) + \sum_{l=\lfloor(-N+1)/2}^{-1} \exp\left(\frac{-l^2}{4\sigma^2 N^2}\right) \exp\left(\frac{2\pi i l p}{N}\right) \right) \\ &= K_2 \left(\sum_{n=\lfloor(-N+1)/2\rfloor}^{\lfloor N/2\rfloor} \exp\left(\frac{-n^2}{4\sigma^2 N^2}\right) \exp\left(\frac{-2\pi i n p}{N}\right) \right) \\ &= K_3 \exp\left(-4\pi^2\sigma^2 p^2\right) \end{aligned} \tag{2.9}$$

where in the last line $K_3 = K_2 N$ and we have completed the square and noted that the sum over n is a constant. Thus the eigenvalues have a Gaussian profile with width $\frac{2\sqrt{2}\pi}{\sigma}$. Also note that the eigenvalues decrease monotonically with the magnitude of p and that, except for λ_0, they occur in pairs with $\lambda_p = \lambda_{-p}$.

Since the eigenvalues occur in pairs, the $(1-\epsilon)$-linear dimension is the smallest $L_{1-\epsilon}$ such that

$$\sum_{p=0}^{\frac{L_{1-\epsilon}}{2}} \lambda_p \geq (1-\epsilon) \sum_{p=0}^{N/2} \lambda_p. \tag{2.10}$$

For large N, approximating both sides of this equation as an integral and canceling

the common prefactor K_3 yields

$$\begin{aligned}
\int_0^{\frac{L_{1-\epsilon}}{2}} e^{-(2\pi\sigma p)^2} dp &\geq (1-\epsilon) \int_0^{\infty} e^{-(2\pi\sigma p)^2} dp = \frac{(1-\epsilon)}{4\sqrt{\pi}\sigma} \\
4\sqrt{\pi}\sigma \int_0^{\frac{L_{1-\epsilon}}{2}} e^{-(2\pi\sigma p)^2} dp &\geq (1-\epsilon) \\
\frac{2}{\sqrt{\pi}} \int_0^{\pi\sigma L_{1-\epsilon}} e^{-z^2} dz &\geq (1-\epsilon) \\
\mathrm{erf}\,(\pi\sigma L_{1-\epsilon}) &\geq (1-\epsilon) \\
L_{1-\epsilon} &= \frac{1}{\sigma} \frac{\mathrm{erf}^{-1}\,(1-\epsilon)}{\pi}
\end{aligned} \tag{2.11}$$

Thus, $L_{1-\epsilon}$ grows as $\frac{1}{\sigma}$, with a proportionality constant that depends on the fraction of variance explained. In particular, for 95% variance explained, we have $L_{0.95} = \frac{1.96}{\sqrt{2}\,\pi\sigma}$.

As we have discussed, translation invariant covariance profiles and their eigenvalues are related by Fourier transforms. According to these uncertainty principles, a function and its Fourier transform cannot both be too localised in their domains [34]. More formally

$$H(f)H(\hat{f}) \geq A \tag{2.12}$$

where $\hat{f}$ denotes the Fourier transform of f and $H(f)$ is some measure of localisation of f and A is a constant. For example, for the Heisenberg uncertainty principle the localisation is the variance of the square of the function, ie $H(f) = \int dx\, x^2 |f(x)|^2$. The Heisenberg uncertainty principle is saturated by a Gaussian function.

Other measures of f include support of f, which is equal to the number of non-zero elements of $f(x)$ when it is discretely sampled from a range. In this case the uncertainty principle is given by

$$Supp|f(x)|\; Supp|\hat{f}(\xi)| \geq N \tag{2.13}$$

where N is the total number of points at which the function is sampled. This relation can be used to show that if a fraction $1-\hat{\epsilon}$ of the covariance profile is concentrated on a set S of size P (meaning that $\sum_{\delta\in S} |c(\delta)| > (1-\hat{\epsilon}) \sum |c(\delta)|$), then the smallest set that contains $1-\epsilon$ of the eigenvalue mass has size at least $N(1-\hat{\epsilon})(1-\epsilon)/P$ [34, 35]. Note

that the size of this set is just the $(1-\epsilon)$-linear dimension and consequently the linear dimension grows inversely with P. These uncertainty principles imply that the spread or the fall-off of the eigenvalue profile is inversely related to the spread or fall-off the covariance profile. Thus covariance profiles concentrated on small sets would have a large number of non-zero eigenvalues which will lead to high linear dimension.

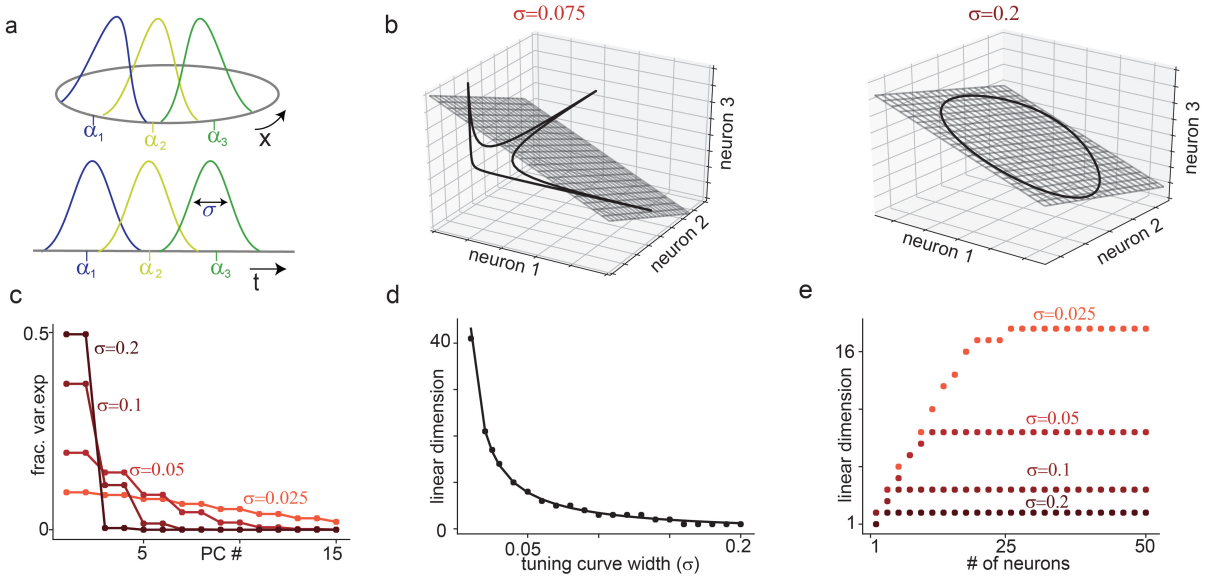

Figure 2.2: **Translation-symmetric tuning to a one-dimensional variable and the inverse relationship between linear dimension and sparsity.** (a) Gaussian tuning curves of 3 neurons encoding a circular (top) or non-circular (bottom) scalar stimulus variable. The non-circular variable example includes tuning to time, as in an epoch code. (b) Black line: Manifold formed by population activity of 3 neurons with Gaussian tuning to a 1-dimensional circular variable. Each axis shows the activity of 1 neuron. Gray: Best fitting 2D linear subspace (i.e., plane spanned by first two principal components). Left and right show an example of narrow ($\sigma = 0.075$) and broad ($\sigma = 0.2$) tuning respectively. For (c)–(e), results shown are for Gaussian tuning to a circular variable, with uniformly spaced tuning curve centers. Circles show numerical simulations and lines show theoretical predictions. (c) Fraction of variance explained by each principal component (equivalently, eigenvalues of covariance matrix) for a population of $N = 50$ neurons. Different curves show different tuning curve widths. (d) Linear dimension of neural data against tuning curve widths, showing that linear dimension grows as $1/\sigma$. (e) Linear dimension against number of neurons in a population for each tuning curve width, showing initial linear growth before saturation at the predicted values shown in (d).

2.3.2 D dimensional translation invariant Gaussian tuning curves

We now turn to the case where the neurons encode D features, or the latent space is D dimensional. For place cells and grid cells, $D = 2$ and correspond to two spatial

dimensions. We consider D dimensional translation invariant Gaussian tuning curves with periodic boundary conditions along each dimension. We assume that $\boldsymbol{x}, \boldsymbol{\alpha} \in [0,1]^D$. The data matrix is given by

$$A(\boldsymbol{x}_p, \boldsymbol{\alpha}_q) = K_1 \exp\left(-\frac{1}{2}(\boldsymbol{x_p} - \boldsymbol{\alpha_q})^T \Sigma^{-1} (\boldsymbol{x_p} - \boldsymbol{\alpha_q})\right) \tag{2.14}$$

The basis vectors of the latent space $\boldsymbol{x}$ can be chosen such that the covariance $\Sigma = \text{diag}(\sigma^2, \cdots, \sigma^2)$ is diagonal.

Since the tuning curves are translation invariant the covariance profile is also translation invariant, $c(\boldsymbol{\alpha_m} - \boldsymbol{\alpha_n}) = c(\boldsymbol{\delta})$ as shown in (Eq. (2.39). For notational convenience we shift the range of $\boldsymbol{\delta}$ so that each component lies within $[-1/2, 1/2]$ rather than $[0,1]$. Thus,

$$\begin{aligned}
c(\boldsymbol{\delta}) &= \int_{\mathcal{S}} d\boldsymbol{x}\, f(\boldsymbol{x}, 0) f(\boldsymbol{x}, \boldsymbol{\delta}) \\
&= K_1^2 \int_{\mathcal{S}} \boldsymbol{dx} \exp\left(-\frac{1}{2}\boldsymbol{x}^T \Sigma^{-1} \boldsymbol{x}\right) \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\delta})^T \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\delta})\right) \\
&= K_1^2 \int_{\mathcal{S}} \boldsymbol{dx} \exp\left[-\left(\boldsymbol{x}^T \Sigma^{-1} \boldsymbol{x} - \frac{\boldsymbol{\delta}^T \Sigma^{-1} \boldsymbol{x}}{2} - \frac{\boldsymbol{x}^T \Sigma^{-1} \boldsymbol{\delta}}{2} + \frac{\boldsymbol{\delta}^T \Sigma^{-1} \boldsymbol{\delta}}{2}\right)\right] \\
&= K_1^2 \int_{\mathcal{S}} \boldsymbol{dx} \exp\left[-\left(\boldsymbol{x} - \frac{\boldsymbol{\delta}}{2}\right)^T \Sigma^{-1} \left(\boldsymbol{x} - \frac{\boldsymbol{\delta}}{2}\right)\right] \exp\left(-\frac{\boldsymbol{\delta^T} \Sigma^{-1} \boldsymbol{\delta}}{4}\right) \\
&= K_1^2 \exp\left(-\frac{\boldsymbol{\delta^T} \Sigma^{-1} \boldsymbol{\delta}}{4}\right) \int_{\mathcal{S}} \boldsymbol{dx} \exp\left[-\left(\boldsymbol{x} - \frac{\boldsymbol{\delta}}{2}\right)^T \Sigma^{-1} \left(\boldsymbol{x} - \frac{\boldsymbol{\delta}}{2}\right)\right] \\
&= K_2 \exp\left(-\frac{\boldsymbol{\delta^T} \Sigma^{-1} \boldsymbol{\delta}}{4}\right).
\end{aligned} \tag{2.15}$$

Here the constant $K_2 = K_1^2 \sqrt{\det(\pi\Sigma)}$. The last equality is approximate but holds when tuning curves are not too wide, meaning that no single neuron responds to the full range of latent variable values, so that integrating over the range of latent variable values is equivalent to integrating over the range of the tuning curve. Thus, neurons with Gaussian tuning curves have a Gaussian covariance profile with covariance twice that of the tuning curve.

The eigenvalues of the covariance matrix are given by a D dimensional Fourier transform. Each eigenvalue can be indexed by a D dimensional Fourier vector $\boldsymbol{p}$, with dth

entry $p_d \in [-N_d/2, \cdots, N_d/2]$ yielding

$$\lambda_{\boldsymbol{p}} = K_3 \exp\left(-4\pi^2 \sum_{d=1}^{D} p_d^2 \sigma^2\right) = K_3 \exp(-4\pi^2\sigma^2 ||\boldsymbol{p}||^2). \tag{2.16}$$

where N_d is the number of neurons along each dimension.

The magnitude of an eigenvalue is thus a function only of $||\boldsymbol{p}||$, the Euclidean distance from the origin in $\boldsymbol{p}$-space. Eigenvalues corresponding to all the lattice point at a fixed distance from the origin will have the same magnitude of eigenvalue. Since the number of lattice points on a sphere increases with the radius of the sphere in the $\boldsymbol{p}$ space, the number of eigenvalues at a given distance from the origin increases as the distance increases. Consequently, as the distance from the origin increases, there will be more eigenvalues but with smaller magnitude. This explains the fall-off of the eigenvalues of the eigenvalues as shown in Fig. 3b,c.

In the continuous limit, we define λ to be a D-dimensional continuous Gaussian function, $\lambda : \mathbb{R}^D \to \mathbb{R}$ as $\lambda(\boldsymbol{p}) = K_3 \exp(-4\pi^2\sigma^2||\boldsymbol{p}||^2)$. The eigenvalues are given by $\lambda(\boldsymbol{p})$ sampled at the integer lattice points of a D-dimensional cube with side length N_d.

Since the magnitude of the eigenvalues fall off with distance from the origin in the $\boldsymbol{p}$ space, summing up the $L_{1-\epsilon}$ largest eigenvalues is equivalent to summing up all eigenvalues for which $||\boldsymbol{p}|| \leq R$ (i.e., eigenvalues corresponding to all lattice points within a ball of radius R). $L_{1-\epsilon}$ will be equal to the number of lattice points in the sphere of radius R. Thus we first compute the smallest radius R such that

$$\sum_{||\boldsymbol{p}|| \leq R} \lambda_{\boldsymbol{p}} \geq (1-\epsilon) \sum_{p} \lambda_{\boldsymbol{p}}. \tag{2.17}$$

We approximate the sum by the integral of $\lambda(\boldsymbol{p})$. We also assume that N_d is large enough so that the sum $\sum_p \lambda_{\boldsymbol{p}}$ can be approximated by $\int_{||\boldsymbol{p}||} d\boldsymbol{p}\lambda(\boldsymbol{p})$ where the integral is over the whole space which we denote by K_4. We seek R such that

$$\frac{1}{K_4} \int_{||\boldsymbol{p}|| \leq R} dp_1 dp_2 \cdots dp_D \exp(-(2\pi\sigma)^2 \sum_{d=1}^{D} p_d^2) \geq (1-\epsilon). \tag{2.18}$$

Converting to radial coordinates with $r^2 = \sum_{d=1}^{D} p_d^2$ and then rescaling r to define $y = \sqrt{8}\pi\sigma r$ yields

$$\begin{aligned}\frac{1}{K_4}\int_{||\boldsymbol{p}||\leq R} dp_1 \dots dp_D \exp(-(2\pi\sigma)^2 \sum_{d=1}^{D} p_d^2) &= \frac{1}{K_5}\int_0^R dr\, r^{D-1}\exp(-(2\pi\sigma)^2 r^2) \\ &= \frac{1}{K_6}\int_0^{\sqrt{8}\pi\sigma R} dy\, y^{D-1}\exp(-y^2/2),\end{aligned} \tag{2.19}$$

where $K_4 = \left(\frac{\pi}{(2\pi\sigma)^2}\right)^{D/2}$, $K_5 = \frac{\Gamma(D/2)}{2(4\pi^2\sigma^2)^{D/2}}$ and $K_6 = 2^{(D/2)-1}\Gamma(D/2)$.

We note that λ is the (unnormalized) probability density function of a multivariate Gaussian distribution and integrating it over a sphere of radius R normalized by the total probability mass, gives us the CDF of a chi distribution which is a distribution of magnitudes of D dimensional Gaussian random vectors. Thus for a random variable Z from the chi distribution the integral in Eq. (2.19) represents the probability $P(Z \leq \sqrt{8}\pi\sigma R)$. We seek to find R such that $P(Z \leq \sqrt{8}\pi\sigma R) > 1 - \epsilon$.

If $\epsilon < 1/2$ (i.e., we are capturing at least 50% of the variance in the data), then by definition R must be at least the median value of Z. Standard results on chi distributions then show that $\sqrt{8}\pi\sigma R \geq \sqrt{D}\left(1 - \frac{2}{9D}\right)^{3/2} > 0.8\sqrt{D}$, where the final inequality holds for $D \geq 2$. Consequently, $R > \frac{0.8}{\sqrt{8}\pi}\frac{\sqrt{D}}{\sigma}$.

Finally, to estimate the linear dimension itself we need to count the number of eigenvalues that lie within a sphere of radius R. Since the eigenvalues correspond to lattice points, the number of eigenvalues is approximately the volume of this D-dimensional sphere yielding

$$L_{1-\epsilon} \approx V_D = \frac{\pi^{D/2}}{\Gamma\left(\frac{D}{2}+1\right)}R^D \approx \frac{1}{\sqrt{D\pi}}\left(\frac{0.4\sqrt{e}}{\sigma\sqrt{\pi}}\right)^D \approx \frac{1}{\sqrt{D\pi}}\left(\frac{K_7}{\sigma}\right)^D \tag{2.20}$$

where K_7 is a constant.To get this final form we have Stirling's approximation for $\Gamma(D/2+1) \sim \sqrt{D\pi}\left(\frac{D}{2e}\right)^{D/2}$ and substituted $R = \frac{0.8}{\sqrt{8}\pi}\frac{\sqrt{D}}{\sigma}$. Consequently the linear dimension grows exponentially with D.

Exponential (or faster) scaling for localized tuning curves can be more generally derived from uncertainty principles. Analogous to the 1D setting, the results of [34] can

be used to show that if a fraction $1-\hat{\epsilon}$ of the covariance profile is concentrated on a set S of size K, then the smallest set that contains $1-\epsilon$ of the eigenvalue mass (i.e., the $(1-\epsilon)$-linear dimension) has size at least $N_D^D(1-\hat{\epsilon})(1-\epsilon)/K$ (where as before N_D is the number of tuning curve centers per dimension). For the case of Gaussian tuning, the size of the set containing 50% of the covariance profile can be bounded above by the number of points in a sphere of radius $\sigma\sqrt{D}$, and this when combined with the uncertainty principle again yields exponential scaling.

2.4 Multiplicative tuning curves

We next consider tuning curves which can be expressed as a product of tuning along each dimension or factors of lower dimensions. For simplicity we assume that the D dimensional tuning curve is a product of D tuning curves,

$$y_n(t) = f(\boldsymbol{x}(t), \boldsymbol{\alpha}_n) = \prod_{d=1}^{D} f_d(x^d(t), \alpha_n^d). \tag{2.21}$$

(Here the superscript denotes the d-th component of a vector, and the f_d's are scalar functions.) Note that in general the multiplicative factors can consist of groups of variables and not just one-dimensional factors, in which case the product would be taken over groups rather than single variables.

A common example of such tuning curves come from multiplicative gain modulation models of attention, where the tuning curve can be written as a product of the tuning to the stimulus and the attentional signal [36] (Fig. 2.4a (top)). Another common example is separable spatio-temporal receptive fields in early visual cortex. Early visual cells respond to specific visual stimuli and their responses change over time. The responses of these visual cells can be expressed as the stimuli convolved with a filter (the receptive field). This filter or receptive field is known as separable when it factorizes into a product of the spatial and the temporal part Fig. 2.4a (bottom)) [37, 38]. In general this model is like a "mean field approximation" for neurons which code multiple features or latent variables and their total tuning can be expressed as a product of tuning for each latent

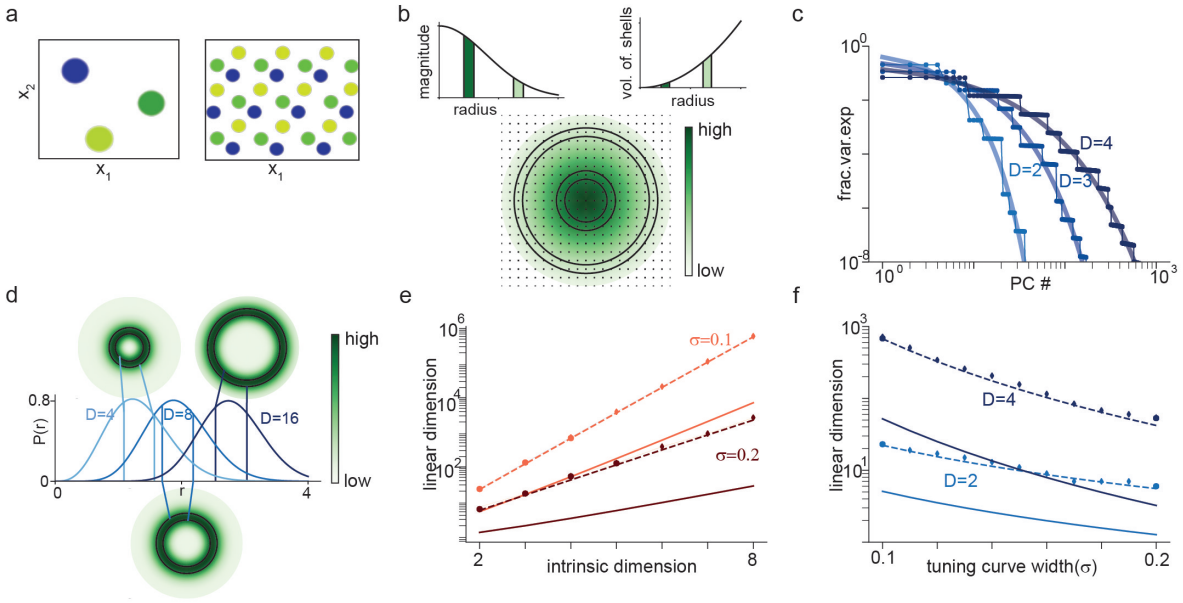

Figure 2.3: **Translation-invariant tuning to a multi-dimensional variable and exponential growth of linear dimension with intrinsic dimension.** (a) Examples of 2d tuning curves, showing schematics of 3 different place cells with different tuning centers in a square arena (left) and 3 grid cells with the same spacing but different phases (right). (b) For Gaussian tuning curves, eigenvalues of the covariance matrix (variance along each PC) are values of a D-dimensional Gaussian at the lattice points of D-dimensional Fourier space. Each lattice point corresponds to one eigenvalue, and the colormap shows its value. Left inset: Decay of eigenvalues with distance from origin in Fourier space. Right: Number of eigenvalues contained in concentric shells of different radii. Circular shells on plot highlight two sets of eigenvalues, with corresponding magnitude and volume of shell shown as shaded region in insets. For a shell close to the origin the eigenvalues have large magnitude but there are fewer eigenvalues as a consequence of the smaller volume. Away from the origin the value of the eigenvalue is lower but there are more such eigenvalues. This trade-off between eigenvalue magnitude and the number of eigenvalues of that magnitude explains the shape of the variance explained vs PC number curve. (c) Fraction of variance explained by each PC (or eigenvalues of covariance matrix) for D-dimensional Gaussian tuning curves and periodic boundary conditions along each dimension. Circles show numerical simulations, thin line represents prediction from Fourier transform of covariance matrix rows, and thicker lines represent theoretically-predicted smooth interpolation. (d) Total probability mass at radius r for a D-dimensional Gaussian (i.e., density function of chi distribution), shown for three different values of D. Circular insets show concentric shells colored by total probability mass at that radius. The bulk of the probability mass lies in a shell of radius $\sim \sqrt{D}/\sigma$. Thus, accounting for most of the variance requires considering all eigenvalues within a sphere of radius at least $\sim \sqrt{D}/\sigma$. (e) Semi-log plot of linear dimension ($\epsilon = 0.05$) vs. intrinsic dimension for Gaussian tuning curves with different widths. Circles show numerical results, solid lines show theoretical lower bound as in Eq. (2.20)(applies whenever $\epsilon \leq 0.5$), and dashed lines show semi-analytic fit using chi distribution. (f) Semi-log plot of linear dimension vs. tuning curve width. Circles and lines as in panel (e).

variable.

We assume that the latent variables $\boldsymbol{x}(t)$ are sampled independently along each dimension. In this case the covariance of two neurons with parameters $\boldsymbol{\alpha_m}$ and $\boldsymbol{\alpha_n}$ is given by

$$\begin{aligned} c(\boldsymbol{\alpha_m}, \boldsymbol{\alpha_n}) &= \int d\boldsymbol{x}\, p(\boldsymbol{x}) f(\boldsymbol{x}, \boldsymbol{\alpha_m}) f(\boldsymbol{x}, \boldsymbol{\alpha_n}) \\ &= \prod \int dx^d\, p(x^d) f_d(x^d, \alpha_m^d) f_d(x^d, \alpha_n^d) \\ &= \prod c^d(\alpha_m^d, \alpha_n^d), \end{aligned} \tag{2.22}$$

where $p(\boldsymbol{x}) = \prod_d p(x^d)$ is the distribution of latent variable values and we have defined $c^d(\alpha_m^d, \alpha_n^d) = \int dx^d p(x^d) f_d(x^d, \alpha_m^d) f_d(x^d, \alpha_n^d)$. Thus the data covariance matrix can be written as the tensor product of matrices corresponding to individual stimulus dimensions, $C = \otimes_{d=1}^{D} C^d$, where $\otimes$ indicates the tensor product and the matrix C^d has entries $C^d_{mn} = C^d(\alpha_m^d, \alpha_n^d)$.

