From the holdings of the Department of Special Collections , Princeton University Library, Princeton, NJ

These documents are either in the public domain, or can only be used for educational and research purposes (Fair use) as per U.S. Copyright law (text below). By accessing this file, all users agree that their use falls in the public domain or within fair use as defined by the copyrightlaw. We do not charge any permission or usage fees for the publication of images of material inour collections, including those provided by Princeton University Library, via our website, catalog, or directly from staff.For more information regarding copyright, credit, and citation please review the information onthe Copyright, Credit and Citation Guidelines page of our website.
https://library.princeton.edu/special-collections/policies/copyright-credit-and-citation-guidelines. If you have any questions, including how to order higher-quality images from the collection, contact Special Collections Public Services via our Ask Us! Form. https://library.princeton.edu/special-collections/ask-us

---

Doctoral Dissertations, 2011-2026 Neuroscience

2018

Representational and Robustness Principles of the Retinal Population Code

Loback, Adrianna Renee

Princeton, NJ : Princeton University

---

http://arks.princeton.edu/ark:/88435/dsp01ks65hf87m

U.S. Copyright law: The copyright law of the United States (Title 17, §108, United States Code) governs the making of photocopies or other reproductions of copyrighted material. Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be used for any purpose other than private study, scholarship, or research. If a user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of fair use that user may be liable for copyright infringement. This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve a violation of copyright law. https://www.copyright.gov/title17/title17.pdf

Representational and Robustness Principles of the Retinal Population Code

Adrianna Renee Loback

A Dissertation
Presented to the Faculty
of Princeton University
in Candidacy for the Degree
of Doctor of Philosophy

Recommended for Acceptance
by the Department of
Neuroscience
Adviser: Professor Michael Berry II

January 2018

© Copyright by Adrianna Renee Loback, 2017.

All rights reserved.

Abstract

A comprehensive theory of neural computation requires an understanding of the fundamental principles underlying the neural population codes that are employed throughout the brain. In this thesis, we focus on principles underlying the ganglion cell population code employed by the retina, which is the earliest circuit involved in visual processing. We investigate these principles using data-driven approaches within frameworks motivated by concepts from other quantitative disciplines, which is a methodological unifying theme. In Chapter 1 we place our work in the broader scientific context with a brief overview of the relevant conceptual background. Then in Chapter 2, we provide an overview of the relevant methodological background that is common to the specific methods and approaches discussed in the subsequent chapters.

As hinted at by the title, there are two key conceptual themes of focus in this thesis: the nature of what the retinal population code represents (e.g. which features of the external visual stimulus space are encoded), and how the retinal system functions in spite of noisy components. While the three main projects discussed in Chapters 3 through 5 are related to one or both of these themes, each is aimed at a different level of analysis. Our work in Chapter 3 focuses on the computational goals of the retinal system, which is investigated via a latent variable model approach. Building on this work, in Chapter 4 we investigate the nature of the representation of the noise-robust population codewords identified in Chapter 3, and discuss preliminary work that focuses on biologically-plausible algorithms for decoding in Chapter 5. We conclude with a discussion of open questions and future directions related to this thesis work.

Acknowledgements

I would like to thank my advisor, Professor Michael Berry, for invaluable guidance throughout my graduate studies. I am very grateful to have had the encouragement and freedom while in Michael's group to pursue the research questions that most excited me, and for Michael's patience in allowing me to grow into the research. Throughout, and especially during the writing and redaction stages, Michael's thoroughness and willingness to discuss ideas greatly influenced my development. I would also like to thank my thesis committee members, Professors Emmanuel Abbe, Bill Bialek, and Jonathan Pillow. I feel very privileged and grateful to have had the opportunity to take courses with both Professors Abbe and Bialek, whose teaching on important and enlightening subjects influenced and contributed to developing my perspective of my research subject. I am very grateful to Jonathan, who has always been very generous in offering his time to talk and offer advice. When juxtaposed with all of his contributions as DGS and to the theoretical neuroscience community in our department (e.g. I am grateful for his establishing of the theoretical neuroscience journal club!), this is even more inspiring.

I have learned something from all members of the Berry group whom I have interacted with over the years. In particular, I owe many thanks to Jason Prentice and Mark Ioffe, who helped me out at the start. I feel very fortunate to have overlapped during my time in the Berry group with Jason Prentice, who introduced me to the machine learning aspects of our field, and whose work influenced the direction of my research project. Other past and present members whom I am especially grateful to for many helpful discussions and for providing a sense of community are: Nikhil Deshmukh, Jan Homann, Sophie Bagur, and Sebastiaan van Opheusden. I would also like to thank Rebecca Khaitman, Everett Morgan, Alex Michaud, and the late Dawn Tindall, who were always immensely helpful and kind in addressing my administrative and program questions.

Now more than ever, I am grateful to those who invested their time and offered mentorship in the past. In particular, I will always be grateful for having had the opportunity as an undergraduate student to conduct summer research internships in Professor Loren Frank's group, which was my first introduction to the field of neuroscience. I am very grateful to Loren for having inspired me to pursue a PhD in neuroscience, and owe many thanks to Jai Yu and Walter German, who were extremely generous in offering their time and mentorship. Likewise, I am very grateful for the encouragement and mentorship of the late Professor Seung-Sup Kim.

I would also like to express my immense gratitude for my friends during my time at Princeton, who have provided an enormous amount of support and fun times: Ingrid Ockert, Georgina Hall, Carly Lay Geronimo, Amanda Siemann, Bianca Dumitrascu, Lili Cai, Stephanie Chan, Nana Kobby Aboagye and Marte Samuelstuen, and Ryan Ly and Kelsey Ockert. Thanks to Niko Markov for diligently organizing the rock climbing group in our department. I would also like to thank the wonderful fellow students on the GWiSE committee, and the Harvard MBB program for the opportunity to serve as a TA during the past summer. This work was partially funded by an NIH T32 predoctoral training fellowship.

Finally, thank you to my parents for their love, and for always instilling and demonstrating by example the importance and value of hard work.

Contents

Abstract iii

Acknowledgements iv

List of Tables x

List of Figures xi

1 Introduction and Scientific Context **1**

1.1 Neural population coding 1

1.2 The retina as a model system 2

1.3 Noise in the nervous system and retina 5

1.4 Marr's three levels of analysis of information processing systems . . . 6

2 Overview of General Methodology and Notation **8**

2.1 Experimental methods 8

2.1.1 Electrophysiology 8

2.1.2 Visual stimulus display 9

2.2 Data pre-processing 10

2.2.1 Spike sorting 10

2.2.2 Population response representation 10

2.3 Modeling the population response probability landscape 12

2.3.1 The scenario 12

2.3.2 Encoding vs. activity models 13

2.3.3 Maximum entropy framework 13
2.3.4 A latent variable model approach 16
2.3.5 Parameter estimation via the EM algorithm 17
2.4 Appendix 22
2.4.1 Derivation of EM updates for a simplified model 22

3 Computational Goals Served by Collective Modes 25
3.1 Introduction 25
3.2 Methodology: Tree hidden Markov model 26
3.2.1 Model specification 27
3.2.2 Maximum likelihood estimation of model parameters 28
3.2.3 $L1$ regularization to mitigate overfitting 35
3.2.4 Computing the MAP latent sequence estimate 37
3.3 Modes encode a distinct feature basis set that includes orientation . . 39
3.3.1 Mode-triggered average analysis 40
3.3.2 Characterization of mode spatial RFs 41
3.4 Error correction endowed by collective modes 51
3.4.1 Background 51
3.4.2 Empirical applicability of the error-correcting code paradigm for the non-repeat stimulus limit 53
3.5 Robustness to independent fittings of the Tree HMM 55

4 The Structure of Retinal Population 'Codewords' 58
4.1 Introduction 58
4.2 Population codewords $\neq$ local maxima for low-repeat stimulus regimes 62
4.2.1 Scarcity of local maxima for low-repeat ensembles 64
4.2.2 Reassessing possible interpretations 68
4.3 Probing the geometry of collective modes 70

4.3.1 The role of sparseness 71
4.3.2 "Soft" local maxima 72
4.3.3 Ridges are a feature of the RGC response probability landscape 77
4.3.4 Collective modes correspond to ridges 83
4.4 Ridges correspond to "neuronal communities" 84
4.5 Discussion 86
4.5.1 Dependence of the population response probability landscape on stimulus ensemble statistics 86
4.5.2 Comparison of two candidate definitions of retinal population codewords 89
4.5.3 Connection to Hebb's cell assemblies 95
4.5.4 Ridges vs. peaks: Which are present during behavior? 96
4.6 Methodology details 97
4.6.1 Fitting the K-pairwise maximum entropy model 97
4.6.2 Selection of the probability model 98
4.6.3 Numerical exploration of the probability landscape 101
4.6.4 Parametric repeat analysis 103
4.6.5 Scaled count distribution 104
4.6.6 Assessing robustness of the set of mapped soft local maxima . 105
4.6.7 Estimating barrier-depths of soft local maxima 105
4.6.8 Visualizing ridges 106
4.6.9 Constructing the ridge union graph 109
4.7 Appendix 110
4.7.1 Control analyses 110
4.7.2 Pseudocode for the ridge search algorithm 113
4.7.3 Supplementary figures 116

5 Biologically-Plausible Algorithms for Learning Retinal Population Codewords **121**

5.1 Introduction 121

5.1.1 Primary biological constraints 122

5.1.2 Bayesian inference implemented by neural circuits 123

5.2 Approach and preliminary results 125

5.2.1 Error correction captured by simpler models 125

5.2.2 The stochastic WTA circuit motif architecture 128

5.2.3 Explicit mapping to a Bernoulli MM 130

5.2.4 The need for normalization (heterosynaptic plasticity) 132

5.3 Discussion and next approaches 137

5.4 Appendix 138

5.4.1 Proof of convergence of STDP learning rules 138

5.4.2 C++ code for the circuit model 141

6 Population-Level Encoding of Invariance in the Retina **150**

6.1 Introduction 150

6.2 Methodology 152

6.2.1 Experimental procedures 152

6.2.2 Optimal decoding analyses 153

6.3 Preliminary results 156

7 Conclusion and Future Directions **158**

Bibliography **163**

List of Tables

3.1 Monopole Mode RF Classification Results 51

4.1 Comparison of two independently-proposed codeword candidates of the RGC population code. Row 2 was shown by past work [87, 100, 116]. Row 3 and rows 4 and 5 for the "Collective Modes" column are restatements of our results. 90

List of Figures

1.1 **Schematic of the retina.** 4

2.1 **Results from two pilot studies supporting binary response representations.** 11

3.1 **Schematic of the structure of the Tree hidden Markov model (HMM).** 26

3.2 **Qualitative categories of mode RFs.** 42

3.3 **Identification of dipole-type collective mode spatial receptive fields.** 44

3.4 **Classification of monopole collective mode spatial receptive fields.** 49

3.5 **Error correcting properties of collective modes.** 54

3.6 **Robustness of identified collective modes over fittings of the model.** 56

3.7 **Robustness of identified collective modes over fittings of the model.** 57

4.1 **Schematic illustrating the concept of local maxima and basins.** 61

4.2 **Local maxima results for the dataset of** 128 **RGCs responding to Movie #1.** 64

4.3 **Experimental design and results for the parametric repeat analysis.** 67
4.4 **The empirical "scaled count distribution", $\widetilde{P}(K)$, vs. spike count.** 72
4.5 **Schematic illustrating the concept of K-soft local maxima and ridge points.** 74
4.6 **Soft local maxima results for two datasets.** 76
4.7 **Schematic illustrating the notions of *u-reachable* and *d-reachable*.** 80
4.8 **Example Type 1 rooted digraphs obtained.** 81
4.9 **Example Type 2 rooted digraphs obtained.** 82
4.10 **Empirical neuronal communities.** 86
4.11 **Schematic of dependence on stimulus ensemble statistics.** . . 88
4.12 **Schematic of biological decoding algorithm for neuronal communities.** 94
4.13 **Comparison of different models to empirical (K=2)-soft local maxima.** 100
4.14 **Control analysis results.** 112
4.15 **Scatter plot of mean firing rates for the raw data.** 116
4.16 **Local maxima results for the Movie #2 dataset ($N = 152$).** . . 116
4.17 **Examples of output rooted digraphs using d-reachable edges.** 117
4.18 **Soft local maxima results for different time bin widths.** 118
4.19 **Example Type 1 ridges for a different choice of time bin.** . . 119
4.20 **Example Type 2 ridge for a different choice of time bin.** . . . 120

5.1 **Error correction performance of latent modes of the Poisson MM.** 126

5.2 **Illustration of the WTA circuit motif model and population input.** 128
5.3 **Performance of the circuit model on a test case.** 135
5.4 **Tuning curve results for the test case.** 136

6.1 **Schematic of the stimulus design.** 153
6.2 **Characterization of responses for pilot studies.** 156
6.3 **Decoding results for the discrimination and invariance tasks.** 157

7.1 **Model goodness-of-fit for V1 data.** 159
7.2 **Training set sanity check.** 160

Chapter 1

Introduction and Scientific Context

Arguably the most complex information-processing system in existence is the human brain. The nascent field of neuroscience has made substantial strides in the past century to further our human understanding of the brain (despite the philosophical conundrum that there is still no consensus as to what it means to "understand the brain"). Yet for each elucidation, there are far more open questions and mysteries. In this chapter, we provide an overview of the relevant conceptual background to the particular problems that will be addressed in this thesis.

1.1 Neural population coding

It is now clear from ample experimental and theoretical evidence that neural circuits throughout the brain encode and transmit information using the temporal patterns of action potentials across large populations of neurons [2, 23, 41, 43, 49, 57, 77, 88, 98, 118]. Yet while the manner in which information is represented by single neurons has been intensively studied [27, 84, 93], the fundamental principles that underlie empirical neural population codes is still a topic of active investigation. The advent of new experimental technologies that enable simultaneous recording from hundreds to thousands of neurons - including multi-electrode array (MEA) technology [68,

111], whole-brain functional imaging via light-sheet microscopy [3], and *in vivo* two-photon [46, 49] and combined microendoscope [127] population calcium imaging - has opened up exciting new possibilities to study this central question.

Beyond the fundamental value of understanding how our brains operate, deeper knowledge of the principles underlying neural population codes would have substantial implications for multiple fields. For example, for informing brain-computer interface (BCI) research for the development of neural prosthetics for clinical applications [45], and for the development of novel solutions to engineering problems that the brain solves.

1.2 The retina as a model system

Neural circuits involved in early sensory processing offer a unique opportunity to study information processing in the brain, because one can define the input to these circuits, which allows for the direct study of the transformation from (known) input to neural output. The retina (schematized in Fig 1.1), which is the system of focus of this thesis work, is the earliest neural circuit involved in visual processing, and as such likewise confers this advantage. Although the retina is the neural part of the eye, it can essentially be considered part of the brain (in fact, the retina develops from an out-pouching of the neuroectoderm, along with the pigment epithelium and the optic stalk, which becomes the optic nerve [93]).

In addition to the above-mentioned advantage, there are three other substantial advantages offered by the retina as an exemplar system for studying neural population coding: (1) it is experimentally accessible; (2) dense multielectrode array technology makes it feasible to record from nearly all of the relevant output of the retina; and (3) substantial evidence supports that the retina performs complex computations, making it useful for scalable insight. Regarding (1), unlike higher order areas in

the brain (e.g. the cortex and hippocampus), there is little known neural feedback projecting to the functioning retina. Consequently, studies of the coupling between visual stimulus input and retinal output can generally be made without needing to take into account feedback from other brain areas that mights otherwise contribute to the computations. In turn, this simplifies the experimental requirements for studies of information processing and computation in the retina. Regarding (2), dense multi-electrode arrays were previously shown by the Berry group to allow for recording from a neural population in the retina that forms an essentially complete representation of a region of visual space [68, 107]. To date, it is still not possible to experimentally record from neural populations in the cortex that analogously correspond to complete representations, due to the increased dimensionality used by higher order brain areas.

Complex computation in the retina

Regarding (3), it is important to emphasize that the retina is not simply a CCD camera, i.e, it does not simply measure the light intensity at each point in space. In fact, many computations thought to be purely cortical in nature have been found in the retina. One prominent example is predictive computations. In the visual illusion called the "flash-lag effect", which is well-studied in the human psychophysics literature, a flash and a moving object that appear in the same location are perceived to be displaced from one another. While it was initially posited that this illusion is attributable to predictive computations that begin in visual cortex, it was shown in [1] that this predictive computation for moving stimuli (which compensates for the inherent transduction delay) begins in the retina. It has also been shown that the retina is capable of predicting a wide class of periodic temporal patterns [105], and more recently, it was demonstrated that retinal ganglion cell activity at a given moment in time contains information about their future activity [81]. Relatedly, the retina is also known to exhibit exaggerated responses to a violation of an expectation

(e.g. the omitted stimulus response [31]). Other well-documented examples of complex computations include the direction selectivity of a subset of ganglion cells (which demonstrates feature extraction) [10, 44], and the differential responses to local and global motion [79].

The layered structure of the retina

Given the above examples of complex computations that the retina is capable of, it is not surprising that its anatomical structure is correspondingly complex (schematized in Fig 1.1). Here we provide some rudimentary background regarding the 'biological hardware' underlying the retina's computational repertoire.

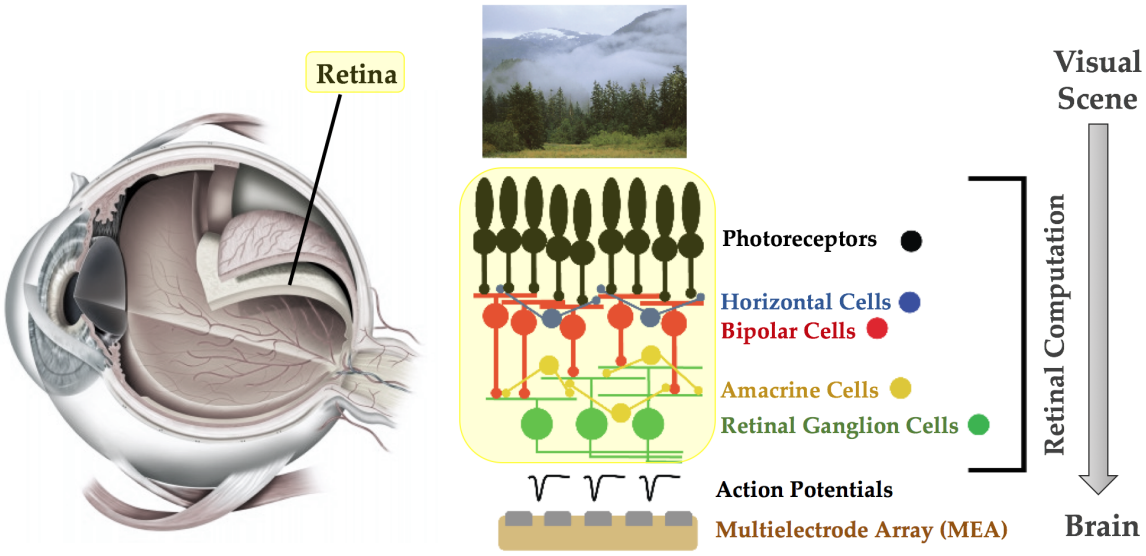

Figure 1.1: **Schematic of the retina.**
(*Left*) The retina sits in the posterior part of the eye. (*Right*) Cartoon, modified from [56], of the major layers and cell types that comprise the feedforward (photoreceptors → bipolar cells → retinal ganglion cells) pathway and lateral connections (horizontal and amacrine cells) in the retinal circuit. (See main text for details).

Unlike a CCD camera with a single layer, there are several layers that comprise the feedforward pathway of the retina through which the visual signal traverses: photons are absorbed by opsins in the outer segments of photoreceptors (black in Fig 1.1),

which activate intracellular protein cascades, which in turn change the current flow across the photoreceptor membrane. Specifically, the biochemical cascade induced by photon absorption results in a reduction in cGMP levels, which causes the closing of ion channels that normally maintain depolarization of the photoreceptor in the dark. This activation is transferred to the bipolar cells (red in Fig 1.1), and onwards to the final layer of retinal ganglion cells (green Fig 1.1), which have axons that comprise the optic nerve that terminates in downstream brain areas. For the data-driven studies presented in this thesis, the data is always MEA recording data taken from the retinal ganglion cell layer; this enables us to directly access the output of the retina that (in the functioning organism) would be transmitted to downstream brain areas. Finally, in addition to the feedforward structure, there are two major classes of cells that provide lateral connectivity: horizontal and amacrine cells (blue and yellow, respectively, in Fig 1.1).

While the effects of perturbations of light incident on the photoreceptors can be modeled as a linear convolution of the light intensity with some filter, such linear models fail for bipolar cells [31] and retinal ganglion cells. Stated another way, non-linear effects begin to be prominent for the other stages of processing in the retina. These multiple nonlinear layers add substantial computational power that simply is not present in single-layer linear networks [54].

1.3 Noise in the nervous system and retina

Noise permeates every level of the nervous system, from the perception of external sensory signals to the generation of motor responses [38]. Some examples of sources of noise that are relevant to the retinal system include sensory noise, cellular noise, electrical noise (e.g. stochastic fluctuations in the electric currents produced by the probabilistic opening and closing of voltage- or ligand-gated ion channels in each

neuron's membrane), and synaptic noise [38]. Perhaps unsurprisingly given these many sources of noise, even when the sensory input and other external conditions are kept as constant as possible, large trial-to-trial variability in the elicited retinal ganglion cell population responses is observed [63, 87, 116]. Yet despite this high variability in the neural population responses, our visual perception can be quite deterministic [7]. One possible explanation for the reconciliation of these facts is that robustness and fault-tolerance are built into the system. We explore this idea of whether fault-tolerance is built into the retinal population code in the latter half of Chapter 3, and then build on these results in Chapters 4 and 5.

1.4 Marr's three levels of analysis of information processing systems

In his seminal 1982 book, neuroscientist David Marr proposed that an information-processing device (such as the retina) must be understood at three levels "before one can be said to have understood it completely": the computational theory level, the representation/algorithm level, and the hardware implementation level [67]. Marr's computational theory level addresses the goal of the computation performed by the information-processing device (e.g. the computational goal of a cash register is to perform arithmetic). His proposed representation/algorithm level addresses how this computational goal is implemented, and the input and output representations used by the device in its implementation (e.g. a cash register uses Arabic numeral input and output representations, and uses the usual rules of base 10 addition for the algorithm). Marr's hardware implementation level addresses how the representation and algorithm are physically realized (e.g. a cash register can be mechanical or electronic). Notice that there can be degeneracies - e.g. an abacus and a cash register share the same computational goal, but differ at the hardware implementation level.

While the three projects discussed in Chapters 3 through 5 are all related to the above-mentioned conceptual theme of robustness in the retina, each is aimed at addressing a different level of analysis, in the sense of Marr's three levels. At the computational theory level, the results in Chapter 3 suggest that one goal of the retinal ganglion cell population is to compute population "codewords" to enable error correction to combat neural noise (and also to compute orientation as an emergent feature at the population level). Our work in Chapter 4 then focuses on advancing our understanding of the nature of the representation of these noise-robust population codewords. Finally, our work in Chapter 5 (which, at the time of writing, is still work in progress) focuses on a related question at the biological hardware implementation level: investigating a mechanistic, biologically plausible spiking circuit model with synaptic plasticity that can learn to extract the noise-robust codewords identified by our previous (non-biologically plausible) machine learning methods.

Chapter 2

Overview of General Methodology and Notation

In this chapter we provide an overview of the general procedures that are either shared among all of the subsequent chapters (in which case, we have decided to list them here as a contained entity for reference rather than extraneously repeat these details in each chapter), or are prerequisite to the specific methods utilized in the subsequent chapters. We first detail the general experimental setup, which applies to all of the data used in this thesis. Then, we provide some introductory-level background on the general probabilistic modeling approach and relevant context. We also introduce the mathematical notation that will be employed throughout.

2.1 Experimental methods

2.1.1 Electrophysiology

All retinal datasets analyzed in the data-driven projects comprising this thesis (Ch. 3 through 6) were acquired via multielectrode array (MEA) recordings from larval tiger salamander (*Ambystoma tigrinum*) retinal ganglion cells responding to visual stimuli.

All experiments were performed by Mark Ioffe, in strict accordance with the Guide for the Care and Use of Laboratory Animals of the NIH (Protocol 1828). Raw voltage traces were digitized (10 kHz sampling rate) and stored for offline analysis using a 252-channel preamplifier (MultiChannel Systems, Germany). Details are provided in [54]. For all recordings, only the responses of neurons that passed the standard tests for waveform stability and lack of refractory period violations were included.

2.1.2 Visual stimulus display

For all retinal experiments (Chapters 3 through 6), visual stimuli were projected onto the multielectrode array (MEA) from a CRT monitor at a frame rate of 60 Hz, and gamma corrected for the display.

For the datasets analyzed in Chapters 3 and 4, all experiments were based on the presentation of two stimulus classes: a natural movie (three experiments), and a binary white noise checkerboard (one experiment). The natural movie consisted of a 7-min gray scale recording of leaves and branches blowing in the wind. We conducted the natural movie experiments with three different designs: (1) in the first, which we will refer to throughout this thesis as "Movie #1", the full 7-min movie was looped (i.e. sequentially repeated) ten times, for a total movie length of approximately 70 min; (2) in the second, which we refer to throughout as "Movie #2", the movie was looped five times, but with a different pixel centered on the recording area for each repeat; (3) in the third, which we refer to as "Movie #4", we interleaved 73 unique movie segments, each 60 s in duration, with 73 repeats of a fixed "target" movie segment that was 60 seconds in duration. Design (2) was constructed this way to provide a non-repeated stimulus: since the patch of retina recorded by the MEA subtended only a small portion of the stimulus, the retinal input was effectively non-repeated over the full recording session. The artificial binary white noise stimulus, which we refer to throughout as "Movie #3", consisted of a 40×40 array of 58 μm

squares with light intensity randomly selected to be either bright or dark every 30 ms. This stimulus was formatted into 30 s periods that alternated between non-repeated stimulation (69 segments) and a repeated 30-s "target" movie segment (69 segments).

2.2 Data pre-processing

2.2.1 Spike sorting

To determine attribution of responses to neurons, spike sorting was performed offline using custom software. The spike sorting algorithm used was developed by Olivier Marre [68] and Dario Amodei [4]. Sebastiaan van Opheusden rewrote substantial portions of the automated algorithm, to improve the speed with which the calculations were performed [120]. The spike sorting was partially automated, with some significant manual input required. Technical details regarding the manual portion of the spike sorting procedure are provided in [54].

2.2.2 Population response representation

For each dataset, unless stated otherwise, we discretized spike trains using a time bin width of 20 ms. This value was chosen based on previous literature [13, 30, 58]. For the data-driven analyses in Chapters 3 through 6, we applied thresholding to obtain a binary response representation for each ganglion cell $i \in [N]$ during each time bin t (where throughout this thesis, N will denote the population size). That is, we set $x_i(t) = 1$ if neuron i fired at least one spike within time bin t, and 0 otherwise. This produces a sequence of binary population response vectors, which we denote as $\vec{x}(t) \equiv (x_1(t), \cdots, x_i(t), \cdots x_N(t)) \in \{0,1\}^N$. Note that this population response is largely conditioned on three aspects: the external stimulus itself, synaptic interactions, and neuronal-network history [24].

Our choice of a binary response representation (using a time bin width of 20 ms) was also based on preliminary results from pilot studies (Fig 2.1), which suggested that this assumption was justified, in that it accounts for $> 80\%$ of the per-cell mutual information. Interpreted from the viewpoint of downstream neurons, this binary response assumption corresponds with assuming a rectangular postsynaptic potential (PSP) kernel of width τ that defines the width of each time bin, and applying thresholding to binarize responses.

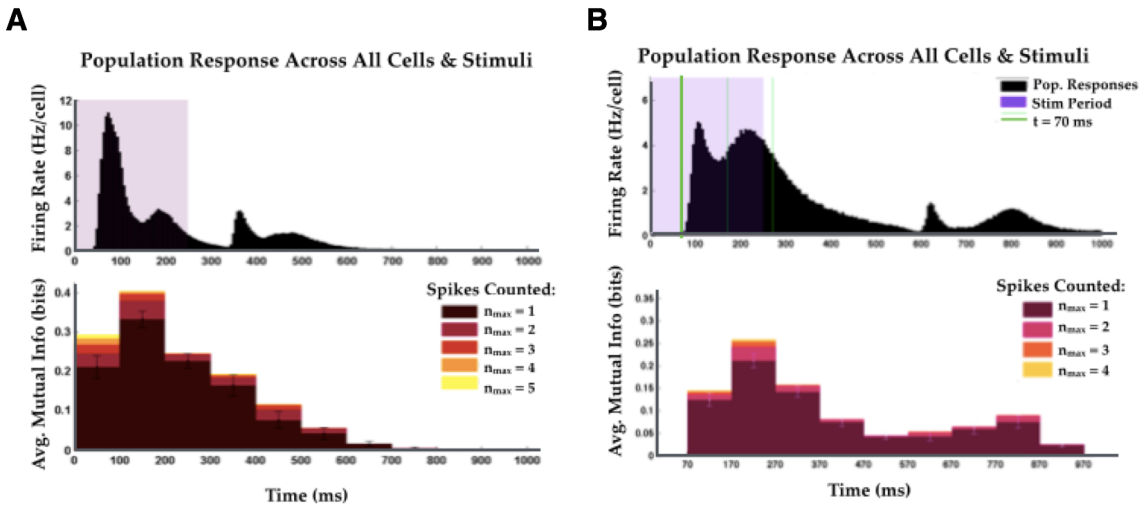

Figure 2.1: **Results from two pilot studies supporting binary response representations.** (A) Results from the first pilot study ($N = 70$ retinal ganglion cells, 80 stimuli). *Top Panel*: Population PSTH across all cells and stimuli. *Bottom Panel*: Mean per cell mutual information (bits) between binary response and stimuli, $I(x_i; S) = -\sum_{x_i=0}^{n_{max}} p(x_i) \log_2 p(x_i) + \frac{1}{|S|} \sum_{k=1}^{|S|} \sum_{x_i=0}^{n_{max}} p(x_i|s_k) \log_2 p(x_i|s_k)$. Stacked bars depict the extra mutual information gained using a set of spike symbols allowing more spikes ($n_{max} > 1$). (B) Results from the second pilot study ($N = 156$ retinal ganglion cells, 16 stimuli); the format is the same as in (A).

2.3 Modeling the population response probability landscape

Fundamental to most conceptual approaches toward investigating neural population coding questions is, as with the single neuron case, characterizing the probability distribution over all neuronal responses. The key additional issue for the multi-neuron scenario however is the nature of correlations among neurons, which fundamentally shapes the probability distribution of population activity.

2.3.1 The scenario

The scenario with which we begin for each project in the subsequent chapters is that we have access to $(T+1)$ total observations of neural (retinal ganglion cell) population responses. As will be consistently used throughout this thesis, we will denote the discrete random variable corresponding with the observed (unweighted) population response during time bin $t \in \mathcal{T} \equiv \{0, 1, \cdots, T\}$ by $X(t)$, which takes values $\vec{x}(t) \equiv (x_i(t); i \in [N])$, where N denotes population size and i indexes neurons in the population. Again, for most cases including the instance of the Tree hidden Markov model (HMM) used in Chapters 3 and 4, we will assume that each neuron's spiking response $x_i(t) \in \{0, 1\}$. We denote the full observed data by

$$\mathbf{x}_0^T \equiv (\vec{x}(t=0), \vec{x}(t=1), \cdots, \vec{x}(t=T)) \in \mathbb{F}_2^{N \times (T+1)} \tag{2.1}$$

and its corresponding RV by X_0^T.

We envision that the system visits the states $\vec{x}$ with some probability $P^*(\vec{x})$. In practice, for network population sizes of about $N > 10$, fundamental sampling issues render it intractable to experimentally determine $P^*(\vec{x})$. This fundamental limitation

necessitates a modeling approach to characterize the joint probability distribution of neuronal activity.

2.3.2 Encoding vs. activity models

Within computational neuroscience, there are two distinct methodologies that have been used to create probabilistic models of network activity. The traditional approach - which includes generalized linear models (GLMs) - attempts to explicitly capture the dependence of neuronal responses on the stimulus. These types of models have been referred to as "encoding models" [124]. The second approach, which we refer to as the "activity model" approach, involves directly modeling the structure of neural population activity, importantly without any reference to the stimulus. An important advantage of activity models is that they correspond to the unsupervised problem actually faced by downstream brain areas, which lack direct access to stimuli in the external environment.

2.3.3 Maximum entropy framework

To date, the most popular class of state-of-the-art activity models used within the field of computational sensory neuroscience has been the class of *maximum entropy (MaxEnt) models* [100, 116]. In analogy to spin configurations in statistical physics, in this short section we will denote the state of the network of ganglion cells during time bin t by a binary vector $\vec{x}(t) \in \{\pm 1\}^N$. The idea underlying the MaxEnt formulation is to represent the true distribution $P^*(\vec{x})$ by the 'least-biased' distribution $P_M(\vec{x})$, in the sense of maximizing the entropy, $S\left[P(\vec{x})\right] \equiv -\int_{\mathbb{R}^N} P(\vec{x}) \log P(\vec{x}) d\vec{x}$, subject to (empirically) known constraints. In practice, with limited sampling, it is possible to experimentally estimate several lowest-order moments of $P^*(\vec{x})$. I.e., in general, a set of measured expectation values $F_\mu = \mathbb{E}\left[f_\mu(\vec{x})\right]$ over $T+1$ samples $\{\vec{x}(0), \cdots, \vec{x}(T)\}$.

In the general MaxEnt formulation, we thus wish to maximize the functional

$$\begin{aligned}\mathcal{L}\left[P(\vec{x})\right] &= -\int P(\vec{x})\log P(\vec{x})d\vec{x} + \lambda_0\left(1-\int P(\vec{x})d\vec{x}\right) + \sum_{\mu=1}^{\eta}\lambda_\mu\left(F_\mu - \int f_\mu(\vec{x})P(\vec{x})d\vec{x}\right) \\ &= \sum_{\mu=0}^{\eta}\lambda_\mu - \int \underbrace{P(\vec{x})\left[\log P(\vec{x}) + \lambda_0 + \sum_{\mu=1}^{\eta}\lambda_\mu f_\mu(\vec{x})\right]}_{\equiv l(\vec{x},P(\vec{x}))} d\vec{x}\end{aligned} \tag{2.2}$$

where $\eta \equiv |\{F_\mu\}|$, and $\{\lambda_\mu\}$ are Lagrange multipliers. Now, we wish to maximize $\mathcal{L}$ with respect to P, so we recall from the calculus of variations that an extremum must correspond with a functional derivative, $\delta\mathcal{L}/\delta P$, satisfying the (generalized) Euler-Lagrange equation:

$$\frac{\delta\mathcal{L}}{\delta P} = \frac{\partial l}{\partial P} - \nabla\frac{\partial l}{\partial \nabla P} = 0 \tag{2.3}$$

(where we assume $l(\vec{x}, P(\vec{x}))$ is twice continuously differentiable). This yields:

$$0 = \underbrace{-\left[\log P(\vec{x}) + \lambda_0 + \sum_{\mu=1}^{\eta}\lambda_\mu f_\mu(\vec{x}) + \frac{1}{P}\cdot P\right]}_{\partial l/\partial P} - \nabla(0) \tag{2.4}$$

$$\implies \log P(\vec{x}) = -(\lambda_0+1) - \sum_{\mu=1}^{\eta}\lambda_\mu f_\mu(\vec{x}) \implies P(\vec{x}) = e^{-(\lambda_0+1)}e^{-\sum_{\mu=1}^{\eta}\lambda_\mu f_\mu(\vec{x})} \tag{2.5}$$

Since λ_0 is determined by the normalization condition of $P(\vec{x})$, we can rewrite Eq. 2.5 as

$$P_M(\vec{x}) = \frac{1}{Z}\exp\left(-\sum_{\mu=1}^{\eta}\lambda_\mu f_\mu(\vec{x})\right) = \frac{1}{Z(\Lambda)}\exp(-\langle\mathbf{\Lambda},\mathbf{f}\rangle) \tag{2.6}$$

where the partition function $Z = \sum_{\vec{x}}\exp\left(-\sum_{\mu=1}^{\eta}\lambda_\mu f_\mu(\vec{x})\right)$, and we have defined $\{f_\mu\} \equiv \mathbf{f}$, $\{\lambda_\mu\} \equiv \mathbf{\Lambda} \in \mathbb{R}^\eta$.

2nd order MaxEnt model

The 2nd order maximum entropy model, also termed the "pairwise" MaxEnt model, was the first MaxEnt model that was introduced to model the probability distribution of RGC population responses [100]. The pairwise model incorporates the following empirically-measured constraints: (i) the mean firing rate $\mathbb{E}_{\vec{x}}[x_i]$ and (ii) two-point correlations $\mathbb{E}_{\vec{x}}[x_i x_{j\neq i}]$. Plugging into Eq. 2.6 yields the solution (which we here denote $P_M^{(2)}$):

$$P_M(\vec{x}) = \frac{1}{Z}\exp\left[-\left(\sum_{i=1}^{N}\lambda_i\pi_i(\vec{x}) + \frac{1}{2}\sum_{i\neq j}\lambda_{ij}\pi_i(\vec{x})\pi_j(\vec{x})\right)\right] \tag{2.7}$$

where the $\frac{1}{2}$ coefficient is due to double counting of the $\binom{N}{2}$ pairs. Equation 2.7 can be rewritten into the more familiar form (for physicists):

$$P_M^{(2)}(\vec{x}) = \frac{1}{Z}e^{-\beta\mathcal{H}(\vec{x})}, \quad \text{where } \mathcal{H} = \left(\sum_{i=1}^{N}h_i x_i + \frac{1}{2}\sum_{i\neq j}J_{ij}x_i x_j\right) \tag{2.8}$$

where $\beta = 1$. Thus, Eq. 2.8, i.e. the solution to the 2nd order MaxEnt model, is exactly the Gibbs measure for a (disordered) Ising spin glass model on a random graph (which is empirically sparse [116]).

kth order MaxEnt model

Similarly, for the kth order model ($k \leq N$), that is, where the constraints are given by (i) $\int_{\vec{x}} P(\vec{x})d\vec{x} = 1$, and (ii) $\{F_\mu\} = \{\mathbb{E}_{\vec{x}} f_\mu(\vec{x})\} = \bigcup_{r=1}^{k}\{\mathbb{E}_{\vec{x}}[\prod_{a=1}^{r} x_{i_a}] : i_1 \neq \cdots \neq i_r\}$, we can plug in the appropriate $\{f_\mu\}$ values into Eq. 2.6 and split the sum into k summands based on the appropriate partition to obtain the Hamiltonian form:

$$P_M^{(k)}(\vec{x}) = \frac{1}{Z}e^{-\beta\mathcal{H}(\vec{x})}, \quad \text{where } \mathcal{H}(\vec{x}) = \sum_{r=1}^{k}\left[\frac{1}{r\,!}\sum_{i_1\neq\cdots\neq i_r}\left(\lambda_{i_1\cdots i_r}\prod_{a=1}^{r}\pi_{i_a}(\vec{x})\right)\right] \tag{2.9}$$

By writing the kth order MaxEnt model solution as a Boltzmann distribution (with $\beta = 1$), note that the Hamiltonian $\mathcal{H}$ corresponds to an underlying random simplicial complex χ, which has a $(k-1)$-simplex for each $\lambda_{i_1 \cdots i_k} \neq 0$ (including a 0-simplex $\forall i \in [N]$). (Whereas the Ising model Hamiltonian of the pairwise model solution corresponds with an underlying random graph, and p-spin glass models for $p \geq 3$ correspond with underlying hypergraphs). Thus, unlike the Ising Hamiltonian, the kth order model Hamiltonians (for $k \geq 3$) can potentially have non-trivial homology groups up to order $(k-1)$, i.e. $H_{k-1}(\chi)$, of the associated (random) simplicial chain complex. In practice, due to sampling limitations for higher-order moments and issues of computational tractability for learning model parameters, kth order MaxEnt models for $k \geq 3$ are not used. However, extensions of the 2nd order MaxEnt model such as the K-Pairwise MaxEnt model [116] have been developed and used.

2.3.4 A latent variable model approach

Latent variable models provide an important tool for the analysis of multivariate data. In the projects described in Chapters 3 though 5, we require an accurate model of the joint distribution of the set of retinal ganglion cell responses. A statistical model specifies the joint distribution of a set of random variables, and it becomes a latent variable model when some of these variables - termed the *latent variables*, which may be discrete (mixture models, hidden Markov models) or continuous (factor-analysis) - are unobservable [11]. Latent variables are typically introduced into a model for two reasons: (1) to reduce dimensionality, and/or (2) the latent quantities figure prominently into the problem and their presence contributes to scientific investigation. Our use of latent variable models in the subsequent chapters will primarily be for the latter reason.

As described in detail in Chapters 3 and 4, one of our early aims was to investigate whether the error-correcting code paradigm is applicable to the retinal population

code (and if so, whether this applicability holds across different stimulus ensemble statistics). Assuming that this paradigm is applicable, then in the inverse to the traditional engineering problem, in our case the 'codewords' have already been chosen by the system and are unobservable to us. From our recording data, we can only observe the (noisy) RGC population responses. Thus, in our specific case, a latent variable model approach - in which latent variables are introduced to correspond to putative 'codewords' - is conducive to enable us to investigate whether or not the error-correcting code paradigm is applicable for the retina.

2.3.5 Parameter estimation via the EM algorithm

In Chapter 3, we will detail our inference procedure for estimating the parameters of what we term the *Tree hidden Markov model* (HMM). For audiences unfamiliar with the EM algorithm for parameter estimation for latent variable models, this section aims to provide some preliminary intuition, by illustrating the general procedure for a simpler latent variable model. Specifically, we will here consider a simple class of mixture models that have emission distributions from the single-parameter exponential family of distributions. As mixture models, these models assume that time bins are statistically independent (unlike the Tree HMM in Chapter 3). For example, for a Bernoulli mixture model, each emission distribution formulates that the probability of a single RGC population response $\vec{x}(t)$ given that the latent state $Z(t) = \alpha$, i.e. $p(\vec{x}(t)|\alpha) \equiv g_\alpha(\vec{x}(t))$, is:

$$g_\alpha\big(\vec{x}(t)\big) = \prod_{i=1}^{N} \big(M_{\alpha,i}\big)^{x_i(t)} \big(1 - M_{\alpha,i}\big)^{(1-x_i(t))} \tag{2.10}$$

Without loss of generality, let $M_\alpha \equiv (M_{\alpha,i}; i \in [N])$ denote the N parameters for the α-th emission distribution. The general approach and algorithm for fitting this class

of model is parallel to that for the Tree HMM used in Chapters 3 and 4, with certain steps being more elaborate for the Tree HMM, as detailed in Chapter 3.

For this scenario, let Θ denote a family of mixture models. Then the likelihood of any one of its members θ is defined as:

$$\begin{aligned} L_{\mathbf{x}_0^T}(\theta) &= \sum_{\alpha=1}^{m} \pi_\alpha \cdot g_\alpha(\mathbf{x}_0^T; M) \\ &= \prod_{t=0}^{T} \sum_{\alpha=1}^{m} \pi_\alpha \cdot g_\alpha\left(\vec{x}(t); M_\alpha\right) \end{aligned} \tag{2.11}$$

where $\theta = (\pi_\alpha, M_{\alpha,i}; \alpha \in [m], i \in [N])$, $\pi_\alpha \in [0,1] \forall \alpha$ and $\sum_{\alpha=1}^{m} \pi_\alpha = 1$, and the second line in Eq. 2.11 follows from the assumption that the observations are exchangeable (i.e. that time bins are statistically independent). Note that we will here assume that the parameter θ is fixed. In log space, this becomes:

$$l_{\mathbf{x}_0^T}(\theta) \equiv \log L_{\mathbf{x}_0^T}(\theta) = \sum_{t=0}^{T} \log \left(\sum_{\alpha=1}^{m} \pi_\alpha \cdot g_\alpha\left(\vec{x}(t); M_\alpha\right) \right) \tag{2.12}$$

Our approach and goal of parameter estimation is to find the maximum likelihood estimate (MLE) of θ:

$$\theta^* = \arg\max_{\theta \in \Theta} l_{\mathbf{x}_0^T}(\theta) \tag{2.13}$$

In the following, we will outline how to do this for the simplified model, to build intuition. First, for convenience, let $q_\theta(\alpha, t)$ denote the joint probability of selecting component α and selecting an observation $\vec{x}(t)$ from that component. That is,

$$q_\theta(\alpha, t) \equiv p(\alpha, \vec{x}(t); \theta) = \pi_\alpha \cdot g_\alpha(\vec{x}(t); M_\alpha) \tag{2.14}$$

This factorization makes explicit the assumption that selecting a latent component is independent of selecting an observation from a component. By Bayes' Theorem,

$$p(\alpha|\vec{x}(t);\theta) = \frac{q_\theta(\alpha,t)}{p(\vec{x}(t);\theta)} = \frac{q_\theta(\alpha,t)}{\sum_{\beta=1}^{m} q_\theta(\beta,t)} \tag{2.15}$$

To derive θ^*, we can make use of Jensen's Inequality, which shows that, since the natural logarithm is a concave function:

$$\log\left(\sum_{\alpha=1}^{m} p_\alpha c_\alpha\right) \geq \sum_{\alpha}^{m} p_\alpha \log c_\alpha \tag{2.16}$$

where the p_α's define a probability distribution. By rearranging terms, $\forall$ values d_α:

$$\begin{aligned} \log\sum_{\alpha=1}^{m} d_\alpha = \log\sum_{\alpha=1}^{m} d_\alpha\frac{p_\alpha}{p_\alpha} &= \log\sum_{\alpha=1}^{m} p_\alpha\left(\frac{d_\alpha}{p_\alpha}\right) \\ &\geq \sum_{\alpha=1}^{m} p_\alpha \log\left(\frac{d_\alpha}{p_\alpha}\right) \end{aligned} \tag{2.17}$$

Thus, applying Jensen's Inequality yields the following bound on the log-likelihood:

$$\begin{aligned} l_{\mathbf{x}_0^T}(\theta) &= \sum_{t=0}^{T}\log\left(\sum_{\alpha=1}^{m}\pi_\alpha\cdot g_\alpha(\vec{x}(t);M_\alpha)\right) \\ &= \sum_{t=0}^{T}\log\left(\sum_{\alpha=1}^{m} q_\theta(\alpha,t)\right) \\ &\geq \underbrace{\sum_{t=0}^{T}\left[\sum_{\alpha=1}^{m} p\left(\alpha|\vec{x}(t);\theta^{(p)}\right)\log\frac{q_\theta(\alpha,t)}{p\left(\alpha|\vec{x}(t);\theta^{(p)}\right)}\right]}_{\equiv b\left(\theta|\theta^{(p)}\right)} \end{aligned} \tag{2.18}$$

since $\sum_{\alpha=1}^{m} p\left(\alpha|\vec{x}(t);\theta^{(p)}\right) = 1$ by assumption. We can then expand, distribute, and split the sum:

$$\begin{aligned} l_{\mathbf{x}_0^T}(\theta) &\geq b\left(\theta|\theta^{(p)}\right) \\ &= \sum_{t=0}^{T}\sum_{\alpha=1}^{m}\left[p(\alpha|\vec{x}(t);\theta^{(p)})\log q_\theta(\alpha,t) - p(\alpha|\vec{x}(t);\theta^{(p)})\log p(\alpha|\vec{x}(t);\theta^{(p)})\right] \\ &= Q\left(\theta|\theta^{(p)}\right) + H\left(\theta^{(p)}|\theta^{(p)}\right) \end{aligned} \tag{2.19}$$

where we have defined the auxiliary function

$$\begin{aligned} Q\left(\theta|\theta^{(p)}\right) &\equiv \sum_{\alpha=1}^{m}\sum_{t=0}^{T} p(\alpha|\vec{x}(t);\theta^{(p)})\log q_\theta(\alpha,t) \\ &= \mathbb{E}\left[\log p\left(Z_0^T,\mathbf{x}_0^T;\theta\right)|\mathbf{x}_0^T;\theta^{(p)}\right] \end{aligned} \tag{2.20}$$

and the function

$$H\left(\theta^{(p)}|\theta^{(p)}\right) \equiv -\mathbb{E}\left[\log p\left(Z_0^T|\mathbf{x}_0^T;\theta^{(p)}\right)|\mathbf{x}_0^T;\theta^{(p)}\right] \tag{2.21}$$

which is the conditional entropy of the complete variable given the observed sequence $\mathbf{x}_0^T$.

Note that an alternative way to derive this is by writing the log-likelihood function as a function of the unobserved variable:

$$l_{\mathbf{x}_0^T}\left(Z_0^T;\theta\right) = \log p(Z_0^T,\mathbf{x}_0^T;\theta) - \log p(Z_0^T|\mathbf{x}_0^T;\theta) \tag{2.22}$$

and taking expectations on both sides with respect to the conditional probability $p(Z_0^T|\mathbf{x}_0^T;\theta^{(p)})$ to obtain:

$$\begin{aligned} l_{\mathbf{x}_0^T}(\theta) &= \mathbb{E}\left[\log p\left(Z_0^T,\mathbf{x}_0^T;\theta\right)|\mathbf{x}_0^T;\theta^{(p)}\right] - \mathbb{E}\left[\log p\left(Z_0^T|\mathbf{x}_0^T;\theta\right)|\mathbf{x}_0^T;\theta^{(p)}\right] \\ &= Q\left(\theta|\theta^{(p)}\right) + H\left(\theta|\theta^{(p)}\right) \end{aligned} \tag{2.23}$$

where the inequality in Eq. 2.19 is satisfied from the inequality [60]

$$H\left(\theta|\theta^{(p)}\right) \geq H\left(\theta^{(p)}|\theta^{(p)}\right) \tag{2.24}$$

Now, Eq. 2.24 shows that $H\left(\theta|\theta^{(p)}\right)$ attains its minimum at $\theta = \theta^{(p)}$, so the gradient $\nabla_\theta\, H(\theta|\theta^{(p)})|_{\theta=\theta^{(p)}} = \vec{0}$. Thus, differentiating both sides of Eq. 2.23 yields

$$\nabla_\theta\, l_{\mathbf{x}_0^T}(\theta)|_{\theta=\theta^{(p)}} = \nabla_\theta\, Q(\theta|\theta^{(p)})|_{\theta=\theta^{(p)}} \tag{2.25}$$

Now, it can be shown that if $Q(\theta|\theta^{(p)}) > Q(\theta^{(p)}|\theta^{(p)})$, then $L_{\mathbf{x}_0^T}(\theta) > L_{\mathbf{x}_0^T}(\theta^{(p)}$. Thus, the best choice - i.e. the one which most increases the likelihood - for the next estimate $\theta^{(p)}$ will be found by maximizing $Q(\theta|\theta^{(p)})$ with respect to θ. Therefore, as is standard, we calculate the new parameters $\theta^{(p+1)}$ by differentiating Q with respect to each parameter, then setting the derivative to 0.

