A SILICON MODEL OF THE PRIMARY VISUAL CORTEX: REPRESENTING FEATURES THROUGH STOCHASTIC VARIATIONS

Paul A. Merolla

A DISSERTATION

in

Bioengineering

Presented to the Faculties of the University of Pennsylvania in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

2006

Kwabena Boahen,	David F. Meaney, Graduate Group Chair
Supervisor of Dissertation	

Abstract

A SILICON MODEL OF THE PRIMARY VISUAL CORTEX: REPRESENTING FEATURES THROUGH STOCHASTIC VARIATIONS

Paul A. Merolla

Supervisor: Kwabena Boahen

In this project, we have presented the bump chip, a silicon model of the primary visual cortex that accounts for the emergence of an orientation map. Two traits set our chip apart from every other man-made system to date: 1) its functional architecture (orientation selectivity) is not specifically engineered *a priori*, and 2) the scaffold for this architecture is innate to the chip itself and exists as an indelible imprint. To achieve this feat, our chip uses a simple feedback network to form patterns of neural activity; these patterns, which are biased by random component mismatch, serve as the scaffold for the selectivity. Therefore, our chip attains all of the benefits of a self-organizing learning system *without* having to go through the tedious process of learning. We propose that cortical maps observed in biology are built using a similar design principle. This idea is supported experimentally; orientation maps in kittens are remarkably robust to experimental manipulations, and appear without the need for visual experience.

Contents

Abstract **iii**

Contents **iv**

List of Tables **x**

List of Figures **xi**

1 Building a visual cortex in silicon **1**

2 Development of orientation maps **4**

2.1 Development of cortical maps 4

2.1.1 Establishment and maintenance of orientation selectivity 5

2.1.1.1 Establishment phase 7

2.1.1.2 Maintenance phase 9

2.1.2 Map stability during development 11

2.1.3 Role of activity in map establishment 14

2.1.4 Origin of map structure 17

2.2 Models of orientation map establishment 21

2.2.1 Correlation-based models 23

2.2.1.1 Correlation-based learning 26

2.2.2 Alternative models 28

2.2.2.1 Haphazard wiring 28

2.2.2.2 Pattern formation using morphogens 30

2.2.2.3 Pattern formation using neural activity 33

2.2.3 Physical constraints 36

3 Neuromorphic bump model **40**

3.1 Pattern formation in V1 40

3.1.1 Complex cells solve a complex problem 44

3.2 Neuromorphic models 46

3.2.1 Neuromorphic building blocks 47

3.2.2 Neuromorphic microcircuit 48

3.3 Bump chip implementation 49

4 Orientation maps *in silico* 52

4.1 Chip testing 53

4.1.1 Setup 53

4.1.1.1 Chip communication 54

4.1.1.2 Board-level spike processing 54

4.1.1.3 GUI 58

4.1.2 Experimental procedure 59

4.2 Innate selectivity 63

4.2.1 Orientation selectivity 63

4.2.1.1 Visualizing SCMs 64

4.2.1.2 Repeated stimulus reliability 66

4.2.1.3 Inter-stimuli reliability 70

4.2.1.4 Smooth mapping 72

4.2.2 Prefered orientation maps 73

4.3 Quantifying selectivity 78

4.3.1 Normalized vector magnitude 80

4.3.2 Information 81

4.3.2.1 General background 81

4.3.2.2 Estimating mutual information 84

4.3.2.3 Results 89

4.3.3 Comparing NVM and information 90

5 Characterizing Network Dynamics **93**

5.1 Dynamic activity patterns 93

5.1.1 Experimental setup 94

5.1.2 Extracting temporal structure 95

5.1.3 Interpreting $\langle \sum_{\mathrm{i}} |\Delta| a_{\mathrm{i}}(t) \rangle$ 96

5.2 Dynamic bumps 98

5.2.1 Identifying bumps 99

5.2.2 Emergent features 101

5.2.3 Estimating consistent motion 103

5.2.4 Extracting local displacements 104

5.2.5 Confined bump motion 107

5.3 Non-classical bump motion 109

5.3.1 Time course of non-classical behaviors 110

5.3.1.1 Time course of $N(t)$ 111

5.3.1.2 Time course of $S_{\mathrm{j}}(t)$ 112

5.3.2 Quantifying non-classical behaviors 113

5.3.2.1 Isolating non-diffusive and diffusive behaviors 114

5.3.2.2 Bump area 118

5.3.3 Staccato motion 120

6 Maps from motion **123**

6.1 Motion regimes 123

6.1.1 Overview of dynamic quantities 124

6.1.2 Dependence on network parameters 126

6.1.2.1 Reducing the parameter space 127

6.1.2.2 Observed diffusion rates 128

6.1.2.3 Observed network dynamics 129

6.2 Dynamic computation 132

6.2.1 PO convergence 132

6.2.2 Selectivity 136

6.3 Mechanistic origins of motion and selectivity 138

6.3.1 Dislodging confined bumps 139

6.3.2 Bump motion with oriented inputs 142

6.3.2.1 Modified experimental procedure and analysis 142

6.3.2.2 Tracking bump orientations 144

6.3.2.3 Verification 148

7 Conclusion **154**

7.1 Bump-based analysis for describing dynamics 156

7.2 Testable predictions for biological maps 158

7.3 Does biology really use noise? 159

Bibliography **161**

List of Tables

6.1 Dynamic quantities 126

6.2 Fits for $\langle C_i(s) \rangle$ 135

List of Figures

2.1 Columnar organization of V1 6

2.2 Innate orientation map 9

2.3 Map stability in developing ferret 13

2.4 Development timeline 14

2.5 Spontaneous LGN activity 15

2.6 Phenotype diversity 19

2.7 Self-organizing orientation map 24

2.8 Center-type correlations for orientation selectivity 27

2.9 Haphazard model of orientation selectivity 30

2.10 Chemical pattern formation 31

2.11 Connection profile and activity pattern 34

2.12 Orientation selectivity using neural pattern formation 37

3.1 Hypothetical ON–OFF segregation 42

3.2 Two layer recurrent network 43

3.3 Microcircuit 46

3.4 Circuit blocks 47

3.5 Silicon microcircuit 50

3.6 Bump chip layout 51

4.1 PCB test setup 55

4.2 GUI 59

4.3 Single condition maps 64

4.4 Normalized single condition maps 67

4.5 Split-half reliability test 68

4.6 Selectivity index and repeatability 71

4.7 Summary of cell reliability across stimuli pairs 73

4.8 Orientation selectivity of the bump chip 75

4.9 Direction selectivity of the bump chip 77

4.10 Map comparison for square and sine -wave inputs 78

4.11 Computing normalized vector magnitude 81

4.12 Bump chip selectivity 82

4.13 PDFs and mutual information for a single cell 89

4.14 Information map of bump chip 90

4.15 NVM vs. Information 92

5.1 Network dynamics 94

5.2 Temporal structure of frame activity 97

5.3 Bump identification 100

5.4 Affect of bump identification on PO 101

5.5 Temporal structure of bump displacements 106

5.6 $\langle \Delta^2 r(t) \rangle$ for ideal matter states 109

5.7 Evidence of non-classical behaviors 110

5.8 Time course of $N(t)$ 112

5.9 Distribution of absolute area changes 113

5.10 Temporal structure of behavior types 116

5.11 Temporal structure of bump areas 119

5.12 Uncovering staccato motion 122

6.1 Map of diffusion rates 129

6.2 Network traits 131

6.3 PO map convergence 134

6.4 PO map comparisons 136

6.5 Comparison of selectivity 137

6.6 Spike dynamics 140

6.7 Distributions of BEOs 145

6.8 Relationship between RBPO and RBEO 147

6.9 Deconstructed PO maps for state 29 150

6.10 Deconstructed PO maps for state 15 151

6.11 Deconstructed PO maps for state 4 152

6.12 Summary of SI between deconstructed PO maps 153

6.13 Removing edge effects 153

Chapter 1

Building a visual cortex in silicon

The patterns of light reflected off of objects offer a fantastic opportunity for biological organisms to gain knowledge about their environment. To take advantage of this informational windfall, mammals have evolved highly specialized visual systems that are able to recognize shapes, detect complex motion patterns, and analyze scenes. In humans, for example, the culmination of vision processing is a visual percept, which represents the skillful combination of relevant aspects of a scene across parallel visual streams while suppressing superfluous details. Artificial vision systems built around traditional computing architectures, on the other hand, have been less successful at recovering information about the three-dimensional world in real-time; even rudimentary tasks such as shape recognition, have proven to be difficult in non-ideal environments.

One promising approach to close the gap between biological and machine vision is to *morph* the structure and function of neural wetware into silicon hardware — a methodology coined neuromorphic engineering [1]. This strategy has proven to be successful for neural structures that have relatively simple organizations. For example, transistor circuits

that mimic ion channels, chemical–electrical connections, and neurons have been brought together to model all five layers of the mammalian retina. The result is a silicon retina that converts light into neural activity similar to its biological counterpart; in addition to operating in real-time, the silicon retina is also power efficient using a mere 60 milliwatts (1,000 times less power than a typical personal computer) [2].

Invigorated by the success of the silicon retina, we would like to morph neural areas that are further downstream in the visual pathway. As we venture deeper into the brain, however, the organization becomes increasingly complex; the primary visual cortex, for instance, systematically maps features such as orientation, direction, color, and eye preference all in the same network. How are we ever going to keep up with this burgeoning complexity? One strategy, which we explore in this project, is to model at the level of development — a methodology coined *metamorphing* [3]. Whereas morphing copies neural circuits into silicon, metamorphing also copies the algorithms necessary to organize these circuits.

In this thesis, we present the bump chip, the first neuromorphic system that can self-organize an orientation map. Consistent with the finding that orientation map development is experience-independent, our design does not rely on learning mechanisms for organizing its selectivity. Instead, our chip uses a simple feedback network to form patterns of neural activity; these patterns, which are biased by random component mismatch, serve as the scaffold for the selectivity. The painstaking task of wiring up orientation selective neurons has been reduced to building an imprecise network that generates the necessary activity patterns. We elaborate on this idea in the following chapters:

Chapter 2 reviews how the primary visual cortex initially becomes selective to orientation from biological and theoretical viewpoints. First, we describe the key experiments that show orientation map establishment does not require vision, and its establishment is robust

to adversity. Then, we offer a critical review of common models that can account for map ontogenesis. We focus on an orientation-selective network that obtains its selectivity by generating spot-like patterns of neural activity; our neuromorphic model is based on this same pattern-formation principle.

Chapter 3 presents the bump chip, a large-scale model of the primary visual cortex that is designed to automatically generate an orientation map.

Chapter 4 characterizes the responses of the bump chip to orientation. We show that the chip responds selectivity to orientation, and this selectivity is repeatable; these results verify that biases introduced during chip fabrication provide a sufficient scaffold for map generation.

Chapter 5 introduces a novel analysis for characterizing the dynamics observed in the bump chip. This analysis reduces chip activity to a collection of bumps, or particles, allowing us to quantify dynamics using intuitive metrics, such as particle diffusion rates. We also explore how activity deviates from our intuition by quantifying non-classical dynamics (such as vanishing, regenerating, and teleporting particles).

Chapter 6 applies our particle-based analysis to determine how dynamics in different network states relate to network computation.

Chapter 7 concludes the thesis with suggestions for experimentalists to evaluate whether neural pattern formation is important for organizing cortical networks.

Chapter 2

Development of orientation maps

2.1 Development of cortical maps

Nearly three and a half centuries ago, John Locke first proposed the idea that at birth, the mind is a *tabula rasa*, or a "blank slate", whose contents are ultimately determined by experiences [4]. The debate that Locke's proposition sparked, which is more commonly known as the nature versus nurture debate, has been raging in religious, philosophical, and legal circles over the centuries [5]. While it is unlikely that these abstract debates will be resolved anytime soon, the literal implications of Locke's *tabula rasa* hypothesis are beginning to be addressed experimentally: that is, to what extent are cortical circuits and their functions specified by innate factors (nature), and to what extent are they shaped by their inputs (nurture)?

It is now known that the initial organization of the cortical structure is largely determined by innate factors. For example, individuals of the same species all have: 1) similarly

parcellated cortical areas, 2) similar topographic connectivity between the areas, and 3) similar basic neural circuits within the areas (interlaminar connectivity) [6]; in a sense, this predetermined cortical structure can be thought of as Locke's blank slate. The functional architecture of the cortex, on the other hand, seemingly requires externally driven events for its development (see Figure 2.1 for a brief review of the organization of V1). For example, even though the qualitative features of orientation maps are largely similar across individuals, the detailed layouts can be quite different (e.g., the specific locations of orientation columns are ostensibly random across individuals). The question that we will explore in this section is how a characteristic, yet individualized orientation map initially becomes imprinted on the cortical slate.

2.1.1 Establishment and maintenance of orientation selectivity

We review the experimental evidence that demonstrates there are two distinct phases in the development of orientation selectivity: an initial phase that establishes selectivity, and a subsequent phase that maintains it [8]. The distinction between these two phases is not arbitrary; the establishment phase, which occurs within the first week of eye opening, is *independent* of visual experience, whereas the maintenance phase requires it. Surprisingly, the notion that cortical development occurs in two phases is relatively recent, even though the division was apparent in the very first experiments performed in the field (see [9] for a historical account). Specifically, Hubel and Wiesel demonstrated in 1965 that orientation responses were present in kittens near the time of eye opening, and selectivity degraded over time when reared in abnormal conditions [10].

For our immediate discussion, we focus on the development of the cat visual system, however, it is expected (or has been shown) that the visual systems in most highly visual

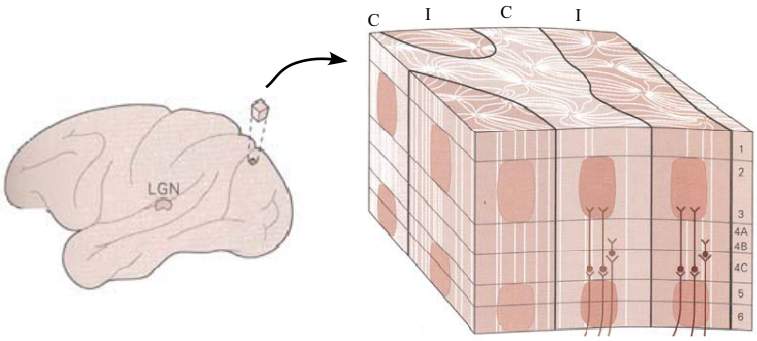


Figure 2.1: Columnar organization of V1

The primary visual cortex (area V1) is a two-dimensional neural sheet organized in six layers, and is the first (and largest) cortical region in the visual pathway. Similar to the neural regions that precede V1 (retina and LGN), visual space is mapped topographically across the network ensuring that nearby cells respond to nearby locations in space. Unlike it inputs, however, V1 also maps higher-order visual features such as orientation, eye, and color preference. There are two organizational principles that the cortex follows: First, cells that extend the thickness of the cortex at each retinotopic position have the same preferences, and form a functional column. And second, nearby columns have preferences that change smoothly across the network (except at fractures), and these preferences repeat at regular intervals. This columnar organization has led to the idea that V1 can be thought of an array of functional modules (or hypercolumns) that represent all combinations of visual features for a particular voxel in space. The highlighted cortical cube depicts one such hypercolumn (this depiction is based on monkey V1). The shaded slab-like regions separated by dark lines correspond to contra (C) and ipsi (I) -lateral eye preferences, spoke-like patterns shown by white lines along the surface correspond to contours of orientation preference, and blob regions correspond to columns involved in color processing [7].

mammals, including humans, monkeys, ferrets, and tree shrews, develop in a similar manner. It is worth noting that the specific phases of development relative to birth can vary across species and is dependent on the different gestation periods, although these phases are roughly equivalent relative to eye opening. For example, cats have a 65 day gestation period and open their eyes around the tenth postnatal day (P10), whereas ferrets have a shorter gestation period (42 days) and open their eyes around P31.

2.1.1.1 Establishment phase

Following the 1965 experiments performed by Hubel and Wiesel, a number of groups have confirmed that cells in V1 are selective to orientation at eye opening [11, 12, 13, 14].[1] While there is a general consensus on the existence of orientation selective cells at eye opening, there are conflicting accounts on how many such cells exist; for example, some studies (Blakemore et al., and Fregnac et al.) find that about 25% of the cells are selective, whereas other studies (Braastad and Heggelund, and Albus and Wolf) find that more than 78% of cells have reliable selectivity. Braastad and Heggelund articulate the difficulties they encountered while recording from single units, which can potentially account for the discrepancies across labs: In their experience, traditional 'handplotting' techniques with static slits was not possible in the youngest kittens because the center of the discharge zone was too variable (even with the optimal stimulus). However, they were able to measure statistically significant tuning curves by presenting slowly moving, long slits, with 4–5s pauses between presentations [14]. Using this procedure, they lost nearly half of their cells during the rather tedious analysis. Therefore, although selectivity exists in single neurons, it is clear that studying cortical development in more than a few cells at a time is a painstaking

[1] There have even been groups that have shown a few cells in V1 are selective to orientation (in ferret) through closed eye lids (i.e., *prior* to eye opening) [15]. However, the firing rates of these cells are quite low and the tuning curves are rather inconsistent.

process [16].

In the ensuing week after eye opening (P11–P17), orientation selectivity in single units mature to near adult levels. Somewhat unexpectedly, this initial maturation is *independent* of visual experience [11, 17] — a result that has been the source of much confusion over the past decades [9]. For example, a number of experiments have attempted to disrupt the formation of orientation selectivity by manipulating visual inputs (e.g., dark rearing or lid suturing) in the weeks following eye opening; however, these experiments were inadvertently exploring the maintenance of selectivity and *not* its initial formation since the manipulations extended past P18 [8]. The seemingly conflicting results of these experiments can be resolved once the establishment phase is appropriately identified.

Around P14, intrinsic signals in V1 become reliable enough to measure prefered orientation (PO) maps using optical imaging techniques; PO is calculated by taking each pixel's average activity for each stimulus orientation, and computing the angle of the vector sum. Consistent with the electrophysiology data, the quality of the maps improve in an experience independent manner until P18 [18]. These early PO maps have traits that are strikingly similar to adult maps, including spatially extended orientation domains that repeat at regular intervals, as well as fractures and pinwheels. We show an example of a map imaged from a kitten near eye opening (Figure 2.2).

In summary, the evidence reviewed thus far demonstrates that the establishment of orientation selectivity is an experience independent process. It seems that the cortical slate does not start out blank after all, but is instead prewired to extract edges from the visual world; next, we will explore what happens to the map when normal visual experience is manipulated past the establishment phase. We note, however, that our main focus in this project is on the establishment of orientation maps; we describe the maintenance phase for

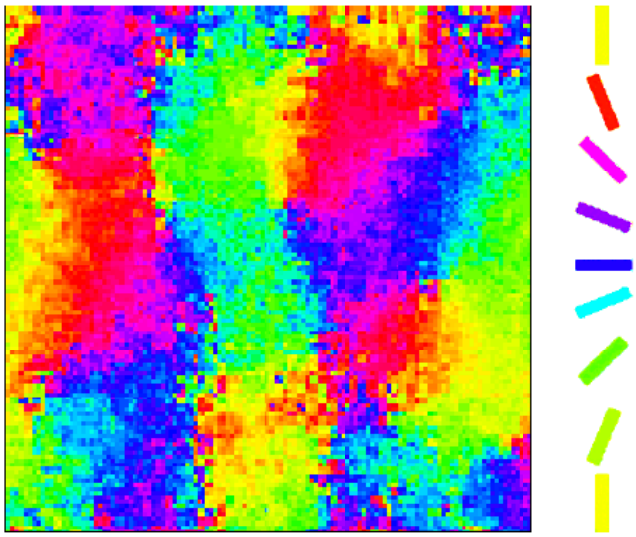


Figure 2.2: Innate orientation map
Orientation map at eye opening in a kitten imaged over a 1.9mm^2 region of cortex [19]. The PO for each pixel is color coded based on the key to the right.

completeness.

2.1.1.2 Maintenance phase

Following the first week of eye opening, there is extensive evidence that shows orientation selectivity can be influenced by visual experience during a critical period (P18–43). We review recent experiments that explore this influence, classified by the type of visual manipulation:

Dark rearing

The animal is raised in complete darkness (until the experiment begins, of course). Crair et al. has shown that prolonged deprivation of vision results in cells becoming progressively less responsive to visual stimuli, as well as a degradation of orientation

selectivity [19].

Lid suturing

The animal's eye lids are surgically shut, thereby removing normal patterned vision; lid suturing differs from dark rearing in that the closed lids allow diffuse light to enter the eyes. Interestingly, this diffuse light causes selectivity to deteriorate at a rate that is drastically faster than dark rearing [20]. These results support the idea that the lack of visual input merely arrests development, whereas abnormal inputs cause cortical circuits to be actively 'miswired' during the maintenance phase [16].

Stripe rearing

The animal is raised in an environment where only a single orientation is present. Sengpiel et al. demonstrated that the overrepresented orientation occupies a larger region of cortex; however, orientations not present in the environment still occupy significant regions of the cortical territory, and all orientations have a similar sharpness of tuning [21]. So although their results confirm that orientation selectivity is largely determined at eye opening, they also suggest that activity can play an *instructive* rather than a permissive role during a critical period, since a portion of cells have modified their preferred orientations based on their inputs. These findings are also consistent with the observation that cardinal orientations, which are more common in nature, are overrepresented in animals that have normal visual experience [22] (but see [23], which suggests that the function of visual experience is to actually equalize innate over-representations).

It is apparent that after the establishment phase, visual inputs can abolish, and in certain cases, modify orientation selectivity. These findings pose an interesting dilemma: Why would the cortex go through all the trouble of creating selective cells, only to discard them in certain situations? It seems like a more efficient strategy would be to create orientation selective cells only after it was evident that such processing were useful (e.g., at least one retina was functioning properly). Perhaps this dilemma is one of the reason why the field has been so reluctant to acknowledge that development has a distinct establishment phase.

2.1.2 Map stability during development

One relevant issue is whether the layout of an orientation map remains constant throughout development (i.e., the map is stable), or if it is constantly being rearranged. That is, should we assign any significance to the initial locations of the preferred orientations, or is the only germane point that a map exists. Naturally, this issue will have profound implications for any model of cortical development.

In 1996, the Stryker lab pioneered a chronic optical imaging technique that allowed them to measure orientation maps in the *same* ferret pups from eye opening through cortical maturation [24]. The main advantage in choosing ferrets as the animal model is that they are born at an earlier stage of cortical development compared to cat (day 42 as opposed to 65), and thus, they offer a more robust preparation at the time when orientation selectivity first develops [25].

In their tour de force experiment, Chapman et al. demonstrated that the detailed struc-

ture of an orientation map is largely constant during development. We reprint an example of their main result (Figure 2.3), which shows both angle and polar maps side by side during map maturation (P31–P42); in the angle map, the hue of each pixel is the preferred orientation, whereas the polar maps include the strength of orientation tuning as the intensity [26]. Even though the selectivities of the early maps (P33–P35) are quite weak (i.e., the polar plots are dark), we can clearly see that the locations, shapes, and colors of the patches in the early angle maps are remarkably similar to the mature map (P42). The map at P31, which was imaged immediately after eye opening, is too noisy (i.e., the optical signal is too weak) to make reliable inferences. However, it seems unlikely that these early maps are substantially different from the mature maps based on supporting single cell measurements.

To quantify the similarity between nascent and mature angle maps, Chapman et al. introduced a measure called the similarity index (SI), which we will use in various forms throughout this dissertation. SI is computed in the following steps: First, the absolute angle difference is computed for each pixel between two maps; because orientation ranges from 0 to 180 degrees, the absolute angle difference is between 0 degrees (same preferred orientation) and 90 degrees (orthogonal preferred orientations). Next, the mean difference is taken over all pixels. Finally, SI is computed by subtracting this mean difference from 90, and normalizing the result to be between 0 and 1 (i.e., divide by 90). A SI that is near 0 indicates that the angle maps are anticorrelated, whereas a value near 1 indicates high correlation; 0.5 signifies no correlation. In the example from Figure 2.3, the SI between an early angle map (P33) and the mature map (P42) is near 0.7, and this value increases with age (presumably due to the maps becoming less noisy).

The fact that an orientation map is relatively consistent is a remarkable result, especially when we consider the massive anatomical reorganization that transpires during the first weeks after eye opening (see Figure 2.4 for a developmental timeline). For one, ocu-

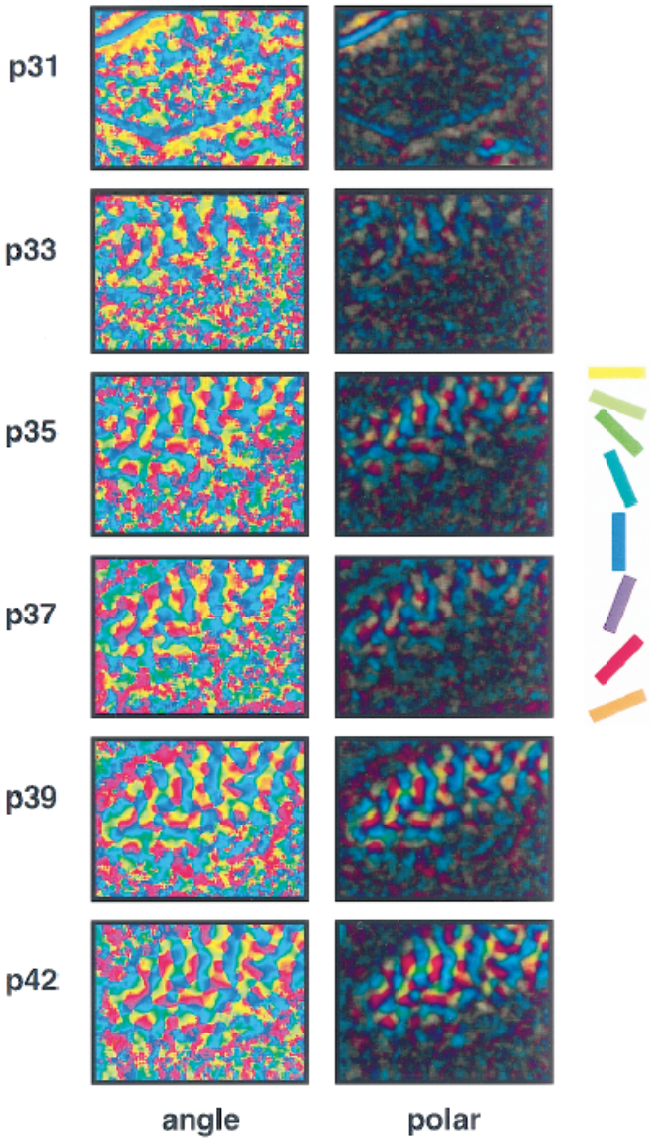


Figure 2.3: Map stability in developing ferret
Imaged orientation maps at different stages of development in the *same* ferret pup; angle maps reveal map structure (left column), and polar plots include orientation tuning strength as color intensity (right column) [24].

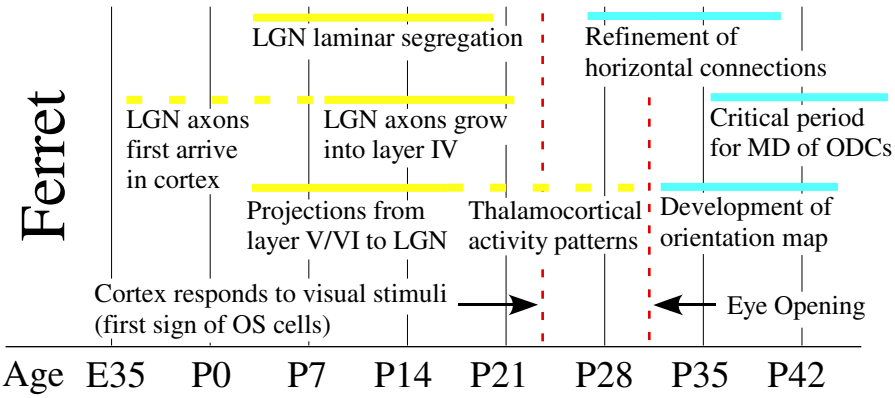


Figure 2.4: Development timeline

Timeline of development in ferret: LGN events (yellow) and cortical events (blue) are indicated; MD and ODC are abbreviations for monocular deprivation and ocular dominance columns, respectively. Note that retinal waves can occur as early as P0 (not shown), and thalamo-cortical activity has been observed (in the LGN) beginning at P24, but may be present earlier.

lar dominance columns are consolidated in this period, with significant changes occurring in the afferent projections to each column [27]; in particular, thalamic projections from the ipsilateral eye segregate into eye specific domains. In addition, long-range horizontal connections (intralaminar) are substantially refined during a similar time frame [20]. Conventional wisdom suggests that the rewiring of feedforward inputs, and/or the refinement of intracortical connections would reorganize the orientation map; it seems however that the opposite case is true — the functional map seems to guide the anatomical reorganization.

2.1.3 Role of activity in map establishment

Although it is clear that visually driven activity does not affect the establishment of an orientation map (Section 2.1.1), it has been proposed that internally generated activity patterns might be involved [28]. For example, a genetically prescribed circuit can generate 'training patterns' that instruct the development of target circuits. In this section, we show that such patterns do in fact exist in the developing cortex, however, these patterns have not yet been

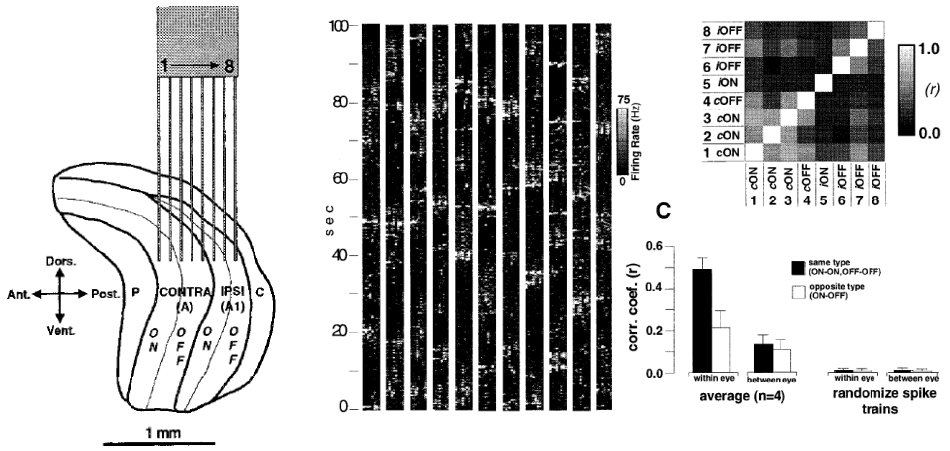


Figure 2.5: Spontaneous LGN activity

A multielectrode array records activity in a developing ferret LGN prior to eye opening *in vivo* (left). Ten successive trials are shown as time series plots (oriented vertically) for a 100 second epoch (middle); the eight columns within each series corresponds to activity at the different electrodes. All pair-wise correlations between the layers are shown in a 8×8 matrix (top right), and summarized for eye and center-type correlations (bottom right) [30].

implicated in instructing map formation; thus, the role of activity is presently unclear.

Two lines of circumstantial evidence support the notion that map development is instructed by activity: First, the early visual pathway exhibits patterns of neural activity around the time that orientation selectivity is thought to originate (one week prior to eye opening) [29, 30]. In particular, both the retina and the thalamo-cortical loop generate spatially and temporally correlated patterns *in vivo* (on their own, and in concert) [31, 30]; we show observed temporal correlations between different LGN layers in a developing ferret (Figure 2.5). Second, computational models have been able to self-organize orientation maps when they are trained on patterns that have center-type (ON–OFF) correlations [32] (we explore these models in Section 2.2.1). It is not yet known if such center-type correlations exist spontaneously (since the retinotopic locations of the LGN layers are difficult to map in experiments like [30]), but it seems possible that center-surround circuitry could coordinate such correlations.

If activity does in fact instruct the development of feature maps, as suggested by a number of groups, it should be possible to disrupt map formation by manipulating the patterns. Weliky directly tested this possibility by artificially stimulating the optic nerve in developing ferrets during P27–P41 (beginning one week prior to eye opening) [30]. Remarkably, neurons in layer IV were unaffected by the artificial correlations and exhibited normal orientation domains and selectivity; neurons in the superficial and deep layers, however, had markedly lower selectivities when compared to control animals. This result strongly suggests that activity-dependent mechanisms do *not* play a significant role in establishing an orientation map (in layer IV); ironically, this paper is often cited as evidence *supporting* activity-instructed development of orientation maps because it was the first to demonstrate that the LGN has eye-specific and center-type correlations.

One possibility is that the artificial stimulation in Weliky's experiment was too intermittent to affect map development in layer IV, since the cuff electrode surrounding the optic nerve was active only 10% of the time. However, this situation is unlikely because the stimulation was able to disrupt the development of superficial and deep layers (i.e., the setup was clearly effective). Another possibility is that the formation of an orientation map occurs prior to P27, and therefore the manipulation was too late to have an effect.

Other attempts to disrupt activity have been difficult to interpret. For example, infusion of tetrodotoxin (TTX) in ferret cortex, which blocks all activity in V1, has been shown to freeze the development of selectivity to an immature state (equivalent to P23) [25]; this demonstrates that a certain level of activity must be present in the cortex for development (permissive role), however, it remains unclear whether the specific activity patterns drive development (instructive role). In a similar experiment, the activity of ON-center ganglion cells in the retina was selectively blocked using 2-amino-4-phosphonobutyric acid (APB), which also resulted in an immature orientation map [33]. This result has been cited as

the "strongest evidence to date" that activity-instructed development plays a role in map formation [32], however, this evidence is not very compelling: First, the experiment still conflates permissive and instructive roles of activity because blocking ON responses may simply reduce activity below a necessary threshold for map development. Second, infusion of APB early in development has resulted in an unresponsive cortex (as opposed to a cortex that only responds to OFF stimuli), which raises the possibility of nonspecific effects. Compelling evidence would need to link the spatio-temporal activity patterns themselves to a predicted (and controllable) change in map formation.

In summary, we have shown that if maps are learned from activity, the process is tightly regulated and the critical period for map specification is early (before P27). However, the lack of evidence for activity-instructed development leaves open the possibility that genetic and molecular factors (which are independent of activity) may be solely responsible for map formation. We describe how orientation maps can develop using activity-independent mechanisms in Section 2.2.2. In fact, our chip model of V1 will obtain its selectivity without the need for activity-instructed learning, verifying that activity is not necessary for map establishment.

2.1.4 Origin of map structure

Despite a wealth of experimental data, we still do not know how an orientation map first becomes forged onto the cortical slate. One promising strategy, which has become quite popular in the past decade, is to explore the role of cellular and molecular mechanisms [8]. The majority of such studies have focused on activity-dependent mechanisms, such as NMDA receptors and neurotrophins — a reflection of the current dogma that map development is instructed by activity. While this mechanism-oriented approach to studying development

may eventually lead to profound insights, orientation maps are seemingly too complex to be reduced to a handful of simple molecular interactions. In fact, recent evidence paints a progressively complicated picture (see [6, 8] for reviews), and it is difficult to see how individual molecular pathways might interact to ultimately determine map structure.

In this section, we take a step back from the low-level molecular mechanisms and explore the broader issue of whether map structure is explicitly encoded in the genes (genetic prespecification), or somehow acquired through a self-organizing process (which is regulated by genes). This issue, which closely parallels the nature–nurture debate, has been repeatedly raised in the literature [18, 8, 32, 34, 35] and may provide insight about the mechanisms that are involved in map formation.

Let us first explore the hypothetical case of genetic prespecification, where an explicit blueprint of an orientation map exists in the genes. For example, consider that every neuron expresses a specific molecular tag (encoded directly in the genes) which 'specifies' its orientation preference. We will argue here, however, that the actual implementation of orientation selectivity (i.e., transforming each tag into a functionally selective neuron) requires multiple well-coordinated events that are not explicitly encoded in the genes — thus the precision of the prespecification is limited by other developmental processes. For example, each neuron must have the following components to achieve selectivity: 1) afferent inputs that maintain local topography (to probe neuron selectivity), 2) appropriate ion channels that allow the neuron to spike (to measure neuron selectivity), and 3) mechanisms that implement orientation selectivity, which would putatively line up thalamic afferents or recurrent connections depending on the tag. Stochastic variations in any of these components could cause deviations between the functional response and the prespecified tag. Our conclusion is that even maps with an explicit genetic origin are in a sense 'acquired' since their expression may be stochastic (due to constraining developmental factors). More

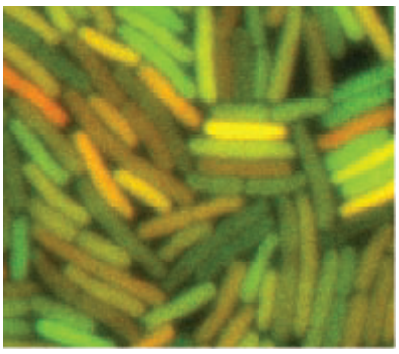


Figure 2.6: Phenotype diversity
Intrinsic noise (or stochastic variations) of green and red fluorescent protein (GFP) reporter gene expression in a population of genetically identical *Escherichia coli*; deviations from equal gene expression are visible as red through yellow to green color variation [36].

fundamentally, recent experiments have shown that gene expression itself can be rather stochastic (see Figure 2.6 for an example), suggesting that an explicit genetic encoding is a straw-man idea.