The eigenvalues of C will then be given by the tensor product of the eigenvalues of C^d. Let $\{\gamma^d_{p_d}, \boldsymbol{u^d_{p_d}}\}$ be the p_dth eigenvalue-eigenvector pair for each C^d. Then

$$\begin{aligned} C\left(\boldsymbol{u^1_{p_1}} \otimes \cdots \otimes \boldsymbol{u^D_{p_D}}\right) &= \left(C^1 \otimes \cdots \otimes C^D\right)\left(\boldsymbol{u^1_{p_1}} \otimes \cdots \otimes \boldsymbol{u^D_{p_D}}\right) \\ &= \left(C^1 \boldsymbol{u^1_{p_1}}\right) \otimes \cdots \otimes \left(C^D \boldsymbol{u^D_{p_D}}\right) \\ &= \left(\gamma^1_{p_1} \boldsymbol{u^1_{p_1}}\right) \otimes \cdots \otimes \left(\gamma^D_{p_D} \boldsymbol{u^D_{p_D}}\right) \\ &= \left(\gamma^1_{p_1} \times \cdots \times \gamma^D_{p_D}\right)\left(\boldsymbol{u^1_{p_1}} \otimes \cdots \otimes \boldsymbol{u^D_{p_D}}\right) \end{aligned} \tag{2.23}$$

Consequently, $\left(\gamma^1_{p_1} \times \cdots \times \gamma^D_{p_D}\right)$ is an eigenvalue of C. Thus the eigenvalues of C are given by all possible products of the eigenvalues of the individual factors.

2.4.1 The linear dimension of multiplicative models grows exponentially with intrinsic dimension

For simplicity we assume that the covariance matrix factorizes along each dimension. Let the eigenvalues of the covariance matrix for each factor be $\{\gamma_1, \ldots, \gamma_{N_D}\}$. The N

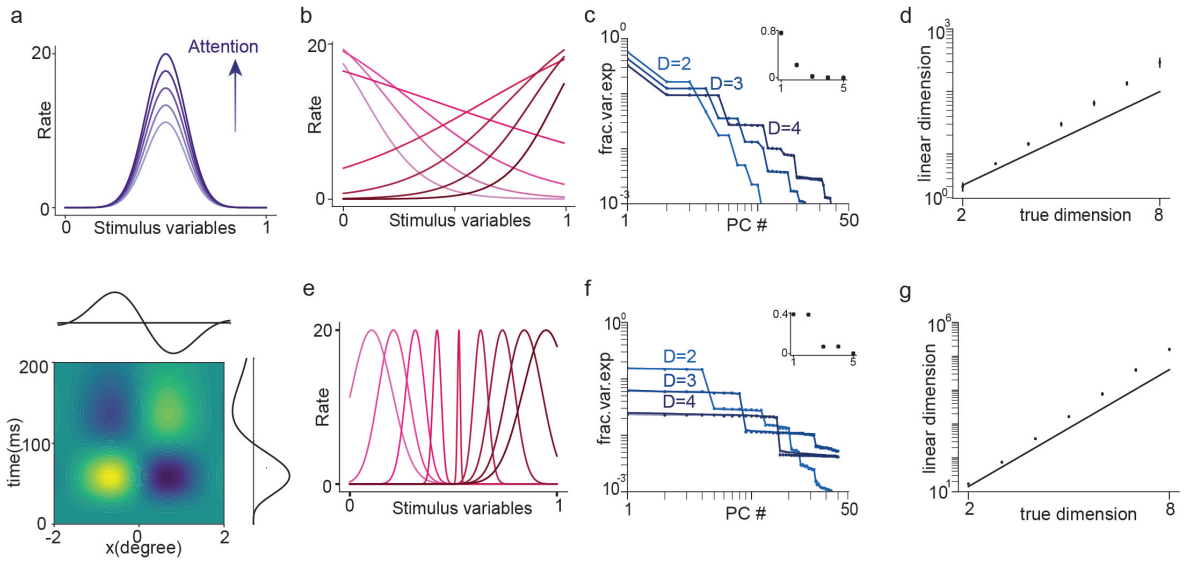

Figure 2.4: **Multiplicative tuning and exponential growth of linear dimension with intrinsic dimension.** (a) Schematics of common examples of multiplicative tuning. Top: Gain modulation of tuning to a sensory stimulus by attention. Bottom: Separable spatio-temporal receptive field of retinal ganglion cell as product of spatial tuning (horizontal) and temporal tuning (vertical). Panels (b)-(d) show results from a multiplicative tuning model where tuning along each dimension is sigmoidal. (b) Sample tuning along each dimension. Tuning curves are sigmoidal with slopes chosen uniformly in range $[-5, 5]$. (c) Fraction of variance explained vs PC number for the model shown in (b) for different values of intrinsic dimension (D). Circles show numerical simulations and lines show result from tensor product of 1D tuning curves. Inset shows the eigenvalues in the 1D case. (d) Linear dimension against intrinsic dimension for the data in (c). Circles show simulations and solid line shows theoretical lower bound of $2^{D(H-0.05)}$, where H is the entropy of the eigenvalue distribution shown in the inset of panel (c). Panels (e)-(g) show results from a multiplicative tuning model where tuning along each dimension is Gaussian. Gaussians are not translation-invariant and the width of the Gaussian depends on position, with tuning sharpest at the center of the stimulus space (as in visual receptive fields). (e) Sample tuning along each dimension. (f), (g) As in (c), (d) but for the model shown in (e).

eigenvalues of the overall covariance matrix C are given by all products of the form $\prod_d \gamma_{p_d}$, where each $p_d \in [1, \ldots, N_D]$. To compute a lower bound on the linear dimension we reframe the problem as a problem in probability theory.

We first divide the set $\{\gamma_1, \ldots, \gamma_{N_D}\}$ by its sum, so that each element denotes the fraction of variance explained along a PC. We consider D independent multinomial random variables, $Z_1 \cdots Z_D$, each having outcome probabilities $\{\gamma_1, \ldots, \gamma_{N_D}\}$. Therefore each eigenvalue of the covariance matrix C corresponds to the probability of a particular realisation of the string $Z_1 \cdots Z_D$. Thus finding the smallest number of eigenvalues whose sum is at at least $1 - \epsilon$ is equivalent to the size of the smallest set of such strings which has a probability at least equal to $1 - \epsilon$.

This subset is often referred to as an ϵ high-probability set [39]. Probabilistic arguments in information theory shows that the size of this set approaches 2^{DH_γ}, where $H_\gamma = -\sum_p \gamma_p \log_2 \gamma_p$ is the Shannon entropy of the distribution $\{\gamma_1, \ldots, \gamma_{N_d}\}$. Since the size of this set corresponds to the $L_{1-\epsilon}$ dimension of our original problem, the linear dimension grows exponentially as 2^{DH_γ} with D. In general, $2^{D(H_\gamma - \delta)}$ provides a lower bound for the size of this set, where $\delta < 1$. We show this lower bound as a solid line in Fig. 4d, h as solid lines with $\delta = 0.05$. This is an arbitrary choice, however and small δ works.

While this is an asymptotic argument, exponential scaling holds for small D as well. For a non-asymptotic lower bound on the linear dimension of these models, consider the case where only two of the eigenvalues of the matrix C^d are nonzero. By normalizing to sum to 1, we can write these eigenvalues as $1 - \gamma$ and γ, for some $\gamma \leq 0.5$. As for the multinomial case, the eigenvalues of C are the tensor product of $[1 - \gamma, \gamma]$ taken D times, i.e. $[1 - \gamma, \gamma] \underbrace{\otimes \cdots \otimes}_{D \text{ times}} [1 - \gamma, \gamma]$. In descending order of magnitude, there is 1 eigenvalue of magnitude $(1 - \gamma)^D$, $\binom{D}{1}$ eigenvalues of magnitude $(1 - \gamma)^{D-1}\gamma$, and so on, with $\binom{D}{k}$ eigenvalues of magnitude $(1 - \gamma)^{D-k}\gamma^k$.

To bound the linear dimension below, note that if F is such that

$$\sum_{k=0}^{F} \binom{D}{k} (1 - \gamma)^{D-k} \gamma^k \leq (1 - \epsilon), \tag{2.24}$$

then the linear dimension $L \geq \sum_{k=0}^{F} \binom{D}{k}$. Thus, we will first find such a F and then use it to estimate a lower bound for the linear dimension L.

Consider a random variable Z distributed according to the binomial distribution with probability γ. That is, $X \sim Bin(D, \gamma)$. Note that $\sum_{k=0}^{F} \binom{D}{k}(1-\gamma)^{D-k}\gamma^k = P(X \leq F)$. The median of this binomial distribution is at least $\lfloor \gamma D \rfloor$. The function $\lfloor . \rfloor$ returns the greatest integer smaller than its argument. Thus if we choose $F = \lfloor \gamma D \rfloor - 1$ we have

$$\sum_{k=0}^{F} \binom{D}{k}(1-\gamma)^{D-k}\gamma^k < 0.5 < 0.95. \tag{2.25}$$

For simplicity, define γ' such that $\gamma' D = \lfloor \gamma D \rfloor$ and note that we can rewrite $F = \rho D$, where $\rho > \gamma - 2/D$ (or, using results bounding the distance of the mean to the median of a binomial, this can be improved to $\rho > \gamma - (1 + \ln(2))/D$). We can now lower bound L as $L \geq \sum_{k=0}^{\rho D} \binom{D}{k}$.

A standard bound on the sum of binomial coefficients [39] yields

$$\sum_{k=0}^{\rho D} \binom{D}{k} \geq \frac{1}{\sqrt{8D\rho(1-\rho)}} 2^{H(\rho)D} = \frac{1}{\sqrt{8\rho(1-\rho)}} 2^{(H_b(\rho) - \frac{log_2(D)}{2D})D} \tag{2.26}$$

where $H_b(\rho) = -\rho \log_2 \rho - (1-\rho)\log_2(1-\rho)$ is the binary entropy function.

Thus, except when D is small enough that $H(\rho) < \frac{\log_2(D)}{2D}$ the lower bound grows exponentially with D, with the exponent asymptotically approaching $H_b(\gamma)D$.

2.5 Discussion

We considered population codes in which neural responses can be expressed as a function of D dimensional latent variables. We considered two classes of such "'tuning curve models'" and found the linear dimensions of the manifold traced out by activities of neural populations. The central result of this work is that for both translation invariant tuning curves and multiplicative tuning curves the linear dimension of these manifolds grow exponentially with D the number of latent variables. Thus we conclude that the manifolds obtained from such models are highly non-linear and linear methods such

as PCA fail dramatically to estimate the number of latent variables that the neural population is encoding.

For translation invariant tuning curves in our model we assumed the tuning curve parameters are spaced uniformly along each dimension. We verify that with this assumption the eigenvalues of the neuron-neuron covariance profile/matrix is given by a Fourier transform. These eigenvalues give the variance along each dimension of a linear subspace that can be fitted to the neural activities over time of this population. Using uncertainty principles relating functions and their Fourier transforms we showed that the linear dimension of translation invariant tuning curves grows at least exponentially. In Section 6.3, we will show that for translation invariant covariance profiles with fixed support, i.e. each neuron is correlated to neurons within a fixed radius in the latent space the linear dimension grows supra exponentially with D, i.e. as $\sqrt{D}^D$. Note that the eigenvalues of the covariance profile can be obtained from a Fourier transform if the tuning curve parameters $\boldsymbol{\alpha}$ of the neurons are distributed uniformly in the D dimensional latent space. However, if the neurons are not distributed uniformly in the latent space, meaning that there are more neurons in a particular region of the latent space and fewer neurons in another region of the latent space, the eigenvalues can no longer be obtained by Fourier transform. We are currently exploring if the eigenvalues of such covariance matrices can be approximated by Fourier transforms.

A commonly used dimensionality measure in the literature is the Participation Ratio (PR) [40]. PR is defined as

$$PR = \frac{\left(\sum_i \lambda_i\right)^2}{\sum_i \lambda_i^2} \tag{2.27}$$

where λ_i's are the eigenvalues of the neuron-neuron covariance matrix. For translation invariant covariance matrices with uniformly spaced tuning curve parameters we can define a similar quantity for the covariance profile centered on a particular neuron

$$PR' = \frac{\left(\sum_n C_{mn}\right)^2}{\sum_n C_{mn}^2} \tag{2.28}$$

The result does not depend on m because of the translation invariant property. There is

an uncertainty relation relating these two quantities

$$PR\,.PR' \geq N \tag{2.29}$$

where N is the total number of neurons. This uncertainty principle again shows us that the fall-off of the covariance profile and the eigenvalues are inversely related. It has been shown in [41], that the PR is equivalent to $L_{1-\epsilon}$ defined by us for $\epsilon \sim 0.15$. Thus our results hold for dimensionality defined by PR as well.

The other class of tuning curves is the multiplicative tuning curves where the total tuning curve can be expressed as a product of tuning along each dimension. We used information theory and bounds on binomial co-efficients to show that the linear dimension grows exponentially with the number of latent variables (D). However, we assumed that the tuning along each dimension is given by the same function. We are currently exploring how this result can be generalized to the case where tuning along each dimension is different and the case where the tuning is a product of groups of latent variables.

The present analysis cautions us against concluding that neural data is high dimensional from linear dimensionality estimation methods. This tells us that if linear dimensions are high, non-linear dimensionality estimation methods or manifold learning methods such as the methods described in [42–44] are required to estimate the dimensions and extract the low dimensional manifold the neural data lies on. From an experimental point of view, high dimensionality of already recorded data might lead scientists to record from more neurons to sample independent directions but this again comes from a linear view of the data-points. We are currently applying these non-linear techniques to the data derived from tuning curve models. Another direction our result points to is a dimensional analysis for data from multiple connected regions. This might give us insight into how representations are transformed from one region to another and specific computations being performed by different regions. For example, [17] hypothesizes that the function of the visual pathway is to untangle object manifolds and make them flatter. If this were true then linear dimensions of representations from subsequent areas in the visual pathway starting from the retina should decrease. Thus our work has important

implications for neural data analysis and can be used by the community to ask questions about the nature of encoding and computations from analysing neural manifolds.

2.6 Appendix

2.6.1 Linear dimension of mean subtracted data is bounded below by the linear dimension of non-mean-subtracted data minus 1

We consider a population of N neurons with activities at time t given by $\boldsymbol{y}(t) = [y_1(t), \cdots, y_N(t)]$. The mean of the population activity is $\boldsymbol{\mu} = \mathbb{E}_t[\boldsymbol{y}(t)]$, where the expectation value is taken over time. The non-mean subtracted covariance matrix is given by $C = \mathbb{E}_t\left[\boldsymbol{y}\boldsymbol{y}^T\right]$. And the mean-subtracted covariance matrix is given by $T = \mathbb{E}_t\left[(\boldsymbol{y}-\boldsymbol{\mu})(\boldsymbol{y}-\boldsymbol{\mu})^T\right]$. Note that $C = T + \boldsymbol{\mu}\boldsymbol{\mu}^T$.

The matrices C, T and $\boldsymbol{\mu}\boldsymbol{\mu}^T$ are Hermitian and positive semi-definite. Moreover, $\boldsymbol{\mu}\boldsymbol{\mu}^T$ is rank 1, with 1 non-zero eigenvalue $||\boldsymbol{\mu}||^2$ and remaining eigenvalues 0. Let the eigenvalues of these matrices be denoted as follows

$$\begin{aligned} C &: \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N \geq 0 \\ T &: \rho_1 \geq \rho_2 \geq \cdots \geq \rho_N \geq 0 \\ \mu\mu^T &: \nu_1 \geq \nu_2 = \cdots = \nu_N = 0 \end{aligned} \tag{2.30}$$

The Weyl inequality for Hermitian matrices yields

$$\rho_a + \nu_b \leq \lambda_p \leq \rho_r + \nu_s \quad a + b - N \geq p \geq r + s - 1 \tag{2.31}$$

With $a = p, b = N, r = p-1, s = 2$ we get

$$\rho_p \leq \lambda_p \leq \rho_{p-1} \quad \text{for } p \geq 2. \tag{2.32}$$

And with $a = 1, b = N, r = 1, s = 1$ we get

$$\rho_1 \leq \lambda_1 \leq \rho_1 + ||\mu||^2 \tag{2.33}$$

Thus we have the eigenvalue bounds

$$0 \leq \rho_N \leq \lambda_N \leq \rho_{N-1} \leq \cdots \leq \lambda_2 \leq \rho_1 \leq \lambda_1 \leq \rho_1 + ||\mu||^2 \tag{2.34}$$

Now assume that the $(1-\epsilon)$ linear dimension of the non-mean-subtracted system is L and that L is at least 2. Thus

$$\frac{\sum_{k=1}^{L-1} \lambda_k}{\sum_{k=1}^{N} \lambda_k} < (1-\epsilon) \tag{2.35}$$

Define $\Delta = \sum_{k=1}^{L-2}(\lambda_k - \rho_k) + \lambda_{L-1}$. Note that $0 \leq \Delta \leq \lambda_1$ by the above inequalities.

$$\begin{aligned}(1-\epsilon) > &\frac{\sum_{k=1}^{L-1} \lambda_k}{\sum_{k=1}^{N} \lambda_k} \geq \frac{\sum_{k=1}^{L-1} \lambda_k - \Delta}{\sum_{k=1}^{N} \lambda_k - \Delta} = \frac{\sum_{k=1}^{L-2} \rho_k}{\sum_{k=1}^{L-2} \rho_k + \sum_{k=L}^{N} \lambda_k} \\ &\geq \frac{\sum_{k=1}^{L-2} \rho_k}{\sum_{k=1}^{L-2} \rho_k + \sum_{k=L-1}^{N-1} \rho_k} = \frac{\sum_{k=1}^{L-2} \rho_k}{\sum_{k=1}^{N-1} \rho_k} \geq \frac{\sum_{k=1}^{L-2} \rho_k}{\sum_{k=1}^{N} \rho_k}\end{aligned} \tag{2.36}$$

Thus $\frac{\sum_{k=1}^{L-2} \rho_k}{\sum_{k=1}^{N} \rho_k} < (1-\epsilon)$.

Consequently if the $(1-\epsilon)$-linear dimension of C is L, the $(1-\epsilon)$-linear dimension of T is at least $L-1$. In general, depending on $||\mu||^2$ the linear dimension of T can be much higher than that of C. Thus, lower bounds on the linear dimension of the non-mean-subtracted case transfer simply to lower bounds on the linear dimension of the mean-subtracted case.

2.6.2 Covariance matrix of translation invariant tuning curves

The neuron-neuron covariance matrix is given by

$$\begin{aligned}C_{mn} &= \frac{1}{N_x} A A^T \\ &= \frac{1}{N_x} \sum_{\boldsymbol{s}} f(\boldsymbol{x_s}, \boldsymbol{\alpha_m}) f(\boldsymbol{x_s}, \boldsymbol{\alpha_n})\end{aligned} \tag{2.37}$$

where N_x is the number of latent variable values. In the limit of large N_x this sample mean will converge to $\mathbb{E}[f(\boldsymbol{x}, \boldsymbol{\alpha_m}) f(\boldsymbol{x}, \boldsymbol{\alpha_n})]_{\boldsymbol{x}}$, where the expectation is taken over the latent variable distribution. Thus the covariance matrix can also be written as

$$C_{mn} = \int_{\mathcal{S}} d\boldsymbol{x}\, p(\boldsymbol{x}) f(\boldsymbol{x}, \boldsymbol{\alpha_m}) f(\boldsymbol{x}, \boldsymbol{\alpha_n}) = \int_{\mathcal{S}} d\boldsymbol{x}\, f(\boldsymbol{x}, \boldsymbol{\alpha_m}) f(\boldsymbol{x}, \boldsymbol{\alpha_n}) \tag{2.38}$$

where $p(\boldsymbol{x})$ is the the distribution over the latent variable, which we assume to be uniform, and the integral $\int_{\mathcal{S}} d\boldsymbol{x} = \int_{\mathcal{S}} dx_1 \cdots dx_D$, is over all points $\boldsymbol{x}$ in the region $\mathcal{S} = [0, 1]^D$. Since the tuning curves are translation invariant we have

$$\begin{aligned} C_{mn} &= \int_{\mathcal{S}} d\boldsymbol{x}\, f(\boldsymbol{x}, \boldsymbol{\alpha_m}) f(\boldsymbol{x}, \boldsymbol{\alpha_n}) \\ &= \int_{\mathcal{S}} d\boldsymbol{x}\, f(\boldsymbol{x} - \boldsymbol{\alpha_n}, \boldsymbol{\alpha_m} - \boldsymbol{\alpha_n}) f(\boldsymbol{x} - \boldsymbol{\alpha_n}, 0) \\ &= \int_{\mathcal{S}-\boldsymbol{\alpha_n}} d\boldsymbol{x}'\, f(\boldsymbol{x}', \boldsymbol{\alpha_m} - \boldsymbol{\alpha_n}) f(\boldsymbol{x}', 0) \\ &= \int_{\mathcal{S}} d\boldsymbol{x}\, f(\boldsymbol{x}, 0) f(\boldsymbol{x}, \boldsymbol{\delta}) \\ &= c(\boldsymbol{\delta}) = c(\boldsymbol{\alpha_m} - \boldsymbol{\alpha_n}). \end{aligned} \tag{2.39}$$

In going from the first to the second line, we have used the translation invariance of the tuning curves ($f(\boldsymbol{x}, \boldsymbol{\alpha}) = f(\boldsymbol{x} - \boldsymbol{\gamma}, \boldsymbol{\alpha} - \boldsymbol{\gamma})$); in going from the second line to the third line we have made a change of variable, $\boldsymbol{x}' = \boldsymbol{x} - \boldsymbol{\alpha_n}$ and defined $\boldsymbol{\delta} = \boldsymbol{\alpha_m} - \boldsymbol{\alpha_n}$; and in going from the third line to fourth line we have used the periodic boundary conditions to shift the integral over $\mathcal{S} - \boldsymbol{\alpha_m}$ to $\mathcal{S}$. c is a periodic function of the difference in tuning curve centers with period 1 along each dimension. Thus the covariance between two neurons is translation invariant for translation invariant tuning curves.

In the one-dimensional case ($D = 1$), if we take the tuning curve centers to be uniformly spaced (i.e., $\alpha_n = n/N$ for $n = 0, \cdots N-1$), then the covariance matrix is circulant

meaning that each row is a shifted version of the row above.

$$\begin{aligned} C_{m+1,n} &= c(\alpha_{m+1} - \alpha_n) \\ &= c((\alpha_{m+1} - 1/N) - (\alpha_n - 1/N)) \\ &= c(\alpha_m - \alpha_{n-1}) \\ &= C_{m,n-1} \end{aligned} \tag{2.40}$$

Note that this holds for $m, n = 1, \cdots N$, since the periodic boundary conditions allow us to define $C_{N+1,n} = C_{1,n}$ and $C_{m,0} = C_{m,N}$.

Finally, note that the results approximately hold if the boundary conditions are hard rather than periodic, provided that the tuning curves are not too wide. In this case, the deviations from perfect translation-invariance caused by the boundary conditions will be mild and restricted to a small subset of neurons.

2.6.3 Eigenvalues of translation invariant covariance matrix

It is well-known that matrices with translation-invariant structure have eigenvalues that are given by the Fourier transform of the function that generates the matrix (i.e., c) [31, 45]. We briefly review those results here.

First, consider tuning curve centers that tile the latent space, forming the points of a lattice with N_d tuning curve centers along the d-th dimension. Thus, the spacing of tuning curve centers along the d-th dimension is $1/N_d$, and there are $N = \prod_d N_d$ neurons in total, in a volume of 1^D. The tuning curve centers are given by

$$\boldsymbol{\alpha_n} = \frac{1}{N_d}(a_1^n, \cdots, a_D^n) \tag{2.41}$$

where each a_d^n belongs to $[0, 1, \cdots, N_d - 1]$.

The covariance matrix has entries $C_{mn} = c(\boldsymbol{\alpha_m} - \boldsymbol{\alpha}_n) = c(\boldsymbol{\delta})$. Note that because of the periodicity of c (as a consequence of the periodic boundary conditions), the vector $\boldsymbol{\delta}$ can be considered to take the same set of possible values as the tuning curve centers.