This idea is encapsulated by the batch EM algorithm [28], which is a standard method that we utilize in this work. Specifically, the EM algorithm alternates two steps: an expectation step (E-step) and a maximization step (M-step). During the E-step, $Q\left(\theta|\theta^{(p)}\right)$ is evaluated for the current parameter estimate $\theta^{(p)}$. Then in the subsequent M-step, the updated estimate $\theta^{(p+1)}$ is found, which satisfies $\nabla_\theta Q\left(\theta|\theta^{(p)}\right)|_{\theta=\theta^{(p+1)}} = \vec{0}$. By Eq. 2.25, if this iterative process converges to θ^*, then

$$\nabla_\theta Q\left(\theta|\theta^*\right)|_{\theta=\theta^*} = \vec{0} \implies \nabla_\theta l_{\mathbf{x}_0^T}(\theta)|_{\theta=\theta^*} = \vec{0} \tag{2.26}$$

Thus, the EM algorithm will converge to a stationary point of the likelihood function. Details illustrating how to derive these updates for one version of this simplified model are provided in the Appendix.

2.4 Appendix

2.4.1 Derivation of EM updates for a simplified model

We will here consider generative models within the family Θ of Poisson mixture models. That is, each emission distribution $g_\alpha(\vec{x}(t);\lambda_\alpha) \equiv p(\vec{x}(t)|\alpha;\lambda_\alpha)$ is assumed to be a product of Poisson distributions, each with a rate parameter $\lambda_{\alpha i}$. This formalizes the assumption that neurons in the input population fire independently within a given context α. I.e.

$$\begin{aligned} g_\alpha(\vec{x}(t);\lambda_\alpha) &= \prod_{i=1}^{N} p(x_i(t)|\alpha;\lambda_{\alpha i}) \\ &= \prod_{i=1}^{N} \frac{e^{-\lambda_{\alpha i}} \cdot \lambda_{\alpha i}^{x_i(t)}}{x_i(t)!} \end{aligned} \tag{2.27}$$

Finding each $\lambda_{\alpha i}^{(p+1)}$

For the M-step of the EM algorithm, we want to calculate the estimate for the next iteration, $(p+1)$, of $\lambda_{\alpha i}$, $\forall\ \alpha \in [m]$ and $\forall\ i \in [N]$:

$$\begin{aligned} \frac{\partial Q}{\partial \lambda_{\alpha i}} &= \frac{\partial}{\partial \lambda_{\alpha i}} \sum_{t=0}^{T} \sum_{\beta=1}^{m} p\left(\beta|\vec{x}(t);\theta^{(p)}\right) \log q_\theta(\beta,t) \\ &= \sum_{t=0}^{T} p\left(\alpha|\vec{x}(t);\theta^{(p)}\right) \cdot \frac{\partial}{\partial \lambda_{\alpha i}} \log q_\theta(\alpha,t) \\ &= \sum_{t=0}^{T} p\left(\alpha|\vec{x}(t);\theta^{(p)}\right) \cdot \frac{\partial}{\partial \lambda_{\alpha i}} \log \left[\pi_\alpha \prod_{i=1}^{N} \frac{e^{-\lambda_{\alpha i}}}{x_i(t)!} \cdot \lambda_{\alpha i}\right] \\ &= \sum_{t=0}^{T} p\left(\alpha|\vec{x}(t);\theta^{(p)}\right) \cdot \frac{\partial}{\partial \lambda_{\alpha i}} \left[-\log(x_i(t)!) - \lambda_{\alpha i} + x_i(t) \log \lambda_{\alpha i}\right] \\ &= \sum_{t=0}^{T} p\left(\alpha|\vec{x}(t);\theta^{(p)}\right) \cdot \left(\frac{x_i(t)}{\lambda_{\alpha i}} - 1\right) \end{aligned} \tag{2.28}$$

where we have used the fact that the derivative of the sum of functions is the sum of the derivative, and that the derivative of every term that does not involve $\lambda_{\alpha i}$ is 0.

We now set the derivative to 0:

$$\begin{aligned} 0 &= \sum_{t=0}^{T} p\left(\alpha|\vec{x}(t);\theta^{(p)}\right) \cdot \left(\frac{x_i(t)}{\lambda_{\alpha i}} - 1\right) \\ &= \sum_{t=0}^{T} p\left(\alpha|\vec{x}(t);\theta^{(p)}\right) \left(\frac{x_i(t)}{\lambda_{\alpha i}}\right) - \underbrace{\sum_{t=0}^{T} p\left(\alpha|\vec{x}(t);\theta^{(p)}\right)}_{\equiv S(\alpha)} \end{aligned} \tag{2.29}$$

where we have defined $S(\alpha)$ for convenience. Now, adding $S(\alpha)$ to both sides yields:

$$\begin{aligned} S(\alpha) &= \sum_{t=0}^{T} p\left(\alpha|\vec{x}(t);\theta^{(p)}\right) \cdot \left(\frac{x_i(t)}{\lambda_{\alpha i}}\right) \\ &= \frac{1}{\lambda_{\alpha i}} \cdot \sum_{t=0}^{T} p\left(\alpha|\vec{x}(t);\theta^{(p)}\right) x_i(t) \end{aligned} \tag{2.30}$$

where we have used the fact that $\lambda_{\alpha i}$ is a constant w.r.t. t. Thus:

$$\lambda_{\alpha i}^{(p+1)} = \frac{\sum_{t=0}^{T} x_i(t) \cdot p\left(\alpha|\vec{x}(t);\theta^{(p)}\right)}{S(\alpha)} \tag{2.31}$$

Finding each $\pi_\alpha^{(p+1)}$

The derivation for updating the π_α's is similar to that of $\lambda_{\alpha i}^{(p+1)}$. However, one key difference is the constraint that the π_α's must constitute a probability distribution. To address this constraint, we use a Langrange multiplier δ:

$$\begin{aligned} 0 = \frac{\partial Q}{\partial \pi_\alpha} &= \frac{\partial}{\partial \pi_\alpha} \sum_{t=0}^{T} \sum_{\beta=1}^{m} p\left(\beta|\vec{x}(t);\theta^{(p)}\right) \log q_\theta(\beta, t) + \delta \cdot \left(\sum_{\beta} p_\beta - 1\right) \\ &= \sum_{t=0}^{T} \frac{\partial}{\partial \pi_\alpha} p\left(\alpha|\vec{x}(t);\theta^{(p)}\right) \log q_\theta(\alpha, t) + \underbrace{\frac{\partial}{\partial \pi_\alpha} \delta(\pi_\alpha - 1)}_{=\delta} \\ &= \sum_{t=0}^{T} \frac{\partial}{\partial \pi_\alpha} p\left(\alpha|\vec{x}(t);\theta^{(p)}\right) \left[\log \pi_\alpha + \sum_{i=1}^{N} \log\left(\frac{e^{-\lambda_{\alpha i}}}{x_i(t)!} \cdot \lambda_{\alpha i}^{x_i(t)}\right)\right] + \delta \\ &= \left(\sum_{t=0}^{T} \frac{p(\alpha|\vec{x}(t);\theta^{(p)}}{\pi_\alpha}\right) + \delta \end{aligned} \tag{2.32}$$

By summing over $\alpha \in [m]$, we see that $\delta = -(T+1)$. Thus,

$$\begin{aligned}(T+1) &= \sum_{t=0}^{T} \frac{p\left(\alpha|\vec{x}(t);\theta^{(p)}\right)}{\pi_\alpha} \\ &= \frac{1}{\pi_\alpha} \cdot \sum_{t=0}^{T} p\left(\alpha|\vec{x}(t);\theta^{(p)}\right)\end{aligned} \tag{2.33}$$

Therefore, the optimal update is:

$$\begin{aligned}\pi_\alpha^{(p+1)} &= \frac{1}{(T+1)} \cdot \sum_{t=0}^{T} p\left(\alpha|\vec{x}(t);\theta^{(p)}\right) \\ &= \frac{1}{(T+1)} \cdot S(\alpha)\end{aligned} \tag{2.34}$$

Hence, for the Poisson Mixture Model, the EM algorithm involves repeating the following two steps until the stopping criteria are reached:

- E-Step: Calculate $p\left(\alpha|\vec{x}(t),\theta^{(p)}\right)$ $\forall$ time bins $t \in \{0,1,\cdots,T\}$ (i.e. using the current estimate of the model parameters, $\theta^{(p)}$).
- M-Step: Update
  - $\lambda_{\alpha i}^{(p+1)} \leftarrow \frac{\sum_{t=0}^{T} x_i(t) \cdot p\left(\alpha|\vec{x}(t);\theta^{(p)}\right)}{S(\alpha)}$
  - $\pi_\alpha^{(p+1)} \leftarrow \frac{1}{(T+1)} \cdot S(\alpha)$

Chapter 3

Computational Goals Served by Collective Modes

3.1 Introduction

Here we present on some results from data-driven work that address questions that, in the parlance of Marr [67], pertain to the computational theory level. That is, questions concerning the computational goals of the retinal ganglion cell (RGC) population. First, we provide details concerning the latent variable model - and our corresponding parameter estimation procedure - that we used in combination with other quantitative approaches to probe these questions. Our first main result suggests that a feature basis set emerges at the population level that is distinct from the features encoded by individual retinal ganglion cells. Of particular interest, although it is well-known that *single* RGCs do not exhibit orientation selectivity in their activity, our results suggest that orientation is computed as an emergent feature at the population level. Our second main result, which serves as the motivation for our work in Chapter 4, suggests that another goal of the retina is to compute population "codewords" to enable error correction to combat neural noise.

3.2 Methodology: Tree hidden Markov model

The Tree hidden Markov model (HMM), which was developed by Jason Prentice and first introduced in our work in [87], is a generative latent variable model and specifically a state-based HMM for the full temporal sequence $\mathbf{x}_0^T$ of recorded ganglion cell population responses. This model posits that there is a non-observable sequence of latent states, the RV for which we will denote by $Z_0^T \equiv (Z(t); t \in \mathcal{T})$, where each $Z(t)$ is a categorical variable that takes values $\in [m] \equiv \{1, \cdots, m\}$. The structure of the model, which we describe in detail in the following section, is schematized in Fig. 3.1.

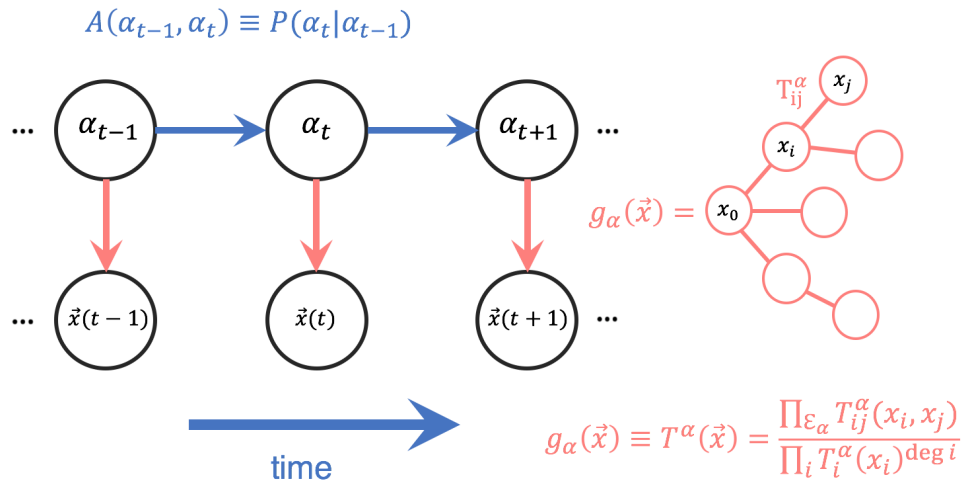

Figure 3.1: **Schematic of the structure of the Tree hidden Markov model (HMM).** The full model includes temporal correlation through a transition probability matrix, $A(\alpha_{t-1}, \alpha_t) \equiv P(\alpha_t|\alpha_{t-1})$, which describes the probability of mode α_t being the latent cause at time step t given that the mode present in the previous time step, $t-1$, was α_{t-1}. The hidden state $Z(t)$, which takes on values $\alpha \in [m]$, evolves probabilistically over time according to the transition matrix, and the observed pattern of spiking at time t, $\vec{x}(t) \in \{0,1\}^N$, has a mode-dependent emission probability, g_α. Shown in red is a schematic representation of the tree graphical structure of each emission distribution.

3.2.1 Model specification

The Tree HMM incorporates temporal correlations between adjacent time bins to model the temporal sequence of observed neural population responses as:

$$P\Big(X_0^T = \mathbf{x}_0^T\Big) = \sum_{\alpha_0^T} \left(\prod_{t=0}^{T} g_{\alpha_t}\Big(\vec{x}(t)\Big) A_{\alpha_{t-1},\alpha_t} \right) \tag{3.1}$$

where $\alpha_0^T \equiv (\alpha_t; t \in \mathcal{T})$ denotes the sequence of latent states (i.e. a possible realization of Z_0^T), where each $\alpha_t \in [m]$, and $g_\alpha(\vec{x}(t)) \equiv p_{X(t)|Z(t)}\left(\vec{x}(t)|\alpha\right)$ denotes the emission distribution for mode α. In practice, we assume that the transition matrix $A \in \mathbb{R}^{m \times m}$, which has entries $A_{\alpha_{t-1},\alpha_t} \equiv p_{Z(t)|Z(t-1)}(\alpha_t|\alpha_{t-1})$, is stationary (i.e. time-independent). Each emission distribution g_α in the model is a tree graphical model (see Ref. [69] and the Appendix).

The model parameters of the Tree HMM, which we will denote by θ, are

$$\theta = (\pi_0, A, \{\theta_{g_\alpha}\}; \alpha \in [m]) \tag{3.2}$$

where π_0 denotes the initial state distribution, i.e.

$$\begin{aligned} &\pi_0 = (\pi_0(\alpha); \alpha \in [m]) \\ &\text{where } \pi_0(\alpha) \equiv P\left(Z(t=0) = \alpha\right) \end{aligned} \tag{3.3}$$

and A denotes the transition matrix as above. Each θ_{g_α} in Eq. 3.2 denotes the model parameters contributed by emission distribution g_α of the Tree HMM. To determine this set θ_{g_α} of contributed parameters, first let $G_\alpha \equiv \Big([N], \mathcal{E}_\alpha\Big)$ denote the underlying graph corresponding with emission distribution g_α. That is, the nodes of the graph G_α are the ganglion cell indices $[N]$ and $\mathcal{E}_\alpha$ denotes the set of edges. Let r_i denote the number of possible responses of neuron i. In practice, we allow the underlying graph to be a forest, so let ρ denote the number of connected components. Then the

number of free parameters contributed by each g_α is:

$$\begin{aligned}
\#\text{ of } g_\alpha \text{ parameters} &= \sum_{(i,j)\in\mathcal{E}_\alpha} r_i r_j - \sum_{i\in[N]} \Big(\deg(i) - 1\Big) r_i - \rho \\
&= \sum_{(i,j)\in\mathcal{E}_\alpha} r_i r_j - \sum_{i\in[N]} r_i \cdot \deg(i) \\
&\quad + \sum_{i\in[N]} r_i - \underbrace{\left(N - \sum_{(i,j)\in\mathcal{E}_\alpha} 1\right)}_{\rho} \\
&= \sum_{(i,j)\in\mathcal{E}_\alpha} (r_i - 1)(r_j - 1) + \sum_{i\in[N]} r_i - N
\end{aligned} \tag{3.4}$$

Since in practice we assume that all neuron responses are binary, $r_i = 2\,\forall i$, hence the total number of free parameters for the entire model is:

$$\begin{aligned}
\text{Total \# of free parameters} &= m(1 + m + N) + \sum_{\alpha=1}^{m} |\mathcal{E}_\alpha| \\
&\leq m^2 + 2mN
\end{aligned} \tag{3.5}$$

where $|\cdot|$ denotes cardinality. If the stationary distribution of the Markov chain is used, then the total number of free parameters reduces to $m(N+1) + \sum_{\alpha=1}^{m} |\mathcal{E}_\alpha| = \mathcal{O}(mN)$.

3.2.2 Maximum likelihood estimation of model parameters

To infer the parameters of the full Tree HMM, we applied a specific form of the EM algorithm for state-based HMMs known as the Baum-Welch algorithm [60]. Note that the second line of Eq. 2.20 is still valid for a hidden Markov model, as alone it makes no assumptions of independence of time bins. For the same reason, Eq. 2.23 also remains valid for a general HMM, and therefore our general argument described in Chapter 2 for the EM algorithm converging to a MLE of the parameters holds for

the Tree HMM. However, the procedure for actually evaluating and maximizing the $Q(\theta|\theta^{(p)})$ function will differ, as we will now detail in the following.

Recall that

$$Q\left(\theta|\theta^{(p)}\right) = \mathbb{E}\left[\log p\left(Z_0^T, \mathbf{x}_0^T; \theta\right)|\mathbf{x}_0^T; \theta^{(p)}\right] \tag{3.6}$$

For our case,

$$p(\alpha_0^T, \mathbf{x}_0^T; \theta) = \pi_0(\alpha_0) g_{\alpha_0}(\vec{x}(0)) \prod_{t=1}^{T} A(\alpha_{t-1}, \alpha_t) g_{\alpha_t}(\vec{x}(t)) \tag{3.7}$$

If we take the logarithm of Eq. 3.7, replace α_0^T with its RV Z_0^T, then take the expectation with respect to $p(\alpha_0^T|\mathbf{x}_0^T; \theta^{(p)}$ and rearrange terms, we obtain that [60]

$$\begin{aligned} Q\left(\theta|\theta^{(p)}\right) = & \overbrace{\sum_{\alpha\in[m]} \log \pi_0(\alpha)\gamma_0^{(p)}(\alpha|\mathbf{x}_0^T) + \sum_{t=1}^{T}\sum_{\alpha,\beta\in[m]} \log A(\alpha,\beta)\xi_{t-1}^{(p)}\left(\alpha,\beta|\mathbf{x}_0^T\right)}^{\equiv Q_1\left(\theta|\theta^{(p)}\right)} \\ & \underbrace{+ \sum_{t=0}^{T}\sum_{\beta\in[m]} \log g_\beta\left(x(t)\right)\gamma_t^{(p)}\left(\beta|\mathbf{x}_0^T\right)}_{\equiv Q_2\left(\theta|\theta^{(p)}\right)} \end{aligned} \tag{3.8}$$

where

$$\gamma_t^{(p)}(\alpha|\mathbf{x}_0^T) \equiv p\left(Z(t) = \alpha|\mathbf{x}_0^T; \theta^{(p)}\right) \tag{3.9}$$

$$\xi_{t-1}^{(p)}(\alpha,\beta|\mathbf{x}_0^T) \equiv p\left(Z(t-1) = \alpha, Z(t) = \beta|\mathbf{x}_0^T; \theta^{(p)}\right) \tag{3.10}$$

M-step

For the Tree HMM, the $\vec{\pi}_0$ and transition matrix A parameters can be optimized analytically, by optimizing $Q_1\left(\theta|\theta^{(p)}\right)$ defined in Eq. 3.8 subject to the normalization constraints, respectively, $\sum_{\alpha\in[m]}\pi_0(\alpha) = 1$ and $\sum_{\beta\in[m]} A(\alpha,\beta) = 1$. Applying the method of Lagrange multipliers, as is standard, yields the following update formulas

for these parameters ($\forall\, \alpha, \beta \in [m]$) for the M-step of the Baum-Welch algorithm:

$$\pi_0(\alpha)^{(p+1)} \leftarrow \gamma_0^{(p)}\left(\alpha|\mathbf{x}_0^T\right) \tag{3.11}$$

$$\begin{aligned} A^{(p+1)}(\alpha,\beta) &\leftarrow \frac{\sum_{t=1}^{T}\xi_{t-1}^{(p)}\left(\alpha,\beta|\mathbf{x}_0^T\right)}{\sum_{\beta=1}^{m}\sum_{t=1}^{T}\xi_{t-1}^{(p)}\left(\alpha,\beta|\mathbf{x}_0^T\right)} \\ &= \frac{\sum_{t=1}^{T}\xi_{t-1}^{(p)}\left(\alpha,\beta|\mathbf{x}_0^T\right)}{\sum_{t=1}^{T}\gamma_{t-1}^{(p)}\left(\alpha|\mathbf{x}_0^T\right)} \end{aligned} \tag{3.12}$$

The remaining parameters define the emission distributions g_α, and are updated by maximizing $Q_2\left(\theta|\theta^{(p)}\right)$ defined in Eq. 3.8. Now, each emission distribution g_α of the Tree HMM is a tree distribution [69], which we will also denote as $T^{(\alpha)}$ in this section, as a reminder. Thus, if we let $G_\alpha\left([N],\mathcal{E}_\alpha\right)$ denote the underlying graph, then by definition each emission distribution takes the form

$$g_\alpha(\vec{x}) \equiv T^{(\alpha)}(\vec{x}) = \frac{\prod_{(i,j)\in\mathcal{E}_\alpha} T_{ij}\left(x_i,x_j\right)}{\prod_{i\in[N]} T_i\left(x_i\right)^{\deg i-1}} \tag{3.13}$$

where deg$i$ denotes the degree of vertex i. Since in our case each neuron's response x_i takes values in $\{0,1\}$, we can see from Eq. 3.4 that the parameters of $T^{(\alpha)}$ are: the choice of tree topology (i.e., the choice of the set of edges, $\mathcal{E}_\alpha$), together with the pairwise distribution $T_{ij}^{(\alpha)}(x_i,x_j)$ on each tree edge $(i,j)\in\mathcal{E}_\alpha$, and the single-cell distribution $T_i^{(\alpha)}(x_i)$ for each neuron. We thus need to maximize $Q_2\left(\theta|\theta^{(p)}\right)$ with respect to these parameters.

Note that we can first rewrite $Q_2\left(\theta|\theta^{(p)}\right)$ as

$$Q_2\left(\theta|\theta^{(p)}\right) = \sum_{\alpha\in[m]}\sum_{t\in\mathcal{T}} P^{(\alpha)}\left(\vec{x}(t)\right)\log T^{(\alpha)}\left(\vec{x}(t)\right) \tag{3.14}$$

where we have defined $P^{(\alpha)}\left(\vec{x}(t)\right) \equiv \gamma_t^{(p)}\left(\alpha|\mathbf{x}_0^T\right)$. Thus, we see that we can update the parameters for each emission distribution $T^{(\alpha)}$ by maximizing the negative cross-

entropy

$$\sum_{t \in \mathcal{T}} P^{(\alpha)}\left(\vec{x}(t)\right) \log T^{(\alpha)}\left(\vec{x}(t)\right) \tag{3.15}$$

Now, Chow and Liu [22] derived an algorithm to solve the problem of finding a single tree distribution T^* that minimizes the Kullback-Leibler divergence from an arbitrary probability distribution P, i.e.

$$\begin{aligned} T^* &= \arg\min_T \sum_{\vec{x}} P(\vec{x}) \log \frac{P(\vec{x})}{T(\vec{x})} \\ &= \arg\max_T \sum_{\vec{x}} P(\vec{x}) \log T(\vec{x}) \end{aligned} \tag{3.16}$$

Thus, for each emission distribution, we can update its parameters by applying the Chow-Liu algorithm [22], which attains an optimum over both structure (i.e. the choice of $\mathcal{E}_\alpha$) and the parameters $\{T_{ij}\}$.

Specifically, define the *dependence graph* under distribution P as the connected graph with edge weights $w(i,j)$ given by

$$w(i,j) = I_{ij} \equiv \sum_{x_i, x_j} P(x_i, x_j) \log \left(\frac{P(x_i, x_j)}{P(x_i) P(x_j)} \right) \tag{3.17}$$

i.e. the mutual information between variables x_i and x_j. Chow and Liu showed that the maximum-weight spanning tree of the dependence graph satisfies Eq. 3.16, where the optimizing parameters T^*_{ij} are equal to the corresponding marginals P_{ij} of the distribution P.

Therefore, we update each emission distribution $T^{(\alpha)}$ via the following procedure:

(1) We compute the empirical pairwise marginals of $P^{(\alpha)} \equiv \gamma_t^{(p)}$:

$$T_{ij}^{(\alpha)}\left(x_i = \sigma, x_j = \sigma'\right) \leftarrow \frac{\sum_{t \in \mathcal{T}} \gamma_t^{(p)}\left(\alpha | \mathbf{x}_0^T\right) \cdot \delta\left(x_i(t) = \sigma\right) \cdot \delta\left(x_j(t) = \sigma'\right)}{\sum_{t \in \mathcal{T}} \gamma_t^{(p)}\left(\alpha | \mathbf{x}_0^T\right)} \tag{3.18}$$

where δ denotes the Kronecker delta.

(2) $\forall i \neq j \in [N]$, we then compute

$$I_{ij}^{(\alpha)} \leftarrow \sum_{\sigma,\sigma'} T_{ij}^{(\alpha)}(x_i = \sigma, x_j = \sigma') \log \left(\frac{T_{ij}^{(\alpha)}(x_i = \sigma, x_j = \sigma')}{T_i^{(\alpha)}(x_i = \sigma) T_j^{(\alpha)}(x_j = \sigma')} \right) \tag{3.19}$$

where $T_i^{(\alpha)}(x_i) = \sum_{\sigma'} T_j^{(\alpha)}(x_j = \sigma')$.

(3) To select the optimal edge set, $\mathcal{E}_\alpha$, we then find the maximum-weight spanning tree of the dependence graph. I.e. the choice of edges that maximizes the sum of the weights, where each edge weight $w(i,j) = I_{ij}^{(\alpha)}$ computed in the previous step. We find the maximum-weight spanning tree using Kruskal's algorithm [125], which (in brief) is a greedy algorithm that iteratively selects the edge with highest weight, out of all edges that would not form a loop when added to the currently-estimated tree.

E-step

To compute the M-step parameter updates in Eqs. 3.11, 3.12 and 3.18 derived above, we need to compute the $\xi_{t-1}^{(p)}\left(\alpha, \beta | \mathbf{x}_0^T\right)$ and $\gamma_t^{(p)}\left(\alpha | \mathbf{x}_0^T\right)$ posterior terms. For a state-based hidden Markov model, an alternative expression for these terms is [60]

$$\xi_{t-1}^{(p)}\left(\alpha_{t-1}, \alpha_t | \mathbf{x}_0^T\right) = \frac{F^{(p)}\left(\alpha_{t-1}, t-1\right) A^{(p)}\left(\alpha_{t-1}, \alpha_t\right) g_{\alpha_t}^{(p)}\left(\vec{x}(t)\right) B^{(p)}\left(\alpha_t, t\right)}{L_{\mathbf{x}_0^T}\left(\theta^{(p)}\right)} \tag{3.20}$$

$$\gamma_t^{(p)}\left(\alpha | \mathbf{x}_0^T\right) = F^{(p)}(\alpha, t) \cdot B^{(p)}(\alpha, t) \tag{3.21}$$

where the *forward variables* and *backward variables*

$$F^{(p)}(\alpha, t) \equiv p\left(Z(t) = \alpha, X_0^t = x_0^t; \theta^{(p)}\right) \tag{3.22}$$

$$B^{(p)}(\alpha, t) \equiv p\left(x_{t+1}^T | Z(t) = \alpha; \theta^{(p)}\right) \tag{3.23}$$

can be computed via the forward-backward algorithm [60]. Specifically, together with the boundary conditions $F(\alpha, 0) = g_\alpha(\vec{x}(0)) \cdot \pi_0(\alpha)$ and $B(\alpha, T) = 1 \; \forall \, \alpha \in [m]$, the forward variables can be computed recursively via a forward pass through the data according to

$$F^{(p)}(\alpha, t) = g_\alpha^{(p)}(\vec{x}) \sum_{\beta=1}^{m} A^{(p)}(\alpha, \beta) \cdot F^{(p)}(\beta, t-1) \tag{3.24}$$

and the backward variables can be computed recursively via a backward pass through the data according to

$$B^{(p)}(\alpha, t) = \sum_{\beta=1}^{m} g_\beta^{(p)}(\vec{x}(t+1)) \cdot A^{(p)}(\alpha, \beta) \cdot B^{(p)}(\beta, t+1) \tag{3.25}$$

Note that if the objective is solely to compute the model parameter updates, then the likelihood does not need to be computed, since the denominator in Eq. 3.20 cancels out in the update equations.

As an implementation detail, we note that naive application of these equations is numerically unstable, as they tend to drive the forward and backward variables toward zero. We therefore instead compute normalized version of $F^{(p)}(\alpha, t)$ and $B^{(p)}(\alpha, t)$, applying the normalization after each time bin. These ideas are organized more concretely in the corresponding pseudocode for our fitting algorithm, which is provided in the following section.

Pseudocode

Here we provide pseudocode for the parameter estimation algorithm. Note that the complete algorithm adds to this a regularization step, as detailed in the subsequent sections.

Algorithm 1 Baum-Welch algorithm for the Tree HMM

Initialize parameters $\theta^{(0)} = \left(\vec{\pi}_0^{(0)}, A^{(0)}, \{g_\alpha^{(0)}\}\right)$
for iter $p = 0 \dots max_iter$ do
▷ Begin E-step
  Set $F(\alpha, 0) \leftarrow g_\alpha^{(p)}(\vec{x}(0))\pi_0^{(p)}(\alpha) \ \forall \alpha \in [m]$ ▷ Begin forward algorithm
  for $t = 1 \dots T$ do
    for $\alpha = 1 \dots m$ do
      $F(\alpha, t) \leftarrow g_\alpha^{(p)}(\vec{x}(t)) \cdot \sum_\beta F(\beta, t-1) A^{(p)}(\alpha, \beta)$
    $F(\alpha, t) \leftarrow F(\alpha, t) / \sum_{\alpha'} F(\alpha', t)$ ▷ Normalize

  Optional: Compute likelihood $L_{\mathbf{x_0^T}}^{(p)} = \sum_\alpha F(\alpha, T)$

  Set $B(\alpha, T) \leftarrow 1 \ \forall \alpha \in [m]$ ▷ Begin backward algorithm
  for $t = T-1 \dots 0$ do
    for $\alpha = 1 \dots m$ do
      $B(\alpha, t) \leftarrow \sum_\beta A^{(p)}(\alpha, \beta) g_\beta^{(p)}(\vec{x}(t+1)) B(\beta, t+1)$
    $B(\alpha, t) \leftarrow B(\alpha, t) / \sum_{\alpha'} B(\alpha', t)$ ▷ Normalize
  Compute $\Gamma(\alpha, t) \equiv \gamma_t^{(p)}\left(\alpha | \mathbf{x}_0^T\right) \leftarrow F(\alpha, t) B(\alpha, t)$
  $\Gamma(\alpha, t) \leftarrow \Gamma(\alpha, t) / \sum_{\alpha'} \Gamma(\alpha', t)$
  Compute $\Xi(\alpha, \beta, t) \equiv \xi_{t-1}^{(p)}\left(\alpha, \beta | \mathbf{x}_0^T\right) \leftarrow F(\alpha, t-1) A^{(p)}(\alpha, \beta) g_\beta^{(p)}(\vec{x}(t)) B(\beta, t)$
  $\Xi(\alpha, \beta, t) \leftarrow \Xi(\alpha, \beta, t) / \sum_{\alpha', \beta'} \Xi(\alpha', \beta', t)$
▷ Begin M-step
  $\pi_0^{(p+1)}(\alpha) \leftarrow \Gamma(\alpha, 0)$
  $A^{(p+1)}(\alpha, \beta) \leftarrow \sum_{t=1}^T \Xi(\alpha, \beta, t) / \sum_{t=1}^T \Gamma(\alpha, t-1)$
  $g_\alpha^{(p+1)}(X_i = \sigma, X_j = \sigma') \leftarrow \sum_{t=0}^T \Gamma(\alpha, t) \delta(x_i(t), \sigma) \delta(x_j(t), \sigma') / \sum_{t=0}^T \Gamma(\alpha, t)$
  $I_{ij}^\alpha \leftarrow \sum_{\sigma, \sigma'} g_\alpha^{(p+1)}(X_i = \sigma, X_j = \sigma') \log \frac{g_\alpha^{(p+1)}(X_i = \sigma, X_j = \sigma')}{g_\alpha^{(p+1)}(X_i = \sigma) g_\alpha^{(p+1)}(X_j = \sigma')}$
  $\mathcal{E}_\alpha^{(p+1)} \leftarrow \mathtt{Kruskal}(\mathtt{I}^\alpha)$ ▷ Get tree edges

Parameter initialization

For all analyses, the same procedure was used to initialize the parameters. Specifically, the initial transition matrix, $A^{(0)} \in \mathbb{R}^{m \times m}$, was set to a uniform distribution in each column, and $\vec{\pi}_0^{(0)}$ was likewise set to a uniform distribution. All tree edges were initialized to zero, i.e. $\mathcal{E}_\alpha^{(0)} = \emptyset$. The remaining parameters of the emission distributions $g_\alpha^{(0)}$ are the mode-dependent firing probabilities for each latent mode and neuron. It is important to choose these differently for each mode, because if

two modes have identical initial parameters the fitting algorithm will never separate them. At the same time, if the initial values of any modes are too extremely different from the data, compared to other modes, they will have their probability driven to zero and effectively drop out of the model. We therefore generated the initial mean vectors randomly, but with only a small amount of variability. Specifically, each component of each mean vector was drawn randomly and independently from the uniform distribution over $[0.45, 0.55]$.

Selecting the optimal latent dimensionality

To select the number of latent states m, also referred to in the literature as the *latent dimensionality* [61], we carried out an n-fold cross-validation procedure in which we randomly generated n partitions of the data into training and test sets. The latent dimensionality was then chosen to maximize the test set likelihood. The test set likelihood, which we also refer to as the cross-validated (CV) likelihood, can be computed via e.g. the forward algorithm [60], where the forward variables are calculated using the parameters θ^* obtained by fitting the Tree HMM to the training set, and the forward pass is implemented over the test data. In practice, we chose $n = 2$.

3.2.3 $L1$ regularization to mitigate overfitting

To prevent overfitting, for all analyses we imposed a regularization that penalizes the addition of tree edges for each emission tree distribution. As derived in [87], this was implemented by estimating the parameters for each emission distribution that maximize the following function (since these results hold for each latent mode individually, below we simplify the previous notation by dropping the α sub- and

superscripts):

$$\mathcal{L} = \Gamma + \sum_{i=1}^{N} h_i \tilde{m}_i + \sum_{(i,j)\in\mathcal{E}} J_{ij}\tilde{C}_{ij} - \eta \sum_{(i,j)\in\mathcal{E}} |J_{ij}| \tag{3.26}$$

where $\tilde{m}_i^\alpha = \sum_t \gamma_t(\alpha|\mathbf{x}_0^T)x_i(t)/\sum_t \gamma_t(\alpha|\mathbf{x}_0^T)$ and $\tilde{C}_{ij}^\alpha = \sum_t \gamma_t(\alpha|\mathbf{x}_0^T)x_i(t)x_j(t)/\sum_t \gamma_t(\alpha|\mathbf{x}_0^T)$ using the notation of the previous section, η denotes the constant regularization parameter, and

$$\Gamma = \sum_{i=1}^{N} \log T_i(0) + \sum_{(i,j)\in\mathcal{E}} \log \frac{T_{ij}(0,0)}{T_i(0)T_j(0)} \tag{3.27}$$

$$h_i = (\deg i - 1)\log \frac{T_i(0)}{T_i(1)} + \sum_{j\in\mathcal{N}(i)} \log \frac{T_{ij}(1,0)}{T_{ij}(0,0)} \tag{3.28}$$

$$J_{ij} = \log \frac{T_{ij}(0,0)T_{ij}(1,1)}{T_{ij}(0,1)T_{ij}(1,0)} \tag{3.29}$$

where $\mathcal{N}(i)$ denotes the set of neurons that are direct neighbors of i on the tree. If we let $m_i \equiv T_i(1)$ and $C_{ij} \equiv T_{ij}(1,1)$, then note that the full distributions $T_i(x_i)$ and $T_{ij}(x_i, x_j)$ defining each emission distribution may be expressed in terms of these quantities. It was shown in [87] that the optimization criteria for Eq. 3.26 are

$$\begin{aligned} m_i &= \tilde{m}_i \\ C_{ij} &= \begin{cases} \tilde{C}_{ij} - \eta, & \widetilde{Cov}(x_i, x_j) > \eta \\ \tilde{C}_{ij} + \eta, & \widetilde{Cov}(x_i, x_j) < -\eta \\ \tilde{m}_i\tilde{m}_j & |\widetilde{Cov}(x_i, x_j)| \le \eta \end{cases} \end{aligned} \tag{3.30}$$

where $\widetilde{Cov}$ denotes the empirical covariance. To incorporate the regularization step derived in [87] into the fitting algorithm, we must modify line 24 of the pseudocode for Algorithm 1, by instead using Eq. 3.30 to calculate m_i^α for all neurons and C_{ij}^α for all neuron pairs, then using these parameters to obtain $T_{ij}^\alpha(x_i, x_j)$.

3.2.4 Computing the MAP latent sequence estimate

After fitting the Tree HMM (i.e., after computing the maximum likelihood estimates of the model parameters as detailed in Section 3.2.2), for various analyses we wanted to be able to identify the most probable sequence of latent modes,

$$\begin{aligned} \hat{\alpha}_0^T &\equiv \arg\max_{\alpha_0^T} p\left(\alpha_0^T | \mathbf{x}_0^T\right) \\ &= \arg\max_{\alpha_0^T} p\left(\alpha_0^T, \mathbf{x}_0^T\right) \end{aligned} \tag{3.31}$$

where the probability distribution in the first line represents the posterior over the full latent state sequences, conditioned on all of the data. I.e., we wanted to use the Tree HMM as a generative classifier. Note that Eq. 3.31 assumes that the model parameters θ are known (fixed). To solve Eq. 3.31 - a solution $\hat{\alpha}_0^T$ of which is referred to as a *MAP sequence estimate* - we use the Viterbi algorithm [60], which is standard, but we review it here for completeness.

The Viterbi algorithm involves computation of the *auxiliary variables*

$$\widetilde{F}(\alpha, t) \equiv \max_{\alpha_0^{t-1}} p\left(Z_0^{t-1} = \alpha_0^{t-1}, Z(t) = \alpha, X_0^t = \mathbf{x}_0^t\right) \tag{3.32}$$

for $\alpha \in [m]$, $1 \leq t \leq T$. Note that the auxiliary variable $\widetilde{F}(\alpha, t)$ is similar to the forward variable $F(\alpha, t)$ in Eq. 3.22, which can be rewritten as

$$F(\alpha, t) = \sum_{\alpha_0^{t-1}} p\left(Z_0^{t-1} = \alpha_0^{t-1}, Z(t) = \alpha, X_0^t = \mathbf{x}_0^t\right) \tag{3.33}$$

I.e. the difference is that the summation in defining $F(\alpha, t)$ is replaced by max in defining $\widetilde{F}(\alpha, t)$. Analogously, each auxiliary variable is computed recursively through

a forward pass over the data, by applying the iterative formula:

$$\widetilde{F}\left(\alpha,t\right)=g_{\alpha}(\vec{x}(t))\cdot\max_{\beta\in[m]}A(\beta,\alpha)\widetilde{F}\left(\beta,t-1\right) \tag{3.34}$$

with the initial value $\widetilde{F}(\alpha,0)=F(\alpha,0)\equiv P\left(Z(0)=\alpha,X(0)=\vec{x}(0)\right)=\pi(\alpha)g_{\alpha}(\vec{x}(0))$. During the same forward pass, we also cache the latent state from which each surviving path emanates:

$$S(\alpha_t,t-1)=\arg\max_{\beta\in[m]}\widetilde{F}(\beta,t-1)A(\beta,\alpha_t) \tag{3.35}$$

This function gives the most probable value of the latent mode identity for time $t-1$, once we know the most probable mode identity for time t.

The full sequence is therefore obtained in a backward pass, by setting $\hat{\alpha}_T=\arg\max_{\beta\in[m]}\widetilde{F}(\beta,T)$ and then iterating $\hat{\alpha}_t=S(\hat{\alpha}_{t+1},t)$, as summarized below.

Algorithm 2 Viterbi algorithm

for each latent state $\alpha=1\ldots m$ **do** ▷ Initializations
  $\widetilde{F}(\alpha,0)\leftarrow\pi_0(\alpha)\cdot g_{\alpha}(\vec{x}(0))$
  $S(\alpha,0)\leftarrow 1$
for time step $t=1,\cdots,T$ **do** ▷ Begin forward pass
  for each latent state $j=1,\cdots,m$ **do**
    $\widetilde{F}(j,t)\leftarrow g_j(\vec{x}(t))\cdot\max_{k\in[m]}\left[A(k,j)\cdot\widetilde{F}(k,t-1)\right]$
    $S(j,t)\leftarrow\arg\max_{k\in[m]}\left[\widetilde{F}(k,t-1)\cdot A(k,j)\right]$
$\hat{\alpha}_T\leftarrow\arg\max_{k\in[m]}\widetilde{F}(k,T)$ ▷ Begin backward pass
for $t=T-1,T-2,\cdots,0$ **do**
  $\hat{\alpha}_t\leftarrow S(\hat{\alpha}_{t+1},t)$

Note that if $p(Z_0^T=\alpha_0^T)$ is uniform prior to the occurrence of the specific $\mathbf{x}_0^T$, then by Baye's rule the sequence of latent states $(\hat{\alpha}_T,\hat{\alpha}_{T-1},\cdots,\hat{\alpha}_t,\cdots,\hat{\alpha}_0)$ obtained via the Viterbi algorithm will also correspond with the maximum likelihood sequence estimate (MLSE). However, the Viterbi algorithm does not require this assumption.

3.3 Modes encode a distinct feature basis set that includes orientation

As with any sensory area, one of the central questions regarding the computational goals of the retina is: What features of external stimulus space are encoded by the retinal ganglion cell (RGC) population? Within computational neuroscience and specifically concerning population analysis of neural data, one popular approach to address this question is to apply decoding algorithms [90]. Many previous decoding approaches used in the field assume knowledge of the stimulus (i.e. are supervised methods, including e.g. nearest-neighbor decoders [89], support vector machines [121] and Bayesian decoders [86]). In contrast, we would like to take the viewpoint of downstream brain areas (e.g. primary visual cortex) that lack direct access to external stimuli. That is, we are interested in an unsupervised decoding approach.

As an *activity model* (i.e. an unsupervised model that does not explicitly account for the stimulus), and moreover as a generative model for which decoding can be performed via the Viterbi algorithm (as described in the preceding section), the Tree hidden Markov model offers one such platform for an unsupervised decoding approach to study this question. Moreover, as reported in [87], we found that the performance of the Tree HMM, as assessed by the log-likelihood evaluated on a 50% held-out test set, compared favorably to previous state-of-the-art activity models, including the K-pairwise maximum entropy model of [116] and the reliable interactions model of [42]. We thus sought to apply an unsupervised decoding approach based on the Tree HMM and Viterbi algorithm to investigate the stimulus features encoded by the RGC population activity. Specifically, our first goal was to characterize the spatial receptive fields of the latent states identified by the Tree HMM fit to RGC population response data, which we refer to throughout as "latent modes" or "collective modes."

3.3.1 Mode-triggered average analysis

To do this, we trained the Tree HMM on the white-noise checkerboard dataset (see *Experimental Methods* in Chapter 2), which was well described by 50 latent modes. We then estimated the most probable (MAP) sequence of latent modes by applying the Viterbi algorithm using the fit model parameters. For each latent mode α, we then estimated its corresponding spatiotemporal receptive field (RF) by computing the "mode-triggered average" (MTA) stimulus obtained in the presence of the white noise checkerboard stimulus. That is, by averaging all stimuli preceding a time bin in which the latent mode was active (as determined by the Viterbi algorithm). This is analogous to the well-established spike-triggered average procedure used to estimate the receptive fields of single neurons [20]. Based on the temporal profiles, we classified cells into ON- and OFF-type.

For analysis of RF shape, we extracted the spatial component of the MTA using singular value decomposition (SVD) across time. Each mode spatial RF was then smoothed by convolving with a two-dimensional Gaussian kernel having a standard deviation of 1 checker, and interpolated by a factor of 4. The center of each mode's spatial RF profile was identified as the spatial position corresponding to the extremal (largest absolute) RF value. To further mitigate the effects of noise, we then restricted all of the following analyses to the 29×29 interpolated checker region (i.e. 7.25×7.25 in original checker units) centered on the spatial RF center. For all modes, this window was sufficiently large enough to encompass the border, estimated visually and then verified post-hoc, of the region contiguous around the RF center outside of which RF magnitudes were below approximately 70% of the extremal value evoked in the RF center.

3.3.2 Characterization of mode spatial RFs

For each of the 50 Tree HMM modes, we then fit a $2D$ Difference of Gaussians (DoG) model via the method of least squares (using the genetic algorithm as our optimization algorithm) to the corresponding smoothed spatial RF, given by:

$$f_{\text{DoG}}(\vec{x}) = K_1 \cdot g(\vec{x}; \vec{\mu}_1, \Sigma_1) - K_2 \cdot g(\vec{x}; \vec{\mu}_2, \Sigma_2) \tag{3.36}$$

where here $\vec{x} \in \mathbb{R}^2$ denotes spatial position, the constant scaling parameters $K_1, K_2 \in (-\infty, \infty)$, g denotes the bivariate Gaussian distribution, $\vec{\mu}_1$ and $\vec{\mu}_2$ denote the means, and Σ_1, Σ_2 denote the covariance matrices of the respective Gaussian distributions. Note that we fit an unconstrained form of the DoG model in that we allowed the means to be spatially shifted, i.e. we allowed for $\vec{\mu}_1 \neq \vec{\mu}_2$, and placed no other constraints on the parameters. For each mode spatial RF, we fit the above $2D$ DoG model a total of 100 times, using different randomly chosen initial parameter values each time. We then chose the best model fit as the one which minimized the mean squared error (MSE). These parameters were used to partition the spatial receptive fields of the latent modes into four qualitative types: 1) intersection, 2) union, 3) oriented dipole, and 4) independent (Fig 3.2, which is taken directly from [87]).

To quantify these results further, we first identified if there were any modes for which the spatial RF was predominantly noise, so that we could exclude these from subsequent analyses. By "predominantly noise", we mean that the spatial RF lacks the fundamental property of containing at least one extremum with an amplitude significantly above the baseline noise level, which we would expect a "good" spatial RF to exhibit. Note that the unconstrained $2D$ DoG model can capture both the case of a spatial RF having one extremum, and the case in which it has two extrema. If the $2D$ DoG model is a good fit in the sense that the χ^2 value equals the number of degrees of freedom, ν, and the spatial RF for each point $\vec{x}$ is independently distributed according

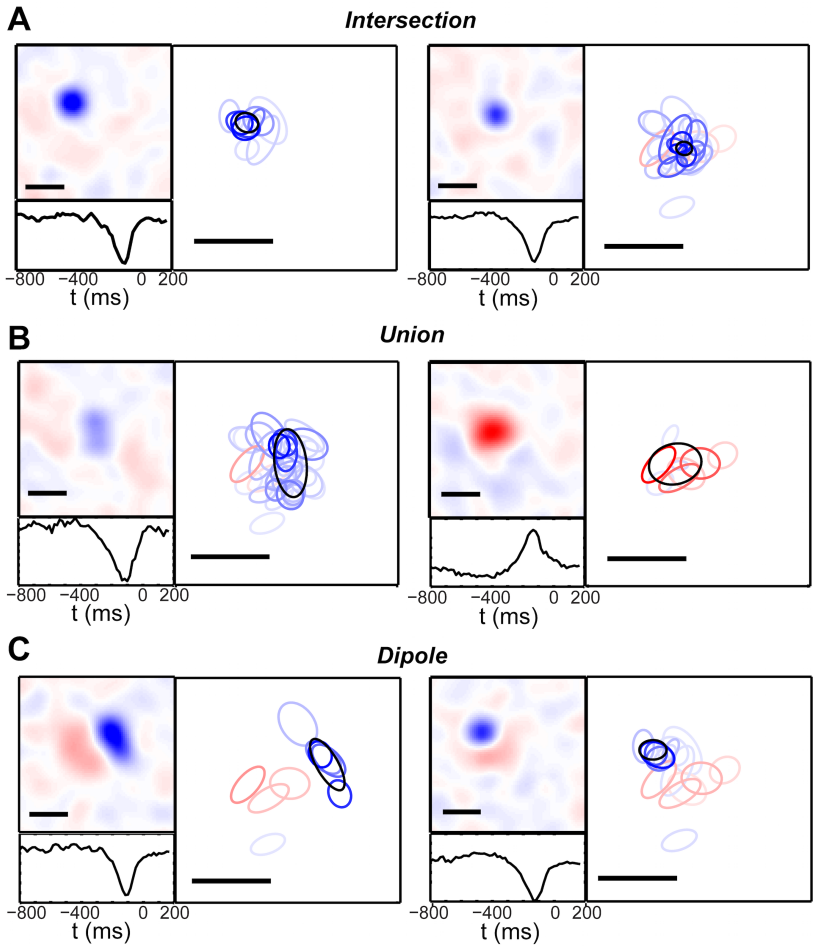

Figure 3.2: **Qualitative categories of mode RFs.**
Figure by Jason Prentice [87]. Shown are examples of types of spatial receptive field of latent modes, measured as the mode-triggered stimulus average during a white noise checkerboard stimulus. In each panel, the left sub-panel exhibits the mode's spatial receptive field (top) and temporal profile (bottom). The right sub-panel shows a 1-standard deviation contour of the best Gaussian fit to the mode receptive field (black ellipse) compared to the same fit to individual cells' spike-triggered average receptive field (red: ON-type cells, blue: OFF-type cells). The saturation of the ellipse's color represents the cell's mode-dependent spiking probability $m_{i\alpha}$. Scale bars = 5 checks. We observed three qualitative categories of mode receptive field: (A) Intersection. These had highly localized, compact, and symmetrical receptive fields located at the intersection of many single-cell fields. Half the modes were of the intersection type. (B) Union. These were larger and more elongated, were spanned by several individual receptive fields, and occurred as both OFF-type (left) and ON-type (right). (C) Dipole. These received a strong contribution from both ON- and OFF-cells, with the two populations forming separate clusters with low overlap. The overall mode receptive field therefore consisted of an oriented pair of ON- and OFF-subfields.

to a normal distribution with the same noise variance - i.e. each $y_j \sim \mathcal{N}(\mu_j, \sigma_f^2)$, where

y_j denotes the observed MTA value at point $\vec{x}_j$ - then:

$$\sigma_f^2 = \frac{1}{\nu} \sum_j \left(y_i - f_{\mathrm{DoG}}(\vec{x}_j; \theta) \right)^2 \equiv \mathrm{MSE} \tag{3.37}$$

where θ denotes the model parameters fit via least squares. That is, under these assumptions we can estimate the variance of the noise as equal to the mean squared error.