However, we might imagine that we could still distinguish between genetic prespecification and self-organization through an 'identical twin' experiment, where maps from genetically identical individuals are compared [24]. In other words, if the observed maps in the twins are statistically similar (i.e., on average, particular molecular tags result in the same functional responses) we could conclude genetic prespecification was responsible for map origin, otherwise if the maps are independent we could rule out genetic prespecification altogether. We show in this section that the identical twin experiment can *not* distinguish between these two scenarios; that is, comparing the two maps simply probes how sensitive map formation is to stochastic variations, and not whether it has an explicit genetic origin. The arguments that we present to debunk the identical twin experiment were inspired by the thoughtful discussions from [32].

The fallacy of the identical twin experiment is apparent when we consider that a map may be acquired (or self-organized), and yet its final structure can still be tightly con-

strained (or 'prespecified'). For example, consider the case where a map's structure is largely determined by geometric boundary conditions (such as the size and shape of V1) — here, identical twins could have similar orientation maps simply because their skulls have similar shapes and grow at similar rates! In this case, measuring the similarity between these maps simply probes how sensitive map development is to stochastic variations; that is, how different do the boundary conditions (V1 geometries) have to be before the maps are appreciably different, and what are the nature of bifurcations if they occur? This discussion should make it clear that a positive result of the identical twin experiment can not distinguish between genetic prespecification and a tightly-regulated self-organizing process, whereas a negative result would demonstrate that map structure is entirely influenced by stochastic variations. Therefore, it is not possible to know at what level genes specify an orientation map through an identical twin experiment.

It should now be clear that the debate surrounding the role of genes in map formation is somewhat futile (even though the literature has a peculiar obsession with this debate). Cortical development is a complex dynamic system whose flow is influenced by many interrelated genetic and environmental factors at each stage. Pondering the genetic origin of orientation selectivity is akin to pondering how a list of ingredients determines the taste of a pastry: the individual ingredients are necessary, but how they are brought together ultimately decides its qualities. To help sort through the complexity of development, we will explore a number of computational models in the following section. We pay particular attention to the factors that constrain each model, and question whether these constraints are likely to be present in biology.

2.2 Models of orientation map establishment

Computational models can be useful tools for exploring how cortical networks develop. Because a model is in essence a mathematical description, all of the unstated assumptions must be made explicit for the model to work. This transparency allows each model to be evaluated in an objective manner, and provides an arena where different theories can be tested and refined, which can ultimately lead to new insights. We consider three criteria when judging a model's merit:

1. Reproducibility

First and foremost, the model should be able to reproduce known experimental results. Models that are more abstract will only be able to capture qualitative aspects of orientation maps (e.g., general map structure), whereas more detailed models are further burdened to reproduce quantitative aspects.

2. Verisimilitude

In the context of modeling, verisimilitude describes the extent to which a model captures (or reflects) reality. Verisimilitude goes beyond the notion of simply reproducing known results, and requires that the model's behavior parallels real-world behaviors under similar constraints. Because the factors that constrain development are typically unknown, it is common to instead explore the inherent constraints of the model, and probe whether these constraints are plausible. For example, if map structure is primarily determined by the network's geometric shape, this prediction can be tested experimentally in V1.

3. Elegance

> There is a popular sentiment that elegant models are more likely to be correct. While there is no fundamental reason why this sentiment must be true, models shrouded in superfluous detail are often difficult to evaluate since they can be as confusing as the process that is being studied.

In this section, we describe different models that can explain the establishment of an orientation map. The majority of models that have been proposed are based on the dogma that map structure emerges from activity-instructed learning; these models are often referred to as *correlation-based models* (CBMs) because typically, the learning is driven by correlations between pre- and post-synaptic activity [32]. CBMs have evolved over the years, keeping pace with the latest experimental results, but their basic traits have remained relatively unchanged. In Section 2.2.1, we provide an overview of CBMs and describe some of their more recent incarnations.

Despite the overwhelming popularity of CBMs, it has been proposed that perhaps learning does *not* play an important role in the establishment of orientation maps; this notion is worth considering because the initial appearance of a map remains surprisingly resilient to experimental manipulations. In Section 2.2.2, we describe alternative models of map development that do not require synaptic plasticity. Our main focus is on one particular model that is able to generate an orientation map using a recurrent network with a noisy connection profile.

One trend that should become apparent in this section is that many different models can reproduce realistic orientation maps. So how do we decide which model best reflects cortical development? Choosing an appropriate model is crucial for this project: we are not only interested in the esoteric questions about development, but like the cortex, our primary goal is to construct a large-scale network that obtains it functional architecture in a robust manner. Toward this end, we consider the verisimilitude and elegance of the different models in Section 2.2.3; specifically, we explore their inherent constraints, and how these constraints relate to cortical ones.

2.2.1 Correlation-based models

In 1973, von der Malsburg was the first to demonstrate that an orientation map can self-organize from a few simple ingredients [37]. His model, which was simulated on a 1MHz UNIVAC, is modest by today's standards, but nonetheless the basic ideas are still quite relevant. The network consisted of 169 computational units arranged in a 2D sheet, where each unit represented a column of neurons with similar response properties. Each column received afferent inputs from all points in visual space (i.e., there was no retinotopy), in addition to short-range excitatory and long-range inhibitory inputs from lateral columns. During a training period, the weights of the afferent inputs were modified via a Hebbian learning rule, and the total weight to each column was kept constant through divisive normalization (thus implementing synaptic competition).

Despite the model's simplicity, the network was able to self-organize an orientation map when trained with patterns of oriented bars (Figure 2.7). After training, individual columns were selective to particular orientations, and this selectivity varied smoothly across the network in spoke-like patterns. Astoundingly, the structure of this learned map was strikingly

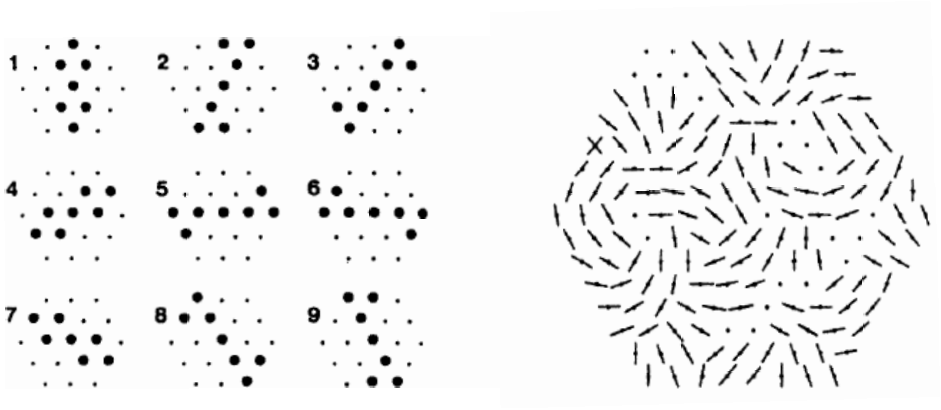


Figure 2.7: Self-organizing orientation map
The different orientation patterns used to train von der Malsburg's network (left), and the resulting orientation preferences after training (right); dots represent cells that do not respond [37].

similar to ones imaged in V1 — 13 years before optical imaging techniques revealed the detailed structure of orientation maps.[2]

The relative ease with which von der Malsburg's model was able to organize a cortical network sparked significant interest in activity-instructed learning. Dozens of similar CBMs have since been proposed (see [34] for a comprehensive review). Miller et al. has noted that practically all of these models share three basic elements [32]:

Synaptic modification

CBMs are all based on some type of Hebbian learning rule [39] where correlations between pre- and post-synaptic activity lead to synaptic modification. The biological mechanisms that are thought to mediate this rule include long-term potentiation (LTP), long-term depression (LTD), and neurotrophic factors (see [8] for an overview of possible mechanisms). Most models ignore the specific details of these mecha-

[2]Prior to optical imaging, Braitenberg and Braintenberg suggested in 1979 that orientation map structure had either a radial or concentric layout based on single unit track recordings [38]. However, their results were far from conclusive, and turned out to be only partially correct.

nisms, and instead use a weight update equation that is computationally efficient.

Competition

Unconstrained synaptic modification can lead to network instabilities in certain situations because the weights can become unbounded. To avoid these unrealistic situations, models typically implement some form of weight normalization (divisive or subtractive) ensuring that synapses compete for efficacy.

Spatial interactions

The hallmark of an orientation map is its smoothly changing selectivity. To reproduce this characteristic structure, CBMs require that nearby cell responses cooperate, and distal cell responses compete (up to a certain distance); these spatial interactions are typically implemented with isotropic short-range excitatory and longer-range inhibitory connections, consistent with connectivity profiles measured in cortical networks [40].

One issue for CBMs is understanding how different training patterns affect map development. For example, is it necessary that the training patterns have well defined edges, similar to the ones that von der Malsburg used, or will other patterns work as well? And perhaps more importantly, what types of patterns can we expect in V1 at the time an orientation map first develops? For one, we already know that the establishment of an orientation map does not require visual experience (see Section 2.1.1.1). This implies that for CBMs to be correct, activity generated *within* the early visual system must be sufficient (e.g., reti-

nal waves and thalamocortical oscillations). In the following section, we describe a CBM that suggests the specific training patterns themselves are unimportant; instead, the only relevant detail is their time-averaged correlations. This model makes specific predictions about the types of correlations that should exist in the developing visual pathway.

2.2.1.1 Correlation-based learning

In 1986, Linsker first proposed the idea that orientation selectivity can develop based entirely of the time-averaged correlation structure of training patterns [41]. Because the details of Linsker's model are quite complex (his model includes seven layers of cells), we describe a more recent version that has been proposed by Miller [42]. In addition to being more concise, Miller's model also adds biological realism by explicitly including ON–OFF -center inputs and eye-specific inputs.

The key idea of Miller's model is that synaptic weight changes are slow compared to the rate at which training patterns are presented; therefore, the time-averaged correlations of the patterns determine the weight changes. Based on this idea, he demonstrated that correlations between units of the same center-type (ON–ON and OFF–OFF inputs) at short retinotopic separations and of opposite center-type (ON–OFF inputs) at moderate retinotopic separations lead to cells with oriented receptive fields (see Figure 2.8).[3] To ensure that nearby cells have similar orientation preferences, this model also included fixed lateral interactions similar to the ones used by von der Malsburg.

Despite being able to generate an orientation map using only center-type correlations, a few of the predictions of Miller's model are inconsistent with biological data. For one, the

[3]Moreover, including binocular correlations can lead to matched orientation maps between the two eyes in addition to ocular dominance columns. In this project, however, we only consider monocular orientation maps.

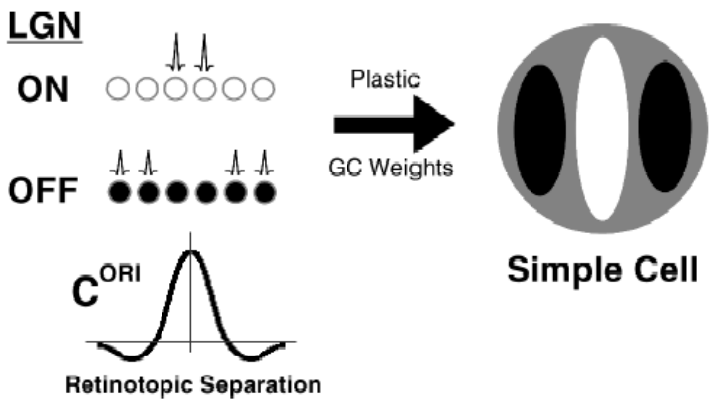


Figure 2.8: Center-type correlations for orientation selectivity
The model proposed by Miller learns an oriented receptive field (right) when the same center-type inputs are coactive with adjacent inputs of the opposite center-type (top left). These statistical correlations are summarized as a function of retinotopic separation (bottom) [32].

power spectra of orientation maps generated by the model are low-pass (i.e., the power is concentrated at the origin), whereas animal maps are distinctly band-pass [34]. Although this difference is difficult to pick out when comparing the maps by eye, it is a rather serious inconsistency: it stems from the fact that orientation domains do not repeat at periodic intervals — suggesting that the model is lacking some crucial element. Another inconsistency is that the subtractive normalization used to constrain synapses in the model is biologically unrealistic because it often leads to saturated synaptic weights [35].

In summary, we have shown that CBMs can indeed account for the emergence of orientation maps using only a few simple ingredients. One key prediction of these models is that characteristic correlations between center-type inputs are required for map development. However, it has not been possible to measure whether these detailed correlation functions are present during cortical development due to experimental limitations.

2.2.2 Alternative models

Throughout this chapter, we have repeatedly suggested that the current dogma in the field, which suggests orientation maps are learned from internally generated activity patterns, might be incorrect. Our discontent with this mainstream view stems from the fact that map establishment remains stubbornly resilient to experimental manipulation. In this section, we explore models that can account for map establishment without resorting to Hebbian mechanisms.

The first model that we explore demonstrates that random sampling of the retinal mosaic can provide the initial orientation bias in V1 (Section 2.2.2.1). We contend that this "haphazard wiring" model is more elegant than CBMs because it can achieve the same result without activity-instructed learning. Next, we explore the possibility that reaction–diffusion interactions can generate periodic patterns, which could serve as a chemical (or molecular) scaffold for a developing orientation map (Section 2.2.2.2). Although pattern formation is an elegant way to organize a system, it is currently not clear how chemical patterns might determine the functional properties of neurons. The final model that we describe is based on this same pattern forming principle, except now, the patterns are generated on the fly in a recurrent network and are represented as electrical (as opposed to chemical) activity (Section 2.2.2.3). This model will serve as the inspiration for the remainder of this thesis.

2.2.2.1 Haphazard wiring

Consider training a CBM many times starting from different random initial conditions, and cataloging all of the afferent weights that lead to an orientation map (after training is com-

plete). We can then ask the following question: What qualities do 'trained' afferents have that others lack? It would quickly become clear that the only difference is that trained ones ensure that nearby cells have similar profiles; the specific afferent weights themselves, however, would be entirely determined by the initial conditions (i.e., they would be stochastic). So in essence, the process of learning can be thought of as randomly assigning cells with selectivity preferences, and then training the network so nearby cells have similar random assignments.

A model proposed by Ringach suggests that we can circumvent this tedious learning process altogether. The basic idea behind his model is that simple cell receptive fields obtain their selectivity by randomly sampling a noisy retinal mosaic (Figure 2.9a); this sampling is akin to randomly assigning cells a particular orientation selectivity. Furthermore, orientation selectivity tends to change smoothly across space (Figure 2.9b) because nearby neurons sample nearby regions of the mosaic (via spatially decaying Gaussian functions that model both the probability of making a connection and a connection's efficacy). In other words, this model replaces the complex process of learning with the simple process of sampling a haphazard arrangement of afferent inputs — a clever and effective shortcut.

Ringach's model offers an elegant strategy for creating an orientation map, however, a number of the model's details are currently inconsistent with data from animal maps. First of all, the orientation domains in his model do not repeat at periodic intervals, which is already apparent in Figure 2.9b; perhaps adding lateral connections would address this issue. Second, this model can only account for the development of an orientation map for one of the eyes (because each retina would have its own unique mosaic). Ringach proposes that the contralateral eye sets the initial map (since these afferents are the first to arrive in the cortex), and the ipsilateral afferents "just go for the ride" (through activity-dependent learning). However, this explanation is inconsistent with reverse-suture experiments where

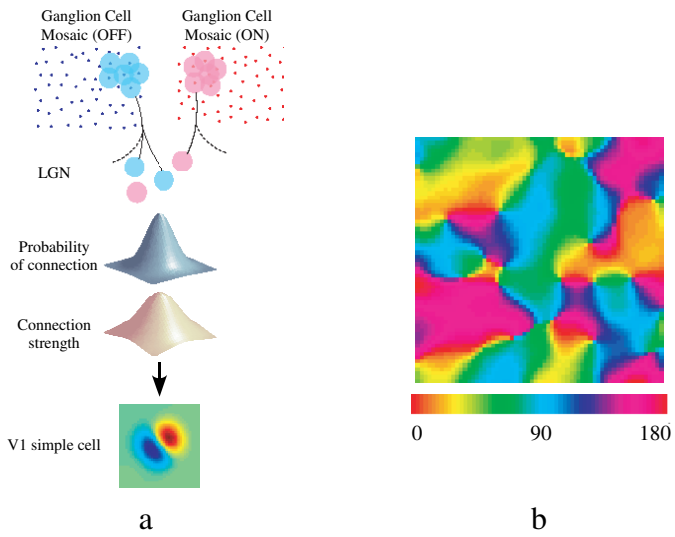
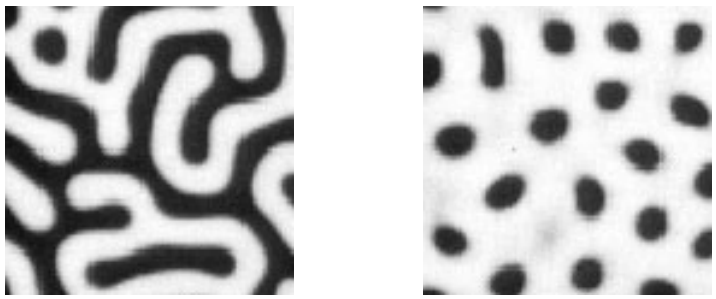


Figure 2.9: Haphazard model of orientation selectivity
(a) Noisy retinal mosaics of ON and OFF center-type ganglion cells project to a layer of LGN cells; RFs are modeled as isotropic two-dimensional Gaussian (shown as blue and red disks). LGN cells connect to cortical targets with a Gaussian probability based on their center locations, and the strength of each connection is also modeled as a Gaussian. The resulting kernel of the cortical RF has distinct subregions and is elongated. (b) The preferred orientation map for the cortical network reveals that selectivity changes smoothly[43].

the two eyes never receive simultaneous inputs and their maps are still in register [44]. Finally, his model suggests that smoothly changing orientation maps should be ubiquitous in visual mammals (i.e., simply from sampling the retinal mosaic). However, it is known that squirrels, rats, and mice, which all rely heavily on their visual sense, do not have smoothly changing orientation maps.

2.2.2.2 Pattern formation using morphogens

Alan Turing first proposed in 1952 that simple reaction–diffusion models can create complex patterns that bear resemblance to ones found in biology [45]. The equations that he developed have been used to replicate the spots on a leopard's coat, the stripes on a zebra's hide [46], and can even account for how these patterns change as an animal ages [47]. Ac-



a b

Figure 2.10: Chemical pattern formation
Reaction–diffusion equations can lead to (a) striped patterns that closely resemble ocular dominance columns (originally discovered by Meinhardt [49]), and (b) spot patterns that resemble a preference map to a particular orientation [46].

cording to Turing, these elaborate patterns are the result of interacting chemicals, which he called *morphogens*. For example, one type of morphogen could result in the darkening of hair (called an activator), whereas another type could bleach it (called an inhibitor); as these morphogens diffuse across the animal's coat, typically at different rates, their reactions with each other can create intricate pigmentation patterns.

One intriguing possibility, which has surfaced in the literature in recent years, is that the functional architecture of the cortex could be specified through a similar reaction–diffusion process [30, 48]. For example, the patterns in Figure 2.10a show a striking resemblance to ocular dominance stripes. A similar idea is that the spot-like patterns in Figure 2.10b could lead to an orientation map; that is, one configuration of spots could correspond to a particular orientation preference, and shifted spot patterns could correspond to different preferences. This idea would explain 1) why orientation maps are smooth (because the spots have spatial extent), and 2) why orientation domains repeat at regular intervals (because the spots pattern is periodic) — both features being natural consequences of reaction–diffusion interactions.

The notion that a reaction–diffusion process creates a scaffold during cortical develop-

ment is a rather compelling idea: For one, it taps into an extremely efficient design strategy; all that is required to create the scaffold is the production of particular chemicals at the appropriate developmental stages. Second, these chemical patterns (or their imprints) can provide an immutable scaffold on which other features are organized, which could account for the remarkable resilience of orientation maps to manipulations. Finally, the reaction–diffusion process can unfold in the presence of many environmental constraints (different size and shape V1 regions), and is therefore robust. Computer scientists, for example, have recently used reaction–diffusion models to add texture to complex meshes without resorting to sophisticated algorithms [46]. It is possible that evolution has been using this same strategy to 'texture' the crumpled cortical sheet for a similar reason.

Despite the aforementioned benefits of using a reaction–diffusion process, there has been no experimental evidence to date that suggests morphogens play a role in cortical development — in fact, the mere existence of morphogens during development is still controversial (although some evidence has recently emerged [50]). In addition, it is not clear how morphogens might determine the functional properties of a neuron. One possibility is that certain morphogens influence cell adhesion such that LGN afferents preferentially connect along a particular axis. Another possibility is that NMDA receptors (or neurotrophic factors) are influenced by morphogen concentrations, ultimately leading to functional differences among neurons; this scenario would make it extremely difficult to untangle the individual contributions of activity-independent and dependent mechanisms. Currently, these ideas are speculative and would require experimental evidence to gain credibility.

2.2.2.3 Pattern formation using neural activity

Thus far, we have shown that simple reaction–diffusion models can generate two-dimensional patterns that look remarkably similar to V1 selectivity maps (eye and orientation preference maps). This similarity, however, is deceptive: the patterns from reaction–diffusion models correspond to chemical concentrations, whereas V1 maps correspond to neuron selectivities. And while it is certainly possible that chemical concentrations could be transformed into functional selectivities (see the previous section for possibilities), there is currently no experimental evidence to support this idea. What if we could cut out this middleman entirely (the chemicals) and generate the patterns directly using neural activity?

In 2001, Ernst et al. proposed a model of V1 that cleverly bridged the gap between pattern formation and orientation selectivity without using chemical interactions [51].[4] Their model substituted activator–inhibitor chemicals with excitatory–inhibitory synaptic connections, and implemented spatial interactions (loosely analogous to chemical diffusion) using Mexican-hat connectivity (local excitatory and distal inhibitory connections implemented as a summed analog connection) (Figure 2.11a). The result is a two-dimensional one-layer recurrent network that can form patches (or bumps) of neural activity when the feedback is sufficiently strong (Figure 2.11b); not surprisingly, these bumps closely resemble the spot-like patterns generated by a simple reaction–diffusion model (Figure 2.10b). Similar to the other V1 models described in this section, the units in this network represent cortical columns, and receive retinotopic afferent inputs (this model only considers a single luminance channel). The inputs in this model, however, are rotationally symmetric and remain fixed throughout the simulation (i.e., there is no feedforward orientation bias).

[4] We note that Ernst et al. did not frame their model in the context of reaction–diffusion models and chemicals; however, we feel that their ideas are best understood as an extension of Turing's original proposal.

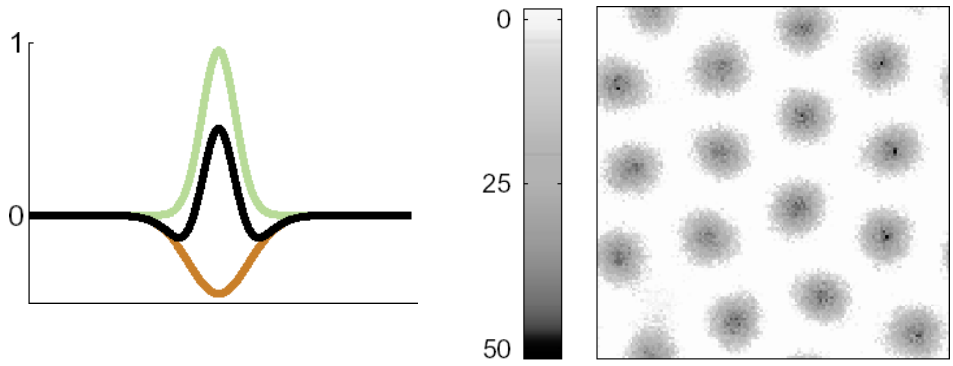


Figure 2.11: Connection profile and activity pattern
Recurrent network proposed by Ernst et al. (a) Normalized connection strength between columns as a function of inter-columnar distance; green, red, and black correspond to excitatory, inhibitory, and summed strengths, respectively. (b) Patterns of neural activity form that resemble the spot-like patterns of reaction–diffusion equations [51].

Now that we can generate spot-like patterns directly using neural activity, the question that remains is how do these patterns determine preferred orientation maps? Consider that there are two requirements to achieve orientation selectivity from spot-like patterns: First, different oriented inputs must elicit different configurations of spots. And second, repeated presentations of the same inputs must yield roughly the same configurations (regardless of the initial state of the network). From the point of view of a single column, these two requirements ensure that its tuning curve is not flat (i.e., different inputs elicit different responses), and its shape is statistically significant (i.e., the tuning curve shape is repeatable for independent trials).

Suppose for the moment that the recurrent network is built to perfection; that is, the properties of each cortical column are exactly matched (thresholds, time constants, and synaptic strengths), and the connections between the columns are also identical as well as symmetric. In other words, there is not even the slightest preexisting bias for activity patterns to favor one location over another. For spot patterns to form (even transiently),

there must be winners (regions of spots) and losers (inhibited regions); in this egalitarian network, however, no region is willing to concede defeat, which leads to the trivial result that no patterns can form. This scenario where no spatial patterns form is quite fragile — even the slightest favoritism will break the symmetry and cause the entire network to choose regions of winners and losers.

Probing the responses of this 'perfect' network to oriented inputs reveals that columns have non-flat tuning curves for individual trials. Before each trial, we impose favoritism by starting the network off with a randomly chosen set of initial conditions; it quickly evolves into a random pattern of bumps. These non-flat tuning curves are simply the consequence of bumps interacting with the oriented inputs, preferentially activating particular regions. However, the shapes of these curves will *not* be consistent from trial to trial. Two factors contribute to the inconsistency: First, the preliminary configuration of bumps is entirely determined by the random initial conditions. Second, the movements of individual bumps are unbiased (i.e., they have an equal likelihood of moving in any direction); this results in the uninteresting case where columns are not really 'selective', but become active simply by chance occurrences.[5] Both of these factors are a direct consequence of the fact that there are no innate biases within the network due to its immaculate perfection.

How can we achieve robust selectivity so that columns respond in a somewhat consistent manner? One possibility is that the intracortical connections could provide an innate bias which constrains network activity. For example, bumps could be 'tethered' thereby confining them to certain regions regardless of the initial conditions; in addition, these pinned bumps might prefer to move in particular directions, resulting in consistent inter-

[5]Note that we are not suggesting bumps are unconstrained and simply follow their inputs. On the contrary, bumps motion is tightly constrained due to interactions with neighboring bumps (assuming the recurrent feedback is strong enough); these interactions, however, will not have any inherent bias that is anything but stochastic.

actions with different orientations. Resisting the temptation to bestow these biases via learning synapses (or systematic anisotropies as in [52]), Ernst et al. shrewdly added random jitter to the strengths of the recurrent connections. And why not — it seems rather foolish to suppose that these connection strengths will be matched in practice.

To test their idea that imperfections can lead to repeatable preferred orientation maps, Ersnt et al. simulated a 128×128 recurrent network with noisy Mexican hat connections. They presented sixteen sinusoidal gratings to the network, each with a different direction (spanning the full circle), and computed the network responses to the eight different orientations by averaging the responses to opposite directions (Figure 2.12a). And *voila*, different oriented inputs led to unique configurations of spot-like patterns and these patterns were repeatable (not shown); the resulting preferred orientation map is shown in Figure 2.12b. Similar to animal maps, this map has smoothly changing orientation selectivity, and selectivities repeat at regular intervals; as we described earlier in Section 2.2.2.2, these characteristics are a direct result of the spot-like patterns.

2.2.3 Physical constraints

The models that we have described in this section demonstrate that it is quite easy to generate an orientation map — at least in the context of software simulations. Map structure can arise from: 1) activity-instructed learning using Hebbian synapses, 2) haphazard sampling of a noisy retinal mosaic, or 3) chemical/electrical pattern formation. But which of these scenarios is best suited for physical implementation? To help answer this question, we explore some of the factors that may constrain orientation map establishment in the developing cortex; our focus is on component variability, which we believe is likely to be an important constraint. Based on our discussion, we conclude that CBMs would not fare

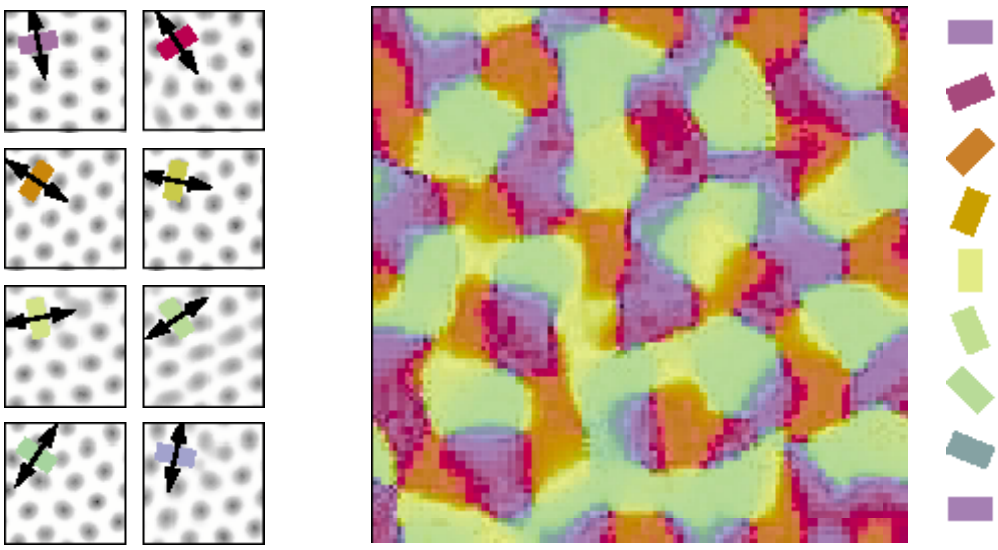


Figure 2.12: Orientation selectivity using neural pattern formation
(a) A network with noisy recurrent connections generates unique patterns based on the eight different input orientations. (b) The corresponding angle map of prefered orientations [51].

well in a large-scale physical system and rule them out for our chip model. Models that embrace component variability, on the other hand, seem like more plausible candidates.

Orientation map development unfolds in the presence of constraints. Certain factors in the cortex may only slightly constrain map development, whereas others may be highly influential. The extent to which different factors affect development are currently unknown, however, we believe that they hold the key to understanding how and why the cortex develops the way it does. And more practically, these constraints can be used to evaluate the verisimilitude of different cortical models. Although difficult to quantify, we suggest that variability is likely to be a crucial constraint throughout the developmental process.

The cortex likely faces similar design challenges as silicon chips, although synaptic density is currently a factor of 100 higher than transistor density (i.e., there are a 100 million synapses per cubic millimeter of cortex compared to 1 million transistors per square millimeter of silicon). One of the biggest challenges facing modern day chip designers is

component variability [53]. As more transistors are crammed into smaller spaces, device characteristics become extremely difficult to control. However, whereas chip designers use abstraction to hide component variability, it is evident at many different scales in biology: At the most basic level, gene expression is itself stochastic (see Figure 2.6). Furthermore, this stochastic scaffold serves as the basis for more complex structures, which ultimately leads to fully organized cortical networks. Even though it is currently unknown the extent that this variability affects development at different stages, it is incontrovertibly pervasive.

The question that we pose for CBMs is whether realistic sources of variability would 'swamp out' the delicate process of learning?[6] Because software models exist in a rather artificial world, their 'virtual' components typically have identical properties. What if: 1) inputs are randomly drawn from a noisy retinal mosaic (as suggested by Ringach), 2) neurons have spontaneous firing rates that vary by a few orders of magnitude, and 3) the Hebbian mechanisms themselves are variable across neurons? We suggest that in these situations, CBMs would essentially learn these inherent biases, and map structure would *not* be instructed by the training patterns. If this is the case, it would seem rather superfluous for the cortex to resort to Hebbian learning when simpler mechanisms can achieve a better outcome.

From our perspective, its seems as if CBMs have conveniently left out a number of likely constraints that might exist in the developing cortex. While these constraints could be a bane for map formation, we argue that they could be a boon by providing a deterministic prepattern for the final map — thereby rendering activity-instructed learning unnecessary. In our opinion, this prepattern would severely compromise the idea that map structure is specified from genetically prespecified training patterns.

[6] To our knowledge, such tests have not yet been performed.

The models proposed by Ringach and Ernst et al., on the other hand, are unique in that they embrace component variability. Unlike CBMs, which can endow a perfectly matched network with functional properties, these models utilize the innate network biases as their scaffold. They boldly raise the question: Why waste important resources to learn a map that is already present?

Chapter 3

Neuromorphic bump model

3.1 Pattern formation in V1

Our goal is to construct a neuromorphic model of V1 that can account for the ontogenesis of an orientation map. But which of the models surveyed from the last chapter should we base our chip design on? In our opinion, the most convincing model to date is the one proposed by Ernst et al. from Section 2.2.2.3 (pattern formation using neural activity): not only is it able to reproduce realistic looking orientation maps using nothing more than a noisy recurrent network, but it is also ripe for implementation. We refer to this model (and its derivatives) as the 'bump model'.

We focus on the bump model for two reasons: First, it does not rely on learning mechanisms for map establishment, which can explain why orientation selectivity remains remarkably resilient throughout development.[1] And second, the bump model utilizes innate

[1] Ringach's model also does not require learning to establish a map — but only for a single eye. To align the map between the two eyes in his model, learning is required.

variability to determine map structure; this opportunism is relevant for us since component variability is pervasive in silicon chips. In contrast, other developmental models fight an uphill battle to endow cells with selectivities: they must either tolerate, or work against the innate biases of the physical system [54].

Despite the admirable qualities of the bump model, the parallels between this model and the primary visual cortex are still unconvincing. For one, the bump model is built around a one-layer recurrent network that collapses excitatory–inhibitory connections into a single analog interaction (for computational simplicity); this simplification, however, is biologically unrealistic because excitatory–inhibitory interactions require distinct neuron types. This detail is more than academic: we will show in Chapter 6 that a two-layer excitatory–inhibitory recurrent network leads to complex and counterintuitive dynamics that are *not* predicted by the simpler one-layer network — and these differences significantly affect map formation. Another issue is that neural dynamics of a column are reduced to rate-based equations (again for computational simplicity). But this simplification ignores the rich temporal dynamics of spiking neurons, which can affect pattern formation in a recurrent network; for example, Laing and Chow demonstrated that a bump of neural activity can lose stability when activity becomes partially synchronized [55].