Next, consider the set of N vectors $\boldsymbol{k_p}$ given by

$$\boldsymbol{k_p} = 2\pi(a_1^p, \cdots, a_D^p), \tag{2.42}$$

where as before each a_d^p belongs to $[0, 1, \cdots, N_d - 1]$. Corresponding to each $\boldsymbol{k_p}$, define the vector $\boldsymbol{v_p} = (e^{i\boldsymbol{k_p}.\boldsymbol{\alpha_1}}, \cdots, e^{i\boldsymbol{k_p}.\boldsymbol{\alpha_N}})$. Multiplying this vector by the matrix C yields

$$\begin{aligned}(C\boldsymbol{v_p})_m &= \sum_{n=1}^{N} c(\boldsymbol{\alpha_m} - \boldsymbol{\alpha_n}) e^{i\boldsymbol{k_p}.\boldsymbol{\alpha_n}} \\ &= \left(\sum_{n=1}^{N} c(\boldsymbol{\alpha_m} - \boldsymbol{\alpha_n}) e^{-i\boldsymbol{k_p}.(\boldsymbol{\alpha_m} - \boldsymbol{\alpha_n})} \right) e^{i\boldsymbol{k_p}.\boldsymbol{\alpha_m}} \\ &= \left(\sum_{n=1}^{N} c(\boldsymbol{\delta_n}) e^{-i\boldsymbol{k_p}.\boldsymbol{\delta_n}} \right) e^{i\boldsymbol{k_p}.\boldsymbol{\alpha_m}}\end{aligned} \tag{2.43}$$

Here we have defined $\boldsymbol{\delta_n} = \boldsymbol{\alpha_m} - \boldsymbol{\alpha_n}$. As a consequence of the uniform spacing of tuning curve centers and periodic boundary conditions, the term in parentheses is a sum over all possible values of $\boldsymbol{\delta_n} = \frac{1}{N_d}(a_1^n, \cdots, a_D^n)$, where as before each $a_d^n \in [0, 1, \cdots N_d - 1]$. Thus, it is independent of both m and n and we can define $\lambda_p = \sum_{n=1}^{N} c(\boldsymbol{\delta_n}) e^{-i\boldsymbol{k_p}.\boldsymbol{\delta_n}}$. Consequently, $\boldsymbol{v_p}$ is an eigenvector of C with eigenvalue λ_p. Note that λ_p is the term with frequency k_p of the discrete Fourier transform of c.

In particular, for the one-dimensional case, C is a circulant matrix. In this case, we have tuning curve centers $\alpha_n = \frac{n}{N}$, where $n \in [0, \cdots, N-1]$. For each p with $0 \leq p \leq N-1$, $\boldsymbol{v_p} = (e^{ik_p.\alpha_1}, \cdots, e^{ik_p.\alpha_N})$ is an eigenvector with eigenvalue $\lambda_p = \sum_{n=0}^{N-1} c(\alpha_n) e^{-2\pi i p\, n/N}$.

Alternatively, if tuning curve centers are uniformly distributed through the latent space, either evenly spaced or randomly chosen, then in the large N limit we can consider the translation invariant kernel $c(\boldsymbol{\alpha_m} - \boldsymbol{\alpha_n})$, which is a continuous function of $\boldsymbol{\alpha_m}, \boldsymbol{\alpha_n} \in \mathbb{R}^D$. This is the continuous generalisation of the covariance matrix for translation invariant tuning curves derived above. The product of the matrix C with a vector can then be approximated by the convolution of the kernel with a function $f_1(\boldsymbol{\alpha_n})$,

$$f_2(\boldsymbol{\alpha_m}) = \int d\boldsymbol{\alpha_n}\, c(\boldsymbol{\alpha_m} - \boldsymbol{\alpha_n})\, f_1(\boldsymbol{\alpha_n}) \tag{2.44}$$

where $\int d\boldsymbol{\alpha_n}$ means an integral over all D dimensional vectors $\boldsymbol{\alpha}_n$. $f(\boldsymbol{\alpha}_n)$ is an eigenfunction if

$$\int d\boldsymbol{\alpha_n}\, c(\boldsymbol{\alpha_m} - \boldsymbol{\alpha_n})\, f(\boldsymbol{\alpha_n}) = \lambda f(\boldsymbol{\alpha_m}) \tag{2.45}$$

Consider $f(\boldsymbol{\alpha}_n) = e^{i\boldsymbol{k}.\boldsymbol{\alpha_n}}$ where $\boldsymbol{k}$ is an arbitrary D dimensional vector.

$$\begin{aligned} &\int d\boldsymbol{\alpha_n}\, c(\boldsymbol{\alpha_m} - \boldsymbol{\alpha_n})\, e^{i\boldsymbol{k}.\boldsymbol{\alpha_n}} \\ &= \int d\boldsymbol{\alpha_n}\, c(\boldsymbol{\alpha_m} - \boldsymbol{\alpha_n})\, e^{-i\boldsymbol{k}.(\boldsymbol{\alpha_m} - \boldsymbol{\alpha_n})}\, e^{i\boldsymbol{k}.\boldsymbol{\alpha_m}} \\ &= \left(\int d\boldsymbol{\delta}\, c(\boldsymbol{\delta})\, e^{-i\boldsymbol{k}.\boldsymbol{\delta}} \right) e^{i\boldsymbol{k}.\boldsymbol{\alpha_m}} \end{aligned} \tag{2.46}$$

As before in going from line 2 to line 3, we have defined a new variable, $\boldsymbol{\alpha_m} - \boldsymbol{\alpha_n} = \boldsymbol{\delta}$ and used the periodic boundary conditions to replace the integral over $\boldsymbol{\alpha_n}$ with an integral over $\boldsymbol{\delta}$. The expression in parenthesis does not depend on $\boldsymbol{\alpha_m}$ or $\boldsymbol{\alpha_n}$ and is the Fourier transform of $c(\boldsymbol{\delta})$. So $e^{i\boldsymbol{k}.\boldsymbol{\alpha_n}}$ is an eigenfunction of $c(\boldsymbol{\alpha_m} - \boldsymbol{\alpha_n})$ with eigenvalue $\int d\boldsymbol{\delta}\, c(\boldsymbol{\delta})\, e^{-i\boldsymbol{k}.\boldsymbol{\delta}}$.

Finally, while these results are for uniformly distributed data with periodic boundary conditions, note that covariance matrices are symmetric and thus normal. Consequently, the effect of perturbations on the eigenvalue spectrum is as mild as possible, and the results should hold approximately for matrices that only approximately satisfy these conditions.

2.6.4 Exponential and faster growth of linear dimension for localized tuning curves from uncertainty principles

We assume that the neurons have N_D equally spaced tuning parameters along each dimension in the hypercube $[-1/2, 1/2]^D$. Thus there are a total of N_D^D neurons in a volume of $\mathbf{1}^D$. We also assume that the covariance profile of a neuron is supported on a fixed radius $R = k\sigma$ around each neuron, such that $k\sigma < 0.5$ since the covariance profile cannot

be supported outside $[-1/2, 1/2]^D$. Thus

$$\begin{aligned} c(\boldsymbol{\alpha_m} - \boldsymbol{\alpha_n}) = c(\boldsymbol{\delta}) > 0, \;\; |\boldsymbol{\delta}| \leq R \\ = 0, \;\; |\boldsymbol{\delta}| > R \end{aligned} \tag{2.47}$$

The support of the covariance profile c is thus given by the number of points inside a sphere of radius R. Since there are N_D^D points in a volume of 1^D, the support of the covariance profile c is,

$$\begin{aligned} |supp(c)| =& Vol_D(R) \times N_D^D \\ =& \frac{\pi^{D/2}}{\Gamma(D/2+1)} (k\sigma)^D N_D^D \sim \frac{1}{\sqrt{D\pi}} \left(\frac{2\pi e}{D}\right)^{D/2} (N_D k\sigma)^D \end{aligned} \tag{2.48}$$

As we have shown for translation-symmetric tuning curves the eigenvalues of the covariance profile are given the Fourier transform of the covariance profile c. Let the eigenvalue profile be $\tilde{c}$. From the uncertainty principle [34]

$$\begin{aligned} |supp(c)||supp^\epsilon(\tilde{c})| \geq& N_D^D(1-\epsilon) \\ |supp^\epsilon(\tilde{c})| \geq& \frac{N_D^D(1-\epsilon)}{|supp(c)|} \\ |supp^\epsilon(\tilde{c})| \geq& (1-\epsilon)\sqrt{D\pi} \left(\frac{D}{2\pi e}\right)^{D/2} \left(\frac{1}{k\sigma}\right)^D \end{aligned} \tag{2.49}$$

Therefore, for translation tuning curves which are localized within a fixed radius, the linear dimension $L_{1-\epsilon} = |supp^\epsilon(\tilde{c})|$ grows faster than exponentially as $\left(k\sigma\sqrt{D}\right)^D$.

Chapter 3

Interactions between brain regions via sparse random connections

3.1 Introduction

The brain can be divided into spatially localized regions which share certain neurobiological properties such as cellular and circuit properties [46]. Identifying and defining the boundaries of parcellated brain areas itself is a subject of active research. The advent of noninvasive imaging techniques like fMRI has produced brain maps with greater detail identifying parcellated brain regions with and connections among them. However, whether each of these areas are involved in a particular brain function or functions are distributed over many regions has been a subject of intense debate. Certain brain areas which are engaged in particular mental functions have indeed been identified [47], for example: the fusiform face area (FFA), which responds selectively to faces [48, 49], the parahippocampal place area (PPA), which responds selectively to places [50], and the extrastriate body area (EBA), which responds selectively to bodies and body parts [51]. Though certain regions are more engaged in specific mental functions than other functions, it has been observed that many brain regions are active during more than one cognitive/mental task. For example, the visual area MT/V5 is involved in perception of visual motion [52, 53] and also contains information about stereo depth [54]. The activity

of more than one brain area during a mental task suggests that different areas dynamically interact with each other to implement a task. Multiple distributed regions underlie coginitive processes such as language [55], cognitive control [56], emotion [57] and social cognition [58]. To understand how interactions between different brain regions enable it to perform various cognitive functions, Olaf Sporns and collaborators in a series of papers [59–62] have proposed a complex network theory to study the organization of brain networks. In this approach, the brain network is a graph where each node can represent a brain region or a neuron or a group of neurons depending on the spatial resolution of the model and edges represent the connectivity between the nodes. In this set-up one can study different questions, like the subnetworks involved in different functions, or dynamics evolving on this network. We take a graph theory approach to study the connectivity between brain regions and the dynamics of information exchange between the regions as a dynamical system evolving on this graph.

Understanding the interactions between brain regions requires a knowledge of connectivity between these regions. Brain regions communicate and exchange information through white matter tracts which are bundles of axons surrounded by myelin sheaths. Brain connectivity maps are created using white matter tract tracing from imaging methods such as diffusion MRI and resting state fMRI [63]. Several white matter tracing studies have shown that there are very few long distance connections in the brain. In particular, the number of connections between neurons has been shown to fall off exponentially with distance [9].

We ask the question: *how can information be efficiently transmitted between distant brain regions using sparse connections?* We do not consider any particular brain region, but model a generic region as a collection of neurons. We assume that the target region has fewer neurons than the source region, thus there is information also gets compressed while being transmitted. Can information be preserved in such kind of communication? We show that information can indeed be conserved in such a situation in Section 3.2.2.

Connectivity in the brain is neither entirely random nor entirely regular [64]. Randomness is a good proxy whenever the exact connectivity between neurons is not known

and randomly connected networks has been studied extensively [65–67]. In this work, we model the sparse connectivity between distant brain regions as sparse random connectivity. In particular we assume that each neuron in a source region is randomly connected to a small fixed number of neurons in the target region. Such a network is an expander graph with high probability. These graphs are well connected in spite of being sparse. We formally define expander graphs in Section 3.2. In this work we consider different neural networks with the expander property and explore the computational and information processing advantages that this property confers on these networks.

In Sections 3.3 and 3.4 we consider long distance interactions between brain regions with sparse expander connections in two different architectures. The first architecture consists of two regions connected by convergent feedforward and reciprocal divergent feedback connections. Such networks are hypothesized to play a role in working memory [68–70] and recall [71–73]. In Section 3.3.1 we define dynamical equations on this network which lead to maintenance of activity patterns corresponding to a sensory signal (working memory) and reconstruction of sensory signal corresponding to recall of a mental image. We discuss how expander connections lead to faster convergence to a fixed point. We discuss how these computations can be recast as compressed sensing problem in signal processing and propose a neurally plausible algorithm which can be used for working memory or recall dynamics.

In Section 3.4 we revisit the problem of communication between distant brain regions as considered in [74]. In particular we ask: can randomly compressed patterns received by the target area be re-expanded in a way to perform computations on it? To answer this, we make a connection between compressed sensing and sparse approximation. We consider an algorithm for sparse approximation known as locally competitive algorithms (LCA) [75]. We explore the consequences of expander connectivity in LCA. It has been hypothesized that the brain forms expanded sparse representations because it is easier to classify them or separate them [76–78]. We check if the sparse expanded patterns can be arbitrarily classified using a perceptron like neuron.

3.2 Expander Graphs

Expander graphs are graphs in which small sets of vertices have large number of neighbors. Therefore, the edges coming out of any small subset of vertices in the graph expands out to reach a large set of vertices which form the neighbor subset. We will quantify this notion below for bipartite expander graphs which have been used for compressed sensing/sparse signal recovery.

A bipartite graph is one in which the vertices can be divided into two sets such that there are no edges in between vertices belonging to the same set. Let $G = (V, E)$ be a bipartite graph, where $V = L_1 \cup L_2$ and $L_1 \cap L_2 = \Phi$. We will call one set of nodes, the left nodes L_1 with $|L_1| = N$ and one set of nodes, the right nodes L_2 with $|L_2| = M$. The degree of a vertex is defined as the number of edges coming out of that vertex. We consider graphs where each vertex in L_1 has c edges. Such graphs are known as graphs with regular left degree. The neighbors of a node A are defined as the nodes which are connected to A by an edge. The neighbor set $\mathcal{N}(S)$ of a subset S of nodes are the neighbors of the nodes in S.

Definition 3.2.1. *The graph described above is an expander graph with parameters* (c, α, ϵ) *if for every subset* S *of left nodes with* $|S| \leq \alpha N$, $|\mathcal{N}(S)| > (1 - \epsilon)c|S|$ *with constants* $\alpha, \epsilon < 1$ *where* $\mathcal{N}(S) \subset L_2$ *is the neighbor set of* S *which is a subset of* L_2.

It turns out that random bipartite graphs with regular left degree are expander graphs with high probability [79]. For example, Fig 3.6b shows a the distribution of the sizes of the neighbor subsets. For this particular random graph the parameters can be inferred as $N = 1000, M = 500, c = 5, \alpha = 0.04, \epsilon = 0.25$. Note here that α depends on M, ϵ.

3.2.1 Can evidence for expander connections be found experimentally?

Suppose an experiment can find all the connections between neurons of two regions of interest with N and M neurons respectively. Let us assume for now that we are not interested in the connections between the neurons with each region. We can express the

connections between these two regions as the $(N + M) \times (N + M)$ adjacency matrix of a bipartite graph where the matrix would have the following block form

$$\begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix} \tag{3.1}$$

This matrix has a typical spectrum of eigenvalues as shown in Fig 3.1c . As can be seen in the figure all eigenvalues occur in positive and negative pairs. Eigenvalues are real since this is a symmetric matrix. There is a gap between eigenvalues of maximum magnitude and the eigenvalues of the second maximum magnitude which is known as the spectral gap and is a feature of the eigenvalues of an expander graph. In general, large expansion implies large spectral gap and large spectral gap implies large expansion [80]. A bound on the spectral gap in bipartite expander graphs has been proven in [81].

3.2.2 Communication in distant brain regions using expander graphs

The model for long distance communication has been set-up in two different ways in the literature. Since any area sends only a few long distance connections, it was assumed in [74], that the information in a source region is first compressed into the activities of a few neurons in the source area which then send long distance axon projections to the target area. The authors assumed that this compression is akin to compression by a random matrix and considered the question, how the target area "makes sense" of this randomly sub-sampled information transmitted to it without any information about how it was sub-sampled. To make matters more difficult the source area and the target area may have different number of neurons. We consider a slightly different set-up where each neuron in the source area is directly connected to a few neurons in the target area and the target area has a smaller number of neurons than the source area thus serving as an information bottleneck. [74] assume that the information is compressed in the source area before being transmitted thus only a few long distance projections are required while we assume

that information is being compressed while being transmitted using a few projections. Our setup is equivalent to [74], when the information in the source area is compressed using sparse random projections. Both of these situations are shown in Fig 3.1d.

As shown in Fig 3.1a, sparse random expander like projections map separate patterns in the source area to separate patterns in the target area. Mathematically, this can be understood from the definition of expander graphs as given in Def. 3.2.1. Using this definition one can show that any vector $\boldsymbol{x}$ of size N, with k non-zero entries with $k \leq \alpha N$, when mapped to a vector of size M by a bipartite expander graph with parameters α, ϵ, c has a ℓ_1 norm at least equal to $(1-2\epsilon)c|x|_1$. Theorem 6.2 and 6.3 (SI Section 6.2) tells us when $M \times N$ binary random matrices with c ones in each of its columns is the adjacency matrix of a bipartite expander graph. Fig 3.1b shows the distribution of $|\mathcal{N}(S)|/c|S|$ for different values of leftnode subset sizes $|S|$. We want to find the parameter value α for this random graph with regular left degree with parameter $N = 1000, M = 500, c = 5, \epsilon = 0.25$. In this Figure, the horizontal dashed line is at $y = 0.75$. Therefore in this example, $\alpha = 0.04$, since the $|\mathcal{N}(S)| > (1-\epsilon)c|S|$ for $|S| < 40$.

3.3 Reciprocally connected networks

Neuroanatomical evidence shows that there are successive levels of convergence in the brain starting from early sensory cortices onto sensory-specific association cortices and to multisensory association cortices, culminating in maximally integrative regions such as in medial temporal lobe cortices and both lateral and medial prefrontal cortices [82]; and the convergence of sensory pathways is reciprocated by successive levels of divergence, from the maximally integrative areas to the multisensory association cortices, to the sensory-specific association cortices, and finally to the early sensory cortices [83–86]. In this section we consider two regions as two sets of neurons with sparse expander like convergent feedforward projections and reciprocal divergent feedback connections. We explore if dynamics evolving over such a network can maintain persistent signals (working memory) and reconstruct signals (recall).

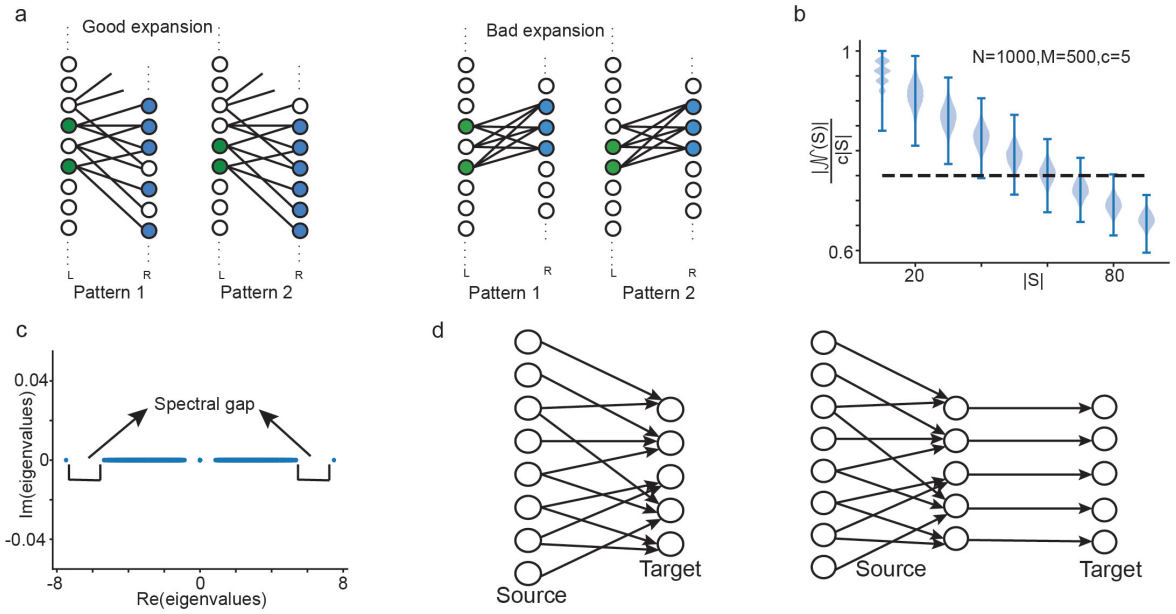

Figure 3.1: **Properties of expander graphs and two models of communication.** a) Two regions connected by sparse expander connections and connections with no expansion. The sparse expander connections separate patterns in the source area to different patterns in the target area where as the connections with no expansion map separate patterns in the source region to same pattern in the target region. b) Violin plot showing the distribution of $\frac{|\mathcal{N}(S)|}{c|S|}$ as function of $|S|$ where S is a subset of L_1 and $\mathcal{N}(S)$ is the neighbor set of S. This distribution was created for a random bipartite graph with regular left degree c. It turns out to be an expander graph with parameters $\epsilon = 0.25, \alpha = 0.04, c = 5$. The dashed line is at 0.75. Note that according to Def. 3.2.1, the y-axis gives $(1 - \epsilon)$ hence we can find the parameters of the expander graph from the plot. c) The spectrum of the adjacency matrix of the expander graph shown in panel b, as in Eq. 3.1. d) The two models of communication discussed below. In the first model, the source regions transmits signals to a target region with a smaller size using sparse random connections. In the second model, information is compressed in the source region and then transmitted to the target region via few long range projections as considered in [74].

3.3.1 Working memory

Working memory is the ability to hold information in mind over periods of seconds or minutes. This information can be sensory information such as the image of an object or the tune of a song or it can be abstract information like a number. Thus, one of the main features of working memory is that it can hold any kind of information. The neural mechanisms underlying this ability have been the subject of intense research both experimentally and computationally [87–89]. One of the main findings of this research is that information is maintained in working memory by the persistent firing of groups of neurons distributed in the brain network [90]. This network consists of the prefrontal

cortex, the parietal and temporal association areas of the cerebral cortex, cingulate and limbic areas, and subcortical structures such as the mediodorsal thalamus and the basal ganglia. Persistent activity is thought to be maintained by the reverberating activity of neurons in these areas. The persistent activity is content specific – different sets of neurons fire persistently to represent different objects or information in the working memory.

We present a network model of two regions connected by random reciprocal connections where content specific activity can be maintained through reverberations in the network. The first region or layer which we call L_1 has N neurons and the second region or layer L_2 has M neurons with $M < N$. A stimulus is presented briefly to L_1 and then removed. In our model the stimulus is a binary signal of size N which also represents the activity of the neurons in L_1. One can imagine that there is some network mechanism prior to L_1 which leads to a particular binary signal for the neurons of L_1 for the presentation of a stimulus. Each neuron in L_1 is randomly connected to c neurons in L_2, as shown in Fig 3.5a.

Such a random regular graph with a fixed left degree is an expander graph with high probability as discussed above. Thus activities corresponding to different stimuli are mapped to different activity patterns in L_2. The dynamics of the network as described below is able to maintain sparse patterns of activity in both L_1 and L_2 even after the stimulus is removed.

Let x and y represent the activities of populations L_1 and L_2 respectively. The network has the following dynamics

$$\begin{aligned} x(t) =& x_0, t < t_{stim} \\ \tau_1 \frac{dx}{dt} =& -x + \phi_x(A^T y), t > t_{stim} \\ y(0) =& 0 \\ \tau_2 \frac{dy}{dt} =& -y + \phi_y(Ax). \end{aligned} \tag{3.2}$$

The stimulus x_0 which is a random binary vector with k ones is presented for time t_{stim}.

ϕ_x and ϕ_y are thresholding functions as follows

$$\begin{aligned}\phi_x(x) =& \frac{1}{1+\exp(-\alpha_x(x-x_{th}))}, x > x_{th} \\ =& 0, x < x_{th}, \\ \phi_y(y) =& \frac{1}{1+\exp(-\alpha_y(y-y_{th}))}, y > y_{th} \\ =& 0, y < y_{th}\end{aligned} \tag{3.3}$$

These thresholding functions were chosen so that the activities of the neurons quickly saturate to 1 after crossing the the threshold. This also prevents the network from becoming unstable due to arbitrarily high firing rates. The values of the various parameters are given in the following Table.