Based on this idea, we excluded a mode spatial RF if the amplitude of its greatest extremum did not exceed the estimated standard deviation of the noise for that mode, σ_f, by at least a factor of Θ (where in practice we chose $\Theta = 4.5$). That is, any mode α which had $\frac{|y_\alpha|}{\sigma_f} < \Theta$ was excluded, where y_α denotes the largest extremum of the observed spatial RF for mode α. As seen in panel B of 3.3 Fig, one of the 50 modes was excluded based on this criterion: mode 35. This result was consistent with visual inspections.

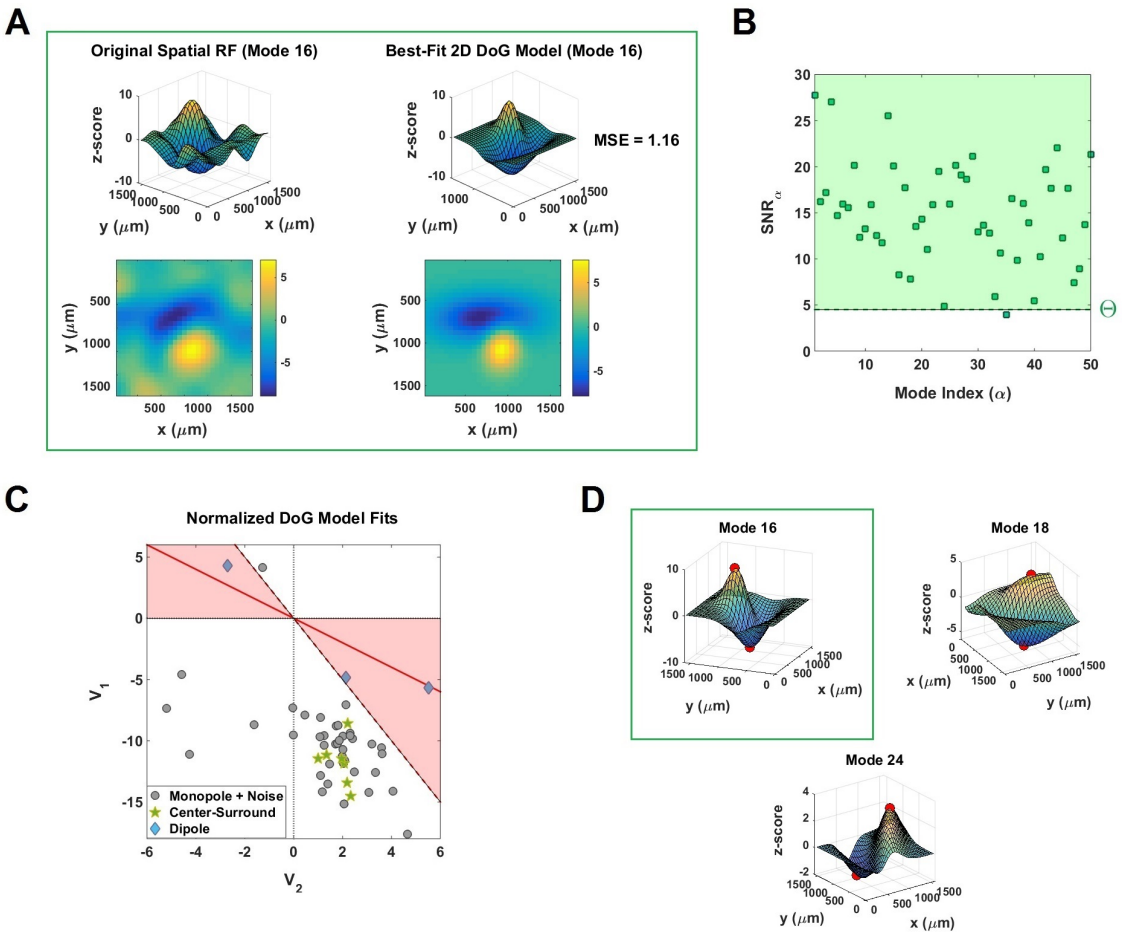

Figure 3.3: **Identification of dipole-type collective mode spatial receptive fields.** (**A**) *Left*: Example smoothed spatial RF of a collective mode, measured by performing SVD on the mode-triggered average (MTA) stimulus. Both the $3D$ surface (top panel) and heat map representation (bottom panel) are shown. *Right*: The corresponding best-fit unconstrained $2D$ Difference of Gaussians (DoG) model for this example mode; the corresponding mean-squared error (MSE) is reported at right. (**B**) Assessment of noise modes. The ratio $\text{SNR}_\alpha \equiv \frac{|y_\alpha|}{\sigma_f}$ (y-axis) for each Tree HMM mode (x-axis); y_α denotes the largest extremum of the observed spatial RF. Green dashed line denotes the chosen threshold value $\Theta = 4.5$. Ratio values above this threshold criterion are represented by the green shaded region. (**C**) Assessment of dipole-type mode spatial RFs. Each point represents one of the 49 non-noise modes. For ease of visualization, each mode's modeled spatial RF profile has been normalized. The extrema of the best-fit $2D$ DoG model were identified via an automatic procedure. Shown is the value of the extremum with the largest amplitude (denoted "V_1", y-axis) vs. the value of the extremum with the second-largest amplitude (denoted "V_2", x-axis). Red solid line denotes the line $y = -x$, and the red dashed line denotes the bound for the chosen threshold criterion $\frac{x}{y} < -0.4$. Red shaded region denotes the region in which this criterion is met; modes within this region were categorized as having dipole-type spatial RFs (blue diamonds). For the remaining modes, the means of the two Gaussians of the best-fit $2D$ DoG model were either nearly identical as in a center-surround organization (green stars), or the second Gaussian merely contributed a negligible "noise blip" (gray circles). (**D**) Spatial RF profile, as modeled by the best-fit DoG model (z-score shown on each z-axis), for the three identified dipole-type modes. Red dots denote the first two largest extrema for each mode.

Oriented dipole analysis

Orientation selectivity is a well-known property of single vertebrate primary visual cortex (V1) neurons [104], but it is not a property of individual retinal ganglion cells. Given that orientation selectivity emerges in the visual pathway, we next asked if signatures of orientation selectivity were present at the *population level* of the RGC code. In this vein, we next investigated whether there were any mode spatial RFs that could be characterized as an oriented dipole, which is indicative of orientation selectivity. Qualitatively, a dipole RF has the fundamental characteristic of containing two extrema of opposite sign, where both extrema are significant.

To quantify the presence or absence of this property, we compared the values of the two largest extrema of the best-fit DoG model. We will denote these values by V_1 (for the value of the extremum having the largest amplitude) and V_2 (for the value of the extremum having the second-largest amplitude).[1] We then classified a given mode α as having a dipole-type spatial RF if its corresponding ratio $\frac{V_2}{V_1} < \Phi < 0$, where in practice we chose the threshold value $\Phi = -0.4$. Note that in practice, for ease of visualization when compiling the global results over all 49 non-noise modes (shown in panel C of 3.3 Fig), we normalized each mode's spatial RF as modeled by the fit DoG model. Specifically, each spatial RF value was rescaled as:

$$\text{Rescaled } f_{\text{DoG}}(\vec{x}_j;\theta) = \frac{f_{\text{DoG}}(\vec{x}_j;\theta)}{\sum_k |f_{\text{DoG}}(\vec{x}_k;\theta)|} \tag{3.38}$$

where $|\cdot|$ denotes the absolute value. Note that we chose this form of rescaling so as to preserve the relative distances between the negative and positive outliers.

As seen in panel C of 3.3 Fig, the above-described criterion yielded three modes (mode 16, 18 and 24) with identified dipole-type RFs. The corresponding fit DoG

[1] Note that if the two bivariate Gaussians do not significantly overlap and $K_1, K_2 \neq 0$, then the values of the two largest extrema will be $f_{\text{DoG}}(\vec{\mu}_1) = K_1 \cdot g(\vec{\mu}_1;\vec{\mu}_1,\Sigma_1) - K_2 \cdot g(\vec{\mu}_1;\vec{\mu}_2;\Sigma_2)$ and $f_{\text{DoG}}(\vec{\mu}_2)$.

model (not normalized) for each of these three identified dipole-type modes is shown in Fig 3.3D. Note that although we did not explicitly stipulate any criterion that the two extrema should be spatially offset in our above identification procedure, we observed that for each of the three identified dipole-type mode RFs, the means of the two Gaussians of the best-fit DoG model were spatially separated (Fig 3.3D). The remaining modes which did not satisfy the above ratio criterion were classified as having monopole spatial RFs. For these modes, in order for the best-fit DoG model to produce a monopole-type RF profile, either the means of the two Gaussians were nearly identical as in a center-surround organization, or the second Gaussian merely contributed a negligible "noise blip".

For the remaining 46 modes which had monopole spatial RFs, we next investigated how these mode spatial RFs compared to those of the individual retinal ganglion cells (RGCs). To quantify the size of mode and individual RGC spatial RFs, we first fit a single $2D$ Gaussian to each smoothed spatial RF profile via nonlinear regression (i.e. using iterative least squares estimation). As with the dipole model fits, for each mode and RGC, nonlinear regression was performed 100 times using different randomly-chosen initial parameter values, and the best-fit $2D$ Gaussian was chosen as the one with minimum MSE. An example best-fit $2D$ Gaussian is shown in 3.4 Fig (panel A).

We then measured the RF radius for each mode and each individual ganglion cell, which we define as the semi-major axis length of the 95% confidence interval ellipse of the best-fit $2D$ Gaussian. Under the null hypothesis of independent visual signaling, one would expect the RF of a firing pattern to approximate the union of the individual cells' RFs [103]. If we interpret a Tree HMM latent mode as a type of definition of neural population 'codeword' (an idea that we will examine in detail in the next section and in Chapter 4), then we would likewise expect the mode RFs to approximate the weighted union of the individual RGCs that contribute to the

mode. Based on this idea, we compared the actual RF radius for each mode α, which we denote by $r_\alpha^{(\text{real})}$, to the radius expected under the null hypothesis of independent visual signaling:

$$r_\alpha^{(\text{null})} \equiv \frac{\sum_{i=1}^{N} m_{i,\alpha} \cdot r_i}{\sum_{i=1}^{N} m_{i,\alpha}} \tag{3.39}$$

where r_i denotes the radius for cell i, and $m_{i,\alpha}$ denotes cell i's mode-dependent firing probability. Note that Eq. 3.39 is a generalization (to the case where the weights $m_{i,\alpha}$ are heterogeneous) of the null model formulation used in Ref. [103]. A scatter plot of $r_\alpha^{(\text{real})}$ vs. $r_\alpha^{(\text{null})}$ for each mode α is shown in panels B and C of 3.4 Fig.

The error bars shown in panels B and C of 3.4 Fig represent the standard deviation associated with each $r_\alpha^{(\text{real})}$ value, which we denote by σ_{r_α}. This value was computed via propagation of error on the fit covariance matrix parameters. In practice, to ensure that the covariance matrix Σ was symmetric positive-definite, we set $\Sigma = LL^T$, where the matrix L is upper-triangular with positive values on the diagonal. We then actually adjusted the parameters L_{11}, L_{21}, and L_{22} when fitting the single $2D$ Gaussian model. Since the radius is defined to be the semi-major axis length of the 95% confidence ellipse, it follows that

$$r_\alpha^{(\text{real})} = \sqrt{5.991} \cdot \sqrt{\lambda_1} \tag{3.40}$$

where λ_1 denotes the largest eigenvalue of Σ. The variance $\sigma_{r_\alpha}^2$ was then estimated as the first two terms of the error propagation equation:

$$\sigma_{r_\alpha} \approx \sigma_{L_{11}}^2 \left(\frac{\partial r_\alpha}{\partial L_{11}} \right)^2 + \sigma_{L_{21}}^2 \left(\frac{\partial r_\alpha}{\partial L_{21}} \right)^2 \tag{3.41}$$

where $\sigma^2_{L_{11}}$ and $\sigma^2_{L_{21}}$ denotes the variances for L_{11} and L_{21}, respectively, estimated via our fitting procedure, and where

$$\begin{aligned}\frac{\partial r_\alpha}{\partial L_{11}} &= \frac{\sqrt{5.991}}{2} \cdot \left[\eta\left(L_{11}L_{21} - \frac{L_{11}^2}{c}\right)\right]^{-1/2} \cdot \eta\left(L_{21} - \frac{2L_{11}}{c}\right) \\ \frac{\partial r_\alpha}{\partial L_{21}} &= \frac{\sqrt{5.991}}{2} \cdot \left[\eta\left(L_{11}L_{21} - \frac{L_{11}^2}{c}\right)\right]^{-1/2} \cdot \eta L_{11}\end{aligned} \tag{3.42}$$

In Eq. (3.42), $\eta \equiv \frac{c}{acb-a^2}$, where a and b denote the first and second component, respectively, of the eigenvector of covariance matrix Σ which corresponds with the largest eigenvalue, and c and d denote the first and second component of the other eigenvector. Note that since we had interpolated each spatial RF by a factor of 4, the terms in Eq. 3.41 were scaled by a factor of $\frac{1}{4}$ to obtain error bar values in units of checkers (see panels B and C of Fig 3.4).

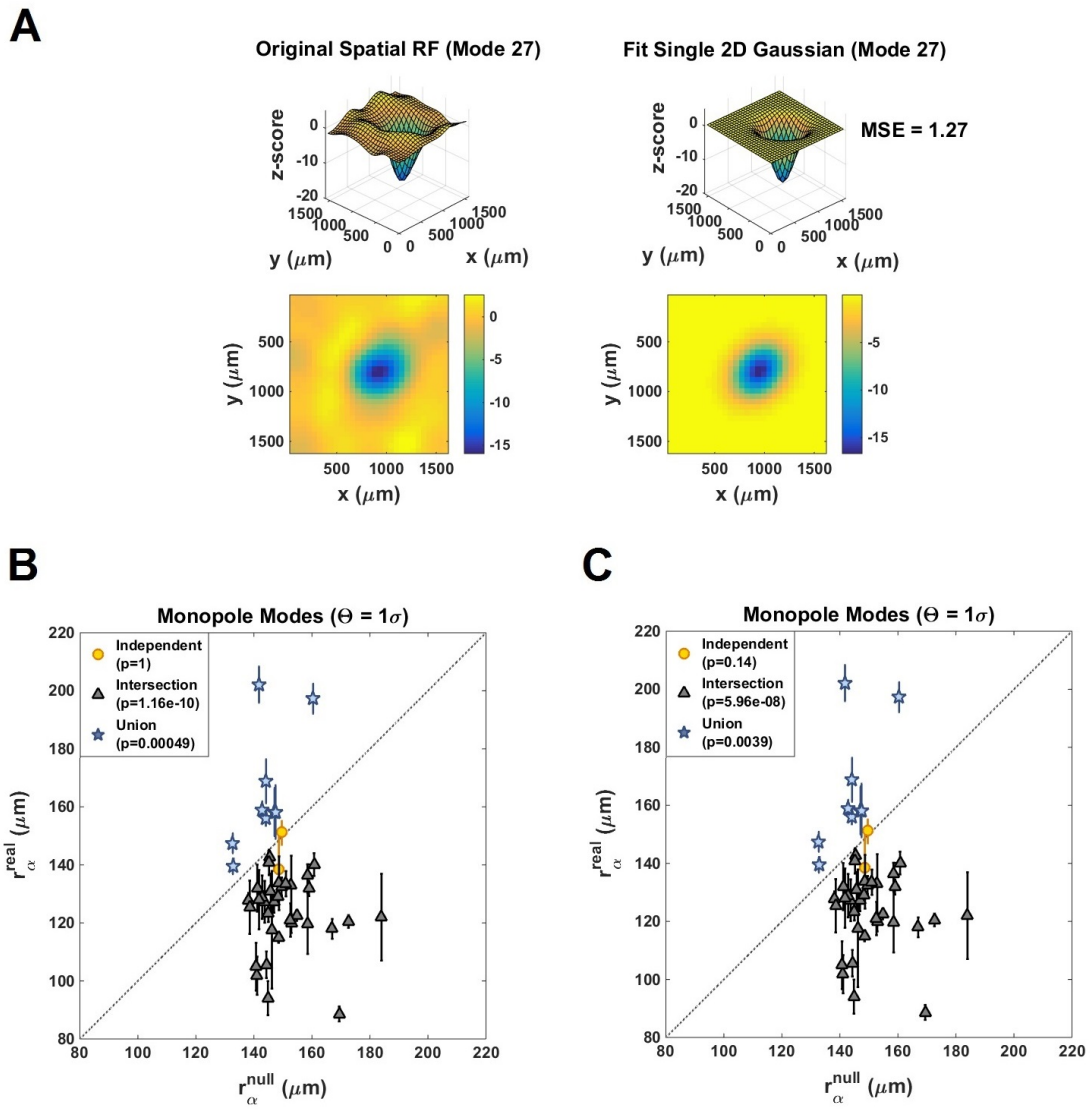

Figure 3.4: **Classification of monopole collective mode spatial receptive fields.** (**A**) *Left*: Smoothed spatial RF of an example monopole mode, measured by performing SVD on the mode-triggered average (MTA) stimulus. Layout as in S1 Fig. *Right*: The corresponding best-fit single $2D$ Gaussian model for this example mode; the corresponding mean-squared error (MSE) is reported at right. (**B**) Classification results obtained when the threshold criterion $\Theta = \sigma_{r_\alpha}$ was used, where σ_{r_α} denotes the estimated standard deviation of the actual radius for mode α (see Eqs. 3.40, 3.41, 3.42). Each point corresponds with one of the 46 modes which were previously identified as having a monopole spatial RF. Shown on the x-axis is the value of the radius predicted by the null model, $r_\alpha^{(\text{null})}$, for the given mode (see Eq. 3.39); the actual value of the radius, $r_\alpha^{(\text{real})}$, is shown on the y-axis. Dashed gray line denotes the line of unity. Each monopole mode was classified as either independent (yellow circles), intersection-type (black triangles), or union-type (blue stars) based on the threshold criterion. The p-value obtained upon performing a Wilcoxon signed-rank test on the set of $\left(r_\alpha^{(\text{real})}, r_\alpha^{(\text{null})}\right)$ pairs assigned to each respective category is shown in the legend (top-left). (**C**) Classification results obtained when the threshold criterion $\Theta = 2\sigma_{r_\alpha}$ was used. Layout is the same as in panel (B).

Classification of intersection, union and independent monopole modes

We next categorized the spatial RF of each of the 46 monopole modes as one of three mutually exclusive types: 1) intersection, 2) union, and 3) independent. A monopole mode α was classified as having an intersection-type spatial RF if

$$r_\alpha^{(\text{null})} - r_\alpha^{(\text{real})} > \Theta \tag{3.43}$$

where $\Theta > 0$ denotes a threshold criterion. Similarly, α was classified as having a union-type spatial RF if

$$r_\alpha^{(\text{real})} - r_\alpha^{(\text{null})} > \Theta \tag{3.44}$$

Finally, if

$$|r_\alpha^{(\text{real})} - r_\alpha^{(\text{null})}| \leq \Theta \tag{3.45}$$

then mode α was classified as having an independent-type spatial RF. In practice, we tested threshold criterion values of $\Theta = \sigma_{r_\alpha}$ and $\Theta = 2\sigma_{r_\alpha}$, where σ_{r_α} denotes the standard deviation of the actual radius $r_\alpha^{(real)}$, which was calculated as previously described. To assess whether the classification of modes obtained using either of these choices for the threshold criterion was appropriate, we performed a Wilcoxon signed-rank test on the set of $\left(r_\alpha^{(\text{real})}, r_\alpha^{(\text{null})}\right)$ pairs assigned to each of the three categories.

The classification and significance results for each criterion choice are summarized in Table 3.1, and shown visually in panels B and C of 3.4 Fig. For the choice of $\Theta = 2\sigma_{r_\alpha}$: 14 modes were classified as independent-type, which was supported by a two-sided Wilcoxon signed-rank test ($p = 0.14 > 0.05$, i.e. insignificant difference between the actual and null model radii values); 24 modes were classified as Intersection-type, which was supported by a left-tailed Wilcoxon signed-rank test ($p = 5.96 \times 10^{-8} < 0.01$, i.e. the classified modes had a significantly smaller radius than predicted by the null model); and 7 modes were classified as union-type, also

supported by a right-tailed Wilcoxon signed-rank test ($p = 0.0039 < 0.01$, i.e. the classified modes had a significantly larger radius than predicted by the null model). For the choice of $\Theta = \sigma_{r_\alpha}$: 2 modes were classified as independent-type, which was strongly supported ($p = 1 > 0.05$); 33 modes were classified as intersection-type, also strongly supported ($p = 1.16 \times 10^{-10} < 0.01$); and 11 modes were classified as union-type, likewise well-supported ($p = 0.00049 < 0.01$).

Table 3.1: Monopole Mode RF Classification Results

	$\Theta = r_\alpha$		$\Theta = 2 \cdot r_\alpha$	
Classification:	**# of Modes**	***p*-value**	**# of Modes**	***p*-value**
Independent	2	1	14	0.14
Intersection	33	1.16×10^{-10}	24	5.96×10^{-8}
Union	11	0.00049	8	0.0039

3.4 Error correction endowed by collective modes

3.4.1 Background

It is well established that noise is prevalent throughout the nervous system [38, 66, 80, 101]. Partitioning noisy output patterns into fault-tolerant clusters corresponding with "codewords" is a long-standing idea that has been utilized in traditional error-correcting codes in communications engineering [109]. In the engineering paradigm, a "codeword" refers to the encoded form of some input information ("message"). The additional information (redundancy) added by the code prior to transmission to generate the codeword is used by a "receiver" to enable the correction of errors induced by noise during transmission.

Conceptually, we can draw an analogy between this paradigm and the retinal code as follows: the input information corresponds to the external visual stimulus (or everything within the bandwidth of the photoreceptors); the code that is used for encoding the external visual information into a form that can be utilized internally by the brain corresponds to the ganglion cell population code; and the population activity patterns that we as experimenters observe at the output of the ganglion cell layer correspond to potentially noise-corrupted versions of the codewords. However, we emphasize that there are also two fundamental disanalogies between the traditional engineering paradigm and our scenario: (1) in our case the original population code is unknown, as are its codewords; (2) we are here thinking about noise within the retina, not noise within a "transmission line" that is separate from the input system. It is thus not conceptually possible in our case to characterize the noise in the transmission line.

Since downstream brain areas lack direct access to external stimuli, biologically plausible decoding must be unsupervised. If the retinal ganglion cell population code were structured to enable the observed noisy activity patterns to be partitioned into clusters in the N-dimensional response space, such that each cluster corresponded to a codeword, then a major advantage would be that decoding and error correction (i.e., mapping noise-corrupted activity patterns to the correct codewords) could be performed via an unsupervised clustering algorithm. Recent studies from multiple groups have suggested that neuronal population codes may indeed be qualitatively structured in this fashion [43, 52, 116].

Note that there is a one-to-one mapping between codewords and clusters. We lack explicit knowledge of the former, but can potentially detect the latter via unsupervised methods. Thus, somewhat at variance with the traditional engineering notation, in Chapter 4 we will refer to the corresponding clusters (which we call "collective modes") as the neural population codewords.

3.4.2 Empirical applicability of the error-correcting code paradigm for the non-repeat stimulus limit

An example of error correction exhibited by the collective modes for the interleaved natural movie (Movie #4; see Chapter 2 for details) dataset is shown in Fig 3.5. For this analysis, we first fit the Tree HMM to the set of population responses ($N = 170$ retinal ganglion cells) evoked by the *non-repeated* portion of the interleaved natural movie ($\sim$73 min; see Chapter 2 for stimulus details). Then, using the learned model parameters, we inferred the collective mode that was active at each time bin t of the fixed 60-s target movie clip, for all 73 repeats of that target movie clip. Inference was implemented by maximizing the posterior probability of the latent collective mode sequence, conditioned on all of the observed spike train data from the beginning of the experiment up until time t: $\alpha^*(t) = \arg\max_{\alpha \in [m]} P_{\text{HMM}}\left(\alpha | \{\vec{x}(0), \cdots, \vec{x}(t)\}\right)$. This maximization was implemented by the Viterbi algorithm [60]. For each time bin t, we then computed the empirical probability, across the 73 repeats of the target clip, of time bin t being mapped to collective mode α: $p(\alpha|t) \equiv \frac{1}{73} \sum_{r=1}^{73} \delta_{\alpha, \phi(\vec{x}_r(t))}$, where $\delta_{a,b}$ denotes the Kronecker delta function (i.e. $\delta_{a,b} = 1$ if $a = b$ and 0 otherwise), $\vec{x}_r(t)$ denotes the population response observed during time bin t for the r-th repeat, and $\phi(\vec{x}_r(t)) \equiv \arg\max_{\alpha \in [m]} P_{\text{HMM}}\left(\alpha | \{\vec{x}_r(0), \cdots, \vec{x}_r(t)\}\right)$ denotes the mapping obtained via the Viterbi algorithm.

As shown in Fig 3.5A, due to noise, a number of distinct population response patterns are typically evoked by the same stimulus. For example, at time bin t_1^* (arrow), every repeat evoked a distinct population response pattern. However, these distinct response patterns were all mapped to the same collective mode identity (mode 43, purple), i.e. $p(\alpha = 43 | t_1^*) = 1$ (Fig 3.5B). This demonstrates that the collective mode identity can enable robustness to noise, reminiscent of the idea of codewords, and hence suggests that the mode representation could constitute a form of error-correcting code. Note that we further quantified the global degree of robustness to

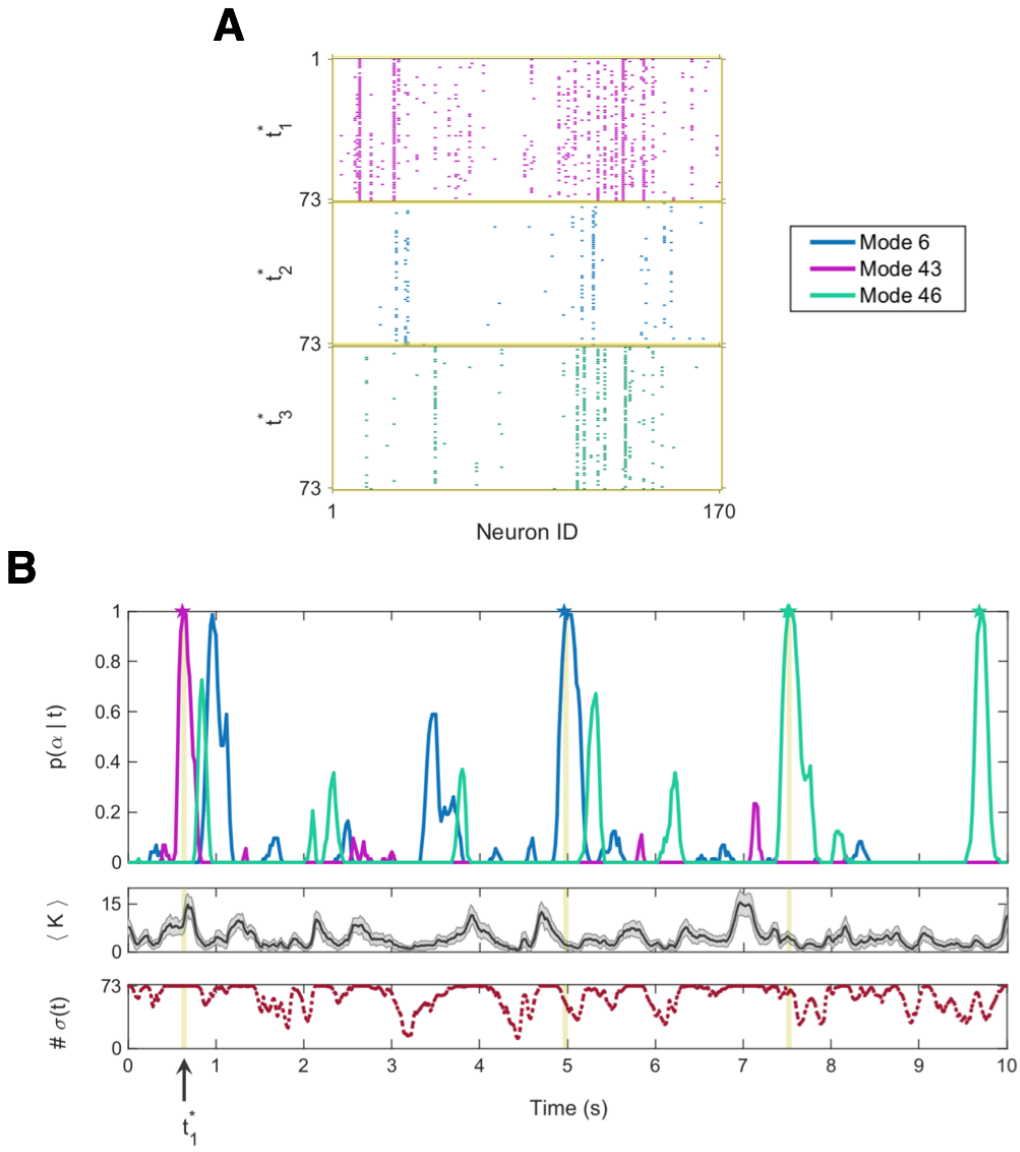

Figure 3.5: **Error correcting properties of collective modes.**
(**A**) Spike rasters of the population responses observed over all 73 repeats of the target movie segment for the interleaved natural movie dataset ($N = 170$ RGCs). A raster is shown for each of three time bins - denoted t_1^*, t_2^*, and t_3^*, and highlighted in yellow in panel (B) - in the target movie segment. Each row denotes one observed population response; responses are displayed in chronological order. White denotes silence; colors denote the collective mode to which the observed population response was mapped via the Viterbi algorithm (see section 3.2.4). (**B**) *Top*: Shown for three collective modes (see key) is the probability (across repeats of the target movie segment) that the population activity was mapped to that particular collective mode at any given time t. Time bins for which this value, denoted $p(\alpha|t)$, reached 1 are indicated by a star. *Center*: Mean population spike count averaged over the repeats; error bars denote one standard deviation. *Bottom*: The number of distinct population responses $\vec{\sigma}(t)$ observed over all 73 repeats at each time t.

noise endowed by mapping the highly-variable RGC population responses to collective modes in [87] using an "information efficiency" measure. For three separate datasets, this measure was found to be significantly above chance. Moreover, results from control analyses demonstrated that this high level of collective mode reproducibility is not explainable by trivial effects [87].

Thus, in terms of computational goals, these results suggest that another goal of the retinal ganglion cell population is to enable error correction to combat neural noise, which is prevalent at all stages throughout the nervous system [38]. Given the predominantly feedforward circuitry from the retina to LGN to primary visual cortex (V1), the retina is potentially amenable to the forward error-correcting code paradigm. For investigations of population codes and robustness in cortical areas however, given the massive presence of recurrent connections and significant role of feedback inter-areal circuits in the cortex [95], we speculate that fault-tolerant ideas that incorporate dynamic output feedback will likely be vital.

3.5 Robustness to independent fittings of the Tree HMM

The Baum-Welch algorithm used to learn the model parameters of the Tree HMM is only guaranteed to converge to a local maximum of the log-likelihood [60]. We thus sought to verify that the identities of the collective modes are robust over distinct, independent fittings of the Tree HMM.

To do so, we first generated two independent subsets of the response data, $\mathcal{D}_1$ and $\mathcal{D}_2$, which were each comprised of 90% of the original data. Specifically, we first found all stretches of the data that either began or ended with at least one all-silent response pattern, $\vec{0}$. This basis for using this method was to prevent breaking temporal correlations between non-silent collective modes. Next, we generated two

independent random permutations by which to choose these blocks to form $\mathcal{D}_1$ and $\mathcal{D}_2$. We then trained the Tree HMM on $\mathcal{D}_1$ to learn a set of model parameters θ_1^*, and trained on $\mathcal{D}_2$ to learn a separate set of parameters θ_2^*. Let ϕ_1 denote the mapping given by the Viterbi algorithm using the learned θ_1^* parameters; i.e. for any time bin t, $\phi_1(t) \equiv \arg\max_{\alpha \in [m_1]} P_{\text{HMM}}\left(\alpha | \{\vec{x}(0), \cdots, \vec{x}(t)\}; \theta_1^*\right)$. Similarly, let $\phi_2(t) \equiv \arg\max_{\alpha \in [m_2]} P_{\text{HMM}}\left(\alpha | \{\vec{x}(0), \cdots, \vec{x}(t)\}; \theta_2^*\right)$. For each collective mode α and β identified via the first and second model fitting, respectively, we computed the conditional probabilities $p_{\alpha|\beta} \equiv P\left(\phi_1(t) = \alpha | \phi_2(t) = \beta\right) = \frac{P(\phi_1(t)=\alpha, \phi_2(t)=\beta)}{P(\phi_2(t)=\beta)}$. Finally, we quantified how closely collective mode α from the first model matched a single collective mode β from the second model by computing the "confusion entropy":

$$\mathcal{H}_{\text{confusion}} = -\sum_{\beta} p_{\alpha|\beta} \log p_{\alpha|\beta} \tag{3.46}$$

Note that this measure has a value of 0 for a perfect match, and that lower values indicate a better match. Results from this analysis show that the identification of collective modes was robust to sampling variation (Figs 3.6 and 3.7).

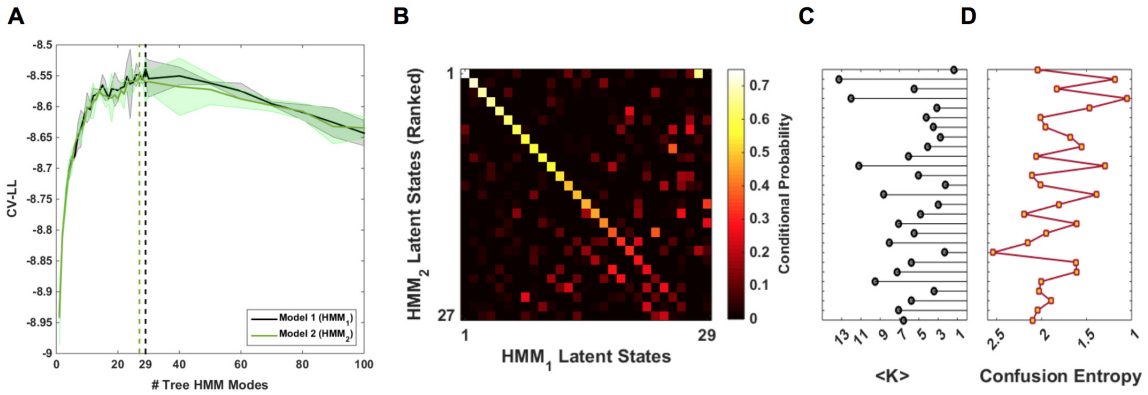

Figure 3.6: **Robustness of identified collective modes over fittings of the model.** Results for the Movie #4 dataset ($N = 170$ neurons). (A) Comparison of the cross-validated log-likelihood curve obtained for the first independent fitting of the Tree HMM (black) vs. for the second fitting of the Tree HMM (green). Dark lines denote the mean over the 2 cross-validation folds; error bars denote one standard deviation. (B) Confusion matrix of the conditional probabilities $p_{\alpha|\beta}$, using the second fit model (HMM_2) as the reference. (C) Mean spike count, $\langle K \rangle$, and (D) confusion entropy of each mode α of the reference model.

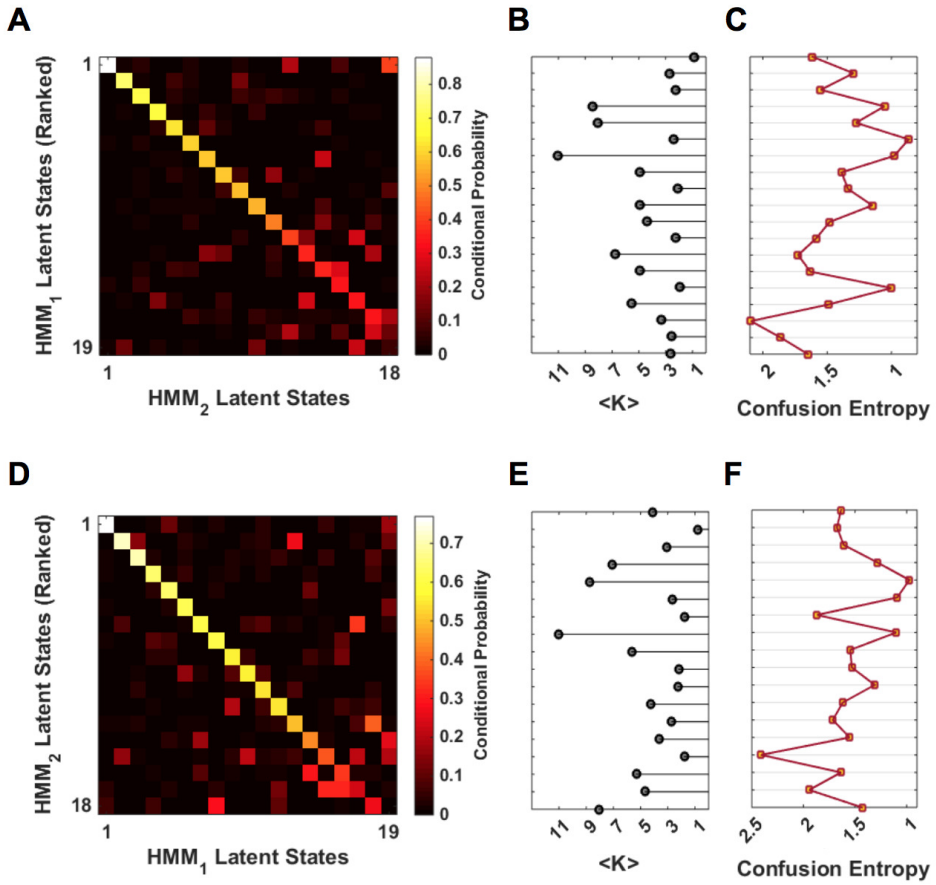

Figure 3.7: **Robustness of identified collective modes over fittings of the model.** Results for the Movie #2 dataset ($N = 152$ neurons). (A) Confusion matrix of each conditional probability value $p_{\alpha|\beta}$, which denotes the empirical probability that any given observed time bin is mapped by the Viterbi algorithm to collective mode α using the fit HMM_1 parameters, given that it was mapped to collective mode β using the fit HMM_2 parameters. The matrix is sorted so that rows are ranked in descending order, and so that the diagonal monotonically decreases to the extent possible. (B) The mean spike count, $\langle K \rangle$, for each collective mode α of the first learned model (HHM_1), averaged over all observed population responses in the full data that are mapped to α via the Viterbi algorithm. (C) The confusion entropy for each collective mode α of HMM_1. (D-F) Same as in panels (A-C), but instead using the second fit model, HMM_2, as the reference.

Chapter 4

The Structure of Retinal Population 'Codewords'

4.1 Introduction

How do correlations affect the code of large neural populations? There are several ideas that have arisen from the past computational neuroscience literature. The oldest is that positive noise correlations can severely limit the encoded information, because they prevent large populations from averaging over the independent noise of neurons [128]. However, this effect can be minimized if the noise correlations are orthogonal in the space of neural activity to the correlations induced by common stimulation [75, 108]. In fact, positive correlations with the right structure can even greatly enhance the encoded information [7]. What these past studies have in common is that they have focused on how correlations quantitatively affect coding fidelity. A second, yet less well-studied question, is how correlations affect the *qualitative structure* of the neural population code.

In the vein of this second approach, a recent idea is that correlations may organize neural population activity into a discrete set of clusters that constitute different

"codewords" [25, 43, 59, 87, 116, 117]. There are three notable advantages of having a code with this type of structure. Firstly, since the receptive field properties of multi-neuronal codewords have been shown to differ significantly from those of constituent single neurons [26, 87, 102], such a clustering operation constitutes a non-trivial computation that potentially changes the feature basis set used at each stage of a neural information processing pathway. Secondly, downstream neurons could use unsupervised learning mechanisms to identify these clusters (and hence learn the new feature basis set). Moreover, the identification of these clusters involves forming a map from many neural activity patterns onto the same cluster, or codeword. Due to this many-onto-one mapping, these population codewords can exhibit *error correction* - namely, that the codeword is more reliably activated by the stimulus than individual activity patterns [87].

Error correction is an idea that originates from communications engineering, which involves the design of codes for data transmission and storage that enable accurate decoding even when the transmitted information has been corrupted by noise [96, 109]. Traditionally, this robustness is achieved by introducing redundancy to partition the space of possible output patterns, so that all noise-corrupted versions of the same input message reliably map to the same subset. Consistent with this notion, the retinal code has been found empirically to be highly redundant [88]. Furthermore, sensory maps in the cortex, such as V1, have many more neurons than do their subcortical sources, implying that these brain areas are even more redundant [8]. Error correction is an appealing principle, because it offers a way to bridge the gap between the noisy activity of neurons and the fact that our perception can be quite deterministic [7].

Due to the combinatorial explosion of possible neural population responses, it is generally intractable to determine the empirical probability distribution of joint activity. This fundamental limitation necessitates a modeling approach. Within

computational neuroscience, there are two distinct methodologies that have been used to create probabilistic models of network activity. The traditional approach - which includes generalized linear models (GLMs) - attempts to explicitly capture the dependence of neuronal responses on the stimulus [85]. These types of models have been referred to as "encoding models" [124]. The second approach, which we call the "activity model" approach [87], involves directly modeling the structure of population activity, without any reference to the stimulus. An important advantage of activity models is that they correspond to the unsupervised problem actually faced by downstream brain areas, which lack direct access to the external stimuli.

One popular class of activity models comes from the Maximum Entropy (MaxEnt) principle (see Chapter 2 for details) [42, 74, 78, 100]. Previous work which fit the K-Pairwise MaxEnt model to activity measured from retinal ganglion cells reported a proliferation of basins in the associated energy landscape [116]. These basins correspond to local peaks in the joint response probability landscape (see Fig 4.1), and can be found by an iterative algorithm that changes the firing state of a single neuron to increase the probability at each step (Fig 4.1C). It was suggested that these local probability peaks could be a candidate for neural population codewords [116].

In this study, we started by uncovering an important problem. Previous studies that have identified putative population codewords as corresponding to local peaks in the probability landscape used experimental data in which a short stimulus segment was repeated many times. However, when we applied the same analysis methods to retinal ganglion cell data under stimulation by non- and mildly-repeated stimulus ensembles, we found almost no local probability peaks. On the other hand, a recent study that used a hidden Markov model (HMM) to describe the ganglion cell population's probability landscape was able to find clusters in the population activity space - called "collective modes" - under non-repeated visual stimulation [87]. Importantly, these collective modes were shown to provide a new feature basis set

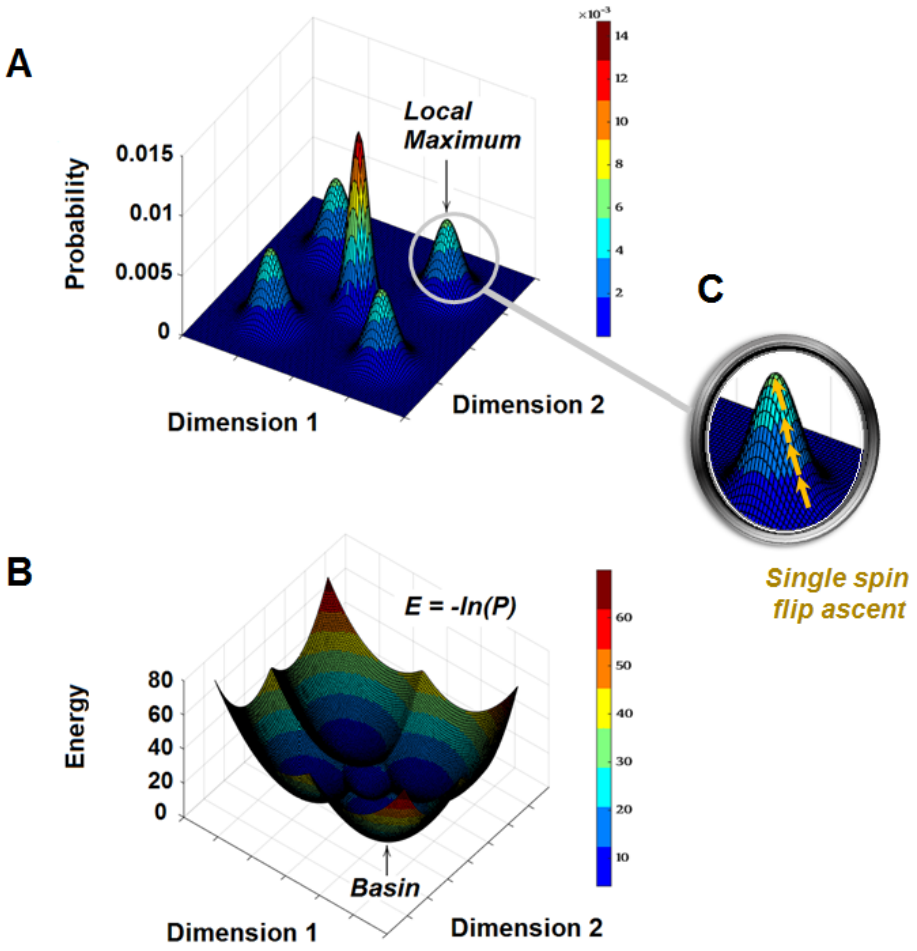

Figure 4.1: **Schematic illustrating the concept of local maxima and basins.** (A) Cartoon illustrating local maxima in a probability landscape. For ease of visualization, the full space of population activity patterns is depicted as a continuous, $2D$ space. The actual domain of the joint probability mass function considered here is discrete and high-dimensional (in particular, an N-dimensional hypercube). (B) Cartoon of basins in the energy landscape corresponding to the probability landscape shown in panel (A). A response state $\vec{\sigma}$ that is a local probability maximum equivalently corresponds to a basin in the associated energy landscape, since the energy $E(\vec{\sigma}) \equiv -\ln P(\vec{\sigma})$. (C) Local maxima in the probability landscape are found via an iterative single spin flip ascent algorithm. Each yellow arrow denotes one iteration.

(including some modes with orientation selectivity), and moreover to enable significant error correction, supporting their candidacy as neural population codewords. Building on this previous work, we confirmed that putative population codewords are identifiably present under non- and mildly-repeated visual stimulation. The first

key result of this chapter is that, at least for the low-repeat regime, they do *not* correspond with local peaks in the probability landscape.

Motivated by this result, we next sought to investigate the geometry of identified population codewords for the non-repeated stimulus regime. To do so, we introduced a new notion of structure, here termed *soft local maxima*, which are local maxima in the space of all neural activity patterns that are restricted to have the same spike count. These soft local maxima were robustly present. Using a novel numerical approach, we then explored their organization across spike count levels. This approach led to an algorithm that links together different soft local maxima into a discrete *ridge*, which is a geometric feature of the joint probability landscape of neural population activity. We found that there was a close correspondence between these ridges and the population codewords found by the hidden Markov model. We argue that depending on the behavioral context, experiments designed with either repeated or non-repeated stimulus ensembles can be relevant, and therefore both geometric structures - local peaks and ridges - can be important for the population code. Finally, we realized that these ridges correspond with *neuronal communities*, a notion from network theory [40]. Our results thus suggest a unified picture linking statistically-defined "codewords" of the retinal population code, geometric structure in the joint response probability landscape, and community structure within the neuronal population.