The primary criticism of the bump model is that the input consists of luminance channel signals — it is not obvious how to replace them with ON and OFF rectified channels. For example, consider for the moment that ON and OFF afferent pathways terminate in distinct sublayers in V1, each within their own recurrent network. In this case, the two networks would have their own innate (noise) connection profiles and the bump patterns would form at independent locations (for the same input orientation). Consequently, there is no guarantee that ON-driven maps would line up with OFF-driven maps, which could result in conflicting orientation signals at the same retinotopic locations (Figure 3.1); these

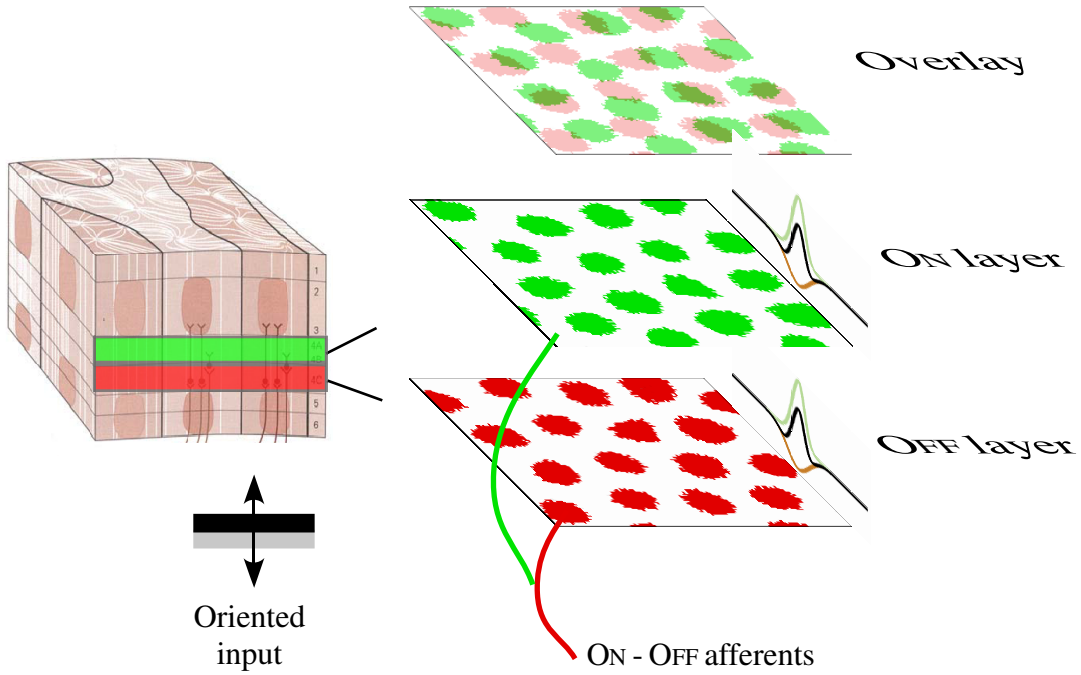


Figure 3.1: Hypothetical ON–OFF segregation

Responses of two recurrent networks to the same oriented input, where each network has its own noisy connection profile (shown to the right of each network). Each network is driven by either ON or OFF afferents (green or red, respectively) that terminate in different cortical layers at the same retinotopic positions. Because bump patterns form at independent locations, the responses of the networks do not match when they are overlayed.

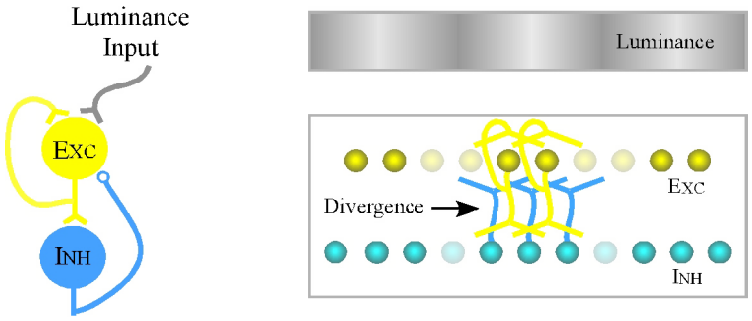


Figure 3.2: Two layer recurrent network
Analog interactions from the Ernst et al. model are replaced by separate EXC (excitatory) and INH (inhibitory) layers: circles represent neurons, curved lines represent axon arbors that end in excitatory synapses (v shape) or inhibitory synapses (open circles). In this two layer network, EXC–INH cells make reciprocal connections, and EXC cells are recurrently connected (left). Connections between neighboring cells (divergence) are also shown to the right. In this example, bumps form (highlighted cells) in regions when the input luminance is high.

conflicting orientation signals would be inconsistent with the basic principle of columnar organization. Our hypothetical example illustrates the complications that crop up when including realistic inputs to the bump model; it is important to note, however, that ON and OFF pathways do *not* segregate into separate cortical lamina.

We make three modifications to the bump model to address these aforementioned issues. The first two modifications are straightforward and are achieved by: 1) expanding the one-layer recurrent network into separate excitatory and inhibitory layers (Figure 3.2); and 2) replacing the rate-based equations with integrate-and-fire (spiking) neurons (described in Section 3.2). Our third modification, which is more complicated, is to replace the luminance channel with ON and OFF rectified channels; we explain this innovation in the following section.

3.1.1 Complex cells solve a complex problem

To our knowledge, orientation maps observed in the cortex are sign-independent; that is, their layout does not change depending on whether light (ON-center) or dark (OFF-center) bars are used as stimuli. In fact, if orientation selectivity did shift based on input sign, the idea of a cortical columnar architecture would be severely compromised. The question that we explore here is how do we incorporate ON and OFF channels in the bump model (which currently has a single luminance channel) to create a sign-independent orientation map? As we described in our previous example, using two parallel recurrent networks will not work because these networks will have their own distinct maps due to their unique connection profiles (see Figure 3.1).

To help us with our task, recall that the orientation map is seeded by innate network biases. In other words, the neural patterns (bumps) interact with stimuli the same way each time because they experience the same innate biases. Based on this insight, we can also ensure that neural patterns interact with the ON and OFF channels the same way each time by subjecting each channel to the same innate biases. We accomplish this by computing both ON and OFF -driven maps in a single recurrent network. With our modification, each EXC cell is driven by both ON and OFF channels at each retinotopic locations (shown in the top half of Figure 3.3a). The result is that bumps interact with stimuli in a repeatable manner, regardless if the stimuli is made up of light or dark bars. Consequently, the orientation map generated by this network will be sign-independent, consistent with orientation maps observed in V1.

The key to generating a sign-independent orientation map in the bump model is treating both channels the same way. However, combining ON–OFF inputs throws away information about contrast sign. For example, the response of an EXC cell to an oriented bar would not

reveal whether the bar was light or dark. We recover the sign by creating two parallel read-out networks, one for ON cells and one for OFF cells (shown in the bottom half of Figure 3.3a); it is important to note that these networks do not affect pattern formation in the EXC–INH recurrent network. These read-out cells derive their responses from two separate signals: an orientation-selective signal from the EXC–INH network that indicates the presence of a bump (independent of sign), and an ON–OFF selection signal that chooses the appropriate response sign. The flow of computation is depicted in Figure 3.3b.

The astute reader may have noticed that the different cells we use in our bump model (sign-dependent and sign-independent) bear a resemblance to the simple and complex cell classifications first described by Hubel and Wiesel. In particular, Hubel and Wiesel classified cells with well-defined ON–OFF subregions as simple (similar to our ON–OFF cells that receive push-pull input), whereas cells with no identifiable subregions were classified as complex (similar to our EXC–INH cells that are excited by both channels). The terms 'simple' and 'complex' reflect a feedforward model of visual processing — stemming from the idea that simple cells first obtain their selectivity from aligned thalamic cells, and the outputs of simple cells combine to create complex cells [56]. In this feedforward view, complex cells are higher in the hierarchy of visual processing because their computation (i.e., being selective to an oriented bar independent of the bar's details) is ostensibly more complex. In our model, however, 'complex' cells have an entirely different role: their purpose is to ensure orientation maps are sign-independent (i.e., patterns that form are subject to the same noise profile regardless of sign). The purpose of 'simple' cells in our model is to recover the sign information that is lost in the orientation computation.

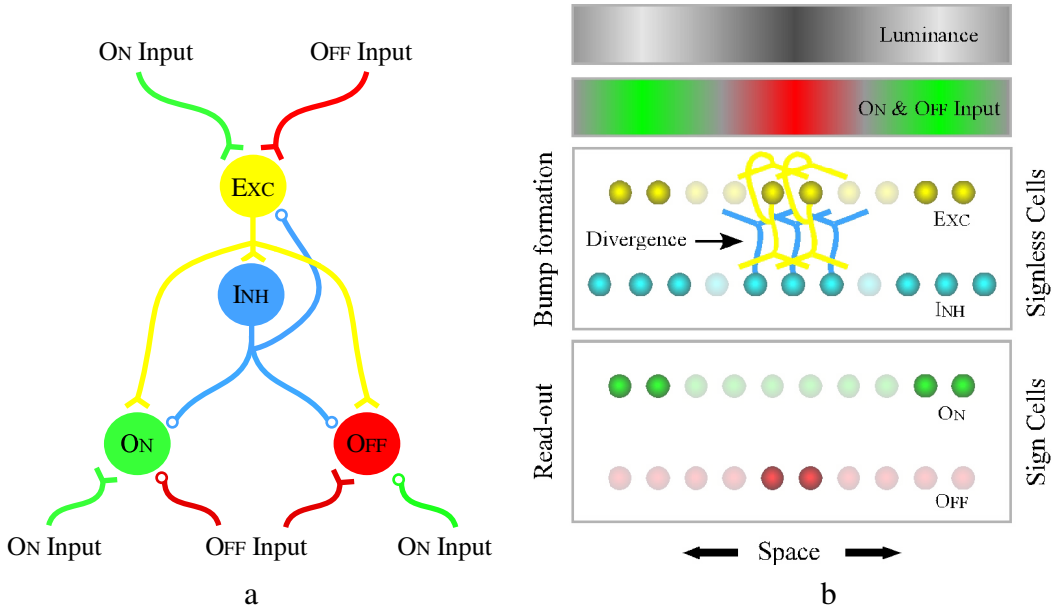


Figure 3.3: Microcircuit

(a) The connections for the modified bump model with sign-independent cells (EXC–INH), and sign-dependent cells (ON–OFF). (b) A sinusoidal input causes bumps to form (highlighted cells) in regions where the luminance is both light and dark. These sign-independent bumps seed bumps in the ON–OFF read-out layers; bumps that match the sign of the input survive, whereas other bumps are extinguished (faded cells) via push-pull inhibition.

3.2 Neuromorphic models

Exploring a detailed large-scale recurrent network is a computationally daunting task that is poorly suited for software modeling. Previous attempts to model such networks in software have sacrificed low-level details that are known to affect neural dynamics (e.g., they simplify spiking with rate-based equations) [55], and have explored only a handful of network states [51, 57]. For these reasons, we have chosen to build our network in silicon using a neuromorphic implementation [1]. This approach has the advantage of: 1) operating in real-time; 2) retaining important low-level details (i.e., current flowing through MOS transistors are analogous to ionic currents in a biological membrane); and 3) supporting large-scale networks ($\approx 10,000$ neurons).

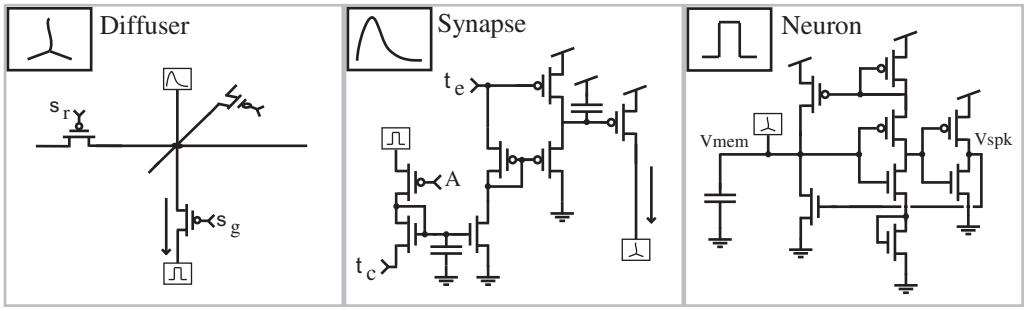


Figure 3.4: Circuit blocks
Transistor implementations for a diffuser, synapse, and neuron are shown along with their input–output interactions.

3.2.1 Neuromorphic building blocks

We build our neuromorphic bump model using three canonical transistor circuits, which are designed to capture the behavior of common neural components (Figure 3.4). The circuits that we describe have been used before in previous neuromorphic systems; the novelty is how we will bring them together in Section 3.2.2 to build our unique microcircuit.

These circuits perform the following functions:

- A **diffuser**, which models axonal arbors, spreads synaptic current to local regions in space with exponential decay [58]. Analogous to resistive divider networks, diffusers are able to efficiently distribute currents to multiple targets. Each diffuser node connects to its six neighbors (although only four connections are shown for simplicity) through transistors that have a pseudo-conductance set by sr, and to its target site through a pseudo-conductance set by sg; the space-constant of the synaptic decay is set by relative levels of sr and sg.

- A **synapse** converts spikes into postsynaptic currents with biologically realistic temporal dynamics. Our synapse achieves this by cascading a current-mirror integrator

circuit [59] with a log-domain low-pass filter. The current-mirror integrator has an impulse response that is inversely proportional to time (the decay rate is set by the voltage tc). This time-extended current pulse is then fed into a log-domain low-pass filter (equivalent to a current-domain resistor-capacitor circuit), which adds a rise time and exponential decay set by the circuit's time constant determined by te.

- A **neuron** integrates excitatory and inhibitory currents on its membrane capacitor and generates a spike (brief voltage pulse) after reaching threshold. Diffusers either inject current (excitatory), or siphon off current (inhibitory) through a current-mirror. Spikes are generated by an inverter with positive feedback (modified from [60]), and the membrane is subsequently reset by the spike signal.

3.2.2 Neuromorphic microcircuit

Based on the diffuser, synapse, and neuron circuit blocks presented in the previous section, we are now able to piece together the bump model's microcircuit (Figure 3.3). For visual clarity, we divide the microcircuit into three sections and label the connections between them: the first section is for the input stage (top panel), the second is for the recurrent network (middle panel), and the third is for the sign read-out network (bottom panel). Each panel depicts two adjacent nodes along with their neighboring connections.

One way to implement EXC–EXC recurrent connections is to tap off of the EXC–INH diffuser network (the $\mathrm{E}+$ signal), and feed it back directly into EXC cells; note that the feedback signal ($\mathrm{E}+$) has a controllable gain factor indicated by the $\times$ bubble. It turns out, however, that this solution is problematic. The crux of the problem is an autapse (self-connection) is introduced that is weighted by the peak of the exponential. In practice, this

autapse resulted in extreme positive feedback and limited the network's operational range. An alternative solution is to re-route EXC spikes sent off chip back into the chip as synaptic events, targeted to neighboring neurons; we describe this digital fanout in more detail in Section 4.1.1. Direct self-excitation is still present through the input diffusers, but the strength of the autapse is reduced. In this case the gain factor is set to 0 thereby eliminating on-chip recurrent feedback.

One unintentional consequence of implementing feedback through off-chip connections is that the rerouted EXC spikes must enter the chip via the ON–OFF afferent pathway. This afferent pathway, however, is prewired to implement a push-pull interaction between ON and OFF channels, which essentially causes the recurrent EXC spikes to cancel themselves out in the read-out network. The result is that these cells are mostly inactive, rendering the read-out network somewhat unusable. For these reasons, we do not explore the responses of the read-out network in any detail in this project. Although unfortunate, eliminating the read-out network is tolerable because it does not affect the pattern formation in the recurrent network in any way.

3.3 Bump chip implementation

We fabricated our bump model in silicon using the TSMC (Taiwan Semiconductor Manufacturing Company) 0.25μm 5-metal layer CMOS process. The final chip, which we call the bump chip, consists of a two-dimensional core of 48×48 pixels (microcircuits) and is surrounded by asynchronous digital circuitry that transmits and receives spikes in real-time; we describe the off-chip communication links in Section 4.1.1. Our design packs nearly 460,000 transistors in 10 mm^2 of silicon area and has 9,216 neurons — the largest

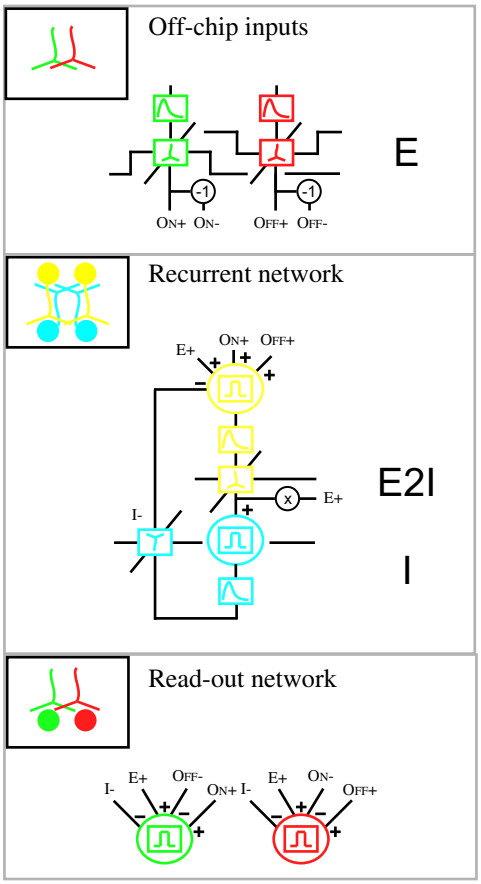


Figure 3.5: Silicon microcircuit

Bump microcircuit built from diffuser, synapse, and neuron circuits; we also use two bubble operators which invert the sign of the current (-1) or multiply it by a gain factor ($\times$). Circuit parameters for each circuit block (defined in Figure 3.4) are referred to as $\mathsf{type}_{\mathsf{par}}$, where type is shown in the right of the network block, and par is labeled in the circuits (e.g., the inhibitory cell synapse parameter A is referenced as I_{A}). The network parameters (in volts) that we use throughout this project (except where noted) are: for INH, $\mathsf{I}_{\mathsf{sr}} = 1.844$, $\mathsf{I}_{\mathsf{sg}} = 1.897$, $\mathsf{I}_{\mathsf{te}} = 2.374$, $\mathsf{I}_{\mathsf{A}} = 1.45$, $\mathsf{I}_{\mathsf{tc}} = 0.061$; for EXC, $\mathsf{E}_{\mathsf{sr}} = 2.046$, $\mathsf{E}_{\mathsf{sg}} = 2.045$, $\mathsf{E}_{\mathsf{te}} = 2.4$, $\mathsf{E}_{\mathsf{A}} = 1.855$, $\mathsf{E}_{\mathsf{tc}} = 0.052$; for their interaction, $\mathsf{E2I}_{\mathsf{sr}} = 2.051$, $\mathsf{E2I}_{\mathsf{sg}} = 2.098$, $\mathsf{E2I}_{\mathsf{te}} = 2.387$, $\mathsf{E2I}_{\mathsf{A}} = 1.602$, $\mathsf{E2I}_{\mathsf{tc}} = 0.05$, and $\mathsf{Vdd} = 2.5$.

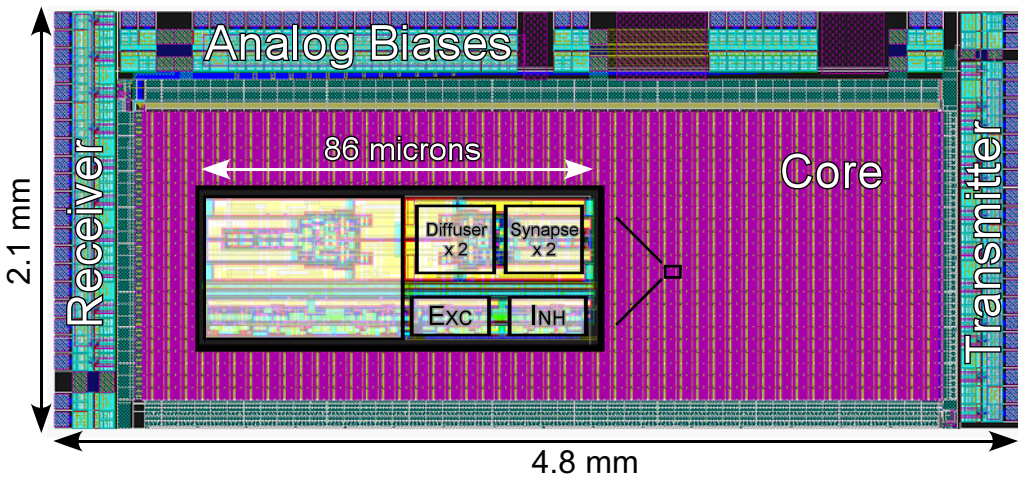


Figure 3.6: Bump chip layout

number of mixed analog–digital neural circuits fabricated to date.

We transformed our transistor circuits into a full-blown chip in two steps: First, we used a CAD program to draw physical transistors and their connections based on the circuit schematics (a process called 'layout'). In our case, we transcribed the microcircuit from Figure 3.5 since this circuit represented the smallest repeatable unit (or pixel) in our network. The layout for our pixel, along with its dimensions, is shown as an inset in Figure 3.6.

The second step was to compile the chip, which involves 1) tiling the pixel into a two-dimensional core, and 2) building transceiver circuitry around the core to handle off-chip spike communication. Luckily, this step was automated using custom place-and-route software called ChipGen (currently maintained by our lab) [61]. The layout for the fully compiled bump chip, along with pads, is shown in Figure 3.6. Notice that the chip has more than a 2:1 aspect ratio; this non-standard shape was chosen so all 23 analog biases would fit along the top edge, thereby eliminating the need for pads on the bottom. Without these bottom pads, we were able to increase the size of our core by nearly 30%!

Chapter 4

Orientation maps *in silico*

In this chapter, we present results from our chip model of V1.

We begin with an overview of our custom hardware–software setup; this setup allows us to simultaneously present stimulus patterns to our chip while recording the network activity, all in real-time (Section 4.1). Next, we show that the chip responds selectivity to oriented gratings, and this selectivity is repeatable for a majority of the cells; this result confirms that innate transistor mismatch can sufficiently constrain network activity so interactions with different orientations are consistent across trials (Section 4.2). Finally, we quantify the selectivity employing two different metrics: normalized vector magnitude and information (Section 4.3).

4.1 Chip testing

Our interest in the establishment of cortical maps has led to the creation of the bump chip — a large-scale model of V1 implemented in silicon. Before we explore how the bump chip responds to different input patterns, which is the focus of the rest of this chapter, we first describe our experimental setup. In our experience, the setup is just as important as the chip model itself: a well-designed setup can provide the flexibility of a software model in a real-time setting.

Past neuromorphic setups were built in a rather *ad hoc* manner, often using sophisticated (and expensive) equipment originally intended for debugging microprocessors and *not* for studying neural systems. While these setups suffice for small-scale neuron chips, they do not perform well for larger systems. This is certainly the case for us: For one, the bump chip is the largest collection of mixed-analog–digital neural circuits that has been fabricated to date, and accordingly, it has significant bandwidth requirements. Moreover, to explore the bump chip's large parameter space, we need to visualize network activity (e.g., selectivity maps, spike rasters, etc.) while interactively changing the parameters; features not offered by past setups. In this section, we present our particular experimental setup that was built to meet our requirements.

4.1.1 Setup

Our setup must perform the following tasks: 1) stimulate the chip's synapses with arbitrary patterns, 2) record the chip's activity (at multiple time resolutions), 3) visualize network activity (both processed and unprocessed), and 4) implement recurrent excitatory connections (fanout). Here, we describe how we implement these features. We start at the

lowest-level, describing how spikes are transferred to and from our chip. Then we move to the board-level, where spike events are processed and routed to their appropriate locations; this board-level spike processing is handled by an off-the-shelf reconfigurable chip (programmable logic). Finally, we present the back-end of the system, built around a graphical user interface (GUI) that can run on most modern PCs; the GUI communicates with our board through a standard USB2.0 interface.

4.1.1.1 Chip communication

Our chip has two independent links: a transmitter link that implements axonal projections *from* our neurons, and a receiver link that implements projections *to* the neurons' synapses. The links use an *address-event representation* (AER) [62], which specifies the source–destination of each projection with a unique row and column address.[1] Signaling an axon event (i.e., a spike) takes an address-event link 40 nanoseconds, which corresponds to a max rate of 25 million events/sec; therefore, an AER link can faithfully model hundreds of thousands of axons in real-time — ensuring that events will reach their destination within 1ms (the timing precision of a biological axon).

4.1.1.2 Board-level spike processing

At the board-level, the transmitter and receiver AER links of the bump chip connect to a field programmable gate array (FPGA) chip, which in our case is programmed to process spike events (Figure 4.1); an FPGA is a large collection (hundreds of thousands) of programmable devices that can be configured offline to implement arbitrary logic functions.

[1] The specific protocol of AER has evolved significantly since its original conception; the bump chip, for example, can send a burst of events that originate within the same row using a word-serial protocol. For specifics on the latest standards, please refer to [63, 64].

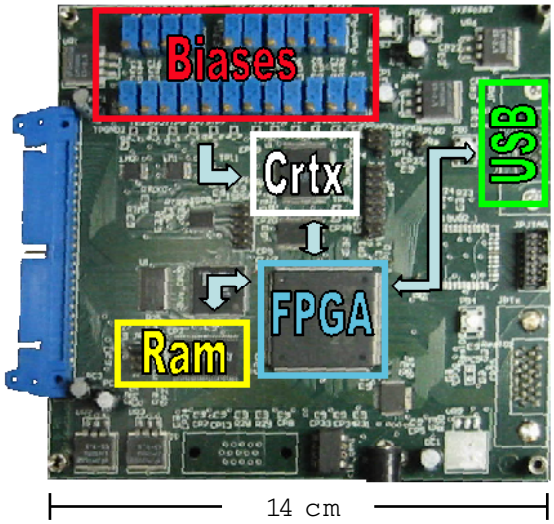


Figure 4.1: PCB test setup
Image of our populated printed-circuit board that was designed to test the bump chip (Crtx). The heart of the board is a programmable logic chip (FPGA), which coordinates digital communication between the transmitter and receiver of the bump chip, a memory chip (RAM), and a USB2.0 link (handled by a second board, which is not shown). The parameters of the bump chip are set by two rows of adjustable voltage sources (Biases).

We have chosen to design our board around an FPGA, instead of using a traditional microcontroller, because the FPGA's hardware can be customized to process spikes; a microcontroller's hardware, on the other hand, is designed for general computing and the additional overhead severely limits real-time spike processing.

We present an overview of the functions of our board:

1. Communicate with PC via USB2.0

Our board provides bidirectional communication between the bump chip and a PC using a USB2.0 link (via a second USB board, which we do not describe here). In brief, we stream address-events to and from a USB chip (Cypress FX2), which stores them in a pair of buffers (4x256x16 bit FIFOs), one pair for each direction. When a

buffer fills, it is transferred to–from a PC at a high-rate (we achieve 7 million spike events/sec); while one buffer is being transfered, the other buffer stores the incoming events, resulting in near seamless transfers. For a more comprehensive description of how address-events are sent over USB2.0, please consult [65].

2. Inject timing events

We inject timing events into the transmitter's event stream once every 50 microseconds. These timing events serve two functions: First, they are used for plotting (spike rasters, etc.), and for analyzing the data offline; without them, we can only determine the relative order of spikes and not their precise times. Second, the timing events set the lower bound on the frame rate of our GUI (described in Section 4.1.1.3). For example, the GUI can only update when it receives a buffer from the USB chip — however when activity is sparse (e.g., the chip is barely firing), the buffers might take a long time to fill up. By injecting timing events at regular intervals, we are able to ensure that the buffers will fill at minimum rate.

3. Deliver synaptic events on time

The PC can send blocks of data (buffers) to our board (via USB), but the timing of these buffers is not consistent. As a result, the PC can not be used directly to stimulate our synapses. For example, consider the case where we wish to stimulate a patch of neurons staggered by exactly 50ms: if each epoch exactly fills a buffer, we have no control when these buffers will arrive; in fact, the PC sends them as fast as it can as determined by the operating system, and the jitter is typically on the order of

milliseconds (but it can be indefinite if the operating system is busy). Our solution is to send both the locations *and* the times of the synaptic events from the PC — the FPGA is then responsible for ensuring the events are delivered at their appropriate times. Returning to our previous example, we would send a patch of neurons along with their times. As long as the PC sends these buffers fast enough so that there are always events in the queue (an easy constraint to meet), the delivery times can be near perfect (at the resolution of 50 nanoseconds, which is determined by the FPGA's clock frequency).

4. Implement fanout with AER

The final function of our board is to implement recurrent excitatory connections (EXC to EXC) using address-events. In our system, an off-chip random-access memory (RAM) stores the addresses of up to 64 postsynaptic cells associated with each presynaptic cell (2,304 of them). Each table of 64 comprises either the addresses of the postsynaptic cells (stored consecutively starting at 1), or a reserved address (stop event), which indicates that there are no more connections; the location of each table in the RAM is indexed by its associated presynaptic cell (e.g., the postsynaptic targets of neuron 4 are stored in the 4th table, etc.). To achieve a fanout, each presynaptic EXC event initiates a read sequence that copies the appropriate postsynaptic events from external RAM memory to local memory located on the FPGA. The entries that are not stop events are sent to cortex chip's receiver, where they are merged with the incoming synaptic events en route from the PC.

Our fanout implementation is the first of its kind that allows each presynaptic cell to connect to an arbitrary number of postsynaptic cells (as long as the total number of connections per cell is less than 64); past implementations required that synaptic

spread was constant. Initially, we wanted the flexibility to reduce (or increase) the amount of recurrent feedback each cell receives — possibly limiting the feedback of overactive cells. We ultimately abandoned this idea because it was not clear how to choose the number of connections each neuron should receive, without making rather arbitrary assumptions. Therefore, our board only implements six nearest-neighbor connections for each cell.

4.1.1.3 GUI

With our custom board, we now have the ability to stream spike events in real-time between our chip and a software application (programmed in Microsoft Visual C++); this GUI handles the high-level functions of our setup, allowing us to visualize chip activity (e.g., processing and plotting incoming spikes in real-time), saving raw and processed spikes, and coordinating the experiments (e.g., presenting sequences of oriented gratings). These high-level functions have the potential to hold up board-level communication — if the buffers fill up faster than the GUI can perform its tasks (e.g., updating the screen). Therefore, we use a commercial USB driver from *Thesycon*, which provides us with the ability to probe the status of the USB chip's buffers (both input and output) without holding up our GUI. With this driver, our GUI is free for the majority of the time to process, plot, and save spike data — occasionally being held up to copy filled buffers to–from memory.

To run a typical experiment, our GUI follows the sequence specified within a table, which is defined within our source-code. The table includes: 1) the locations of the stimulus files (which contain synaptic events and times), 2) the sequence of the stimulus presentations, 3) the desired timing resolution for the saved data, 4) the locations of the output files, and 5) the desired visualization of the data (spike rasters, activity maps, selectivity

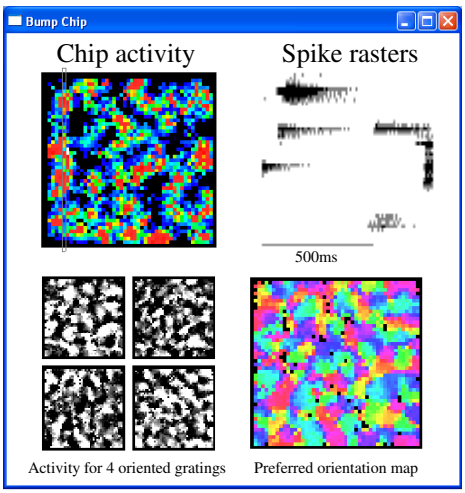


Figure 4.2: GUI
Screen shot of the real-time GUI during a typical experiment.

maps, etc.). A screenshot of one such experiment (Figure 4.2) shows the activity of the chip across space, saved activity for four different oriented gratings, spike rasters for one column of neurons, and the resulting preferred orientation map. These GUI components are shown here to demonstrate our real-time setup; we describe their use in more detail in the following chapters.

4.1.2 Experimental procedure

To explore the orientation selectivity (OS) of our network, we present full-field gratings to the chip and record the resulting responses, following a procedure similar to imaging experiments in V1. Of particular interest are the trial-to-trial network responses, and their averages, which are traditionally referred to as *single condition maps* (where each condition corresponds to a different orientation).

We perform our experiment in the following steps:

1. Create drifting gratings

First, we create two-dimensional drifting sinusoids (S) that are defined for positions $\{y_i, x_i\}$ and time t by:

$$S_i(t) = \cos\Big(\frac{x_i \cos(\theta_k) + y_i \sin(\theta_k) + \omega t}{\lambda}\Big), \tag{4.1}$$

where θ_k is orientation, λ is the period, and ω is the speed. Next, we separate S into positive (S^+) and negative (S^-) channels:

$$S_i^+(t) = \max(S_i(t), 0) \tag{4.2}$$

$$S_i^-(t) = |\min(S_i(t), 0)|. \tag{4.3}$$

Note that each channel is defined to be strictly positive (luminance). For one set of experiments, we will use these half-wave rectified sinusoids as the input to our network (after they are transformed into spikes). The majority of our experiments, however, use square-wave gratings (G), which are created by applying the following non-linear transformations:

$$\{G_i^+(t), G_i^-(t)\} = \begin{cases} 1 & \text{if } \{S_i^+(t), S_i^-(t)\} > c, \\ 0 & \text{if } \{S_i^+(t), S_i^-(t)\} \leq c, \end{cases} \tag{4.4}$$

where c sets the inter-bar distance. We will show later on that both square-wave and sinusoidal gratings lead to similar results — the square-wave gratings are preferred

because they drive the bumps more effectively (shown in Section 4.2.2).

2. Generate afferent spike trains

The next step is to transform the grating stimuli into afferent spikes trains that can drive our cortical network. We accomplish this by generating ON and OFF spike events using Poisson event generators (simple neuron model) that are modulated by grating waveforms; the resulting events project retinotopically to the ON and OFF synapse circuits on our chip.

The probability that a spikes occurs in a given time interval is:

$$\{p_i^{\text{ON}}(t), p_i^{\text{OFF}}(t)\} = \{G_i^+(t), G_i^-(t)\}\mathrm{F}dt, \tag{4.5}$$

where F is the peak event rate, and dt is the length of the interval. We determine an event occurred by drawing numbers with equal probability from the interval $[0, 1]$, denoted by $u_i(t)$, and assigning spikes to the subset locations where $u_i(t) \geq 1-p_i(t)$. This procedure approximates a Poisson point process when p is small.

We use the following parameters for gratings: $\lambda = 16$ pixels, $\omega = 32\pi$ pixels/s (1Hz temporal frequency), $c = 0.8$, and $\theta_k = \pi k/8 - \pi$ for $k = 0, \ldots, 16$ (-180 to 180 degrees); and for spike generation: $\mathrm{F} = 117$Hz for sinusoids, $\mathrm{F} = 183$Hz for square-waves, and $dt = 0.5$ms. The value of F is chosen so that events are generated at a mean rate of 75Hz; this choice results in the cortical cells receiving synaptic events at a mean rate of 75Hz (recall that cortical neurons are driven by both ON and OFF events). These spike-events are computed off-line and the results are saved into

files that specify the locations and times of the events.

3. Stimulate and record

The final step in our experimental procedure is to drive our chip with the precomputed afferent spike trains (step 2), while recording its responses (for a review on the detailed real-time setup, please refer to Section 4.1.1). For each experiment, we choose a trial duration D (i.e., how many grating cycles are presented in sequence), the number of repetitions R, and the set of stimuli k (i.e., which grating directions are used). Then, during each stimulus presentation (direction θ_k and repetition r), we record the number of spikes that cell i elicits to cycle t, and denote these spike counts as $f_{i,k}(t, r)$.

We obtain single condition maps (SCMs) by averaging the network activity over each trial to a particular stimulus condition. Formally, we compute:

$$T_{i,k}(r) = \frac{1}{D-2} \sum_{t=3}^{D} f_{i,k}(t, r). \tag{4.6}$$

Note that we disregard the first two cycles of each trial, giving the network 2s to form bumps. Often, we refer to $\langle T_{\mathrm{i_o},k} \rangle$ (averaged across repetitions) for a particular cell $\mathrm{i_o}$ as its *tuning curve*; from this point of view, our SCMs are the collection of tuning curves for all the cells in the network, and we use these two terms (SCMs and tuning curves) interchangeably. In this regard, our terminology is different from the imaging literature; each pixel of their SCMs represents the collective activity of hundreds of neurons.

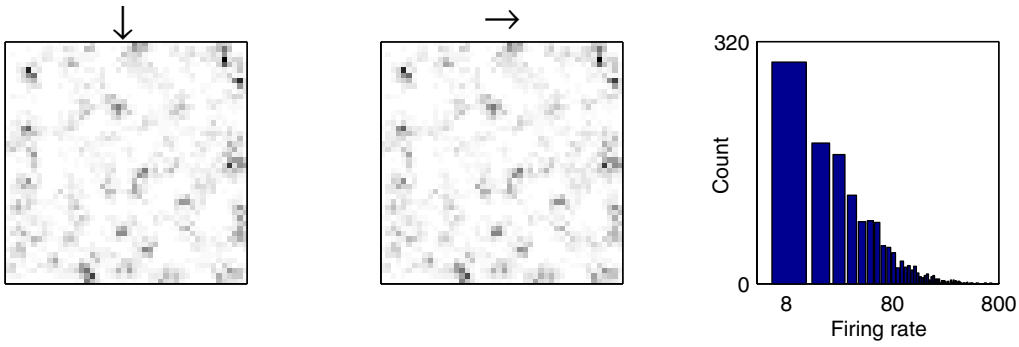
4.2 Innate selectivity

In this section, we explore how the bumps in our heterogeneous network interact with oriented stimuli. Our starting point is to examine the SCMs directly (Section 4.2.1). After verifying that our SCMs are in fact selective (and reliably so), we then explore how orientation preferences are distributed across space by plotting them as a color-coded map (Section 4.2.2).