Parameters	Values
t_{stim}	$2 \times \tau_2$
τ_1	5ms
τ_2	20ms
α_x	8
α_y	8
x_{th}	[0.8c]
y_{th}	0

Table 3.1: Parameter values for Eq.3.2

To gain intuition into why this particular connection structure is helpful, let us consider only one active neuron in the pattern x_0. This neuron has c outgoing connections to L_2 and c incoming connections from L_2. Thus this neuron receives exactly c inputs. Since the threshold is $\lceil(0.8c)\rceil$, this neuron remains active ($\lceil . \rceil$ is a function that gives the closest integer greater than the argument). For any other neuron in L_1 to get activated, it has to be connected to at least $\lceil(0.8c)\rceil$ of the c neurons the original active neuron is connected to. The probability of this happening given by $\frac{\binom{c}{\lceil(0.8c)\rceil}}{\binom{M}{c}}$ which is 3.9×10^{-12} for $M = 500, c = 5$. An error occurs in a situation where the pattern of activity that is maintained differs from the original activity pattern. We quantify the error by $|x(t) - x_0|_1$.

Now suppose a small set S of neurons in L_1 are active. Let $\mathcal{N}(S)$ be the set of neurons in L_2 that they are connected to. The probability of selecting a neuron outside

S is $(N-|S|)/|S|$. A neuron outside S will be activated by reciprocal connections from L_2 if it is connected to more than $\lceil(0.8c)\rceil$ neurons in L_2. Let A_{L_1} be a node outside S. Let X be the number of neighbors A_{L_1} has in $\mathcal{N}(S)$. Since A_{L_1} has a total of c neighbors, it must have X neighbors in $\mathcal{N}(S)$ and $c-X$ neighbors in $L_2/\mathcal{N}(S)$. The probability for such an occurrence is given by

$$\begin{aligned} P(X=\lceil 0.8c\rceil) =& \frac{\binom{|\mathcal{N}(S)|}{\lceil(0.8c)\rceil}\binom{M-|\mathcal{N}(S)|}{c-\lceil(0.8c)\rceil}}{\binom{M}{c}} \\ P(error=1 \;\Big|\; |S|) =& P(X=\lceil 0.8c\rceil)\times\frac{N-|S|}{|S|}\times P_{max}(|\mathcal{N}(S)| \;\Big|\; |S|) \end{aligned} \tag{3.4}$$

The distribution of $|\mathcal{N}(S)|$ for each $|S|$ is shown in Fig. 1b. Using the probability for most probable $|\mathcal{N}(S)|$ given $|S|$, we find the probability for an error as shown in Fig.3.2c. Thus a binary pattern with $k=25, N=1000, M=500, c=5$ ones can be maintained quite accurately since the probability for an error is roughly 0.009. We also note that the delay period activity might not be exactly the same as the stimulus activity and a few errors might be allowed as long as the stimulus identity can be decoded from the activity of the delay period [70].

We would like to emphasize that the connectivity in our model is different from Hopfield networks [6] whose connection weights are constructed from the patterns that the network can store as fixed point. The expander property of random regular graphs combined with the non-linearity allows our model to maintain any binary pattern up to a given number of nonzero elements without any errors through reverberatory activity between areas. A natural next step would be to extend this model to include multiple areas as persistent activity related to working memory has been observed in multiple brain areas [90].

The same network structure with two layers of neurons connected to each other to maintain persistent activity was used in [70]. The three main differences between [70] and our model are: 1) In [70], the sparse random reciprocal connections are modelled by an Erdos-Renyi graph where each connection occurs with a small probability p where as in our model each neuron in L_1 sends and receives a small fixed number of connections, 2)

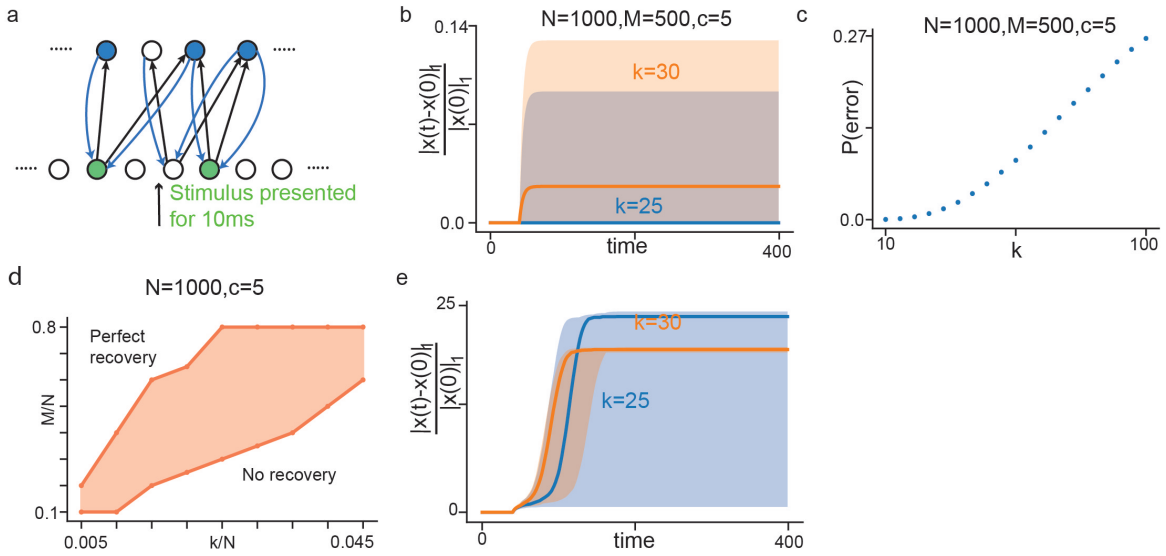

Figure 3.2: **Performance of reconstruction dynamics as in Eq.**(3.2)**.** (a) Cartoon of the network architecture and dynamics. Stimulus as a binary signal is presented to L_1 neurons for a short period of time and then removed. Signal is maintained in the network through reverberatory activity. (b) Normalized error defined as $\frac{|x(t)-x(0)|_1}{|x(0)|_1}$ as a function time for $|x(0)|_1 = k = 25$ (blue) and $k = 30$ (orange). Solid lines are medians and shaded area represents 100% confidence interval. For more details see Section 3.4. (c) Probability of an error verses k as in Eq. 3.4 (d) Shaded area corresponds to region in $M - k$ plane where perfect reconstruction is possible. (e) Normalized error as a function time for dynamics as in Eq. 3.2 on a Erdos-Renyi bipartite graph where the connection between a neuron in L_1 and L_2 occurs with probability $p = c/M$ with parameters as in panel (b). Note that the normalized error is $>> 1$ which means that the signal being maintained has many more active units than the original signal.

In [70], the first layer was either considered to have a ring structure with lateral inhibition or to be a Hopfield network with structured inhibition which is closer to our models. We have compared the reconstruction errors of a network with random regular connections with Erdos-Renyi connections where the probability $p = c/M$ so that the average sparsity in the Erdos-Renyi case is the same as the random regular case in Figure 3.2e . We see that the network with random regular connections is able to maintain original pattern with fewer errors than the network with Erdos-Renyi connections. 3) We consider a firing rate model whereas [70] have considered a linear-nonlinear spiking model.

We note that this simple network with just reciprocal excitatory connections is able to maintain any binary sparse pattern with number of non-zero elements determined by the parameters N, M, c. For random bipartite graphs with regular left degree, these parameters can be determined probabilistically. A more physiologically plausible network

should include broad inhibition in each of the areas [91–93] to ensure excitation-inhibition balance.

3.3.2 Future extensions of the working memory model

This problem can also be recast as sparse signal reconstruction problem. Suppose the stimulus x_0 is presented for long enough such that the neurons in L_2 reaches a steady state $\phi_y(A\boldsymbol{x_0})$. Since the neurons in L_1 have a smaller time constant, the activity corresponding to x_0 decays quickly, and is reconstructed from the activity of neurons in L_2. This is a classic problem called compressed sensing in applied math and has been well studied [94]. We will describe this problem briefly in Section 3.6.1. In Section 3.6.3 we will describe how expander graphs have been used in compressed sensing. We propose an algorithm, similar to these algorithms to reconstruct sparse binary signals which we call the signed transpose algorithm as described in Section 3.3.5. As a next step, we are trying to include inhibition in such a way that the network dynamics mimic the signed transpose algorithm, thereby increasing the number of patterns that the network dynamics can maintain. In addition to this, as already mentioned we need to include global inhibition to maintain excitation-inhibition balance.

The above model is a combination of binary neurons with firing rate models. We start with an activity pattern which is represented by neurons being on or off and combine it with a firing rate model for neurons to maintain persistent activity. The firing rates for the neurons are still bounded by 1 due to the saturating non-linearity. As a next step We will consider the case where the stimulus is still represented by the neurons which are active but the firing rates of the neurons are not bounded by 1. In such a model the activities have to be stabilized by inhibition. Such a model would correspond more closely to physiological neurons with analog firing rates. In addition, if this model is able to reproduce low irregular firing rates of the neurons which are active during the delay period it will take our model one-step closer to experimentally observed delay period activity.

Working memory related activity has been observed in multiple brain areas [90]. In our

model, content specific activity is maintained in two areas through reciprocal connections. A natural next step is to extend this model beyond two areas to include multiple areas connected by reciprocal connections.

3.3.3 Recall

We are able to recall sensory information such as mental images, full songs from cues such as words, situations, beginning of a song, a tune similar to the song. Such cues can be hypothesized to produce content specific activity in the pre-frontal cortex (PFC). We can think of the L_2 neurons in our model as PFC. The top-down connections from PFC to a sensory cortex (L_1 neurons) can activate the pattern of activity corresponding to the information one is trying to recall. This would be same as reconstructing the initial pattern of activity representing the object in L_1. Such reconstruction of representations of information in the sensory cortex through top-down activation and reciprocal connections has been hypothesized as a mechanism for recall [71, 95].

In our model this corresponds to initial conditions

$$\begin{aligned} x(0) &= x_0 \\ y(0) &= Ax_0 \end{aligned} \tag{3.5}$$

where x_0 is the activity pattern representing some sensory information. The pattern in L_1 is no longer held fixed for t_{stim}. We show that the initial activity pattern can be reconstructed in L_1, for sparse binary patterns with a given number of non-zero elements k determined by N, M, c, the parameters of the network. This is exactly the reconstruction problem that compressed sensing solves. We show that the dynamics described in Eq. 3.2 can recover sparse binary patterns of activity which corresponds to recall of the sensory information. Note that we use the same parameters for the network dynamics as in Table 1.

We also considered the case of pattern completion. The cue maybe such that some of the neurons in L_1 are also activated due to the cue. In this case we observe that the

dynamics is able to reconstruct the original activity pattern x_0 if

$$\begin{aligned} x(0) &= \hat{x}_0 \\ y(0) &= Ax_0 \end{aligned} \tag{3.6}$$

where $\hat{x}_{0_i} = x_{0_i}$ for $i \in S \subset T$ where T is the set of indices where $x_{0_j} = 1$. Thus $\hat{x}_0$ is equal to x_0 in a few indices. We assumed that S is randomly chosen from T and $|S| = 0.4|T|$. Thus $\hat{x_0}$ is nonzero in 40% of the indices where x_0 is 1. An example of this is shown in Fig 3.3 d . We noticed that reconstruction fails with this dynamics if $y(0) \neq Ax_0$. In other words the current dynamics will fail to recover the original pattern if the cue does not activate the original sensed vector Ax_0 in L_2. The performance of this dynamics is shown in Figure 3.3 b,d. An interesting question We are going to consider next is if the above mentioned extensions of the model (see Section 2.1.1) is able to reconstruct the original vector when the cue only activates the sensed vector in L_2 partially.

3.3.4 Simulation details for Figure 2 and Figure 3

Sparse random binary vectors of sparsity k were created by selecting k random indices of a length N 0-vector and setting them to 1. The $M \times N$ random binary connectivity matrix A between the two layers were created by setting c randomly selected indices in each column of the matrix A to 1. The input to the layer 1 was held fixed for $2 \times \tau_2 = 10ms$. Eq. (3.2) was integrated using the Euler method with the parameters given in Table 1 and $dt = 1ms$. Fig. 3.2b shows the normalized difference of the ℓ_1 norm of the original vector and the reconstructed vector over time. For each sparsity k, the dynamics was run 50 times with different inputs of sparsity k. The lines represent the median and the shaded region represents the spread in the error of the reconstructed vector in the first layer. For Fig. 3.2d, a connectivity matrix A was created for each combination of N, M, c. The reconstruction dynamics of Eq. (3.2) was run 50 times for different inputs with sparsity k. The shaded region corresponds to the area where the fraction of times there was perfect recovery was > 0 and ≤ 1 for combinations M, k. For Fig. 3.2e, each entry of the connectivity matrix is an i.i.d Bernoulli random variable with $A_{ij} \sim Bernoulli(c/M)$. The

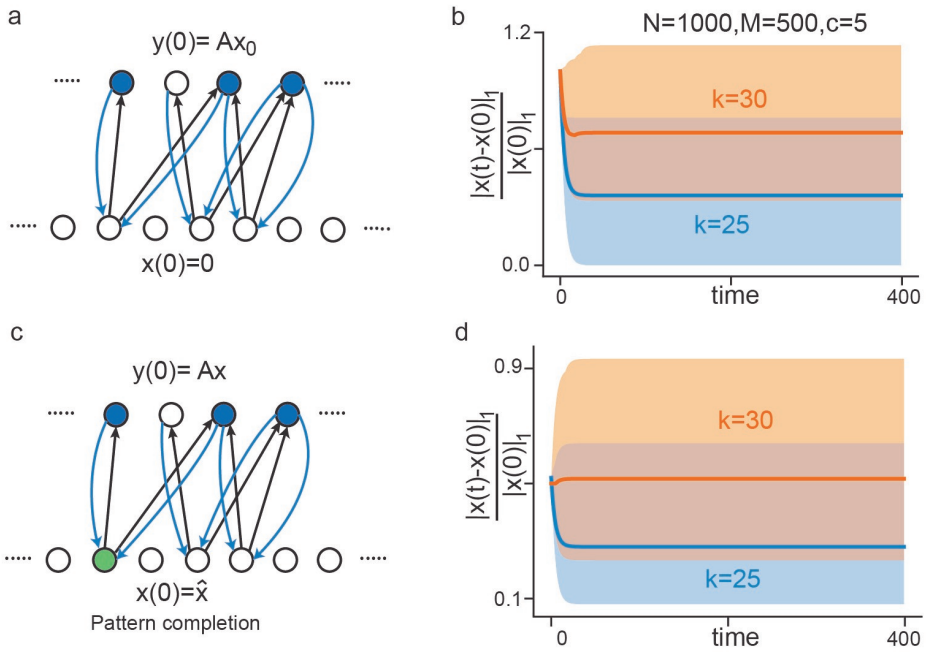

Figure 3.3: **Recall as reconstruction of original activity pattern starting from two different initial conditions and using the dynamics in** (3.2). (a) Initial conditions where the neurons in L_1 are completely inactive and the neurons in L_2 are have activity given by $\boldsymbol{y} = A\boldsymbol{x_0}$. (b) Ratio of errors in activity of L_1 and original activity over time. Lines are median and shaded area is 100% confidence interval. For $|\boldsymbol{x}(0)|_1 = k = 25$, perfect reconstruction is possible since the minimum number of errors $|\boldsymbol{x}(t) - \boldsymbol{x}(0)|_1$ goes to 0 with time whereas for k=30 it remains positive. For k=25, the median of $|\boldsymbol{x}(t) - \boldsymbol{x}(0)|_1 = 8$ while for k=30, the median of $|\boldsymbol{x}(t) - \boldsymbol{x}(0)|_1 = 18$. (c) Initial condition in which the L_1 neurons are partially active to correspond to the original signal. (d) Ratio of errors and k with time for initial conditions as in (c). For k=25, the median of $|\boldsymbol{x}(t) - \boldsymbol{x}(0)|_1 = 7$ while for k=30, the median of $|\boldsymbol{x}(t) - \boldsymbol{x}(0)|_1 = 16$.

recall and the pattern completion dynamics were implemented using the same equations and same parameters. For Fig. 3.3, the initial condition for the dynamics in Eq.(3.2) was $\boldsymbol{x}(0) = x_0$ and nothing was held fixed for t_{stim}. For Fig. 3.3, $\boldsymbol{x}(0) = \hat{x}$ where $\hat{x}_i = x_i$ for 40% of the indices where $x_i = 1$. As in Fig. 3.2, the dynamics was run 50 times for different inputs with given sparsity. The lines represent the median errors and the shaded regions represent the spread.

3.3.5 Signed transpose algorithm

Expander graphs have been used in different fields like computer science, signal processing, error correcting codes. The property of expander graphs that small subsets of nodes

have large neighbor subsets has been used in error correcting codes [96] and compressed sensing [97] as described in Section 5.3. We have modified this algorithm slightly to propose the following approach, which we call the signed transpose algorithm.

Let $\boldsymbol{x}$ be the original signal and A be the measurement matrix and $Ax = y$ be the sensed signal. let $\hat{\boldsymbol{x}}$ denote the signal being reconstructed. The vector $\boldsymbol{g} = \boldsymbol{y} - A\hat{\boldsymbol{x}}$ is known as the gap vector. The algorithm is as follows:

1. Start with $\hat{\boldsymbol{x}} = 0$.

2. If $g = y - A\hat{\boldsymbol{x}} = 0$ then terminate. If not, find a node $\hat{x}_i$ which is connected to $> c/2$ non-zero gaps.

3. If this node is connected to $> c/2$ positive gaps then $\hat{x}_i = 1$, otherwise if it is connected to $\geq c/2$ negative gaps then $\hat{x}_j = 0$. Go to 2.

We note that this algorithm works for only sparse binary vectors. As shown in Section 5.3, for an expander graph with parameters $\alpha, c, 1/4$, and $|x|_{l_0} < \alpha N$ we can always find a node $\hat{x}_i$ which is connected to $> c/2$ unique gaps. This algorithm is slightly different from the rest, in that a node in L_1 is turned on if it is connected to $> c/2$ positive gaps and turned off if it is connected to $\geq c/2$ negative gaps. This algorithm is able to reconstruct vectors with greater number of nonzero elements k than the bit-flip (see Section 3.6.3) and the greedy gap algorithms (see Section 3.6.3). The reconstruction performance for different combinations of k, N, M, c is shown in Figure 3.4. We are currently trying to develop semi-analytical results to explain this performance. We think the results in [81] might be helpful in finding analytical limits of the performance. In particular we would like to find M/N as a function of k/N which would be required for perfect reconstruction. The results from simulations for two different values of c are shown in Fig. 3.4b. Such a result will help us find the scaling of M with k as in other traditional compressed sensing results [94, 97–100]. Fig. 3.4d shows that the reconstruction is perfect for c in the range $7 - 12$. We would like to check if this range is possible to find theoretically as well. The algorithm converges faster for small values of c as shown in Fig. 3.5f. As a next step we would like to implement this dynamics in a neurally plausible way to model working memory or recall as discussed above.

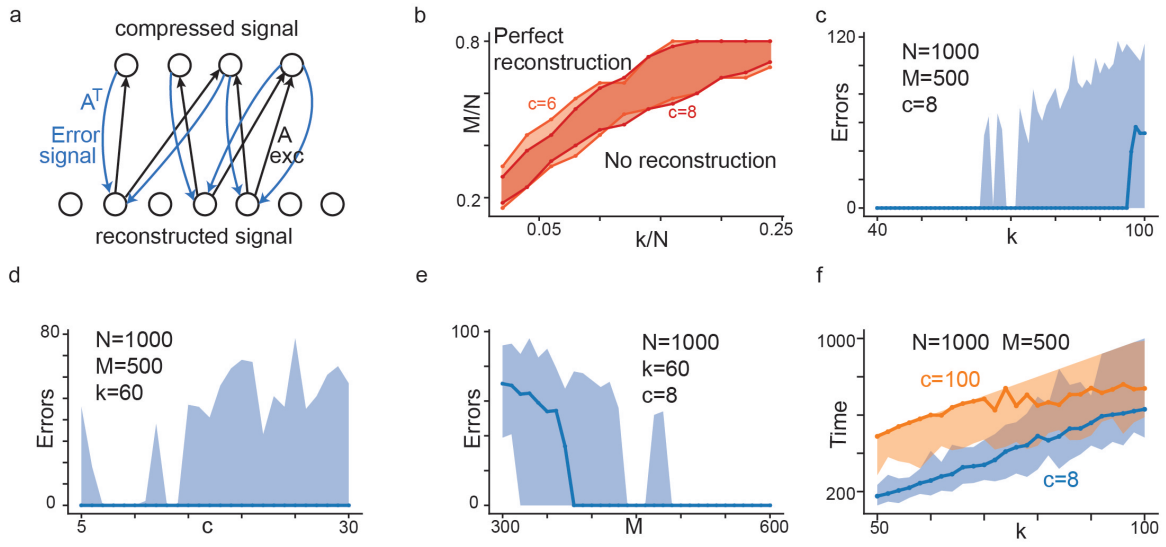

Figure 3.4: **Performance of the signed transpose algorithm for various combinations of the parameters** N, M, k, c**.** (a) Network used in the signed transpose algorithm. Original signal is represented in the bottom layer and compressed signal is represented in the top layer. The two layers are connected by a sparse binary random matrix with fixed column sums. For all the following results, N=1000. (b) Shaded region corresponds to the area in the $M - k$ plane where perfect recovery is possible for two different values of c. (c)-(e) correspond to various slices in the parameter space. The thick lines with the dots correspond to the median error/time for 50 runs. The shaded region corresponds to the spread in the error/convergence time for 50 runs. For each plot the values of the parameters N, M, k, c are shown on the plot. (c) $|\boldsymbol{x}_{rec} - \boldsymbol{x_0}|_1$ as a function of $|\boldsymbol{x_0}|_1$. (d) $|\boldsymbol{x}_{rec} - \boldsymbol{x_0}|_1$ as a function of left degree c of the left regular random graph used to connect the two layers. (e) $|\boldsymbol{x}_{rec} - \boldsymbol{x_0}|_1$ as a function of size of the compressed signal (M). (f) No. of time-steps required for the algorithm to converge as a function of sparsity (k) for two different values of c.

3.3.6 Details of simulations and results for signed transpose algorithm

Sparse binary vectors were created by selecting k random indices of a zero vector of size N and setting them to 1. Sparse random binary $M \times N$ matrices were created by selecting c indices from each column randomly and setting them to 1. For Fig.3.4, a measurement matrix A with parameters was created with parameters N, M, c in each run the signed transpose algorithm with measurement matrix A was used to reconstruct a random sparse binary vector with k ones. The algorithm was run 50 times.The shaded region corresponds to the area where the fraction of times there was perfect reconstruction was > 0 and ≤ 1. Fig. 3.4 c,d,e show slices in the parameter space. For each of these plots, one variable was varied keeping the other 3 fixed and the algorithm was run 50 times. The lines show the median errors and the shaded area shows the spread.

3.4 Compression and sparse dictionary learning as a model for communication and computation in the brain

3.4.1 Sparse coding

Sparse coding is the principle that sensory information or stimuli are encoded by a few active neurons in a population. We have been implicitly assuming that the populations we consider follow this principle so far, since each stimulus is represented as a sparse pattern of activity in our neural population. Experimental evidence for sparse coding has been found in several different sensory modalities in a variety of animals. V1 neurons in primates produce sparse responses when stimulated with naturalistic stimuli, i.e. stimuli resembling images that occur during natural vision [101]. De Weese et al. [102] have found that neurons in the auditory cortex of rats can produce a single spike in response to certain stimuli and the responses do not change across trials. Odor evoked responses

in the olfactory cortex are sparse [103].

Sparse coding confers several advantages in computation for a neural population [104]. It is useful in formation of associations and storing patterns in memory as we have already seen in the last section. Sparse codes are energy efficient since they require a few neurons to be active to represent a stimulus.

Sparse approximation Let $\boldsymbol{s} \in \mathbb{R}^M$ be a signal. Let $\mathcal{D} = \{\boldsymbol{\phi_1}, \cdots, \boldsymbol{\phi_N}\}$ be a set of N basis vectors with $N > M$. In this situation there are infinitely many solutions for the equation $\boldsymbol{s} = \sum_{i=1}^{N} a_i \boldsymbol{\phi_i}$ for $\boldsymbol{a}$. The optimal sparse approximation seeks to find $\boldsymbol{a}$ such that it has the smallest possible number of non-zero entries, thus solving the following optimization problem

$$\min_{\boldsymbol{a}} ||\boldsymbol{a}||_0 \text{ subject to } \boldsymbol{s} = \sum_{i=1}^{N} a_i \boldsymbol{\phi_i}. \tag{3.7}$$

where $|\boldsymbol{a}|_0$ represents ℓ_0 norm of $\boldsymbol{a}$ or the number of nonzero entries of $\boldsymbol{a}$. This problem is NP-hard [105]. The signal processing community has found a solution by replacing ℓ_0 norm by ℓ_1 [106] norm which makes it a convex optimization problem

$$\min_{\boldsymbol{a}} ||\boldsymbol{a}||_1 \text{ subject to } \boldsymbol{s} = \sum_{i=1}^{N} a_i \boldsymbol{\phi_i}. \tag{3.8}$$

It has been shown that this substitution leads to the same sparse approximation [107].