4.2 Population codewords $\neq$ local maxima for low-repeat stimulus regimes

Previous studies that used the Maximum Entropy framework (see Methods) to explore the probability landscape of neural population activity have reported a proliferation of local probability maxima [116]. However, whether or how this feature depends on

the stimulus ensemble was unknown. One potential confound of these studies, is that they have exclusively used highly-repeated stimulus ensembles. For a highly-repeated stimulus ensemble, a small set of points in stimulus space is sampled on many trials. It is thus plausible that the previously reported local maxima correspond to 'average' response patterns elicited by the stimuli comprising the short movie segment that was repeated, with the width of the local peaks being attributable to scatter around that average due to neural noise.

To investigate whether a proliferation of local maxima is a general feature that is independent of the stimulus repeat structure, we first applied the same K-Pairwise MaxEnt model used in [116] to the measured responses of $N = 128$ retinal ganglion cells to Movie #1, which was a mildly-repeated natural movie stimulus. In response to Movie #1, the mean firing rate averaged over ganglion cells, $\langle r(t) \rangle$, was 2.70 ± 0.21 spikes/s/neuron (mean $\pm$ SEM). The probability of a cell firing a spike in any given time bin t, which we denote $p(t) \equiv \langle r(t) \rangle \triangle t$ (where $\triangle t = 0.02$ s), was 0.054. The measured responses were thus within the sparse firing regime.

After fitting the MaxEnt model to the data (see Methods), we then searched for local maxima of the modeled probability landscape using the same numerical method as in [116], which we term "single spin flip ascent". In brief, this involves taking each population activity pattern observed in the data, and moving "uphill" on the modeled joint response probability landscape. The termination patterns of this algorithm are by definition local maxima of the probability landscape, when only allowing the state of a single neuron to change in each iteration. Note that this choice of defining a local maximum as the single-flip-stable ascent pattern establishes an upper bound on the number of local maxima [19]. The single spin flip ascent algorithm can be thought of as a mapping from the N-dimensional population response space to the set of local probability maxima.

4.2.1 Scarcity of local maxima for low-repeat ensembles

We applied the single spin flip ascent algorithm to the probability landscape obtained by fitting the MaxEnt model to the Movie #1 dataset. This resulted in 65 unique local probability maxima (Fig 4.2A). However, of the 175,002 observed population responses in the data, 99.71% were mapped via single spin flip ascent to the all-silent local maximum (i.e. $\vec{0}$), and 0.037% were mapped to the rank-2 local maximum. This is in stark contrast to the ∼50% of responses that were mapped via single spin flip ascent to the all-silent local maximum for the highly-repeated stimulus dataset in [116].

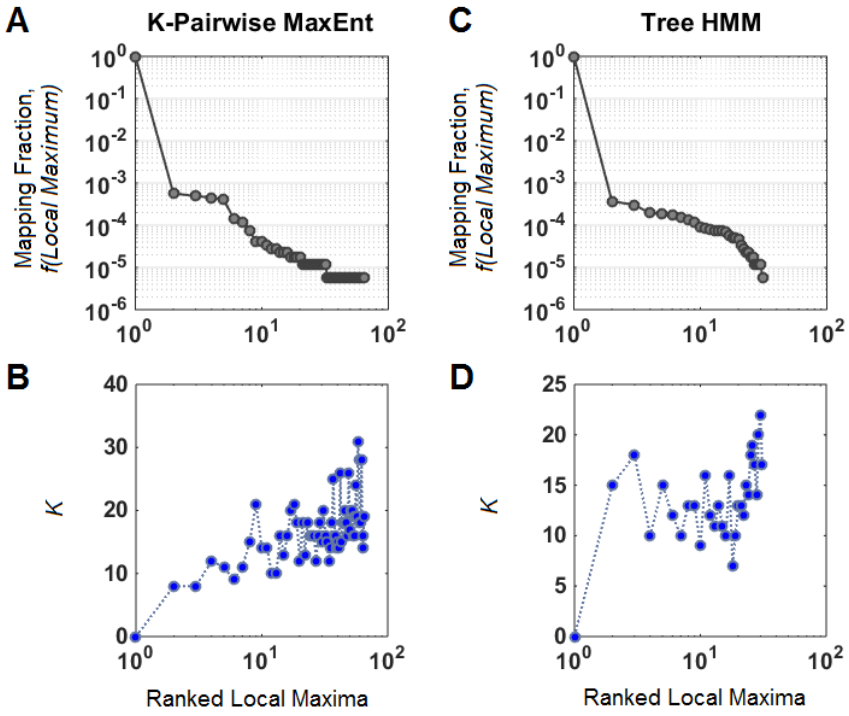

Figure 4.2: **Local maxima results for the dataset of** 128 **RGCs responding to Movie** #1**.** Pairwise Maximum Entropy model (8385 parameters) (A,B) or Tree hidden Markov model (8379 parameters) (C,D) as the underlying probability model. (A) Log-log plot of the proportion of the 175,002 population responses observed in the data that were mapped via single spin flip ascent - denoted as the "mapping fraction", f(Local Maximum) - to the corresponding unique local maximum indicated on the x-axis (ranked). (B) The spike count, denoted by $K \equiv \sum_{i=1}^{N} \sigma_i$, for each of the 65 unique local maxima identified (x-axis, ranked). (C,D) The panels are as described in (A) and (B), but for the hidden Markov model results.

We also fit the Tree hidden Markov model to this dataset (see Chapter 3 for details), and likewise performed single spin flip ascent on the corresponding probability landscape. Since the number of free parameters constitutes an important consideration for model comparison, we selected the number of HMM latent states so as to match the total number of free parameters as closely as possible to the K-Pairwise MaxEnt model. As shown in Fig 4.2C, 31 unique local maxima were found after performing single spin flip ascent on the probability landscape modeled by the HMM. Of the 175,002 observed population responses in the data, 99.74% were mapped to the all-silent local maximum, and 0.056% were mapped to the rank-2 local maximum. This result is consistent with that obtained for the K-Pairwise MaxEnt model, indicating that the paucity of local maxima is an intrinsic feature of the probability landscape rather than an artifact of one specific model.

To investigate the limit of the non-repeated stimulus regime, we next performed the above procedure on a dataset of the responses of 152 ganglion cells to Movie #2, which was a *non-repeated* natural movie stimulus. The mean firing rate elicited by Movie #2, averaged over ganglion cells, was 1.10 ± 0.07 spikes/s/neuron (mean $\pm$ SEM). Correspondingly, the probability of a neuron firing a spike in any given time bin was again sparse, $p(t) = 0.022$. For this non-repeated stimulus dataset, performing single spin flip ascent on the probability landscape modeled by the Tree HMM identified only two local maxima. Moreover, a staggering 99.95% of the 90,001 observed population responses were mapped via single spin flip ascent to the all-silent local maximum.

A systematic analysis

To probe the relationship between stimulus repeat structure and the prevalence of non-silent local maxima more systematically, we next performed an analysis that allowed us to take the number of stimulus repeats as a parameter. Specifically, two

separate experiments were performed in which we presented one of two different movie stimuli: a binary white-noise checkerboard movie (Movie #3), or a natural movie (Movie #4). Both movie stimuli were similarly designed to have unique movie segments that were *interleaved* with repeated presentations of a "target" movie segment (see section 2 for details). A schematic illustrating the experimental setup for the interleaved white-noise stimulus ensemble is shown in Fig 4.3A.

We verified that for both datasets, each ganglion cell's average firing rate during the repeated vs. unique movie segments was statistically identical. For Movie #3, the average firing rate across ganglion cells recorded during the repeated segments and unique segments was, respectively, 0.86 ± 0.08 spikes/s/neuron and 0.84 ± 0.08 spikes/s/neuron (mean $\pm$ SEM). Correspondingly, the probability of a neuron firing a spike in any given time bin was $p(t) = 0.0172$ and $p(t) = 0.0168$, respectively. For Movie #4, the average firing rate across ganglion cells recorded during the repeated clips and unique clips was, respectively, 1.40 ± 0.01 spikes/s/neuron and 1.46 ± 0.01 spikes/s/neuron. The probability of a neuron firing a spike in any given time bin was $p(t) = 0.028$ and $p(t) = 0.029$, respectively.

For each of the two movie stimuli, we then generated "subset movies", which were comprised of a subset of the movie segments that appeared in the original full-length movie (see Methods). Different subset movies included a different ratio of the number of repeated target segments to unique movie segments. Note that the duration of each subset movie was the same, so as to eliminate the potentially confounding effect of different stimulus durations on our sampling of population neural activity. To ensure that there was no dependence on the specific choice or ordering of the movie segments included, we also performed multiple independent random permutations to generate the subset movie corresponding to each repeat ratio. For each permutation and for each repeat ratio, we then fit the Tree hidden Markov model to the set of ganglion cell population responses observed during the corresponding subset movie. The cross-

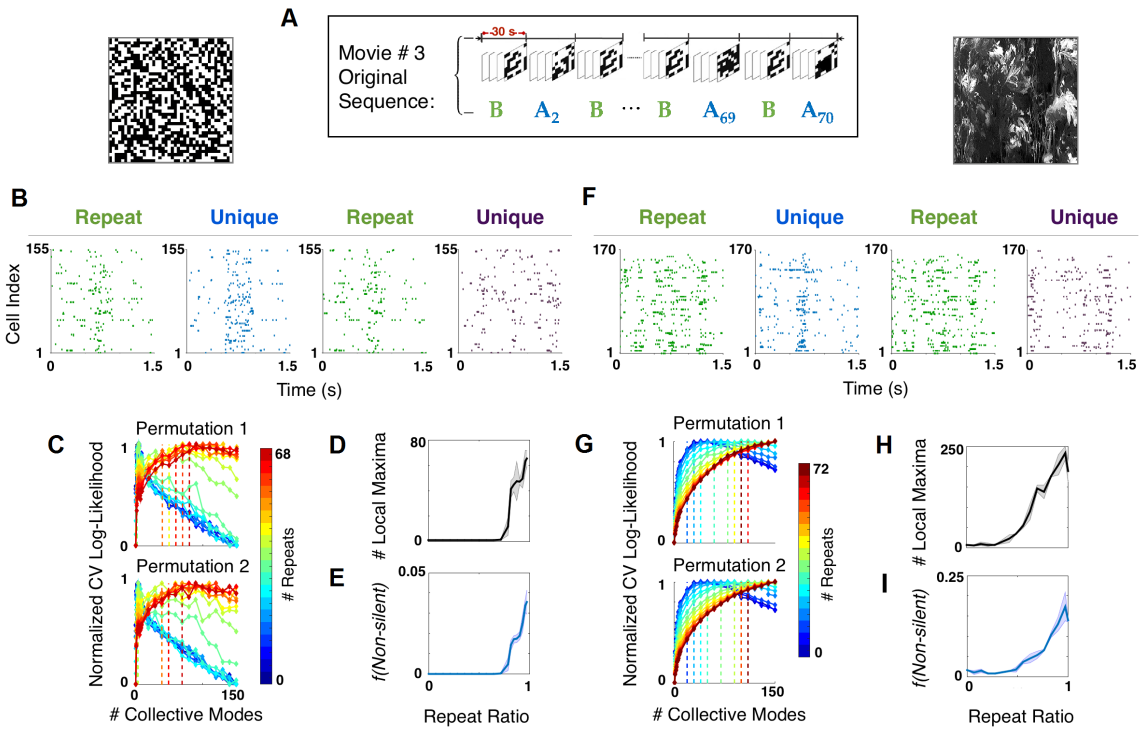

Figure 4.3: **Experimental design and results for the parametric repeat analysis.** (A) Cartoon of the movie stimulus design. Unique movie segments (denoted by blue A's) were interleaved with repeated pesentations of a "target" movie segment (denoted by green B's). (B-E) Results for the dataset of 155 ganglion cells responding to Movie #3. (B) Examples of raw spike rasters elicited by repeated (green) versus non-repeated movie segments (blue and purple). (C) Normalized cross-validated log-likelihood (CV-LL, y-axis) vs. the number of Tree HMM latent states for each repeat ratio (see color key). Dashed lines denote the optimal number of latent states (see *Methods*). (D) Total number of unique local maxima (black) found vs. repeat ratio. Error bars denote SEM. (E) Proportion of the 105,000 population responses observed during each respective subset movie that were mapped via single spin flip ascent to a non-silent local maximum. (F-I) Results for the dataset of 170 ganglion cells responding to Movie #4.

validated log-likelihood results (used to select the optimal number of latent states, or "collective modes"; see Chapter 3 for details) for different repeat ratios are shown in Fig 4.3C and 4.3G. For both datasets, there was a general shift toward a larger optimal number of collective modes as the number of included repeats increased.

After fitting the Tree HMM to the sequence of ganglion cell population responses observed for each subset movie, we performed single spin flip ascent on the modeled probability landscape. For both datasets, there was a predominantly monotonic increase in the number of unique local maxima as a function of repeat ratio. Specifically,

for the white-noise (Movie #3) dataset, we found 1 ± 0 unique local maximum corresponding with the case of 0 included repeats, which increased to 65 ± 6 local maxima when all movie segments were repeats (Fig 4.3D). Likewise, for the interleaved natural movie (Movie #4) dataset, increasing the repeat ratio from 0 to 1 corresponded with a drastic increase in the average number of identified local maxima, from 6 ± 0 to 232.5 ± 16.5 (Fig 4.3H).

Moreover, for both datasets, the weight of non-silent local maxima - that is, the percentage of observed population responses that were mapped by single spin flip ascent to a non-silent local maximum - increased mostly monotonically as a function of the repeat ratio. In particular, for the interleaved white-noise dataset we found that the weight of non-silent local maxima increased from 0% for the non-repeated case, to $3.54\% \pm 0.59\%$ for the all-repeat case (Fig 3E). For the interleaved natural movie dataset, the effect was stronger, with the corresponding value being $1.63\% \pm 0.01\%$ for the non-repeated case, and $17.34\% \pm 3.54\%$ for the all-repeat case (Fig 4.3I).

In summary, we have found that the qualitative structure of the probability landscape of ganglion cell population activity strongly depended on the visual stimulus ensemble. Specifically, consistent with past work, we found that when particular stimuli were repeated sufficiently, this resulted in the incorporation of non-silent local maxima in the probability landscape. However, our first key novel result is that within the non- and mildly-repeated stimulus regimes, for both artificial and natural movie stimuli the modeled response probability landscape was instead essentially comprised of a single global peak, arranged around the all-silent activity pattern, $\vec{0}$.

4.2.2 Reassessing possible interpretations

Clearly, a mapping of neural activity onto local maxima in the low-repeat stimulus regime results in an encoding scheme with a prohibitively low code rate [65], as it corresponds to a dictionary comprised of only the all-silent codeword. Hence, this does

not form a very useful population coding scheme. To motivate our next directions, we need to first understand how to best interpret this result. One possibility is that the error-correcting code paradigm does not apply in the low-repeat stimulus limit. An alternative possibility is that neural population activity can still be mapped to error-correcting codewords, but that these codewords do not resemble local peaks in the probability landscape. In support of the latter interpretation, our previous work in [87] showed, by applying the Tree hidden Markov model, that the N-dimensional response space of ganglion cell population activity can be organized into clusters. These clusters, termed "collective modes," were shown to provide a new basis set of visual features different from the receptive fields of single ganglion cells, and were moreover shown to enable significant error correction (see Fig 11 in [87]). Importantly, this error correction property was demonstrated for the non-repeated stimulus regime.

An example of error correction exhibited by the collective modes for the interleaved natural movie (Movie #4) dataset is shown in Fig 3.5A in Chapter 3. For this analysis, as in [87], we fit the Tree HMM to the population responses evoked by the *non-repeated* portion of the stimulus ensemble. Then, using these learned parameters, we applied the Viterbi algorithm [60] to map the sequence of population responses observed for each of the 73 repeats of the fixed 60-s target movie segment to the most likely sequence of collective modes (see Chapter 3 for details). As shown in Fig 3.5B, due to noise, a number of distinct population response patterns are typically evoked by the same stimulus. For example, at time bin t_1^* (arrow), every repeat evoked a distinct population response pattern. However, these distinct response patterns were all mapped to the same collective mode identity (mode 43, purple in Fig 3.5A), i.e. $p(\alpha = 43|t_1^*) = 1$. This demonstrates that the collective mode identity can enable robustness to noise, and hence suggests that the mode representation could constitute a form of error-correcting code.

The global degree of robustness to noise endowed by mapping the highly-variable population responses to collective modes was further quantified in [87] using an "information efficiency" measure. For three separate datasets, this measure was found to be significantly above chance. Moreover, results from control analyses demonstrated that this high level of collective mode reproducibility is not explainable by trivial effects (see Fig 11E in [87]). Finally, as shown in the Appendix, we verified that the collective mode identities are robust to independent fittings of the hidden Markov model. Combined with these other results, our analyses support the conclusion that the error-correcting population code paradigm is still applicable to the retinal population code under low-repeat stimulation, but that the "codewords" are not local probability maxima.

4.3 Probing the geometry of collective modes

Our previous results in [87] suggest that a better codeword candidate is the collective modes identified via the Tree HMM approach. As detailed in Chapter 3, the Tree HMM is a generative statistical model that formalizes the fact that the population codewords are not directly observable by representing them as latent (hidden) states, which have a one-to-one correspondence with the collective modes. As a statistical modeling approach, it importantly does not make explicit assumptions about the geometry of the population codewords. This thus leaves open the question: What is the geometry of the collective modes in the response probability landscape? As fitting the HMM is not a biologically plausible computation, it would be useful to know whether the collective modes have other structural correlates that could potentially be detected by biologically plausible algorithms. We thus next sought to characterize the geometry of the collective modes.

4.3.1 The role of sparseness

As a first step, we note that neural activity of retinal ganglion cells was very sparse across all of the stimulus ensembles tested, both artificial and natural. This leads to a simple intuition about why we see essentially no local peaks in the probability landscape in the low-repeat stimulus regime. Due to the sparseness of neural activity, it may be quite unlikely that an activity pattern with K spikes has higher probability than a pattern with $K-1$ spikes formed by switching one of its spiking neurons to silent.

To gain a sense of how strongly sparseness affects the probability landscape, we plotted the empirical probability of finding activity patterns with K spikes (Fig 4.4). For all stimulus ensembles examined, this probability monotonically decreased with K. Moreover, when we fit a log-linear model to this function at low K values ($0 \leq K \leq 4$), the slope for the moderately-repeated Movie #1 dataset was a factor of $\sim$50-fold decrease in probability per extra spike. For the non-repeated Movie #2 and non-repeated white-noise checkerboard dataset, the fit slope was even higher, being a factor of an 80-fold and 90-fold decrease in probability per extra spike, respectively. This indicates that sparseness controls the probability of activity patterns quite powerfully.

The powerful impact of sparseness led us to postulate that while local maxima may be largely absent in the full probability landscape, perhaps there are local maxima within the restricted space of activity patterns with shared spike count, K. More generally, we can restate this postulate as assuming that there is low-dimensional structure in the high-dimensional probability landscape that could be exploited by downstream brain areas. After all, there are various "no free lunch" theorems that show that, in the absence of low-dimensional structure, very little can be done in high-dimensional learning problems with limited samples [76].

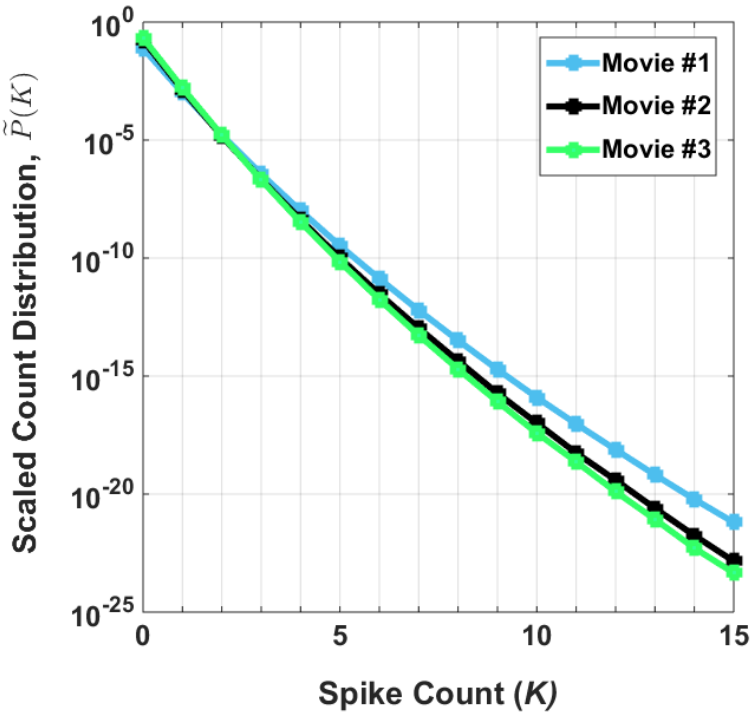

Figure 4.4: **The empirical "scaled count distribution", $\widetilde{P}(K)$, vs. spike count.** The scaled count distribution is defined as the empirical probability of observing a population response with K spikes, normalized by the analytical number of possible joint response patterns with spike count K. Results are shown for three of the different datasets analyzed in the present paper: Movie #1 (blue; $N = 128$ ganglion cells), Movie #2 (black; $N = 152$ ganglion cells), and the non-repeated version of Movie #3 (green; $N = 155$ ganglion cells). Note that the y-axis is shown on a log scale.

4.3.2 "Soft" local maxima

To explore this idea, we investigated whether the response probability landscape for the non-repeated stimulus regime contains what we term *K-soft local maxima*. Intuitively, a K-soft local maximum is a local probability maximum when we restrict our search to the metric subspace (of the full joint response space, which is the N-dimensional Hamming cube) defined on the set of all activity patterns with fixed spike count, K. We will subsequently refer to each such metric subspace as the *K-th spike count level*.

The geometric intuition behind this definition is illustrated in Fig 4.5. Specifically, in its most simplified form, our motivating intuition was that the global probability landscape resembles a "mountain" with the global peak given by the all-silent state. Coming down the mountain (in the direction of higher K) are a number of ridges. Along each ridgeline, the probability is a local maximum of activity patterns at constant spike count, K (blue and green curves, Fig 4.5B). But the strong decrease of probability with K (Fig 4.5C) prevents these activity patterns (such as points γ_{10} and γ_{15}, Fig 4.5A) from being true local maxima. We will show in subsequent sections that this is a qualitatively viable picture of the actual, high-dimensional joint response probability landscape in the low-repeat stimulus regime.

If we wish to find nearby activity patterns that preserve the same spike count, then if one neuron's silence is 'flipped' (changed) to spiking, this necessitates changing another neuron's spiking to silence (and vice versa). Thus, given an activity pattern with spike count K, its "neighbors" in the K-th spike count level will be those that differ by a Hamming distance of 2. In other words, soft local maxima are the stable-probability-ascent patterns when we allow for flipping exactly two opposing neuron response states. Formally, this can be formulated as follows:

Definition 4.1. A response state $\vec{\gamma} \in \{0,1\}^N$ is a *K-soft local maximum* if and only if $P[\vec{\gamma}] > P[\vec{\sigma}] \;\; \forall\, \vec{\sigma}$ such that $w_H(\vec{\sigma}) = w_H(\vec{\gamma}) = K$ and $d_H(\vec{\sigma}, \vec{\gamma}) = 2$, where $w_H(\vec{\sigma}) \equiv \sum_{i=1}^{N} \sigma_i$ denotes spike count (Hamming weight) and $d_H(\vec{\sigma}, \vec{\gamma}) \equiv \sum_{i=1}^{N} |\sigma_i - \gamma_i|$ denotes Hamming distance.

To our knowledge, this concept, which we introduced in [63], is new in the neuroscience literature.

In practice, we used an iterative algorithm that we term *opposite sign neuron pair relaxation* to find the K-soft local maxima for a given spike count K (see Methods for details). Note that our choice of this algorithm is *not* simply the next obvious "higher order" search technique from the single spin slip ascent algorithm. Rather,

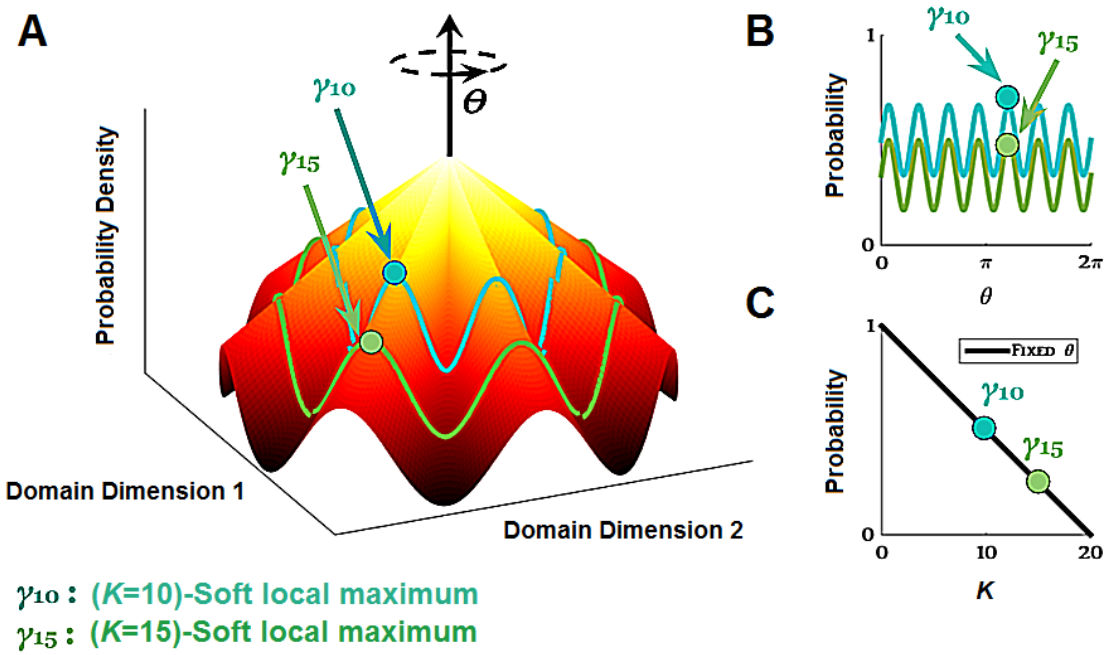

Figure 4.5: **Schematic illustrating the concept of K-soft local maxima and ridge points.** (A) Cartoon of a multivariate probability mass function and its "ambient" (i.e. prior to projection) domain. For ease of visualization, the ambient domain is shown in 2D, and is represented in polar coordinates. Here we denote the radial coordinate by K, and the polar coordinate by θ. The analogue of the K-soft local maxima defined in Def. (4.1) for this conceptual example are the points that are local maxima of the conditional probability $P(\theta|K)$, i.e. after we project onto a fixed value of K. For illustration, two example K-soft local maxima are shown, denoted by circles and labeled γ_{10} (blue) and γ_{15} (green; see arrows). (B) After projecting onto the radial coordinate for the respective values of K, we see that γ_{10} is a local maximum of $P(\theta|K = 10)$ and likewise that γ_{15} is a local maximum of $P(\theta|K = 15)$. (C) Illustrated is the projection onto the polar coordinate, for an arbitrary fixed value of θ. Since soft local maxima are local maxima of a function defined on a lower-dimensional subspace of the original domain, they are candidates for a type of "ridge point," and as shown in Panel (A) could theoretically form discrete "ridges" that span across different K levels.

it was motivated by the sparseness of neural activity, and follows directly out of necessity when imposing the constraint to preserve spike count. Note also that this algorithm is used here solely as a numerical tool to explore the probability landscape over all possible ganglion cell activity patterns; it does not relate in any way to the real dynamics of retinal light responses. To check for the presence (or absence) of soft local maxima in the probability landscape, we implemented the opposite sign neuron pair relaxation algorithm for two large datasets. Based on the results of another analysis (see Methods), which demonstrated that the Tree HMM more accurately captures the

empirical ($K = 2$)-soft local maxima than the K-Pairwise MaxEnt model, we chose the Tree HMM as our underlying probability model.

We first examined the dataset of 152 ganglion cells responding to the non-repeated natural movie (Movie #2). For all spike counts $K > 1$ examined, we found many soft local maxima (Fig 4.6A). Moreover, the number of identified unique soft local maxima monotonically increased with spike count. We observed qualitatively similar results when we performed this analysis on the same dataset, but with responses discretized using a different temporal bin width (see the Appendix). Since the opposite sign neuron pair relaxation algorithm is stochastic, we checked how robust the mapped set of unique K-soft local maxima was. To do so, for each value of K we performed 100 independent iterations of the opposite sign neuron pair relaxation algorithm (see Methods). We then computed the mean pairwise overlap ratio between the identified sets of unique K-soft local maxima for each of the $\binom{100}{2}$ iteration pairs. As seen in Fig 4.6A (upper left-hand inset), for $1 \leq K \leq 3$, the identity of the mapped set of unique soft local maxima was perfectly conserved. For $4 \leq K \leq 7$, the mean pairwise overlap ratio was also large ($> 94\%$), indicating a high degree of robustness.

We next examined the responses of 170 ganglion cells to the original version of Movie #4. We similarly found a proliferation of soft local maxima for all spike counts examined, and that the number of unique soft local maxima monotonically increased with K (Fig 4.6C). For both datasets, the distributions of the proportion of observed population responses mapped to each K-soft local maximum was nearly uniform for the top-ranked soft local maxima (Fig 4.6B,D). This is in stark contrast to the analogous local maxima results, in which $> 99\%$ of observed population responses were mapped by single spin flip ascent to a single local maximum (Fig 4.2A,C).

We have shown that soft local maxima are present in the probability landscape of ganglion cell population responses under both non- and moderately-repeated visual stimulation. But how pronounced is each soft local maximum in terms of its peak-

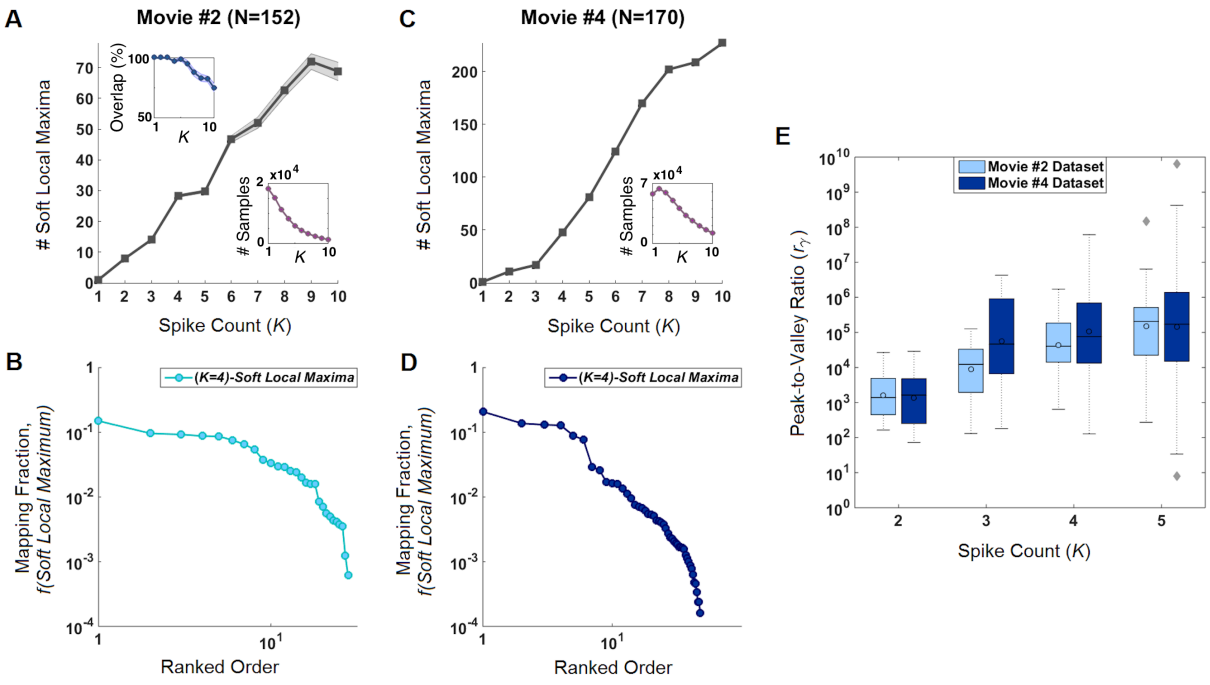

Figure 4.6: **Soft local maxima results for two datasets.**
Shown are the soft local maxima results for (A,B) the dataset of 152 ganglion cells responding to Movie #2; and (C,D) the dataset of 170 ganglion cells responding to the original version of Movie #4. (A) Shown in grey is the mean number of unique K-soft local maxima identified (y-axis) for each examined spike count level K, averaged over 100 mapping iterations. Error bars denote one standard deviation over the 100 iterations. *Upper Left Inset*: The mean pairwise overlap ratio (see the Methods) for each value of K (blue), averaged over all 4950 iteration pairs. Error bars denote one standard deviation. *Lower Right Inset*: Total number of observed population responses in the data (purple) for each spike count level. (B) Example results for a specific spike count level K, arbitrarily chosen here to be $K = 4$. Shown is a log-log plot of the proportion of the 8149 observed responses in the data with 4 spikes that were mapped to the corresponding ($K = 4$)-soft local maximum indicated on the x-axis. (E) Box plot of the estimated peak-to-valley ratios of identified K-soft local maxima, denoted $r_{\vec{\gamma}}$, for each spike count level K (see Methods).

to-valley ratio in the probability landscape? To quantify this, we computed a proxy measure that we denote $r_{\vec{\gamma}}$, which is a lower bound on the peak-to-valley ratio of a given soft local maximum $\vec{\gamma}$ (see Methods). This ratio was quite high for nearly all soft local maxima, and moreover systematically increased with spike count (see Fig 4.6E).

4.3.3 Ridges are a feature of the RGC response probability landscape

So far we have focused on the set of all activity patterns having a fixed number of spikes, K. Within the disciplines of computer vision and differential geometry, there is a well-studied notion of ridges, which are curves or hypersurfaces composed of "ridge points" [36, 48]. Although multiple definitions for ridge points exist, one popular definition is the *height definition* [36]. This definition stipulates that a necessary condition for a point $\vec{\sigma}$ to be a ridge point of a multivariate function g is that it must be a *generalized maximum*, which conceptually is a local maximum of g when we restrict our search to a subspace of the function's domain. Although generalized maxima are technically only defined for continuous functions on vector space domains, we can see that the K-soft local maxima defined in Definition 4.1 are a type of discrete analogue, where we have chosen the restricted metric subspace of the domain (which is the Hamming cube in our case) to be the set of all activity patterns with a fixed spike count, K.

Motivated by this potential conceptual connection to ridge points, we next sought to investigate whether the soft local maxima found in the previous section are naturally organized *across* spike count levels such that they comprise discrete "ridges". Toward this purpose, we introduced a new definition in our work in [63]:

Definition 4.2. A soft local maximum $\vec{\gamma}$ is *u-reachable* from another soft local maximum $\vec{\sigma}$ if and only if $w_H(\vec{\gamma}) = w_H(\vec{\sigma}) + 1$, and $\exists$ neuron $i \in$ the set of neurons that have a silent response, $\mathcal{S}(\vec{\sigma})$, such that:

(i) Performing opposite sign neuron pair relaxation on $\vec{\sigma}^{(i)}$ results in $\vec{\gamma}$, and

(ii) $P[\vec{\sigma}^{(i)}] < P[\vec{\sigma}]$ (to ensure that $\vec{\sigma}^{(i)}$ is not a local maximum)

where $w_H(\cdot)$ denotes spike count, $\vec{\sigma}^{(i)}$ denotes the population response pattern that differs from $\vec{\sigma}$ only in switching neuron i's state from silent to active, and $\mathcal{S}(\vec{\sigma}) \equiv \{\text{neurons } j \mid \sigma_j = 0\}$.

We describe the intuition for this definition in the following. Intuitively, we want to explore whether there are natural 'links' between soft local maxima across the different spike count levels. If we focus on two spike count levels K and $K+1$, then we can state this more concretely as wanting to identify the $(K+1)$-soft local maxima that are 'linked' in some intuitive way to a given K-soft local maximum of interest, $\vec{\sigma}_K$. One natural way to think of being 'linked' is in terms of being as 'near' as possible, with regard to minimizing the number of computations needed to transition from $\vec{\sigma}_K$ to a $(K+1)$-soft local maximum. If the allowable computations are single spin flips (i.e. modifying one neuron's output activity) and opposite sign neuron pair relaxation, then the minimum number of computations possible to accomplish this transition is one implementation of each. This is because transitioning from $\vec{\sigma}_K$ to any activity pattern in the $(K+1)$-th spike count level will require at a minimum flipping one of $\vec{\sigma}_K$'s silent neurons to an active state. We let $\vec{\sigma}_{K+1}$ denote the activity pattern that differs from $\vec{\sigma}_K$ only in having one extra active neuron. Since $\vec{\sigma}_{K+1}$ is not guaranteed to be a soft local maximum, we must additionally allow for one implementation of opposite sign neuron pair relaxation applied to $\vec{\sigma}_{K+1}$, which will ensure transitioning to a $(K+1)$-soft local maximum, $\vec{\gamma}_{K+1}$. When it is possible to transition from a K-soft local maximum $\vec{\sigma}_K$ to a $(K+1)$-soft local maximum $\vec{\gamma}_{K+1}$ via the composition of one spin flip and one implementation of opposite sign neuron pair relaxation, then we say that $\vec{\gamma}_{K+1}$ is *u-reachable* from $\vec{\sigma}_K$ (see Definition 4.2 and Fig 4.7A). This formalism allows us to define a progression of 'linked' soft local maxima spanning across low to high spike count levels.

An analogous notion for exploring the organization of soft local maxima across high to low spike count levels is similarly defined (also see Fig 4.7B):

Definition 4.3. A soft local maximum $\vec{\gamma}$ is *d-reachable* from another soft local maximum $\vec{\sigma}$ if and only if $w_H(\vec{\gamma}) = w_H(\vec{\sigma}) - 1$, and $\exists$ neuron $i \in$ the set of neurons that have an active response, $\mathcal{A}(\vec{\sigma})$, such that:

(i) Performing opposite sign neuron pair relaxation on $\vec{\sigma}^{(i)}$ results in $\vec{\gamma}$, and

(ii) $P[\vec{\sigma}^{(i)}] > P[\vec{\sigma}]$

where $\vec{\sigma}^{(i)}$ denotes the population response pattern that differs from $\vec{\sigma}$ only in switching neuron i's state from active to silent, and $\mathcal{A}(\vec{\sigma}) \equiv \{\text{neurons } j \mid \sigma_j = 1\}$.

To visualize how the soft local maxima are organized across low to high spike count levels, we used a standard class of search algorithm [113]. We will here refer to our specific variant of this algorithm as the *ridge search algorithm*, and we provide an intuition for how it works in the following (see Methods and the Appendix for details). Our ridge search algorithm takes as input a single soft local maximum of interest, $\vec{\gamma}_{\text{root}}$. We will denote the spike count of $\vec{\gamma}_{\text{root}}$ by $K_{\min}$, as this specifies the starting (and thus lowest) spike count level. In practice we chose $K_{\min} = 4$, and chose each input to be one of the identified $(K = 4)$-soft local maxima in the preceding section.

Starting at $\vec{\gamma}_{\text{root}}$, the algorithm then proceeds to find all soft local maxima at the next higher spike count level, $K_{\min} + 1$, that are 'linked' in terms of being u-reachable from $\vec{\gamma}_{\text{root}}$. We call the set of $(K_{\min} + 1)$-soft local maxima that are u-reachable from $\vec{\gamma}_{\text{root}}$ the *neighborhood* of $\vec{\gamma}_{\text{root}}$. For each $(K_{\min} + 1)$-soft local maximum in the neighborhood of $\vec{\gamma}_{\text{root}}$, the algorithm then proceeds to find *its* neighborhood of linked soft local maxima that reside in the next higher spike count level, $K_{\min} + 2$. This procedure is iterated recursively up to a specified maximum spike count level, $K_{\max}$.

In this way, our ridge search algorithm traces out all of the connection paths that start at the input soft local maximum $\vec{\gamma}_{\text{root}}$ and extend out to the highest specified spike count level. This information can be readily represented by a standard structure

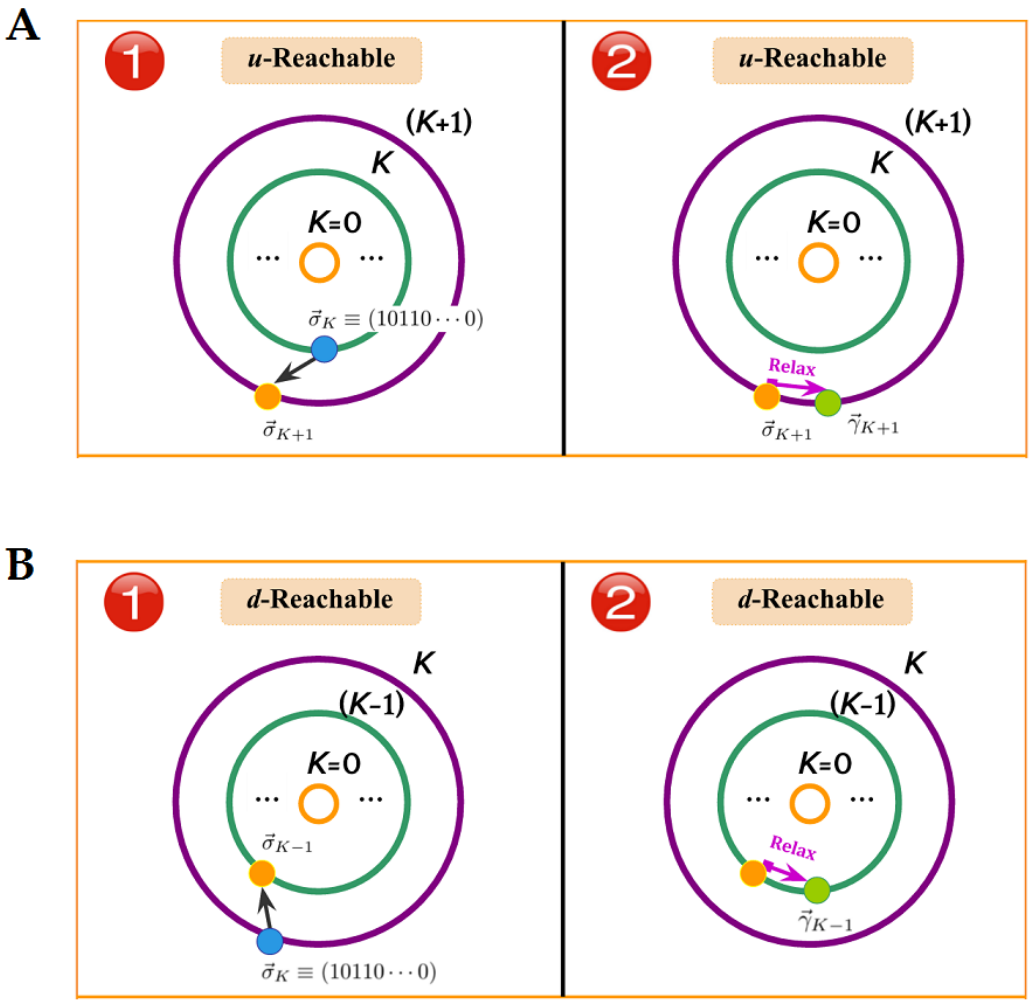

Figure 4.7: **Schematic illustrating the notions of *u-reachable* and *d-reachable*.** (A) In this cartoon diagram, $\vec{\sigma}_K$ (blue dot) represents a K-soft local maximum, and $\vec{\gamma}_{K+1}$ (green dot) represents a $(K+1)$-soft local maximum. The different metric subspaces Ω_K of the full joint response space $\{0,1\}^N$, which are defined as $\Omega_K \equiv \{\vec{\sigma} \in \{0,1\}^N \mid w_H(\vec{\sigma}) = K\}$ and which we also refer to as "spike count levels", are represented by colored concentric circles. The center-most yellow circle represents the 0th spike count level. In this example, $\vec{\gamma}_{K+1}$ is u-reachable from $\vec{\sigma}_K$. This is because: (1) there exists a silent neuron i such that changing neuron i's instantaneous response to spiking (represented by the black arrow) will (2) result in a joint response pattern $\vec{\sigma}_{K+1}$ (orange dot) at the $(K+1)$-th spike count level that will be mapped via opposite sign neuron pair relaxation (represented by the purple arrow) to $\vec{\gamma}_{K+1}$. (B) In this cartoon diagram, the $(K-1)$-soft local maximum $\vec{\gamma}_{K-1}$ (green dot) is d-reachable from the K-soft local maximum $\vec{\sigma}_K$ (blue dot).

in graph theory called a *rooted digraph* [115]. We chose this graph theoretic representation to visualize the output of our ridge search algorithm (see Methods), because it is a $2D$ representation and hence inherently conducive to visualization, whereas the

soft local maxima reside in a high-dimensional space. Shown in Figs 4.8 and 4.9 are example digraph visualizations that were obtained when we applied our ridge search algorithm to the dataset of 152 ganglion cells responding to the non-repeated natural movie (Movie #2).

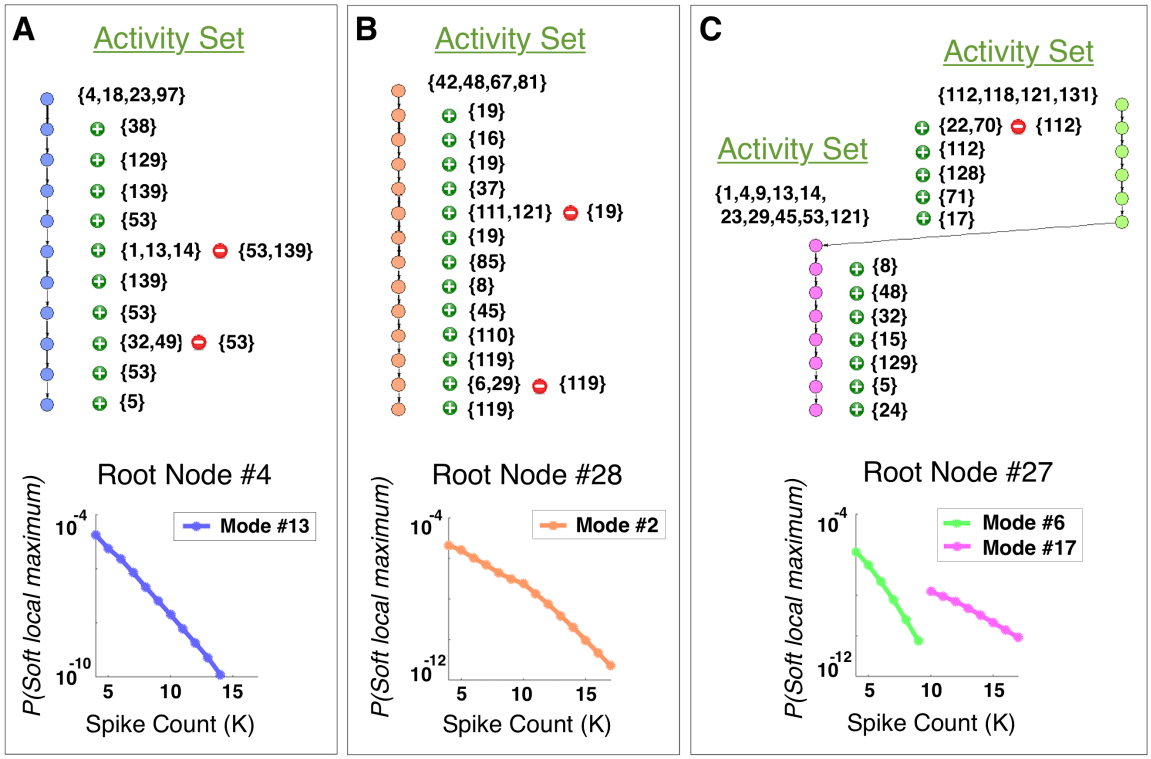

Figure 4.8: **Example Type 1 rooted digraphs obtained.**
Results shown are for the dataset of 152 ganglion cells responding to Movie #2. (A) Output digraph obtained via the ridge search algorithm when the unique $(K = 4)$-soft local maximum with rank 4 is the input. Labeled next to each node is the "activity set" of the corresponding soft local maximum $\vec{\gamma}$, defined as $\mathcal{A}(\vec{\gamma}) \equiv \{\text{neurons } i \,|\gamma_i = 1\}$. For ease of visualization, the activity set of the starting node is written out in full. The activity set of each subsequent $(K+1)$-soft local maximum, $\vec{\gamma}_{K+1}$, is then denoted by the neuron(s) that is added to (green plus symbol) or removed from (red minus symbol) the previous K-soft local maximum's activity set to achieve $\mathcal{A}(\vec{\gamma}_{K+1})$. Node color denotes the MAP estimate of the associated collective mode. There was one distinct ridge for this example; all soft local maxima were associated with the same collective mode (blue). Below: Probability, as modeled by the Tree HMM, of each depicted soft local maximum vs. its spike count, K. (B,C) Same format as in panel (A), but for two other input $(K = 4)$-soft local maxima.