4.2.1 Orientation selectivity

Before applying traditional techniques to characterize the chip's selectivity (Sections 4.2.2 and 4.3), which involve collapsing the SCMs into smaller (more manageable) datasets, we first characterize the SCMs directly. Our intention is to get a feel for the raw SCMs (i.e., the tuning curves), since they are the basis for the more traditional measures that we use later on.

Our first order of business is to simply visualize the SCMs so that we can get a rough estimate of the chip's selectivity (Section 4.2.1.1). After verifying that different stimuli do in fact lead to noticeably different maps, we then characterize the reliability of the responses to the *same* stimulus (Section 4.2.1.2), and to *different* stimuli (Section 4.2.1.3) for independent trials; in this analysis, we consider all the pairwise combinations between the different SCMs. After confirming that a subset of our cells have non-flat (statistically significant) tuning curves, we finally explore the average trend of the tuning curves (Section 4.2.1.4). Specifically, we test that the differences in cell responses are modulated by the differences in orientations for each stimuli pair; a smooth mapping between responses and orientation difference is observed in biological tuning curves.



a b c

Figure 4.3: Single condition maps
SCM elicited from a horizontal (a) and vertical (b) grating. Cell responses from (a) shown as a histogram with logarithmic scale (c).

4.2.1.1 Visualizing SCMs

In this section, we check that our network is selective to orientation by visually inspecting the SCMs ($T_{i,k}$). For instance, we plot $\langle T_{i,k}(r) \rangle$ for two orthogonal orientations, $k = 0$ (Figure 4.3a) and $k = 4$ (Figure 4.3b), which correspond to -180 and -90 degree gratings (square-wave), respectively. For this experiment (and the rest of the experiments in this section), the trial duration (D) is 100 cycles, and the number of trial repetitions (R) is 50; that is, the tuning curve of each cell is an 8×50 table where the eight rows represent the different orientations and the 50 columns represent the different trials, where each trial is the mean rate over 98 cycles ($D - 2$).

These two SCMs in Figure 4.3 look almost identical, suggesting that our network is not very selective. However, before we throw in the towel, we should also examine the distribution of the underlying responses across cells (Figure 4.3c). This histogram reveals that cell-to-cell firing rates vary by almost 2.9 orders of magnitude, with a small percentage of cells reaching rates up to 800Hz. On the other hand, the stimulus-to-stimulus rate changes for each cell vary by only 1.8 orders of magnitude (1 to 70Hz, data not shown). In

other words, the intensities of our SCM plots are determined by the cell-to-cell variations across the network, and *not* across stimuli. One solution is to simply plot the difference of these maps; this method, however, requires $\frac{Q^2-2Q}{2}$ difference maps for all unique pair-wise stimulus combinations (24 in our case) — not a parsimonious representation.

Our method to eliminate the cell-to-cell variability is to normalize their tuning curves; this normalization will allow us to compare the cells using the same intensity scale. Our normalization is computed as:

$$R_{i,k}(r) = \frac{T_{i,k}(r) - \min(T_{i,k}(r))_k}{\max(T_{i,k}(r))_k - \min(T_{i,k}(r))_k}, \tag{4.7}$$

where the $\max$ and $\min$ functions are computed across stimuli (indicated by the subscript k). In essence, $R_{i,k}(r)$ scales a cell's tuning curve to be between 0 (min response) and 1 (max response) without changing its relative shape; the notion of an absolute reference for all cells has been lost, but this is exactly our intention. We refer to these normalized single condition maps ($R_{i,k}$) as NSCMs; it is important to note that NSCMs are *only* used for visualization, and all other analyses use the raw data directly.

We plot $\langle R_{i,k}(r) \rangle$ for eight orientations ($k = 0, \ldots, 7$), ranging from -180 to -22.5 degrees (Figure 4.4). Unlike our previous plots, NSCMs reveal periodic blob-like regions that are visibly distinct for different stimuli, characteristic of orientation selectivity. In essence, these eight plots show the tuning curves for all of the cells on our chip (with their values normalized between 0 and 1).

Although we are off to a good start, since the chip's NSCMs are similar in appearance to the ones imaged in V1, the results still seem unsatisfactory. Specifically, we do not have a good handle on the repeatability of the underlying responses that make up $R_{i,k}$. We

address this issue in the following section.

4.2.1.2 Repeated stimulus reliability

Here, we quantify the reliability of $T_{i,k}$, and in particular, we check that cell responses are repeatable across trials when stimulus orientation is the same. For example, if bump positions are influenced by initial conditions instead of oriented gratings, then we would expect the SCMs to vary considerably for independent trials under identical stimulus conditions (since the initial conditions are random for each trial). Reliability is relevant for our model, since ours is the first of its kind to use spikes, instead of rates, to obtain orientation selectivity; in particular, spikes can add response variability (in coupled systems) due to their discrete nature [55].

Three related techniques are explored to gauge response repeatability. First we use a split-half reliability test to determine if cell responses in independent trials are correlated [66]. Then, we introduce a handy metric, called the selectivity index, that quantifies the relative change (or dispersion) in a cell's response; this metric is used to identify cells that change their rates even when the stimulus is the same (i.e., their responses are unreliable). Finally, we employ a standard statistical test to determine if the trial-to-trial rate deviations of cells can be accounted for by sampling errors.

1. Split-half reliability test

To perform a split-half reliability test, we split $T_{i,k}$ into two groups, $T^1_{i,k}$ and $T^2_{i,k}$ such that each group consists of $\frac{D}{2}$ different trials; then we compare the mean response of each cell in the two groups when the stimulus orientations are identical.

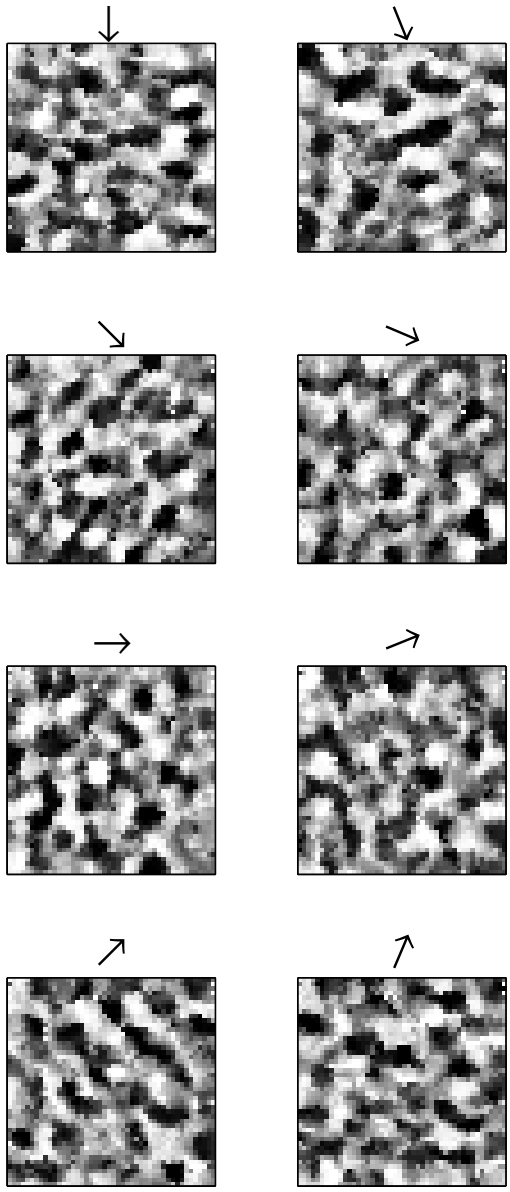
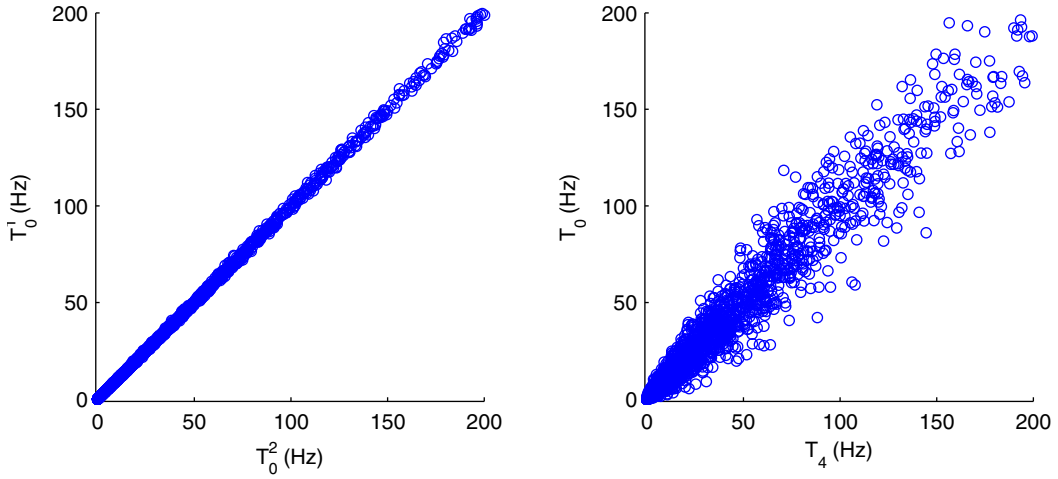


Figure 4.4: Normalized single condition maps
Chip activity (normalized) for eight directions. The intensity scale is from 0 (white) to 1 (dark).



a b

Figure 4.5: Split-half reliability test

Comparison of cell firing rates for independent trials when stimulus orientations are the same ($-180°$) (a) and orthogonal ($-180°$ and $-90°$) (b).

For an example, we plot $\langle T^1_{i,0}(r)\rangle$ against $\langle T^2_{i,0}(r)\rangle$, where the 0 indicates the stimulus was a $-180°$ grating (Figure 4.5a); note that we only show firing rates up to 200Hz, since only 3.6% of our cells fall in the 200–800Hz range. As expected for reliable data, the cell responses from the independent groups have a near perfect linear relationship with a slope of 1. For comparison, we also plot $\langle T_{i,4}(r)\rangle$ against $\langle T_{i,0}(r)\rangle$ (Figure 4.5b), which reflects the case where the two groups have orthogonal stimuli; this case is explored in detail in Section 4.2.1.3.

2. Selectivity index

To quantify the dispersion of a cell's response (i.e., the deviation from the straight line in Figure 4.5a), we introduce a selectivity index [67] (also known as the Rayleigh-Michelson ratio); this measure allows us to identify (and eliminate) a class of unreliable cells right off the bat. Specifically, we compute:

$$z(a,b) = \frac{|a-b|}{a+b}, \tag{4.8}$$

where a and b represent firing rates of a cell in two conditions. The useful property of z is that it measures the *proportion* between a and b, regardless of their underlying values. For example, if $z = 1/4$, this implies that a is 40% larger than b (or visa-versa, depending on whether a or b is larger); this percentage is calculated by $1 - \frac{1-z}{1+z}$. Since here, we are interested in the repeatability of our responses, a and b are measured in independent trials for stimuli with the same orientation — our hope is that the selectivity is low (or, repeatability is high).

We compute $z(T^1_{i,0}, T^2_{i,0})$ for each cell and plot them against their mean firing rate $\langle T_{i,0} \rangle$ (Figure 4.6a). The selectivity index reveals that a portion of cells, skewed towards low firing rates, do in fact change their rates by a significant proportion, which we interpret as being unreliable. We would like to eliminate these unreliable cells from our analysis, however, we must consider that an unreliable cell for this particular orientation might be reliable for a different orientation. Our solution is to classify cells that do not fire more than a mean rate of 2Hz to *any* orientation as unresponsive, marked by blue points. This classification was chosen since cells that fire more than 2Hz rarely have z values greater than 0.15.

3. Statistical test

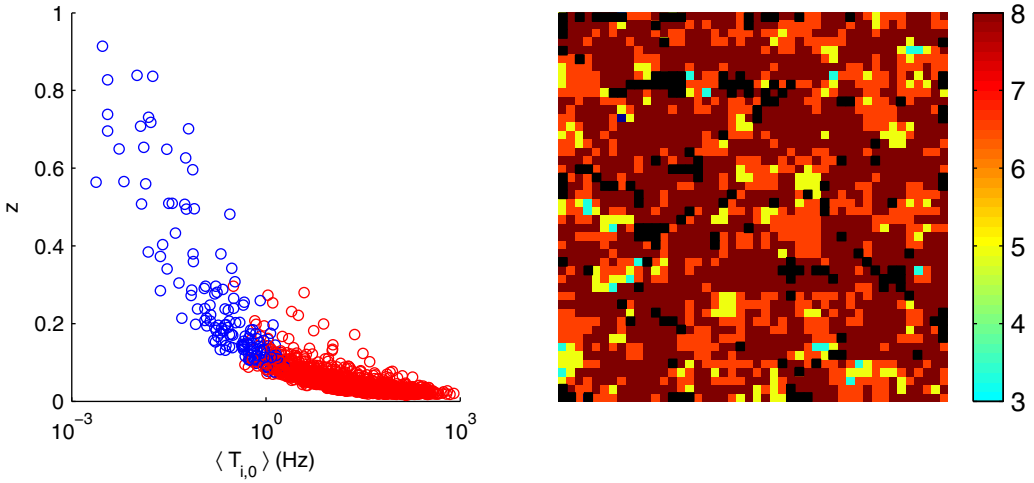
After eliminating the unresponsive cells (8.3% of the population), we now characterize the reliability of the remaining cells using a standard statistical test. Specifically, we employ a z-test ($p < 0.05$) where our null hypothesis, denoted by H_0,

is: $\langle T^1_{i,k}(r) - T^2_{i,k}(r)\rangle = 0$ (i.e., the difference in a cell's firing rate for independent trials is zero when the stimulus orientation is the same); this test is repeated for all cells, for each of the eight stimuli ($k = \{0, \ldots, 7\}$). If we can not reject H_0, we are inclined to believe that the observed response deviations can be explained by chance occurrences (i.e., the rates are statistically equivalent). We summarize our results by recording each instance where H_0 is accepted, referred to as r_k, and count these instances for each orientation (Figure 4.6b). In this plot, a count of eight indicates a cell's mean rate was reliable in the split-half trials for all of the orientation, while lower counts signify that there were unreliable trials; we mark the unresponsive cells in black. As can be clearly seen from the plot, H_0 is accepted in the majority of trials, and 62% of cells respond reliably for all eight stimuli. To verify that a z-test is appropriate, we also used a bootstrap procedure (500 bootstrap samples of $\langle T^1_{i,k}(r) - T^2_{i,k}(r)\rangle$ using $p < 0.05$); the two procedures agree on 98.4% of the cells.

4.2.1.3 Inter-stimuli reliability

Our aim in this section is to test whether different orientations change a cell's rate more than chance. For example, consider the relationship between $\langle T_{i,0}\rangle$ and $\langle T_{i,4}\rangle$, which correspond to -180° and -90° orientations, respectively (Figure 4.5b); note that here, we no longer split the trials into two groups. This plot reveals another positive correlation, although this correlation is noticeably more disperse than the one in Figure 4.5a.

The fact that a cell's response is correlated with itself, even when the stimuli are different, is perhaps not too surprising. In the cortex, for instance, cells with spontaneous firing rates will typically have a baseline response even for non-optimal stimuli, and often the baseline will increase for all stimuli [68]. On the chip, cells are correlated across dif-



a b

Figure 4.6: Selectivity index and repeatability

(a) Cell selectivity indexes (z) for independent trials when the stimulus is $-180°$, plotted against the their average firing rates (over these trials); blue markers signify cells that have a mean firing rate less than 2Hz to any orientation (unresponsive) whereas red markers are all other cells (responsive). (b) The number of instances where a cell's response is statistically equivalent in independent trials over all orientations.

ferent stimuli for a simple reason: if they participate in a bump for one orientation, there is a higher chance that they will participate in a bump for other orientations. In both of these examples, the correlations are caused by factors that are stimulus-independent (and putatively cell, or network dependent).

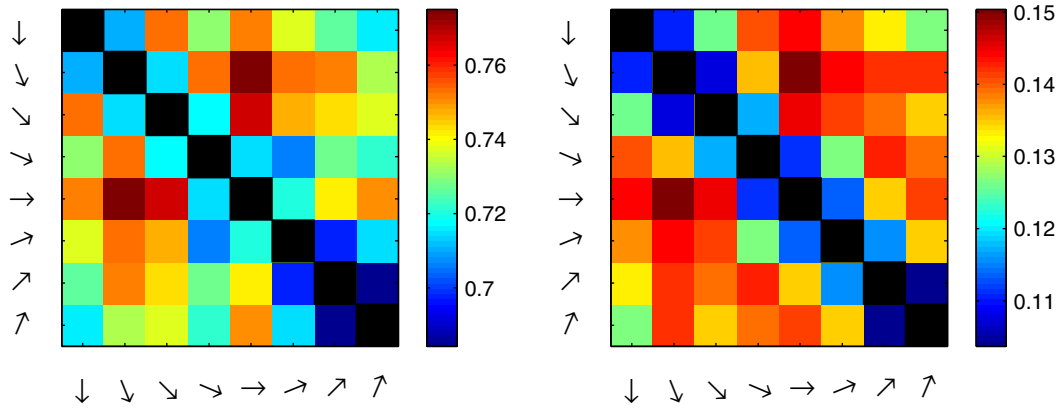
To determine if the deviations observed in Figure 4.5b are significant, we again use a z-test — except now we are interested in cells that reject H_0 (the null hypothesis), or accept the alternative hypothesis (H_1) that states the observed rate deviations are beyond chance level. Formally, we define H_1 as: $\langle T_{i,k_1}(r) - T_{i,k_2}(r) \rangle \neq 0$, where $k_1 = 0, \ldots, 7$ and $k_2 = 0, \ldots, 7$ when $k_1 \neq k_2$ (again only considering responsive cells). Then, for each $\{k_1, k_2\}$ stimulus pair, we record the cells that are selective, or H_1 is accepted when the stimuli are different (denoted by $r^d_{k_1,k_2}$), *and* their responses to the individual stimuli were consistent, or H_0 was accepted for the same stimuli (ensuring that the rate change is not

due to the cell being unreliable for one or both orientations); the conjuction of these events is denoted by $r^d_{k_1,k_2} \cap r_{k_1} \cap r_{k_2}$. We report the results (as the proportion out of the total population) in the symmetric 8×8 matrix (Figure 4.7a), where each entry corresponds to a particular row–column stimulus pair; note that the diagonal entries are set to black. Again, for verification, we used a bootstrap procedure (500 samples with $p < 0.05$), which was consistent with our z-test on 99.3% of the dataset.

Figure 4.7a reveals that similar orientations have a smaller proportion of selective cells (the strip of blue straddling the diagonal), whereas orthogonal orientations result in larger proportions; however, the differences in these proportions are relatively small, and the majority of cells are able to discriminate between any orientation pair.

4.2.1.4 Smooth mapping

This preceding analysis confirms that a subset of our cells are in fact selective to orientation, and this selectivity is consistent in independent trials. It does not, however, reveal the shape, or magnitude, of the selectivity. To uncover the amount our tuning curves change for different orientation pairs, we enlist the previously described selectivity index (z). Specifically, we compute the average z for each of the selective cells from Figure 4.7a for each stimuli pair. This result is again shown as an 8×8 matrix (Figure 4.7b). Here, we see that the smooth mapping does exist: similar orientations result in $\approx 19\%$ rate changes ($z = 0.11$), whereas orthogonal orientations result in $\approx 25\%$ rate changes ($z = 0.14$). These differences are not dramatic, but they exist nonetheless. It is important to note that the trend in Figure 4.7b is not simply due to the trend in Figure 4.7a, since the number of selective cells should not affect their magnitudes (although the two trends may be caused by the same underlying factors).



a b

Figure 4.7: Summary of cell reliability across stimuli pairs
(a) The proportion of cells that change their firing rates more than chance for each unique stimuli pair; the diagonal entries (self comparisons) are set to black. (b) The average selectivity index for all the cells reported in (a).

4.2.2 Prefered orientation maps

In this section, we collapse the SCMs into one map by computing each cell's prefered orientation (PO). The resulting PO map then allows us to visualize how orientation preferences are distributed across our network, using only a single plot.

The PO of a cell should summarize its tuning curve. A popular method is to use a vector sum, which is computed as:

$$\Theta_i = (1/2)\arg\Big(\sum_{k=0}^{\mathrm{Q}} T_{i,k}\mathrm{e}^{(2j\theta_k)}\Big), \tag{4.9}$$

where Q is the number of stimuli, $j = \sqrt{-1}$, and the $\arg()$ function returns the phase angle of the summed vector. The factor of two in the exponent scales θ_k so that orthogonal orientations point in opposite directions (i.e., their vectors subtract).

There are two issues that crop up when plotting PO maps. First, we are not able to directly map orientation to intensity since orientation is periodic, and intensity is not. One solution to this dilemma, originally introduced by Blasdel and Salama in 1986, is to assign a color to each pixel based on its PO using complementary colors (e.g., red and blue) to represent orthogonal orientations [69].

After adopting the PO–color mapping strategy, a second plotting issue arises: every pixel will be mapped to a color regardless of the underlying responses. In other words, weak responses (noise) cannot be differentiated from strong responses (signal). Therefore, it is common practice to use color intensity to differentiate between signal and noise. In our case, we have decided to set the intensity of each color based on the repeatability of our PO signal. Specifically, we measure the similarity index (SI) (see Section 2.1.2 for a description) to compare cells' POs in independent trials; the underlying idea is that reliable cells will report the same orientation in different trials (SI=1), whereas unreliable cells, mostly influenced by noise, will report random orientations (SI=0.5). Finally, we determine the intensity of each cell as: $\frac{\langle SI_i \rangle - 0.5}{0.5}$, where $\langle SI_i \rangle$ is the mean SI taken over all trial repetitions for cell i.

We plot our chip's PO map (Figure 4.8) employing the techniques described in this section. This map was computed by presenting 16 square-wave gratings that span the full circle (-180 to 157.5 degrees) and averaging the resulting SCMs with opposite directions, or $T_{i,k} = 0.5(T_{i,k} + T_{i,k+8})$ for $k = 0, \ldots, 7$; the experiment parameters are: the trial duration (D) is 100 cycles, and the number of trial repetitions (R) is 25. Figure 4.8 reveals that, like the cortex, nearby cells on our chip tend to have similar POs, and these preferences repeat at regular intervals. Moreover, the regions of smoothly changing POs are interspersed by regions of rapid change (fractures) that converge at either a line or a point.

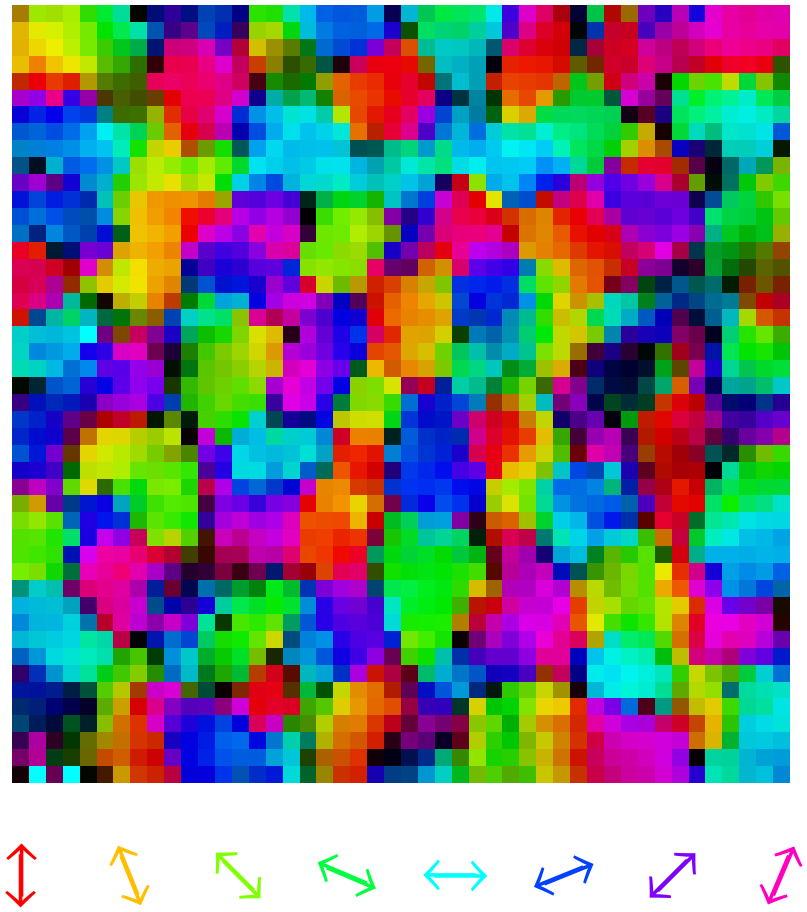


Figure 4.8: Orientation selectivity of the bump chip
Spatial distribution of preferred orientations to square-wave grating inputs. Activity for opposite directions was averaged before computing the angle map.

It is important to note a key difference when comparing our PO map with the ones imaged in cortex: a pixel in our map is derived from the exact tuning curve of a cell, whereas in an imaged map, it reflects the average tuning over a population of cells (> 100). This difference comes into play when orientation signals change rapidly (e.g., near fractures). Specifically, optical signals in these regions will be noisy because of co-localized conflicting orientation signals, and as a result, it is not clear whether the noise can be attributed to the individual cells, or the imaging procedure. In our chip experiments, however, noisy signals are always the result of the individual cell responses.

Although we are primarily interested in orientation selectivity in this project, we also plot the preferred direction (PD) map (Figure 4.9) for completeness. The PD map, unlike the PO map, considers all 16 directions; specifically, it assigns opposite colors to the same orientations with opposite directions. We observe that nearby cells have similar direction preferences, which again repeat at regular intervals. The spatial extent of these direction columns are about half the size of the orientation columns, consistent with maps observed in the cortex. Note that because direction selectivity is markedly lower than orientation selectivity (data not shown), we plot the PD map with the intensity set to one.

Finally, we determine if our maps are selective to spatio-temporal correlations in the inputs (i.e., drifting oriented inputs), or to the precise spatial patterns of the inputs (e.g., square-wave versus sinusoid gratings). To accomplish this, we compare the PO map created with square-wave gratings (Figure 4.10a) with one created with sinusoid gratings (Figure 4.10b). At a glance, the structure and colors of the two maps look very similar. To quantify their similarity, we compute the SI (described previously) for each cell (Figure 4.10c). Although there are a few regions where the maps report different POs, the majority of cells have nearly identical POs: the average SI is 0.77. This result confirms that our cells are in fact selective to orientation, and not a particular grating type.

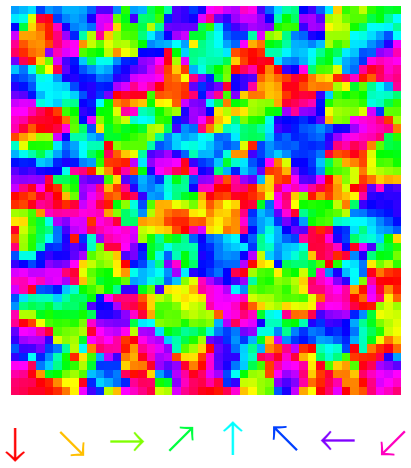
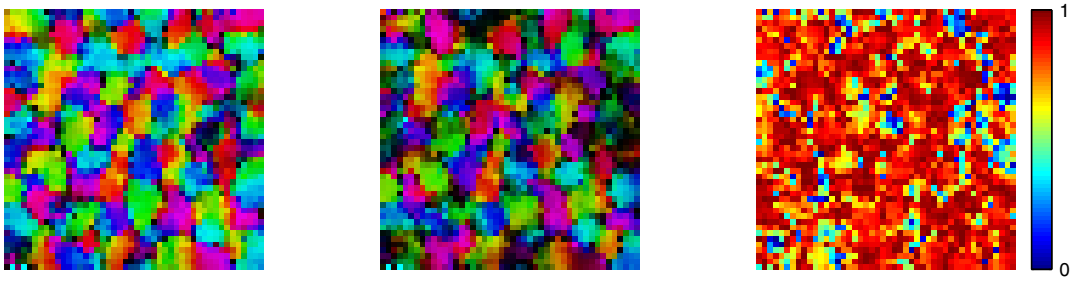


Figure 4.9: Direction selectivity of the bump chip
Spatial distribution of preferred directions.

One noticeable difference between the two maps, however, is that the one created from sinusoids is darker in certain regions (typically regions where the two maps diverge). This observation is not too surprising: in the case of half-wave rectified sinusoids, cells are always being stimulated *regardless* of orientation (except at the zero crossings) since each cell receives both ON and OFF inputs. Square-wave gratings, on the other hand, concentrate their stimulation in space along the axis of orientation and in time near the 0 (ON) and 180 (OFF) degree phases. In other words, the edges of the square-wave gratings are more effective at driving bumps and the resulting maps are less noisy; in light of this result, we no longer consider sinusoid gratings as inputs. However, we expect that the qualitative aspects of our results would be similar if we used sinusoid gratings, but we would require more trials to obtain the same reliability.



a b c

Figure 4.10: Map comparison for square and sine -wave inputs
PO maps created from square-wave (a) and sinusoid (b) gratings; the two maps are compared using the similarity index (c).

4.3 Quantifying selectivity

In addition to estimating a cell's PO, we are also interested in quantifying its selectivity (with a single number). The conventional method for measuring selectivity is to characterize how tight a cell's tuning curve is around its PO. Note that our previous selectivity measure (Section 4.2.1.2) is inappropriate here since it considers all unique stimulus pairs (i.e., many numbers). Common metrics that fit this bill include bandwidth, which approximates the width of the tuning curve near its PO [70], and normalized vector magnitude (NVM), which measures how well the PO captures the entire tuning curve (described in detail in Section 4.3.1). For our analysis, we focus on NVM since it is a robust measure of selectivity in the presence of noise [68].

Implicit in these conventional measures of selectivity (bandwidth and NVM) are two assumptions. The first assumption is that each cell's tuning curve has a single mode near its PO. While this preconceived notion is perhaps justified in mature cortex, we do not wish to make any unwarranted assumptions in our chip (or developing cortex). Therefore, we

entertain the possibility that each cell can encode information about multiple orientations (e.g., the possibility that multiple PO maps exist simultaneously). Examples include cells that are selective to crosses (i.e., they respond to cardinal orientations but not oblique ones), or wedges (i.e., they responds to all orientations except one). Such cells have in fact been observed in single unit recordings in immature cortex (recording through closed eye lids in ferret), although, they were not described using this description [15].

The second assumption is that a cell's tuning curve accurately reflects its response distribution $f_{i,k}$; recall that $f_{i,k}$ represents the number of spikes a cell elicits for one cycle at orientation k. A prevailing view in neuroscience is that the distribution of spike counts in a sufficiently long time window (such as 1s) can be captured with a few parameters (typically a Poisson or Gaussian distribution). The cells in our network do not necessarily fall into this category — for instance, we sometimes observe bimodal distributions of spike counts. We attribute these two modes to shifting bumps: one mode, characterized by a cell firing at a low rate, occurs when a cell is not participating in a bump; the second mode, characterized by a high firing rate, occurs when a cell is part of a bump (i.e., the bump has shifted). It should be noted that bimodal response distributions have also been observed in the cortex; in fact, it has been shown that up to 60% of complex cells in cat visual cortex exhibit bimodal membrane-potential distributions (termed *two-state* fluctuations) [71].

How should we characterize the selectivity of a cell that does not follow one, or both of these assumptions? One possibility is to take a case-by-case approach — first checking if a cell's response can be parameterized, then testing the number of encoded POs. However, this solution seems tedious. A more elegant approach, which we explore in Section 4.3.2, is to compute how much information a cell can convey about its stimulus. Unlike NVM, information is independent of any particular coding strategy, and is computed directly from the underlying response distributions, instead of just the tuning curve.

4.3.1 Normalized vector magnitude

Normalized vector magnitude (NVM) quantifies the degree that a cell's PO is consistent with its tuning curve. It is computed by dividing the vector sum of the averaged responses by the magnitude sum, or formally:

$$M_i = \frac{|\sum_k T_{i,k} \mathrm{e}^{j2\theta_k}|}{\sum_k T_{i,k}} \tag{4.10}$$

where $j = \sqrt{-1}$, and θ is expressed in radians. A cell that is poorly tuned and responds in equal parts to orthogonal orientations (Figure 4.11a) will have a low NVM (≈ 0), whereas a cell that is exceptionally tuned and only responds to similar orientations (Figure 4.11b) will have a high NVM (≈ 1). We note that our measure of NVM is related to a measure found in the literature called *circular variance* (cv), where $cv_i = 1 - M_i$ [68].

We compute the NVMs (M_i) for the population of responsive cells (recall that responsive cells are characterized by an average firing rate of more than 2Hz to at least one orientation), using the same dataset as in Section 4.2.1.1 (square-wave gratings). The distribution of NVMs across the network is shown (Figure 4.12a). Here, we set the values of cells that are unresponsive (but not inactive) to 0; typically, these unresponsive cells can have artificially high NVM because their tuning curves resemble a delta function. We also plot the NVMs in a histogram, only considering responsive cells (Figure 4.12b); this distribution has a mean of 0.084.

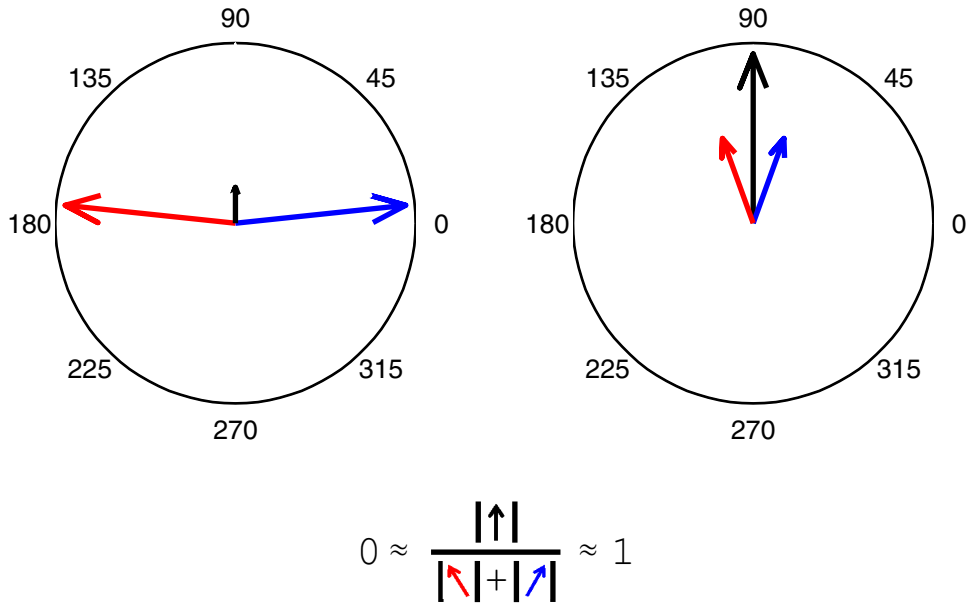


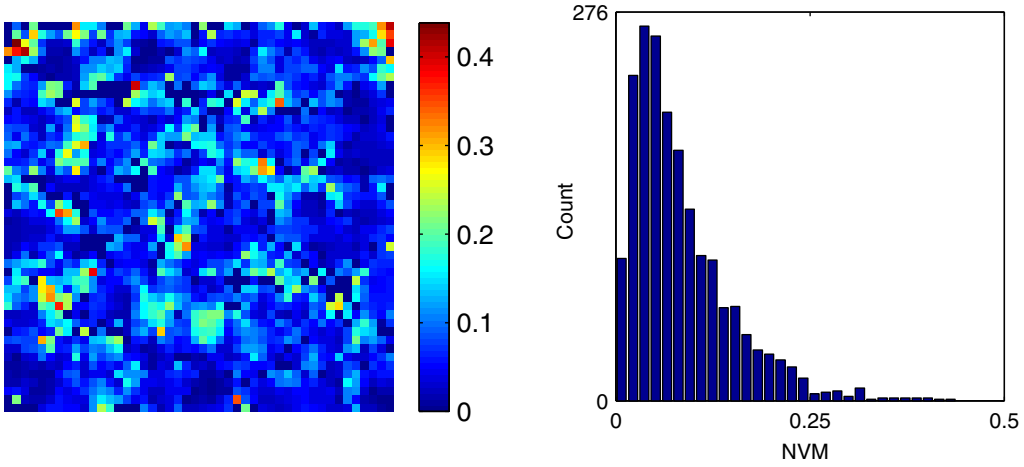
Figure 4.11: Computing normalized vector magnitude
Vector responses for a selective cell (left) and an unselective cell (right), and (bottom) calculation of NVM using vector components (blue and red) and sum (black).