Olshausen and Field in their famous work [108, 109] have shown that natural images have sparse structure and the dictionary elements obtained from optimizing for sparsity resemble receptive fields of $V1$ neurons. They have also shown that the coefficients a_i obtained by them are statistically independent i.e. $P(\boldsymbol{a}) = \prod_i P(a_i)$, thus satisfying the efficient coding hypothesis. Note that the efficient coding hypothesis [110] does not imply sparse coding and sparse structure can only be obtained if it is present in the data as in natural images. Natural sounds have also been shown to have such a sparse decomposition [111].

Sparse expanded representations have been observed in granule cells of cerebellar cortex and *Drosophilla* mushroom body. The purpose of these systems is pattern discrimination and associative learning [76, 77]. The following papers have considered sparse

expanded formed by randomly mapping a dense pattern onto a large number of neurons. Ref. [78] have used clustered sensory stimuli and projections that incorporate the clustered structure of the inputs to map the sensory stimulus onto a large number of neurons. They have shown that these expanded representations reduce distance between intra-cluster patterns and increase distance between intercluster patterns thus aiding cluster categorization. [40] consider a similar set up in which the input patterns are mapped to much larger mixed layer and each neuron in the mixed layer receives a fixed number of input. The authors have shown that sparse connectivity between the input layer and mixed layer, i.e. low fixed synaptic degree leads to a high dimensionality of the mixed layer representations. The dimensionality of the mixed layer representations can be directly related to the performance of a perceptron classifier which associates each pattern to a valence ± 1 with equal probability. In these examples, the sparse expanded patterns are formed by random mapping from a small source area to a large target area to facilitate associative learning. This is different from expressing a set of vectors as a sparse combination of an overcomplete dictionary of basis vectors.

3.4.2 Locally competitive algorithms (LCA) for sparse dictionary learning

The network dynamics of a two layer neural network as shown in Fig 3.5 can be used to obtain a sparse approximation of a signal $\boldsymbol{s}$ [75]. The activity of the neurons in the first layer represent the signal $\boldsymbol{s}$. The equilibrium firing rates $\boldsymbol{a}$ of the neurons in the second layer gives the sparse approximation of $\boldsymbol{s}$. Let A be a $M \times N$ matrix whose columns are given by ϕ_i the elements of $\mathcal{D}$, $A = [\phi_1 \cdots \phi_N]$. Assume that the ϕ_i's are normalized to 1. In the set-up considered in [75], ϕ_i's can be considered to be the receptive fields of the neurons in the second layer. The neurons in the first layer and the second layer are connected by the matrix A^T, such that $(A^T)_{ij}$ represents the strength of connection between j-th neuron in layer one and i-th neuron in layer two. The neurons in the second layer have recurrent inhibitory connections given by the weight matrix $A^T A - I$. This inhibition induces competition between neurons having similar receptive fields which

would be given by large overlap between any two columns of A. The columns of A are normalized so that the diagonal elements of A^TA are one and subtracting the identity matrix I from this connectivity matrix ensures that there is no self inhibition. This competition allows a more active neuron i with receptive field $\boldsymbol{\phi}_i$ suppress a less active neuron j with similar receptive field $\boldsymbol{\phi}_j$, when the inner product $(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j)$ is high. For this reason, this algorithm is known as a locally competitive algorithm (LCA). The dynamics settles into an equilibrium where only a few neurons in the second layer are active thus building a sparse representation of the signal $\boldsymbol{s}$.

Let $\boldsymbol{u}(t)$ be the membrane potential of the neurons of the second layer and $\boldsymbol{a}(t)$ be the firing rates of the neurons in the second layer. The dynamics of the neurons are given by

$$\begin{aligned} \tau\frac{d\boldsymbol{u}(t)}{dt} &= -\boldsymbol{u} + A^T s - (A^TA - I)\boldsymbol{a}(t) \\ \boldsymbol{a}(t) &= T_\lambda\left(\boldsymbol{u}(t)\right) \end{aligned} \tag{3.9}$$

where the firing rate $\boldsymbol{a}(t)$ is given by a nonlinear function $T_\lambda(.)$ of the membrane potential $\boldsymbol{u}(t)$. In general T_λ is a thresholding function which is 0 below a threshold and equal to the input above the threshold. The most general form considered in [75] is given by

$$T_{\alpha,\gamma,\lambda}(u_m) = \frac{u_m - \alpha\lambda}{1 + e^{\gamma(u_m - \lambda)}} \tag{3.10}$$

In signal processing literature, $T_{0,\infty,\lambda} = lim_{\gamma\to\infty}T_{0,\gamma,\lambda}$ is known as the hard thresholding function which is the one we have used in our simulations.

The authors of Ref.[75] have also analyzed the stability of the equilibrium points of the above dynamics with two notions of stability. The first notion is the one usually considered in nonlinear dynamics, where a fixed point is defined as asymptotically stable if the system returns to the fixed point asymptotically if it is slightly perturbed. The second notion of stability is that the firing rates of the neurons do not grow exponentially. They have identified that the following criteria should hold for both notions of stability.

Criteria for stability Define $\mathcal{M}_{\boldsymbol{u}(t)}$ as the set of neurons in $\{1, \cdots N\}$ whose mem-

brane potential $u_m(t)$ is above the threshold, $\mathcal{M}_{\boldsymbol{u}(t)} = \{m : u_m(t) > \lambda\}$. Then LCA meets the *stability criterion* if the set of basis vectors $\{\boldsymbol{\phi}_m | m \in \mathcal{M}_{\boldsymbol{u(t)}}\}$ is linearly independent. It is interesting to note here that this stability criterion is met by matrices A which satisfies the restricted isometry Eq. (3.14), defined by [94] for measurement matrices in compressed sensing. This property implies that any set T of the columns of the matrix A with $|T| < S$ behaves approximately as an orthonormal system. Though the criteria for stability just demands linear independence which is weaker than demanding that the vectors be orthonormal. However, an orthonormal set of vectors is of course linearly independent.

3.4.3 Compressed sensing and sparse dictionary learning

We note here that compressed sensing and sparse approximation are essentially the same problem if the signal $\boldsymbol{s}$ is obtained by compressing a sparse signal $\boldsymbol{x}$ using a measurement matrix A.

$$\boldsymbol{s} = A\boldsymbol{x} \tag{3.11}$$

The compressed sensing problem is deterministic in the sense that it solves to find the particular sparse $\boldsymbol{x}$ from $\boldsymbol{s}$ rather than finding any sparse approximation of $\boldsymbol{s}$. Thus, it is not surprising that the stability criterion for LCA matches the restricted isometry property of compressed sensing. Since there has been considerable research on the classes of matrices A that satisfy the restricted isometry property, this tells us the classes of matrices A that can be used for LCA. It would be interesting what these A's tell us when viewed as collection of basis vectors and if any of them make sense as receptive fields of neurons. We are still trying to uncover if there is a deeper mathematical reason for why the stability criteria for LCA and the restricted isometry property from compressed sensing turns out to be the same.

We want to note another similarity here between LCA and the signed transpose algorithm proposed earlier. In LCA, while using hard thresholding when $\boldsymbol{u}(t) > \lambda$ the RHS of (3.9) can be expressed as $A^T(\boldsymbol{s} - A\boldsymbol{u}(t)) = A^T\boldsymbol{gap}$ where $\boldsymbol{gap} = \boldsymbol{s} - A\boldsymbol{u}(t)$. In the signed transpose algorithm we calculate the same quantity and apply a thresholding

function on $A^T\boldsymbol{gap}$ while updating neurons in the first layer. Note that the layer structure in the signed transpose and LCA are switched, the first layer or the layer containing the sparse signal in the signed transpose set-up is the second layer or the layer constructing the sparse approximation in the LCA set-up. Therefore the layer reconstructing the sparse signal in the signed transpose set-up receives a similar input as received by the layer constructing the sparse approximation in LCA. The main difference between the two algorithms is this: in the signed transpose the threshold is applied afterwards and neurons are updated asynchronously (at each time step a neuron is chosen at random and updated). We are currently exploring the relationship between applying the non-linearity (the thresholding function) at different points and the relationship between the synchronous and asynchronous dynamics.

3.4.4 LCA using expander graphs

We implemented the LCA algorithm with the adjacency matrix of a bipartite random left regular graph as the connectivity matrix A. We used this algorithm to perform compressed sensing in a way described above. A sparse binary signal $\boldsymbol{x}$ was compressed to form the signal $\boldsymbol{s} = A\boldsymbol{x}$. Then the algorithm was run on $\boldsymbol{s}$ to find a sparse approximation $\boldsymbol{a}$ in the (3.9). The performance of the algorithm is measured by $\boldsymbol{error} = |\boldsymbol{x} - \boldsymbol{a}|_1$. Fig. 3.5b shows the region in the $M - k$ plane where $|\boldsymbol{error}|_1 < 0.1$.

[81] shows that random rectangular zero-one matrices with fixed row and column sums are full rank with high probability. We think this result holds for random binary matrices with fixed column sums that we consider as well which is something we are trying to prove. We think the other results in this paper might help us to prove bounds on the compressed sensing algorithms we have considered which is also something we would like to explore. We also note that the adjacency matrices of random regular bipartite graph has the $RIP-1$ property as discussed in Section 6.2. This property can be used to upper bound the angle between subspaces spanned by disjoint subsets of columns of A as shown in SI analogous to matrices with $RIP-2$ property [94].

The columns of a random binary $M \times N$ matrix with a fixed number of ones probably

do not make sense as receptive fields of neurons. We are exploring the possibility that there might be a transformation taking overcomplete sets of visual or auditory receptive fields to a matrix with few ones in each column. Even if such a transformation does not exist, random regular graphs might be a connectivity strategy used by the brain. Since these connections are sparse, they reduce wiring costs and uses fewer neurons to perform a computation. Thus they are energy efficient. As discussed here they can also be used to construct sparse expanded representations using the LCA algorithm.

3.4.5 Details of LCA simulations and plots

Sparse random binary vectors $\boldsymbol{x}$ of length N with sparsity k were created by randomly choosing k indices of a length N 0-vector and setting them to 1. The $M \times N$ adjacency matrix A of sparse random left regular graphs were created in a similar manner, c indices from M were chosen randomly in each column and set to 1. All the columns were normalized to 1. In the case of random Gaussian matrix, each entry was drawn from $\mathcal{N}(0, 1)$ The sparse binary random vectors $\boldsymbol{x}$ were compressed using A to obtain $\boldsymbol{s} = A\boldsymbol{x}$. $\boldsymbol{s}$ serves as the input to the LCA network. The equations (3.9) were integrated using the Euler method, with $\tau = 0.1, dt = 1, \lambda = 0.5$. We assumed that the algorithm converges when $|\boldsymbol{a}(t) - \boldsymbol{a}(t - dt)|_{\ell_2} < 10^{-4}$. For Fig. 3.5b, a matrix A is created for each M. The LCA is run 50 times for different random binary vectors $\boldsymbol{x}$ which produce different compressed vectors $\boldsymbol{s}$. Errors are computed using the ℓ_1 norm of the difference $|\boldsymbol{a}(t) - \boldsymbol{x}|_{\ell_1}$. The shaded region in Fig. 3.5 is the area where the $|\boldsymbol{error}|_1 < 0.1$ in at least 1 run. Errors are plotted against k, c, M in Fig. 3c,e,f respectively. These plots show that sparse random binary matrices as connectivity perform better than random Gaussian matrices. Fig. 3.5 shows that the algorithm converges faster when A is a sparse binary random matrix.

3.4.6 Compression as a form of communication between brain regions and re-expansion for computation

We consider communication between spatially localized brain regions as in [74]. As discussed in the introduction, the number of connections between neurons fall-off with distance [9]. All neurons in a source region do not send axonal projections to all neurons in a target region [112]. Thus the representation in the source region undergoes some sort of compression before being transmitted and the target region does not know how it was compressed. The problem considered in [74] is how does the target region make sense of this subsampled information incoming from the source region. They propose that the representations are expanded into sparse representations on which computations can now be performed akin to expanded representations considered in [75, 78].

We consider a slightly different set-up where we assume that the source region is larger than the target region, thus information has to be compressed while being transmitted. Furthermore, each neuron in the source region is connected to a few neurons in the target region thus the connectivity between the two regions have the sparse expander structure. In Fig 1, we have shown that information is preserved during such kind of communication.

In this section, we explore if the compressed patterns in the target region can be re-expanded to perform any kind of computation. In particular, we consider the pattern separation problem as in [40]. Each pattern in the source area is associated with a valence ± 1 with equal probability. The compressed pattern in target region is re-expanded using the LCA algorithm. The A' used in the LCA algorithm is not the same as the one used for communication, thus the re-expanded pattern is different than the original pattern. A classifier is trained on the re-expanded pattern, to associate them with ± 1 which were associated with the original patterns. The performance of this classifier is shown in Fig. 3.6 bottom panels. For the simulations given above, the classifier can associate the patterns with their given valences perfectly. More details of the simulations are given in the methods section below.

We check if the same process can be repeated multiple times. So we consider another random association of the patterns with valences ± 1. The expanded pattern in target

region 1 is compressed while being communicated to target region 2. It is re-expanded to form a sparse representation in target region two. A classifier is then trained on this re-expanded pattern to associate them with the second set of valences. The performance of this second classification is also shown in Fig. 3.6 bottom right panel. This shows that this method of compression and re-expansion can be repeated multiple times, and relevant information can be extracted from the re-expanded patterns in different brain regions.

We have used bipartite expander connections to model the projections between distant brain regions as well as to model connectivity within a region to construct sparse expanded representations. We have shown that these connections perform extremely well, both when the patterns are compressed while being transmitted as well as for the LCA algorithm which can be used to form sparse expanded representations with multiple generic neural populations.

3.4.7 Details of simulations and plots

For the simulation results shown in Fig 3.6 we used the following region sizes. A_{comp_1} : $(500, 1000)$, $A_{exp_1} : (1200, 500)$, $A_{comp_2} : (600, 1200)$, $A_{exp_2} : (1200, 600)$. All the matrices were constructed as random binary matrices with c 1s in each column. The LCA was run as described above. For each sparsity, we took 100 patterns randomly associated with ± 1. We trained a perceptron like classifier with weights $\boldsymbol{W}$ and bias b. The weights W and the bias b were trained to minimize the loss $tanh(W.\boldsymbol{a} + b) - valence(\boldsymbol{x})$, where $\boldsymbol{a}$ is the expanded pattern corresponding to $\boldsymbol{x}$ and $valence(\boldsymbol{x})$ is the sign associated with $\boldsymbol{x}$. The classifier was trained in tensorflow using the Adam optimizer to minimise the loss function. As can be seen from the bottom panels of Fig 3.6 re-expanded patterns in each region could be separated perfectly.

3.4.8 Future extensions of the communication model

We are planning to consider different compression and expansion ratios to evaluate the performance. We are also planning to use dense matrices for compression and re-

expansion to compare the performance of these matrices with sparse connectivity matrices.

3.5 Discussion

We considered distant brain regions interacting via long range sparse connections. We modeled these connections with expander graphs and explored the consequences of sparse expander connections on a few computations that these regions might perform. We found that for regions connected with feedforward convergent connections and reciprocal feedback divergent connections, sparse connectivity gives the network more flexibility. Networks with sparse connectivity can maintain or reconstruct patterns with greater number of non-zero elements without error than networks with dense connectivity. We also found that sparse connectivity leads to faster convergence in the dynamics. We have already discussed the future directions we would consider for this model in Section 3.2. We would like to make this model more biophysically plausible by including excitation-inhibition balance, adding multiple areas, having neurons with any positive firing rates and let network mechanisms stabilize the dynamics.

In addition to this, We believe that this network can be used for associative learning. For example, certain experiments have observed that in a task where the subject has to identify an instrument by its sound or its image, neurons in the visual cortex are also activated when a sound is played and neurons in the auditory cortex is activated when an image is shown [113, 114]. We think such models where activity in one area is associated with activity in one area can activate activity in another area can be modeled by the two layer reciprocal network we considered. The first layer can be divided into multiple areas which project to an integrative region such as the association cortex. Then the activity in one area can activate associated patterns in another area through the divergent feedback connections from the association cortex to the early sensory cortices [73]. Recently Steinberg and Sompolinsky used a network with the same structure for associative learning [115].

We also considered a different architecture where the patterns compressed from the source region to target region are re-expanded to form sparse representations. Such sparse re-expansion could either be a sparse approximation of a vector [104] or a sparse expansion for pattern separation [78]. An interesting question is if optimising for one of these gives the other too. In particular if expanded representation is built such that the over complete dictionary is optimised to give the sparsest representations for each vector will this also optimize the pattern separation performance? Different regions in the brain might build sparse representations for different purposes. For example, in the visual cortex the brain might want to represent an image as a sparse combination of edges so the random connectivity will not work whereas in other areas it might just want a sparse expanded representation for pattern separation where the random connectivity works perfectly well. The LCA algorithm suggests that the same algorithm can be used to build sparse representations irrespective of the purpose.

We want to note that we did not consider optimising the connectivity (dictionary elements) for building the sparse representation. Instead we just considered sparse expander connectivity for the LCA network. We used LCA for the compressed sensing problem as discussed in Section 4.3 . We found that the sparse expander connection indeed perform well to solve this compressed sensing problem. The problems of sparse approximation and compressed sensing are related and have drawn inspiration from each other for their solutions. Overcomplete dictionaries that optimize sparse representations for different data sets are different. For example, the overcomplete dictionary for natural images is different from the overcomplete dictionary for natural sounds. From our simulations we find $M \times N$ random binary matrices with fixed column sums or $M \times N$ random Gaussian matrices serve as good connectivity matrices for LCA for finding sparse approximations of any compressed patterns with positive entries. We would like to check if specific over complete dictionaries optimized for specific datasets can build better sparse representations. We would also like to explore if the transformations relating specific over complete dictionaries to random matrices make sense.

In this work we have drawn on different areas in applied mathematics such as graph

theory, compressed sensing, sparse approximations and dynamical systems to model and study two computations the brain might perform. We hope that these ideas can be extended to study specific brain regions and their computations, for example the interactions of the prefrontal cortex with other areas to model cognitive control. We also hope that brain connectivity maps will help us check is sparse expander connectivity indeed exists in the brain which would validate using such connectivity and methods to study brain functions.

3.6 Appendix

3.6.1 Compressed sensing

Let $x \in \mathbb{R}^N$ be a signal and A be a $M \times N$ measurement matrix, with $M < N$. The measured vector $y \in \mathbb{R}^M$ is

$$y = Ax \tag{3.12}$$

Can x be reconstructed with knowledge of A and y. In general this is not a well-posed problem since the number of unknowns(N) are more than the number of equations(M).

Now suppose that x is sparse, ie it has only K nonzero components with $K << N$. In this case, Candes et. al. [94, 98] proved that x can be recovered by solving the optimisation problem

$$\min||x||_{l_1} \quad \text{subject to} \quad y = Ax \tag{3.13}$$

if A satisfies certain properties. Let A_T, $T \subset \{1, \cdots, N\}$, be the $n \times |T|$ matrix which extracts the columns in A corresponding to the indices in T. Then [94] defines the restricted isometry constants δ_S as the smallest quantities such that

$$(1 - \delta_S)||x||_2^2 \leq ||Ax||_2^2 \leq (1 + \delta_S)||x||_2^2 \tag{3.14}$$

for all subsets T with $|T| < S$. This property requires that any set of columns of A with cardinality less than S behaves like an orthonormal system. The reconstruction is exact

when $\delta_{3S} + 3\delta_{4S} < 2$.

For matrices A whose entries are random i.i.d Gaussian with mean 0 and variance $1/N$, the above condition is satisfied with overwhelming probability when [116–118]

$$S < C.N/log(M/N) \tag{3.15}$$

3.6.2 Theorems on expander graphs

For a subset S of the left nodes we define $N_{unique}(S) \subseteq N(S)$ to be the subset of $N(S)$ which receive only one edge from S. In other words, the right nodes in $N_{unique}(S)$ are unique neighbors of left nodes in S.

Theorem 3.6.1. *For any subset S of left nodes $|S| \leq \alpha N$, $|N_{unique}(S)| \geq (1 - 2\epsilon)c|S|$.*

Proof. Let us call the nodes in $N(S)$ which receive more than 1 edge $N{>}1(S)$. From the expansion property of expander graphs we have $|N_{>1}(S)| + |N_{unique}(S)| > (1-\epsilon)c|S|$. The total number of edges coming out of S, $c|S|$. Each node in $N_{>1}(S)$ must receive atleast 2 edges, so counting the number of edges coming out of S we have $|N_{unique}(S)| + 2|N_{>1}(S)| \leq c|S|$. So, $|N_{>1}(S)| \leq \frac{1}{2}(c|S| - N_{unique}(S))$. Substituting the maximum value possible for $|N_{>1}(S)|$ in the above inequality, we get $|N_{unique}(S)| \geq (1 - 2\epsilon)|S|$. □

Definition 3.6.1. *An $M \times N$ matrix A is said to satisfy $RIP_{p,k,\delta}$ if, for any k-sparse vector x, we have*

$$|x|_{\ell_p} \leq ||Ax||_{\ell_p} \leq (1 + \delta)||x||_{\ell_p} \tag{3.16}$$

Theorem 3.6.2. *Consider any $M \times N$ matrix,A that is an adjacency matrix of a bipartite expander graph with parameters (c, α, ϵ). The scaled matrix $A/c^{(1/p)}$ satisfies the $RIP_{(p,\alpha N,\delta)}$ with $\delta = C\epsilon$ for some absolute constant C.*

Theorem 3.6.3. *Consider a $M \times N$ matrix, A that has c ones in each column. If for some scaling factor $S > 0$, S A satisfies the $RIP_{(1,k,\delta)}$ property then A is an adjacency matrix of an expander graph with parameters $(k/N, \epsilon)$ where*

$$\epsilon = \left(1 - \frac{1}{1+\delta}\right) / (2 - \sqrt{2}) \tag{3.17}$$

For the proofs of these two theorems see [99].

3.6.3 Compressed sensing using expander graphs

Bit flip algorithm

Expander graphs have been used to build error correcting codes in the famous work by [96]. In their set up, the left nodes of a bipartite expander graph represents a binary message x and the right nodes represent constraints y. The constraint equations are given by $Ax = y(mod2) = 0$ for correct messages, where A is the adjacency matrix of an expander graph with parameters $(\alpha, c, 1/4)$. For a corrupt message x', some of the constraint nodes are going to be unsatisfied (there values will be 1). The above paper shows that it possible to correct a message with less than $\alpha N/2$ errors with their algorithm. We modify their arguments for recovery (compressed sensing of sparse binary signals).

In the compressed sensing set-up, the left nodes represent the binary signal x, the adjacency matrix A of the expander graph connecting the left nodes and the right nodes represents the measurement matrix and the right nodes $y = Ax\ (mod2)$ represents the measured vector. Compressed sensing asks if we can recover x once we know A and y. Note that in this case, unlike the error correcting code case, $y = Ax \neq 0$, but we know its components.

The algorithm is given by:

1. Start with $\hat{x} = 0$.

2. Let $\hat{y} = A\hat{x}$. Unsatisfied constraints in the compressed sensing set up are given by the components where $\hat{y} \neq y$.

3. Find the components of $\hat{x}$ connected to $> c/2$ unsatisfied constraints. If no such component of $\hat{x}$ exist, then terminate. Otherwise pick a random node which is connected to $> c/2$ unsatisfied constraints and flip its value.

Note that since we always flip a node which is connected to more unsatisfied constraints than satisfied constraints, the number of unsatisfied constraints decreases.

Theorem 3.6.4. *If $|x - \hat{x}|_{l_0} \leq \alpha N$ we can always find some component to flip. If $|x|_{l_0} \leq \alpha N/2$ we can recover x where A is an adjacency matrix for an expander with $(\alpha, c, 1/4)$.*

Proof. Let S be the set of components where $x - \hat{x} \neq 0$. If $\hat{y}$ is a unique neighbor of an element in S it is unsatisfied by the definition of an unsatisfied constraint. We have shown that for a subset S of left nodes with $|S| \leq \alpha N$, the set of unique neighbors of S, $N_{unique}S$ satisfies $|N_{unique}(S)| > c|S|/2$ for an expander with the above parameters. Each node in S has c neighbors. If $\leq c/2$ of these neighbors for each node are unique, then the number of unique neighbors of S would be $\leq c|S|/2$, which is a contradiction.Therefore there is at least one node in S with $> c/2$ unique neighbors. This implies we can find a left node $\hat{x}_i$ which is connected to $> c/2$ right nodes where $y \neq \hat{y}$. So we can always find a node to flip.