Since we can view soft local maxima as analogous to ridge points, this approach leads to an algorithmic definition of a discrete ridge. Specifically, a discrete ridge is the set of soft local maxima that are organized across low to high spike count levels

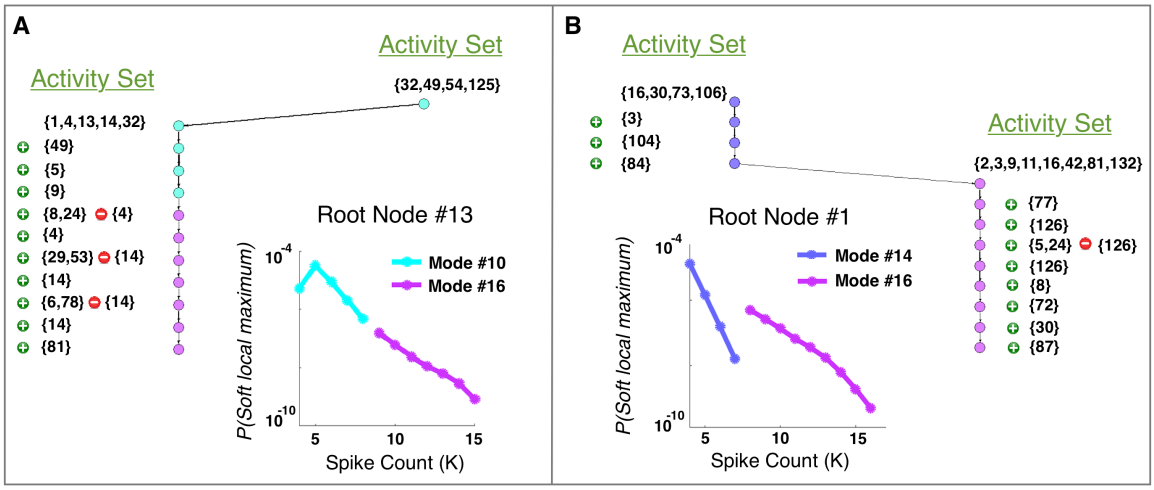

Figure 4.9: **Example Type 2 rooted digraphs obtained.**
Results shown are for the dataset of 152 ganglion cells responding to Movie #2. (A) Digraph obtained when the $(K = 4)$-soft local maximum with rank 13 was taken as the input to the ridge search algorithm. Notation is the same as in Fig 4.8. There was one distinct ridge; however, the soft local maxima comprising this ridge were associated with two possible collective modes: #10 (cyan) and #16 (violet). (B) Digraph obtained when the $(K = 4)$-soft local maximum with rank 1 was taken as the input. There were two distinct ridges. The first, which was comprised of soft local maxima that were all associated with the same collective mode (#14, blue), was Type 1. Soft local maxima comprising the second ridge were associated with collective mode #16 (violet). Since mode #16 corresponded with three distinct ridges, two of which are shown here in panels (A) and (B), this second ridge was classified as Type 2.

such that each respective pair of K- and $(K+1)$-soft local maxima is 'connected', in terms of being u-reachable. Note that in principle, the ridge search algorithm could trace out a single ridge starting at an input $\vec{\gamma}_{\text{root}}$ (as in the case when there is only one u-reachable soft local maximum at each successively higher spike count level), or could trace out multiple ridges. The number of ridges traced/visualized is a specific property of the dataset. We observed that many of the digraphs for the Movie #2 dataset exhibited a single ridge (Fig 4.8A,B).

In principle, it is also possible that a discrete ridge descending from $\vec{\gamma}_{\text{root}}$ could terminate at a lower spike count level K_* than the arbitrarily-chosen $K_{\max}$. An example occurrence of this scenario is shown in Fig 4.8C. To aid in visualizing distinct ridges, we further assigned an x- and y-coordinate to each node in the digraph visualizations. The intuition behind this x-coordinate assignment was to quantify whether

or not there is a 'jump' between ridges based on the amount of overlap between the different soft local maxima (see Methods for formal details).

4.3.4 Collective modes correspond to ridges

For each soft local maximum $\vec{\gamma}$ in each digraph, we also computed the maximum *a posteriori* (MAP) estimate of the collective mode that was most likely to be associated with it. To visualize this information, we then assigned a color to each node in the digraphs, which uniquely identifies the associated collective mode. Overall, there was a strong tendency for each distinct ridge to be comprised of soft local maxima that were associated with the same collective mode (Fig 4.8).

To quantify the degree of correspondence between ridges and collective modes, we first identified the set of all unique ridges based on the compiled digraph results (for each dataset analyzed), and then classified each unique ridge as belonging to one of two disjoint categories: *Type 1* or *Type 2*. Formally, we classified a ridge as Type 1 if it corresponded with exactly one collective mode, which moreover uniquely corresponded with the ridge (that is, if there was a bijection between the ridge and a single collective mode). Conceptually, a Type 1 classification denotes that the given ridge has a 'perfect' correspondence with one of the collective modes. We classified any ridge that was not Type 1 as Type 2. Note that we chose to classify on the basis of ridges, rather than on the basis of the digraph examples, to avoid potentially double-counting ridges when computing global statistics.

Applying this classification scheme to the dataset of 152 ganglion cells responding to Movie #2, we found that 13/17 identified unique ridges, i.e. 76.5%, were Type 1. To investigate if this result is also obtained for a different temporal binning of the data, we performed our full analysis procedure on the same Movie #2 dataset, but with the responses binned at $\triangle t = 16.7$ ms (i.e. matching the 60 Hz frame rate of the video). We likewise observed a strong correspondence between ridges and

collective modes for $\triangle t = 16.7$ ms; specifically, 12/14 identified unique ridges (i.e. 85.7%) were Type 1. Comparably, for the dataset of 170 ganglion cells responding to the original version of Movie #4, we found that 32/44 identified unique ridges, i.e. 73%, were Type 1. Hence, for both a non-repeated as well as a moderately-repeated stimulus ensemble, and for two different time bin widths used to discretize responses, we found that there was a substantial correspondence between the statistically-defined collective modes and the geometrically-defined ridges. In the remaining cases where there was not a 'perfect' correspondence, either the ridge was associated with two collective modes (Fig 4.9A), or a collective mode was associated with more than one ridge (Fig 4.9B).

4.4 Ridges correspond to "neuronal communities"

As shown in Fig 4.8, even in the few cases where there was not a perfect one-to-one correspondence between a given geometric ridge and a collective mode, intriguingly, we consistently found that each identified ridge corresponded with a specific group of active neurons. Specifically, for each ridge, which spans across multiple spike count levels, K, we observed that the active neuron sets of the soft local maxima comprising that ridge were nested (Figs 4.8 and 4.9). (By the "active neuron set" of a population response pattern $\vec{\gamma}$, we mean the set of neurons that have an instantaneous spiking response for that pattern, i.e. $\mathcal{A}(\vec{\gamma}) \equiv \{\text{neurons } i \mid \gamma_i = 1\}$). Stated another way, we found that population response patterns within each ridge exhibit active neurons that are members of an identifiable group of ganglion cells, combined with silence of all neurons outside of this group. We call the identifiable group of active neurons that is specific to each ridge the *neuronal community* associated with that ridge.

We next investigated how the full population of ganglion cells is organized in terms of these ridge-associated neuronal communities. To do this, for each ganglion

cell i (where $1 \leq i \leq N$) we recorded all of the neuronal communities it was a member of, based on the compiled digraph results from the previous section. We then visualized this information in the form of an undirected graph, a standard structure in graph theory [115]. Since this graph visualizes the compilation of the ridge-associated neuronal community results for an entire dataset, we refer to it as the *ridge union graph*. Each node in the ridge union graph represents a neuron in the population of ganglion cells, and an edge (i, j) is present if ganglion cells i and j are members of at least one common neuronal community (see Methods for details).

The ridge union graph for the dataset of 152 ganglion cells responding to the non-repeated natural movie is shown in Fig 4.10. As seen in Fig 4.10, mixed membership - that is, the case of a neuron having membership in more than one neuronal community - was prevalent. Specifically, averaged over the 83 retinal ganglion cells that were assigned to at least one neuronal community, a given ganglion cell belonged to a mean ($\pm$ SEM) of 2.5 ± 0.18 neuronal communities. This high degree of mixed membership may be a signature of a combinatorial neural population code.

In network science, *community structure* refers to the occurrence of groups of nodes in a network that are more densely connected internally than with the rest of the network [40]. Thus, another way to state our result is that we observed that the geometric picture of ridges in the joint response probability landscape naturally maps to one of communities in the population of ganglion cells. We performed two control analyses (see Methods for details) to verify that this organization of the ganglion cell population into many, relatively small neuronal communities is a specific property of the data, and not solely attributable to the underlying probability model.

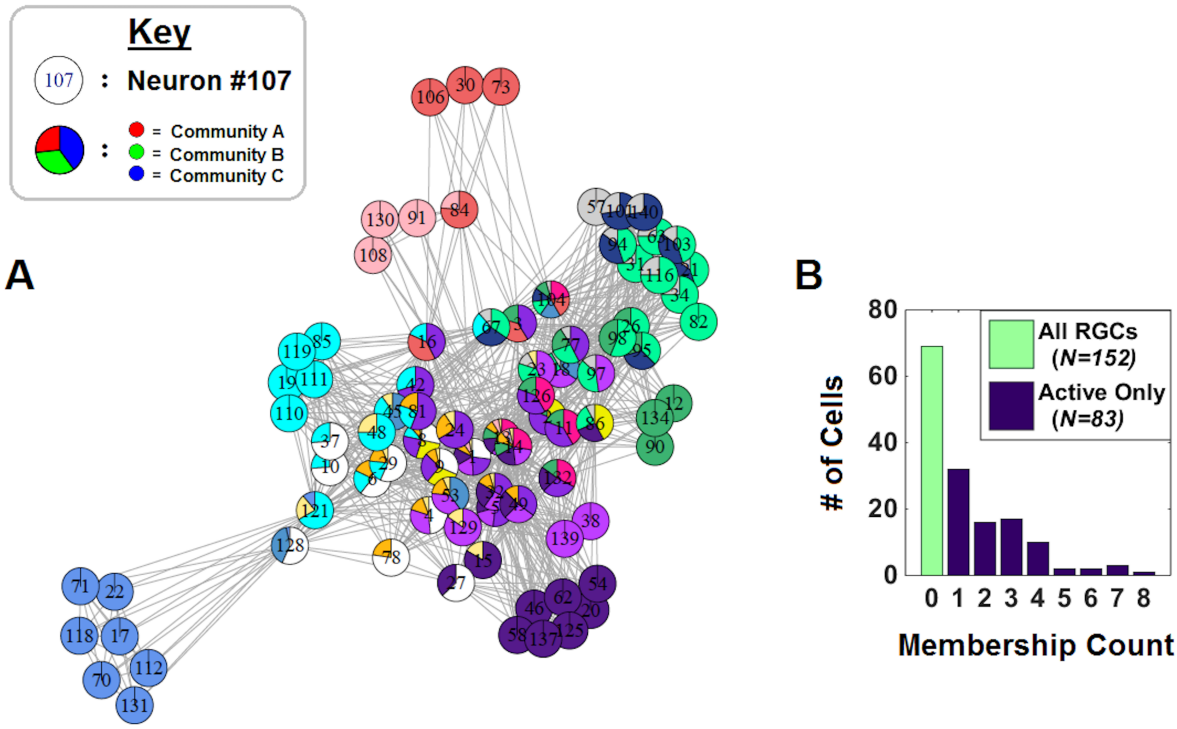

Figure 4.10: **Empirical neuronal communities.**
(A) Shown is the *ridge union graph* for the dataset of 152 ganglion cells responding to Movie #2. Nodes represent individual ganglion cells in the population; each inscribed number indicates the index of the corresponding ganglion cell. The presence of an edge (i, j) denotes that neurons i and j share at least one common ridge-associated neuronal community. Each color uniquely identifies one of the 17 distinct ridge-associated neuronal communities that were identified for this dataset. Ganglion cells that exhibited mixed membership are depicted with "pie" nodes (see Key). (B) Histogram of the distribution of the degree of mixed membership across the population. Represented on the x-axis is the "membership count", defined as the number of ridge-associated communities a given ganglion cell was a member of. Green denotes the same as purple, except that it also includes the 69 ganglion cells that were not a member of any community.

4.5 Discussion

4.5.1 Dependence of the population response probability landscape on stimulus ensemble statistics

Arising from the maximum entropy literature has been the proposal that neuronal population codewords may correspond with local peaks in the joint response probability landscape [100, 116] (Fig 4.1). However, these studies analyzed population activity

driven by many repeats of a short visual stimulus segment, which corresponds to a narrow range of variability in stimulus space. Here, we found that under low-repeat, i.e. broadly variable, stimulus ensembles, the joint response probability landscape was instead mostly devoid of local peaks (Figs 4.2 and 4.3).

Why might the repeat structure of the stimulus ensemble affect the probability landscape so strongly? Our suggestion is that each stimulus elicits an average population response, and that repetition of the same stimulus produces scatter around that average response due to neural noise (Fig 4.11Ai). If the scatter due to noise (represented by black dots in Fig 4.11Ai) is small compared to the separation of the average response (red dot in Fig 4.11Ai) from responses evoked by other stimuli, then sampling over repeats will produce a local peak in the probability landscape. Such local peaks can be detected by an iterative hill climbing algorithm (Fig 4.11Aii). We expect that in the limit of many repeats of few stimuli, this scenario will be achieved.

However, in the case of a broader variety of stimuli, there is greater likelihood that average responses will be less well separated in response space. In this case, local peaks merge together (Fig 4.11B). In principle, this merging could simply result in fewer local peaks with negligible additional structure in the probability landscape. However, neural activity is often sparse, which causes there to be relatively more average population responses with low spike count than high. Due to this bias, the merged peaks can instead form a ridge. In this case, the same hill climbing algorithm will ascend to the all-silent state, which will be the global peak for sufficiently sparse neural activity (Fig 4.11Bii). Notice that when the probability landscape is structured into ridges, the iterative single spin flip ascent algorithm can map neural activity states onto the all-silent global peak even if those states have high spike count. This is the result we found in our analysis (Figs 4.2 and 4.3).

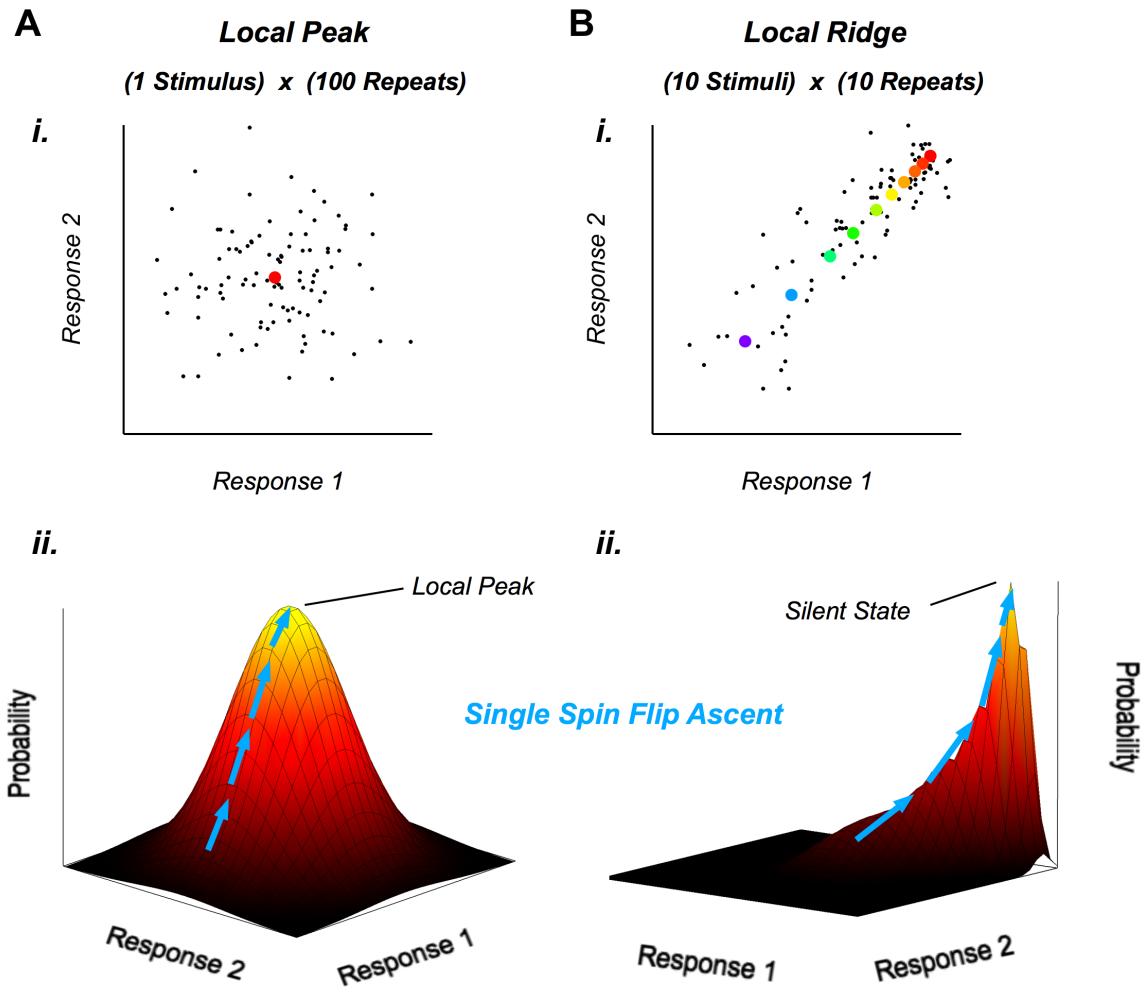

Figure 4.11: **Schematic of dependence on stimulus ensemble statistics.** (A) *i.* Example of neural responses shown in a 2D space triggered by 100 repeats of the same stimulus (black dots) along with the average response (red circle). *ii.* The corresponding probability landscape contains a local peak, and single spin flip ascent maps neural activity to the local peak (blue arrows; each arrow represents one iteration comprising the ascent algorithm). (B) *i.* Example of neural responses triggered by 10 repeats of 10 different stimuli (black dots) along with average responses (circles with different colors for each of the 10 stimuli). *ii.* Here, the corresponding probability landscape is a local ridge, and single spin flip ascent maps neural activity to the all-silent state, regardless of spike count (blue arrows).

4.5.2 Comparison of two candidate definitions of retinal population codewords

Energy basins, which are equivalent to local probability peaks [1], and collective modes are candidate definitions of population codewords that were proposed independently by past studies that applied fundamentally different modeling approaches. In past work that used the Maximum Entropy framework, energy basins in the modeled energy landscape were shown to exhibit error correction for the high-repeat stimulus regime [116]. Likewise, using a hidden Markov model framework, it was recently shown that collective modes (which are the clusters of population responses that map to the same latent state of the HMM) exhibit strong error correction, and moreover provide a new feature basis set than ganglion cell receptive fields [87]. The explicit relationship between these candidate codewords was previously unknown, and is an empirical result of the present study (summarized in Table 4.1).

Our results in the preceding section demonstrate that these concepts are not equivalent (see Fig 4.4C). Further, they support the conclusion that collective modes are a better population codeword candidate, as they are robustly present in both the low-repeat and high-repeat stimulus regimes. We argue that both of these stimulus regimes are relevant in different behavioral contexts.

Note that we did not perform the full ridge analysis on the probability landscape modeled by the K-Pairwise Maximum Entropy model, because results from another analysis (see Methods) demonstrated that the K-Pairwise MaxEnt model does a much poorer job of capturing the empirical ($K = 2$)-soft local maxima than the Tree HMM.

[1] Because "energy" is *defined* as negative log probability, $E \equiv -\ln P$, local energy basins necessarily have the geometry of local peaks in the probability landscape (see Fig 4.1) [100]

	Candidate Codeword	
	Energy Basin	Collective Mode
Error-Correcting Algorithm	single spin flip ascent	MAP estimation
High-Repeat Stimulus Regime	present	present
Low-Repeat Stimulus Regime	absent	present
Geometry in Prob. Landscape	local maxima[†]	ridges[‡]
Correlate in Network of Neurons	unknown	neuronal community

Table 4.1: Comparison of two independently-proposed codeword candidates of the RGC population code. Row 2 was shown by past work [87, 100, 116]. Row 3 and rows 4 and 5 for the "Collective Modes" column are restatements of our results.

[†]By definition

[‡]At least for the low-repeat stimulus ensemble regime

Comparison to other definitions of neural population "codewords"

There have been many different proposals for how to study population neural codes. One particularly formative approach has been to apply information-theoretic techniques [14, 90, 93, 102]. In past work that applied such information-theoretic methods to quantify combinatorial coding in the salamander and guinea pig retina, it was found that whereas synchronous spiking was mostly redundant, combinations of spiking and silence were generally synergistic [102]. Our community results are consistent with and can be viewed as a generalization of this previous result to populations of $N > 100$ neurons.

Another recent approach has been to investigate the possibility of clustering in the response space using a semantic distance metric [43]. In this work, it was shown that the studied 20-bit ganglion cell population responses nontrivially cluster in the response space based on semantic similarity. Unfortunately, we cannot directly compare

the ridges or neuronal communities found here with the previously reported semantic clusters, since the approach used in [43] is a supervised approach that requires a high-repeat stimulus regime. In contrast, our analyses were applied to the non-repeated stimulus regime. Given our finding that the ganglion cell population response probability landscape is proliferated by local maxima for the high-repeat regime, we expect that semantic clusters will correspond geometrically to local maxima.

However, we speculate that the probability landscape in the vicinity of such local peaks still has an extended, ridge-type geometry. The finding in [43] that simple linear and bilinear distance metrics do not correspond with semantic distance is thus consistent with our findings, as we would not expect simple linear or bilinear clustering routines to succeed well in capturing a ridge-type geometry. The reason for this intuition is that each ridge extends out from the all-silent state in a different "direction" in the response space (Fig 4.5), while linear and bilinear distance metrics use the same parameters everywhere on the probability landscape. Finally, our neuronal community results are also highly consistent with the finding that population responses belonging to the same semantic cluster exhibited a common subset of neurons that were always active, and other neurons that were always silent [43].

Other structural correlates of collective modes

It is important to emphasize that for both the Maximum Entropy models and the Tree hidden Markov model, the associated model fitting and error-correction algorithms are non-biologically plausible computations. In order to explore biologically plausible algorithms that can learn the population codewords identified by the HMM (see Chapter 5), it would be useful to understand whether the collective modes have other, potentially biologically-relevant structural correlates. The second main goal of this paper was to address this question. Our main results are that the statistically-defined collective modes (1) have the geometry of *ridges* in the probability landscape

of neural population activity; and (2) closely correspond to *neuronal communities* within the population of ganglion cells, a notion from network science [40].

The mapping between the ridges (and hence collective modes) and neuronal communities arises from the following nontrivial property: for each given ridge, which spans across multiple spike count levels, we observed that the active neuron sets of the soft local maxima constituting that ridge were nested (Figs 4.7 and 4.8). Stated another way, we found that all soft local maxima within a given ridge exhibit active neurons that are members of an identifiable group of ganglion cells - which we call the *neuronal community* associated with that ridge - combined with silence of all neurons outside of this group. Thus, the identity of the active and especially the silent neurons appears to be crucial in defining the population codeword. In contrast, the identity of the community is invariant to the precise number of active neurons above some threshold. Fundamentally, it is this invariance to the number of active neurons that gives rise to error correction.

Note that formally, the community identities are sensitive to the highest spike count level we allow our ridge search algorithm to explore out to, $K_{\max}$. However, due to the nested property and subsequent invariance of community identity to spike count, even if a given ridge extends to a higher spike count level $K_* > K_{\max}$, the associated community we identify by exploring out to only $K_{\max}$ will be very likely be conserved. Intuitively, if we think of community identity as the value of the "angle" of neural activity in the response space, θ in Fig 5, then the nested property implies that a given ridge will have the same value of θ for K_* and $K_{\max}$. We therefore expect that the choice of $K_{\max}$ will not qualitatively affect our results. Also, note that we have here enumerated only the ridges that start at spike count level $K_{\min} = 4$, or branch from a ridge starting at $K_{\min} = 4$. We would expect to find additional ridges (and thus communities) starting at higher $K_{\min}$ values, as we hope to enumerate in future work. The number of communities reported here is therefore a lower bound.

Biologically plausible decoding of neuronal communities

Finally, if this type of structure plays a role in neural coding, then it may be important for downstream brain areas to be able to identify neuronal communities from population activity. Here we determined community identity by starting with the ridge union graph for the ganglion cell population (Fig 4.10), and then using established simple community detection algorithms (see Methods). However, downstream brain areas do not have access to this network information.

Interestingly though, a simple, biologically plausible decoding algorithm exists for detecting neuronal communities. This algorithm (schematized in Fig 4.12) consists of feedforward excitatory and inhibitory synapses from the neural population onto a given readout unit, which will fire only if the community is present in its input population activity. Specifically, for each member of the active set of neurons, the synapse should be excitatory. If, for example, these synapses had uniform weight, then the threshold of the readout unit would need to be $K_{\min}$, with an output that saturates with just one spike for $K > K_{\min}$. In this manner, the readout unit would be active if at least a criterion number of neurons in the active set of the community fire spikes.

For each member of the silent neuron set, there would then need to be a disynaptic pathway where activity drives a local inhibitory cell, which feeds forward onto the readout neuron. If this inhibitory neuron could strictly veto the firing of the readout unit - as is expected for chandelier-type inhibitory cells in the neocortex - this would enforce the condition that the readout unit would be silent if any of the silent set of input neurons fires. Finally, by making different choices of which cells have an excitatory synapse and which have a disynaptic inhibitory connection, different readout units would be selective for different neuronal communities within the same input neural population.

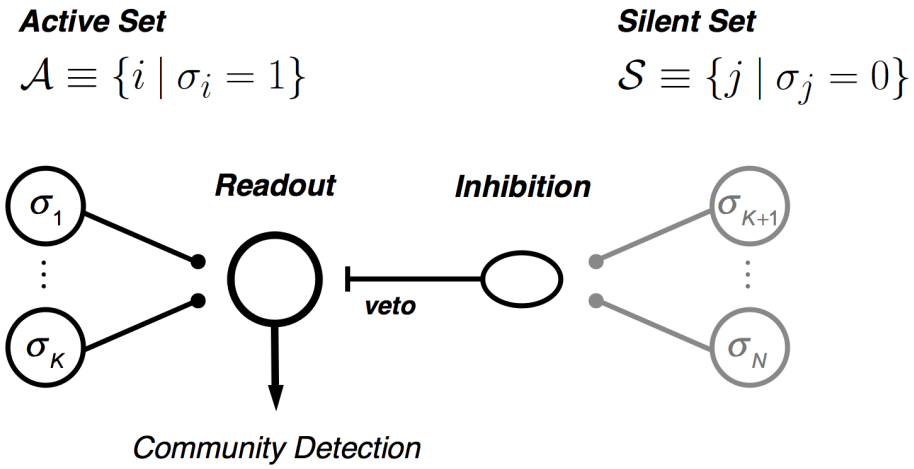

Figure 4.12: **Schematic of biological decoding algorithm for neuronal communities.** All of the K neurons in the active set (black circles, left side) make excitatory synapses onto the readout unit (large circle). The readout unit can be active as long as its input excitation is at least $K_{\min}$. All of the $(N-K)$ neurons in the silent set (grey circles, right) make excitatory synapses onto a local inhibitory interneuron (black oval). This neuron then feeds forward and vetoes activity in the readout unit. This vetoing operation ensures that all members of the silent set must be silent in order for the readout unit to be active.

Note that we are here imagining that the recognition of neuronal communities might occur within the neocortex. Of course, retinal ganglion cells project directly to the lateral geniculate nucleus (LGN). However, relay cells in the LGN have receptive fields very similar to those of retinal ganglion cells [112], consistent with each relay cell receiving driving inputs from one or a few ganglion cells [110, 119]. Thus, we are assuming here that the population code of the LGN will be quite similar to that of the retina. Another major retinal target is the superior colliculus/optic tectum (SC). While chandelier-type inhibitory interneurons have not been described in the SC, this circuit does possess multiple winner-take-all (WTA) mechanisms that underlie the creation of a salience map [55]. These sources of WTA inhibition are functionally similar to the veto operation embodied in this proposed biologically-plausible decoder.

4.5.3 Connection to Hebb's cell assemblies

The organization of the ganglion cell population into a community structure with a high degree of mixed membership (Fig 11) is reminiscent of the "cell assemblies" concept hypothesized by Donald Hebb [50]. The first four properties of Hebb's cell assembly are (paraphrased; see [50, 98]): (i) overlapping set coding of information items; (ii) sparse coding; (iii) dynamic construction and reconstruction; and (iv) dynamic persistence. Our neuronal community results are highly consistent with properties (i) and (ii). Specifically, we found that communities are overlapping sets of neurons: for the non-repeated natural movie dataset, a given ganglion cell was a member of on average 2.5 ± 0.18 different communities, and a given community overlapped with on average 10.7 ± 0.59 other communities. Moreover, any individual community contained a small subset (mean 12.1 ± 1.2 ganglion cells) of the 152 total neurons in the population, consistent with (ii). We cannot comment on how our results relate to Hebb's properties (iii) and (iv), since these are dynamic properties, and the geometric ridges and consequently communities are features that were extracted from the modeled static probability landscape.

Our neuronal community results are also consistent with the final property of Hebb's cell assembly concept: (v) dynamic completion, i.e. activation of a large enough subset of a cell assembly results in activation of the complete cell assembly [50]. Although intrinsic dynamics are not applicable in our case, there is a strong parallel between property (v) and our observation that population responses that had instantaneous activation of a large enough subset of the neuronal community were typically mapped to the same collective mode. It was shown in [87] that the collective modes exhibited a high degree of error correction, which combined with the present work suggests that communities likewise exhibit error correction. In combination, the three properties of overlapping coding sets, sparse coding, and robust completion were previously shown to offer theoretical advantages to overcome the deficiencies of the

single-neuron doctrine [9, 62, 126]. Relatedly, in past work that modeled computation in visual cortex using groups of neurons called "cliques" that satisfied the above three properties, it was derived that a circuit with this structure can take neurons that by themselves are crude and highly unreliable, and create aggregate units that are both extremely precise and highly reliable [73].

4.5.4 Ridges vs. peaks: Which are present during behavior?

The fact that the structure of the probability landscape of ganglion cell population activity is qualitatively different in the high-repeat versus low-repeat stimulus regimes raises an obvious question: which regime corresponds to the visual stimuli falling on the eye during real behavior? Answering this question depends on understanding at a fundamental level how a "stimulus ensemble" is defined from the viewpoint of the retina and downstream brain areas. In particular, there are two important considerations: (1) the duration of sampling of the stimulus space, and (2) whether readout processing is context-specific.

Regarding (1), during natural behavior readout circuits can potentially accumulate sampling over long time periods. In the case that sampling occurs over the entire lifetime of the organism, then the stimulus ensemble will be over all possible natural visual stimuli in the animal's environment. As natural stimuli have been shown to possess conserved second-order statistics and other regularities like contrast scaling [32], such lifetime sampling can converge to a stable result. For the retinal code in this limit, the stimulus ensemble would thus lack repeated features. Thus, when sampling occurs over the lifetime of the animal, we would expect that the probability landscape more closely resembles the ridge (i.e. non-repeated stimulus) regime studied here.

However, regarding (2), past work has shown that the neural code can be multiplexed, so that there are multiple context-specific interpretations that serve different

purposes [37, 70, 122, 123]. It is possible that some downstream circuits that read out the ganglion cell population code could be gated by contextual signals, i.e. that these readout circuits only sample from the ganglion cell population under particular contexts. For example, an animal being in different environmental locations could constitute different contexts, signaled by feedback from the hippocampus; another example is running versus stationary movement, which is known to induce contextual modulation in V1 responses [6]. These types of contextual signals could therefore restrict sampling to subsets of the visual environment in which some visual features repeat frequently. In this case, the probability landscape of ganglion cell population activity would have at least some local peaks that downstream circuits might benefit from identifying.

The generality of ridge-like population codes

There is no aspect of our approach that explicitly refers to properties of the retina: We made no assumptions about cells types, or receptive field properties of neurons, or functional models of feedforward sensory processing (such as the linear-nonlinear model) [20, 94]. In fact, nothing in our approach refers to visual processing or even sensory systems. The abstract nature of this approach therefore suggests that similar results might be found for population neural codes in many regions of the central brain.

4.6 Methodology details

4.6.1 Fitting the K-pairwise maximum entropy model

The analytical form of the K-Pairwise MaxEnt distribution is [116]:

$$P^{(2,K)}\left(\vec{\sigma}\right) = \frac{1}{Z}\, e^{-\mathcal{H}(\vec{\sigma})}\,, \text{ where } \;\; Z = \sum_{\vec{\sigma}\in\{0,1\}^N} e^{-\mathcal{H}(\vec{\sigma})} \tag{4.1}$$

and where the "energy function" $\mathcal{H}$ is given by:

$$\mathcal{H}\left(\vec{\sigma}\right) = \sum_{i=1}^{N} h_i \sigma_i + \frac{1}{2} \sum_{i \neq j} J_{ij} \sigma_i \sigma_j + \sum_{K=0}^{N} \lambda_K \delta_{K, w_H(\vec{\sigma})} \tag{4.2}$$

where $\delta_{a,b}$ is the Kronecker delta function (i.e. $\delta_{a,b} = 1$ if $a = b$ and 0 otherwise), and $w_H(\vec{\sigma}) \equiv \sum_{i=1}^{N} \sigma_i$ is the Hamming weight of the population response $\vec{\sigma} \in \{0,1\}^N$. We used the same learning procedure presented in [116] to compute the parameters of the Hamiltonian for the K-Pairwise Maximum Entropy model given measured constraints. The proof of convergence for the core of this L1-regularized maximum entropy algorithm is given in [35]. The code used to fit the model was written in C++ and Matlab.

4.6.2 Selection of the probability model

We considered two distinct activity models of the joint probability mass function (p.m.f.) of measured ganglion cell population activity: the K-Pairwise MaxEnt model and the Tree hidden Markov model [87, 116]. For our purposes, it is important that the model accurately captures the soft local maxima present (or not) in the empirical probability landscape. Due to limited sampling, it is intractable to determine the full empirical probability landscape and hence to make a complete comparison. However, population responses with low spike count are far better sampled than those with high spike count. We thus computed the empirical ($K = 2$)-soft local maxima.

This was done by performing n iterations ($n = 15$ in practice) of the following cross-validation procedure: For each iteration, we

1. Randomly split the data (i.e. all observed population responses with 2 spikes) into two training sets.

2. For each of the two training sets, we estimated the empirical p.m.f. of an observed population response $\vec{\sigma}$ as:

$$\widehat{P}_{\text{empirical}}\left(\vec{\sigma}\right) = \frac{\sum_{t=1}^{T/2} \delta_{\vec{\sigma},\vec{\sigma}(t)}}{T/2} \tag{4.3}$$

where $\vec{\sigma}(t)$ denotes the t-th observed population response, $T/2$ the total number of training samples, and $\delta_{a,b}$ the Kronecker delta function.

3. For each of the two training sets, we then performed opposite sign neuron pair relaxation on all unique observed responses in the given training half, using the respective estimated empirical p.m.f., to find $(K = 2)$-soft local maxima.

We then computed the union of all unique empirical $(K = 2)$-soft local maxima found over the $2n$ iteration halves. Over 15 iterations (i.e. 30 different splits of the data), we found a total of 68 unique $(K = 2)$-soft local maxima. To check reliability, we then computed the proportion of occurrence of each unique $(K = 2)$-soft local maxima, defined as:

$$\text{Proportion of Occurrence of } \vec{\gamma} := \frac{\sum_{h=1}^{n} \mathbb{1}_{\mathcal{S}_h}(\vec{\gamma})}{2n} \tag{4.4}$$

where $\vec{\gamma}$ denotes the soft local maximum of interest, $\mathcal{S}_h$ denotes the set of unique $(K = 2)$-soft local maxima found after performing opposite sign neuron pair relaxation on the data in half h (where $1 \leq h \leq 2n$), and $\mathbb{1}$ denotes the indicator function (i.e. $\mathbb{1}_{\mathcal{S}_h}(\vec{\gamma}) = 1$ if $\vec{\gamma} \in \mathcal{S}_h$, 0 otherwise). Intuitively, the proportion of occurrence measures what fraction of all $2n$ data halves contain a given soft local maximum, $\vec{\gamma}$. The resulting proportions of occurrence are shown in Fig 4.13 (black trace).

It is possible that differences obtained between the two splits of the data in the exact identities of the empirical soft local maxima could arise from noise, attributable to limited sampling upon halving the data. To account for this possibility, we also performed a "relaxed" version of the above empirical analysis.

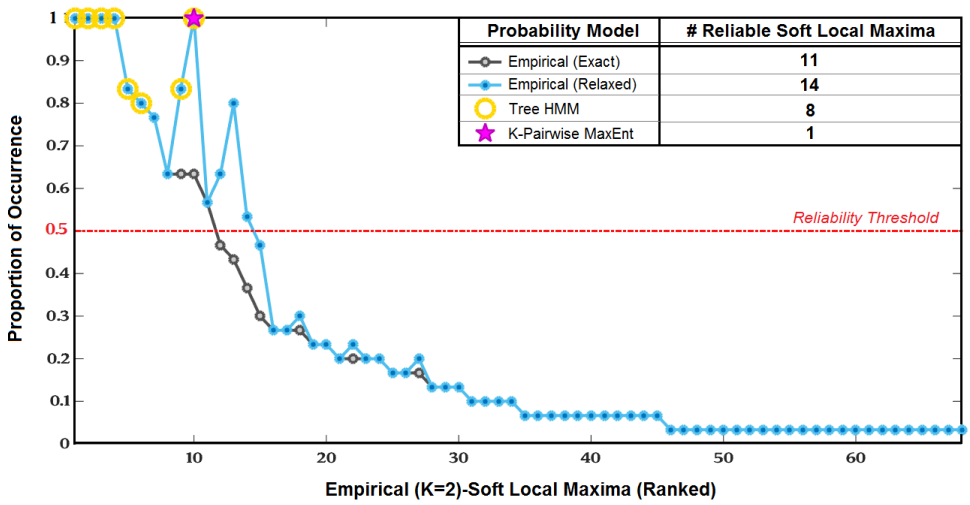

Figure 4.13: **Comparison of different models to empirical (K=2)-soft local maxima.** Shown in grey/blue is the exact/relaxed proportion of occurrence of each of the 68 ($K = 2$)-soft local maxima found using the halving procedure with the empirical probabilities (ranked). Soft local maxima found using the Tree HMM are denoted with a yellow annulus, and those found using the K-pairwise maximum entropy model are denoted with a pink star. The number of ($K = 2$)-soft local maxima captured by each model with a proportion of occurrence greater than the chosen reliability threshold of $\Theta = 0.5$ is shown in the table inset.

In the relaxed version, for each unique empirical ($K = 2$)-soft local maxima found, we performed the following: For each of the n iterations, we checked if the given ($K = 2$)-soft local maximum - which we will call $\vec{\gamma}_1$ - was found when using one data half but not the other. If so, then we performed neuron pair relaxation on $\vec{\gamma}_1$, importantly using the p.m.f. estimated from the data in the second half. If this resulted in a ($K = 2$)-soft local maximum that had been found originally in the second half - we will denote this $\vec{\gamma}_2$ - then we say that $\vec{\gamma}_1$ and $\vec{\gamma}_2$ are equivalent. Intuitively, we interpret $\vec{\gamma}_1$ and $\vec{\gamma}_2$ as being different only by sampling noise that shifted the local maximum in the first half, $\vec{\gamma}_1$, to a new activity pattern, $\vec{\gamma}_2$, in the second half of the data. (In mathematical terms, we identify $\vec{\gamma}_1$ and $\vec{\gamma}_2$ as members of the same equivalence class). In this case, we updated the proportion of occurrence of $\vec{\gamma}_1$ by adding a term of $\frac{1}{2n}$. Otherwise, the proportion of occurrence of $\vec{\gamma}_1$ was unmodified. The updated proportions of occurrence for the relaxed empirical analysis are shown in Fig 4.13 (light blue trace).

In summary, the Tree HMM results matched the empirical results the best, capturing 8 of the reliable empirical ($K = 2$)-soft local maxima (Fig 4.13). In contrast, the K-Pairwise MaxEnt model captured only one ($K = 2$)-soft local maximum.

4.6.3 Numerical exploration of the probability landscape

Finding local maxima

In this work, we use the same definition of local maxima of the joint response probability landscape that was used previously in [116]. That is, we defined local probability maxima as the single-flip-stable ascent patterns. This definition is equivalent to what has been termed a "(δ=1, ρ=Hamming distance)-mode" in applied math [19]:

Definition 4.4. A point is a (δ, ρ)-*mode* if and only if its probability is higher than all points within distance δ under a distance metric ρ.

To find local maxima of the high-dimensional joint RGC response probability landscape, we used the same iterative algorithm as in [116]. We refer to this algorithm as "single spin flip ascent". Conceptually, this iterative algorithm is implemented as follows: Let s index the current iteration step. The algorithm is initialized at $s = 0$ by starting with a population response $\vec{\sigma} \in \{0, 1\}^N$ in the data, which we denote as $\vec{\sigma}^{(0)} = \vec{\sigma}$. Within each iteration $s > 0$, the algorithm then 'flips' the response of neuron i, where i is chosen in accordance with a random permutation of the N neuron indices. (A different random permutation is independently generated for each iteration). That is, we set $\sigma_i^{(s)} = \sigma_i^{(s-1)} \oplus 1$, where $\oplus$ denotes mod 2 addition. The flip is retained if the new resulting configuration $\vec{\sigma}^{(s)}$ has lower energy (or equivalently, higher probability) than the previous iteration's pattern $\vec{\sigma}^{(s-1)}$, i.e. if $-\log P\big(\vec{\sigma}^{(s)}\big) < -\log P\big(\vec{\sigma}^{(s-1)}\big)$. In this case, the algorithm continues to the next iteration step $s + 1$. Otherwise, if flipping does not increase the probability, then the flip is not accepted, and the algorithm proceeds to try each of the other neuron indices according to the given

permutation. When none of the N neurons can be flipped, the resulting pattern is recorded as a local maximum. Implementation of the single spin flip ascent algorithm was done in Matlab.

Finding soft local maxima

To find soft local maxima of the modeled probability landscape for a given spike count K (see section 3.3), we used an iterative algorithm that we refer to as "opposite sign neuron pair relaxation". Conceptually, this algorithm is implemented as follows: Let s denote the index of the current iteration step. The algorithm is initialized at $s = 0$ by starting with a population response $\vec{\sigma}(t) \in \{0, 1\}^N$ having K spikes that appears in the data, denoted $\vec{\sigma}^{(0)} = \vec{\sigma}(t)$. Within each iteration $s > 0$, the algorithm then randomly selects a pair of neurons (i, j) that have opposite instantaneous responses according to the population response vector $\vec{\sigma}^{(s-1)}$, meaning that $\sigma_i^{(s-1)} = \sigma_j^{(s-1)} \oplus 1$, where $\oplus$ denotes mod 2 addition. The algorithm then proceeds by 'flipping' the response state of both neurons, by which we mean that we set $\sigma_i^{(s)} = \sigma_i^{(s-1)} \oplus 1$ and set $\sigma_j^{(s)} = \sigma_j^{(s-1)} \oplus 1$. In other words, the active neuron of the pair is made silent, and vice versa. This flipping alteration is retained if the resulting pattern, $\vec{\sigma}^{(s)}$, has a higher probability, i.e. if $P\left(\vec{\sigma}^{(s)}\right) > P\left(\vec{\sigma}^{(s-1)}\right)$. In this case, the algorithm continues to the next iteration step $s + 1$. On the other hand, if flipping does not increase the probability, then the flipping alteration is not accepted, and the algorithm proceeds to try each of the other neuron pairs with opposite instantaneous responses according to a random permutation. If none of the $K \cdot (N - K)$ neuron pairs with opposite instantaneous responses can be flipped to increases the probability, then the algorithm terminates at iteration s. By Definition 4.1, the terminating pattern $\vec{\sigma}^{(s-1)}$ is a K-soft local maximum.

4.6.4 Parametric repeat analysis

We here detail the design and implementation of the parametric repeat analysis reported in this chapter, which aimed to investigate the relation between the repeat structure of the stimulus ensemble and prevalence of local probability maxima. As described in section 2, each of the two original visual stimulus ensembles used for this analysis was a movie (Movie #3 or Movie #4) that was designed to alternate between a presentation of a unique (i.e. non-repeated) movie segment, and a fixed "target" movie segment. I.e. the unique movie segments and the repeated presentations of the target movie segment were interleaved. Derived from the original, full-length movie, we then generated a range of distinct shorter-duration movie stimuli that we refer to as "subset movies". Before detailing how these subset movies were constructed, we first introduce some notation: Let A denote a unique movie segment, and let B denote a repeated movie segment. We then define $n_A =$ the number of presentations of a unique movie segment, and define $n_B =$ the number of repeated presentations of the fixed target movie segment. We define $n_{\text{total}} = n_A + n_B$, and define the *repeat ratio*, ρ, of the subset movie as $\rho = n_B/n_{\text{total}}$. Note that $0 \leq \rho \leq 1$.

For the parametric repeat analysis for Movie #3, we always set $n_{\text{total}} = 70$ total movie segments comprising each subset movie, corresponding with a duration of 2100 s (or 105,000 time bins). For each repeat ratio $\rho = n_B/n_{\text{total}}$ examined, we generated an associated subset movie by selecting n_B of the repeated movie segments in the original Movie #3. (For example, for a repeat ratio of $\rho = 0.1$, we chose 7 repeat movie segments and 63 unique movie segments to include in the associated subset movie). The specific choice of which n_B of the 68 repeat target segments to include, and which n_A of the 70 unique movie segments to include in the subset movie was made in accordance with one of five random sequence permutations. For the parametric repeat analysis for Movie #4, we always set $n_{\text{total}} = 72$ total movie segments comprising each subset movie, corresponding with a duration of 4320 s (or 216,000 time bins).

Let n_ρ denote the number of distinct repeat ratios examined, and let n_k denote the number of random sequence permutations. Then the total number of subset movies was $n_\rho \cdot n_k$. For each subset movie, we fit the Tree HMM to the population response data restricted to that subset movie. That is, from the original population spiking data, we selected the sequence of population response patterns that were elicited only by the sequence of movie segments included in the given subset movie. This restricted set of population spiking data corresponded to an $N \times 105000$ binary matrix for the analysis using Movie #3 (or $N \times 216000$ matrix for the analysis using Movie #4). The local maxima results reported in Fig 4.3 were obtained by then performing the single spin flip ascent algorithm on the joint response probability landscape corresponding with each subset movie (i.e. repeat ratio), as modeled by the Tree HMM.

4.6.5 Scaled count distribution

To characterize the effect of spike count on the empirical joint response probability landscape (Fig **??**), we defined the "scaled count distribution", denoted $\widetilde{P}(K)$, as:

$$\widetilde{P}\,(K) \equiv \frac{P_{\text{empirical}}(K)}{\binom{N}{K}} \tag{4.5}$$

where

$$P_{\text{empirical}}(K) \equiv \frac{\sum_{t=1}^{T} \delta_{K,w_H(\vec{\sigma}(t))}}{T} \tag{4.6}$$

where $w_H\big(\vec{\sigma}(t)\big)$ denotes the spike count of the population response observed in time bin t, T denotes the total number of time bins in the data, and $\delta_{a,b}$ is the Kronecker delta function.

4.6.6 Assessing robustness of the set of mapped soft local maxima

To investigate how robust the mapping from the dataset of all observed responses with K spikes to the set of unique K-soft local maxima was, we performed 100 independent implementations of our opposite sign neuron pair relaxation algorithm for the Movie #2 dataset. For each implementation, a distinct random sequence of permutations was used for the choice of neuron pairs when performing each iteration of the algorithm. For each value of K, we then computed the mean pairwise overlap ratio between the identified set of unique K-soft local maxima for each implementation pair (l, m) (there was a total of $\binom{100}{2} = 4950$ implementation pairs), defined as:

$$\text{Mean Pairwise Overlap Ratio} \equiv \frac{\frac{1}{\binom{100}{2}} \sum_{l<m} \left|\mathcal{U}_l \cap \mathcal{U}_m\right|}{\max_{l=1}^{100} \left(\left|\mathcal{U}_l\right|\right)} \tag{4.7}$$

where l and m index one of the 100 mapping implementations performed, $\mathcal{U}_l$ denotes the set of unique K-soft local maxima obtained for the l-th mapping implementation, and $|\cdot|$ denotes set cardinality. Note that the the mean pairwise overlap ratio values are reported as percentages in Fig 4.5A.

4.6.7 Estimating barrier-depths of soft local maxima

To quantify how pronounced each identified soft local maximum $\vec{\gamma}$ was in terms of its associated peak-to-valley ratio in the probability landscape, we computed a proxy measure that we denote $r_{\vec{\gamma}}$. Formally, this is defined as:

$$r_{\vec{\gamma}} \equiv \arg\max_{\vec{\sigma}_s} \left[\frac{P\left(\vec{\gamma}\right)}{P\left(\vec{\sigma}_s\right)}\right] \tag{4.8}$$

where the argmax is taken over all populations responses $\vec{\sigma}_s$ observed in the data that are mapped (via opposite sign neuron pair relaxation) to $\vec{\gamma}$. Note that because our

search is only over population responses $\vec{\sigma}_s$ observed in the real data, $r_{\vec{\gamma}}$ is a *lower bound* on the true peak-to-valley ratio associated with soft local maximum $\vec{\gamma}$.

To investigate the relationship between $r_{\vec{\gamma}}$ and spike count level, we performed the following analysis: For each spike count level K (in practice we examined $2 \leq K \leq 5$, because the set of identified K-soft local maxima was perfectly robust for this range), we computed the mean $r_{\vec{\gamma}}$, averaged over all K-soft local maxima. We also computed the interquartile range of the $r_{\vec{\gamma}}$ values for each spike count level K (see Fig 4.5E).

4.6.8 Visualizing ridges

To visualize specific examples of how soft local maxima were organized across low to high spike count (K) levels, we used a type of breadth-first search (BFS) algorithm [113]. We call our variant of this algorithm the *ridge search algorithm*. The input to the ridge search algorithm is a given "root" soft local maximum, $\vec{\gamma}_{\text{root}}$, and the output is a rooted digraph that is specific to the input. The nodes of this output rooted digraph represent soft local maxima, and a directed edge $(\vec{\gamma}_K, \vec{\gamma}_{K+1})$ is present if and only if soft local maximum $\vec{\gamma}_{K+1}$ is u-reachable from $\vec{\gamma}_K$, as defined in Definition 4.2. Given a fixed spike count level $K_{\min}$, we chose each input to be one of the $K_{\min}$-soft local maxima found by performing opposite sign neuron pair relaxation on the probability landscape obtained by fitting the Tree HMM to the data.