4.3.2 Information

Cells with relatively low values of NVM can represent one of two scenarios: One possibility is that these cells are unselective to orientation. Or, another possibility is that these cells use a non-conventional encoding, which NVM is ill-equipped to measure. To distinguish between these possibilities, we use information theory, which provides us with a measure of selectivity that does not assume a particular coding strategy.

4.3.2.1 General background

Information theory, originally developed by Shannon in 1948, offers a mathematical framework for quantifying information transmission over a communication channel [72]. We



a b

Figure 4.12: Bump chip selectivity
NVM distribution (M_i) across space (a), and in a histogram (b).

offer a brief review: Consider encoding symbols s_k $(\mathrm{k} = 1, \ldots, \mathrm{Q})$ over a noisy channel such that the probabilities of observing the responses r_c $(\mathrm{c} = 1, \ldots, \mathrm{W})$ to each symbol are exactly known; we often refer to r_c as the *code words*. The number of bits (i.e., the number of yes–no questions) that it takes to specify the entire response distribution can be calculated by its entropy, or

$$H_{\mathrm{r}} = -\sum_{c} p(r_c)\log_2 p(r_c), \tag{4.11}$$

where $p(r_c)$ is the probability of observing r_c; the negative sign ensures that entropy is strictly positive since $p(r_c) \leq 1$ and hence $\log_2 p(r_c) \leq 0$ for all c. H_{r} represents the *maximum* amount of information that could be encoded. In a similar manner, we can also calculate the entropy of the response given the stimulus, or

$$H_{\mathrm{r|s}} = -\sum_{k} p(s_k) \sum_{c} p(r_c|s_k)\log_2 p(r_c|s_k), \tag{4.12}$$

where $p(r_c|s_k)$ is the conditional probability of observing r_c given s_k, and $p(s_k)$ is the probability of observing s_k (in a controlled experiment, $p(s_k)$ typically equals $1/\mathrm{Q}$). $H_{\mathrm{r|s}}$ is often referred to as response noise, since it measures the uncertainty remaining in the response when the stimulus conditions are known [73].

Shannon cleverly realized that information is the response entropy (H_{r}) less the noise entropy ($H_{\mathrm{r|s}}$). In other words, by removing the variability of the response that is independent of the stimulus (i.e., noise), we are left with the stimulus dependent variability (i.e., information). Formally, we write

$$I_{\mathrm{r,s}} = H_{\mathrm{r}} - H_{\mathrm{r|s}}, \tag{4.13}$$

where $I_{\mathrm{r,s}}$ is termed the *mutual information*. In the case that the responses are unrelated to the stimuli, or $p(r_c|s_k) = p(r_c)$, we observe that $I_{\mathrm{r,s}} = 0$ since $H_{\mathrm{r|s}} = H_{\mathrm{r}}$. In the other extreme where the response is unique for each stimulus and the stimuli are equally likely, we observe that $I_{\mathrm{r,s}} = \log_2 \mathrm{Q}$ since $H_{\mathrm{r|s}} = 0$.

Extending the ideas of information theory into the realm of neuroscience seem natural; after all, a neuron is simply a biological communication channel that encodes sensory events (either directly, or through downstream neurons) using spike times. In practice, however, one quickly runs into difficulty when directly applying Shannon's theory to the neural code. For example, since we do not know the nature of the neural code, the safest approach is to consider all the possible spike configurations within a biologically relevant time window as potential code words. Following an example from [74], where the time window of interest is 300ms, a neuron's average firing rate is 30Hz, and spikes are seg-

mented into 5ms bins, we estimate that there are nearly 10^{11} possible spike trains! Thus, with limited data (such as the lifetime of a cell), any attempt to measure the probabilities of these different code words will be severely biased.

Despite major challenges, researchers have successfully used information theory to quantify neural communication for certain niche problems in neuroscience. One application has been to pit rate codes against timing codes, attempting to address the heated question: do the precise times of spikes encode additional information about stimuli? Another application has been geared towards uncovering a neuron's capacity (or max information rate), without regard to which aspects of the stimulus are being encoded. Our requirements of information theory (which are described next in Section 4.3.2.2) are not as sophisticated as these more common applications; specifically, we are able to reduce the number code words by neglecting timing information, which simplifies our analysis significantly. For this reason, we do not expound upon information theory and timing codes any further; the interested reader should consult [74, 73] for overviews.

4.3.2.2 Estimating mutual information

We use information theory to quantify how well the neurons from our chip can encode their inputs. Two simplifications make this problem tractable: First, we restrict our inputs to drifting gratings that vary only in their orientations. Second, we consider the output of a neuron to be its spike count during one stimulus presentation (i.e., one grating cycle). This simplification is rooted in the idea that, because OS is a network phenomena (bumps interacting with gratings), each neuron should have a chance to sample at least one full cycle before its response is considered. Furthermore, by only considering spike counts, we are assuming that a neuron's OS is determined by the total amount of time it participates in

a bump, rather than the precise times of participation.

With these simplifications in hand, we compute the information of our cells by following Shannon's recipe — first estimating the probabilities, then computing their entropies. Our detailed procedure is described in the following steps:

1. Estimating $p(r_c), p(r_c|s_k)$

The first step is to estimate $p(r_c)$ and $p(r_c|s_k)$ from the observed spike count distribution $f_{i,k}$ for a particular neuron $i = \mathrm{n}$. This deceptively simple task brings to light an important dilemma: how should we choose the code words (r_c) for this neuron? One approach is to subdivide $f_{\mathrm{n},k}$ into W equal sized bins, such that each bin is a different code word. However, using equal sized bins is inefficient when $f_{\mathrm{n},k}$ has a wide range of values that are not evenly distributed (e.g., a bimodal distribution); the inefficiency arises because a large number of bins are needed to capture the local changes (e.g., the regions near the two modes), yet the majority of the bins remain unoccupied.

A more elegant solution is to adopt a variable binning technique that equalizes the number of samples in each bin. The basic approach is to stretch each bin until it contains $\frac{1}{\mathrm{W}}$ of the samples from f_{n} (recall that dropping the subscript k refers to all stimuli). For example, if $\mathrm{W} = 2$, then each bin (r_c) will contain half of the data points, and the two bins are split by f_{n}'s median. Using this approach, we immediately see that $p(r_c) = \frac{1}{\mathrm{W}}$ for all c (by definition).

To estimate the conditional probabilities, we count the number of instances where $f_{\mathrm{n},k} \in r_c$ (i.e., how many samples of $f_{\mathrm{n},k}$ fall within each bin), denoted by $y_{\mathrm{n},k}(c)$, and divide the results by the total number of samples for that trial ($\frac{\mathrm{N}}{\mathrm{Q}}$); formally, we

write that $p(r_c|s_k) = (\frac{\mathrm{Q}}{\mathrm{N}})y_{\mathrm{n},k}(c)$. The number of bins (W) is left as a parameter; we note that when W is too small, our information estimate will be low (since our choice of code words will neglect relevant details), and when it is too large, there is a sampling bias (described next).

2. Estimating entropy

To estimate the entropies H_{r} and $H_{\mathrm{r|s}}$, we simply plug in our estimates $p(r_c)$ and $p(r_c|s_k)$ into Equations 4.11 and 4.12. We immediately notice that $H_{\mathrm{r}} = \log_2 \mathrm{W}$ since $p(r_c) = \frac{1}{\mathrm{W}}$; this calculation reveals that equalizing the number of samples in each bin is equivalent to normalizing their entropies (i.e., the size of each bin has been chosen such that its entropy is exactly $\frac{1}{\mathrm{W}}\log_2 \mathrm{W}$ bits).

Because $p(r_c)$ and $p(r_c|s_k)$ are estimated from a finite number of observations, their naive entropy estimates are *downwardly* biased. This bias can be approximated as:

$$B = \frac{1}{\ln 2}\frac{w_o - 1}{2u}, \tag{4.14}$$

where w_o is the number of bins (code words), and u is the number of samples (see [75] for a detailed derivation); note that this bias is inversely proportional to the average number of samples in each bin (for large u). The debiased entropy is then calculated as $H = H^{\mathrm{B}} + B$, where H^{B} is the naive entropy estimate. This correction can be applied directly to H_{r} using $w_o = \mathrm{W}$ and $u = \mathrm{N}$. In the case of $H_{\mathrm{r|s}}$, we first calculate the entropy bias to each stimulus using $u = \frac{\mathrm{N}}{\mathrm{Q}}$ (i.e., the number of samples for each unique stimulus presentation); one caveat is that w_o should

represent the number of bins that *might* be occupied — thus, it is not clear how to treat empty bins (if they happen to occur). Heeding the advice from [75], we again use $w_o = \mathrm{W}$. The justification for this choice is that when the number of trials is sufficiently large, such that each bin has a reasonable chance of being occupied (i.e, $\frac{\mathrm{N}}{\mathrm{Q}} \times p(r_c) \gg 1$), then using all of the code words is a fair (and conservative) estimate. Then, to compute $H_{\mathrm{r|s}}$, we multiply the debiased entropies for each stimulus by $p(s_k)$ and sum the results (following Equation 4.12). where w_o is the number of bins (code words), and u is the number of samples (see [75] for a detailed derivation); note that this bias is inversely proportional to the average number of samples in each bin (for large u). The debiased entropy is then calculated as $H = H^{\mathrm{B}} + B$, where H^{B} is the naive entropy estimate. This correction can be applied directly to H_{r} using $w_o = \mathrm{W}$ and $u = \mathrm{N}$. In the case of $H_{\mathrm{r|s}}$, we first calculate the entropy bias to each stimulus using $u = \frac{\mathrm{N}}{\mathrm{Q}}$ (i.e., the number of samples for each unique stimulus presentation); one caveat is that w_o should represent the number of bins that *might* be occupied — thus, it is not clear how to treat empty bins (if they happen to occur). Heeding the advice from [75], we again use $w_o = \mathrm{W}$. The justification for this choice is that when the number of trials is sufficiently large, such that each bin has a reasonable chance of being occupied (i.e, $\frac{\mathrm{N}}{\mathrm{Q}} \times p(r_c) \gg 1$), then using all of the code words is a fair (and conservative) estimate. Then, to compute $H_{\mathrm{r|s}}$, we multiply the debiased entropies for each stimulus by $p(s_k)$ and sum the results (following Equation 4.12).

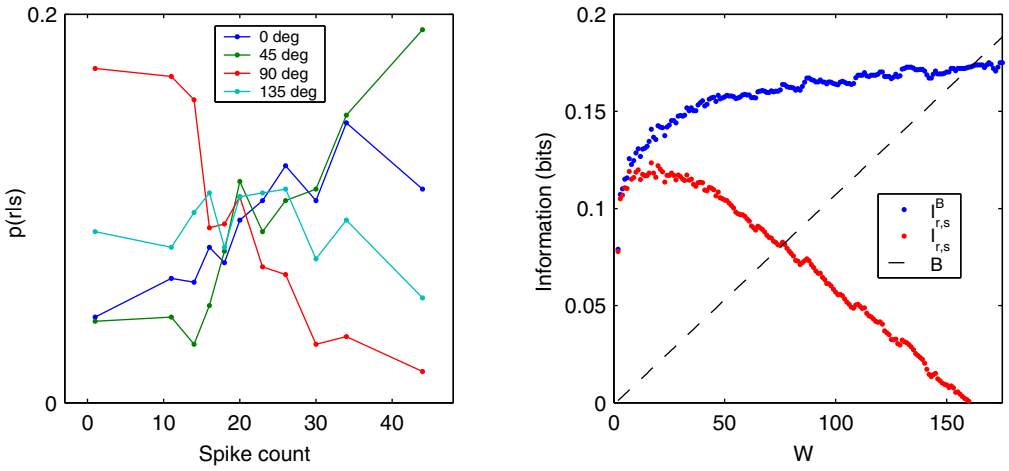
3. Computing mutual information

To compute mutual information, we plug in our debiased entropy estimates into Equation 4.13. For instance, consider a cell that responds to a single cycle of an oriented grating with known probabilities (Figure 4.13a); in this example, the number of total presentations (N) is 2000, the number of orientations (Q) is 4, and the number of

bins (W) is 11. Since the conditional probabilities ($p(r_c|s_k)$) are clearly distinct for the different stimulus conditions, this cell conveys non-zero information about orientation in its responses; specifically, the response entropy (H_{r}) is $3.4527 + 0.0036$ bits (naive + bias), the conditional entropy ($H_{\mathrm{r|s}}$) is $3.3220 + 0.0144$ bits, and the mutual information is $0.1307 - 0.0108$ bits (out a maximum of 2 bits). Note that the downward entropy biases result in an *upward* information bias.

We then vary the number bins (i.e., treat W as our parameter w_o) to discover the maximum amount of information (debiased) that this cell can transmit. The idea behind this strategy is straightforward: Consider adding a code word to a cell's response repertoire (i.e., increasing w_o by 1), and comparing how this additional code word affects the naive information estimate ($I^{\mathrm{B}}_{\mathrm{r,s}}(w_o)$) and the bias ($B(w_o)$). In the case that naive information increases *more* than the bias, then we can deduce that the current value of w_o underestimates the number of code words that this cell can signal; if naive information increases the *same* as the bias, then w_o is a reasonable estimate of the cell's code word resolution; and finally, if naive information increases *less* than the bias, then w_o overestimates the number of code words for this cell.

Using the same cell from our previous example, we plot its naive information ($I^{\mathrm{B}}_{\mathrm{r,s}}(w_o)$), debiased information ($I_{\mathrm{r,s}}(w_o)$), and bias ($B(w_o)$) for values of w_o ranging from 2 to 175 (Figure 4.13b). We observe that $I_{\mathrm{r,s}}(w_o)$ first increases (w_o between 2 and 7), plateaus (w_o between 8 and 30), and then decreases (w_o greater than 30) — the relationship that we expected. Therefore, we estimate the information of this cell as the peak in $I_{\mathrm{r,s}}(w_o)$, which in this case is near 0.12 bits.

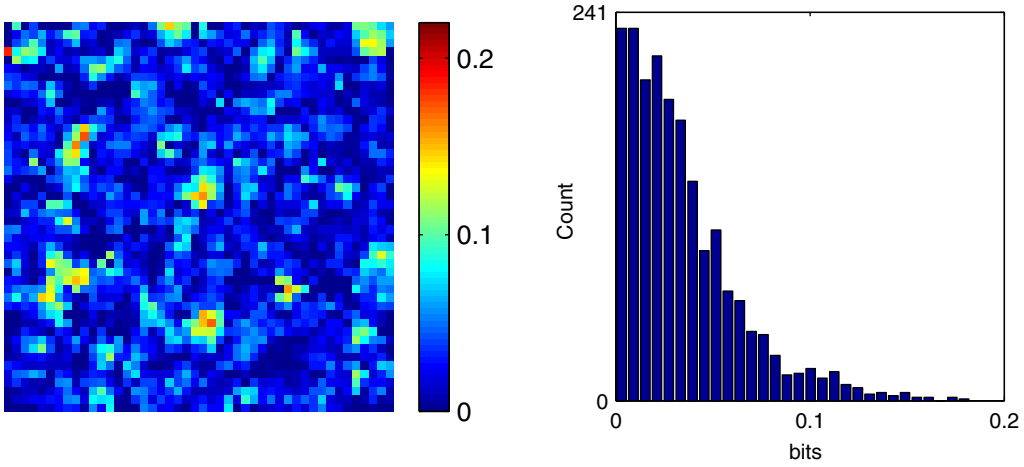


a b

Figure 4.13: PDFs and mutual information for a single cell
(a) Conditional PDFs of a random cell for 4 unique orientations where the number of code words considered is 11 (W); note that the bin edges, which are marked by points, vary in size. (b) Max information as a function of W for the same cell.

4.3.2.3 Results

We apply the procedure developed in Section 4.3.2.2 to compute how much information each cell in our network carries about orientation. In these experiments, we consider four directions from -180 to -45 degrees, where the trial duration (D) is 3 cycles, and the number of trial repetitions (R) is 500; following our standard protocol, we only consider the population of responsive cells. The spatial distribution of information is shown (Figure 4.14a) where again, the unresponsive cells are set to 0. We can see that cell's with similar information values tend to cluster in space, which suggests that information may be related to NVM, and preferred bump locations; this prospect is explored in the following section. We also show the results in a histogram, considering only responsive cells (Figure 4.14b); this distribution has a mean of 0.034 bits, and a median of 0.027 bits.



a b

Figure 4.14: Information map of bump chip
Information distribution across space (a), and as a histogram (b).

4.3.3 Comparing NVM and information

Now that we have measured the information and NVM for each cell, we can now explore their underlying relationship. To uncover how each measure varies across the network, we rank each cell's information and NVM values (independently), and compute the information rank minus the NVM rank (Figure 4.15a); the normalized difference varies from -1 to 1. In this plot, regions of red indicate cells whose information ranks higher than NVM, where regions of blue indicate the opposite; the color saturation is set by the magnitude of the difference (black signifies unresponsive cells). This analysis has uncovered continuous regions (patches) where either information or NVM dominate, interspersed by patches of equal rank (white). Blue patches typically surround unresponsive cells, consistent with the notion that cell's with low firing rates have high NVMs.

We examine the underlying responses for two cells, one that has high information and low NVM (marked by X1 in Figure 4.15a) in the second row of the figure, and one that has

low information and high NVM (marked by X2) in the third row; Figure 4.15(c) and (e) show the tuning curves, where (e) and (f) show the conditional probabilities of the trial-to-trial responses (using 11 bins). The tuning curve of cell X1 shows that, even though the cell is selective for orientation, it responds with similar rates for orthogonal orientations (-180 and -90°). Thus, the NVM is low even though cell is selective. In contrast, the tuning curve of cell X2 reveals that this cell has a single well-defined PO near -10°. However, its information is relatively low since the conditional probabilities (Figure 4.15(f)) have significant overlap. In fact, the majority of its time is spent not firing for all orientations.

We also observe that the mean rate of cell X1 is higher than cell X2, which suggests that the information–NVM relation could also depend on how much a cell participates in bumps. To test this possibility, we have plotted information against NVM, and use color to indicate the mean firing rate (Figure 4.15b); here, the colors are set by their firing rate rank (i.e., the proportion of cells that have lower rates). From this plot, we observe a striking relationship that clearly shows cells with low NVM typically have higher rates (putatively, closer to the center of the bumps), whereas high NVM corresponds to low mean rates; this NVM–firing relationship has been observed in adult macaque V1 [68]. It would be interesting to see if the information–mean rate relationship that we observe also exists in the cortex.

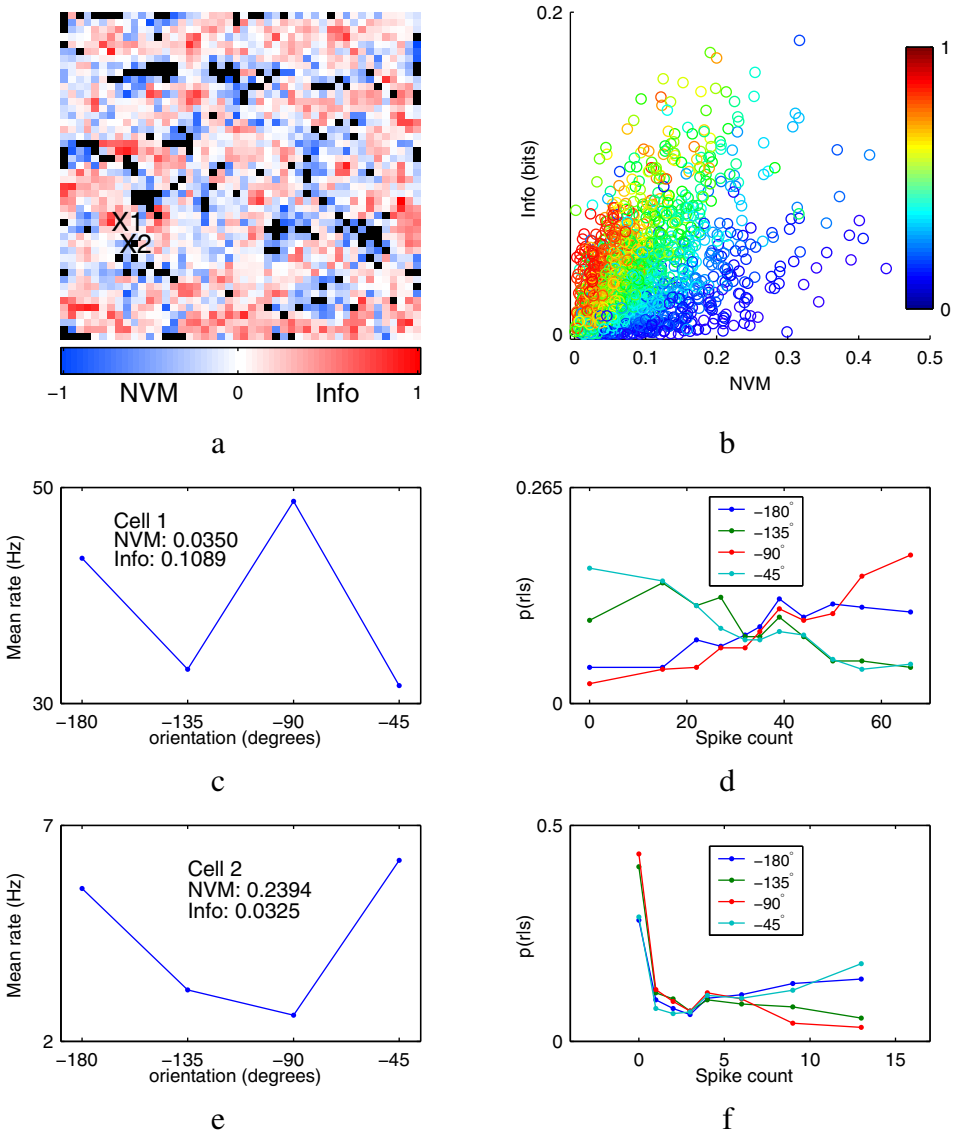


Figure 4.15: NVM vs. Information

(a) Spatial distribution of rank differences between NVM (blue) and information (red); black indicates unresponsive cells. (b) Comparison of NVM, information, and mean rate (indicated by color). Example tuning curve and conditional probabilities for cell X1 (c) and (e), and cell X2 (d) and (f); cells are marked in (a).

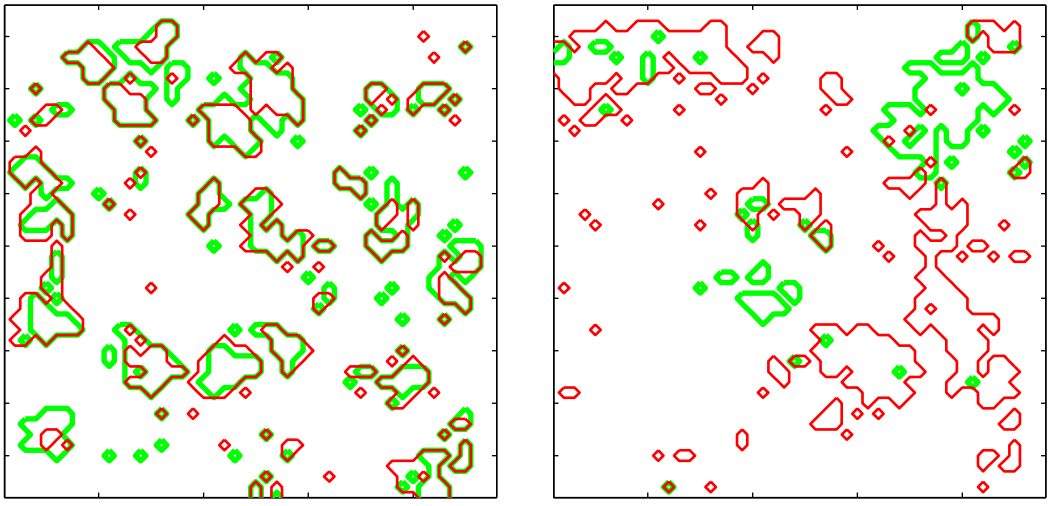
Chapter 5

Characterizing Network Dynamics

In this chapter, we develop techniques for characterizing the complex spatio-temporal dynamics that we observe in our chip model of V1. We will use these techniques in the following chapter to explore how dynamics affect the formation of PO maps.

5.1 Dynamic activity patterns

The neural activity patterns that emerge in our heterogenous recurrent network exhibit complex dynamic behavior: over time, the patterns can change shape, coalesce with neighbors, disappear, or diffuse. These dynamics are qualitatively different when the network is tuned to different parameter regimes. For instance, if we look at two frames of activity 50ms apart when the network has different levels of recurrent feedback (Figure 5.1(a) and (b)), the dynamics are visibly different: in Figure 5.1a, the patterns of activity are mostly immobile except for occasional shifts, while in Figure 5.1b the patterns are more fluid. In this section,



a b

Figure 5.1: Network dynamics

Contour plots of chip activity 50ms apart (green and red contours represent the earlier and later time intervals, respectively) for low recurrent inhibition and excitation (a) and high recurrent inhibition and excitation (b). The contours represent cells that have fired at least 1 spike in a 10ms frame.

we move beyond these informal descriptions, and introduce a method for quantifying *how* these dynamics are different.

5.1.1 Experimental setup

In previous experiments, we were primarily concerned with how vigorously cells responded to a particular stimulus — the precise timing of the responses did not affect our results. However, since we are now interested in characterizing dynamics, this timing information is also important. Accordingly, we have modified our experimental setup.

The first modification is to replace the previous stimuli (drifting gratings), with spatially and temporally uncorrelated spike trains (Poisson statistics). In contrast to drifting gratings, these Poisson spike trains are able to jump start activity in our network without

inducing stimulus-related correlations; such correlations could potentially swamp out the innate network dynamics. The rate of our new stimulus is chosen to be 75Hz, which drives the network with the same mean rate as the gratings. The second modification is that during post-processing, we now keep track of when network activity occurred with a 10ms timing resolution. With this modified setup, a typical 20s experiment produces a 2,000 frame movie of network activity (at 100fps), denoted by $A_{\mathrm{i}}(t)$ where i references a cell's location and t is a frame index representing 10ms time increments. In the subsequent analysis, activity within each frame is capped at five spikes, limiting the instantaneous firing to 500Hz. The network state that we use throughout this section is indicated in Figure 3.5. This state does not have any special significance — it was chosen simply because it exhibits moderate dynamics.

5.1.2 Extracting temporal structure

Our goal is to classify the complex spatio-temporal dynamics that we observe in our network. Here, we outline our general strategy for extracting the temporal structure of quantifiable aspects of network dynamics, and provide a simple example to illustrate the method.

1. **Chose a dynamic quantity**

   The first step is to identify a quantity of interest, derived from network activity, that captures an essential feature of the dynamics. For our present example, we track cell activity within a frame as our quantity, or $A_{\mathrm{i}}(t)$.

2. **Track quantity changes**

   The next step is to construct a difference distribution, which represents the amount our chosen quantity can change over a given time interval. For this example, we

compute the total absolute difference of cell activity between frames separated by the interval to. We notate this distribution by $\sum_{\mathrm{i}} |\Delta| a_{\mathrm{i}}(\mathrm{to})$; here, $a_{\mathrm{i}}(\mathrm{to})$ represents all the instances of $A_{\mathrm{i}}(t)$ that are separated by interval to (denoted by a lowercase naming convention), Δ is the traditional difference operator, and the sum is computed across cells. To illustrate how activity between frames changes over a 10ms time interval, we plot the $\sum_{\mathrm{i}} |\Delta| a_{\mathrm{i}}(10\mathrm{ms})$ distribution (Figure 5.2a).

3. Extract temporal structure

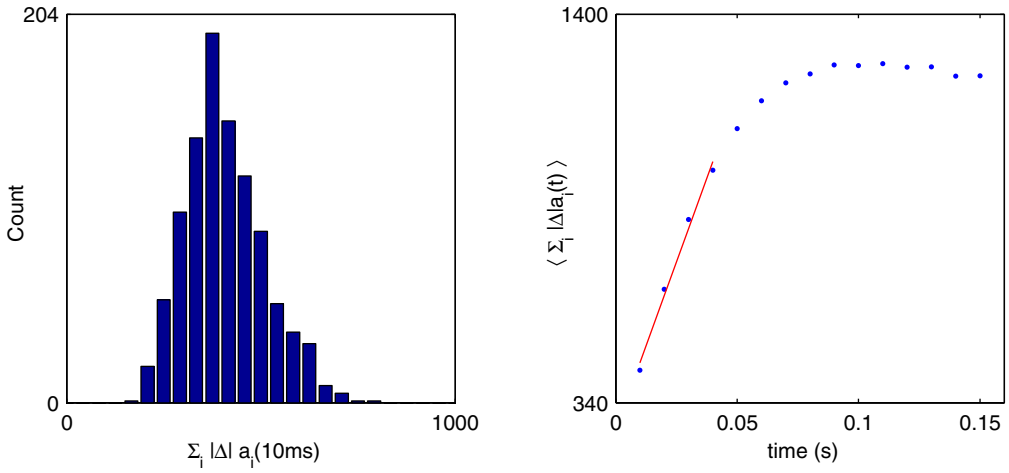
Next, we characterize how the difference distribution depends on the time interval to. Our intuitive notion is that in a well-behaved system, most quantities remain relatively constant over short time intervals, and drift apart monotonically as time increases. We explore this relationship by plotting the mean of the difference distribution for different times, or in our example, $\langle \sum_{\mathrm{i}} |\Delta| a_{\mathrm{i}}(t) \rangle$ (Figure 5.2b).

4. Fit time dependence

The final step is to fit the empirically discovered temporal structure for our quantity using a simple functional form. In our example, we use a piece-wise linear function with two pieces: the first piece extends from 0 to 70ms and has a slope of xx spks/s (red line, Figure 5.2b), and the second piece extends from 80 to 150ms and has a constant value of yy spks/s (fit not shown). The values of these fits can be subsequently used for classifying the dynamics.

5.1.3 Interpreting $\langle \sum_{\mathrm{i}} |\Delta| a_{\mathrm{i}}(t) \rangle$

In our previous example, we demonstrated that the expected absolute difference of cell activity between frames, or $\langle \sum_{\mathrm{i}} |\Delta| a_{\mathrm{i}}(t) \rangle$, has underlying temporal structure. We would



a b

Figure 5.2: Temporal structure of frame activity
(a) The difference distribution of total absolute activity for frames separated by 10ms. (b) The mean of these distributions for different time intervals.

like to relate this temporal structure to how the individual patterns (bumps) in the network move over time, since these component motions are the basis for the complex dynamics that we observe. We show in this subsection that although tracking cell activity provides insight into global network dynamics, it does not have a straightforward interpretation for individual bump motion.

To understand how $\langle \sum_i |\Delta| a_i(t) \rangle$ relates to network activity, we can think of it as estimating the global activity change between any two frames. From this perspective, the temporal structure of $\langle \sum_i |\Delta| a_i(t) \rangle$ has the following interpretation: activity within frames will become progressively different over 70ms, at which point their differences will reach a plateau (Figure 5.2b). In other words, we can predict the time difference between two randomly chosen frames simply by measuring the global change in their patterns; however, if the frames occurred more than 70ms apart, we are unable to predict the additional time difference (i.e., 90ms is indistinguishable from 90s).

The interpretation of $\langle \sum_{\mathrm{i}} |\Delta| a_{\mathrm{i}}(t) \rangle$ in terms of dynamic global activity patterns is clear, however, the relation between this global activity and individually moving bumps is ambiguous. For example, $\langle \sum_{\mathrm{i}} |\Delta| a_{\mathrm{i}}(t) \rangle$ is insensitive to one bump transporting a large distance or many bumps diffusing a small distance between any two frames; both cases would result in similar $\langle |\Delta| a_{\mathrm{i}}(t) \rangle_i$ distributions. Since there is no *a priori* constraint on how bumps should behave, and in fact, non-diffusive behavior is often evident (Figure 5.1b), we are unable to attribute the temporal structure to physically-rooted motion.

To shift our focus from the dynamics of global activity to the motion of individual bumps, we need to track bump-derived quantities explicitly. In the following section, we develop techniques to extract the temporal structure of relevant bump quantities. Specifically, we measure the rate at which bumps diffuse in our network, using a modified version of the procedure outlined in Section 5.1.2. This brings us one step closer to understanding how bumps move in our heterogeneous recurrent network.

5.2 Dynamic bumps

As we demonstrated in the previous section, the activity patterns in our network have underlying temporal structure, however, a straightforward interpretation of this temporal structure in terms of moving bumps was not possible. To uncover the temporal structure of bumps, we need to explicitly identify bumps within frames and track their motion between frames. Identifying bumps within a frame is tractable; tracking bumps between frames is more difficult. Instead of developing a sophisticated motion detection algorithm, we employ a clever technique that can estimate consistent bump motion. This technique enables us to draw parallels between bump motion and particle motion from ideal matter states, and

perhaps more important, how these parallels diverge.

5.2.1 Identifying bumps

The process of identifying bumps can be relatively easy if the appropriate assumptions are made. The first step is to find a relevant time frame for identifying bumps that is long enough to resolve a bump's shape yet short enough so that a bump's motion is not smeared: we use a 10ms frame, as mentioned in Section 5.1.1, which was determined by trial and error (ultimately a subjective decision). Second, the concept of a bump needs to be explicitly defined: we consider activity to be part of a bump if at least half of its neighbors are simultaneously active within a frame. The final step is to develop an algorithm that is capable of identifying bumps: we implement a space-filling algorithm that labels contiguous regions of activity which meet our bump definition; the set of cells that fall within this contiguous region are denoted by $p(j)$, where j is the bump label.[1] Then, each bump is defined as

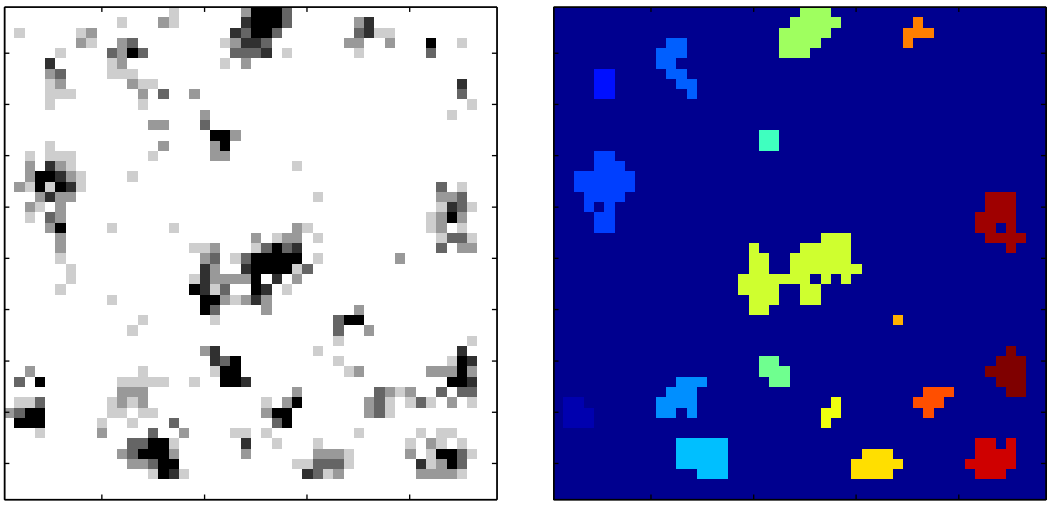
$$B_{i,j}(t) = \begin{cases} A_i(t) & \text{if } i \in p(j), \\ 0 & \text{otherwise.} \end{cases} \tag{5.1}$$

Finally, we decompose each frame of activity (Figure 5.3a) into

$$Ad_{\mathrm{i}}(t) = \sum_{j=1}^{N(t)} B_{\mathrm{ij}}(t) \tag{5.2}$$

(Figure 5.3b) where $N(t)$ is the total number of identified bumps in frame t. Note that the

[1] Notice that our bump labeling procedure routinely discards activity that does not have active neighbors, which is different from the traditional 4-connected and 8-connected algorithms; these standard algorithms maintain a one-to-one mapping with activity.