Since $|x|_{l_0} \leq \alpha N/2$, the maximum number of unsatisfied constraints that the nodes in the S it can be connected to is $c\alpha N/2$. We have shown above that we can always find a node to flip if $|S| < \alpha N$. Thus the algorithm fails if $|S| \geq \alpha N$ during the algorithm. For this to happen, there must be a step when $|x - \hat{x}|_{l_0} = \alpha N$.At this step, $|N_{unique}(S)| > c|S|/2 = c\alpha n/2$. This is a contradiction since the maximum number of unsatisfied constraints possible is $c\alpha N/2$ and the algorithm always decreases the number of unsatisfied constraints. Therefore, this algorithm can recover x whenever $|x| < \alpha N/2$. □

Greedy gap algorithm

The greedy gap algorithm also uses the adjacency matrix of an expander graph as the measurement matrix and relies on the number of unique neighbors of the left nodes but its update rule is a bit different from the bit flip algorithm. We can only recover a binary signal using the bit flip algorithm but we can recover a sparse signal with arbitrary components using the greedy gap algorithm. [100],[97] We assume that $|x|_{l_0} \leq \alpha N/2$.

Like before, we start with a estimate vector $\hat{x}$. We define a gap vector

$$g = y - A\hat{x} \tag{3.18}$$

The algorithm is as follows:

1. Start with $\hat{x} = 0_{1\times N}$.

2. If $g = 0$, declare that $\hat{x}$ is the solution and terminate. If not, then we can find i such that $\hat{x}_i$ is connected to $> c/2$ identical gaps g.

3. Set $\hat{x}_i = \hat{x}_i + g$. Go to 2.

Since the $|x|_{l_0} < \alpha N/2$, we know from 3.6.1 that there must be at least one component of $\hat{x}$ that is connected to more than $c/2$ unique neighbors for an expander $(\alpha, c, 1/4)$. The value of the gap g for all these unique neighbors are identical. Therefore one can always find a component to update at the first step.

Note that the algorithm always reduces the number of nonzero components of the gap vector. The number of nonzero gaps at the beginning can be at most $c\alpha N/2$. The algorithm will fail if at some time $|x-\hat{x}|_{l_0} > \alpha N$. If this has to happen, then at some time $|x - \hat{x}|_{l_0} = \alpha N$. At this time according to 3.6.1 the number of unique neighbors of these components is $> c\alpha N/2$. The gaps which are unique neighbors of the components of $|x-\hat{x}|_{l_0}$ are also non zero. So the number of non zero components of gap becomes greater than $c\alpha N/2$. This is a contradiction since the algorithm always reduces the number of nonzero components of the gap vector. Therefore using this algorithm we can recover any signal x with $|x|_{l_0} < \alpha N/2$.

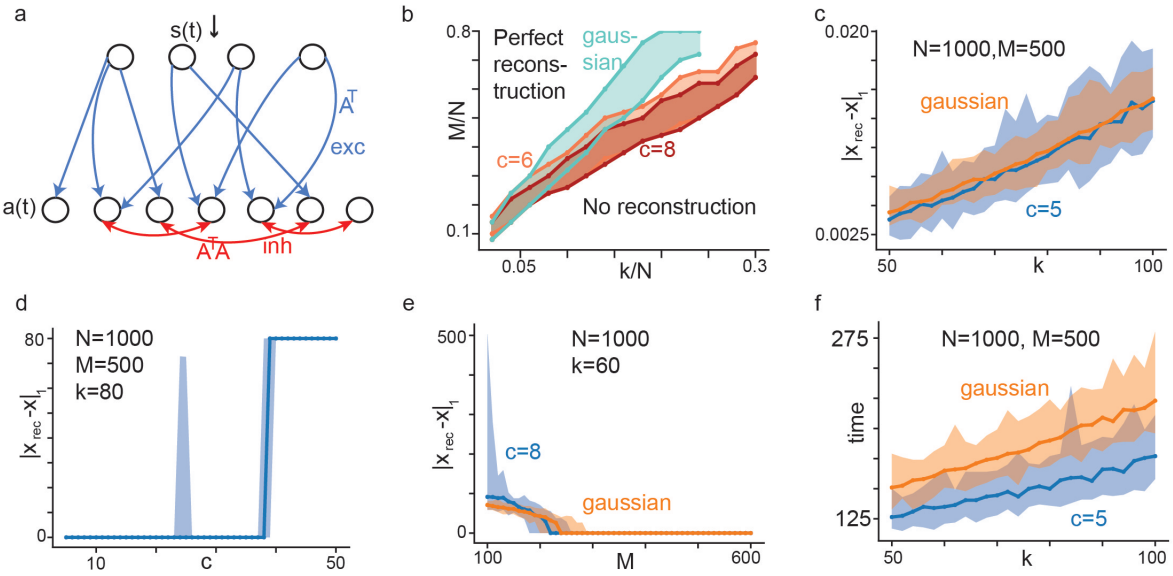

Figure 3.5: **LCA for compressed sensing using adjacency matrix of bipartite expander graphs as** A (a) The LCA neural network as considered in [75]. The first layer represents the sigal $\boldsymbol{s}(t)$ and the sparse approximation is given by firing rates $\boldsymbol{a}(t)$ of second layer. (b) Shaded region in the $M-k$ plane shows the area where $|\boldsymbol{a}-\boldsymbol{x}|_1 < 0.1$ for at least 1 run out of 50 runs.(See details in Section 3.4.5). The orange and light red areas correspond to LCA with random left regular graph with left-degrees 6 and 8 respectively. The turquoise area corresponds to LCA with a matrix where each entry is iid Gaussian drawn from $\mathcal{N}(0,1)$. The sparse random matrix performs better than the Gaussian random matrix. The next 3 plots show errors verses the various parameters in the model. Blue lines and shaded regions are from LCA using random regular graph with left degree c. Orange lines and areas are from LCA with Gaussian random matrix with i.i.d entries $\mathcal{N}(0,1)$ (c) Reconstruction error verses sparsity of original vector. Thick lines represent median and shaded area represents spread of the error over 50 runs (d) Reconstruction error verses left degree of random left regular graph used in LCA. (e) Reconstruction error vs size of compressed signal(M) that is input into the LCA algorithm. (f) Number of timesteps the algorithm required to converge verses the sparsity of the original vector. LCA with Gaussian random matrices require more time than adjacency matrix of random graphs with regular left degree.

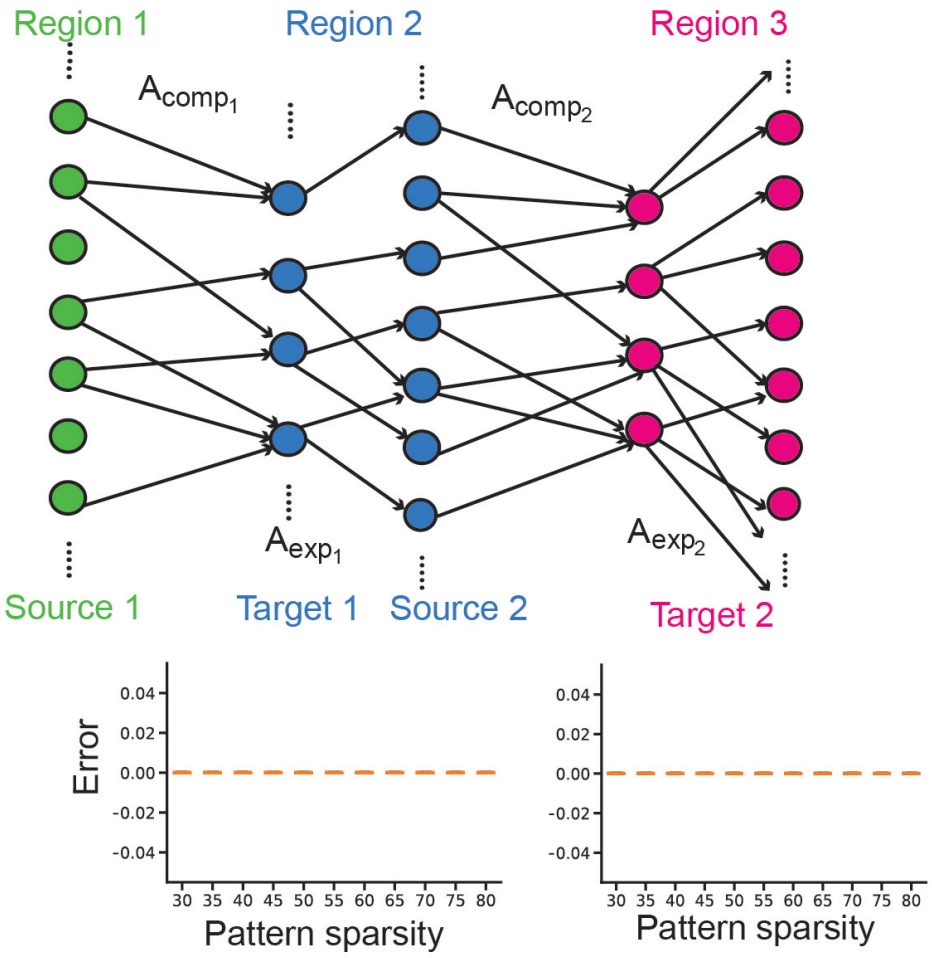

Figure 3.6: **Model of communication between spatially localized brain regions as in [74].** Neural representations in source region 1 are compressed while being transmitted to target region 2 using A_{comp_1}. They are re-expanded in region 2 using LCA with connectivity A_{exp_1}. The recurrent inhibitory connections required for LCA are not shown in the figure. Each source pattern is associated with ± 1 with equal probability. A classifier is trained to discriminate the patterns according to the associations. The performance of this classifier is shown below region 1 (left). The classifier network is also not shown in the figure. The exppanded pattern in region 2 is compressed while being transmitted to region 3 using A_{comp_2}. This pattern is re-expanded using LCA in region 3 with A_{exp_2}. The classifier is now trained to learn a different set of associations ± 1. The performance of this classifier is shown below region 3 (right). For more details of region sizes and simulations see Section 3.4.7 .

Chapter 4

Biological neural network for the localisation of sound

4.1 Introduction

The interaction of sound with the outer ear (pinna) modifies the energy in certain frequency bands in the spectrum of a broadband noise in a way dependent on the location of the sound in the vertical plane.The function mapping the original sound spectrum to the modified one after interaction with the outer ear is known as the head related transfer function (HRTF) [119–121]. More specifically, the interaction of the sound with the outer ear produces notches (frequency bands where energy is decreased) in the spectrum; the frequency and magnitude of these notches are a function of the angle of elevation of the sound source.

There is evidence that there are neurons in the auditory pathway dedicated to the processing of these notches. These neurons are located in the dorsal cochlear nucleus (DCN) and the inferior colliculus (IC) and are sensitive to the frequency and the magnitude of the notches. Therefore they are believed to be important in the localisation of sound in the vertical plane.

Fig 4.1 shows the neural circuit that has been proposed to describe the response of the type 4 neurons of the DCN and the type O neurons of the IC [122], [123]. In this paper,

we build a firing rate model at steady state for the neural circuit proposed above. This is a feed forward neural network with each layer consisting of the neurons which are relevant for the neural circuit which process these notches. Our model describes how neurons up to the pre-cortex area are tuned to spectral notches essential for the localisation of sound in the vertical plane.

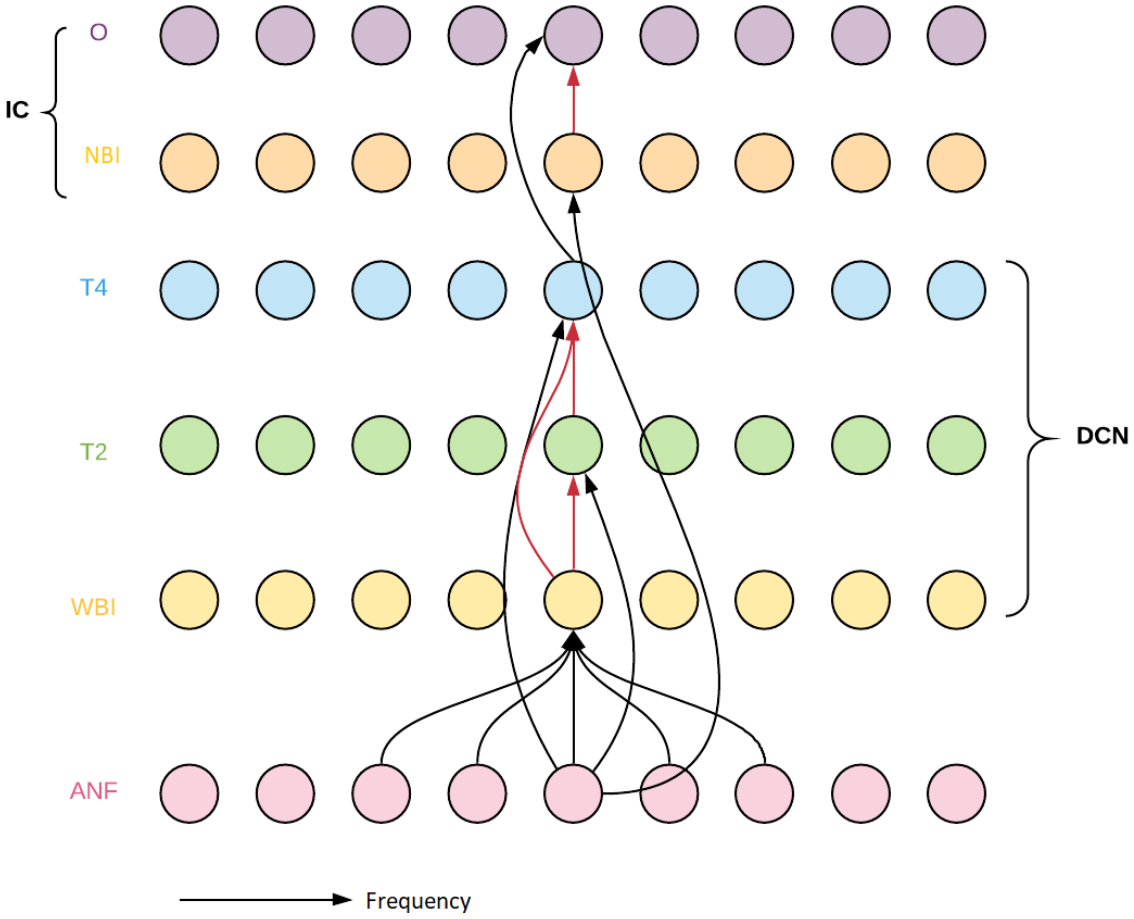

Figure 4.1: **Model Schematics.** Schematic of feedforward neural network with 6 layers. Each layer is labelled on the left. Black arrows represent excitatory connections and red arrows represent inhibitory connections. The arrows only represent the connections, the actual weight matrices are given in 4.5.

We consider the response of the relevant neurons to three kinds of stimuli that have been experimentally recorded : pure tones, broad band noise and notched noise which is a simulation of the notch produced by the HRTF on the spectrum of the sound. The different stimuli that we will be considering for our model are: a) pure tone at a Best Frequency (BF), b) pure tone frequencies swept across the entire range of BFs at

different intensity levels, c) broadband noise at different spectrum levels d) Notched noise at different spectral intensity levels, e) notch center frequencies swept across the entire range. The best frequency(BF) of a neuron is defined as the frequency of pure tone for which the neuron responds above its spontaneous rate at the lowest intensity. For a broad band noise we consider a white noise with equal power in each of the frequency components with a certain center and width. We are of course most interested in the notched noise response of the type 4 neurons and the type O neurons since these are sensitive to the frequency of the notches. The complex response properties of these neurons to the above mentioned stimuli show the non trivial integration properties of these neurons which makes the modelling task a challenge.

We consider a firing rate model at steady state. One of the reasons for this is that the experimental results that we compare our results with are rate vs level functions for the different neurons or rate vs best frequency of the neurons. Since we do not have experimental data on rate as a temporal function for the different neurons we are considering, we forego it at our first pass for modelling this circuit. The second reason for considering a model at steady state is because we are interested in the interactions between the different populations of neurons in our model and how it gives rise to complex response properties.

Our motivation in developing this model is to see how the localization network responds when damaged by a neurodegenerative disease such as Alzheimer's, which is known to impair the ability of afflicted individuals to localize sound, particularly in crowded environments[124]. Moreover, there is substantial evidence to suggest that the auditory brain stem itself is directly impacted by Alzheimer's driven neurodegeneration[125].

We show that our model is able to reproduce the responses of the neurons recorded in the DCN and the IC at least qualitatively. Though we are not able to capture all the features of the experimentally recorded neurons, we are able to reproduce the features that are important in the localisation of sound. In section 2, we describe our model in more details including the behaviour of the neurons in the different layers and the connections between the layers. In section 3, we describe the successes and the shortcomings of our

model. In section 4, we discuss the novel features of our model.

We have built a rate based population model to explain the responses of the type 4 neurons and the type O neurons. Our model architecture has been inspired by the model by Blum and Reed [126],[127]. However, Blum and Reed did not construct a model with neurons logarithmically arranged in frequency as the auditory fibers are known to be. Their neurons were also not correlated with observed best frequencies. Additionally they did not consider responses to notched noise sweeps and their model did not extend to the type O neurons which have been discovered more recently. The arrangement of the neurons according to their BFs and the integration of inputs over some range of frequencies around the BF was inspired by the models of Hancock and Voight [128],[129]. However, the models of Hancock and Voight are spiking models and hence it is more difficult to simulate and study the effects of integration of the inputs between different kinds of neurons. Our model is different from that of [130] in that it is a population model and involves interactions between populations of neurons and has more realistic weight functions connecting populations of neurons.

4.2 The Model

The model is a feed forward neural network consisting of six layers of neurons. Each layer has a hundred neurons that are tonotopically arranged, ie, they are arranged according to the frequency to which they are most sensitive. In our model, we have the BFs of the neurons spanning four octaves, starting from 1.25 kHz and going up to 20 kHz with 25 neurons in each octave. So the BF of each successive neuron is $2^{0.04}$ times the BF of the previous neuron. This is a reasonable hearing range for humans. We refer to all the neurons below by their BFs.

It is experimentally known that the tonotopic arrangement of the neurons are preserved in the higher areas of the auditory pathway, so we have arranged our model in this way, which was also done in [129] and [126]. However in [126], the authors did not map their frequencies into the actual range of hearing frequencies like we do. Our model

is also easily scalable in the number of neurons. Increasing the number of neurons will increase the number of neurons per octave and decrease the difference in the BFs.

The first layer consists of the auditory nerve fibers which receive all auditory stimuli. The auditory nerve fibers innervate the neurons in the DCN which is the first relay station in the auditory pathway. The neurons in the second, third and the fourth layer in our model lie in the DCN. The neurons in the DCN project to neurons in the IC. The neurons in the fifth and sixth layer of our model lie in the IC. We will describe the neurons in each of these layers and their experimentally recorded response properties in details below.

Auditory Nerve Fibers

In this section, we define the rate function to model the response of an auditory nerve fiber for pure tone stimulus based on its receptive field. Then we generalise it to a response for more complicated stimulus like broad band noise. The auditory nerve fibers innervate the hair cells which are present on the basilar membrane. The basilar membrane is a thin coiled membrane with a gradient in tension and thickness along its axis. So the different regions of the basilar membrane have different resonant frequencies. The auditory nerve fibers connected to a particular region the basilar membrane is most sensitive to the resonant frequency of that region. The auditory nerve fibers are tonotopically arranged because they connect to different regions of the basilar membrane.

In neuroscience literature [131], the rate of a neuron as a function of a stimulus parameter s, $r = f(s)$, is defined as a tuning curve. By this definition, (4.1) would be called the tuning curve of the ANFs for a pure tone. The tuning curve of an ANF is defined as the minimum intensity as a function of pure tone frequency that elicits a response from the ANF above its spontaneous rate. We first define the rate of an ANF in the first sense, ie as a function of stimulus parameters,the frequency of a pure tone f_{PT},

and the intensity of a pure tone s.

$$\begin{aligned} r_{ANF}(f_{BF}, f_{PT}, s) =& \theta(s - s_{th}) \times h(s - s_{th}) \times exp\left(-\left(\frac{\alpha(f_{BF} - f_{PT})}{s}\right)^2\right), \\ h(x) =& \frac{ax^2}{x^2 + b}, \\ \theta(x) =& \begin{cases} 1, & x \geq 0, \\ 0, & x < 0. \end{cases} \end{aligned} \tag{4.1}$$

This equation is inspired by tuning curves for other neurons found in the literature. This is the first such function to be defined for an ANF to our knowledge. All other models have only considered rate as a function of intensity, which is the second factor on the right side of (4.1). This was modelled based on the observation that as the intensity increases, the frequency range over which the pure tone elicits a response from an ANF also increases.The first figure in Fig 4.2 , shows the rate of an ANF as a function of the frequency of pure tones at different intensities according to (4.1). The second figure in Fig 4.2 , shows the saturating function $h(x)$ for different parameters. Table 4.1 gives the values of the parameters in (4.1) and their description.

Table 4.1: **Parameters in** (4.1)

Parameters	Function of the Parameter
s_{th}	Threshold for ANF (20 dB)
a	Determines the value of saturation of firing rate (200)
b	Determines the slope of the saturating function h
α	Determines the width of the tuning curve Fig 4.2 (a)

On the left of Fig 4.3 we have the tuning curves obtained experimentally. On the right of Fig 4.3 we have the tuning curves obtained from (4.1). To get the tuning curves $s(f_{BF}, f_{PT})$ from (4.1), we set $r_{ANF}(s, f_{PT}, f_{BF}) = s_{th}$. Since the function (4.1) is symmetric about f_{BF} for the pure tone frequencies we do not get the long tail extending to the lower frequencies at higher intensities as is seen experimentally. We would need to modify (4.1) to include this feature.

We model a broadband noise to have a constant flat spectrum over a range of frequencies. We assume that the response of an ANF to such a stimuli is a saturating function

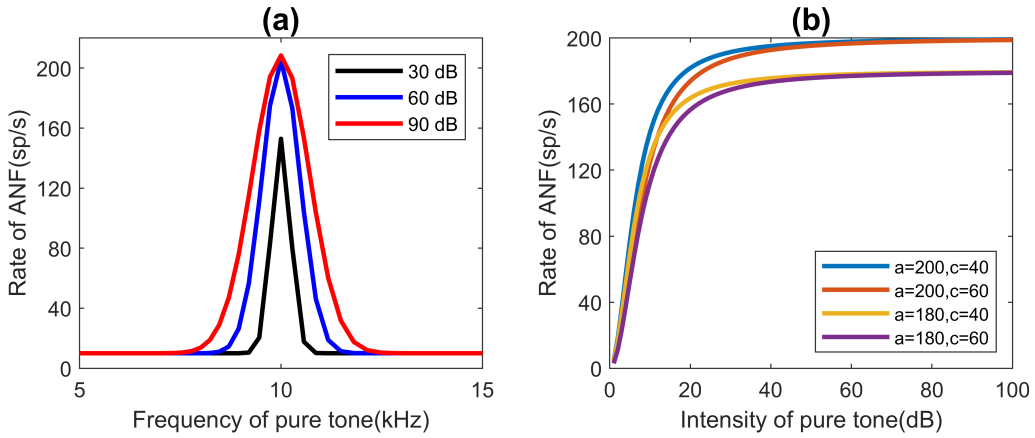

Figure 4.2: **Response of Auditory Nerve Fibers (ANF) as a function of frequency and intensity.** a) Rate vs frequency of pure tones for an ANF(BF=10 kHz) at different intensities ($\alpha = 90$) obtained from (4.1). b) The saturating function with different parameters.

of the power in the frequency component corresponding to the BF of the ANF. So the ANFs whose BFs lie under the nonzero power spectrum of the noise respond to the noise and the others do not.The first figure in Fig 4.4 shows the noise stimulus and the second one shows the response of the ANFs to the broadband noise on the left. Similarly Fig 4.5 shows a notched noise stimulus on the left and the rate of the ANFs vs the BFs of the ANFs for the stimulus to the right.

Neurons in the DCN

The neurons in the DCN and the connections between them are shown in Fig 4.1(a). This neural circuit was proposed in [122] to explain the responses of the type 4 neurons of the DCN. They conjectured the existence of the neurons which they called the WBIs which receive weak inhibitory inputs from a wide range of ANF frequencies. So they respond weakly to pure tones and strongly to broadband noise. They showed that the responses of the type 4 neurons and the type 2 neurons could be explained if one considered the inhibition from these cells. The onset chopper neurons in the Ventral Cochlear Nucleus(VCN) are known to have responses similar to that of the WBIs [133].