Note that the ridge search algorithm is non-deterministic, since one of its component subroutines is the non-deterministic opposite sign neuron pair relaxation algorithm. To incorporate our confidence level about the digraph edges computed by the ridge search algorithm, we thus did the following: For each neuron i in the silent set of the current K-soft local maximum, $\mathcal{S}(\vec{\gamma}) \equiv \{\text{neurons } j \,|\gamma_j = 0\}$, we performed 10 iterations of opposite sign neuron pair relaxation on the population response pattern in the $(K+1)$-th spike count level that was obtained when neuron i's response was changed to spiking. For each iteration, we used a different, independent choice of

the random permutation. We then set the weight of each directed edge $\left(\vec{\gamma}, \vec{\gamma}_{K+1}\right)$, which we denote by $w(\vec{\gamma}, \vec{\gamma}_{K+1})$, as the proportion of the 10 relaxation iterations that resulted in $\vec{\gamma}_{K+1}$. To mitigate effects due to noise, for each unique $(K+1)$-soft local maximum $\vec{\gamma}_{(K+1)}$ reached after performing the above procedure on a given $\vec{\gamma}$, we further only included the directed edge $\left(\vec{\gamma}, \vec{\gamma}_{K+1}\right)$ if $w(\vec{\gamma}, \vec{\gamma}_{K+1})$ exceeded a reliability threshold, Θ. In practice, we used $\Theta = 0.3$. The computed weight of each edge is represented by edge thickness in the digraph visualizations (Figs 4.8 and 4.9).

To aid with visualizing distinct ridges, we also assigned an x- and y-coordinate to each node $\vec{\gamma}_K$ (i.e. soft local maximum) in the output rooted digraphs as follows:

1. The y-coordinate, denoted $y(\vec{\gamma}_K)$, was set to K (i.e. the spike-count of $\vec{\gamma}_K$).

2. Our choice of formulation for the x-coordinate was based on two motivating criteria: First, we wanted to be able to visualize potential "jumps" in ridge organization. These jumps can arise due to the current identified ridge either terminating at a lower spike count level than the one we chose arbitrarily to explore out to, or branching into multiple distinct ridges. Examples of this scenario can be seen in Figs 4.8C and 4.8B. Second, we wanted to visualize only "genuine" jumps or branching, with respect to the notion of neuronal communities. Specifically, after inspection of preliminary digraph visualizations, we noticed that 'connected' (i.e. u-reachable) soft local maxima typically had a set of active neurons that was a subset of a larger unique group (which we call the *neuronal community*). We thus post-hoc formulated our definition of the x-coordinate for nodes in our digraph visualizations to aid us in examining how prevalent this property was.

   To do this, we needed to define an overlap measure between the activity sets of different soft local maxima. The natural distance metric based on overlap is:

$$d_O\left(\vec{\gamma}_K, \vec{\gamma}_G\right) \equiv \left|\mathcal{A}\left(\vec{\gamma}_K\right) \cap \mathcal{A}\left(\vec{\gamma}_G\right)\right| - K \tag{4.9}$$

where $G > K$ denotes a higher spike-count level, $\mathcal{A}(\vec{\gamma}) := \{i|\gamma_i = 1\}$ denotes the "activity set" (i.e. the set of active neurons) of soft local maximum $\vec{\gamma}$, and $|\cdot|$ denotes set cardinality. Note that if the activity set of a K-soft local maximum is a subset of the activity set of a soft local maximum at a higher spike count level, $G > K$, then this distance is 0. I.e. if $\mathcal{A}(\vec{\gamma}_K) \subset \mathcal{A}(\vec{\gamma}_G)$, then $|\mathcal{A}(\vec{\gamma}_K) \cap \mathcal{A}(\vec{\gamma}_G)| = K$, and thus $d_O(\vec{\gamma}_K, \vec{\gamma}_G) = 0$.

The input to our procedure is the digraph information - that is, the set of nodes (soft local maxima) and the set of edges connecting the nodes; the output is the assigned x-coordinate of each node. In brief, our procedure is initialized by setting the x-coordinate of the soft local maximum at the highest spike count level, which we denote $\vec{\gamma}_{K_{\max}}$, to 0. That is, in our notation, we set $x(\vec{\gamma}_{K_{\max}}) = 0$. Our procedure then works backward to compute the x-coordinate of each node at a successively lower spike count level. Specifically, for each node $\vec{\gamma}_K$ where $K_{\min} < K < K_{\max}$, we compute its *path neighborhood*, $\mathcal{N}_p(\vec{\gamma}_K)$, which is the set of all soft local maxima $\vec{\gamma}_G$ residing in a higher spike count level $G > K$ such that there is a path in the digraph connecting $\vec{\gamma}_K$ and $\vec{\gamma}_G$. If the path neighborhood is empty, i.e. $\mathcal{N}_p(\vec{\gamma}_K) = \emptyset$, then this means that $\vec{\gamma}_K$ is situated at the end of a distinct branch. In this case, we assign $x(\vec{\gamma}_K) = x(\vec{\gamma}_{K_{\max}}) + b$, where b is an offset. Otherwise, if $\mathcal{N}_p(\vec{\gamma}_K) \neq \emptyset$, then we found the soft local maximum in the path neighborhood that was closest to $\vec{\gamma}_K$ in terms of overlap, which we denote $\vec{\gamma}^*_G$. Finally, we assigned $\vec{\gamma}_K$ the same x-coordinate as this nearest $\vec{\gamma}^*_G$, plus an offset equal to the overlap distance between $\vec{\gamma}_K$ and $\vec{\gamma}^*_G$:

$$x(\vec{\gamma}_K) = x(\vec{\gamma}^*_G) + d_O(\vec{\gamma}_K, \vec{\gamma}^*_G) \tag{4.10}$$

$$\text{where} \quad \vec{\gamma}^*_G \equiv \underset{\vec{\gamma}_G \in \mathcal{N}_p(\vec{\gamma}_K)}{\arg\min} \; d_O(\vec{\gamma}_K, \vec{\gamma}_G) \tag{4.11}$$

In the case of a tie (e.g. if a ($K = 4$)-soft local maximum $\vec{\gamma}_4$ had complete overlap of its activity set with both $\vec{\gamma}_5$ and $\vec{\gamma}_6$), we chose the soft local maximum at the nearest spike count level to be $\vec{\gamma}_G^*$. (I.e. in the above example, we would choose $\vec{\gamma}_G^* = \vec{\gamma}_5$, and would assign $x(\vec{\gamma}_4) = x(\vec{\gamma}_5)$).

In addition, we computed the MAP (maximum *a posteriori*) estimate α of the Tree HMM latent state for the soft local maximum $\vec{\gamma}$ associated with each digraph node:

$$\begin{aligned} \hat{\alpha}_{\text{MAP}}(\vec{\gamma}) &\equiv \arg\max_{\alpha} P(\alpha|\vec{\gamma}) \\ &= \arg\max_{\alpha} \left[\frac{P(\vec{\gamma}|\alpha)P(\alpha)}{\sum_{\beta} P(\vec{\gamma}|\beta)P(\beta)} \right] \\ &= \arg\max_{\alpha} \left[\psi_{\alpha} g_{\alpha}(\vec{\gamma}) \right] \end{aligned} \tag{4.12}$$

where ψ denotes the stationary distribution of the Markov chain, and $g_{\alpha}(\cdot)$ is the emission distribution for mode α. Visualization of the rooted digraphs computed via the ridge search algorithm (examples of which are shown in Figs 4.8 and 4.9) was automated using the *igraph* and *network* packages in R.

We also investigated the organization of soft local maxima across high to low spike count (K) levels; example results are shown in later subsections. For this analysis, a rooted digraph was constructed, but where each root node was taken to be a soft local maximum found at a high K level, and where a directed edge $(\vec{\gamma}_K, \vec{\gamma}_{K-1})$ was added if and only if soft local maximum $\vec{\gamma}_{K-1}$ is d-reachable from $\vec{\gamma}_K$ (see Definition 4.3). Note that we report the u-reachable version (i.e. starting from a low spike count and progressing out to higher spike count levels) in the main text, as we have more reliable sampling of the data at low spike count levels.

4.6.9 Constructing the ridge union graph

In network science, *community structure* refers to the occurrence of groups of nodes in a network that are more densely connected internally than with the rest of the

network [40]. In parallel with this concept, we observed that we can map the geometric picture of ridges in the joint response probability landscape to one of communities in the population of ganglion cells: We start with the empty graph (i.e. no edges) in which each node represents a ganglion cell in the population. For each ridge, we then compute the union of the set of active neurons for each soft local maximum comprising that ridge. This union set, which we call the *neuronal community* associated with the ridge, is then represented as a clique in the undirected graph. That is, we add all-to-all connectivity in the graph between the neurons in the union set. When we have added in the clique corresponding with each ridge, we call the resulting network the *ridge union graph* (see Fig 4.10A).

The inverse process of identifying the neuronal communities given the ridge union graph can be readily implemented by simple community detection algorithms. Specifically, all neuronal communities can be identified in this case via the Bron-Kerbosch algorithm, which is a well-known algorithm for finding the maximal cliques in an undirected graph [15]. Construction of the ridge union graph and simple community detection (see Fig 4.10A) was done using the *igraph* package in R.

4.7 Appendix

4.7.1 Control analyses

We sought to ascertain that correlations in the data were necessary to give rise to the soft local maxima and discrete ridge structures observed here, as opposed to these features trivially arising from the underlying probability model. Due to the large population sizes of the datasets we analyzed, it is intractable to determine the *empirical* joint response p.m.f. It is thus not possible to directly investigate the presence of these geometric features in the empirical probability landscape. We thus compared the results obtained for the true dataset of 152 ganglion cells responding to

Movie #2 to those obtained by fitting the hidden Markov model to two manipulated cases in which only limited structure was preserved. Each control is detailed below.

Heterogeneous firing rates shuffled control

To generate this control, for each neuron i in the population we performed a "complete shuffle" on neuron i's discretized spike train (binned into T total 20 ms time bins) occurring in the data, by implementing a random permutation of the T time bins. Note that an independent random permutation was implemented for each neuron. This complete shuffling procedure eliminates all signal and noise correlations among neurons in the population, but retains the firing rate of each neuron over the course of the experimental recording. Thus, the control 'dataset' of joint responses generated via this procedure corresponds with a population of neurons that fire independently, but have heterogeneous firing rates that match the original data.

Homogeneous firing rates shuffled control

To generate this control, we first computed the average mean rate from the data, $\left\langle r_i \right\rangle_{i=1}^{N}$, averaged over the entire population of N ganglion cells, where $r_i \equiv \left(\sum_{t=1}^{T} \sigma_i(t) \right) / T$. We then simulated an independent population of matched size N in which each neuron i was assigned to have its firing rate $r_i = \rho \equiv \left\langle r_i \right\rangle_{i=1}^{N}$. Thus, the control 'dataset' of joint responses generated via this procedure corresponds to a population of neurons that fire independently, and moreover have homogeneous firing rates.

Results for the control analyses

As seen in Fig 4.14A, there was a categorical difference between the cross-validated log-likelihood (CV-LL) curves obtained when the HMM was fit to the original data, versus when the HMM was fit to the simulated control "data". Whereas the CV-LL

curve for the original data exhibited a well-defined peak at 19 collective modes, when fit to each control dataset, the CV-LL curve instead achieved its maximum at only one collective mode.

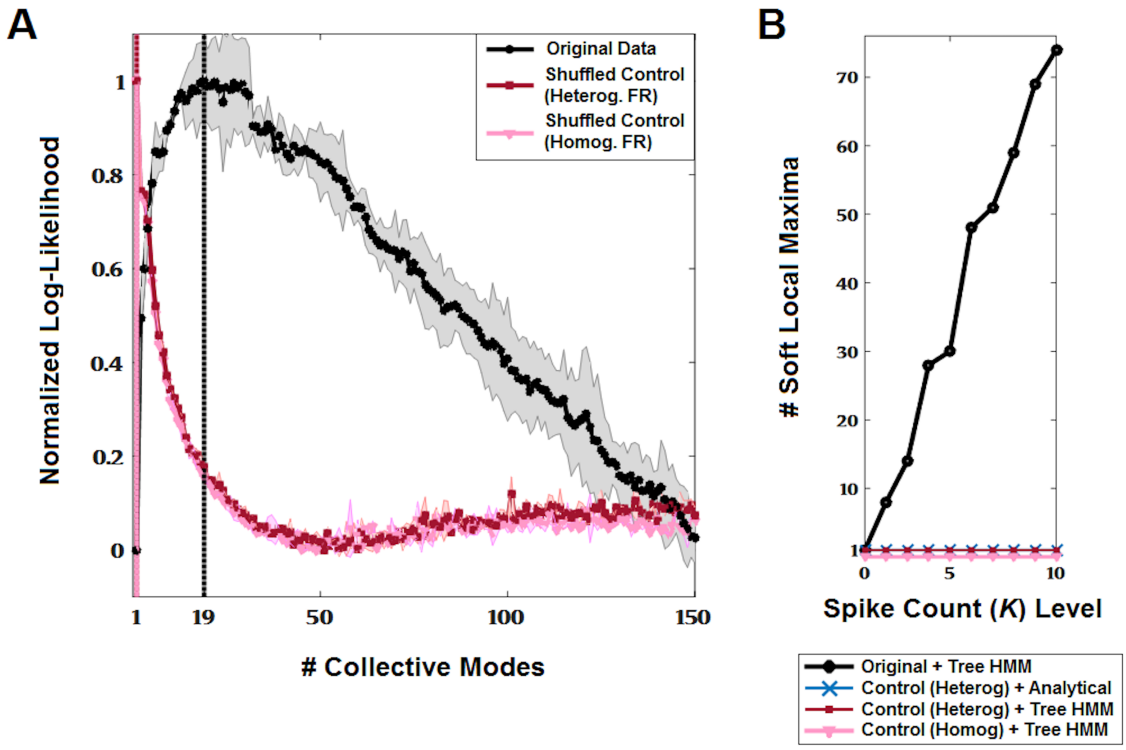

Figure 4.14: **Control analysis results.**
(A) Normalized cross-validated log-likelihood (CV-LL) as a function of the number of HMM collective modes for: the original dataset of 152 ganglion cell responses to Movie #2 (black), the heterogeneous firing rate shuffled control (red), and the homogeneous firing rate shuffled control (pink). Shown in bold is the mean normalized CV-LL over all n cross-validation folds (in practice taken to be $n = 2$). Shaded error bars denote SEM over cross-validation folds. Each colored dashed line indicates the optimal latent dimensionality for the corresponding dataset (see Key for colors). (B) Shown is the number of unique K-soft local maxima identified via opposite sign neuron pair relaxation, as a function of spike count K. Results are shown for four cases: the original dataset with the Tree HMM chosen as the underlying probability model (black circles); the heterogeneous firing rate shuffled control with the analytical independent model as the underlying probability model (blue crosses); the heterogeneous firing rate shuffled control with the fit Tree HMM as the underlying probability model (red squares); and the homogeneous firing rate shuffled control with the Tree HMM as the underlying probability model (pink triangles).

The soft local maxima results were likewise substantially different for the original data versus the controls (Fig 4.14B). In particular, there was a monotonic increase in

the proliferation of soft local maxima at all spike count levels for the original data, with 74 unique soft local maxima identified at spike count level $K = 10$. In contrast, the modeled probability landscapes for both controls exhibited either zero or one unique soft local maximum across all spike count levels.

Specifically, for the first control, in which the ganglion cell firing rates in the original data were preserved and thus heterogeneous across the population, one unique soft local maximum was present at each spike count level. As expected analytically for an independent neuron population with heterogeneous firing rates, we confirmed that the single soft local maximum identified at each spike count level K corresponded with the K ganglion cells that had the highest firing rates. This scenario corresponds with a ganglion cell population that is organized into one, single neuronal community. In this case, the ridge union graph corresponds to the complete graph (i.e. it has all-to-all connectivity between all nodes). Thus, in the case of a population of heterogeneous neurons that fire independently, there is no community structure.

For the second control, in which each neuron was moreover assigned the same firing rate, no soft local maxima were found for any of the tested spike count levels. This is consistent with the analytical result and expectation for an independent population of homogeneous neurons, as the symmetry in the homogeneous case implies that the probability landscape depends only on K (and thus all joint responses within the same spike count level have identical probability).

In summary, results from the control analyses support that the soft local maxima results - and by extension the ridge and neuronal community results - are non-trivially dependent on the empirical correlation structure of the measured neural activity.

4.7.2 Pseudocode for the ridge search algorithm

Here we provide pseudocode for our *ridge search algorithm*, which is based on the classic breadth-first search algorithm. We use a non-recursive implementation that

utilizes a queue, denoted Q in the pseudocode below, which is a standard type of list. The basic operations on a queue are *enqueue*, which inserts an element at the end of the list, and *dequeue*, which returns and deletes the element at the front. In our implementation, we cache the information about each node (soft local maximum) in an object v, which has the following attributes:

- v.nodeID : the index of the node in the digraph
- v.pattern : the soft local maximum $\vec{\gamma} \in \{0,1\}^N$ that this node represents
- $v.K$: the spike count of v.pattern
- v.parent : the parent node of v (i.e. if there is a directed edge (p, v) in the digraph, then we say that node p is the parent of v)
- v.neighborhood : the set of all $(v.K+1)$-soft local maxima that are u-reachable from v.pattern
- v.edgeweights : the weight of each edge $(v.\text{pattern}, \vec{\gamma}_{K+1})$, where $\vec{\gamma}_{K+1} \in v.\text{neighborhood}$
- v.collectivemode : the MAP estimate of the collective mode that is associated with v.pattern

Each object v also has the following methods (i.e. the functions that can access and modify its data):

- v.FindNeighborhood : computes the *neighborhood* of v.pattern, which is the set of all unique soft local maxima in the $(v.K+1)$-th spike count level that are u-reachable from v.pattern; also computes the associated edgeweights
- v.MAPestimation : computes the maximum *a posteriori* (MAP) estimate of the Tree HMM latent state (i.e. *collective mode*) associated with v.pattern

Input: Starting "root" K-soft local maximum, $\vec{\gamma}_{\text{root}} \in \{0,1\}^N$; $K_{\max}$; threshold Θ

Output: Adjacency matrix for the rooted digraph that is specific to the input $\vec{\gamma}_{\text{root}}$

Algorithm 3 Ridge search algorithm

Initialize empty $\mathcal{D}$: list of objects, v, for each node in the digraph
$v \leftarrow$ InitializeObject($\vec{\gamma}_{\text{root}}$)
Add v to $\mathcal{D}$
Initialize $Q \leftarrow \emptyset$
Q.enqueue(v)
while $Q \neq \emptyset$ **do**
    $v \Leftarrow Q$.dequeue()
    v.FindNeighborhood(Tree HMM parameters, Θ)
    v.MAPestimation(Tree HMM parameters)
    Update $\mathcal{D}\{v.\text{nodeID}\}$
    for each soft local maximum $\vec{\gamma}_{K+1} \in v$.neighborhood **do**
        $n \Leftarrow$ InitializeObject($\vec{\gamma}_{K+1}$)
        n.parent $\Leftarrow v$.nodeID
        if (n has not yet been observed) AND ($n.K < K_{\max}$) **then**
            Add n to $\mathcal{D}$
            Q.enqueue(n)
$A \leftarrow$ ComputeAdjacencyMatrix($\mathcal{D}$)

4.7.3 Supplementary figures

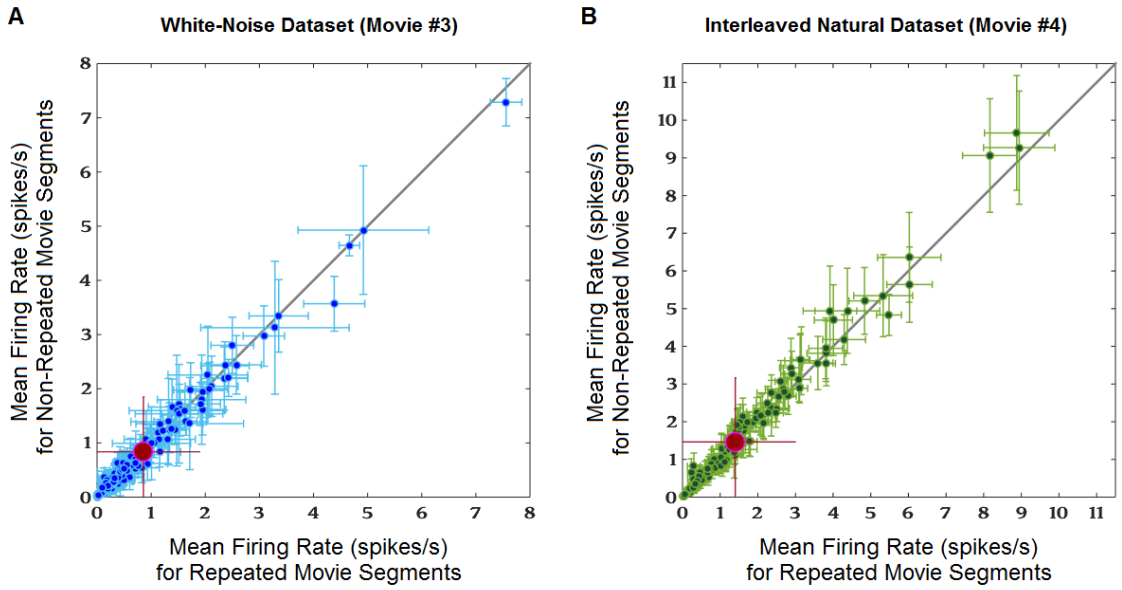

Figure 4.15: **Scatter plot of mean firing rates for the raw data.**
(A) Results for the Movie #3 dataset ($N = 155$). Each blue dot represents the raw firing rate (spikes/s) for one ganglion cell. Shown is each neuron's mean firing rate during the *non-repeated* movie segments, averaged over all unique movie segments comprising the original movie (y-axis), versus its mean firing rate during the *repeated* movie segments (x-axis). Error bars denote one standard deviation over movie segments. (B) Results for the Movie #4 dataset.

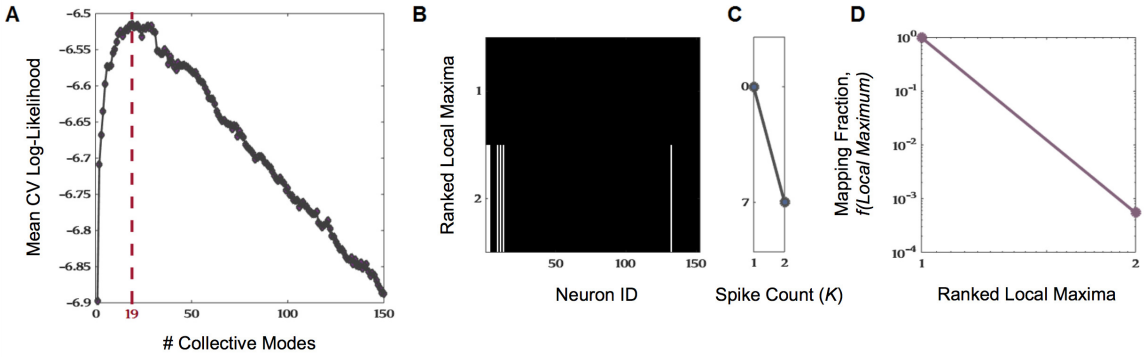

Figure 4.16: **Local maxima results for the Movie #2 dataset ($N = 152$).**
(A) Cross-validated log-likelihood (CV-LL) averaged over the two cross-validation folds (y-axis), as a function of the number of Tree HMM latent states (collective modes) (x-axis). Red dashed line denotes the optimal latent dimensionality, which corresponds to the peak of the CV-LL curve. (B) Each row represents the binary representation of an identified unique soft local maximum. (C) The spike count of each associated soft local maximum shown in panel (B). (D) Plot of the proportion of the 90001 population responses that were mapped via the single spin flip ascent algorithm to the corresponding soft local maximum (x-axis).

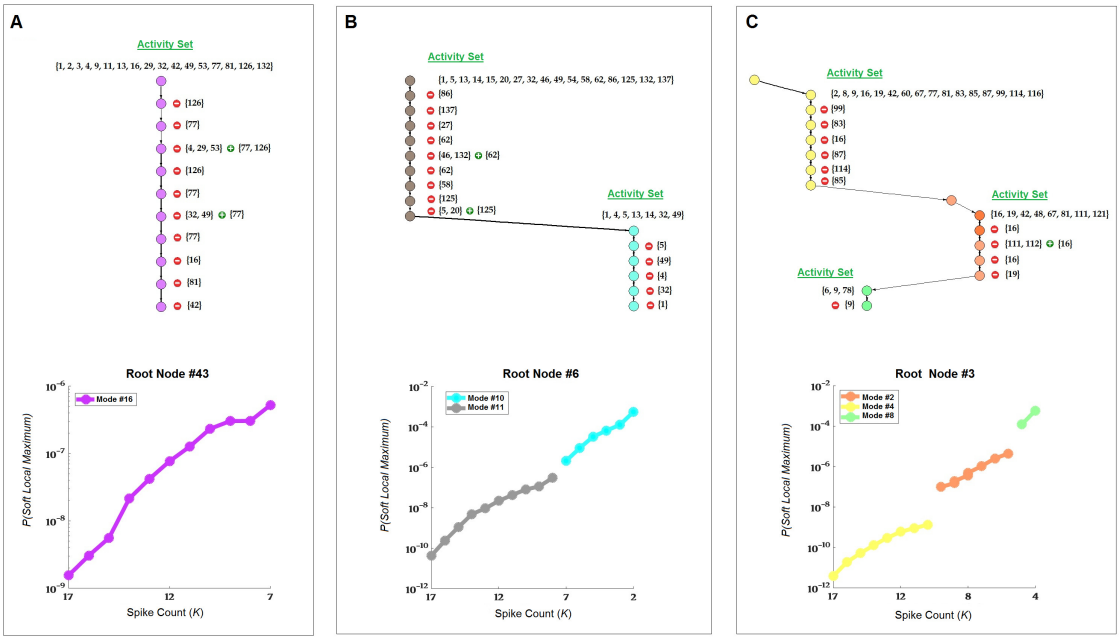

Figure 4.17: **Examples of output rooted digraphs using d-reachable edges.** (A) Output rooted digraph obtained via our ridge search algorithm when we took the unique ($K = 17$)-soft local maximum with rank 43 as the input root node. Notation is the same as in Figs 4.8 and 4.9. The rooted weighted digraph obtained when the ($K = 17$)-soft local maximum with rank 6 and 3 was taken as the input root node is shown in (B) and (C), respectively.

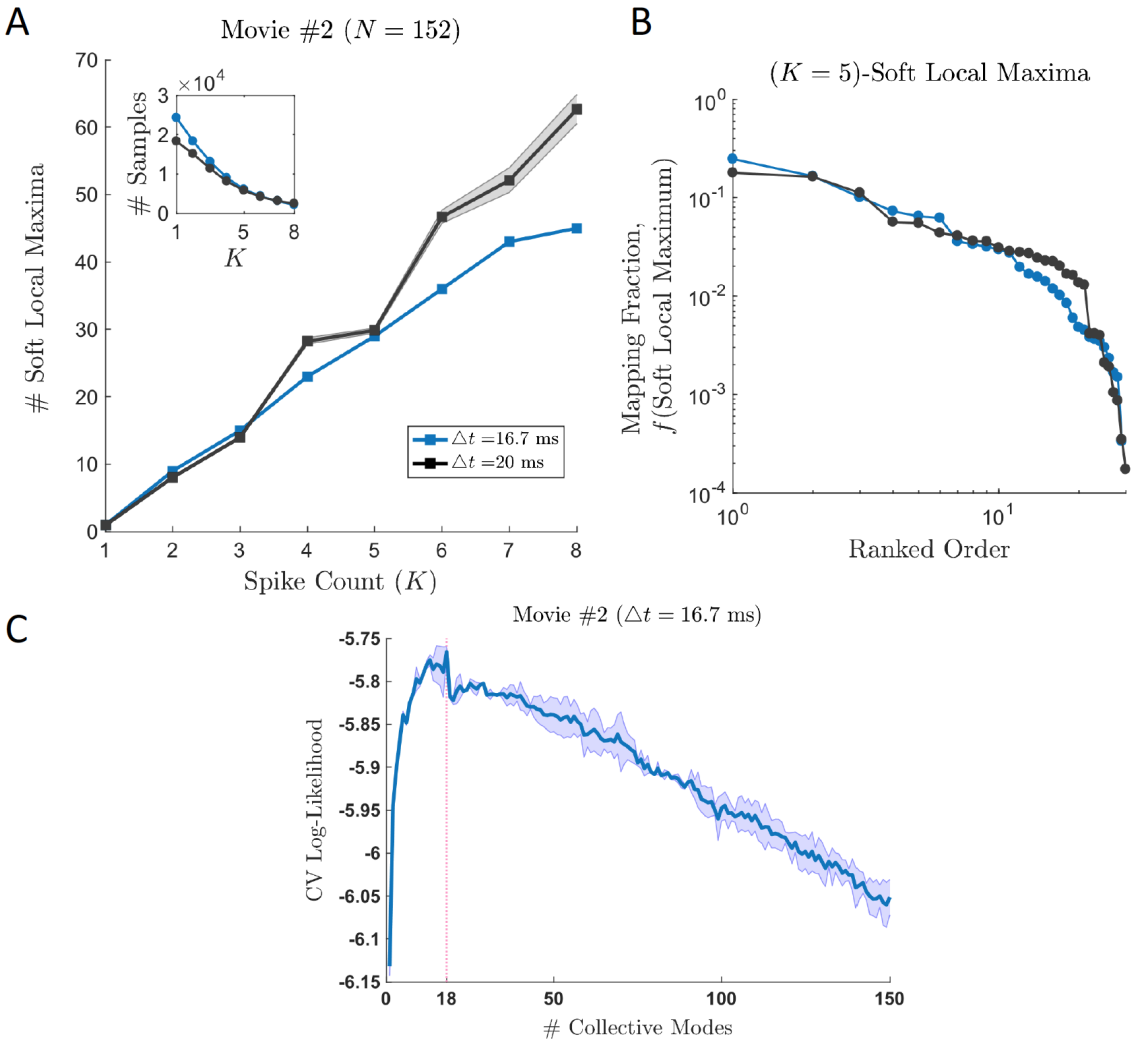

Figure 4.18: **Soft local maxima results for different time bin widths.** Comparison of soft local maxima results for the Movie #2 dataset when responses were discretized using a time bin width of $\triangle t = 20$ ms (grey) vs. $\triangle t = 16.7$ ms (blue; see color key). (A) The number of unique K-soft local maxima identified (y-axis) for each examined spike count level, K. *Inset*: Total number of population response patterns in the data for each spike count. (B) Log-log plot of the proportion of observed responses in the data with 5 spikes that were mapped to the corresponding ($K = 5$)-soft local maximum indicated on the x-axis. (C) Cross-validated log-likelihood (CV-LL) results when $\triangle t = 16.7$ ms. Error bars denote one standard deviation over the two cross-validation folds. Pink dashed line denotes the optimal latent dimensionality.

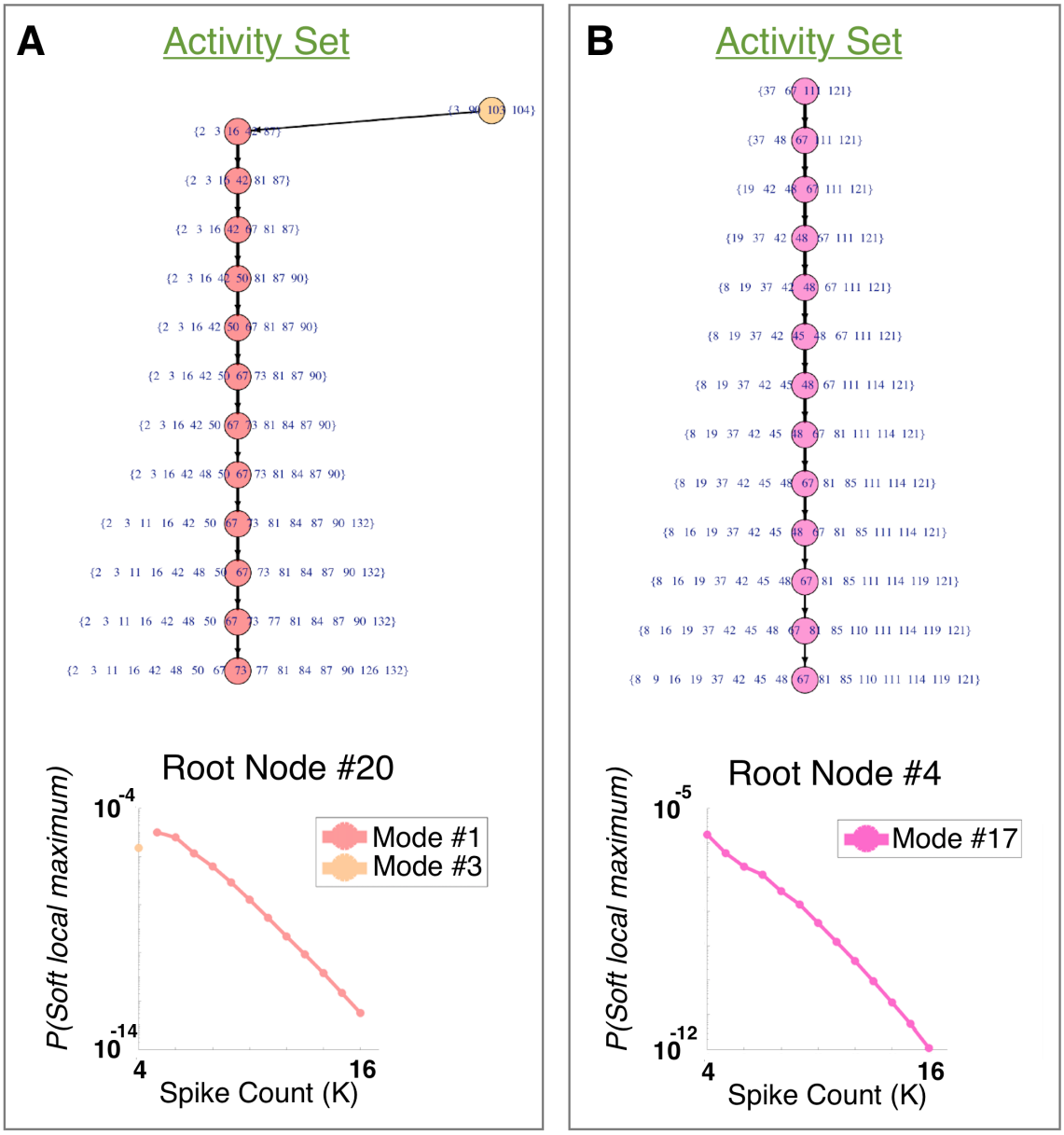

Figure 4.19: **Example Type 1 ridges for a different choice of time bin.** Examples of two rooted digraphs with Type 1 ridges obtained (via our ridge search algorithm) for the Movie #2 dataset when responses were discretized using a time bin width of $\triangle t = 16.7$ ms. Labeled above each node is the "activity set" of the corresponding soft local maximum $\vec{\gamma}$, $\mathcal{A}(\vec{\gamma}) \equiv \{\text{neurons } i | \gamma_i = 1\}$. Node color denotes the maximum *a posteriori* (MAP) estimate of the associated collective mode (see color key). Each digraph example shown in (A) and (B) corresponds with one distinct ridge. Bottom panels: Probability, as modeled by the Tree HMM, of each depicted soft local maximum versus its spike count, K.

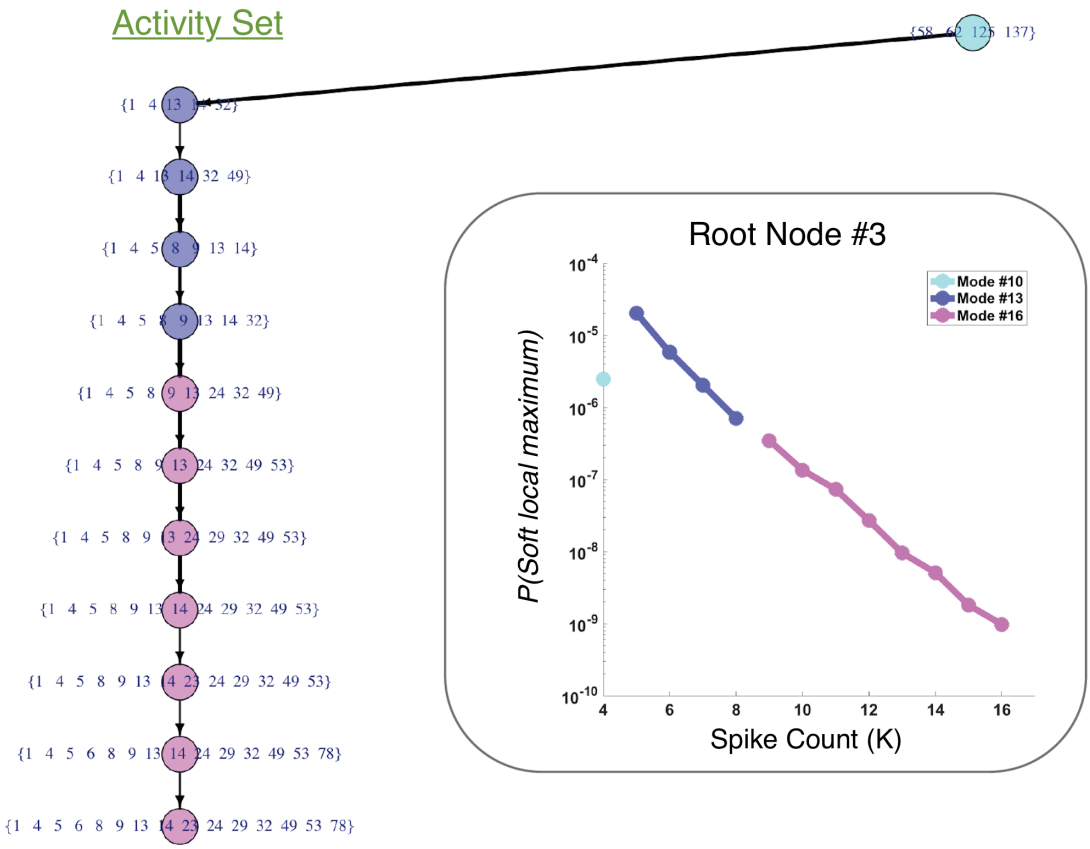

Figure 4.20: **Example Type 2 ridge for a different choice of time bin.** Example of a rooted digraph with a Type 2 ridge obtained for the Movie #2 dataset when the responses were discretized using a temporal bin width of $\triangle t = 16.7$ ms. Format is the same as in Fig 4.19. There was one distinct ridge for this digraph; however, the soft local maxima comprising this ridge were associated with two possible collective modes: mode #13 (lavender) and #16 (pink).

Chapter 5

Biologically-Plausible Algorithms for Learning Retinal Population Codewords

5.1 Introduction

The Baum-Welch algorithm that we used in [63, 87] to learn the Tree hidden Markov model parameters and to subsequently identify the RGC population codewords (as detailed in Chapters 3 and 4) is not a biologically-plausible algorithm. For example, it is restricted to offline learning. So, how could downstream processing areas, such as primary visual cortex (V1), extract the previously-identified error-robust codewords? This question, which is situated at Marr's "hardware implementation" level [67], is the focus of this chapter (the project for which is work in progress).

Ultimately, our goal is posed as follows: We would like to find a biologically-plausible spiking circuit model that is capable of learning the latent RGC population codewords that were previously identified by the Tree HMM. Note that, importantly, we want the circuit model to be able to successfully extract the latent codewords for

actual recording data. Conceptually, we are envisioning modeling the feedforward synapses from LGN - which we assume uses a largely conserved version of the RGC population code - to layer 4 of cortical area V1.

5.1.1 Primary biological constraints

In general, any such circuit model under consideration would need to satisfy three primary criteria to be considered biologically plausible [83]:

1). Biologically plausible learning algorithms must be formulated in the **online** (streaming), rather than offline (or batch) setting;

2). Learning rules must primarily be **local**, although there are known exceptions (e.g. heterosynaptic plasticity can allow for quasi-local and global scaling [21]);

3). Learning should ideally be **unsupervised** (though note that this is not a hard constraint)

Regarding criterion (1), this means that the input data is streamed to the algorithm sequentially, one sample $\vec{x}(t)$ at a time, and that the corresponding output of the circuit must be computed before the next input sample arrives. While a stochastic and online approximation algorithm for the EM algorithm was introduced in [18], this work was not in the context of biologically-plausible neural networks with synaptic plasticity rules.

Regarding criterion (2), there is an extensive literature on the documented ability of neurobiological networks to be modified by experience (see e.g. [12]), and extensive experimental findings to place constraints on biologically-plausible learning rules. The most extensively analyzed learning rules are those that involve changes of synaptic weights, which can be partitioned into two categories: associative and nonassociative. Associative synaptic plasticity rules depend on the interaction of two or more state

variables, e.g. the state of activity in the presynaptic *and* postsynaptic neurons. Possibly the best known associative learning rule is the Hebbian rule for synaptic modification [12], which requires a specific temporal relationship between the pre- and postsynaptic neuronal activities. The requirement to be purely local means that the synaptic weight update can depend only on the activity of the two neurons a synapse connects (as is the case, for example, in Hebbian learning). Activities of other neurons are not physically available to a synapse, and therefore including them into learning rules is generally biologically implausible. (Note that most modern artificial neural networks, such as backpropagation-based deep learning networks, rely on nonlocal learning rules).

The criterion of unsupervised learning in (3) is not a hard constraint; in contrast to the retina, layer 4 of V1 receives feedback projections from layer 6 that could potentially provide a supervision signal. However, as conceptually we are interested in first investigating if the population codewords could be learned during the feedforward pathway alone from LGN to layer 4 of V1, our first priority is to investigate an unsupervised learning approach. Thus, we will take an approach where we assume that the only information that the circuit model has access to is: (i) the current input population activity pattern (i.e. the current presynaptic activity); and (ii) the current synaptic weights, and hence the current postsynaptic activity (and any lateral network inhibitory signals).

5.1.2 Bayesian inference implemented by neural circuits

Numerous experimental evidence suggests that the brain applies principles of Bayesian inference for analyzing sensory stimuli [34, 91]. In the Bayesian inference framework, an observed sample or state is interpreted as the result of an underlying cause that cannot be observed directly. This framework is thus potentially well-suited for the

important task of probabilistic perception [39], in which latent causes that explain noisy and potentially ambiguous sensory inputs have to be inferred.

Correspondingly, within the theoretical neuroscience literature, many simulation-based studies have investigated Bayesian inference as a theoretical framework for guiding the search for biologically-grounded neuron or neural network models that can perform inference. For example, in [29], it was shown that a single neuron could be modeled as a hidden Markov model with two possible states. In [91], the possibility to carry out Bayesian probabilistic computations in recurrent networks of rate-based neurons was investigated, and this was then extended in [92] to networks of spiking neurons. Both of these approaches relied on belief propagation however, and did not consider any neurally plausible learning mechanism.

More recently, using a normative approach, it was demonstrated in [47] that a certain class of implicit generative models can be represented and learned online by a spiking neural network model in the context of unsupervised competitive Hebbian learning. Specifically, [47] used a winner-take-all (WTA) circuit model, which is a ubiquitous network motif (i.e. a subgraph [51]) found in cortical microcircuits [33]. It was shown that this WTA circuit model could be directly mapped to a Poisson mixture model (MM) that the network implicitly represents, and that the proposed STDP learning rule drives the synaptic weight configuration of the WTA network to converge to the maximum likelihood parameter estimates of the Poisson MM given the input population responses [47]. Thus, [47] essentially showed that their proposed STDP learning rule in a WTA circuit motif model can act as a biologically-plausible proxy to the online stochastic EM algorithm for fitting a Poisson MM to the input population responses.

It is important to emphasize that all of these past studies were simulation-based, and typically incorporated simplifying assumptions for analytical tractability that may likely be violated for real neural data (see the following section).

5.2 Approach and preliminary results

Given that the approach in [47] satisfies all three of our main desired criteria for biological plausibility, we first considered investigating the suitability of this WTA circuit model for extracting the RGC population codewords that we found in our previous work from actual data [63, 87].

5.2.1 Error correction captured by simpler models

One obvious potential problem is that the approach in [47] corresponds with a biologically-plausible algorithm for fitting an independent Poisson mixture model (with uniform mixing weights), and not the Tree HMM that we used in our prior work to identify the noise-robust RGC population codewords. Unlike the Tree HMM and Tree mixture model, the Poisson MM cannot capture noise correlations (because it assumes that neuronal responses are independent conditioned on the latent state). However, the Poisson MM can capture signal correlations. As shown in Fig 4.14 in the proceeding chapter, we found that the collective modes and ridges are non-trivially dependent on the empirical correlation structure of the measured retinal ganglion cell activity. As this control analysis had abolished *both* signal and noise correlations, we cannot conclude however whether only incorporating signal correlations would be mostly sufficient for capturing the RGC population codewords, or whether the noise correlations also play a fundamental role. This is an empirical question that depends on the nature of the population code used by the retina.

Thus, as a first step, we investigated if the Poisson MM fit via the standard batch EM algorithm - which places an upper performance bound on the biologically-plausible approximation method used in [47] - can capture latent modes that also exhibit error correction. To assess this, as in Chapter 3, we fit a Poisson MM via the standard batch EM algorithm to the set of RGC population responses (allowing for

each population response pattern $\vec{x}(t) \in \mathbb{Z}^N_{\geq 0}$) evoked by the non-repeated portion of the interleaved natural movie (Movie #4; see Chapter 2 for stimulus details). Then, using the learned model parameters, we inferred the latent mode that was active at each time bin t of the fixed 60-s target movie clip, for all 73 repeats of that target movie clip. Inference was implemented via standard maximum *a posteriori* (MAP) estimation. As shown in Fig 5.1A, we were able to find time bins (shaded in yellow) for which the many distinct population response patterns elicited by the same stimulus (see bottom panel) could be robustly mapped to the same latent mode identity. For example, for the three shaded time bins shown in Fig 5.1A, $p(\alpha|t) > 0.9$.

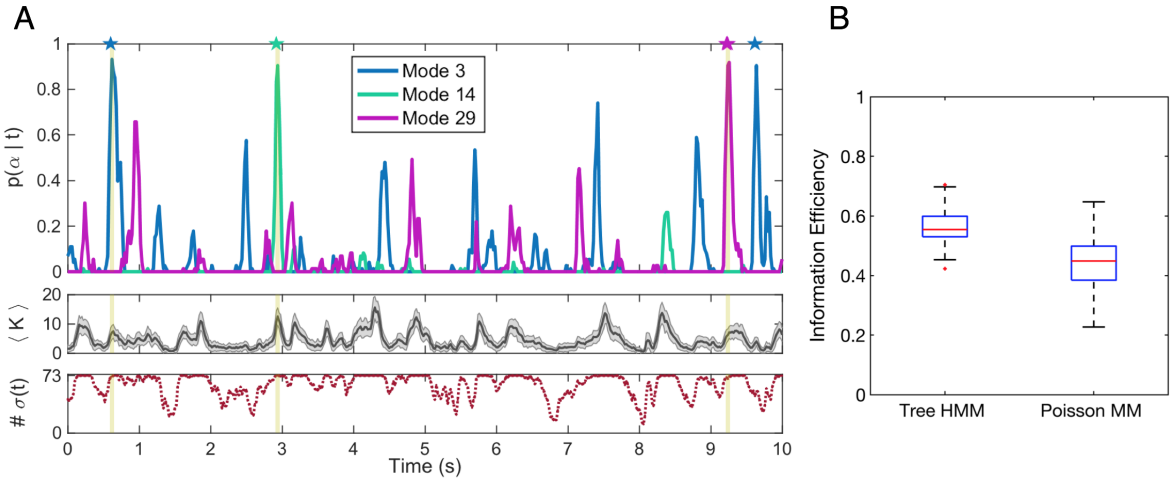

Figure 5.1: **Error correction performance of latent modes of the Poisson MM.** Error correction performance for the latent modes ($m = 35$) identified by the Poisson mixture model (MM) fit to the interleaved natural movie dataset ($N = 170$ RGCs) via the batch EM algorithm. (**A**) *Top*: Shown for three collective modes (see key) is the probability (across repeats of the target movie segment) that the population activity was mapped to that particular collective mode at any given time t. Time bins for which this value, denoted $p(\alpha|t)$, reached > 0.9 are indicated by a star. *Center*: Mean population spike count averaged over the repeats; error bars denote one standard deviation. *Bottom*: The number of distinct population responses observed over all 73 repeats at each time t. (**B**) Comparison of the information efficiency of latent mode activation across repeats of the same stimulus for the Tree HMM ($m = 50$ latent modes) vs. Poisson MM ($m = 35$ latent modes). (Information efficiency = mutual information about stimulus / mode output entropy; see main text). Box plots depict the interquartile range of this quantity over all latent modes (red line = median).

To compare the global degree of fault tolerance enabled by mapping noisy RGC population responses to latent modes for the Poisson MM versus the Tree HMM, we next computed the information efficiency (see [87]) of latent mode activation across repeats of the same stimulus for each model. As in [87], to compute the information efficiency, for each latent mode we computed the fraction of stimulus repeats on which that mode was active at time bin t, denoted $r(t)$, and its time average $\bar{r} = \frac{1}{T+1}\sum_t r(t)$. We then defined the output entropy of mode occurrence, S_{out}, and its noise entropy S_{noise}, as follows:

$$S_{\text{out}} = -\bar{r}\log_2\bar{r} - (1-\bar{r})\log_2(1-\bar{r}) \tag{5.1}$$

$$S_{\text{noise}} = -\frac{1}{T+1}\sum_{t=0}^{T} r(t)\log_2 r(t) + (1-r(t))\log_2(1-r(t)) \tag{5.2}$$

The information efficiency was then defined to be the mutual information divided by the output entropy: $(S_{\text{out}} - S_{\text{noise}})/S_{\text{out}}$. As shown in Fig 5.1B, as expected, the median information efficiency of the latent modes identified by the Tree HMM was higher than for the Poisson MM. The median value for the Poisson MM was still substantially higher than the information efficiency computed for the random partition control in [87] though, which showed that the chance value had a median of 0.14.