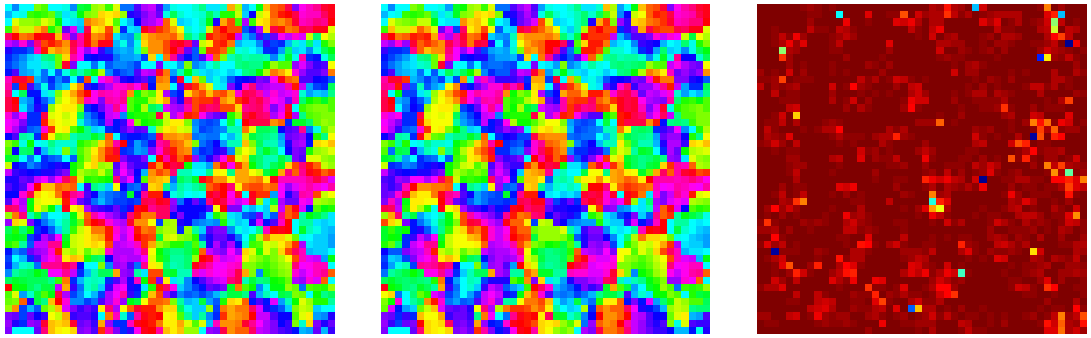
a b

Figure 5.3: Bump identification
A typical frame of cell activity (a) and its corresponding identified bumps (b), where each bump is marked by its unique integer label.

number of bumps identified for each frame can vary.

Of course, our bump identification process has some drawbacks. One issue is that when two nearby regions of contiguous activity are connected for brief periods of time (through thin activity bridges), our algorithm will classify this pattern as one bump instead of two distinct bumps. It could be argued that this configuration should be classified as one bump — it boils down to semantics. Even with this issue, we believe that our method for identifying bumps is superior to conventional dimension reduction algorithms, such as PCA, and ICA. In particular, the results from these algorithms (basis vectors) are not constrained to have a one-to-one mapping with bumps.

As a check to ensure that bumps, as we have defined them, are in fact the primary mechanism for creating PO maps, we have re-analyzed data from a previous PO map experiment (Figure 4.8) in the following way: First we recast our data into high-temporal resolution format. Next we filter the data by identifying bumps as described in this section. Finally,



a b c

Figure 5.4: Affect of bump identification on PO
PO map computed from raw data (a) and identified bumps (b), and their corresponding SI (c).

we recompute the PO map only considering the bump data (Figure 5.4a), and compare the similarity between this map and the original PO map (Figure 5.4b). The similarity between these two maps (Figure 5.4c) is comforting, not only because it validates that our bump identification algorithm does not discard important information, but also because it justifies our assumption that bumps capture the quintessential network behavior.

5.2.2 Emergent features

Bump identification allows us to characterize regions of contiguous network activity as if they were their own unique objects. From these objects emerge a wide array of quantifiable features — some of which can be used measure bump motion. Here, we describe the quantities that we keep track of; our strategy is to capture enough first-order information about each bump so that if we wanted to, we could reconstruct a crude representation of network activity. From this perspective, our procedure is essentially a dimension reducing technique. However, unlike other dimension reduction algorithms, our method is purely descriptive and is not constrained to represent the original data set optimally.

We derive the following quantities from bumps[2]:

1. Centroid

The centroid for bump j is defined by the pair of points:

$$R_{\{y,x\}_\mathrm{j}}(t) = (\sum_{i=1}^{\mathrm{N_c}} B_\mathrm{ij}(t))^{-1} \sum_{i=1}^{\mathrm{N_c}} \{y,x\}_\mathrm{i} \times B_\mathrm{ij}(t), \tag{5.3}$$

where $\{y,x\}_\mathrm{i}$ are the y and x coordinates for cell i, and $\mathrm{N_c}$ are the number of cells in the network. Note that the magnitude of activity within each bump influences the calculation (i.e., we are effectively computing the center-of-mass), but at most by a factor of 5 since activity is capped at 500Hz. We use the centroid extensively in subsequent analysis since it can be used to track how far bumps move in a given time interval (e.g., diffusion rates).

2. Area

The area for bump j is simply calculated as:

$$S_\mathrm{j}(t) = \sum_{i=1}^{\mathrm{N_c}} \min(B_\mathrm{ij}(t), 1), \tag{5.4}$$

where we use the $\min$ function to transform B_ij to a binary representation; this transformation ensures that each cell contributes equally to the area, since we are only interested in the bump's shape, and not its density.

3. Orientation

Each bump, if not circularly symmetric, will have an axis of orientation when fit by

[2]In practice, all of these quantities are computed using the built-in routines from the matlab image processing toolbox.

an ellipse. In our analysis we keep track of the orientation of the ellipse's major axis with respect to the cardinal axis.

4. Eccentricity

Eccentricity is a measure of how much a bump deviates from a circular approximation that varies from 0 (perfect circle) to 1 (line). The eccentricity is computed as the ratio of the distance between the foci of the ellipse and its major axis length (both of which are fit parameters). Currently, we do not use eccentricity in any of our subsequent analyses; we make a point to include it, however, because it is required to 'reconstruct' network activity from our derived bump quantities.

Therefore, a frame is represented by N groups of these four quantities (one for each bump). This can result in a significant data reduction since 48×48 points are now represented by $N \times 4$ quantities per frame.

5.2.3 Estimating consistent motion

Identifying bump quantities within a frame is relatively easy — tracking them between frames to estimate motion, however, is more difficult. The main difficultly is that the more bumps change, the less certain we are about which bump is which. This problem is exacerbated when bumps behave less like billiard balls and more like quantum particles. For example, if a bump disappears or transports — or if two (or more) bumps coalesce or annihilate — it is not clear how to measure bump motion since a bump in one frame may not have a direct counterpart in subsequent frames. We solve this difficult bump tracking problem by only considering dynamics that are consistent, inspired by a strategy used in physics experiments.

Physicists have developed a technique that is capable of estimating the mean-squared displacement of particles[3] over time without tracking individual particle motion explicitly. Their brute-force strategy is to compute all of the pair-wise distances between particles separated by a given time interval; the distances that occur most often (region around the peak of the distribution) corresponds to the correct displacement. In other words, if particles consistently travel a certain distance over a given time interval, then these consistent displacements (signal) will out number the inconsistent displacements (noise). Of course, if the distances between neighboring particles are also correlated, multiple peaks will exist.

This same technique can also be used to estimate bump motion — as long as the motion type (quantum or classical) contributes to an identifiable peak in the pair-wise distance distribution. For example, if the quantum-like behavior is spurious, it will be automatically segregated away from this peak; however, if the quantum-like behavior results in consistent bump motion, then it will contribute to either the main peak or an ancillary peak. It is worthwhile to point out, however, that these quantum behaviors can also potentially complicate our analysis; we explore these complications in detail in Section 5.3. Regardless of this caveat, the strategy that we have described provides a simple method for measuring bump motion, without the need for explicitly tracking bumps. In the next section, we employ this method to measure the temporal structure of local bump displacement.

5.2.4 Extracting local displacements

Now that we are capable of identifying bumps (Section 5.2.1) and estimating their motion (Section 5.2.3), our current goal is to extract the temporal structure of an emergent bump

[3]The particles in these physics experiments are large enough such that quantum effects can be ignored (they are typically beads in a viscous liquid). Accordingly, all further descriptions of particles refer to classical motion unless noted otherwise.

feature. Taking our cue from the physicists, we have decided to estimate the mean-squared displacement of bumps with the intention of ultimately drawing parallels between moving bumps and moving particles — and highlighting where the parallels diverge.

To measure the temporal structure of the mean-squared displacement of bumps, we follow the procedure from Section 5.1.2.

1. Compute bump centroids

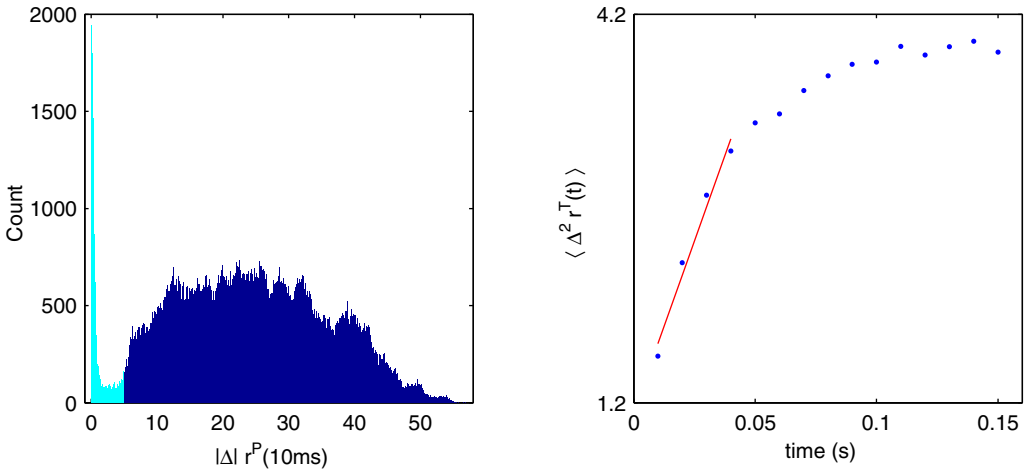
The first step is to identify bumps within frames (Section 5.2.1) and compute their centroids (Section 5.2.2), denoted by $R_{\{y,x\}_\mathrm{j}}$.

2. Track centroid changes

Next, we calculate the pair-wise distances of all the centroids separated by time to. For this calculation, we construct the $\Delta^2 r^\mathrm{P}(\mathrm{t}_0))$ distribution, where $r^\mathrm{P}(\mathrm{t}_0)$ represents all of the possible pair-wise combinations of centroids separated by time to (the superscript P refers to pair-wise); accordingly, this distribution consists of $\prod N(t)N(t+\mathrm{to})$ samples (recall that $N(t)$ is the number of bumps within a frame). To illustrate a typical distribution, we plot $|\Delta| r^\mathrm{P}(10\mathrm{ms})$ (Figure 5.5a). For our subsequent analysis, we only consider the first peak of this distribution (lighter region). By only considering this initial region, we are in essence grouping pairs of bumps together based on the condition that the bumps exist near each other at different times. A more detailed interpretation of $|\Delta| r^\mathrm{P}(10\mathrm{ms})$'s interesting shape is discussed in Section 5.2.5.

3. Extract local displacements

The next step is to characterize how the mean-squared displacement depends on the time interval, or $\langle \Delta^2 r^\mathrm{T}(t) \rangle$ (Figure 5.5b). Here, $r^\mathrm{T}(t)$ corresponds to the subset of bump pairs that are $\leq \mathrm{T}$ pixels from each other, where the value of T is chosen to



a b

Figure 5.5: Temporal structure of bump displacements
(a) Pairwise displacements between all bumps in frames separated by 10ms ($|\Delta| r^{\mathrm{P}}(10\mathrm{ms})$); we plot the absolute difference ($|\Delta|$) because it is easier to visualize. Bump displacements $\leq$ T (lighter region) correspond to bumps from $r^{\mathrm{T}}(t)$. (b) The mean-squared displacement for different time intervals ($\langle \Delta^2 r^{\mathrm{T}}(t) \rangle$); the linear region corresponds to the diffusion constant.

contain the initial peak up to its trough (4 pixels). The implication of restricting our analysis this initial region is described in detail in the following sections.

4. Fit local motion

Finally, we explore the structure of $\langle \Delta^2 r^{\mathrm{T}}(t) \rangle$ by fitting it with a two-piece linear function: the first piece extends from 0 to 70ms and has a slope of xx spks/s, and the second piece extends from 80 to 150ms and has a constant value of yy spks/s (fits are shown in Figure 5.5b).

5.2.5 Confined bump motion

To interpret the temporal structure of $\langle \Delta^2 r^{\mathrm{T}}(t) \rangle$, we again look to physics for insight. Our strategy is to draw parallels between moving particles in matter to moving bumps in our network. For example, the mean-squared displacement of particles in gaseous, liquid, and solid states of matter have characteristic shapes (Figure 5.6) that reflect different types of particle interactions. Specifically, in a gaseous state $\langle \Delta^2 r(t) \rangle$ has a quadratic time dependence, characteristic of non-interacting particles; in a liquid state $\langle \Delta^2 r(t) \rangle$ has a linear time dependence (the slope is proportional to the well-known diffusion constant of the liquid), characteristic of moderate particle interactions; and in a solid state $\langle \Delta^2 r(t) \rangle$ has a linear time dependence that saturates, characteristic of confined particle motion (i.e., strong particle interactions).

The temporal structure of $\langle \Delta^2 r^{\mathrm{T}}(t) \rangle$ for bumps (Figure 5.5b) resembles that of a solid, suggesting that the motion of bumps are similar to particles tethered by springs. From this perspective, the initial linear region (10 to 60ms) represents the mean squared distance a bump will travel, which is proportional to how long we wait (up until ≈ 60ms); the slope of this linear region (D) is the well-known diffusion constant. For longer time intervals (> 90ms), the expected distance tapers off (near a value of S). This saturating behavior is indicative of a confined bump; it will never travel beyond a certain distance regardless of how long we wait. However, we must consider the possibility that $\langle \Delta^2 r^{\mathrm{T}}(t) \rangle$ saturates because we only use the initial part of the distribution (lighter region of Figure 5.5a) to compute the mean; that is, by ignoring all bump displacements greater than 16 pixels^2, we are artificially inducing the saturation. We rule out this possibility because S has a value that is significantly smaller than the 16 pixel^2 cutoff.

The multiple humps observed in the $|\Delta| r^{\mathrm{P}}(\mathrm{t}))$ distribution (Figure 5.5a) offer further

evidence that, in this network state, bumps diffuse in a confined area. The initial peak is the result of bumps moving a small distance (or not moving at all), whereas the nearby trough represents distances where bumps are unlikely to travel. The k subsequent peaks correspond (at least in part) to the mean distance between k-th nearest neighbors; these peaks exist because on average, neighboring bumps are consistently the same distance from each other.[4] It becomes clear now why we must restrict ourselves to only the initial region when using this physics-inspired technique — otherwise the mean calculation would be corrupted with neighboring bump correlations.

At this point, we might be tempted to draw a strong analogy between bump dynamics and particles vibrating in a crystalline lattice — but we must be careful when invoking such a parallel. One caveat is that the mechanism of bump diffusion could be fundamentally different from confined particle motion, even though both bumps and confined particles have similar $\langle \Delta^2 r(t) \rangle$ shapes. For example, bumps could be popping in to and out of existence and the appearance of confined motion is simply a consequence of this non-diffusive behavior. Alternatively (or simultaneously), bumps could be expanding and contracting (unlike particles), which could also account for some of the observed diffusion. A second caveat is that by only considering the initial region of the $\Delta^2 r^{\mathrm{P}}(t)$ distribution, we could be ignoring important dynamics that happen to fall outside of this initial region. For example, bumps that disappear and reappear a distance greater than T (e.g., they teleport), could be completely ignored in our $\langle \Delta^2 r^{\mathrm{T}}(t) \rangle$ calculation. In the following section, we attempt to characterize these higher-order bump behaviors, building on the techniques that we developed in this section. By characterizing how bump motion is different from particle motion, we can develop a better understanding of how bumps move in our network.

[4] We verified that these subsequent peaks occur at locations similar to the $|\Delta| r^{\mathrm{P}}(0)$ distribution (data not shown).

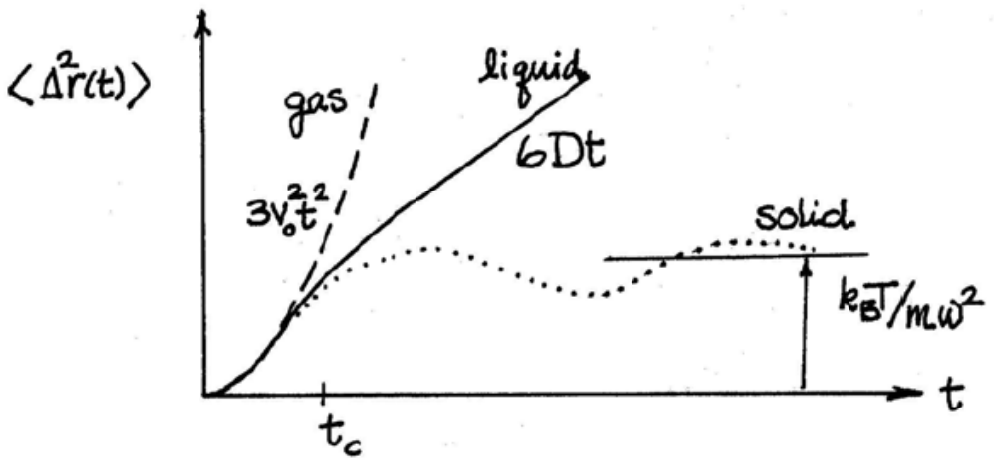
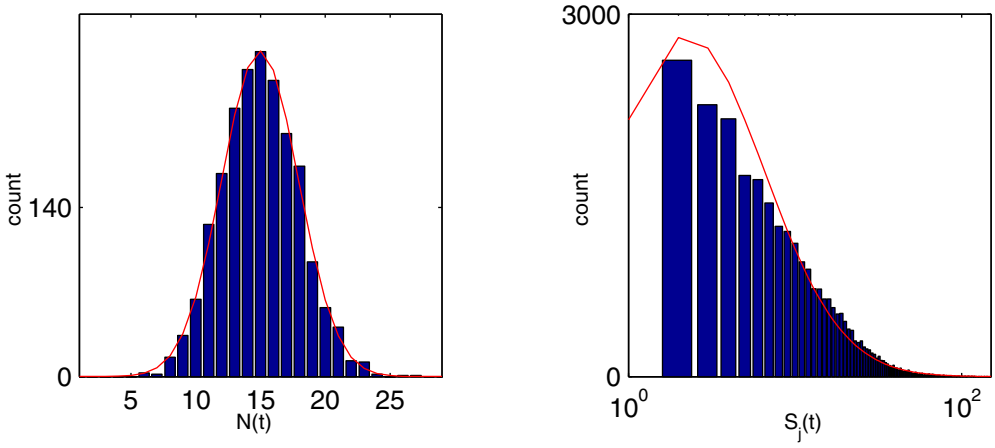


Figure 5.6: $\langle \Delta^2 r(t) \rangle$ for ideal matter states

5.3 Non-classical bump motion

Our concern that bump motion is fundamentally different from particle motion arises from two observations. The first observation is that the number of bumps, $N(t)$, varies considerably from frame to frame (Figure 5.7a) due to bumps vanishing, coalescing, and splitting (we refer to these behaviors collectively as being *non-diffusive*). The second observation is that the size of these bumps, $S_{\mathrm{j}}(t)$ also varies considerably (Figure 5.7b). Since both of these behaviors deviate from the traditional notion of classical particle behavior, we refer to them as *non-classical*. In the following sections, we characterize these non-classical behaviors so that we can ultimately determine their role in bump dynamics.

Our plan of attack is threefold: First, we verify that non-classical behaviors are relevant on the same time scale as bump diffusion (Section 5.3.1). This analysis confirms that both $N(t)$ (the number of bumps per frame) and $S_{\mathrm{j}}(t)$ (the size of bumps) do in fact vary on a 150ms time scale, and justifies further analysis on them. Second, we explore how these behaviors relate to confined bump movements, or our $\langle \Delta^2 r^{\mathrm{T}}(t) \rangle$ result (Section 5.3.2).



a b

Figure 5.7: Evidence of non-classical behaviors
(a) Measured distribution of the number of bumps per frame and its fit to a normal dist (mean=15.0, std=3.0). (b) Measured distribution of bump areas and its log-normal fit (mean=11.7, std=12.6).

This is accomplished by directly examining the sets of bumps used to calculate diffusion (i.e., the $r^{\mathrm{T}}(t)$ distributions), and classifying their non-classical characteristics. Finally, we explore the possibility that there is a class of bump motion that is ignored in our current $\langle \Delta^2 r^{\mathrm{T}}(t) \rangle$ result, since the only signature of this motion lies beyond the T cutoff (Section 5.3.3). We dubb this yet uncharacterized motion type as *staccato*, since it reflects sudden and disjointed bump movements, and present a method to quantify its extent.

5.3.1 Time course of non-classical behaviors

In this section, we show that both $N(t)$ (the number of bumps per frame) and $S_{\mathrm{j}}(t)$ (the size of bumps), change on a 150ms time scale — our concern is that if these variables changed at a slow pace, they may not play a significant role in bump dynamics. Our analysis is not meant to be exhaustive, but instead provides a cursory look at the relevant time course of

these behaviors.

5.3.1.1 Time course of $N(t)$

Here, we gauge the pace at which $N(t)$ changes using two simple techniques. The first technique is to plot $N(t)$ along with its normalized autocorrelation (top and bottom plots of Figure 5.8a, respectively). These results indicate that the number of bumps within each frame does indeed vary over the first 100ms, and that $N(t)$ has a characteristic temporal structure. However, we find it difficult to gauge the serial structure of $N(t)$ from these two plots. Our second technique is to construct a 2D histogram of the joint distribution $n(t, t + \mathrm{to})$ for two values of to.[5]

To help interpret $n(t, t + \mathrm{to})$, consider the scenario where vanishing (or regenerating) bumps is rare and the number of bumps between frames remains relatively constant. In this case, the majority of points in the 2D histogram will lie near the diagonal, since the number of bumps in one frame corresponds to a similar number of bumps in later frames. Now consider the case where changes in $N(t)$ occur often (or equivalently, the time between frames is long enough so that the rare non-diffusive behaviors aggregate). Here, we expect that the 2D histogram will be circularly distributed, since the number of bumps in one frame will only be loosely correlated with the count in later frames. In our network, both types of distributions are observed, depending on the interval between frames. For a 10ms time interval (Figure 5.8a) the distribution is elongated, indicative of small to moderate non-diffusive behavior; for a 40ms time interval (Figure 5.8b) the distribution is circular, indicative of more pronounced non-diffusive behavior. This latter result clearly shows that non-diffusive behaviors are not only present on a short time scale, but perhaps

[5] We use the same lowercase notation as before (i.e., $n(\mathrm{to})$ represents all the instances of $N(t)$ separated by the interval to), except now, we index these instances by t.

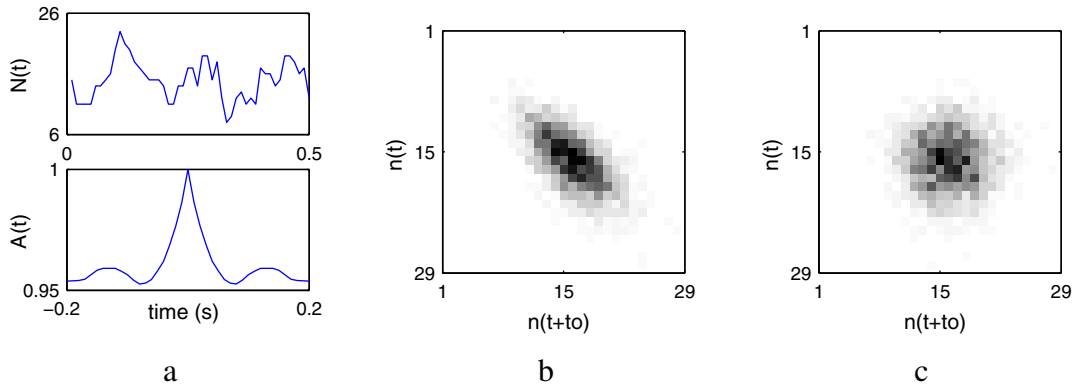


Figure 5.8: Time course of $N(t)$
(a) A 500ms snippet of the $N(t)$ time series (top) and the normalized autocorrelation of $N(t)$ (bottom) computed from 20s of data. Joint distribution of $n(t, t+\mathrm{to})$ where to is 10ms (b) and 40ms (c).

even dominant.

5.3.1.2 Time course of $S_{\mathrm{j}}(t)$

Determining if $S_{\mathrm{j}}(t)$ changes on a 150ms time scale is a more difficult task (since we do not know which bumps in one frame correspond to which bumps in a later frame). One possibility is to apply the procedure that we outlined in Section 5.2.4 and look for consistent size changes in the $s_{\mathrm{j}}^{\mathrm{P}}(t)$ distribution. It turns out, however, that this approach is not suitable for bump sizes; specifically $|\Delta| s_{\mathrm{j}}^{\mathrm{P}}(t)$ does not have an identifiable peak (see Figure 5.9). In other words, it is not possible to clearly identify which bump corresponds to which bump based solely on changes in area. In the following section, we will explore a technique that allows us to circumvent this problem; the basic idea is that we can use the subset of bumps previously determined from Figure 5.5b, and probe how the area of this subset changes over time.

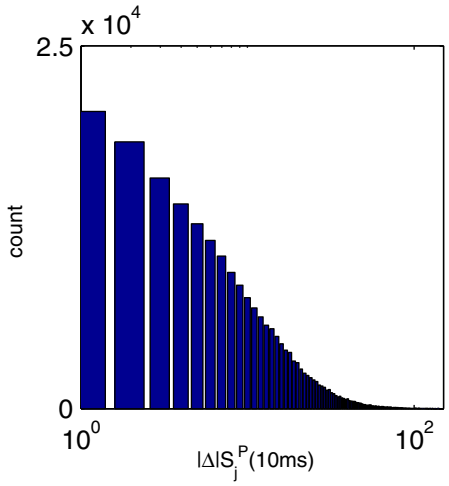


Figure 5.9: Distribution of absolute area changes
The lack of an identifiable peak suggests that changes in bump area are not consistent between frames.

5.3.2 Quantifying non-classical behaviors

Thus far, we have shown that non-classical behaviors exist (Figure 5.7), and that they are relevant over short time intervals (Figures 5.8 and 5.9), thereby confirming that bump motion is fundamentally different from confined particle motion. In this section, we extract the temporal structure of these behaviors, and quantify their role in bump motion, using a simple yet powerful strategy. Instead of starting our analysis from scratch, which would entail searching for temporal structure with no *a priori* information (a hard if not impossible task, as shown in Figure 5.9a), our strategy is to simply use the previously identified bump pairings from our bump diffusion calculations (i.e., the bump pairs from the $r^{\mathrm{T}}(t)$ distributions). Using these pairings, we are then able to directly quantify the non-classical behaviors. In particular, we calculate the likelihood that 0, 1, or more than 1 bump exists near an original at different time lags (i.e., do bumps tend to disappear, remain the same, or split) (Section 5.3.2.1), and the expected amount a bump will change in size over time (Section 5.3.2.2). This strategy is not only practical, since we have already calculated all of

the necessary bump pairings, but it also allows us to uncover their behaviors with minimal assumptions about which bumps are the 'same' across frames.

Before diving into our analysis, we recap how bumps in different frames are paired together based on their geometry (a review of Section 5.2.4). The algorithm can be understood in the following steps: First, a circle is drawn around the center of each bump (called an *original* bump) such that the circle's radius is determined by the cutoff T from the $\langle \Delta^2 r^{\mathrm{T}}(t) \rangle$ distribution. By choosing T to contain the initial region up until the second peak (Figure 5.5a), we are in essence allowing the circle to be as large as possible without encompassing neighboring bumps. Next, we keep track of the bumps that subsequently fall within this circle, pairing them together (i.e., the original and each subsequent bump) based on their time lag. We refer to this distribution as $r_{\mathrm{j}}^{\mathrm{T}}(t)$, where j references the original bump, and T indicates that only bumps that fall within bump j's circle are considered; note that $r_{\mathrm{j}}^{\mathrm{T}}(t)$ also keeps track of the instances when no bumps fall within the circle. Based on these bump pairings, we are then able to classify their non-classical behaviors (i.e., do bumps vanish, change in size, etc.).

5.3.2.1 Isolating non-diffusive and diffusive behaviors

Here, we segregate non-diffusive and diffusive behaviors of moving bumps by calculating the likelihood that 0, 1, or more than 1 bump exists near an original bump at different time lags: Non-diffusive behavior is identified by keeping track of when bumps vanish and reappear near an original bump, whereas diffusive behavior is identified by tracking bumps that exist sequentially in time near an original bump.

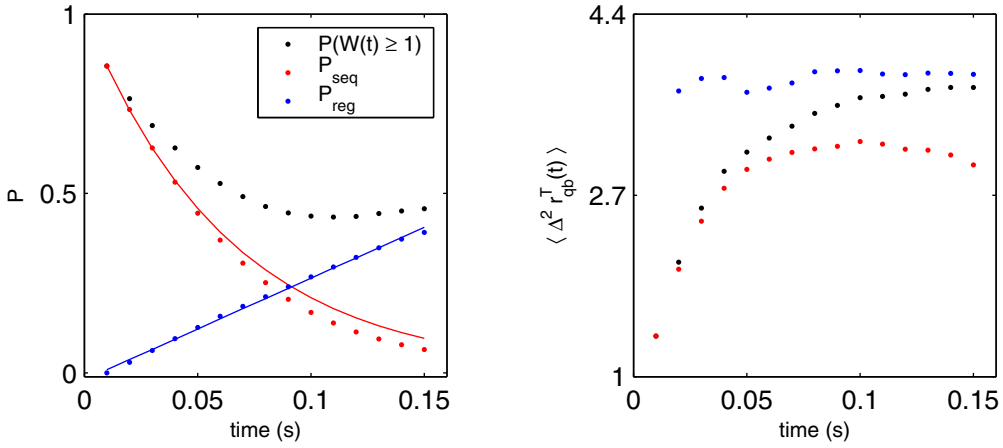
To compute the likelihoods, we construct the space $W(t, \mathrm{j}) = \Gamma(r_{\mathrm{j}}^{\mathrm{T}}(t))$, where Γ speci-

fies a counting operator that returns the number of bumps that are within T pixels of bump j; the argument j is omitted if all bumps are to be considered in this space. $W(t)$ is then used to calculate the probability that bumps tend to vanish, $P(W(t) = 0)$, remain $P(W(t) = 1)$, or split, $P(W(t) > 1)$, for different time lags. We also explore the probability that a bump existed sequentially, $P_{\mathrm{seq}}(t)$, or was regenerated, $P_{\mathrm{reg}}(t)$. After segregating each bump into its appropriate space, we are then able to characterize the mean-squared displacement of the different behaviors separately. Specifically, we calculate $\langle \Delta^2 r^{\mathrm{T}}_{\mathrm{qb}}(t) \rangle$, where qb indexes the subset of the bump pairs that exhibit a specific type of behavior (e.g., if qb is P_{seq}, then only bumps that existed sequentially are considered in the mean calculation).

The first probability that we calculate, $P(W(t) > 1)$, is practically nil (data not shown). This perhaps surprising result suggests that bumps rarely split, or if they do split, the fragmented pieces do not fall within the original bump's circle. Taking this into account, we instead compute $P(W(t) \geq 1)$ (black dots in Figure 5.10a), which automatically implies $P(W(t) = 0)$ since $P(W(t) \geq 1) = 1 - P(W(t) = 0)$.

The temporal structure of $P(W(t) \geq 1)$ decreases from 0.88 to 0.44 over the first 100ms, which suggests that bumps tend to disappear over this time interval (i.e., P(W(t) = 0) increases from 0.22 to 0.66); notably, this trend reverses direction after the 100ms mark. This reversal highlights the fact that $P(W(t) \geq 1)$ encompasses both bumps disappearing and reappearing (degenerative and regenerative processes, respectively). To tease apart these different processes, we explore the probability that at least one bump existed within bump j's circle in the last t frames (i.e., bumps exist sequentially), which is calculated as:

$$P_{\mathrm{seq}}(t) = P(W(10\mathrm{ms}, \mathrm{j}) \geq 1 \ \ \& \ \ \ldots \ \ \& \ \ W(t, \mathrm{j}) \geq 1) \tag{5.5}$$



a b

Figure 5.10: Temporal structure of behavior types
(a) The probability that at least one bump exists near an original (black points), bumps exist sequentially (red points), and bumps are regenerated (blue points) for different time lags. (b) Mean-squared displacement segregated by the corresponding behavior types (qb).

(red dots in Figure 5.10a). A simple Markov process $M(t) = P_{\text{seq}}(10\text{ms})^{t/0.01}$ (red line in Figure 5.10a) fits P_{seq} fairly well (mean-squared error is 0.003), indicating that the probability a bump exists in a given frame is mostly determined by the likelihood that it existed in the previous frame. We also explore the likelihood that a bump vanished at least once extending back an interval of t (i.e., the bump is regenerated), which is calculated as:

$$P_{\text{reg}}(t) = P(W(10\text{ms}, \text{j}) = 0 \mid \ldots \mid W(t, \text{j}) = 0) \tag{5.6}$$

(blue dots in Figure 5.10a); a linear fit for $P_{\text{reg}}(t)$ is also shown, which is in good agreement. Note that $P(W(t) \geq 1) = P_{\text{reg}}(t) + P_{\text{seq}}(t)$, since every bump has either been regenerated, or has existed sequentially.

At this point, we have successfully classified the different behavior types, and are now in a position to uncover their relation to our previous diffusion results. First, we explore

the mean-squared displacement of bumps that behave like classical particles (i.e., bumps that exist sequentially), denoted by $\langle \Delta^2 r^{\mathrm{T}}_{\mathrm{P_{seq}}}(t) \rangle$ (red points in Figure 5.10b). For comparison, we also plot $\langle \Delta^2 r^{\mathrm{T}}(t) \rangle$ (black points in Figure 5.10b), which also corresponds to $\langle \Delta^2 r^{\mathrm{T}}_{\mathrm{P(W(t)\geq 1)}}(t) \rangle$. The shape of $\langle \Delta^2 r^{\mathrm{T}}_{\mathrm{P_{seq}}}(t) \rangle$ is similar to $\langle \Delta^2 r^{\mathrm{T}}(t) \rangle$ — both consist of a linear time dependence that saturates, matching the stereotyped shape of confined motion. One difference, however, is that $\langle \Delta^2 r^{\mathrm{T}}(t) \rangle$ has a small but distinct positive slope past 70ms, whereas $\langle \Delta^2 r^{\mathrm{T}}_{\mathrm{P_{seq}}}(t) \rangle$ fully saturates. The origin of this excessive diffusion becomes apparent when we observe that the mean-squared displacement of regenerated bumps, or $\langle \Delta^2 r^{\mathrm{T}}_{\mathrm{P_{reg}}}(t) \rangle$ (blue points in Figure 5.10b) is constant. This constant displacement, combined with the fact that the probability of observing a regenerated bump increases linearly with time (blue points in Figure 5.10a), accounts for the linear increase of $\langle \Delta^2 r^{\mathrm{T}}(t) \rangle$; that is,

$$\langle \Delta^2 r^{\mathrm{T}}(t) \rangle = \frac{P_{\mathrm{seq}}(t)}{P(W(t) \geq 1)} \langle \Delta^2 r^{\mathrm{T}}_{\mathrm{P_{seq}}}(t) \rangle + \frac{P_{\mathrm{reg}}(t)}{P(W(t) \geq 1)} \langle \Delta^2 r^{\mathrm{T}}_{\mathrm{P_{reg}}}(t) \rangle. \tag{5.7}$$

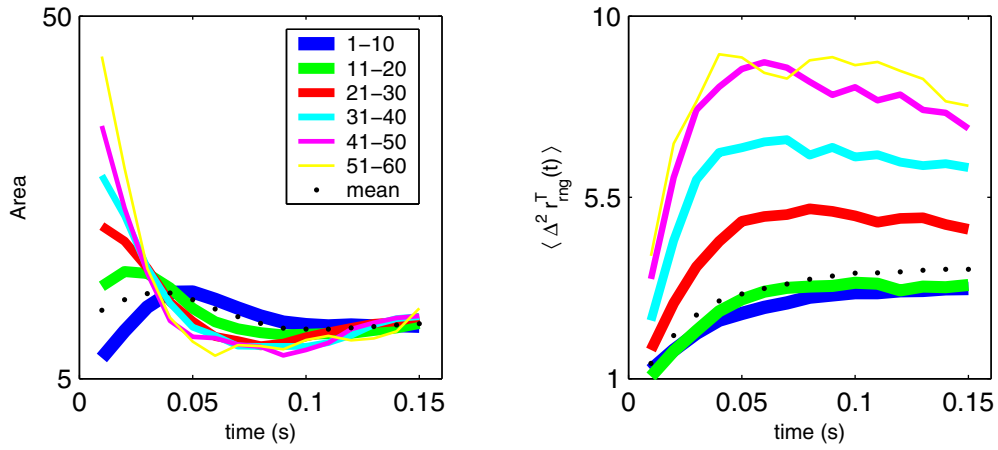
Another interesting feature uncovered by this analysis is that the displacements of regenerated bumps are larger than that of sequential bumps (saturation of 4 pix compared to 2.9 pix). This is characteristic of attractive–repulsive dynamics: bumps, initially attracted to a fixed site, feed this attraction throughout their existence — but once extinguished, newly formed bumps experience a repulsion from this previously occupied site.