The type 2 neurons are excited by the ANFs and inhibited by the WBIs. The response properties of these neurons as a result of the interaction of the above two kinds of neurons

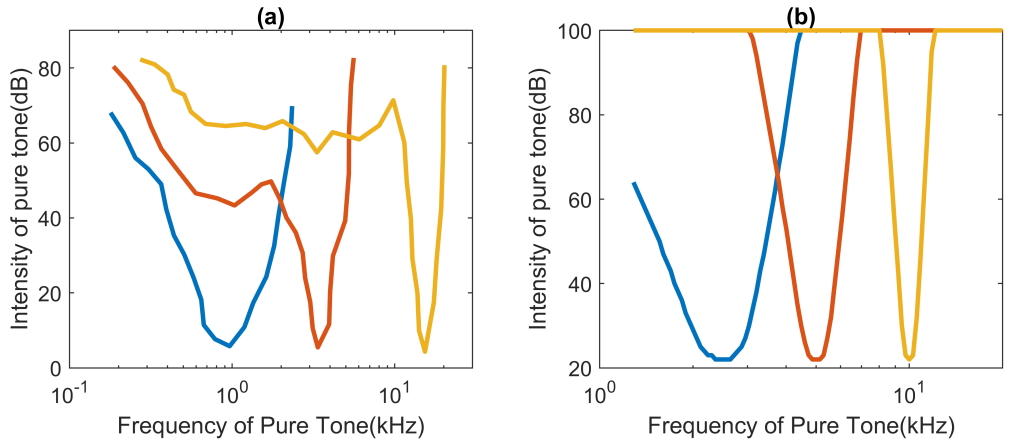

Figure 4.3: **Comparison of experimental and model tuning curves for ANFs.** a) Tuning curves for ANFs obtained experimentally. The tuning curve for an ANF is intensity as a function of pure tone $s(f_{pt})$, at which the ANF responds above its spontaneous rate. Adapted from ([132]) b) Tuning curves obtained from the model of the ANFs above.

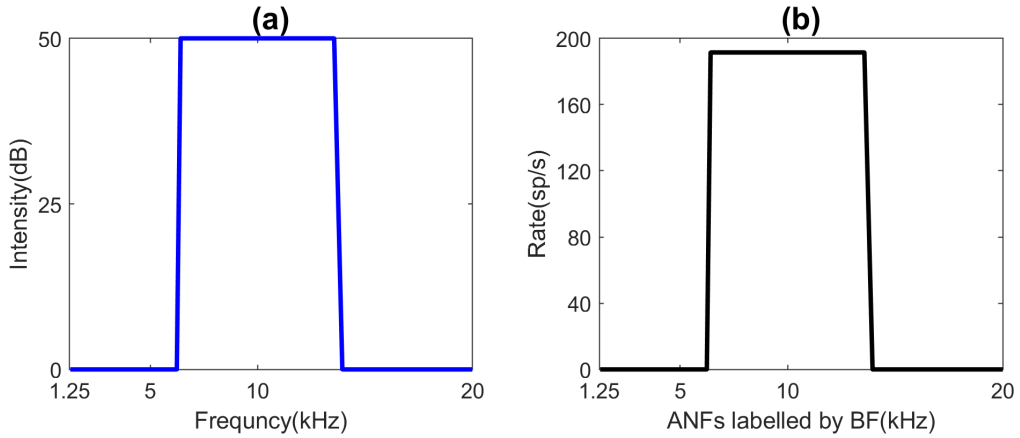

Figure 4.4: **Response of ANFs to broadband noise.** a) Broadband noise with a constant power spectrum of 50 dB centered on 5kHz having a width of 7kHz. b) Response of ANFs labelled by their BFs to the broadband noise on the left.

are discussed in the next section. The type 4 neurons are excited by the ANFs and inhibited by the type 2 neurons and the WBIs.

We give the input output functions of the above neurons and the weight matrices connecting them in 4.5. The input output functions are the same as that in [126] but we do not have any justification for these except that they work well. The weight functions that we use are closely related to how receptive fields of neurons are modelled. This kind of weight functions have not been previously used to model this neural network. Our model helps shed light on the receptive fields of the different neurons and their frequency

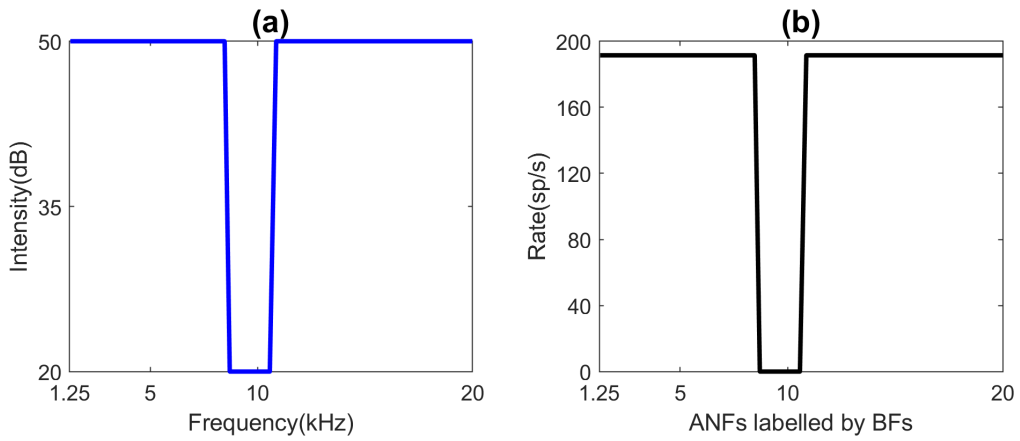

Figure 4.5: **Response of ANFs to notched noise.** a) Notch of width 1.6 kHz centered at 9 kHz in a broadband noise of 14 kHz. b) Rate of ANFs labelled by their BFs for the notched noise described in (a).

integration properties.

Neurons in the IC

Davis et al[123] have recorded the response of the neurons in the inferior colliculus to pure tone sweeps and notched noise sweeps. They show that the type O neurons of the IC are sensitive to the position of the notches. They have shown that these neurons receive direct excitatory inputs from the type 4 neurons of the DCN. They also propose a local circuit in the IC to explain the response of the type O neurons to the different stimuli described above. They conjecture the existence of narrow band inhibitors which inhibit the type O neurons and are excited by frequencies in a narrow range lying below the BF of the type O neuron which it inhibits. They call these neurons the Narrow Band Inhibitors(NBI). They also conjectured the existence of neurons which receive input from a wide range of frequencies and excite the type O neurons called the Wide Band Excitators (WBE). For our model, we only consider the NBIs which are necessary to reproduce the sensitivity of the type O neurons to notched noises. We will describe below what we fail to reproduce as a result of not considering the WBEs. It has been conjectured in [123] that the type I and type V units in the IC might be the narrow band inhibitors and the onset units in the IC could be the WBEs but more experiments are needed to confirm these conjectures.

4.3 Results

The type 2 neurons respond strongly to pure tones at their BF and are characterised by their non-monotonic rate level functions. Fig 4.6(a) shows the rate level function obtained from our model. We get a non-monotonic rate level function as predicted by experiments. This is because, as intensity increases, the range of ANFs which respond to a pure tone increases. Therefore the rate of the WBIs as the intensity of the pure tone increases. Since the WBIs inhibit the type 2 neurons, the increase in their rate brings down the rate of the type 2 neurons at higher intensities of pure tones. This lends some support to the widening of the response curves of the ANFs as the intensity increases and to the existence of WBIs. Fig 4.6(b) shows that the overall rate of the type 2 neurons decreases as the width of the broadband noise increases. This can again be explained by the fact that the response of the WBIs increases as the width of the broadband noise increases. Type 2 neurons are also experimentally found to have a high threshold which leads to the neurons being completely shut off if the input is below the threshold. In our simulations the type 2 neurons are completely shut off by a notched noise when the notch lies above its BF.

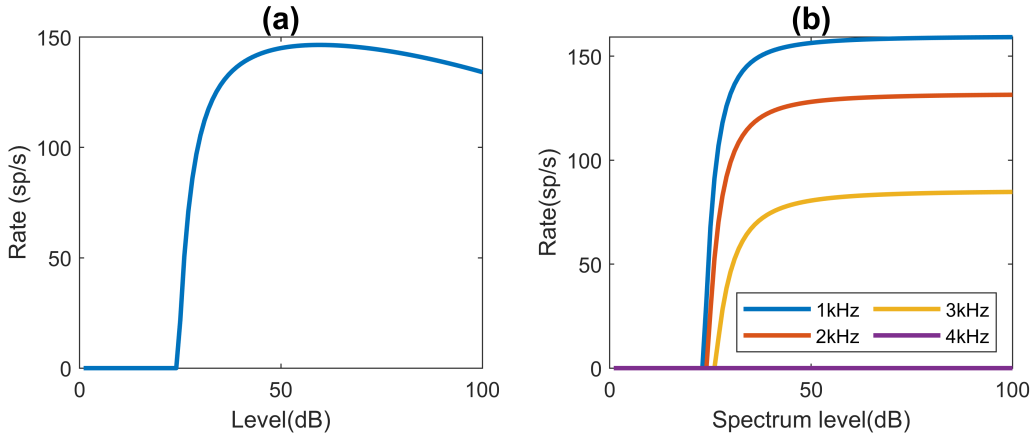

Figure 4.6: **Response of type 2 neurons.** a) Rate vs level curve for the response of a type 4 neuron to a pure tone at BF. b) Rate vs level curve of a type 2 neuron for noise. We can see that the rate decreases as the width of the broadband noise increases which is also observed experimentally.

The type 4 neurons are excited for a narrow range of frequencies at low levels(dB) and are inhibited or shut off for higher intensities. Fig 4.7(a), shows the rate vs level curve

for a pure tone at its BF for a type 4 neuron. The type 4 neurons are modelled such that they have a spontaneous rate of 30 kHz. We see that the rate increases above the spontaneous rate as intensity increases above 20 dB which is the threshold for the ANFs. At low intensities the type 4 neurons only receive excitatory inputs from the ANFs since the type 2 neurons have a high threshold as we discussed above and the WBIs respond very weakly at low intensities. As the intensity increases, the inhibition of the type 2 neurons and the WBIs soon overcome the excitation from the ANFs. So we have a narrow peak in the rate vs level curve for the type 4 neurons for pure tones. Fig 4.7(b) shows the rate vs spectrum level of broadband noise for type 4 neurons. The type 4 neurons are excited above their spontaneous rates by broadband noise of different widths. The rate goes down as the width of the broadband noise increases because the inhibitory inputs from the WBIs increase.

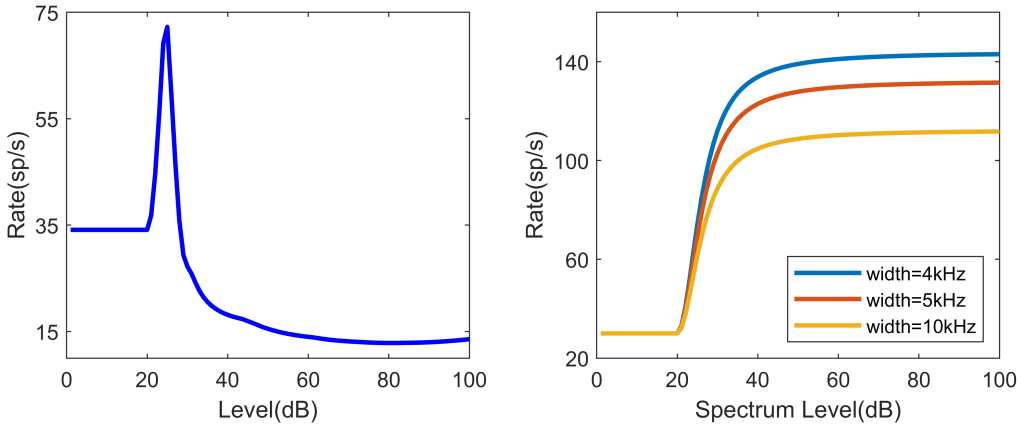

Figure 4.7: **Response of a single type 4 neuron to pure tone at BF and broadband noise.** a) Rate vs level curve for a type 4 neuron to a pure tone at its BF. It is excited above its spontaneous rate in a narrow frequency range above the threshold and it is inhibited for higher decibels. b) Rate vs spectrum level curves of a type 4 neuron for broadband noise.

Fig 4.8(a) shows the response of the type 4 neurons for pure tone sweeps at different levels. As we discussed above, the type 4 neurons are excited above their spontaneous rates for a narrow range of frequencies at low levels. At higher levels, the rate of the type 4 neurons fall below the spontaneous rate. We failed to reproduce the inhibition of the type 4 neurons for pure tones in the entire frequency range for higher levels. There is inhibition only over the range from which the WBIs receive inputs. Fig 4.8(b) shows

the response of a type 4 neuron with BF 9.2 kHz to notched noise sweeps. The notch has a width of 1.6 kHz and its center is swept over 3 octaves. The type 4 neurons are inhibited below their spontaneous rate when the notch lies over the best frequency of the neuron and they are excited above their spontaneous rate when the notch center moves away from the best frequency. We have not been able to reproduce the different ranges of inhibitions of the neurons at different spectrum levels of the notched noise. Experiments have shown that there is maximum inhibition at intermediate levels.

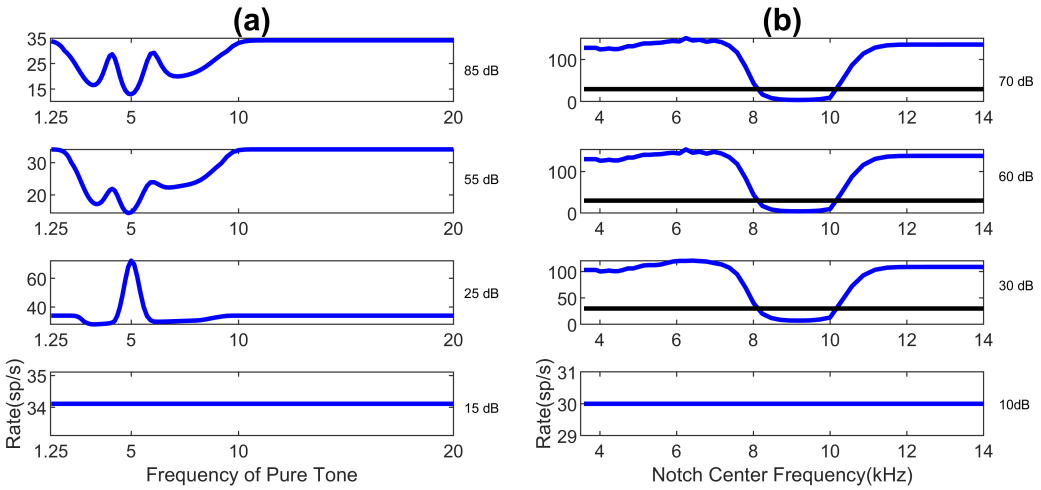

Figure 4.8: **Responses of Type 4 neurons to pure tone sweeps and notched signal sweeps.** a)Rate vs frequency of pure tone of a type 4 neuron with BF 5 kHz for pure tone sweeps at different levels. b) Rate vs notch center frequency of a notched noise with notch width 1.6 kHz which is swept over 3 octaves, of a type 4 neuron with BF 9.2 kHz.

Fig 4.9(a) shows the rate vs pure tone frequency curve for a type O neuron with BF 5 kHz at different levels. The type O neurons are excited above spontaneous rates at low levels and a narrow range of pure tone frequencies. This behaviour is similar to the type 4 neurons of the DCN. The drop in the rates just below the BF is due to the inhibition from the NBIs whose BFs lie in a narrow range of frequencies below the BF of the type O neuron which they inhibit. As the intensity increases the rates of the type 4 neurons themselves fall below spontaneous rates and hence the type O neurons are no longer excited above their spontaneous rates either. Fig 4.9(b) shows the rate vs notch center frequency curves for a type O neuron with BF 9.2 kHz at different spectrum levels of the notched noise. The type O neuron is excited above its spontaneous rate when the notch

lies just below the BF of the neuron. This is because in our model the type O neurons are inhibited by NBIs whose BFs lie below the BF of the type O neuron it inhibits. So when the notch lies over the BFs of the NBIs, the type O neurons do not receive any inhibition and are excited by the type 4 neurons. The width of the excitation depends on the integration of inputs from the type 4 neurons and the NBIs.

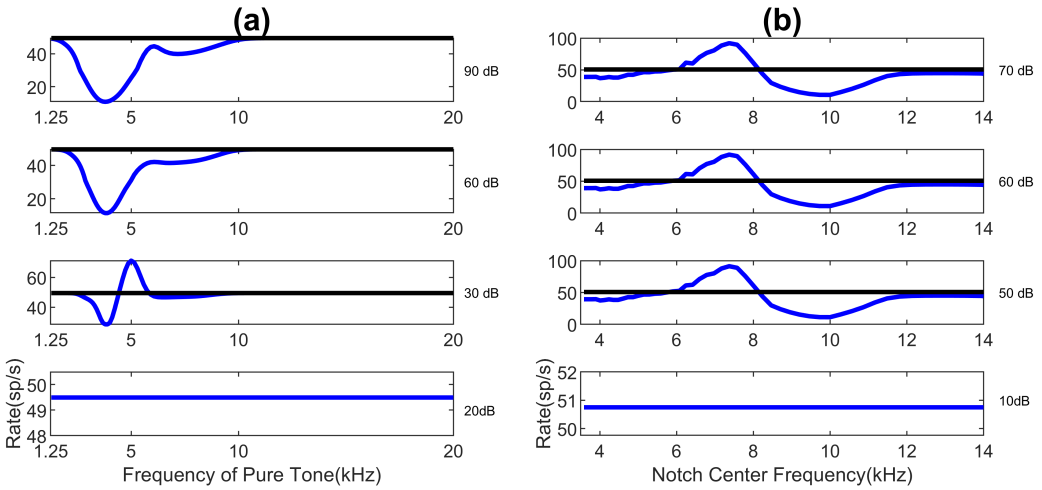

Figure 4.9: **Response of type O neurons to pure tone sweeps and notched noise sweeps at different levels.** The black lines represent the spontaneous rates. a) Rate vs pure tone frequency of a type O neuron with BF 5 kHz. The neuron is excited for a narrow range of pure tone frequencies at a low intensity like type 4 neurons. b) Rate vs notch center frequency of a type O neuron with BF 9.2 KHz. The notch had a width of 1.6 kHz and was swept across a range of 2 octaves from 3.5 kHz to 14 kHz.

4.4 Discussion

We have been able to reproduce the main features of the response of the type 4 neurons and the type O neurons to notched noises and shown their sensitivity to the frequency of notches as found in experiments. The type 4 neurons of the DCN have been experimentally found to be excited by a narrow range of pure tone frequencies at BF at low intensities and inhibited at high intensities. They are inhibited by notched noise when the notch lies over the BF of the neuron. The width of the inhibitory regions for both the pure tone and the notched noise do not match the experimental results but we have reproduced the two main features characterising the type 4 neuron mentioned above as shown in Fig 4.8.

We have also reproduced the general characteristics for the response of the type O neurons as shown in Fig 4.9. The pure tone response characteristics are the same as that of the type 4 neurons of the DCN where they have a narrow region of excitation at low intensities and are inhibited at high intensities. However, unlike the type O neurons these neurons are excited when the notch center lies below their BF. We have not been able to reproduce the broad excitation of the type O neurons at low intensities. This could be because we have not included the wideband excitators that were conjectured in [123].

We use an approximation for the responses of the auditory nerve fibers for complex stimuli like broadband noise and notched noise. The response of the auditory nerve fibers are crucial since it it is the first layer and all subsequent layers receive inputs from the ANFs, which might be one of the reasons that some of the response features to complex stimuli for the type 4 neurons and type O neurons are missing. More accurate responses for the ANFs have to be modeled by filter functions on the stimuli [134],[111] which are in the time domain. For this work we have assumed our neurons to be in steady state. To match experimental results more closely, we would have to simulate a dynamical model for all our neurons and Fourier transform it to obtain rate vs frequency curves. Even so, rate vs intensity curves would be difficult to obtain from such models. We leave the development of such a dynamical model to future work. As we have mentioned earlier, the parameters in the weight matrices of our model might not be optimal since they were found by a trial and error method. Optimising these parameters would also bring the responses closer to their experimental analogs.

Recently, convolutional neural nets (CNNs) have been used to model retinal responses to natural stimuli and to determine the receptive fields of neurons in the visual pathway [135]. This neural network can also be thought of as a CNN with a single filter (a frequency dependent filter). It could be trained on various natural stimuli with experimentally found responses to find the receptive fields of the neurons. We could then compare the proposed receptive fields to the receptive fields obtained by such training. In a different approach, we could train our neural network on natural stimuli with experimentally obtained responses to optimise the parameters in the weight functions defined below.

This would be a more algorithmic approach to determine these parameters.

As the model stands now, we have shown that the neurons in the DCN and IC are important in the processing of notches. A full mapping between the position of these notches to the vertical location of the sound has not yet been found. It is yet to be experimentally found how the inputs from the IC are processed in the auditory cortex. The final aim of this neural circuit would be to have a mapping between the position of the notch in the spectrum to the location of the source of the sound.

4.5 Supplementary Information

Mathematical Description of the Model

In this section we give the equations for our feed forward neural network. As has been already discussed, we are considering a rate model at its fixed point. This can be easily generalised to a dynamical model by looking at the stimuli as a function of time and then performing Fourier transforms to obtain rate vs frequency plots. The different weights that we use are given in Fig 4.10. The WBIs receives inputs from ANFs 1.6 octaves around their BFs with strength 0.04.

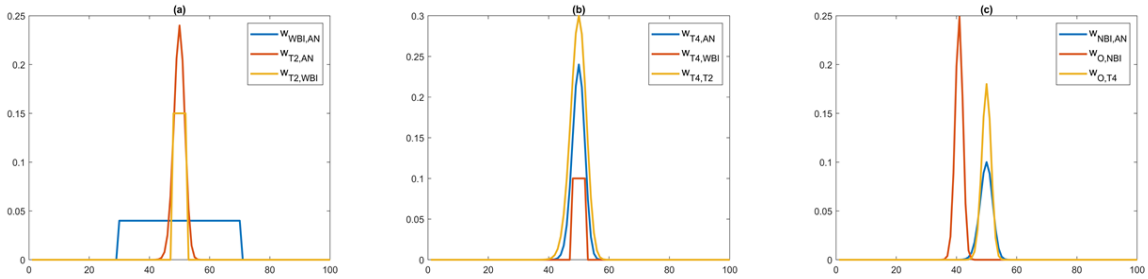

Figure 4.10: **Weights connecting the different neurons in our model** a) Weights from ANFs to WBI(*blue*) and weights from WBI and ANFs to T2 with BF 10 kHz. (b) Weights from ANF, WBI, T2 to T4 neuron with BF 10 kHz. (c)Weights from ANF to NBI(*blue*) and weights from NBI, T4 to O.

The type 2 neurons receive excitatory inputs from the ANFs and inhibitory connections from the WBIs.All the neurons are labelled by their BFs. For example, a ANF with BF f is labelled as f_{ANF}.

$$I_{T2} = W_{T2,ANF} r_{ANF} - W_{T2,WBI} r_{WBI} \tag{4.2}$$

$$W_{T2,ANF} = 0.24 exp\left(-\left(\frac{f_{T2} - f_{ANF}}{0.07 f_{ANF}}\right)^2\right)$$

$$W_{T2,WBI} = \begin{cases} 0.15, & 2^{-0.08} f_{WBI} \leq f_{T2} \leq 2^{0.08} f_{WBI}, \\ 0, & \text{otherwise.} \end{cases} \tag{4.3}$$

$f(x)$ gives the input output function for ANFs. This input output function was taken from [126]

$$\begin{aligned} r_{T2} =& f(I_{T2}), \\ f(x) =& \frac{ax}{x+b}. \end{aligned} \tag{4.4}$$

In our model, $a = 250\, sp/s$, $b = 70\, sp/s$ and the T2 neurons have a threshold of $35 sp/s$. Note that the units of the parameter a and b are such that r_{T2} has the units of spikes/s.

Type 4 neurons receive excitatory connections from ANFs and inhibitory connections from type 2 neurons and WBIs.

$$I_{T4} = W_{T4,ANF} r_{ANF} - W_{T4,WBI} r_{WBI} - W_{T4,T2} r_{T2} \tag{4.5}$$

$$W_{T4,ANF} = 0.22 exp\left(-\left(\frac{f_{T4} - f_{ANF}}{0.08 f_{T4}}\right)\right) \tag{4.6}$$

$$
\begin{aligned}
r_{T4} =& g(inp_T 4), \\
g(x) =& \begin{cases} ce^{\frac{x}{d}}, & x \leq 0, \\ x + c, & x > 0. \end{cases}
\end{aligned} \tag{4.7}
$$

For our model, $c = 30, d = 30$.