Given this result, as a first step and to gain better intuition, we decided to next investigate the performance of the WTA circuit model with competitive Hebbian plasticity presented in [47] for extracting fault-tolerant latent modes. We detail this circuit model in the next section.

5.2.2 The stochastic WTA circuit motif architecture

The structure of the WTA circuit motif model is illustrated in Fig 5.2, which depicts the circuit at one point in time (i.e. during one discrete time bin, t).

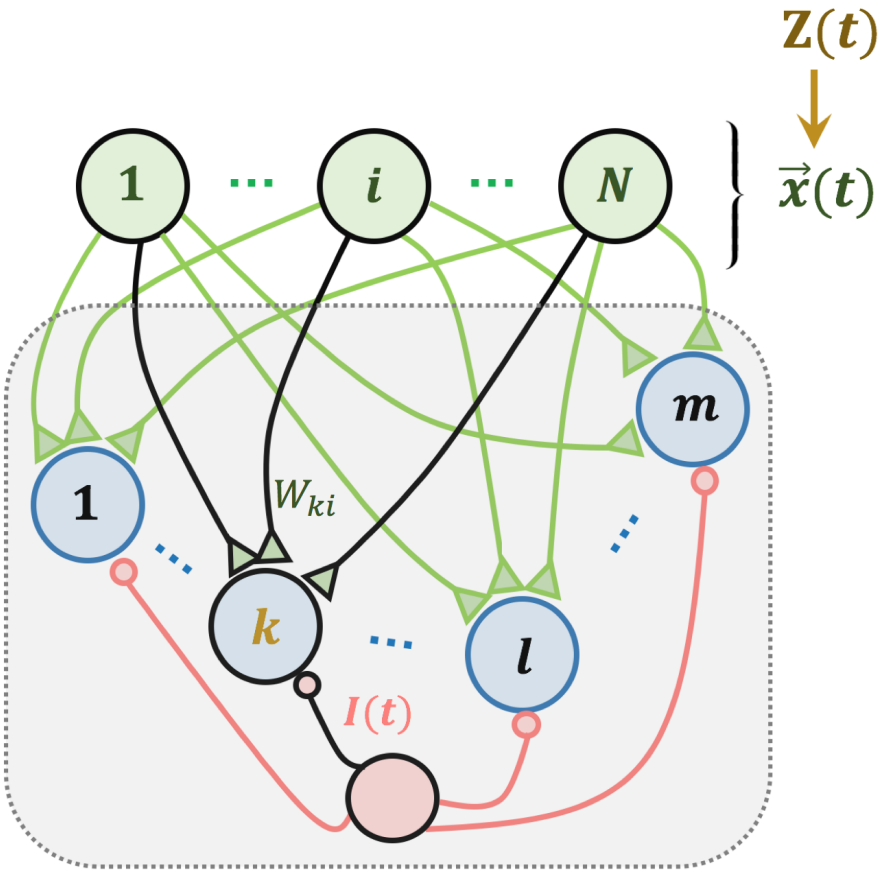

Figure 5.2: **Illustration of the WTA circuit motif model and population input.** Encompassed in grey are the components comprising the WTA circuit motif that, via STDP update rules and heterosynaptic plasticity, learns an implicit generative model to infer the current value k of the latent variable $Z(t)$, given only the input of the afferent neural population (green circles), $\vec{x}(t) \in \{0,1\}^N$. The structure of the network consists of m readout neurons (blue) that receive feedforward inputs (green synapses) from the afferent neurons. Interneurons (pink circle) install soft winner-take-all behavior by injecting a global inhibition $I(t)$ (common to all readout neurons, but dependent on time, t) in response to the network's spiking activity. If inference is performed correctly, then (only) readout neuron k will emit a spike. The local and quasi-local plasticity means that a given readout neuron's activity can only affect synapses that it participates in (for example, those shown in black for readout neuron k).

The scenario in this framework is as follows: The soft winner-take-all (WTA) network (highlighted in grey in Fig 5.2) is interpreted as the downstream network, i.e. in our case layer 4 of cortical area V1. The m excitatory readout neurons that comprise

the WTA network (schematized as blue circles in Fig 5.2) receive feedforward inputs (green synapses in Fig 5.2) from a population of N neurons in the upstream network (schematized as green circles in Fig 5.2).

The membrane potential of each readout neuron $k \in [m] \equiv \{1, \cdots, m\}$ at time t, denoted $u_k(t)$, consists of an excitatory and inhibitory contribution [47]:

$$u_k(t) = \sum_{i \in [N]} W_{ki}(t) x_i(t) - I(t) \tag{5.3}$$

where $W_{ki}(t)$ denotes the synaptic efficacy - commonly called the "synaptic weight" - of the synapse connecting neuron i in the afferent neuron population to readout neuron k of the WTA network (see Fig 5.2). The term $x_i(t)$ models the unweighted output spike train of the i-th sensory neuron after filtering according to the filtering properties of the postsynaptic membrane. In practice, we assume a step function with width τ (where τ denotes the discrete time bin width) for the EPSP kernel, which is consistent with the binning response representation used for the Tree HMM analyses. The term $W_{ki}(t)x_i(t)$ thus represents the contribution of previous spikes from the i-th sensory neuron to the membrane potential u_k of readout neuron k at time t.

$I(t)$ denotes the contribution to the membrane potential due to lateral inhibition (schematized in pink in Fig 5.2), which is common to all readout neurons. As in [47], we model $I(t)$ such that the total firing activity of the readout circuit, denoted $\hat{\nu}$, is approximately constant (in practice, we set $\hat{\nu} = 1$ spike/time bin). This corresponds with an inhibitory contribution given by

$$I(t) = \log \sum_{k \in [m]} \exp\left(\mathbf{W_k}(t)^\intercal \vec{x}(t)\right) - \log \hat{\nu} \tag{5.4}$$

where $\mathbf{W_k} \equiv (W_{k1}, \cdots, W_{kN}) \in \mathbb{R}^N$. This lateral inhibition serves to introduce competition among the readout neurons. For the choice of $\hat{\nu} = 1$ spike/time bin,

strict winner-take-all competition is introduced; note however that for $\hat{\nu} > 1$, a softer version of winner-take-all competition can be used.

5.2.3 Explicit mapping to a Bernoulli MM

The theoretical analysis presented in [47] relies on the notion of an implicit generative model implemented in the WTA network. The idea is that the parameters of the network, which are the synaptic weights $W \in \mathbb{R}^{m \times N}$, correspond with the parameters of a generative latent variable model (see Chapter 2 for background). This generative model is referred to as "implicit", because our interest is not in generating samples from the modeled multivariate probability distribution, and this is not what the network does. Rather, the idea is that each of the m readout neurons in the WTA circuit will, after learning (see the next section), correspond with one of the m latent modes of the generative model. In this way, the goal is that after learning the proper setting of the synaptic weights, the WTA network will in essence be able to map each (possibly noise-corrupted) input population response, $\vec{x}(t)$, to the correct latent state $k \in [m]$. The mapping to latent state k is represented by readout neuron k of the WTA network firing an action potential during time bin t (whereas the other readout neurons should be silent).

For our purposes then, we would like the the WTA network to represent an implicit generative model in which the latent variable $Z(t)$ corresponds to the RGC population codeword at time t. Unfortunately, the WTA network architecture shown in Fig 5.2 cannot represent the Tree HMM that we used as the generative model in our past work to identify the fault-tolerant clusters [63, 87]. However, given the result in Fig 5.1 that a simpler mixture model can still extract latent clusters that exhibit a significant (albeit lesser) degree of fault-tolerance, we still investigated the use of this architecture as a preliminary step.

Although [47] presented their simulation results for a WTA circuit model that represents a Poisson mixture model (MM), they showed that their framework can be generalized to all mixture models with emission distributions $p(\vec{x}(t)|Z(t) = k; \mathbf{W_k})$ from the single-parameter exponential family of distributions. Given that the Tree HMM analyses in our past work had assumed a binary response representation, i.e. assumed that each $\vec{x}(t) \in \{0,1\}^N$ (which is well-justified for the retina [13]), for the sake of comparison we wanted to use the same input response representation. We thus chose to first examine a WTA circuit model that represents an implicit Bernoulli mixture model, which models the joint probability of the afferent population responses as:

$$p(\vec{x}(t); \theta) = \sum_{k \in [m]} \omega_k \cdot \prod_{i \in [N]} \pi_{ki}^{x_i(t)} \left(1 - \pi_{ki}^{(1-x_i(t))}\right) \tag{5.5}$$

where $\omega_k \equiv p(k; \theta)$ is the *mixing weight* for component k, and $\theta = \{\pi_{ki}\}$ denotes the model parameters, where $k \in [m]$, $i \in [N]$. The relationship between the Bernoulli MM parameters $\{\pi_{ki}\}$ and the WTA network parameters $\{W_{ki}\}$ is given by

$$W_{ki} = \log\left(\frac{\pi_{ki}}{1 - \pi_{ki}}\right) \tag{5.6}$$

Note that, as in [47], we assume that the mixing weights ω_k are uniform for the preliminary analysis.

STDP learning rules for the Bernoulli MM WTA circuit

The general STDP learning rule proposed in [47] is:

$$\triangle \mathbf{W_k} \leftarrow \xi \cdot y_k(t) \cdot \mathbf{H}_A^{-1}(\mathbf{W_k}) \left[\vec{x} - \nabla A(\mathbf{W_k})\right] \tag{5.7}$$

where ξ denotes the learning rate, $y_k(t) \in \{0,1\}$ denotes the response of readout neuron k during time bin t, $\mathbf{W_k}$ denotes the current value (i.e. at time t) of the

synaptic weights ($W_{ki} : i \in [N]$), and $\mathbf{H}_A$ denotes the Hessian matrix of the real-valued function $A : \mathbb{R}^N \to \mathbb{R}$. For the Bernoulli MM, $A(\mathbf{W_k}) = c \sum_{i \in [N]} \log(1 + e^{W_{ki}})$, where c denotes a positive constant. Thus, for the WTA circuit model that represents an implicit Bernoulli MM, the local STDP update rules are:

$$\triangle W_{ki} \leftarrow \xi \cdot y_k(t) \left[\frac{x_i(t) \cdot (1 + e^{W_{ki}})^2}{c e^{W_{ki}}} - (1 + e^{W_{ki}}) \right] \tag{5.8}$$

5.2.4 The need for normalization (heterosynaptic plasticity)

In the Appendix, the details are provided for the proof of convergence of the STDP learning rule in Eq. 5.7 to drive the network parameters $\{W_{ki}\}$ to coincide with local maxima of the expected log-likelihood for a mixture model with m exponential family components (and uniform mixing weights). However, a crucial requirement for this convergence is that the WTA readout neurons must sample from the correct posterior. I.e., formally, the theory requires that $p(y_k(t) = 1 | \vec{x}(t); \mathbf{W_k}) = p(Z(t) = k | \vec{x}(t); \pi_{\mathbf{k}})$ for $\hat{\nu} = 1$ spike/time bin, where $y_k(t)$ denotes the response of readout neuron k at time t.

Since there is an explicit correspondence between $\{W_{ki}\}$ and $\{\pi_{ki}\}$, note that we can use these interchangeably. Letting $q_W(k, \vec{x}) \equiv p(Z = k, \vec{x}; \mathbf{W_k})$, by Baye's theorem:

$$\begin{aligned} p(Z = k | \vec{x}; \mathbf{W_k}) &= \frac{q_W(k, \vec{x})}{p(\vec{x}; W)} = \frac{q_W(k, \vec{x})}{\sum_{l \in [m]} q_W(l, \vec{x})} \\ &= \frac{p(k; W) \cdot p(\vec{x} | k; \mathbf{W_k})}{\sum_{l \in [m]} p(l; W) \cdot p(\vec{x} | l; \mathbf{W_l})} \\ &= \frac{\exp\left[\mathbf{W_k}^\intercal \vec{x}(t) - A(\mathbf{W_k})\right]}{\sum_{l \in [m]} \exp\left[\mathbf{W_l}^\intercal \vec{x}(t) - A(\mathbf{W_l})\right]} \end{aligned} \tag{5.9}$$

where the third line follows from the assumption of uniform mixing weights, and from writing in exponential family form $p(\vec{x} | k; \mathbf{W_k}) = h(\vec{x}) \exp(\mathbf{W_k}^\intercal \vec{x} - A(\mathbf{W_k}))$. Now,

for the circuit model, the probability of readout neuron k emitting an action potential in time bin t given the input population response $\vec{x}(t)$ is:

$$\begin{aligned} p(y_k(t) = 1|\vec{x}(t); \mathbf{W_k}) &= e^{u_k(t)} \\ &= \underbrace{\hat{\nu}}_{1} \cdot \frac{\exp\left[\mathbf{W_k}^\intercal \vec{x}(t)\right]}{\sum_{l\in[m]} \exp\left[\mathbf{W_l}^\intercal \vec{x}(t)\right]} \end{aligned} \tag{5.10}$$

Thus, we see that the stipulation that the WTA readout neurons must sample from the correct posterior is satisfied when $A(\mathbf{W_k})$ is constant for all readout neurons k.

To satisfy this requirement for the WTA circuit representing the Poisson MM, [47] made the assumption that every input population response pattern $\vec{x}(t)$ has the same spike count, i.e. that $\sum_{i=1}^{N} x_i(t) = A_0$ for some constant A_0. They showed that for a Poisson MM, this assumption implies that $A(\mathbf{W_k}) = c^{-1}A_0 \ \forall k$. For the Bernoulli MM, this implication no longer holds. More importantly however, for the case of real data, this simplifying assumption is grossly violated (the spike counts of RGC population responses are highly heterogeneous; specifically, the spike count distribution is generally positively skewed with a broad tail).

Rather than imposing constraints on the input population responses, we instead normalize each synaptic weight W_{ki} of the network after each time step, according to:

$$\hat{W_{ki}}(t) = \log\left(\exp\left[\frac{A_0}{c} \cdot \frac{\log(1 + e^{W_{ki}(t)})}{\mathcal{Z}_k(t)}\right] - 1\right) \tag{5.11}$$

where $\hat{W_{ki}}(t)$ denotes the normalized value, $\mathcal{Z}_k(t) \equiv \sum_i \log(1 + e^{W_{ki}(t)})$ and A_0 is an arbitrary constant that is chosen as a hyperparameter. By simple algebra, this ensures that $A(\mathbf{W_k}) = A_0 \ \forall k$ when the implicit generative model is a Bernoulli MM. We argue that, qualitatively, this normalization is biologically plausible, as multiple forms of heterosynaptic plasticity that enable such quasi-local synaptic modifications and aid in homeostasis have been documented experimentally [12]. For a more quantitative compatibility with biologically plausible neural computations, a linear approximation

of Eq. 5.11 could instead be substituted, as in practice we found that there is some tolerance to small differences in the $A(\mathbf{W_k})$ values.

Note that while in theory $A(\mathbf{W_k})$ could be incorporated directly into the circuit model (by modifying the membrane potential model, $u_k(t)$) to achieve the necessary correspondence, this approach is problematic for two reasons. First, the corresponding membrane potential model would be less biologically plausible. Secondly, we found that this incorporation makes the circuit model unstable in practice.

Preliminary results from test cases

To test the performance of the above WTA circuit model with normalized synaptic weights for the implicit Bernoulli MM representation, we created a test case with samples generated from a Bernoulli MM with known parameters. In this toy simulated test case, there are $N = 200$ neurons in the afferent (input) population, and $m = 5$ known latent states. As shown in Fig 5.3A, this test case has a broad range of input spike counts.

The normalization procedure described above has the potential disadvantage that the network will only be able to learn a proper subspace of the full space of all possible Bernoulli mixture models. Specially, the space of Bernoulli mixture models that satisfy the constraint that $A(\mathbf{W_k}) = -c\sum_{i=1}^{N}\log(1-\pi_{ki})$ is constant for all k (for some arbitrary constant A_0). However, the advantage is that this normalization ensures that the network is guaranteed to converge, via the above STDP learning rule dynamics, to a Bernoulli MM in this subclass. To gain some intuition for how this restriction influences the approximative performance of the network, we also designed this test case to contain groups of afferent neurons of different sizes that correspond with each known latent state (an idea that is consistent with our neuronal community results in [63]). We also added variability in the π_{ki} parameter values (Fig 5.3B). This heterogeneity induces distinct $A(\mathbf{W_k})$ values for the actual known

Bernoulli MM from which the N-dimensional samples are generated (Fig 5.3C).

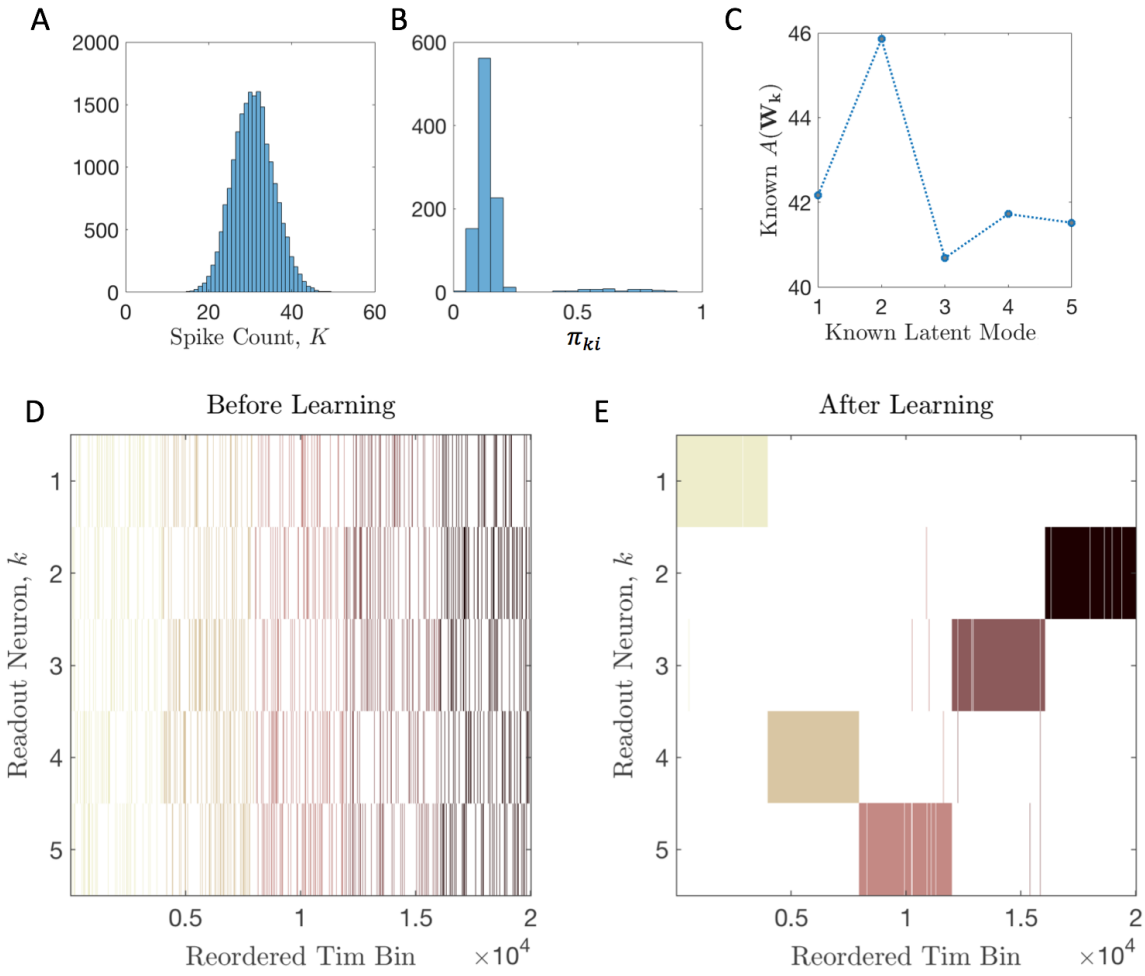

Figure 5.3: **Performance of the circuit model on a test case.**
Shown in panels (A-C) are characteristics of the simulated test case, which had $N = 200$ neurons in the simulated input population and $m = 5$ latent states. (**A**) The spike count distribution of the input population responses (over 20,000 generated samples). (**B**) The distribution of known π_{ki} parameter values for the Bernoulli MM from which the samples are generated. (**C**) This test case was constructed so that the actual $A(\mathbf{W_k})$ values are different across distinct k. (**D**) Raster plot of the performance of the WTA network *before* learning, i.e. with randomized synaptic weights W_{ki}. x-axis denotes the time bin, reordered according to the known latent state value. Colors denote the known latent state ID. Each row represents the spike train of the labeled readout neuron (y-axis). (**E**) Raster plot of the performance of the WTA network *after* learning. Same format as in (D).

The performance of the WTA circuit model (with normalized synaptic weights) on the above-described test case is shown in Fig 5.3. Whereas prior to learning

(Fig 5.3D), i.e. when the synaptic weights W_{ki} are random, the readout neurons exhibit no selectivity for the latent modes. In stark contrast, after learning via the STDP procedure, each readout neuron is specialized to represent one latent state, as demonstrated by the block structure in Fig 5.3E. To further quantify this selectivity, we computed each readout neuron's tuning curve as a function of the known latent states. As shown in Fig 5.4, each readout neuron exhibited a sharp tuning curve.

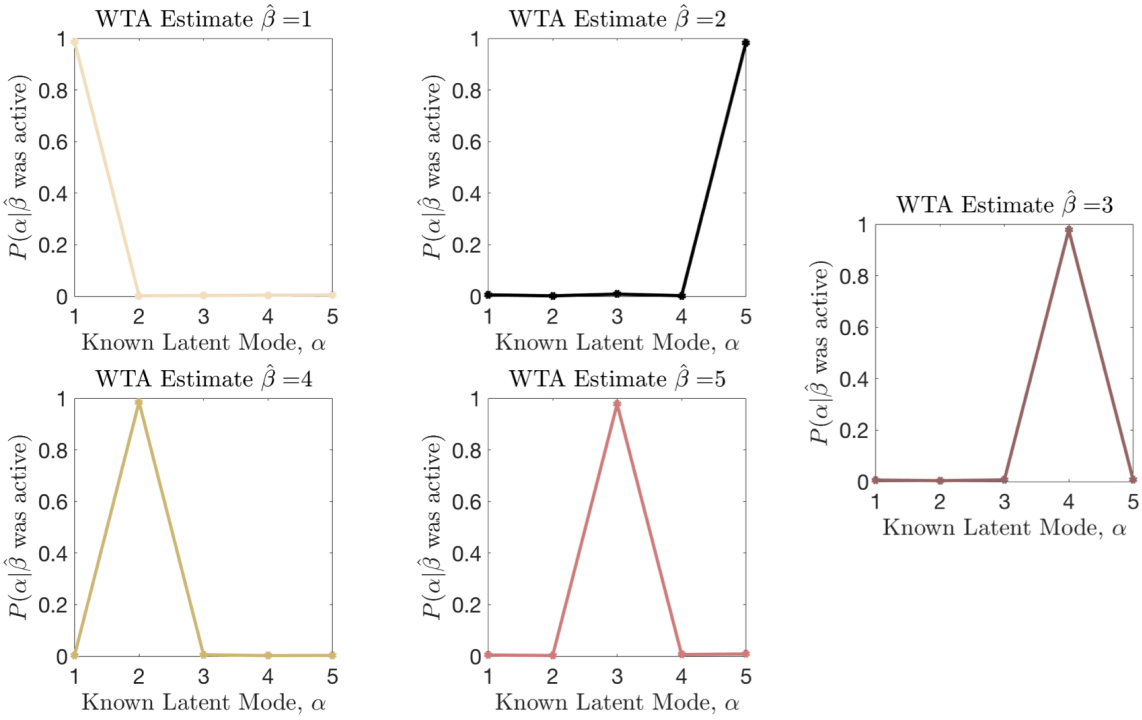

Figure 5.4: **Tuning curve results for the test case.**
Shown is the performance of the WTA circuit model with normalization (the same network as in Fig 5.3) *after* learning, quantified in terms of the tuning curves of each readout neuron in the WTA circuit. Each panels depicts the tuning curve for one of the five readout neurons after learning, as a function of the known latent mode (quantified over 20,000 presented samples). The color of each tuning curve represents the latent mode ID for which that readout neuron exhibits the sharpest tuning (colors corresponds with the same latent states as in Fig 5.3).

5.3 Discussion and next approaches

The next immediate step that is in progress is to extend the above circuit model to enable it to represent an implicit Bernoulli mixture model that potentially has non-uniform mixing weights, ideally without imposing restrictive assumptions on the input. This modification will be important to extend the applicability of the circuit model to actual input response data [87]. After we have incorporated a learning rule for the mixing weights that enables successful representation of an implicit Bernoulli MM with non-uniform mixing weights, we will then investigate the performance of the resulting circuit model on actual RGC population response data.

If the above is not successful, an alternative normative approach is to investigate a different circuit architecture and learning rule that would correspond with implicitly representing the Tree HMM or Tree mixture model. A weaker version of this would be to use the same or similar circuit architecture (see Fig 5.2), but to modify the STDP learning rule to enable capturing noise correlations. For example, we have thus far used a purely local update rule (see Eq. 5.8). However, the quasi-local generalization in Eq. 5.7 could potentially capture conditional dependencies among the inputs via non-zero off-diagonal terms of the Hessian matrix $\mathbf{H}_A(\mathbf{W_k})$. (For the case of the independent mixture models discussed here, which assume that input neuronal responses are independent conditioned on the latent mode, all non-diagonal terms of the Hessian matrix will be 0). This would enable the implicit generative model to potentially capture noise correlations that may be important for extracting the fault-tolerant clusters from the actual RGC population response data.

5.4 Appendix

5.4.1 Proof of convergence of STDP learning rules

Here, for completeness, we reiterate the proof given in [47], expanding on some details. Consider an MM with m exponential family components and uniform mixing weights:

$$\begin{aligned} p(\mathbf{x};\theta) &= \sum_{k\in[m]} \frac{1}{m} \cdot p\left(\vec{x}|k;\mathbf{W_k}\right) \\ &= \frac{1}{m} \cdot h(\vec{x}) \sum_{k\in[m]} \exp(\underbrace{\mathbf{W_k}^\intercal \vec{x} - A(\mathbf{W_k})}_{u_k}) \end{aligned} \tag{5.12}$$

where $\theta \equiv W \in \mathbb{R}^{m\times N}$ denotes the parameters of the model, and the log-partition function $A(\cdot) : \mathbb{R}^N \to \mathbb{R}$ is a real-valued function of the parameter vector $\mathbf{W}_k$. The goal of the generative model framework is to maximize the expected log-likelihood of the data under the model, $\mathbb{E}_{p^*(\vec{x})}\left[\log p(\vec{x};W)\right]$, i.e. to learn parameters W^* such that

$$\begin{aligned} W^* &\equiv \arg\max_W \mathbb{E}_{p^*(\vec{x})}\left[\log\left(\frac{1}{m}\cdot h(\vec{x})\sum_{k=1}^{m}\exp\left(u_k\right)\right)\right] \\ &= \arg\max_W \underbrace{\mathbb{E}_{p^*(\vec{x})}\left[\log\left(\sum_{k=1}^{m}\exp\left(u_k\right)\right)\right]}_{\equiv\phi(W)} \end{aligned} \tag{5.13}$$

where $p^*(\vec{x})$ denotes the empirical input distribution.

We want to show that the expected update of the STDP learning rule converges to a local optimum of $\phi(W)$ defined in Eq. 5.13. To do this, we must first show that the necessary conditions for local extrema of $\phi(W)$ are satisfied by the equilibrium points of the STDP learning rule. Of course by the first identification theorem a necessary condition for a local extremum W^* of $\phi(W)$ is that the gradient

$$\nabla_{\mathbf{W_k}}\phi|_{W^*} = \mathbf{0} \in \mathbb{R}^N \;\; \forall k \tag{5.14}$$

This implies that

$$\begin{aligned}\mathbf{0} &= \mathbb{E}_{p^*(\vec{x})}\left[\frac{1}{\sum_{l=1}^{m} e^{u_l}} \cdot \left(e^{\mathbf{W_k}^\intercal \vec{x}} \cdot \vec{x} \cdot e^{-A(\mathbf{W_k})} + e^{-A(\mathbf{W_k})} \cdot (-\nabla A(\mathbf{W_k})) \cdot e^{\mathbf{W_k}^\intercal \vec{x}}\right)\right] \\ &= \mathbb{E}_{p^*(\vec{x})}\left[\vec{x} \cdot \frac{e^{u_k}}{\sum_{l=1}^{m} e^{u_l}} - \nabla A(\mathbf{W_k}) \cdot \frac{e^{u_k}}{\sum_{l=1}^{m} e^{u_l}}\right] \\ &= \mathbb{E}_{p^*(\vec{x})}\left[\vec{x} \cdot p(k|\mathbf{x}; W^*)\right] - \nabla A(\mathbf{W_k}) \cdot \mathbb{E}_{p^*(\vec{x})}\left[p(k|x; W^*)\right]\end{aligned} \tag{5.15}$$

Therefore, a local extremum W^* must satisfy

$$\nabla A({\mathbf{W_k}}^*) = \frac{\mathbb{E}_{p^*(\vec{x})}\left[\vec{x} \cdot p(k|\mathbf{x}; W^*)\right]}{\mathbb{E}_{p^*(\vec{x})}\left[p(k|\mathbf{x}; W^*)\right]} \tag{5.16}$$

Thus, we must show that the equilibrium points of the learning rule introduced in [47],

$$\triangle \mathbf{W_k} = \xi \cdot y_k \cdot \mathbf{H}_A^{-1}(\mathbf{W_k})[\mathbf{x} - \nabla A(\mathbf{W_k})] \tag{5.17}$$

likewise satisfy Eq. 5.16, where $\mathbf{H}_{A,ij}(\mathbf{W_k})$ denotes the Hessian matrix of $A(\cdot)$.

By definition, an equilibrium point satisfies $\mathbb{E}_{p^*(\vec{x})}\left[\triangle \mathbf{W_k}\right] = \mathbf{0}$. Substituting Eq. 5.17 into this definition yields:

$$\mathbf{0} = \mathbb{E}_{p^*(\vec{x})}\left[\xi \cdot y_k \cdot \mathbf{H}_A^{-1}(\mathbf{W_k})\left(\vec{x} - \nabla A(\mathbf{W_k})\right)\right] \tag{5.18}$$

This can be rewritten, using the fact that $p(y_k = 1|\vec{x}; W) = p(k|\vec{x}; W)$, as

$$\mathbb{E}_{p^*(\vec{x})}\left[p(k|\vec{x}; W) \cdot \mathbf{H}_A^{-1}(\mathbf{W_k})\vec{x}\right] = \mathbb{E}_{p^*(\vec{x})}\left[p(k|\vec{x}; W)\mathbf{H}_A^{-1}(\mathbf{W_k})\right] \cdot \nabla A(\mathbf{W_k}) \tag{5.19}$$

Since $\mathbf{H}_A(\mathbf{W_k})$ is independent of $\vec{x}$, multiplying both sides of Eq. 5.19 by it yields

$$\begin{aligned}\mathbb{E}_{p^*(\vec{x})}\left[p(k|\vec{x}; W) \cdot I_N\vec{x}\right] &= \mathbb{E}_{p^*(\vec{x})}\left[p(k|\vec{x}; W)I_N\right] \cdot \nabla A(\mathbf{W_k}) \\ &= \nabla A(\mathbf{W_k}) \cdot \mathbb{E}_{p^*(\vec{x})}\left[p(k|\vec{x}; W)\right]\end{aligned} \tag{5.20}$$

Rearrangement of terms of Eq. 5.20 then yields Eq. 5.16, as desired.

Next, we need to show that the STDP learning rule specified in Eq. 5.17 always drives the parameters in the right direction, by showing that the dot product $\nabla\phi(W) \cdot \mathbb{E}_{p^*(\vec{x})}[\triangle W] \geq 0$. As derived in Eq. 5.15, differentiating $\phi(W)$ wrt $\mathbf{W_k}$ gives

$$\nabla_{\mathbf{W_k}}\phi|_W = \mathbb{E}_{p^*(\vec{x})}\left[p(k|\vec{x};W)(\vec{x} - \nabla A(\mathbf{W_k}))\right] \tag{5.21}$$

Expanding the dot product and substituting this in yields

$$\begin{aligned}
\nabla\phi(W) \cdot \mathbb{E}_{p^*(\vec{x})}[\triangle W] &= \sum_{k=1}^{m} (\nabla_{\mathbf{W_k}}\phi|_W)^\intercal \mathbb{E}_{p^*(\vec{x})}[\triangle \mathbf{W_k}] \\
&= \sum_{k=1}^{m} \mathbb{E}_{p^*(\vec{x})}\left[p(k|\vec{x};W)(\vec{x} - \nabla A(\mathbf{W_k}))^\intercal\right] \cdot \xi \\
&\quad \cdot \mathbf{H}_A^{-1}(\mathbf{W_k}) \underbrace{\left(\mathbb{E}_{p^*(\vec{x})}[p(k|\vec{x};W)(\vec{x} - \nabla A(\mathbf{W_k}))]\right)}_{\equiv \mathbf{a_k}} \\
&= \sum_{k=1}^{m} \xi \mathbf{a_k}^\intercal \mathbf{H}_A^{-1}(\mathbf{W_k}) \mathbf{a_k} \\
&\geq 0
\end{aligned} \tag{5.22}$$

where the last line follows, because $\mathbf{H}_A^{-1}(\mathbf{W_k})$ is a positive definite matrix.

5.4.2 C++ code for the circuit model

Below we provide our code (written in C++, but note that we also have a Mex file version available for Matlab) for the WTA circuit model with normalized synaptic weights, which represents an implicit Bernoulli MM.

```
//-----------------------------------------------------------------------
//  BernoulliWTA_TrainTest_UniPriors_Norm.h
//  Copyright 2017 Adrianna. All rights reserved.
//-----------------------------------------------------------------------

#ifndef BernoulliWTA_TrainTest_UniPriors_Norm_h
#define BernoulliWTA_TrainTest_UniPriors_Norm_h

#include <iostream>
#include <gsl/gsl_rng.h>
#include <vector>
#include <string>
#include <map>

using namespace std;

// ********************** myMatrix ****************************
template <class T>
class myMatrix
{
public:
    myMatrix();
    myMatrix(vector<T>&, int, int);
    void assign(vector<T>&, int, int);
    int get_m();
    int get_N();

    const T& at(const int k, const int i) const;
    const T& at(const int k) const;
    void addto(const int k, const int i, T a);
    void assign_entry(const int k, const int i, T a);

    vector<T>* data();

private:
    vector<T> matrix_data;
    int m,N;
};
// -- myMatrix Definition: --
template <class T>
myMatrix<T>::myMatrix() : m(0),N(0) {};

template <class T>
myMatrix<T>::myMatrix(vector<T>& _data, int _m, int _N) : m(_m), N(
    _N) {
    if (_data.size() != _m*_N) {
        cerr << "Matrix dimensions must agree." << endl;
        m = 0; N = 0;
    } else {
        matrix_data = _data;
    }
}

template <class T>
```

```
void myMatrix<T>::assign(vector<T>& _data, int _m, int _N) {
    if (_data.size() != _m*_N) {
        cerr << "Matrix dimensions must agree." << endl;
        m = 0; N = 0;
    } else {
        m = _m; N = _N;
        matrix_data = _data;
    }
    return;
}

template <class T>
int myMatrix<T>::get_m() { return m; }

template <class T>
int myMatrix<T>::get_N() { return N; }

template <class T>
const T& myMatrix<T>::at(const int k, const int i) const {
    return matrix_data[i*m + k];
}

template <class T>
void myMatrix<T>::addto(const int k, const int i, T a) {
    matrix_data[i*m + k] += a;
}

template <class T>
void myMatrix<T>::assign_entry(const int k, const int i, T a) {
    matrix_data[i*m + k] = a;
}

template <class T>
vector<T>* myMatrix<T>::data() { return &matrix_data; }

// ********************* paramsStruct ***************************
template<typename T>
struct paramsStruct
{
    myMatrix<T> W_star;     //the learned feedforward weights
    vector<T> Converg_avgW;
};

// ************************* RNG ******************************
class RNG
{
public:
    RNG();
    ~RNG();
    double uniform(double min, double max);
    int discrete(const vector<double>&);
    bool bernoulli(double);
    vector<int> randperm(int);
    double gaussian(double sigma, double mu);
private:
    gsl_rng* rng_pr;
};

// ************************ Spike *****************************
struct Spike
{
    double time;
    int bin;
    int neuron_ind;
};

```

```
// ******************** SpikeComparison ************************
class SpikeComparison
{
public:
    bool operator() (const Spike& lhs, const Spike& rhs) const
    {
        return (lhs.bin < rhs.bin);
    }
};

// ******************** WTACircuitModel ************************
class WTACircuitModel
{
public:
    WTACircuitModel(const string& filename, double binsize, int N,
   int m);
    paramsStruct<double> train_via_STDP(double binsize);
    vector<vector<bool> > test_WTA(double binsize);
    ~WTACircuitModel();
protected:
    //Constant params:
    int N;          //afferent input neuron population size
    int m;          //number of readout neurons
    double D;       //EPSP kernel param
    double tau1;    //EPSP kernel param
    double tau2;    //EPSP kernel param
    double mu_w;    //mean for initializing synaptic weights
    double sigma_w; //std for initializing synaptic weights
    double c;       //constant for weight update rule
    double xi;      //learning rate
    double nu_hat;  //total network instantaneous firing rate
    RNG* rng;

    //Dynamic/learned params:
    myMatrix<double> W;
    double current_inhib;

    //Data-handling:
    vector<Spike> all_spiketimes;
    vector<double> current_x;
    vector<Spike> sort_spikes(vector<Spike>&);

    //Normalization:
    double A0;                  //constant hyperparam
    vector<double> A_vec;
    double compute_A_Wk(int k);
    void normSynWeights(int k);   //normalizes the synaptic weights

    //To check convergence of learning:
    vector<double> deltaW;

    //Training via STDP:
    double EPSP_kernel(double delta, double binsize);
    void   compute_unweighted_spkvec(int n_t, double binsize);
    void   compute_current_inhibition();
    double compute_uk_hat(int k);
    double compute_rho_k(int k);
    double calcAvg_deltaW();      //for convergence tracking
};
#endif /* BernoulliWTA_TrainTest_UniPriors_Norm_h */
```

Listing 5.1: Header file

```cpp
//-------------------------------------------------------------------
//  BernoulliWTA_TrainTest_UniPriors_Norm.cpp
//  Copyright 2017 Adrianna. All rights reserved.
//-------------------------------------------------------------------

#include "BernoulliWTA_TrainTest_UniPriors_Norm.h"
#include <iostream>
#include <fstream>
#include <vector>
#include <queue>
#include <cmath>
#include <algorithm>
#include <exception>
#include <random>
#include <gsl/gsl_rng.h>
#include <gsl/gsl_randist.h>

int main(int argc, char *argv[]) {

    // ** Initializations: **
    int N          = 170;      //afferent neuron population size
    int m          = 5;        //# of readout neurons
    double binsize = 200;      //time bin width (units: 10 kHz)

    string datafile = "TestCase.txt"; //input spike times

    // ** WTA Spiking Circuit Model **
    WTACircuitModel WTA_obj(datafile, binsize, N, m);

    // ** Train on Observed Data & Return Learned Model Params: **
    paramsStruct<double> learned_W_b = WTA_obj.train_via_STDP(
   binsize);

    // ** Write Learned W Param Results to Output File: **
    string outfile1 = "Wstar_WTAwSTDP.txt";
    ofstream W_output(outfile1);
    for (int k=0; k<m; k++) {
        for (int i=0; i<N; i++) {
            W_output << learned_W_b.W_star.at(k,i) << " ";
        }
        W_output << endl;
    }
    W_output.close();

    // ** Test the Trained Network & Return Readout Output: **
    vector<vector<bool> > readout = WTA_obj.test_WTA(binsize);

    // ** Write Readout Neuron Responses to Output File: **
    string outfile4 = "ReadoutResps_WTAwSTDP.txt";
    ofstream y_output(outfile4);
    for (int n_t=0; n_t<readout.size(); n_t++) {
        vector<bool> y = readout[n_t];
        for (int k=0; k<m; k++) {
            y_output << y[k] << " ";
        }
        y_output << endl;
    }
    y_output.close();

    return 0;
}

// ************************* RNG Methods *************************
RNG::RNG() {
    // Initialize mersenne twister RNG
```

```
    rng_pr = gsl_rng_alloc(gsl_rng_mt19937);
}
RNG::~RNG() {
    gsl_rng_free(rng_pr);
}
double RNG::uniform(double min, double max) {
    double u = gsl_rng_uniform(rng_pr);
    return ((max-min)*u)+min;
}
int RNG::discrete(const vector<double>& p) {
    double u = gsl_rng_uniform(rng_pr);
    double c = p[0];
    int ix=0;
    while (c<u) {
        ix++;
        c += p[ix];
    }
    return ix;
}
bool RNG::bernoulli(double p) {
    return (gsl_rng_uniform(rng_pr) < p);
}
vector<int> RNG::randperm(int nmax) {
    //Description: This fn is analogous to Matlab's randperm(nmax)
    vector<int> nvals (nmax);
    for (int i=0; i<nmax; i++) {
        nvals[i] = i;
    }
    for (int i=0; i<nmax; i++) {
        // select random integer ix between i and nmax-1
        // swap i with ix
        unsigned long ix = i + gsl_rng_uniform_int(rng_pr, nmax-i);
        int tmp   = nvals[i];
        nvals[i]  = nvals[ix];
        nvals[ix] = tmp;
    }
    return nvals;
}
double RNG::gaussian(double sigma, double mu) {
    double g = gsl_ran_gaussian(rng_pr, sigma);
    return g+mu;
}

// ******************* WTACircuitModel Methods *********************
//Constructor:
WTACircuitModel::WTACircuitModel(const string& filename, double
   binsize, int N, int m) : N(N), m(m)
{
    rng = new RNG();

    //-- Instantiate constant parameters: --
    mu_w    = -2.9;
    sigma_w = 0.1;
    xi      = 0.001;
    nu_hat  = 1;       //units: spikes per time bin
    c       = 1.15;
    A0      = 30;

    //-- Load input population spike times: --
    cout << "Loading in spike time data..." << endl;
    vector<Spike> all_spikes;
    ifstream infile;
    infile.open(filename);
    double st;
    int nidx;
```

```
130     while( infile >> st >> nidx ) {
            Spike s;
132         s.time = st;          //Units: 10 kHz
            s.bin  = floor(st/binsize);
134         s.neuron_ind = nidx;
            all_spikes.push_back(s);
136     }
        infile.close();
138
        //-- Now sort spikes to be in chronological order: --
140     all_spiketimes = sort_spikes(all_spikes);
        cout << "Sorted spikes..." << endl;
142
        //-- Initialize W matrix: --
144     vector<double> W_int_vec;
        for (int ind=0; ind<(m*N); ind++) {
146         W_int_vec.push_back(rng->gaussian(sigma_w, mu_w));
            deltaW.push_back(0);
148     }
        W.assign(W_int_vec,m,N);
150
        //-- Apply the Normalization to Synaptic Weights: --
152     for (int k=0; k<m; k++) {
            normSynWeights(k);
154         A_vec.push_back(compute_A_Wk(k));
        }
156 }

158 WTACircuitModel::~WTACircuitModel() {
        delete rng;
160 }

162 vector<Spike> WTACircuitModel::sort_spikes(vector<Spike>& a_spikes)
    {
        sort(a_spikes.begin(), a_spikes.end(), SpikeComparison());
164     return a_spikes;
166 }

168 double WTACircuitModel::compute_A_Wk(int k) {
        double A_Wk = 0;
        for (int i=0; i<N; i++) {
170         A_Wk += log(1 + exp(W.at(k,i)));
        }
172     return c*A_Wk;
174 }

176 void WTACircuitModel::normSynWeights(int k) {
        double Z_k = 0;
        for (int j=0; j<N; j++) {
178         Z_k += log(1 + exp(W.at(k, j)));
        }
180     // Update each W_{ki} for the given k:
        for (int i=0; i<N; i++) {
182         double norm_Wki = log( exp((A0/c) * ( log(1 + exp(W.at(k,i))
    ) / Z_k )) - 1 );
            W.assign_entry(k,i, norm_Wki);
184     }
186 }

    paramsStruct<double> WTACircuitModel::train_via_STDP(double binsize)
        {
188     paramsStruct<double> params_learned;
        Spike last_s  = all_spiketimes.back();
190     int first_bin = all_spiketimes[0].bin;
        int n_T = last_s.bin;
192     vector<double> Converg_avgW(n_T+1 - first_bin);
```

```
194     for (int n_t=first_bin; n_t<=n_T; n_t++) {
196         std::vector<bool> y(m,0);
198         //(0a) Compute x(n_t) for the current time bin n_t:
            compute_unweighted_spkvec(n_t, binsize);
200
            //(0b) Compute i(n_t) (which is independent of k)
202         compute_current_inhibition();
204         double check_homog_netrate = 0;
206         for (int k=0; k<m; k++) {
                //(1) Compute inst. firing rate of neuron k
208             double rho_k = compute_rho_k(k);
                check_homog_netrate += rho_k;
210
                //(2) Draw from Bernoulli dist., &
212             //(3) Set y_k=1 if above is successful
                bool draw_k = rng->bernoulli(rho_k);
214             if (draw_k) { y[k] = 1; }
            }
216
            if (abs(check_homog_netrate-nu_hat)>0.001) { cerr << "Non-
    homogeneous network rate" << endl; }
218
            //** Apply STDP Updates: **//
220         for (int k=0; k<m; k++) {
                // -- Update synaptic weights: --
222             for (int i=0; i<N; i++) {
                    double deltaW_ki = xi * double(y[k]) * ( ((current_x
    [i]*pow(1+exp(W.at(k,i)),2))/(c*exp(W.at(k,i)))) - (1+exp(W.at(k,
    i))) );
224                 deltaW[i*m + k] = deltaW_ki;
                    W.addto(k, i, deltaW_ki);
226             }
            }
228
            //** Normalize the Synaptic Weights: **
230         for (int k=0; k<m; k++) {
                normSynWeights(k);
232             A_vec[k] = compute_A_Wk(k);
            }
234         //** Check Convergence: **
            Converg_avgW[n_t - first_bin] = calcAvg_deltaW();
236         cout << "Finished n_t = " << n_t << endl;
        }
238
        //Assign learned parameters
240     params_learned.W_star = W;
        params_learned.Converg_avgW = Converg_avgW;
242     return params_learned;
}
244
void WTACircuitModel::compute_unweighted_spkvec(int n_t, double
    binsize) {
246     double t = binsize*(n_t+1); //proxy for discretizing time
        int spk_ind = 0;
248     std::vector<double> x(N,0.0);
        Spike last_spk = all_spiketimes.back();
250     int n_T        = last_spk.bin;
252     Spike s = all_spiketimes[spk_ind];
        if (n_t < n_T) {
254         while (s.bin <= n_t) {
```

```
            x[s.neuron_ind] += EPSP_kernel(t-s.time, binsize);
            spk_ind++;
            s = all_spiketimes[spk_ind]; //move to next spike
        }
    }
    else {
        vector<Spike>::iterator it;
        for (it = all_spiketimes.begin(); it != all_spiketimes.end()
    ; ++it) {
            x[it->neuron_ind] += EPSP_kernel(t - it->time, binsize);
        }
    }
    current_x = x; //update for time bin n_t
}

double WTACircuitModel::EPSP_kernel(double delta, double binsize) {
    double y;
    if (delta <= binsize) { y=1; }
    else {y=0;}
    return y;
}

void WTACircuitModel::compute_current_inhibition() {
    double temp_summand = 0; //init
    for (int k=0; k<m; k++) {
        temp_summand += exp( compute_uk_hat(k) );
    }
    if (temp_summand < 0) { cerr << "Error: Negative values for log
    argument.\n"; }
    current_inhib = log(temp_summand) - log(nu_hat);
}

double WTACircuitModel::compute_uk_hat(int k) {
    double dotprod = 0;
    for (int i=0; i<N; i++) {
        dotprod += ( W.at(k,i) * current_x[i] );
    }
    return dotprod;
}

double WTACircuitModel::compute_rho_k(int k) {
    double rho_k = exp( compute_uk_hat(k) - current_inhib );
    return rho_k;
}

double WTACircuitModel::calcAvg_deltaW() {
    double sum = 0.0;
    for(std::size_t i = 0; i < deltaW.size(); i++)
        sum += deltaW.at(i);
    return sum/deltaW.size();
}

vector<vector<bool> > WTACircuitModel::test_WTA(double binsize) {
    Spike last_s  = all_spiketimes.back();
    int first_bin = all_spiketimes[0].bin;
    int n_T = last_s.bin;

    vector<vector<bool> > y_Cache(n_T-first_bin+1);

    for (int n_t=first_bin; n_t<=n_T; n_t++) {

        vector<bool> y(m,0);

        //(0a) Compute x(n_t) for the current time bin n_t:
        compute_unweighted_spkvec(n_t, binsize);

```

```
        //(0b) Compute i(n_t) (which is independent of k)
320     compute_current_inhibition();

322     for (int k=0; k<m; k++) {
            //(1) Compute \rho_k(n_t)
324         double rho_k = compute_rho_k(k);

326         //(2) Draw from Bernoulli dist., &
            //(3) Set y_k=1 if above is successful
328         bool draw_k = rng->bernoulli(rho_k);
            if (draw_k) { y[k] = 1; }
330     }

332     //** Cache vector y(n_t) \in {0,1}^m: **//
        y_Cache[n_t-first_bin] = y;
334
        cout << "Finished n_t = " << n_t << endl;
336 }
    return y_Cache;
338 }
```

Listing 5.2: Source file

Chapter 6

Population-Level Encoding of Invariance in the Retina

6.1 Introduction

Two major problems in sensory neuroscience are (1) determining what stimulus-specific information is encoded by dynamic patterns of ensemble neural activity, and (2) how the brain's sensory processing systems solve the *invariance problem*: that is, the ability to produce selective representations of relevant stimulus content in a manner that is tolerant to changes along 'irrelevant' stimulus dimensions [97]. For example, regarding (2), the visual system is tasked with extracting stimulus content (e.g. object identity) from the spatiotemporal light pattern falling on photoreceptors. However, visual information originating from the same content can differ (e.g. the same object viewed from different angles), requiring the system to extract content immune or *invariant* to these differences.