5.3.2.2 Bump area

Here, we characterize how bumps change in area over time. To explore these area changes, we first construct the space $U(t, \mathrm{j}) = \Lambda(r_{\mathrm{j}}^{\mathrm{T}}(t))$, where Λ returns the area of the bump that falls within an original bump's circle (contingent on its existence); essentially, the operation that Λ performs is to find the subset of bumps that meet certain criteria (e.g., bumps that exist in $r_{\mathrm{j}}^{\mathrm{T}}(t)$) and lookup their areas in $S_{\mathrm{j}}(t)$. Next, we compute $\langle U(t) \rangle$ (black dots in Figure 5.11a). The shape of $\langle U(t) \rangle$ reveals that bump area does in fact exhibit temporal structure; however, the observation that $S_{\mathrm{j}}(t)$ is log-normally distributed suggests that $\langle U(t) \rangle$ may not accurately reflect the underlying distribution of bump sizes (i.e., bumps of different sizes may not have the same characteristic temporal structure shape). Accordingly, we have also calculated $\langle U(t, \mathrm{rng}) \rangle$, where rng indexes original bumps that fall within a range of areas; the range considered here is from 1 to 60 pixels^2 in increments of 10 (solid lines in Figure 5.11a). The widths of the lines in this figure are determined by the $\log$ of the number of bumps for each range; note that we have excluded bumps that have an area greater than 60 pixels^2, since their sample space is too small to estimate their dynamics.

These results paint an interesting picture of how the network reacts to bumps of different sizes: Large bumps (greater than 14 pix^2) initially decrease in area, whereas small bumps (less than 10 pix^2) initially increase in area, both tending towards a size of 12 pix^2. Furthermore, during the first 40ms, the rate at which these bumps change in area is roughly proportional to how far away they are from their final size. The fact that there is ringing (i.e., bump sizes overshoot their final value) suggests that an active feedback mechanism influences bump size.

We also explore how the initial size of a bump relates to bump diffusion (Figure 5.11b). This classification reveals that larger bumps move faster and further than smaller bumps,



a b

Figure 5.11: Temporal structure of bump areas

(a) The expected area of a bump at a certain time lag, where each trace represents a different bump size. The width of each line represents the log of the number of bumps in each sample size (149255 and 1463 samples for bump sizes ranging from 1-10 and 51-60, respectively. (b) Mean-squared displacement segregated by the corresponding area divisions.

uncovering another difference between physical particles and bumps: if bump size was equivalent to mass, then we would expect that larger bumps would diffuse at a slower pace. We have the opposite case, which suggests that bump size may reflect energy instead. This non-physical behavior may also reflect that the centers of bumps are more difficult to measure at larger sizes. However, since the number of large bumps is minimal and their influence on the mean diffusion is small (black points), we do not explore their properties any further. One interesting outcome of this analysis is that the characteristic shape of $\langle \Delta^2 r_{\mathrm{j}}^{\mathrm{T}}(t) \rangle$ (linear region followed by saturation) seems to be independent of bump size, which suggests that bump sizes are not related to the mechanism underlying confined motion.

5.3.3 Staccato motion

To complete our analysis of bump motion, we must also consider the possibility that our current technique for estimating diffusion ignores an entire class of bump movements. For example, we currently classify two bumps in different frames as being potentially the same object when they are less than T pixels from each other (i.e., the $r^{\mathrm{T}}(t)$ distribution). By default, all other bump pairings, denoted by $r^{\overline{\mathrm{T}}}(t)$, are classified as distinct objects (i.e., all other permutations of neighboring bumps). In this section, we show that the previously ignored $r^{\overline{\mathrm{T}}}(t)$ distribution can itself be segregated into two classes: one class that unquestionably corresponds to separate objects, and a second class where this distinction between the same and separate objects is ambiguous. We refer to this second class as staccato-like motion, since it can be characterized by sudden and disjointed bump movements. Whether the bumps involved in staccato motion should be considered as the same (e.g., bumps that teleport) or separate (e.g., neighboring bumps that are temporally correlated) object(s) is open for debate. Regardless of the interpretation, the extent at which staccato motion is present in our network must be quantified, since it may reflect important dynamics.

We define staccato motion as two bumps, separated in time, that do not have a counterpart (i.e., a nearby bump less than T pixels away) in each other's frame. Therefore, our strategy to uncover staccato motion is to search for bump pairs from the $r^{\overline{\mathrm{T}}}(t)$ distribution that could in fact be considered as the same objects. We identify these bumps in two steps:

The first step is to eliminate all of the bump pairs from $r^{\overline{\mathrm{T}}}(t)$ that clearly correspond to separate objects — a potentially difficult task since we do not know which bump corresponds to which bump in different frames. Our solution to this problem is to simply find the closest bump near each original bump in an earlier frame; we denote the distribution that contains all of these closest bump pairings in frames separated by to as $r^{\mathrm{C}}(\mathrm{to})$.

By considering only the closest bumps, we are ensuring that each bump has exactly one counterpart (in contrast, $r^{\mathrm{P}}(\mathrm{to})$ considers all possible bump pairs). For an example, we show $|\Delta| r^{\mathrm{C}}(40\mathrm{ms})$ (Figure 5.12a) alongside $|\Delta| r^{\mathrm{P}}(40\mathrm{ms})$ (Figure 5.12b) for comparison. We can think of each value in $|\Delta| r^{\mathrm{C}}(\mathrm{to})$ as representing the smallest sized circle, centered around an original bump, required before encountering a bump in a later frame. Circle radii larger than T (darker region of Figure 5.12a) represent the subset of bump pairs that were previously considered separate objects, but may in fact correspond to staccato motion.

The second step is to determine which of these remaining bump pairs (i.e., the instances of $r^{\mathrm{C}}(\mathrm{to})$ that are $> \mathrm{T}$ pixels apart) that may have teleported. Our strategy is to examine each closest bump (from the later frame) and decide whether it has a counterpart less than T pixels away (in the original bump frame). If the closest bump does have an counterpart in an earlier frame, then this closest bump clearly corresponds to a neighboring bump that has diffused; in other words, this bump has not teleported. However, if the closest bump does *not* have an counterpart in an earlier frame, the bump has essentially teleported. For an example, we show the bump pairs in $|\Delta| r^{\mathrm{C}}(40\mathrm{ms})$ that reflect staccato motion (red bars in Figure 5.12c), and those that do not (dark blue bars in Figure 5.12c).

To quantify staccato motion, we calculate the proportion of the bump pairs that qualify as staccato out of the potential staccato bump pairs (area of the red region divided by area of the red and dark blue regions in Figure 5.12c). In the example where to is 40ms, this staccato index is 0.4. We also compute the staccato index for different time lags (Figure 5.12d), which has a shape that is similar to the mean-squared displacement of bumps.

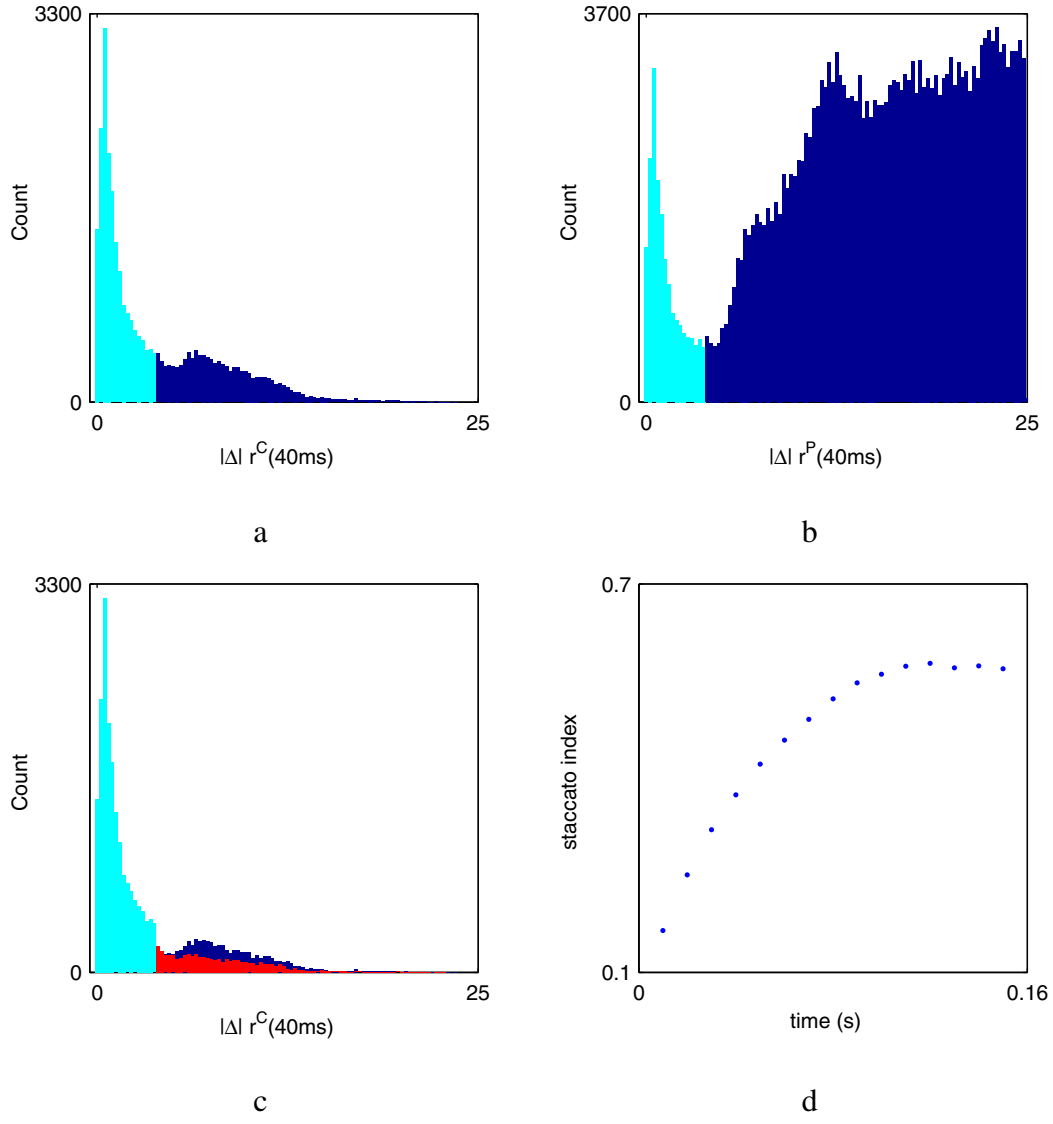


Figure 5.12: Uncovering staccato motion
Distribution of absolute displacement between closest bumps (a), and all pairwise bumps (b) for frames separated by 40ms. (c) The subset of bumps from (a) that are classified as staccato motion (red bars), and neighbors (dark blue bars). (d) The staccato index, which is computed as the ratio of the area of red bars in (c) over the dark blue bars in (a), for different time lags.

Chapter 6

Maps from motion

In this chapter, we explore how selectivity maps emerge from dynamic bumps.

We begin by characterizing the different types of dynamics that are possible in our network (Section 6.1); this analysis applies the techniques that we developed in Chapter 5. Next, we explore how these different dynamics relate to network computations (Section 6.2); in particular, we look at the speed at which orientation maps converge, and the selectivity of the maps (using measures of information and NVM). Finally, we uncover the neural mechanisms that allow bumps to move in the presence of heterogeneity, and examine how these mechanisms relate to computation (Section 6.3).

6.1 Motion regimes

Previously, we introduced a novel strategy for characterizing the dynamics in our recurrent network. Our approach was to first classify bumps (or contiguous regions of activity) within

frames, and then estimate their movements between frames — capitalizing on the idea that the bumps themselves are the underlying objects that drive network activity. Using this approach, we developed a number of techniques in Chapter 5 that quantified how bump motion was similar to, and different from particle motion.

In this chapter, we use these techniques to determine how bump motion relates to orientation selectivity. Intuitively, we know that a relation must exist; for example, if oriented gratings are unable to dislodge bumps from their preferred positions (i.e., the bumps are immobile), neurons can not be selective. We hypothesize that to achieve selectivity, bumps must be mobile enough so that different oriented stimuli can shift them, but also stable enough so that their interactions with the stimuli are consistent.

Before we can tease apart the relationship between bump motion and selectivity, we must first characterize the types of dynamics that are possible in our network. We tackle this problem in two steps: First, we provide an overview of the specific quantities that we employ to track bump motion (Section 6.1.1). Then, we systematically explore how these quantities change when the network parameters are varied (Section 6.1.2).

6.1.1 Overview of dynamic quantities

Here, we explicitly define the quantities that we will use to characterize dynamics — all of which are derived from Chapter 5. Because there are quite a few quantities, our analysis can quickly become overwhelming if not properly organized. To avoid confusion, we arrange them in a table, and adopt a simplified naming convention; in addition, we also provide a description of each quantity (including a brief derivation) in the following text.

The quantities that we track are divided into three classes, as indicated by the different

colored bullets in Table 6.1:

1. *Diffusive motion* (●)

We characterize diffusive motion by extracting the diffusion rate (D_{seq}) and the value where diffusion saturates (S_{seq}) of the bumps that exist sequentially (see Section 5.3.2.1 for a review). By only considering sequential bumps, we ensure that non-diffusive behaviors (e.g., bumps vanishing and reappearing) do not influence our estimates; for example, Figure 5.10b clearly demonstrates that regenerated bumps inflate $\langle \Delta^2 r^{\text{T}}_{\text{P(W(t)}\geq 1)}(t) \rangle$, especially when t is greater than 50ms. Formally, we compute D_{seq} as the slope of $\langle \Delta^2 r^{\text{T}}_{\text{P}_{\text{seq}}}(t) \rangle$ for $t = 10$ to 50ms, and S_{seq} as the average value of $\langle \Delta^2 r^{\text{T}}_{\text{P}_{\text{seq}}}(t) \rangle$ for $t = 90$ to 150ms.[1]

2. *Non-diffusive motion* (●)

We characterize non-diffusive motion by tracking the mean displacement of regenerated bumps (S_{reg}) and the amount that this displacement is beyond the saturation (S_{jmp}). Formally, we compute S_{reg} as the mean of $\langle \Delta^2 r^{\text{T}}_{\text{P}_{\text{reg}}}(t) \rangle$ for $t = 20$ to 150ms, and S_{jmp} as $S_{\text{reg}} - S_{\text{seq}}$.

3. *General network characteristics* (●)

In addition to measuring bump displacements and rates, we also track some general characteristics of our network. In particular, we consider: the probability that a bump exists after 10ms (P_{seq}); the probability a bump has been regenerated after 150ms (P_{seq}); and the staccato index (P_{stc}), which quantifies the proportion of bumps that have teleported (i.e., bumps that have vanished and reappeared more than T pixels away). We also measure the average area of bumps (A_{avg}) and the average number of bumps (N_{avg}) in each frame; while A_{avg} and N_{avg} are not technically dy-

[1] Our derivations in this section use the notation from Chapter 5; their specific definitions can be located via the section references in Table 6.1.

Table 6.1: Dynamic quantities

Parameter Name	Quant	Section Ref	Description
• Diffusion rate	D_{seq}	Section 5.3.2.1	diffusion rate of sequential bumps (10-50ms)
• Saturation	S_{seq}	Section 5.3.2.1	displacement of saturation (90-150ms)
• Displacement regenerated	S_{reg}	Section 5.3.2.1	displacement of regenerated bumps
• Displacement jump	S_{jmp}	Section 5.3.2.1	displacement of regenerated bump beyond S_{seq}
• Probability sequential	P_{seq}	Section 5.3.2.1	prob. bump will exist after 10ms
• Probability regeneration	P_{seq}	Section 5.3.2.1	prob. bump has been regenerated in 150ms
• Proportion staccato	P_{stc}	Section 5.3.3	proport. of bumps that are staccato (90-150ms)
• Bump area	A_{avg}	Section 5.3	average bump area
• Bump count	N_{avg}	Section 5.3	average number of bumps

namic quantities, they provide insight about the state of the network. Formally, we compute P_{seq} as $P_{\text{seq}}(10\text{ms})$, P_{reg} as $P_{\text{reg}}(150\text{ms})$, and P_{stc} as the average value of the staccato index for $t = 90$ to 150ms. Note that both P_{seq} and P_{reg} sample just a single value of the measured relationships, under the assumption is that these samples reflect the entire relationship. A_{avg} and N_{avg} are simply the means of $S_j(t)$ and $N(t)$, respectively.

6.1.2 Dependence on network parameters

As we noted earlier (Section 5.1), the bump dynamics in our network change qualitatively when we vary the parameters. Our goal in this section is to systematically map out these dynamics using the quantities from Table 6.1. Because the parameter space is astronomic, we must constrain our search. Our strategy is to first find a suitable network operating point that exhibits interesting bump dynamics, and then determine a subset of parameters that samples a wide-range of dynamics.

6.1.2.1 Reducing the parameter space

To simplify our search in the high-dimensional parameter space, we first find a suitable network operating point: Individual parameters are initially set to reasonable values based on an analysis of the underlying circuits (and previous circuit simulations). Then, we adjust each parameter until we observe bumps with interesting dynamics (defined subjectively at this point); this process occurs interactively with our real-time setup. For example, if INH cells are not participating in the network, we can increase the parameter E2I until they become active. This procedure is more of an art than a science, due to the fact that the physical circuits can deviate significantly from their simulations; moreover, the relation between the parameters and network behavior is often counterintuitive, as we will shortly see (Section 6.1.2.2). The final values that define our network operating point are indicated in Figure 3.5 (from Chapter 3).

After settling on an operating point, we now constrain our analysis to the post-synaptic current amplitudes ($\mathrm{E_A}$ and $\mathrm{I_A}$). These two parameters were selected because they sample a wide-range of different dynamics from visual inspection; moreover, postsynaptic current amplitudes are likely candidates for adjustment in a cortical network through synaptic plasticity.[2] Because, two parameters still constitute a large space, we further reduce their range by cutting out regions where: 1) bumps do not self-sustain for more than 10s in response to an impulse (low hysteresis); 2) bumps do not pack the entire network (few bump interactions); and 3) more than 28 neurons have an average firing-rate above 1,000Hz (unstable). The borders of the three regions were uncovered using the real-time GUI; for instance, when the network becomes unstable as we are sweeping through the parameter space, a

[2]There is also evidence that intralayer connectivity patterns in V1 are modified during development, and in particular, long-range connections have been implicated in map consolidation [20]. It is unclear whether these connections could play a role in dynamics.

red box flashes in the GUI. We demarcate the borders of regions 1 (low hysteresis) and 3 (unstable) with x's (Figure 6.1). It is important to note that these regions are empirically defined and do not imply underlying bifurcations.

6.1.2.2 Observed diffusion rates

Having carved out a region in the two-dimensional parameter space $\{\mathrm{E_A}, \mathrm{I_A}\}$, we compute D_{seq} for points in that range (Figure 6.1); we initially focus on bump diffusion rates simply because this quantity has an intuitive physical correlate (i.e., D_{seq} measures the expected distance bumps move in a given time interval). We observe in this figure that D_{seq} is mapped smoothly across the region: the lowest rate (14.5 pix^2/s) occurs when both postsynaptic amplitudes are low (state 1), and the highest rate (41.5 pix^2/s) occurs when both amplitudes are high (state 29); recall that decreasing the voltages $\mathrm{E_A}$ and $\mathrm{I_A}$ result in an *increase* in postsynaptic amplitudes since they connect to pmos transistors. We also note that even though diffusion is correlated with both excitatory and inhibitory amplitudes, the relationship with excitation is noticeably greater.

Our result that increased postsynaptic amplitudes (feedback) corresponds to higher bump diffusion is somewhat surprising; intuition suggests that strong feedback would serve to pin the bumps to their preferred locations — resulting in decreased diffusion. This logic suffices for a one-layer recurrent network that collapses excitatory and inhibitory connections into a single analog interaction (such as the original bump model from [51]). However, because our network consists of two-layers, the resulting dynamics can become significantly more complex, and often leads to counterintuitive behaviors. In our network, for example, increasing feedback will cause activity to ping-pong between the excitatory and inhibitory layers, effectively dislodging bumps from their preferred locations. We present

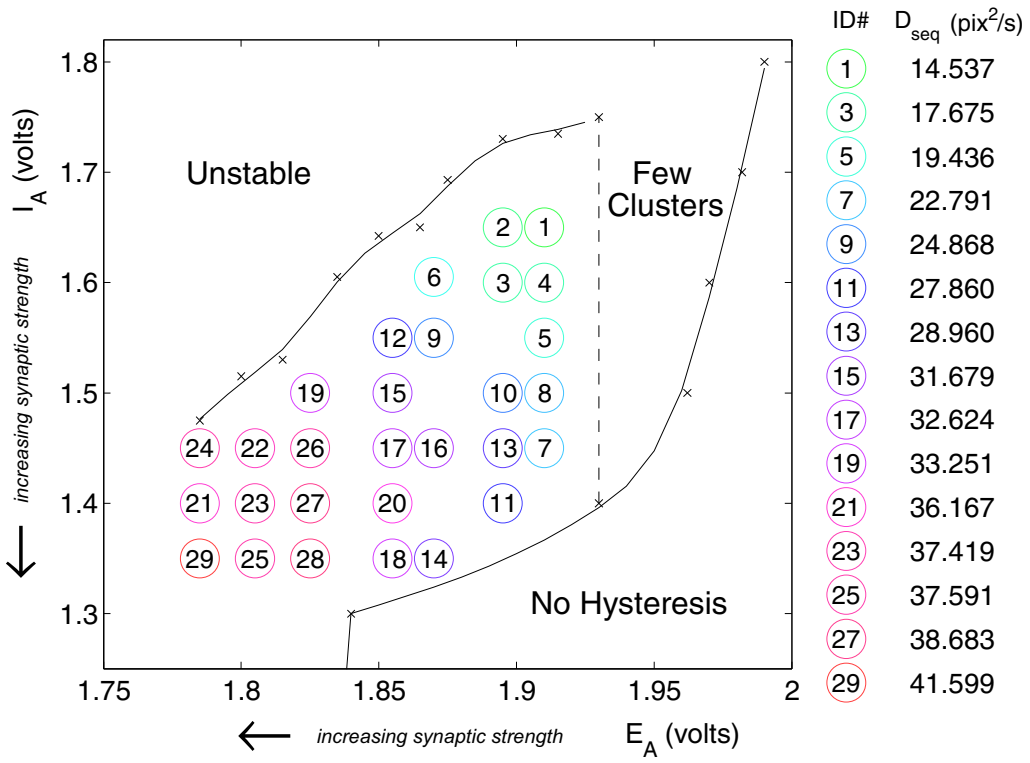


Figure 6.1: Map of diffusion rates
Bump diffusion rates (numbered) are shown for different voltages $\mathrm{I_A}$ and $\mathrm{E_A}$.

a characterization of these dynamics later in this chapter (Section 6.3.1).

6.1.2.3 Observed network dynamics

Because D_{seq} does not capture the full spectrum of dynamics in our network, we also compute the remaining quantities from Table 6.1 for each network state (Figure 6.2). In these figures, we plot a subset of the quantities (y-axes) against the diffusion rate (x-axes), where each marker corresponds to a unique state from Figure 6.1. We have chosen to compare each quantity directly with diffusion (D_{seq}) for simplicity; that is, D_{seq} is used as a reference for the parameters since it maps smoothly across the parameter space. As we will see shortly, D_{seq} turns out to be useful in its own right for deciphering these other

dynamic quantities.

In Figure 6.2a, we observe that the saturation displacement (S_{seq}) and regeneration displacement (S_{reg}) both have near linear relationships with D_{seq}. This relationship between S_{seq} and D_{seq} is not too surprising: faster bump diffusion (greater slope between 10 and 50ms) will naturally lead to a greater saturation displacement (average value between 90 and 150ms). It follows that if bumps move a greater distance in a given interval, then the distance at which bumps regenerate (S_{reg}) will also increase because of attractive-repulsive dynamics (i.e., when a bump regenerates, it initially experiences a repulsive force from the location of the previous bump). It is interesting to note that S_{jmp} is nearly constant across network states, which indicates that the repulsion displacement is independent of diffusion rate.

Next, in Figure 6.2b, we track the probability that a bump exists after 10ms (P_{seq}), and that a bump has been regenerated after 150ms (P_{reg}). We observe an inverse relationship between P_{seq} and D_{seq}, indicating that bumps vanish less when bump diffusion is low. Surprisingly, the relationship between P_{reg} and D_{seq} is concave down and peaks near medium diffusion rates (≈ 30 pix^2/s). We suggest that the reason P_{reg} decreases at high diffusion rates is a consequence of network oscillations; these oscillations can suppress bumps from being regenerated on the timescale that we are looking at (150ms). If we looked farther out in time, however, we would expect to see more regenerated bumps. We present evidence to support this idea in Section 6.3.1, when we explore the mechanisms underlying bump diffusion.

Next, we track the relationship between the staccato index (P_{stc}) with diffusion (Figure 6.2c). This figure shows that the percentage of bumps that can be considered staccato increases linearly with diffusion. This relationship is consistent with our previous result that

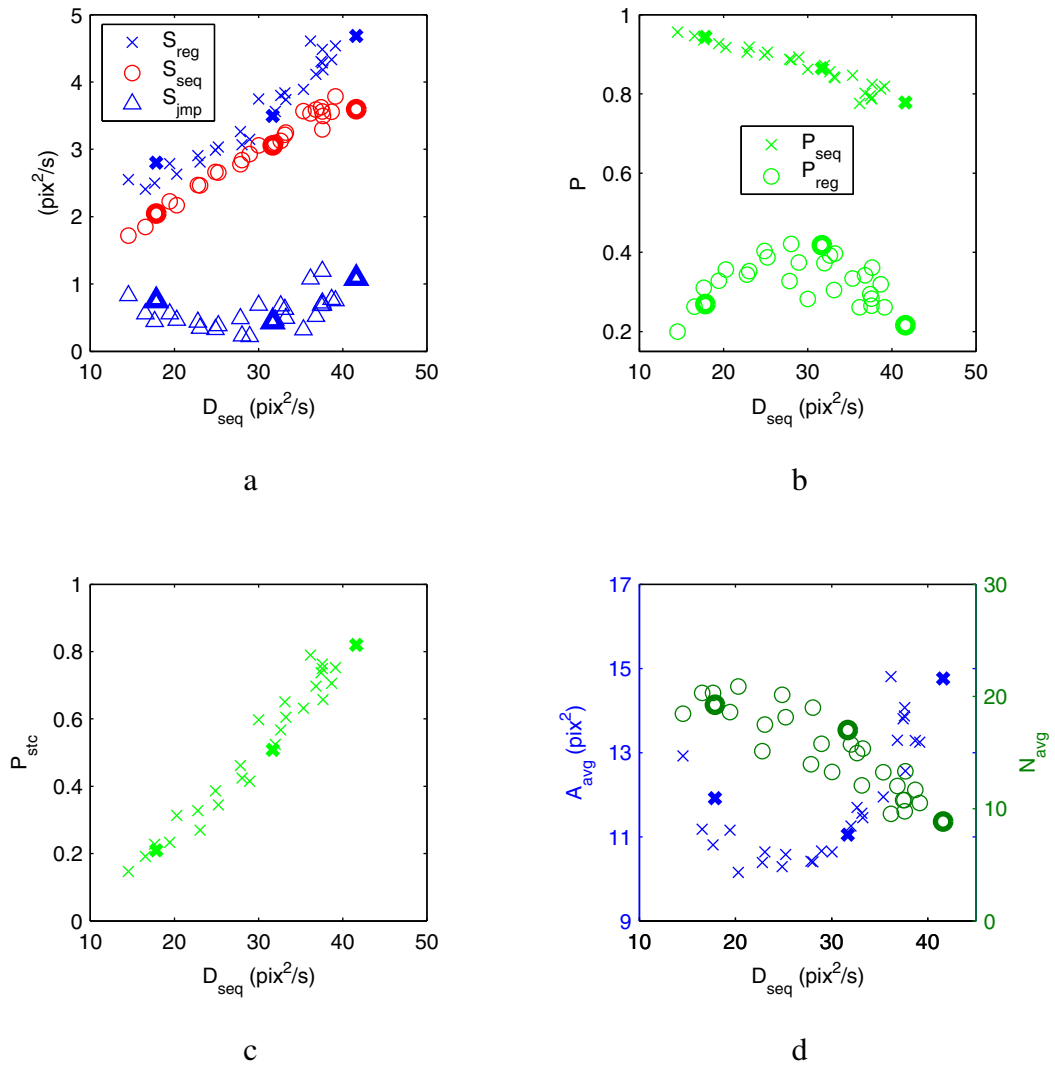


Figure 6.2: Network traits

Relationships between quantities from Table 6.1 and diffusion rates for each state. Bold markers represent states whose computational properties are explored in detail.

high diffusion suppresses bump regeneration for sometime in a local region; this suppression would putatively promote bump regeneration at adjacent locations, thereby setting up the right conditions for bumps to 'teleport'.

Finally, we track the average area (A_{avg}) and number of bumps (N_{avg}) for each state (Figure 6.2d). Unlike the other quantities that we have measured, A_{avg} and N_{avg} do not measure bump dynamics directly, however, they still provide useful information about the network's state. We observe that A_{avg} is concave up and has a minimum when diffusion rates are medium. On the other hand, N_{avg} decreases approximately linearly with increasing diffusion. From these parameters, we can estimate that the packing density is largest when diffusion is low (i.e., $N_{\text{avg}} \times A_{\text{avg}}$ is large).

6.2 Dynamic computation

The preceding analysis demonstrates that our network can exhibit a wide-range of bump dynamics. Our goal in this section is to characterize how these dynamics affect computation — ultimately uncovering which types of bump motion lead to the best maps. To distinguish the quality of the orientation maps, we measure the speed at which they converge (Section 6.2.1), and their selectivity (Section 6.2.2).

6.2.1 PO convergence

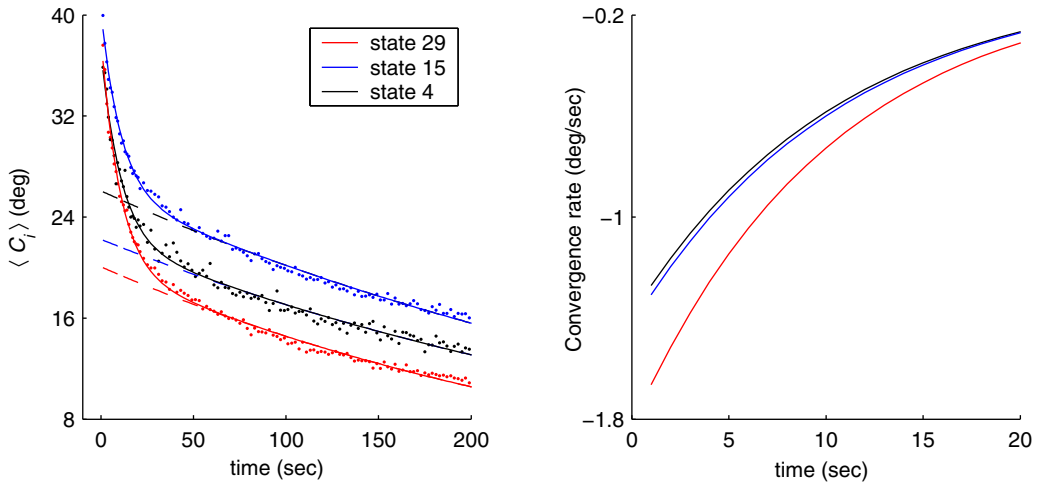
Here, we explore the relationship between bump motion, and the speed at which orientation maps form. Our expectation is that fast moving bumps will give cells more opportunities to respond (or not respond) to oriented stimuli, whereas slow moving bumps will require more

stimuli presentations for the same opportunities. To test this expectation, we introduce a simple analysis that estimates how quickly an orientation map approaches its converged state. Then, we use this analysis to measure the convergence rates for three network states with slow, medium, and fast dynamics.

Consider that we have access to the exact response distributions of a cell to single cycle oriented gratings (i.e., one distribution for each orientation). In this case, we can compute the *true* PO of the cell by taking the vector sum of the mean response to each orientation (see Equation 4.9). The question that we are interested in here is: how many independent samples from these distributions are required before the sampled PO approaches the true PO of the cell? To answer this question (through measurement), we define the convergence of cell i, denoted by $C_i(s)$, as the distance between the sampled PO (computed from s samples) and its true PO.[3] Furthermore, the *convergence rate* is simply the derivative of $C_i(s)$; because each sample corresponds to a presentation of a single grating cycle (1 sec), the convergence rate is measured in degrees/sec.

To measure $C_i(s)$ for the cells in our network, we make the following choices: First, we approximate the exact probability distributions of cell responses with empirical ones, using data from a long experiment; in particular, we present 4 orientations, each for 1,000 cycles. Second, to compute the distance between the sampled PO (considering parts of the trial) and the true PO (considering the entire trial), we employ the similarity index (SI), as defined in [24]; recall that SI measures the difference between two angles on a circular axis. In our calculation, we do not normalize SI but instead keep the measure in units of degrees, where 90 degrees represents opposite, and 0 degrees represents identical POs. Finally, to estimate $C_i(s)$ for each cell, we take 20 independent samples of size s for each cell, and

[3] We note that to determine if the sampled PO falls within a particular confidence interval, we can not assume that the sampling errors will be distributed normally (since orientation is a circular variable). Refer to [76] for a review on estimating sampling errors in this situation.



a b

Figure 6.3: PO map convergence

(a) Comparison of PO map convergence ($\langle C_i(s) \rangle$) for three network states with shown fits (solid and dashed lines). (b) Comparison of convergence rates computed from fit parameters.

compute the mean SI for each cell taken over the samples; the average SI across all cells ($\langle C_i(s) \rangle$) is then reported for each map.

We compute $\langle C_i(s) \rangle$ for three different network states where s ranges from 2 to 200s (Figure 6.3). These three states, which are shown by bold markers in Figure 6.2, were chosen because they sample unique combinations of dynamics. In particular, states 4, 15, and 29 reflect low, medium, and high diffusion rates, respectively. Figure 6.3a shows that the cells in state 29 approach their long-term POs at a faster pace than the other states, thereby confirming our original expectation that fast diffusion leads to a fast converging map (a 12 degree difference after 200s). Surprisingly, the map obtained from the medium diffusion rate does not follow expected trend: the average cell in state 15 is nearly 20 degrees away from its final PO after 200s, whereas state 4 is 16 degrees away. This result indicates that relationship between diffusion and map convergence is not as simple as we originally anticipated.

Table 6.2: Fits for $\langle C_i(s)\rangle$

State	a, [CI]	b, [CI]	c, [CI]	d, [CI]
29	18.1,$[17.5, 18.6]$	-0.098,$[-0.104, -0.092]$	20.1,$[19.8, 20.4]$	-0.0032,$[-0.0034, -0.0038]$
15	14.1,$[13.6, 14.7]$	-0.094,$[-0.101, -0.087]$	26,$[25.8, 26.4]$	-0.0026,$[-0.0027, -0.0025]$
4	14.8,$[14.0, 15.6]$	-0.093,$[-0.103, -0.083]$	22.2,$[21.8, 22.7]$	-0.0026,$[-0.0028, -0.0025]$

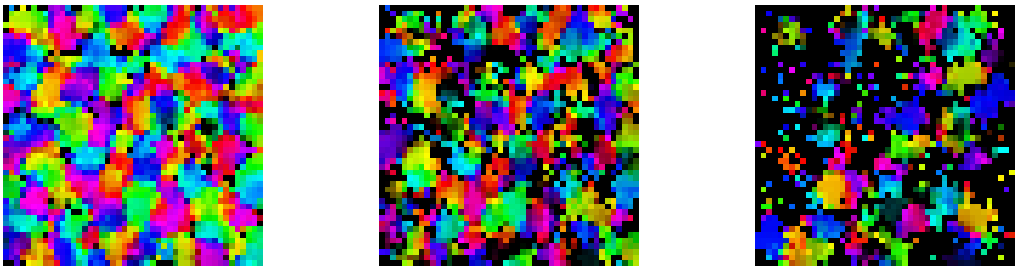
To quantify the convergence of each state, we parameterize $\langle C_i(s)\rangle$ using a sum of two exponentials:

$$a\mathrm{e}^{b\cdot s} + c\mathrm{e}^{d\cdot s}, \tag{6.1}$$

where $\{a, b, c, d\}$ are the parameters. The fits, which are shown in Figure 6.3a, use solid lines to represent the sum of both exponentials, and dashed lines to represent the slower of the two exponentials. The particular values of the fit are indicated in Table 6.2, along with confidence intervals.