The type O neurons of the IC are excited by the type 4 neurons of the DCN and inhibited by the narrow band inhibitor(NBI) in the IC.

$$
I_O = W_{O,T4} r_{T4} - W_{O,NBI} r_{NBI} \tag{4.8}
$$

The input output function for type O units were the same as that of the type 4 units.

Chapter 5

Conclusion

In this dissertation we have studied different questions related to structure neural activity patterns or neural representations an how the structure, connectivity of and the computations performed by networks and neural representations are linked. In the second chapter we found that neural manifolds formed by common population codes are extremely nonlinear. We are exploring a few future directions. What kind of non-linear dimensionality reduction methods can we use to find the non-linear structure of these manifolds? Another question we are interested in exploring is how does the dimensionality of representation vary over different regions in the brain and what it means for the computations performed by these regions?

In the third chapter we considered sets of neurons interacting through expander like connections. This is an ongoing work and some of the directions we are considering to finish it are outlined in Chapter 3 itself. In future work, I would like to explore a network of densely connected recurrently connected subnetworks interacting with each other through sparse expander like connections and what kind of computations these networks can perform? Another direction I am interested in exploring is the relation between efficient coding and sparse coding.

In the fourth chapter we investigated the interaction between different populations of neurons in the auditory pathway which give rise to the complex repsonses of these neurons. We would like to extend this network to the cortex and produce an actual map

between the position of the notches to the angle of elevation of a sound source. We would also like to explore ways to damage this network to study the effects of neurodegenerative diseases like Alzheimer's on the auditory pathway.

Bibliography

1. Jazayeri, M. & Ostojic, S. Interpreting neural computations by examining intrinsic and embedding dimensionality of neural activity. *Curr. Opin. Neurobiol.* **70,** 113–120 (2021).

2. Chaudhuri, R., Gerçek, B., Pandey, B., Peyrache, A. & Fiete, I. The intrinsic attractor manifold and population dynamics of a canonical cognitive circuit across waking and sleep. *Nat. Neurosci.* **22,** 1512–1520 (2019).

3. Stringer, C., Michaelos, M., Tsyboulski, D., Lindo, S. E. & Pachitariu, M. High-precision coding in visual cortex. *Cell* **184,** 2767–2778 (2021).

4. Pashkovski, S. L. *et al.* Structure and flexibility in cortical representations of odour space. *Nature* **583,** 253–258 (2020).

5. Churchland, M. M. *et al.* Neural population dynamics during reaching. *Nature* **487,** 51–56 (2012).

6. Hopfield, J. J. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences* **79,** 2554–2558 (1982).

7. Ben-Yishai, R., Bar-Or, R. L. & Sompolinsky, H. Theory of orientation tuning in visual cortex. *Proceedings of the National Academy of Sciences* **92,** 3844–3848 (1995).

8. Mastrogiuseppe, F. & Ostojic, S. Linking connectivity, dynamics, and computations in low-rank recurrent neural networks. *Neuron* **99,** 609–623 (2018).

9. Ercsey-Ravasz, M. *et al.* A predictive network model of cerebral cortical connectivity based on a distance rule. *Neuron* **80,** 184–197 (2013).

10. Georgopoulos, A. P., Schwartz, A. B. & Kettner, R. E. Neuronal population coding of movement direction. *Science* **233,** 1416–1419 (1986).

11. Tanaka, K., Saito, H.-a., Fukada, Y. & Moriya, M. Coding visual images of objects in the inferotemporal cortex of the macaque monkey. *Journal of neurophysiology* **66,** 170–189 (1991).

12. Fujita, I., Tanaka, K., Ito, M. & Cheng, K. Columns for visual features of objects in monkey inferotemporal cortex. *Nature* **360,** 343–346 (1992).

13. Tsunoda, K., Yamane, Y., Nishizaki, M. & Tanifuji, M. Complex objects are represented in macaque inferotemporal cortex by the combination of feature columns. *Nature neuroscience* **4,** 832–838 (2001).

14. Pasupathy, A. & Connor, C. E. Responses to contour features in macaque area V4. *Journal of neurophysiology* **82,** 2490–2502 (1999).

15. Pasupathy, A. & Connor, C. E. Population coding of shape in area V4. *Nature neuroscience* **5,** 1332–1338 (2002).

16. Machens, C. K., Romo, R. & Brody, C. D. Functional, but not anatomical, separation of "what" and "when" in prefrontal cortex. *J. Neurosci.* **30,** 350–360 (2010).

17. DiCarlo, J. J. & Cox, D. D. Untangling invariant object recognition. *Trends Cogn. Sci.* **11,** 333–341 (2007).

18. Chung, S., Lee, D. D. & Sompolinsky, H. Classification and geometry of general perceptual manifolds. *Physical Review X* **8,** 031003 (2018).

19. Kiang, N., Sachs, M. B. & Peake, W. Shapes of tuning curves for single auditory-nerve fibers. *J. Acoust. Soc. Am* **42,** 1341–1342 (1967).

20. Pasupathy, A. & Connor, C. E. Shape representation in area V4: position-specific tuning for boundary conformation. *Journal of neurophysiology* (2001).

21. Sadagopan, S. & Wang, X. Level invariant representation of sounds by populations of neurons in primary auditory cortex. *Journal of Neuroscience* **28,** 3415–3426 (2008).

22. Mazer, J. A., Vinje, W. E., McDermott, J., Schiller, P. H. & Gallant, J. L. Spatial frequency and orientation tuning dynamics in area V1. *Proceedings of the National Academy of Sciences* **99,** 1645–1650 (2002).

23. Schwartz, E. L., Desimone, R., Albright, T. D. & Gross, C. G. Shape recognition and inferior temporal neurons. *Proceedings of the National Academy of Sciences* **80,** 5776–5778 (1983).

24. Stokes, M. G. *et al.* Dynamic coding for cognitive control in prefrontal cortex. *Neuron* **78,** 364–375 (2013).

25. Nieder, A. & Miller, E. K. Coding of cognitive magnitude: Compressed scaling of numerical information in the primate prefrontal cortex. *Neuron* **37,** 149–157 (2003).

26. Nieder, A. & Miller, E. K. A parieto-frontal network for visual numerical information in the monkey. *Proc. Natl. Acad. Sci.* **101,** 7457–7462 (2004).

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

28. Roweis, S. T. & Saul, L. K. Nonlinear dimensionality reduction by locally linear embedding. *Science* **290,** 2323–2326 (2000).

29. Tenenbaum, J. B., Silva, V. d. & Langford, J. C. A global geometric framework for nonlinear dimensionality reduction. *Science* **290,** 2319–2323 (2000).

30. Park, M. *et al.* Bayesian manifold learning: the locally linear latent variable model (LL-LVM). *Adv Neural Inf Process Syst.* **28** (2015).

31. Gray, R. M. Toeplitz and circulant matrices: A review (2006).

32. O'Keefe, J. & Burgess, N. Geometric determinants of the place fields of hippocampal neurons. *Nature* **381,** 425–428 (1996).

33. Piazza, M., Izard, V., Pinel, P., Le Bihan, D. & Dehaene, S. Tuning curves for approximate numerosity in the human intraparietal sulcus. *Neuron* **44,** 547–555 (2004).

34. Wigderson, A. & Wigderson, Y. The uncertainty principle: variations on a theme. *Bulletin of the American Mathematical Society* **58,** 225–261 (2021).

35. Donoho, D. L. & Stark, P. B. Uncertainty principles and signal recovery. *SIAM Journal on Applied Mathematics* **49,** 906–931 (1989).

36. Salinas, E. & Abbott, L. F. A model of multiplicative neural responses in parietal cortex. *Proc. Natl. Acad. Sci.* **93,** 11956–11961 (1996).

37. DeAngelis, G. C., Ohzawa, I. & Freeman, R. Spatiotemporal organization of simple-cell receptive fields in the cat's striate cortex. I. General characteristics and postnatal development. *J. Neurophysiol.* **69,** 1091–1117 (1993).

38. Lindeberg, T. A computational theory of visual receptive fields. *Biological cybernetics* **107,** 589–635 (2013).

39. Cover, T. M. & Thomas, J. A. Information theory and statistics. *Elements of information theory* **1,** 279–335 (1991).

40. Litwin-Kumar, A., Harris, K. D., Axel, R., Sompolinsky, H. & Abbott, L. Optimal degrees of synaptic connectivity. *Neuron* **93,** 1153–1164 (2017).

41. Gao, P. *et al.* A theory of multineuronal dimensionality, dynamics and measurement. *bioRxiv,* 214262 (2017).

42. Levina, E. & Bickel, P. Maximum likelihood estimation of intrinsic dimension. *Adv Neural Inf Process Syst.* **17** (2004).

43. Low, R. J., Lewallen, S., Aronov, D., Nevers, R. & Tank, D. W. Probing variability in a cognitive map using manifold inference from neural dynamics. *bioRxiv,* 418939 (2018).

44. McInnes, L., Healy, J. & Melville, J. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426* (2018).

45. Gray, R. On the asymptotic eigenvalue distribution of Toeplitz matrices. *IEEE Trans. Inf. Theory* **18,** 725–730 (1972).

46. Blumensath, T. *et al.* Spatially constrained hierarchical parcellation of the brain with resting-state fMRI. *Neuroimage* **76,** 313–324 (2013).

47. Kanwisher, N. Functional specificity in the human brain: a window into the functional architecture of the mind. *Proceedings of the National Academy of Sciences* **107,** 11163–11170 (2010).

48. Kanwisher, N., McDermott, J. & Chun, M. M. The fusiform face area: a module in human extrastriate cortex specialized for face perception. *Journal of neuroscience* **17,** 4302–4311 (1997).

49. McCarthy, G., Puce, A., Gore, J. C. & Allison, T. Face-specific processing in the human fusiform gyrus. *Journal of cognitive neuroscience* **9,** 605–610 (1997).

50. Epstein, R. & Kanwisher, N. A cortical representation of the local visual environment. *Nature* **392,** 598–601 (1998).

51. Downing, P. E., Jiang, Y., Shuman, M. & Kanwisher, N. A cortical area selective for visual processing of the human body. *Science* **293,** 2470–2473 (2001).

52. Albright, T. D. Direction and orientation selectivity of neurons in visual area MT of the macaque. *Journal of neurophysiology* **52,** 1106–1130 (1984).

53. Zeki, S. *et al.* A direct demonstration of functional specialization in human visual cortex. *Journal of neuroscience* **11,** 641–649 (1991).

54. DeAngelis, G. C., Cumming, B. G. & Newsome, W. T. Cortical area MT and the perception of stereoscopic depth. *Nature* **394,** 677–680 (1998).

55. Friederici, A. D. & Gierhan, S. M. The language network. *Current opinion in neurobiology* **23,** 250–254 (2013).

56. Power, J. D. & Petersen, S. E. Control-related systems in the human brain. *Current opinion in neurobiology* **23,** 223–228 (2013).

57. Pessoa, L. *et al.* Beyond brain regions: Network perspective of cognition–emotion interactions. *Behavioral and Brain Sciences* **35,** 158 (2012).

58. Barrett, L. F. & Satpute, A. B. Large-scale brain networks in affective and social neuroscience: towards an integrative functional architecture of the brain. *Current opinion in neurobiology* **23,** 361–372 (2013).

59. Sporns, O., Chialvo, D. R., Kaiser, M. & Hilgetag, C. C. Organization, development and function of complex brain networks. *Trends in cognitive sciences* **8,** 418–425 (2004).

60. Bullmore, E. & Sporns, O. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature reviews neuroscience* **10,** 186–198 (2009).

61. Rubinov, M. & Sporns, O. Complex network measures of brain connectivity: uses and interpretations. *Neuroimage* **52,** 1059–1069 (2010).

62. Sporns, O. Graph theory methods: applications in brain networks. *Dialogues in clinical neuroscience* (2022).

63. Modha, D. S. & Singh, R. Network architecture of the long-distance pathways in the macaque brain. *Proceedings of the National Academy of Sciences* **107,** 13485–13490 (2010).

64. Harris, K. D. & Mrsic-Flogel, T. D. Cortical connectivity and sensory coding. *Nature* **503,** 51–58 (2013).

65. Sompolinsky, H., Crisanti, A. & Sommers, H.-J. Chaos in random neural networks. *Physical review letters* **61,** 259 (1988).

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

67. Brunel, N. Dynamics of sparsely connected networks of excitatory and inhibitory spiking neurons. *Journal of computational neuroscience* **8,** 183–208 (2000).

68. Wang, X.-J. Synaptic reverberation underlying mnemonic persistent activity. *Trends in neurosciences* **24,** 455–463 (2001).

69. Constantinidis, C. & Wang, X.-J. A neural circuit basis for spatial working memory. *The Neuroscientist* **10,** 553–565 (2004).

70. Bouchacourt, F. & Buschman, T. J. A flexible model of working memory. *Neuron* **103,** 147–160 (2019).

71. Damasio, A. R. The brain binds entities and events by multiregional activation from convergence zones. *Neural computation* **1,** 123–132 (1989).

72. Damasio, A. R. Time-locked multiregional retroactivation: A systems-level proposal for the neural substrates of recall and recognition. *Cognition* **33,** 25–62 (1989).

73. Meyer, K. & Damasio, A. Convergence and divergence in a neural architecture for recognition and memory. *Trends in neurosciences* **32,** 376–382 (2009).

74. Isely, G., Hillar, C. & Sommer, F. Deciphering subsampled data: adaptive compressive sampling as a principle of brain communication. *Advances in neural information processing systems* **23** (2010).

75. Rozell, C. J., Johnson, D. H., Baraniuk, R. G. & Olshausen, B. A. Sparse coding via thresholding and local competition in neural circuits. *Neural computation* **20,** 2526–2563 (2008).

76. Albus, J. S. A theory of cerebellar function. *Mathematical biosciences* **10,** 25–61 (1971).

77. Marr, D. & Thach, W. T. A theory of cerebellar cortex. *From the Retina to the Neocortex: Selected Papers of David Marr,* 11–50 (1991).

78. Babadi, B. & Sompolinsky, H. Sparseness and expansion in sensory representations. *Neuron* **83,** 1213–1226 (2014).

79. Capalbo, M., Reingold, O., Vadhan, S. & Wigderson, A. *Randomness conductors and constant-degree expansion beyond the degree/2 barrier Proceedings of the 34th Annual ACM Symposium on Theory of Computing* (2002), 659–668.

80. Hoory, S., Linial, N. & Wigderson, A. Expander graphs and their applications. *Bulletin of the American Mathematical Society* **43,** 439–561 (2006).

81. Brito, G., Dumitriu, I. & Harris, K. D. Spectral gap in random bipartite biregular graphs and applications. *Combinatorics, Probability and Computing* **31,** 229–267 (2022).

82. Bonner, M. F., Peelle, J. E., Cook, P. A. & Grossman, M. Heteromodal conceptual processing in the angular gyrus. *Neuroimage* **71,** 175–186 (2013).

83. Vogt, B. A. & Pandya, D. N. Cortico-cortical connections of somatic sensory cortex (areas 3, 1 and 2) in the rhesus monkey. *Journal of comparative neurology* **177,** 179–191 (1978).

84. Rockland, K. S. & Pandya, D. N. Laminar origins and terminations of cortical connections of the occipital lobe in the rhesus monkey. *Brain research* **179,** 3–20 (1979).

85. Pandya, D. N. Anatomy of the auditory cortex. *Revue neurologique* **151,** 486–494 (1995).

86. Clavagnier, S., Falchier, A. & Kennedy, H. Long-distance feedback projections to area V1: implications for multisensory integration, spatial awareness, and visual consciousness. *Cognitive, Affective, and Behavioral Neuroscience* **4,** 117–126 (2004).

87. Fuster, J. M. & Alexander, G. E. Neuron activity related to short-term memory. *Science* **173,** 652–654 (1971).

88. Funahashi, S., Bruce, C. J. & Goldman-Rakic, P. S. Mnemonic coding of visual space in the monkey's dorsolateral prefrontal cortex. *Journal of neurophysiology* **61,** 331–349 (1989).

89. Compte, A., Brunel, N., Goldman-Rakic, P. S. & Wang, X.-J. Synaptic mechanisms and network dynamics underlying spatial working memory in a cortical network model. *Cerebral cortex* **10,** 910–923 (2000).

90. Constantinidis, C. & Procyk, E. The primate working memory networks. *Cognitive, Affective, and Behavioral Neuroscience* **4,** 444–465 (2004).

91. Marino, J. *et al.* Invariant computations in local cortical networks with balanced excitation and inhibition. *Nature neuroscience* **8,** 194–201 (2005).

92. Vogels, T. P. & Abbott, L. F. Signal propagation and logic gating in networks of integrate-and-fire neurons. *Journal of neuroscience* **25,** 10786–10795 (2005).

93. Vogels, T. P., Sprekeler, H., Zenke, F., Clopath, C. & Gerstner, W. Inhibitory plasticity balances excitation and inhibition in sensory pathways and memory networks. *Science* **334,** 1569–1573 (2011).

94. Candes, E. J. & Tao, T. Decoding by linear programming. *IEEE transactions on information theory* **51,** 4203–4215 (2005).

95. Becker, S. & Lim, J. A computational model of prefrontal control in free recall: strategic memory use in the California Verbal Learning Task. *Journal of Cognitive Neuroscience* **15,** 821–832 (2003).

96. Sipser, M. & Spielman, D. A. Expander codes. *IEEE transactions on Information Theory* **42,** 1710–1722 (1996).

97. Xu, W. & Hassibi, B. *Efficient compressive sensing with deterministic guarantees using expander graphs 2007 IEEE Information Theory Workshop* (2007), 414–419.

98. Candes, E. J., Romberg, J. K. & Tao, T. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences* **59,** 1207–1223 (2006).

99. Berinde, R., Gilbert, A. C., Indyk, P., Karloff, H. & Strauss, M. J. *Combining geometry and combinatorics: A unified approach to sparse signal recovery 2008 46th Annual Allerton Conference on Communication, Control, and Computing* (2008), 798–805.

100. Jafarpour, S., Xu, W., Hassibi, B. & Calderbank, R. Efficient and robust compressed sensing using optimized expander graphs. *IEEE Transactions on Information Theory* **55,** 4299–4308 (2009).

101. Vinje, W. E. & Gallant, J. L. Natural stimulation of the nonclassical receptive field increases information transmission efficiency in V1. *Journal of Neuroscience* **22,** 2904–2915 (2002).

102. DeWeese, M. R., Wehr, M. & Zador, A. M. Binary spiking in auditory cortex. *Journal of Neuroscience* **23,** 7940–7949 (2003).

103. Poo, C. & Isaacson, J. S. Odor representations in olfactory cortex:"sparse" coding, global inhibition, and oscillations. *Neuron* **62,** 850–861 (2009).

104. Olshausen, B. A. & Field, D. J. Sparse coding of sensory inputs. *Current opinion in neurobiology* **14,** 481–487 (2004).

105. Natarajan, B. K. Sparse approximate solutions to linear systems. *SIAM journal on computing* **24,** 227–234 (1995).

106. Chen, S. S., Donoho, D. L. & Saunders, M. A. Atomic decomposition by basis pursuit. *SIAM review* **43,** 129–159 (2001).

107. Donoho, D. L. & Elad, M. Optimally sparse representation in general (nonorthogonal) dictionaries via 1 minimization. *Proceedings of the National Academy of Sciences* **100,** 2197–2202 (2003).

108. Olshausen, B. A. & Field, D. J. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature* **381,** 607–609 (1996).

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

110. Barlow, H. B. *et al.* Possible principles underlying the transformation of sensory messages. *Sensory communication* **1,** 217–233 (1961).

111. Lewicki, M. S. Efficient coding of natural sounds. *Nature neuroscience* **5,** 356–363 (2002).

112. Schuz, A., Chaimow, D., Liewald, D. & Dortenman, M. Quantitative aspects of corticocortical connections: a tracer study in the mouse. *Cerebral cortex* **16,** 1474–1486 (2006).

113. Meyer, K. *et al.* Predicting visual stimuli on the basis of activity in auditory cortices. *Nature neuroscience* **13,** 667–668 (2010).

114. Meyer, K., Kaplan, J. T., Essex, R., Damasio, H. & Damasio, A. Seeing touch is correlated with content-specific activity in primary somatosensory cortex. *Cerebral Cortex* **21,** 2113–2121 (2011).

115. Steinberg, J. & Sompolinsky, H. Associative memory of structured knowledge. *Scientific Reports* **12,** 21808 (2022).

116. Szarek, S. J. Condition numbers of random matrices. *Journal of Complexity* **7,** 131–149 (1991).

117. Candes, E. J. & Tao, T. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE transactions on information theory* **52,** 5406–5425 (2006).

118. Donoho, D. L. For most large underdetermined systems of equations, the minimal ℓ_1-norm near-solution approximates the sparsest near-solution. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences* **59,** 907–934 (2006).

119. May, B. J. & Huang, A. Y. Spectral cues for sound localization in cats: A model for discharge rate representations in the auditory nerve. *The Journal of the Acoustical Society of America* **101,** 2705–2719 (1997).

120. Musicant, A. D., Chan, J. C. & Hind, J. E. Direction-dependent spectral properties of cat external ear: New data and cross-species comparisons. *The Journal of the Acoustical Society of America* **87,** 757–781 (1990).

121. Rice, J. J., May, B. J., Spirou, G. A. & Young, E. D. Pinna-based spectral cues for sound localization in cat. *Hearing research* **58,** 132–152 (1992).

122. Nelken, I. & Young, E. D. Two separate inhibitory mechanisms shape the responses of dorsal cochlear nucleus type IV units to narrowband and wideband stimuli. *Journal of neurophysiology* **71,** 2446–2462 (1994).

123. Davis, K. A., Ramachandran, R. & May, B. J. Auditory processing of spectral cues for sound localization in the inferior colliculus. *Journal of the Association for Research in Otolaryngology* **4,** 148–163 (2003).

124. Tuwaig, M. *et al.* Deficit in Central Auditory Processing as a Biomarker of Pre-Clinical Alzheimer's Disease. *Journal of Alzheimers Disease* **60,** 1589–1600. ISSN: 1387-2877. %3CGo%20to%20ISI%3E://WOS:000414612200033 (2017).

125. Mansour, Y., Blackburn, K., Gonzalez-Gonzalez, L. O., Calderon-Garciduenas, L. & Kuleszaa, R. J. Auditory Brainstem Dysfunction, Non-Invasive Biomarkers for Early Diagnosis and Monitoring of Alzheimer's Disease in Young Urban Residents Exposed to Air Pollution. *Journal of Alzheimers Disease* **67,** 1147–1155. ISSN: 1387-2877. %3CGo%20to%20ISI%3E://WOS:000460435000003 (2019).

126. Blum, J. J., Reed, M. C. & Davies, J. M. A computational model for signal processing by the dorsal cochlear nucleus. II. Responses to broadband and notch noise. *The Journal of the Acoustical Society of America* **98,** 181–191 (1995).

127. Reed, M. C. & Blum, J. J. A computational model for signal processing by the dorsal cochlear nucleus. I. Responses to pure tones. *The Journal of the Acoustical Society of America* **97,** 425–438 (1995).

128. Hancock, K. E., Davis, K. A. & Voigt, H. F. Modeling inhibition of type II units in the dorsal cochlear nucleus. *Biological cybernetics* **76,** 419–428 (1997).

129. Hancock, K. E. & Voigt, H. F. Wideband inhibition of dorsal cochlear nucleus type IV units in cat: a computational model. *Annals of biomedical engineering* **27,** 73–87 (1999).

130. Johnson, J. S., O'Connor, K. N. & Sutter, M. L. Segregating two simultaneous sounds in elevation using temporal envelope: Human psychophysics and a physiological model. *The Journal of the Acoustical Society of America* **138,** 33–43 (2015).

131. Dayan, P. & Abbott, L. F. *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems* ISBN: 0262541858 (The MIT Press, 2005).

132. Salimpour, Y. & Abolhassani, M. D. *Auditory wavelet transform based on auditory wavelet families 2006 International Conference of the IEEE Engineering in Medicine and Biology Society* (2006), 1731–1734.

133. Palmer, A. & Winter, I. *Best frequency (BF) threshold reductions caused by off-BF non-excitatory tones in onset units of the cochlear nucleus Sixteenth Midwinter Research Meeting of the Association for Research in Otolaryngology* (1993), 123.

134. Carney, L. H. A model for the responses of low-frequency auditory-nerve fibers in cat. *The Journal of the Acoustical Society of America* **93,** 401–417 (1993).

135. Maheswaranathan, N. *et al.* Deep learning models reveal internal structure and diverse computations in the retina under natural scenes. *bioRxiv,* 340943 (2018).