Invariance in the brain has traditionally been studied at the *single cell* level, where it is defined as a neuron maintaining its preference - i.e., the peak of its firing rate tuning curve - for particular stimulus variable(s) regardless of changes in 'irrelevant'

parameters [97]. Some examples of single-cell invariance in the visual system include: complex cells in primary visual cortex (V1), which exhibit approximate translation- and contrast-invariant orientation tuning [16, 53]; neurons in higher visual cortical areas which exhibit more complex forms, e.g. object-invariance [114]; and neurons in primate area MT and in avian tectum that exhibit form-cue invariant motion selectivity [64]. Commensurate with all of these findings, several extensive modeling studies have attempted to explain the emergence of *single cell* invariant responses [17, 53, 72].

In contrast, the question of whether and how information about invariance may also be encoded in *population* neural responses has been relatively unexplored. Given that there is substantial support that the brain employs population/combinatorial codes, one might naturally suspect that population-level invariant representations also exist. One reason why a population-level encoding scheme of invariance might be useful, is that it could allow for fewer processing stages to "read out" the invariant information. A general theme of the previous single-cell modeling work is the use of a hierarchical (i.e. multi-stage) processing scheme. While hierarchical processing schemes make sense in the visual cortex, they are less justified in explaining e.g. form-cue invariant responses of SGC-I neurons in the tectum, which, due to their monosynaptic contact with retinal ganglion cells [64], must obtain this property within a single decoding stage. While it is known that single RGCs do not exhibit the canonical invariances found in V1 complex cells (although see [71]), whether invariant properties are encoded in RGC *population* responses remains an open question.

This short chapter presents some preliminary results from a rotation project that I completed in my first year of graduate school, which focused on this question. While the results are very preliminary and the analysis methods are relatively simple compared to later work (which hopefully demonstrates progress!), this chapter is

included primarily with the hope that it may be useful in the off chance that anyone in the future in the Berry lab decides to investigate questions in this direction.

6.2 Methodology

For the purposes of this project, we will define invariance as a representation manifest as a population response that can be selectively mapped to an external stimulus invariant to 'irrelevant' stimulus dimension(s). Thus, to investigate population-level invariance, we must assume models of downstream processing areas whose input are the empirical RGC population responses, and whose mapped output are estimates of external stimuli (or stimulus features). In an engineered system, decoders are matched to encoders, which translate "messages" (e.g. external stimuli in our case) into codewords. Note that in our case, the neural circuit constitutes both the encoder and channel. Moreover, unlike the methods in Chapters 3-5, the decoding approaches used here are *supervised* methods, and consequently are not biologically plausible. However, one advantage of the decoding analysis approach described below is that it sets lower bounds on the mutual information [82].

6.2.1 Experimental procedures

Electrophysiology

For this project, MEA recordings of RGC responses were obtained according to the procedures described in Chapter 2. All experiments were performed by Mark Ioffe.

Visual stimulus design

In these pilot studies, the visual display consisted of an ensemble of stationary bar stimuli of constant size (34×139 pixels; total display 800×800 pixels). Each stimulus trial was $1\,s$ in duration, which consisted of 250 ms of stimulus presentation, followed

by 750 ms of grey screen (to prevent memory effects). Each unique bar stimulus identity was characterized by two parameters: position and orientation. The goal of this (highly artificial) stimulus design was to enable offline implementation of decoding analyses subsuming two paradigms: a "Discrimination" task and "Invariance" task, depicted in Fig 6.1 (see caption for details).

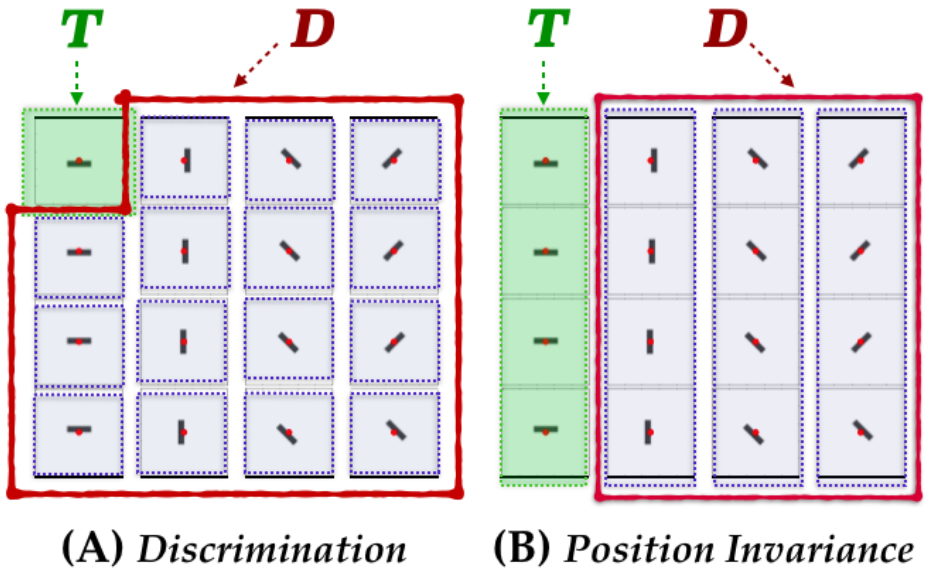

Figure 6.1: **Schematic of the stimulus design.**
(**A**) Example of one trial of the *Discrimination Task*. Each square in the matrix represents a unique bar stimulus, given by an $(orientation, position)$ pair (x-axis is orientation, y-axis is position). T corresponds with the *target* stimulus (green), i.e. the stimulus actually presented on that trial. D corresponds will all other stimuli. The task is, for each trial, to correctly discriminate T from D. (**B**) Example of one trial of the *Position Invariance Task*. Each column corresponds with a "unique stimulus"; i.e. all stimuli with the same *orientation* are considered equivalent. The task is to discriminate the target "stimulus", T, from D.

6.2.2 Optimal decoding analyses

Background on correlations in neural populations

Let $\vec{x}(t) = (x_1(t), \cdots, x_N(t)) \in \{0,1\}^N$ denote the population response (to external stimuli) of the population of N neurons at time t. This response is largely conditioned by three aspects: (1) the stimulus itself, (2) synaptic interactions, and (3) neuronal-

network history [24]. These are accounted for, respectively, by what are referred to in the neuroscience parlance as (1) signal correlations, (2) noise correlations, and (3) temporal correlations. The neurons in the network are said by neuroscientists to display *signal correlations* if their individual responses are *not* activity independent, that is, $P(\vec{x}(t)) \neq \prod_{i=1}^{N} P(x_i(t))$ (where $x_i(t)$ denotes the response of the ith neuron at time t, and $P(\cdot)$ denotes the empirical probability) [5]. In contrast, *noise correlations* exist if the neurons' responses are not conditionally independent given the stimulus, i.e. $P(\vec{x}(t)|s) \neq \prod_{i=1}^{N} P(x_i(t)|s)$. If the responses of the population depend on past responses, i.e. the system exhibits memory, then *temporal correlations* are said to exist.

Decoders used

For the pilot analyses, we used *maximum a posteriori* (MAP) optimal decoding, which in the two-discrimination case (using the notation in Fig 6.1) has the decoding rule:

$$\hat{s} = \begin{cases} T & , \, if \;\; \frac{P_M(\vec{x}|T)P(T)}{P_M(\vec{x}|D)P(D)} \geq \Theta \\ D & , \, otherwise \end{cases} \tag{6.1}$$

where $\hat{s}$ denotes the estimated stimulus, $P_M(\cdot)$ denotes a *modeled* probability density, and Θ denotes a (learned) threshold. As in [106], Θ was chosen so that the $hit\,rate = (1 - miss\,rate)$ is $> 99\%$. For all decoding analyses, cross-validation was used (i.e. the decoder was trained and tested on disjoint portions of the data). We initially investigated using the following three MAP decoders:

(i) *Spike Count Decoder*: Assumes that RGC firing probabilities do not carry additional information than spike counts, i.e. $P_M(\vec{x}|D) = \mathbb{E}_{s \in D}\left[P(K|s)\right]$, where $K \equiv \#$ of spikes summed over the population (and $\mathbb{E}_{s \in D}$ denotes the average over all distractor stimuli).

(ii) *Independent Decoder*: Assumes that responses x_i of individual RGCs in the population are *activity independent*, i.e. $P_M(\vec{x}|D) = \prod_{i=1}^{N} P(x_i|D)$, where $P(x_i|D) = \mathbb{E}_{s\in D}\left[P(x_i|s)\right]$.

(iii) *Mixture Decoder*: Fully incorporates signal correlations, but assumes that RGC responses are conditionally independent, i.e. $P_M(\vec{x}|D) = \mathbb{E}_{s\in D}\left[\prod_{i=1}^{N} P(x_i|s)\right]$.

Evaluating decoder performance

For this simple task, a decoder's performance can be fully described by its confusion matrix [99].[1] A single number often termed the "transmission information", $I(\mathcal{R},\mathcal{S})$ can be computed from the (normalized) confusion matrix, which is given by

$$\begin{aligned} I(\mathcal{R},\mathcal{S}) = -\sum_{a=1}^{|\mathcal{R}|}\left[\left(\sum_{b=1}^{|\mathcal{S}|} N^*(S_b,R_a)P(S_b)\right)\log_2\left(\sum_{b=1}^{|\mathcal{S}|} N^*(S_b,R_a)P(S_b)\right)\right] \\ + \sum_{b=1}^{|\mathcal{S}|}\sum_{a=1}^{|\mathcal{R}|} N^*(S_b,R_a)P(S_b)\log_2 N^*(S_b,R_a) \end{aligned} \tag{6.2}$$

where $\mathcal{R}$ denotes the set of "response classes" (here $\mathcal{R} = \{R_T, R_D\}$), $\mathcal{S}$ denotes the set of "stimulus classes" (here $\mathcal{S} = \{S_T, S_D\}$), and N^* denotes the normalized confusion matrix:

$$N^* := \frac{1}{n}\begin{pmatrix} \frac{\sum_{k=1}^{n} N_{TT}^{(k)}}{|T|} & \frac{\sum_{k=1}^{n} N_{TD}^{(k)}}{|T|} \\ \frac{\sum_{k=1}^{n} N_{DT}^{(k)}}{|D|} & \frac{\sum_{k=1}^{n} N_{DD}^{(k)}}{|D|} \end{pmatrix}$$

where n denotes the total number of cross-validation iterations, and e.g. $N_{TT}^{(k)}$ and $N_{DT}^{(k)}$ denote, respectively, the number of hits and false positives (false alarms) on cross-validation iteration $k \in [n]$.

In our case, $0 \leq I(\mathcal{R},\mathcal{S}) \leq -\left[P(S_T)\log_2(P(S_T)) + P(S_D)\log_2(P(S_D))\right]$, where the bounds correspond, respectively, to uniform random and perfect decoding. Here

[1] *N.B.* Due to sampling issues, it is intractable to compute the actual mutual information between the population responses and stimuli.

the *fractional transmission information*, defined as $\mathcal{I}_F = \frac{Actual\ I(\mathcal{R},\mathcal{S})}{\max I(\mathcal{R},\mathcal{S})}$, was used to quantify decoder performance. Unlike the false alarm rate, this measure can be used to compare decoding performance between tasks differing in the number of stimuli.

6.3 Preliminary results

For both of the pilot studies conducted, we found that the chosen stimulus parameter values were sufficient to drive retinal ganglion cell responses (6.2). Moreover, as shown in fig 6.2B, we found that the mean per-cell mutual information did not significantly differ across different parameter values.

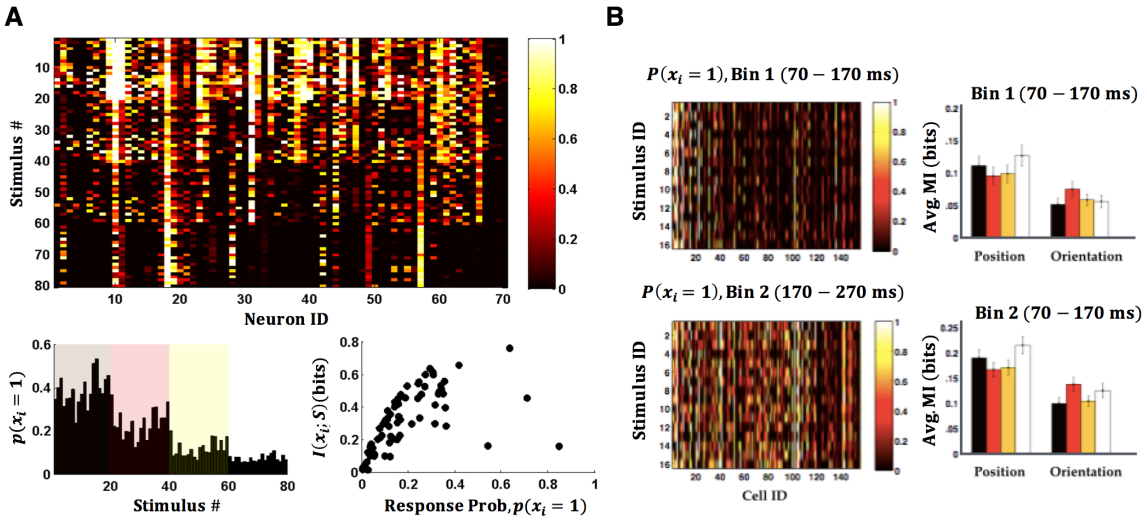

Figure 6.2: **Characterization of responses for pilot studies.**
(**A**) (*Top Panel*) Matrix of firing probabilities for each ganglion cell in the first time bin for the first pilot study ($N = 70$ neurons). Color scale denotes firing probability, $p(x_i = 1|s)$. (*Bottom Left*) Plotted is $\mathbb{E}_{i\in[N]}p(x_i = 1|s)$ vs. each stimulus s. (**B**) Results for the second pilot study ($N = 156$ neurons). (*Left Panels*) Same as in (*A*). (*Right Panels*) Mean per-cell mutual information (MI; bits) for all stimuli sharing a given stimulus parameter (e.g. position 1). (I.e. the bar for 'position 1' corresponds with $\mathbb{E}_i\left[I(x_i; \text{orientation}|\text{position 1})\right]$).

For both pilot studies, we found that the Mixture Decoder performed significantly better than the Independent Decoder on a discrimination task where stimulus identity is given by a (*position*, *orientation*) pair (Fig 6.3B,C), suggesting an important role

of signal correlations. Interestingly, the Mixture and Independent Decoder perform (nearly) perfectly on the Position Invariance Task, suggesting position invariance, but both performed poorly on the Orientation Invariance Task (Fig 6.3E).

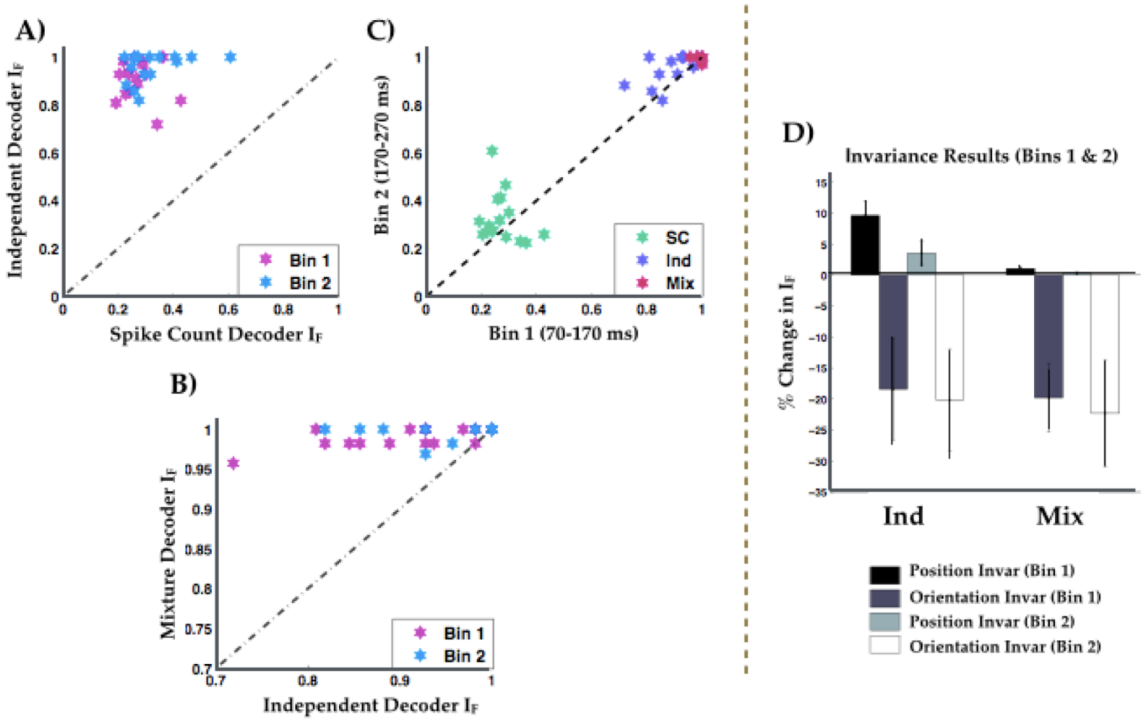

Figure 6.3: **Decoding results for the discrimination and invariance tasks.** (**A**) Scatter plot comparing the performance (quantified as fractional transmission information, $\mathcal{I}_F$) of the Spike Count vs. Independent Decoder on the Discrimination task (see Fig 6.1 for details). Performance when responses in bins 1 or 2 were used are denoted in pink and blue, respectively. (**B**) Same as (*A*), but for the Independent vs. Mixture Decoder. (**C**) Bin 1 vs. bin 2 performance for the Spike Count (SC, green), Independent (Ind, purple), and Mixture (Mix, pink) Decoders. (**D**) Percent change in $\mathcal{I}_F$ for the Position and Orientation Invariance tasks vs. the original Discrimination task. Both the Independent and Mixture Decoders perform better on the Position, but worse on the Orientation, Invariance task relative to the Discrimination task.

These very preliminary results suggest that information about position invariance is encoded in RGC population responses. Moreover, they show that an assumption of conditional independence of retinal ganglion cell responses is sufficient to decode position, but not orientation, invariance.

Chapter 7

Conclusion and Future Directions

We have used a data-driven approach within a framework informed by concepts from disparate fields - e.g. the notion of codewords from communications engineering and Bayesian inference from statistics - to investigate properties of the retinal ganglion cell population code. Our results in Chapter 3 suggested that emergent features are represented at the population level, and that another goal of the retinal system is to compute population "codewords" to enable error correction to combat neural noise. In Chapter 4, we then showed that, for the naturalistic low-repeat stimulus regime, these RGC population codewords do not correspond with local maxima of the population response probability landscape, as had been previously assumed. Rather, they correspond geometrically to "ridges" in the probability landscape, and moreover with what we term "neuronal communities" within the network of ganglion cells. These results naturally raise the question of how downstream brain areas such as V1 (primary visual cortex) could extract these population codewords, which we made a start at considering in Chapter 5.

In the following, we discuss some questions raised by this work and corresponding future directions.

Extensions to visual cortex data

There is no aspect of our approach in Chapters 3 and 4 that explicitly refers to properties of the retina: We made no assumptions about cells types, or receptive field properties of neurons, or functional models of feedforward sensory processing. In fact, nothing in our approach in these chapters refers to visual processing or even sensory systems. In theory then, the abstract nature of this approach raises the question of whether these results concerning the retinal ganglion cell population code might generalize to population codes in other regions of the central brain.

Though we have not investigated this question, in a very preliminary analysis, we did fit the Tree hidden Markov model to calcium imaging data of V1 population responses (which was coarsely thresholded to transform the $\triangle$F/F values to a binary representation). As shown in Fig 7.1, the Tree HMM was able to largely capture the first and second order moments.

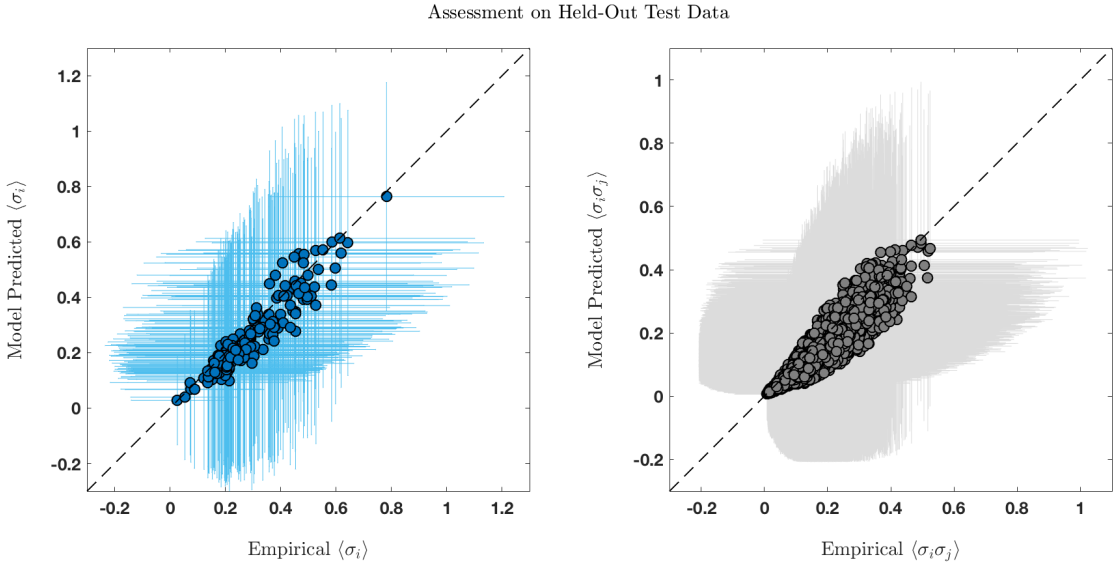

Figure 7.1: **Model goodness-of-fit for V1 data.**
(*Left panel*) Blue points denote mean first order moments predicted by the Tree HMM (y-axis) vs. the empirical mean values (x-axis). Blue lines denote one standard deviation. (*Right panel*) Same format, but for the second order moments.

For the analysis shown in Fig 7.1, we computed the model-predicted moments from 100,000 samples generated from the fit Tree HMM (where $m = 61$ latent modes was used as the latent dimensionality, as determined by 2-fold cross-validation). The Tree HMM was trained on the first half of the binarized V1 data, and these estimated model parameters were then used to test the performance of the model on the second held-out half of the dataset. (I.e. the empirical first and second order moments were calculated from the held-out test set). As a sanity check, we also verified that the empirical and model-predicted first and second order moments more closely aligned when we trained and tested on the entire dataset (Fig 7.2).

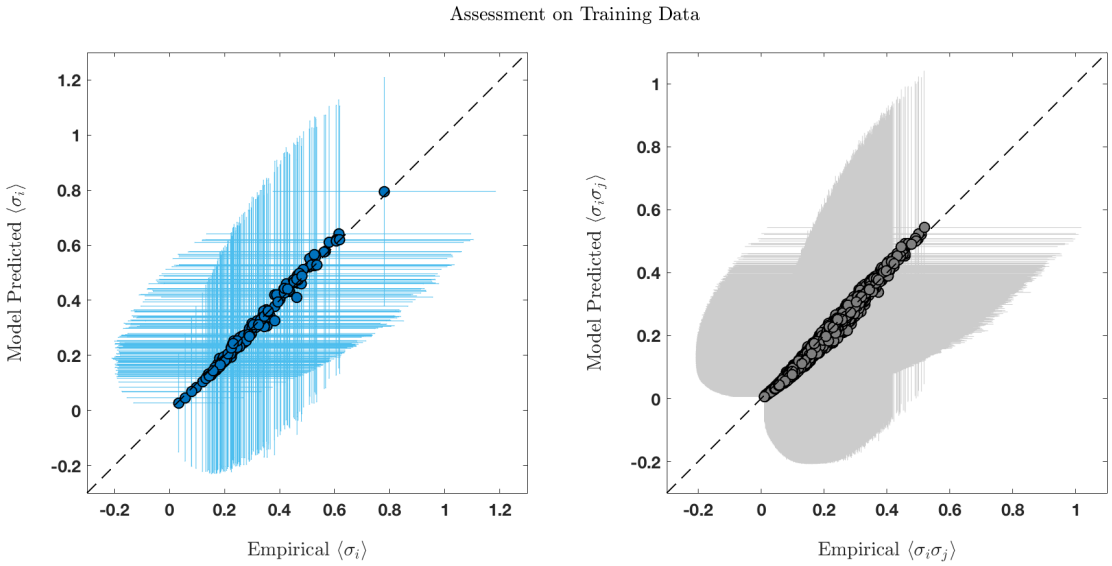

Figure 7.2: **Training set sanity check.**
Correspondence when the Tree HMM was both trained and tested on the entire dataset.

For analyses of cortical data, given the recurrent and feedback projections that are prevalent within cortical circuit architectures, we speculate that dynamics will need to be more seriously taken into account. For example, the history dependence of the Tree HMM may need to be extended beyond first-order Markovian, and the potential role of recurrence and feedback would likely need to be considered when investigating fault-tolerant properties of population codes in these brain regions.

Relations between RGCs in the same neuronal community

One question raised by our neuronal community results in Chapter 4 is how retinal ganglion cells that are members of the same neuronal community are functionally related. Though we have not investigated this question, from a visual neuroscience perspective, one potentially interesting direction would be to characterize how RGCs of the same neuronal community are related in terms of their spatial receptive fields. In classic and seminal work by Hubel and Wiesel in 1962 [53], a theory for how single V1 neurons could develop orientation tuning - despite the lack of orientation tuning exhibited by single retinal ganglion cells - was proposed. The Hubel and Wiesel model predicts a specific arrangement of the spatial receptive fields of the retinal ganglion cells. In terms of making connections with this classical work, it may thus be interesting to investigate whether the spatial receptive fields of the RGCs in the same neuronal community exhibit such an arrangement, especially for neuronal communities that correspond with collective modes that had exhibited orientation tuning [87].

Optimal design principles

Another question raised by our results in Chapters 3 and 4 is *why* the RGC population code is structured in this manner. In this vein, one potentially interesting direction would be to take an engineering viewpoint within the context of sparse coding in which one were to assume constraints on the codewords suggested by our previous results. In particular, it may be interesting to investigate how this structure relates to the fundamental trade-off between code rate and robustness (fault-tolerance).

Formalizing the ridge mapping

Finally, the ridges in Chapter 4 were algorithmically defined. Thus, an open question is whether we could formalize the ridge mapping. Relatedly, our ridge-search

algorithm presented in Chapter 4 works on a local basis; it is thus an open question whether the ridges could be globally defined. Very speculatively, a computational topology approach that incorporates global structure (e.g. separatrix persistence) may be an interesting avenue to investigate.

Bibliography

[1] Berry M.J. 2nd., I.H. Brivanlou, T.A. Jordan, and M. Meister. Anticipation of moving stimuli by the retina. *Nature*, 398(334-338), 1999.

[2] Ghazanfar A.A. and M.L. Nicolelis. Non-linear processing of tactile information by thalamocortical ensembles. *Journal of Neurophysiology*, 78:506–510, 1997.

[3] M.B. Ahrens, M.B. Orger, D.N. Robson, J.M. Li, and P.J. Keller. Whole-brain functional imaging at cellular resolution using light-sheet microscopy. *Nature Methods*, 12:413–420, 2013.

[4] D. Amodei. *Network-scale electrophysiology: Measuring and understanding the collective behavior of neural circuits.* PhD thesis, Princeton University, 2011.

[5] B.B. Averbeck, P.E. Latham, , and A. Pouget. Neural correlations, population coding, and computation. *Nature Rev. Neurosci.*, 7:358–366, 2006.

[6] A. Ayaz, A.B. Saleem, M.L. Scholvinck, and M. Carandini. Locomotion controls spatial integration in mouse visual cortex. *Current Biology*, 23:890–894, 2013.

[7] R. Azeredo da Silveira and M.J. 2nd. Berry. High-fidelity coding with correlated neurons. *Computational Biology*, 11:e1003970, 2014.

[8] H. Barlow. Redundancy reduction revisited. *Network*, 3:241–253, 2001.

[9] H.B. Barlow. Single units and sensation: a doctrine for perceptual psychology? *Perception*, 1:371–394, 1972.

[10] H.B. Barlow and R.M. Hill. Selective sensitivity to direction of movement in ganglion cells of the rabbit retina. *Science*, 139(412-414), 1963.

[11] D.J. Bartholomew. *Latent Variable Models and Factor Analysis: A Unified Approach*. Wiley, 2011.

[12] D. A. Baxter and J. H. Byrne. *The Neurobiology of Neural Networks*, chapter Learning rules from neurobiology, pages 71–105. MIT Press, 1993.

[13] M.J. Berry, D.K. Warland, and M. Meister. The structure and precision of retinal spike trains. *Proceedings of the National Academy of Sciences USA*, 97(10):5411–5416, 1997.

[14] N. Brenner, S.P. Strong, R. Koberle, W. Bialek, and R.R. de Ruyter van Stevenick. Synergy in a neural code. *Neural Computation*, 12:1531–1552, 2000.

[15] C. Bron and J. Kerbosch. Algorithm 457: finding all cliques of an undirected graph. *Communications of the ACM*, 16(9):575–577, 1973.

[16] M. Carandini, J.B. Demb, M. Mante, D.J. Tolhurst, Y. Dan, B.A. Olshausen, J.L. Gallant, and N.C. Rust. Do we know what the early visual system does? *Journal of Neuroscience*, 25(46):10577–10597, 2005.

[17] M. Carandini and D.G. Heeger. Summation and division by neurons in primate visual cortex. *Science*, 264:1333–1336, 1994.

[18] G. Celeux and J. Diebolt. The sem algorithm: A probabilistic teacher algorithm derived from the em algorithm for the mixture problem. *Comput Statist Quater*, 2:73–82, 1985.

[19] C. Chen, H. Liu, D.N. Metaxas, and T. Zhao. Mode estimation for high fimensional discrete tree graphical models. *Advances in Neural Information Processing Systems*, 27:5533–5551, 2014.

[20] E.J. Chichilnisky. A simple white noise analysis of neuronal light responses. *Network*, 12(2):199–213, 2001.

[21] M. Chistiakova, N.M. Bannon, J.Y. Chen, M. Bazhenov, and M. Volgushev. Homeostatic role of heterosynaptic plasticity: models and experiments. *Frontiers in Computational Neuroscience*, 9(89), 2015.

[22] C.K. Chow and C.N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3):462–467, 1968.

[23] P.S. Churchland and T.J. Sejnowski. *The computational brain*. MIT Press, Cambridge, MA, 1994.

[24] R. Cofre and B. Cessac. Exact computation of the maximum-entropy potential of spiking neural-network models. *Physical Review E*, 89, 2014.

[25] C. Curto, V. Itskov, K. Morrison, Z. Roth, and J.L. Walker. Combinatorial neural codes from a mathematical coding theory perspective. *Neural Computation*, 25(7):1891–1925, 2013.

[26] Y. Dan, J.M. Alonso, W.M. Usrey, and R.C. Reid. Coding of visual information by precisely correlated spikes in the lateral geniculate nucleus. *Nature Neuroscience*, 1(6):501–507, 1998.

[27] P. Dayan and L. Abbott. *Theoretical neuroscience*. MIT Press, Cambridge, MA, 2001.

[28] A.P Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society Series B*, 39(1):1–38, 1977.

[29] S. Deneve. Bayesian spiking neurons i: Inference. *Neural Computation*, 20(1):91–117, 2008.

[30] G. Desbordes, J. Jin, C. Weng, N.A. Lesica, G.B. Stanley, and J.M. Alonso. Timing precision in population coding of natural scenes in the early visual system. *PLoS Biology*, 6(12):e324, 2008.

[31] N.R. Deshmukh. *Complex computation in the retina*. PhD thesis, Princeton University, 2015.

[32] D.W. Dong and J.J. Atick. Statistics of natural time-varying images. *Network*, 6:345–358, 1995.

[33] R.J. Douglas and A.C. Martin. Neuronal circuits of the neocortex. *Annual Review of Neuroscience*, 27:419–451, 2004.

[34] K. Doya, S. Ishii, A. Pouget, and R.P. Rao. *Bayesian Brain: Probabilistic Approaches to Neural Coding*. MIT Press, 2011.

[35] M. Dudik, S.J. Phillips, and R.E. Schapire. Performance guarantees for regularized maximum entropy density estimation. In *Proceedings 17th Annual Conference on Learning Theory*, 2004.

[36] D. Eberly, R. Gardner, B. Morse, S. Pizer, and C. Scharlach. Ridges for image analysis. *Journal of Mathematical Imaging Visualization*, 4:353–373, 1994.

[37] A.L. Fairhall, J.D. Lewen, W. Bialek, and R.R. de Ruyter van Stevenick. Efficiency and ambiguity in an adaptive neural code. *Nature*, 412:787–792, 2001.

[38] A.A. Faisal, L.P. Selen, and D.M. Wolpert. Noise in the nervous system. *Nature Reviews Neuroscience*, 9:292–303, 2008.

[39] J. Fiser, P. Berkes, G. Orban, and M. Lengyel. Statistically optimal perception and learning: from behavior to neural representation. *Trends in Cogn Sciences*, 14:119–130, 2010.

[40] S. Fortunato. Community detection in graphs. *Physics Reports*, 485(3):75–132, 2010.

[41] S. Fujisawa, A. Amarasingham, M.T. Harrison, and G. Buzsaki. Behavior-dependent short-term assembly dynamics in the medial prefrontal cortex. *Nature Neuroscience*, 11:823–833, 2008.

[42] E. Ganmor, R. Segev, and E. Schneidman. Sparse low-order interaction network underlies a highly correlated and learnable neural population code. *Proceedings of the National Academy of Sciences of the United States of America*, 108(23):9679–84, 2011.

[43] E. Ganmor, R. Segev, and E. Schneidman. A thesaurus for a neural population code. *Elife*, 4:e06134, 2015.

[44] T. Gollisch and M. Meister. Eye smarter than scientists believed: Neural computations in circuits of the retina. *Neuron*, 65(2):150–164, 2013.

[45] M.D. Golub, B.M. Yu, and S.M. Chase. Internal models for interpreting neural population activity during sensorimotor control. *Elife*, page 10.7554/eLife.10015, 2015.

[46] D.S. Greenberg, A.R. Houweling, and Kerr J.D. Population imaging of ongoing neuronal activity in the visual cortex of awake rats. *Nature Neuroscience*, 11:749–751, 2008.

[47] S. Habenschuss, H. Puhr, and W. Maass. Emergence of optimal decoding of population codes through stdp. *Neural Computation*, 25:1371–1407, 2013.

[48] R. Haralick. Ridges and valleys in digital images. *Graphical Models and Image Processing*, 22:28–38, 1983.

[49] C.D. Harvey, P. Coen, and D.W. Tank. Choice-specific sequences in parietal cortex during a virtual-navigation decision task. *Nature*, 484:62–68, 2012.

[50] D.O. Hebb. *The Organization of Behavior*. Wiley and Sons, New York, 1949.

[51] Y. Hu, J. Trousdale, K. Josic, and E. Shea-Brown. Motif statistics and spike correlations in neuronal networks. *Journal of Statistical Mechanics: Theory and Experiment*, (03):P03012, 2013.

[52] H. Huang and T. Toyoizumi. Clustering of neural code words revealed by a first-order phase transition. *Physical Review E*, 93:062416, 2016.

[53] D.H. Hubel and T.N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *Journal of Physiology*, 160(1):106–154, 1961.

[54] M.L. Ioffe. *Adaptation in Coding by Large Populations of Neurons in the Retina*. PhD thesis, Princeton University, 2017.

[55] L. Itti and C. Koch. Computational modelling of visual attention. *Nature Reviews Neuroscience*, 2:194–203, 2001.

[56] I.L. Jones, T.L. Russell, K. Farrow, M. Fiscella, F. Franke, J. Muller, D. Jackel, and A. Hierlemann. A method for electrophysiological characterization of hamster retinal ganglion cells using a high-density cmos microelectrode array. *Frontiers in Neuroscience*, 9, 2015.

[57] J.F. Kalaska and D.J. Crammond. Cerebral cortical mechanisms of reaching movements. *Science*, 255:1517–1523, 1992.

[58] P. Kara, P. Reinagel, and R.C. Reid. Low response variability in simultaneously recorded retinal, thalamic, and cortical neurons. *Neuron*, 27(3):635–646, 2000.

[59] R. Kiani, H. Esteky, K. Mirpour, and K. Tanaka. Object category structure in response patterns of neuronal population in monkey inferior temporal cortex. *Journal of Neurophysiology*, 97:4296–4309, 2007.

[60] H. Kobayashi, B.L. Mark, and W. Turin. *Probability, Random Processes, and Probability, Random Processes, and Statistical Analysis.* MIT Press, 2012.

[61] K.C. Lakshmanan, P.T. Sadtler, E.C. Tyler-Kabara, A.P. Batista, and B.M. Yu. Extracting low-dimensional latent structure from time series in the presence of delays. *Neural Computation*, 27:1825–1856, 2015.

[62] C.R. Legendy. On the scheme by which the human brain stores information. *Mathematical Biosciences*, 1:555–597, 1967.

[63] A.R. Loback, J.S. Prentice, M.L. Ioffe, and Berry M.J. 2nd. Noise-robust modes of the retinal population code have the geometry of "ridges" and correspond with neuronal communities. *Neural Computation*, 29(12), 2017.

[64] H. Luksch, R. Khanbabie, and R. Wessel. Synaptic dynamics mediate sensitivity to motion independent of stimulus details. *Nature Neuroscience*, 7(4):380–388, 2004.

[65] F.J. MacWilliams and N.J.A. Sloane. *The Theory of Error-Correcting Codes.* Elsevier Ltd., Amsterdam, Netherlands, 1977.

[66] Z.F. Mainen and T.J. Sejnowski. Reliability of spike timing in neocortical neurons. *Science*, 268:1503–1506, 1995.

[67] D. Marr. *Vision.* W.H. Freeman and Co., 1982.

[68] O. Marre, D. Amodei, N. Deshmukh, K. Sadeghi, F. Soo, T.E. Holy, and M.J. Berry. Mapping a complete neural population in the retina. *Journal of Neuroscience*, 43:14859–14868, 2012.

[69] M. Meila and T. Jaakkola. Tractable bayesian learning of tree belief networks. *Statistical Computing*, 1(77-92), 16.

[70] M. Meister and M.J. 2nd. Berry. The neural code of the retina. *Neuron*, 22:435–450, 1999.

[71] B.W. Mel, D.L. Ruderman, and K.A. Archie. Translation-invariant orientation tuning in visual "complex" cells could derive from intradendritic computations. *Journal of Neuroscience*, 18(11):4325–4334, 1998.

[72] F. Michler, R. Eckhorn, and T. Wachtler. Using spatiotemporal correlations to learn topographic maps for invariant object recognition. *Journal of Neurophysiology*, 102(2):953–964, 2009.

[73] D.A. Miller and S.W. Zucker. Computing with self-excitatory cliques: A model and an application to hyperacuity-scale computation in visual cortex. *Neural Computation*, 11(1):21–66, 1999.

[74] T. Mora, S. Deny, and O. Marre. Dynamical criticality in the collective activity of a population of retinal neurons. *Physics Review Letters*, 114:078105, 2015.

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

[76] S. Negahab, P. Ravikumar, M.J. Wainwright, and B. Yu. A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. *Statistical Science*, 27(4):538–557, 2012.

[77] M.L. Nicolelis, E.E. Fanselow, and A.A. Ghazanfar. Hebb's dream: The resurgence of cell assemblies. *Neuron*, 19:219–221, 1997.

[78] I.E. Ohiorhenuan, F. Mechler, K.P. Purpura, A.M. Schmid, Q. Hu, and J.D. Victor. Sparse coding and high-order correlations in fine-scale cortical networks. *Nature*, 466:617–621, 2010.

[79] B.P. Olveczky, S.A. Baccus, and M. Meister. Segregation of object and background motion in the retina. *Nature*, 423(401-408), 2003.

[80] L.C. Osborne, S.G. Lisberger, and W. Bialek. A sensory source for motor variation. *Nature*, 437:412–416, 2005.

[81] S.E. Palmer, O. Marre, Berry M.J. 2nd., and W. Bialek. Predictive information in a sensory population. *Proceedings of the National Academy of Sciences USA*, 112(22):6908–6913, 2015.

[82] L. Paninski, J. Pillow, and J. Lewi. *Computational Neuroscience: Progress in Brain Research*, chapter Statistical models for neural encoding, decoding, and optimal stimulus design, pages 493–507. Elsevier Ltd., 2006.

[83] C. Pehlevan and D.B. Chklovskii. A normative theory of adaptive dimensionality reduction in neural networks. *NIPS*, 2015.

[84] D. Perkel and T. Bullock. Neural coding. *Neuroscience Research Program Bulletin*, 6:221–343, 1968.

[85] J.W. Pillow, J. Shlens, L. Paninski, A. Sher, A.M. Litke, E.J. Chichilnisky, and E.P. Simoncelli. Spatio-temporal correlations and visual signalling in a complete neuronal population. *Nature*, 454:995–999, 2008.

[86] A. Pouget, P. Dayan, and R. Zemel. Information processing with population codes. *Nature Rev. Neurosci.*, 1:125–132, 2000.

[87] J.S. Prentice, O. Marre, M.L. Ioffe, A.R. Loback, G. Tkacik, and M.J. Berry. Error-robust modes of the retinal population code. *PLoS Computational Biology*, (e1005148), 2016.

[88] J.L. Puchalla, E. Schneidman, R.A. Harris, and M.J. 2nd. Berry. Redundancy in the population code of the retina. *Neuron*, 46:493–504, 2005.

[89] R. Quian-Quiroga, L.H. Snyder, A.P. Batista, H. Cui, and R.A. Andersen. Movement intention is better predicted than attention in the posterior parietal cortex. *Journal of Neuroscience*, 26:3615–3620, 2006.

[90] R. Quian-Quiroga and P. Stefano. Extracting information from neuronal populations: information theory and decoding approaches. *Nature Reviews*, 10:173–185, 2009.

[91] R.P. Rao. Bayesian computation in recurrent neural circuits. *Neural Computation*, 16(1):1–38, 2004.

[92] R.P. Rao. Hierarchical bayesian inference in networks of spiking neurons. *Advances in Neural Information Processing Systems*, 27(1113-1120), 2005.

[93] F. Rieke, D. Warland, R. de Ruyter van Steveninck, and W. Bialek. *Spikes: exploring the neural code*. MIT Press, Cambridge, MA, 1999.

[94] R.W. Rodieck. Quantitative analysis of cat retinal ganglion cell response to visual stimuli. *Vision Research*, 5(11):583–601, 1965.

[95] J. S. Rodney and K.A.C. Martin. Recurrent neuronal circuits in the neocortex. *Current Biology*, 17(13), 2007.

[96] R.M. Roth. *Introduction to Coding Theory*. Cambridge University Press, 2006.

[97] N.C. Rust and A.A. Stocker. Ambiguity and invariance: two fundamental challenges for visual processing. *Current Opinion in Neurobiology*, 20:382–388, 2010.

[98] Y. Sakurai. Population coding by cell assemblies - what it really is in the brain. *Neuroscience Research*, 26:1–16, 1996.

[99] M.T. Schaub and S.R. Schultz. The ising decoder: reading out the activity of large neural ensembles. *Journal of Computational Neuroscience*, 32:101–118, 2012.

[100] E. Schneidman, M.J. 2nd Berry, R. Segev, and W. Bialek. Weak pairwise correlations imply strongly correlated network states in a neural population. *Nature*, 440:1007–1012, 2006.

[101] E. Schneidman, B. Freedman, and I. Segev. Ion channel stochasticity may be critical in determining the reliability and precision of spike timing. *Neural Computation*, 10:1679–1703, 1998.

[102] E. Schneidman, J.L. Puchalla, R. Segev, R.A. Harris, W. Bialek, and M.J. 2nd. Berry. Synergy from silence in a combinatorial neural code. *Journal of Neuroscience*, 44:15732–15741, 2011.

[103] M.J. Schnitzer and M. Meister. Multineuronal firing patterns in the signal from eye to brain. *Neuron*, 37(499-511), 2003.

[104] B. Scholl, A.Y. Tan, J. Corey, and N.J. Priebe. Emergence of orientation selectivity in the mammalian visual pathway. *Journal of Neuroscience*, 33(26):10616–10624, 2013.

[105] G. Schwartz, R.A. Harris, D. Shrom, and Berry M.J. 2nd. Detection and prediction of periodic patterns by the retina. *Nature Neuroscience*, 10:552–554, 2007.

[106] G. Schwartz, J. Macke, D. Amodei, H. Tang, and Berry MJ 2nd. Low error discrimination using a correlated population code. *Journal of Neurophysiology*, 108:1069–1088, 2012.

[107] R. Segev, J. Goodhouse, J.L. Puchalla, and Berry M.J. 2nd. Recording spikes from a large fraction of the ganglion cells in a retinal patch. *Nature Neuroscience*, 7:1155–1162, 2004.

[108] M. Shamir and H. Sompolinsky. Implications of neuronal diversity on population coding. *Neural Computation*, 18:1951–1986, 2006.

[109] C.E. Shannon. A mathematical theory of communication. *Bell Labs Technical Journal*, 27:379–423, 1948.

[110] S.M. Sherman. The thalamus is more than just a relay. *Current Opinion in Neurobiology*, 17:417–422, 2007.

[111] J. Shlens, G.D. Field, J.L. Gauthier, M. Greschner, A. Sher, A.M. Litke, and E.J. Chichilnisky. The structure of large-scale synchronized firing in primate retina. *Journal of Neuroscience*, 15:5002–5031, 2009.

[112] L.C. Sincich, J.C. Horton, and T.O. Sharpee. Preserving information in neural transmission. *Journal of Neuroscience*, 29:6207–6216, 2009.

[113] S. Skiena. *The Algorithm Design Manual*. Springer, 2008.

[114] H. Tamura and K. Tanaka. Visual response properties of cells in the ventral and dorsal parts of the macaque inferotemporal cortex. *Cerebral Cortex*, 11:384–399, 2001.

[115] K. Thulasiraman and M.S. Swamy. *Graphs: Theory and Algorithms*. Wiley-Interscience, 1998.

[116] G. Tkacik, O. Marre, D. Amodei, E. Schneidman, W. Bialek, and Berry M.J. 2nd. Searching for collective behavior in a large network of sensory neurons. *PLoS Computational Biology*, 10(1):e1003408, 2014.

[117] G. Tkacik, J.S. Prentice, V. Balasubramanian, and E. Schneidman. Optimal population coding by noisy spiking neurons. *Proceedings of the National Academy of Sciences USA*, 32:14419–14424, 2010.

[118] W. Truccolo, L.R. Hochberg, and J.P. Donoghue. Collective dynamics in human and monkey sensorimotor cortex: predicting single neuron spikes. *Nature Neuroscience*, 18:105–111, 2008.

[119] W.M. Usrey, J.B. Reppas, and R.C. Reid. Paired-spike interactions and synaptic efficacy of retinal inputs to the thalamus. *Nature*, 295(6700):384–387, 1998.

[120] S.C.F. van Opheusden. Critical states in retinal population codes. Master's thesis, Universiteit Leiden, 2013.

[121] R.J. Vapnik. *Statistical Learning Theory*. Wiley & Sons, New York, 1998.

[122] J.D. Victor and K.P. Purpura. Nature and precision of temporal coding in visual cortex: A metric-space analysis. *Journal of Neurophysiology*, 76(2):1310–1327, 1996.

[123] J.D. Victor and K.P. Purpura. Spatial phase and the temporal structure of the response to gratings in v1. *Journal of Neurophysiology*, 80(2):554–571, 1998.

[124] M. Vidne, Y. Ahmadian, J. Shlens, J.W. Pillow, J. Kulkarni, A.M. Litke, E.J. Chichilnisky, E.P. Simoncelli, and L. Paninski. Modeling the impact of common noise inputs on the network activity of retinal ganglion cells. *Journal of Computational Neuroscience*, 33(1):97–121, 2012.

[125] M.A. Weiss. *Data Structures and Algorithm Analysis in C++*, volume 4. Pearson Education, Inc., 2014.

[126] W.A. Wickelgren. Webs, cell assemblies, and chunking in neural nets. *Concepts in Neuroscience*, 1:1–53, 1992.

[127] Y. Ziv, L.D. Burns, E.D. Cocker, E.O. Hamel, K.K. Ghosh, L.J. Kitch, A. El Gamal, and M.J. Schnitzer. Long-term dynamics of ca1 hippocampal place codes. *Nature Neuroscience*, 16:264–266, 2013.

[128] E. Zohary, M.N. Shadlen, and W.T. Newsome. Correlated neuronal discharge rate and its implications for psychophysical performance. *Nature*, 370:140–143, 1994.