Using the parameters from Table 6.2, we estimate the convergence rates of the maps in the three different states analytically (Figure 6.3b). These curves reveals that state 29 does in fact converge at a faster rate than the other two states. In addition, we can also observe that state 4 and 15 have nearly identical rates even though PO map from state 4 comes closer to its final state compared to state 15.

It is also worthwhile to note that the percentage of responsive cells (map coverage) in the three network states change in a consistent manner with bump diffusion rates. In particular, states 4, 15, and 29 have 42%, 71%, and 88% of cells responding, respectively. This relationship is not too surprising: we expect that diffusive bumps would cover a larger distance, subsequently allowing more cells in the network to participate. To illustrate this point, we plot the three PO maps for these states and set non-responsive cells to black



a b c

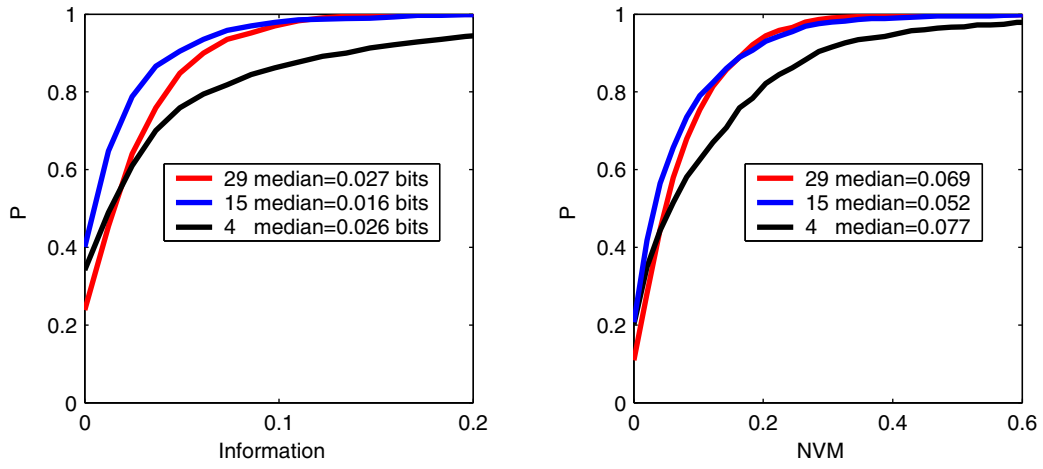
Figure 6.4: PO map comparisons
Orientation maps for network states with high (a), medium (b), and low (c) bump diffusion rates.

(Figure 6.4).

6.2.2 Selectivity

We have demonstrated that higher bump mobility leads to superior maps. For instance, cells in state 29 'know' their PO in fewer trials on average (Figure 6.3). In addition, the percentage of cells that participate in the maps is also greater at this high diffusion rate (Figure 6.4). Our aim in this section is to determine whether these superior map qualities come at the expense of selectivity.

Following the techniques described in Chapter 4 for quantifying selectivity, we compute the information (Figure 6.5a) and NVM (Figure 6.5b) for each cell when the network is tuned in the three different states, and plot the results in cumulative distributions; note that each distribution only considers responsive cells (i.e., the non-black cells in Figure 6.4). Two trends are evident: First, we observe that there is a crossover between state 4 and state 29 near the medians (more noticeable in Figure 6.5b). This suggests that low



a b

Figure 6.5: Comparison of selectivity
Cumulative distribution of information (a), and NVM (b) for responsive cells in the network for three states. Note that we only plot the distributions up to 0.2 bits for information, and 0.6 for NVM, to increase the visibility in the lower regions.

diffusion (state 4) favors extremes (either low or high selectivities), whereas high diffusion (state 29) is more evenly distributed. The second observation is that the medium diffusive state (15) consistently trails behind both information and NVM measures relative to the other two states. This is consistent with our result from Section 6.2.1 that shows state 15 also lags behind in convergence; that is, we expect that more trials would be required to determine a cell's PO when selectivity is low. So unfortunately, the relationship between bump diffusion and network computation (information and NVM) is not straightforward (i.e., it is non-monotonic).

However, the differences in selectivity between the states are not overly dramatic (with these current results, not much can really be determined beside the aforementioned trends). The result that we wish to emphasize is that the high diffusive state does not sacrifice selectivity for its superior convergence rate and coverage factor.

6.3 Mechanistic origins of motion and selectivity

Although we have made significant progress in characterizing the behavior of the bump chip, our current analysis has *not* been able to uncover the fundamental mechanisms that are responsible for map formation. This is problematic: we need to have a firm understanding of the these mechanisms if we are to judge the viability of our model (e.g., to make testable predictions that can be validated in biology). In a sense, our project up to this point has done little more than transformed an unexplained phenomena in the cortex (i.e, the initial appearance of PO maps) into an unexplained phenomena in a silicon chip (albeit in a more controlled setting).

Our findings, however, have not left us completely in the dark. For example, we discovered in this chapter that innate bump motion (measured when the network is driven with uncorrelated inputs) is linked to the formation of PO maps: this result suggests that unearthing the mechanisms responsible for innate bump movements may provide clues about selectivity. Following this lead, we explore how bumps are able to move in the presence of heterogeneity by directly observing the patterns of activity in EXC and INH cells for different levels of feedback (Section 6.3.1).

Another promising possibility is to use our bump-based analysis to explore how moving bumps interact with gratings. For example, this analysis will allow us to compute the likelihood that a bump will form in its 'correct' location (according to the converged PO map), based on the bump's features (e.g., its area or shape). Furthermore, we can also track how these likelihoods evolve over time; that is, do bumps with particular features diffuse to their correct locations, and if so, on what time scale? As one can quickly see, the possibilities are extensive. In our case, we have chosen to explore how bumps orient themselves (i.e., their major-axis of orientation when fit by an ellipse) when driven by

oriented stimuli (Section 6.3.2).

The insights that we obtain in this section will be used to piece together a coherent story for how the bumps in our network obtain their orientation selectivity; this story will serve as the basis for the testable predictions that we will make in the final chapter (Chapter 7).

6.3.1 Dislodging confined bumps

Here, we explore the surprising result that the bumps in our network become more mobile as the postsynaptic amplitudes are increased (as shown in Figure 6.1). Our hope is that by understanding the spike patterns that promote innate bump motion, we will gain insight into the relationship between bump dynamics and map formation.

Our first approach is to simply view the spikes of coupled EXC–INH cells over a small snippet of time; in this example, we consider ten cells of each type that are in the same column. We observe that in state 29, which corresponds to a high diffusion rate (Figure 6.6a), activity alternates between the two layers — first EXC responses build for about 80ms (green rasters), which then recruits massive activity in INH cells (black rasters). State 15 exhibits a similar spike pattern (Figure 6.6b), except here, the interlayer interactions occur on a slower time scale (EXC responses build for nearly 200ms). In contrast, state 4 has no discernable global oscillations over the shown timescale (Figure 6.6c).

To check if these observed spike patterns are in fact characteristic for our network, we have also computed the autocorrelation (AC) for a 10×10 grid of EXC cells, considering 50s of data. We summarize our findings by first normalizing the AC for each cell so that its amplitude at lag 0 is unity, and then computing the average AC across all cells (Figure 6.6d). The average ACs for states 29 and 15 indicate that the oscillations observed in the

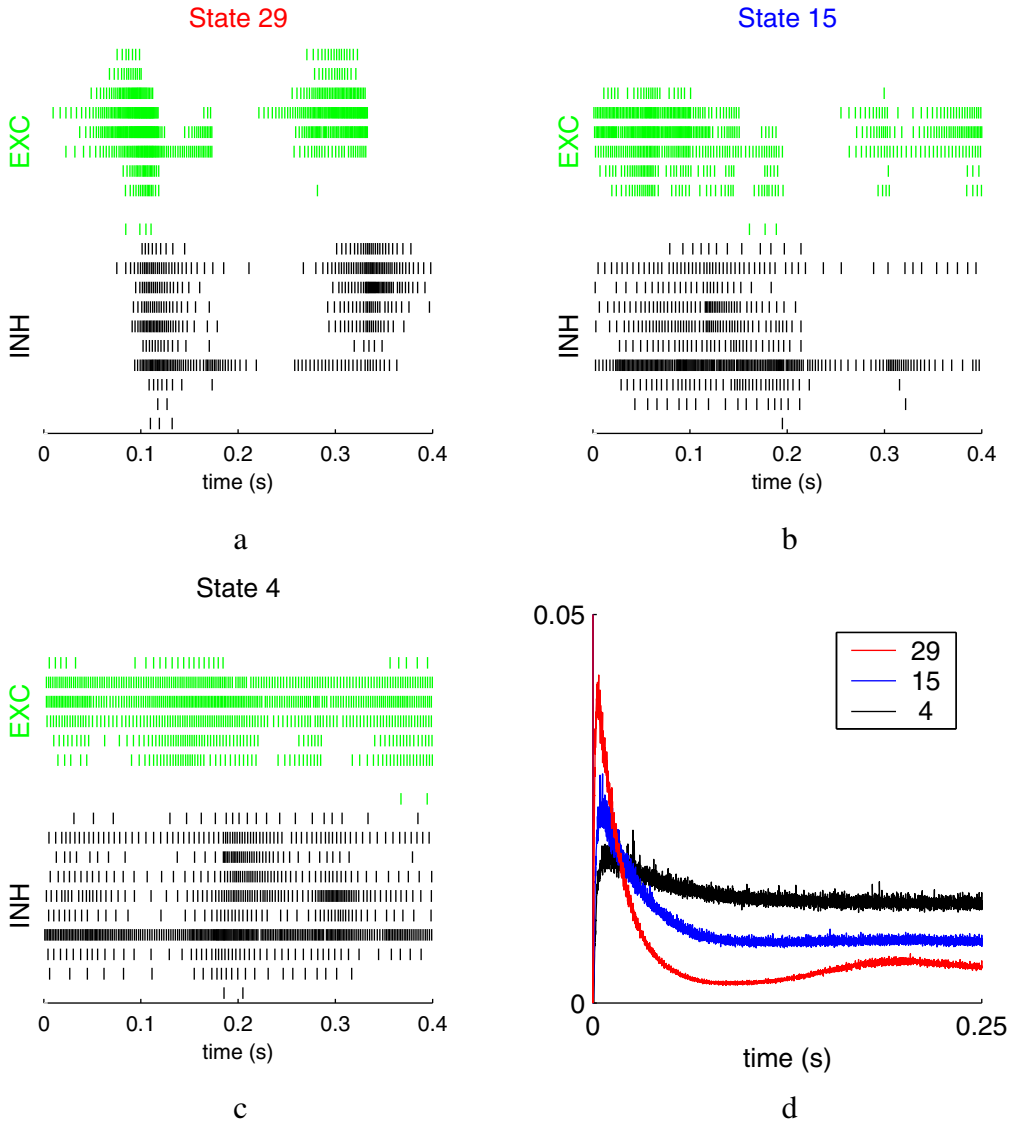


Figure 6.6: Spike dynamics
(a)–(c) Rasters for ten EXC (green, top) and INH (black, bottom) neurons in different states. (d) The average autocorrelation, which is calculated by considering EXC spike over a 50s interval at 0.1ms bin resolution. The autocorrelation is normalized for each neuron individually and averaged over 100 neurons.

rasters are stereotypical responses; both states 29 and 15 decay rapidly over the first 100ms, and then tail off (or slightly rebound) some time later. The average AC of state 4, on the other hand, decays monotonically (resembling a power law).

These observed oscillations shed light on a number of puzzling results that we have encountered in this chapter. For one, it helps explain why increased recurrent feedback spurs bump mobility (recall that increased feedback serves to pin bumps to their preferred locations in a single-layer network). We can now clearly see increased feedback in a two-layer network causes a ping-pong of activity between the layers; furthermore, these oscillations become shorter and more focused as the coupling between EXC and INH cells is increased. We propose that these focused bursts of inhibitory and excitatory activity are able to dislodge bumps from their preferred locations.

Another puzzling result that we encountered was that bump regeneration (P_{reg}) decreased at high diffusion rates (see Figure 6.2b). Again, the oscillations explain why this is the case: When a bump vanishes (the end of an excitatory epoch in the Figure 6.6a), the inhibitory epoch is near its peak. This inhibition, which lasts for an additional 100-200ms, effectively suppresses bump regeneration on the time scale that we are looking over. At lower diffusion rates, however, the inhibitory oscillation is more diffuse, allowing bumps to regenerate at earlier times.

In conclusion, we have shown that our two-layer network exhibits diverse dynamic behaviors due to excitatory–inhibitory oscillations. In a sense, these oscillations seem to have evened the playing field: neurons that are inherently overactive (causing pinned bumps) can now be actively dislodged through coordinated inhibitory activity; this allows less fortunate neurons an opportunity to participate — an opportunity not offered in the states with no oscillations.

6.3.2 Bump motion with oriented inputs

Up to this point, our bump analysis has been used to quantify innate bump dynamics (when the network is driven with uncorrelated inputs). One exciting prospect is that this same analysis can be used to measure how bumps interact with oriented gratings, potentially elucidating the mechanisms that underly selectivity. Our excitement is justified: as we will show shortly, this analysis is capable of exposing non-obvious relationships between bumps and PO maps that would otherwise be difficult to pick out. Specifically, we report results that uncover a surprising relationship between the way bumps orient themselves relative to the stimulus, and the final selectivity map. Before we present these results, we first upgrade our modified experimental procedure and bump tracking techniques to handle gratings.

6.3.2.1 Modified experimental procedure and analysis

The goal of our current set of experiments is to characterize bump dynamics while the network is being stimulated with gratings. We perform these experiments by following the procedure described in Section 5.1.1, except here, we replace the Poisson inputs with square-wave gratings. Specifically, we present four square-wave gratings to our network with orientations that range between -180 and -45 degrees; each trial, which corresponds to a different stimulus, lasts for 75s. The resulting activity is then binned at a 10ms resolution, which corresponds to four movies with 7,500 frames; we denote these movies by $A_{i,k}(t)$ where i indexes a cell's location, k indexes the orientation of the stimulus ($k = 0 \ldots 3$), and t is the frame index. As before, the number of spikes in each frame is capped at five, limiting the instantaneous rate of overactive cells to 500Hz.

Next, we characterize the dynamics of $A_{i,k}(t)$ using the techniques developed in Chap-

ter 5; this analysis is performed for each trial independently. Specifically, we identify bumps in each frame, and extract their features (Sections 5.2.1 and 5.2.2). For our current analysis, we track each bump's centroid, area, and major axis of orientation when fit by an ellipse; we refer to the bump's ellipsoid orientation as BEO. In addition to these standard features, we also record where the center of each bump lands in relation to the converged PO map and NVM maps (which were computed in a separate experiment following the procedure in Section 4.1.2); we refer to the PO of the cell that the bump's center lands on as a bump's PO, or BPO. After reducing activity to a collection of bumps, we can then construct a difference distribution, $r^{\mathrm{T}}(t)$, for each trial. These distributions will allow us to estimate the temporal structure of bump features between frames following the examples in Section 5.2.4.

The final point to note is that in addition to measuring BEOs and BPOs, we also record their values *relative* to the orientation of the stimulus (as opposed to being measured relative to the cardinal axis of the array). That is, we compute the absolute angular difference on a circular axis (in orientation space) between the measured values and the stimulus orientation; we refer to these stimulus relative measures as RBEO and RBPO (where the R is an abbreviation for relative). For example, if a bump has a BEO of -10 degrees, we would record a RBEO of 55 degrees when the stimulus is a -135 degree grating, since 55 is the difference between -135 and -10 on a circular axis in orientation space. As we will see, measuring these orientations relative to the stimulus simplifies our analysis, since we are primarily interested in how these quantities change in relation to the stimulus; one notational benefit is that for these relative measures, we do not have to keep track of four separate $r^{\mathrm{T}}(t)$ distributions, but can instead simply concatenate them into a single distribution.

6.3.2.2 Tracking bump orientations

Here, we explore the possibility that the way in which bumps contort to oriented inputs is somehow related to the formation of PO maps. The inspiration behind this idea comes from the observation that bumps are not perfect circles, but are actually quite malleable. Therefore, it seems likely that driving the network with drifting oriented inputs could result in stimulus-induced biases in bump shapes.

To test this possibility, we have performed the experiment outlined in Section 6.3.2.1 for network states 29, 15, and 4. The resulting distributions of BEOs are shown for each state (Figure 6.7a–c). We note that 1) only bumps with areas larger than 10 pix^2 are considered, which effectively removes spurious contributions from smaller bumps,[4] and 2) the common modes of BEOs (Figure 6.7d) have been subtracted out to remove any systematic biases.[5]

For states 29 and 15, the majority of bumps orient themselves parallel to their stimulus orientation. This observation is perhaps expected — if we are driving the network along a particular axis, it seems logical that neural activity will take the shape of the inputs. But can these observed bump elongations account for OS? For example, we might hypothesize that OS is the combination of bumps being forced to match the shape of their inputs, while simultaneously being constrained by their preferred locations. Although this would be a welcomed conclusion, the evidence that we have amassed so far does not support

[4]This simplification was made because many of the smaller bumps report orientations that fall along the cardinal axes simply because they are symmetric. The likelihood observing large bumps that are symmetric, however, is greatly reduced.

[5]The bumps in state 4 have an overwhelming preference for -135 degrees, regardless of the stimulus. For example, nearly 7,000 bumps in this state were oriented along a -135 degree axis; however, once the common mode is removed, there are less than 200 bumps that prefer -135. This indicates that although a bias exists, it is not stimulus specific (otherwise, we would have a problem obtaining PO maps). We would like to emphasize that this bias is *not* introduced by our analysis, but is instead a chip-related effect. We have not explored the source of this bias in any extent; the potential culprit is likely a supply gradient, which is exacerbated in state 4 because the neurons are firing at higher rates (i.e., the circuits are drawing more current, which can cause an IR drop across length of the chip).

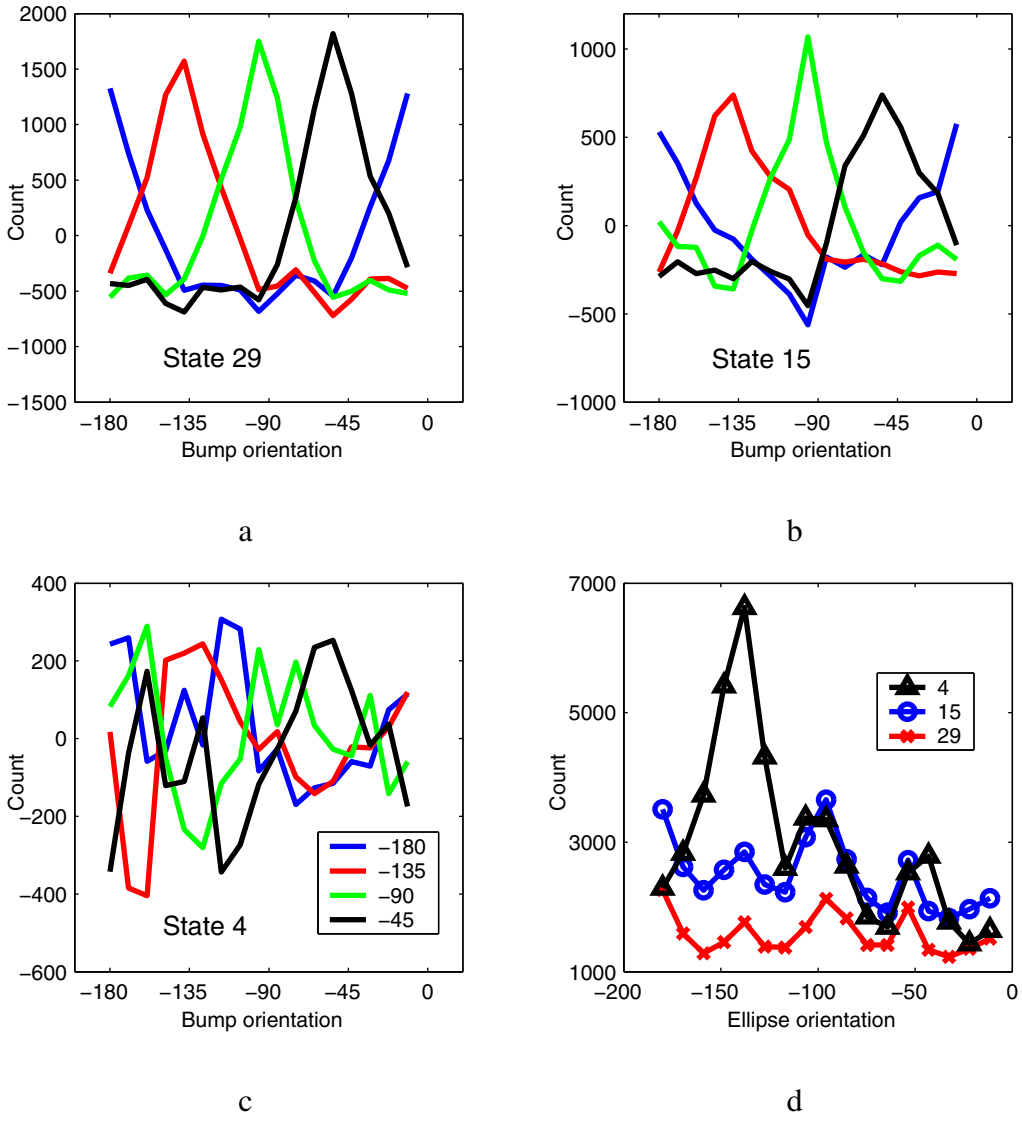


Figure 6.7: Distributions of BEOs

Ellipsoid orientations of bumps (greater than 10 pix^2) when the network is driven (a–c). The common orientations for each state (d) are subtracted out of the individual histograms. Note that state 4 has an overwhelming bias at -135 degrees; however, the bias is common to all stimuli.

it. For starters, we would predict that states 29 and 15 would be highly selective, in par with the dramatic bump elongations. As we have consistently found, however, selectivity across multiple measures is moderate to low (considering the selectivity index, NVM, and information). Other opposing evidence is that the distributions of BEOs for state 4 is quite noisy, yet this state outperforms state 15 in both terms of selectivity and convergence rates. Clearly, there must be other factors involved in map formation.

Because the relationship between BEOs and selectivity is still unclear, we employ a more direct analysis that tracks how close bumps land to their 'correct' locations based on their shapes (i.e., the BPOs). For example, consider the hypothetical scenario where bumps that are parallel to their inputs (i.e., their RBEOs are close to 0) *are* in fact responsible for OS, as suggested earlier, whereas RBEOs much larger than 0 do not contribute. One method of exposing this relationship is to first group bumps based on their RBEOs (e.g., bumps that are parallel, nearly parallel, etc. to the stimulus), and then track the RBPOs for each group (i.e., where the bumps in each group form according to the final PO map relative to the stimulus). Continuing with our hypothetical example, the group with low RBEOs (parallel to the stimulus) would have low RBPOs, since they would land in regions that are near their correct POs, whereas the other groups (not parallel to the stimulus) have random RBPOs, since they would land in arbitrary locations.

We performed an experiment based on the above example to see if the way in which bumps orient themselves to a grating (RBEO) predicts if they form in the correct locations (RBPO). Specifically, we first separated bumps into four groups based on how they orient themselves to the grating: parallel (RBEO $\in$ 0–22.5), nearly parallel (RBEO $\in$ 22.5–45), nearly perpendicular (RBEO $\in$ 45–67.5), and perpendicular (RBEO $\in$ 67.5–90). Then, based on these different groups, we tracked if bumps form in near correct locations (RBPO is relatively small), or random locations (RBPO is 45). The results are shown in Figure 6.8

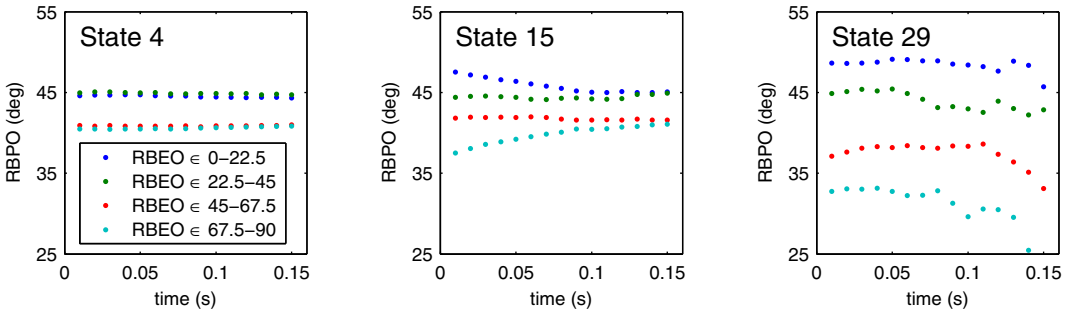


Figure 6.8: Relationship between RBPO and RBEO
Bumps that orient themselves perpendicular to the input grating (RBEO $>$ 45 degrees) tend to form in 'correct' locations (their RBPOs are lower); By contrast, bumps that orient themselves parallel to the input tend to form in incorrect locations. This trend is visible in all three dynamic states.

for all three states.

These plots show a surprising trend that is consistent for all the states: bumps that are *perpendicular* to the stimulus (RBEOs that are greater than 67.5 degrees) tend to form closer to their correct locations, whereas bumps that form parallel to the stimulus (RBEOs that are less than 22.5 degrees) land in either random — or even incorrect locations (RBPOs are greater than 45 degrees). But how could bumps that form perpendicular to the stimulus be better off? For one, it would explain why selectivity has consistently been low: the number of parallel bumps far outnumber the perpendicular bumps (see Figure 6.7).

After significant reflection, this idea may not be so radical. Perhaps the selectivity in our network occurs when bumps are 'dragged' along with the grating; that is, maybe bumps are selective to spatio-temporal energy rather than just elongated inputs. Another interesting possibility is that selectivity may be determined entirely by the recurrence, and *not* by direct stimulation; in fact, our results suggest that cells directly driven by the grating are actually less selective. Before we become preoccupied with speculations, we first confirm that this

exciting result it not simply an artefact of our analysis.

6.3.2.3 Verification

In this section we verify our result that large RBEOs tend to form in the correct locations. Our strategy is based on the following idea: If the subset of bumps that have large RBEOs are in fact responsible for OS, then a PO map created from this subset should closely resemble the final PO map (considering all activity); on the other hand, PO maps created from bumps with small RBEOs should bear little resemblance to the final map.

Based on this premise, we decompose $A_{i,k}(t)$ into four separate channels, where each channel only considers a range of RBEOs (e.g., parallel, nearly parallel, etc.). In practice, this decomposition is achieved by first identifying bumps within each frame, and then segregating the activity of each bump into its appropriate channel based on its RBEO. The result is four decomposed activity movies, each of which have four dimensions (one for each stimulus orientation).

Next, we compute the PO map for each of the four channels. These maps are shown for states 29, 15, and 4 in Figures 6.9, 6.10, and 6.11 (b, d, f, and h), respectively; we also show for comparison the original PO map (without separating the channels) in the central plots (e). To quantify the similarity between the decomposed PO maps and the final map, we compute their similarity indexes (a, c, g, and i). From these plots, we see that as proposed, bumps with large RBEOs result in PO maps that resemble the final map (i), whereas bumps with low RBEOs bear little resemblance (a); this trend persists in all states. One deviation from our predictions, however, is that moderate RBEOs (bumps with nearly perpendicular and parallel orientations) are *more* similar to the final map than the one with high RBEOs.

We hypothesize why this might be the case later in Chapter 7.

To gauge the relationship between the deconstructed PO maps, we also compute the mean SI taken across cells for each state, and plot them in a 4×4 matrix (Figure 6.12); in these plots, the main diagonal compares each deconstructed PO maps with the final map (instead of trivially reporting self-map comparisons). These intermap comparisons reveal that PO maps made from perpendicular and nearly perpendicular bumps are quite similar (reddish box in the bottom right of the matrix); however the similarity between nearly perpendicular and nearly parallel maps is close to random (0.5).

In Figure 6.9b and h, we observe a peculiar characteristic along the edges of the array: in Figure 6.9b, horizontal bumps dominate along the top and bottom edges (horizontal streaks of red), whereas vertical bumps dominate along the left and right edges (vertical streaks of green–blue); the opposite can be observed in Figure 6.9h (vertical streaks of red and horizontal streaks of green–blue). This brings up the prospect that the square geometry of our array may introduce an orientation bias. We are confident that this peculiarity is simply an artefact of the deconstruction procedure: specifically, it is the trivial result that vertical bumps can not form along a horizontal edge, and visa-versa. To support this idea, we run a control experiment where the gratings are masked by a circular aperture, which removes any edge-related bias (Figure 6.13a); the region of stimulation is indicated by the bright circle. As expected, this PO map is nearly identical to the PO map obtained when the entire array is stimulated (Figure 6.13b), dispelling any concern about edge-induced biases. One perhaps meaningful observation is that the similarity between these PO maps (Figure 6.13c) is similar in regions outside the circle of stimulation. This evidence supports our idea that the recurrence is driving selectivity, and not direct stimulation from the gratings.

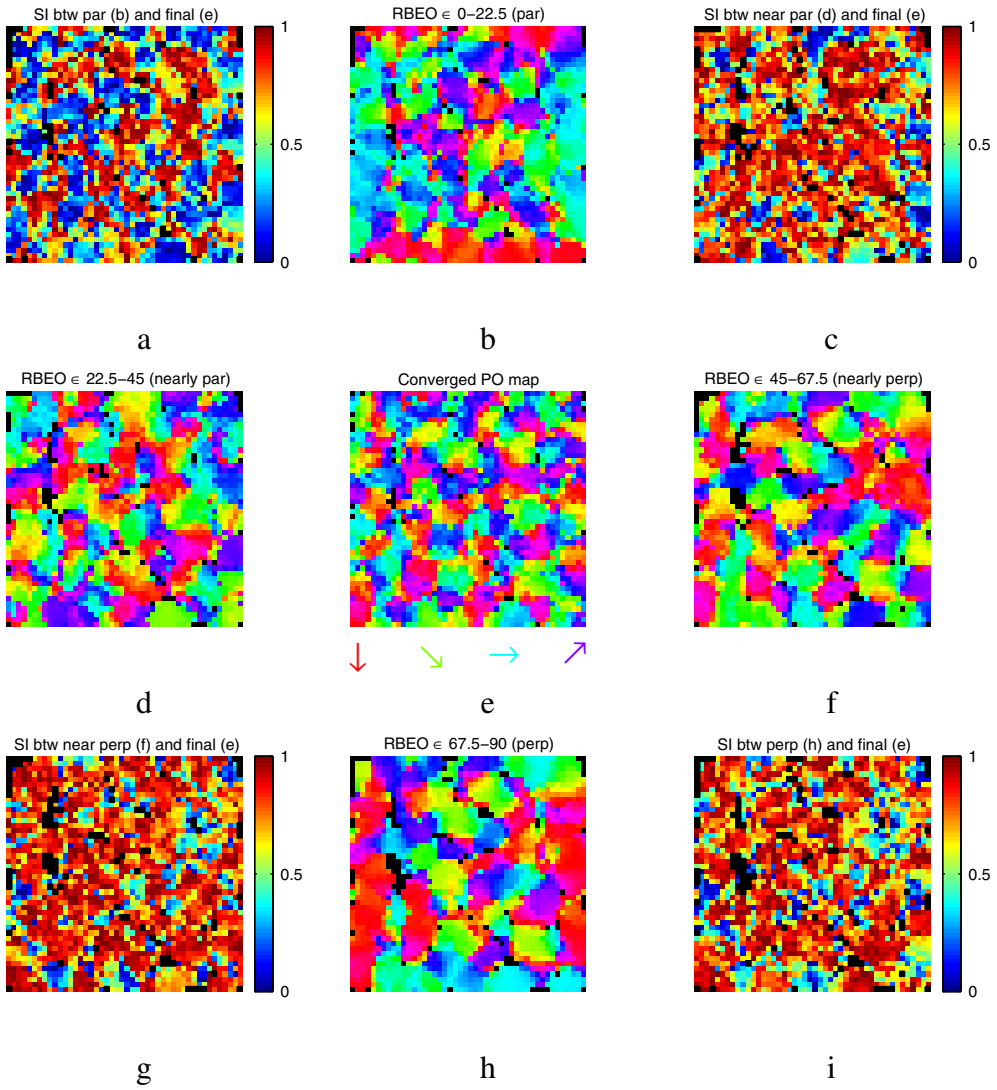


Figure 6.9: Deconstructed PO maps for state 29

(b,d,f,h) PO maps created from subsets of bumps based on how they orient themselves to an input grating (indicated by their RBEO labels). (e) PO map created from entire set of bumps. (a,c,g,i) Comparisons of PO maps created from subset and the entire set using the similarity index (SI). Notice that the PO map created from parallel (par) oriented bumps does not correlate well with the PO map created from the entire set, whereas all other RBEOs (perp, near perp, and near par) do correlate well. Boundary effects can be seen in (b) and (h); this is an artefact, and is simply a result of parallel bumps not being able to form at a perpendicular edge, and visa-versa.

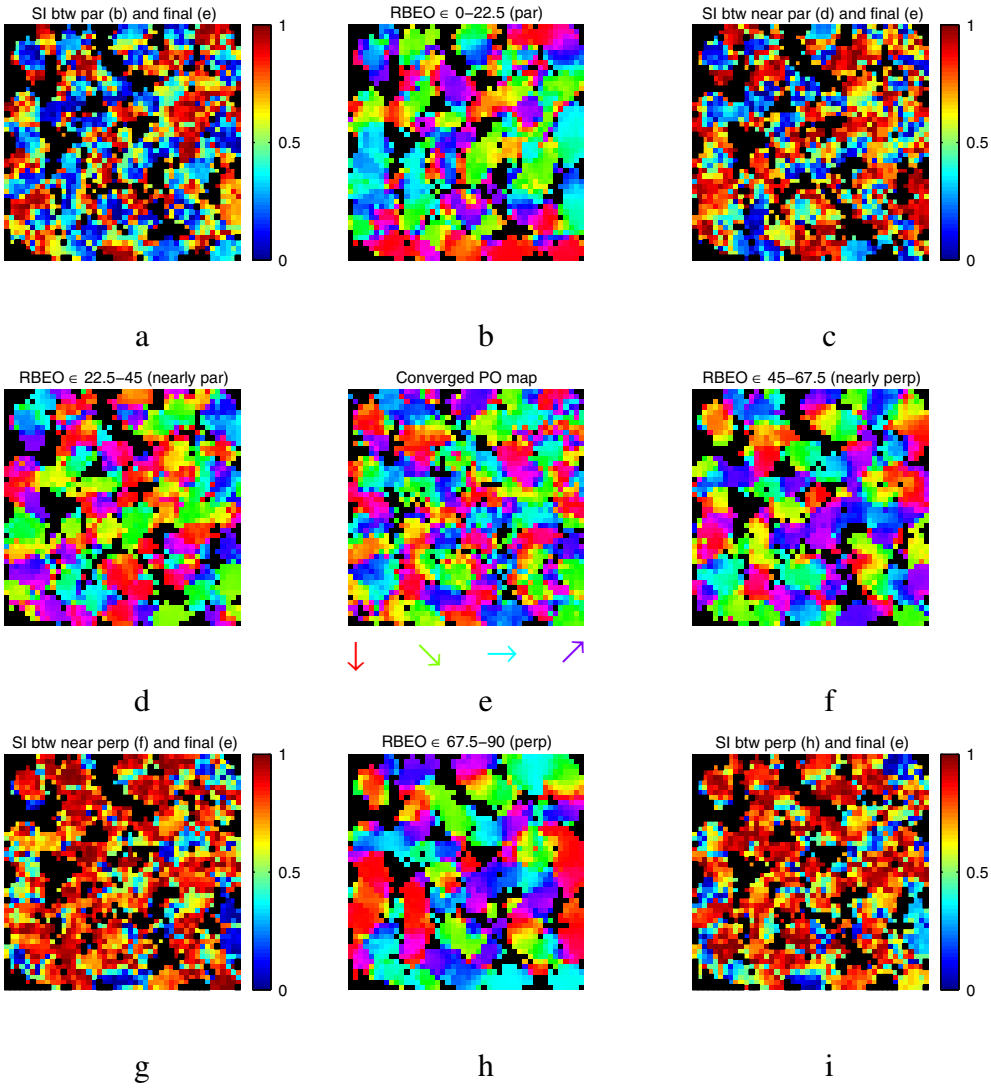


Figure 6.10: Deconstructed PO maps for state 15

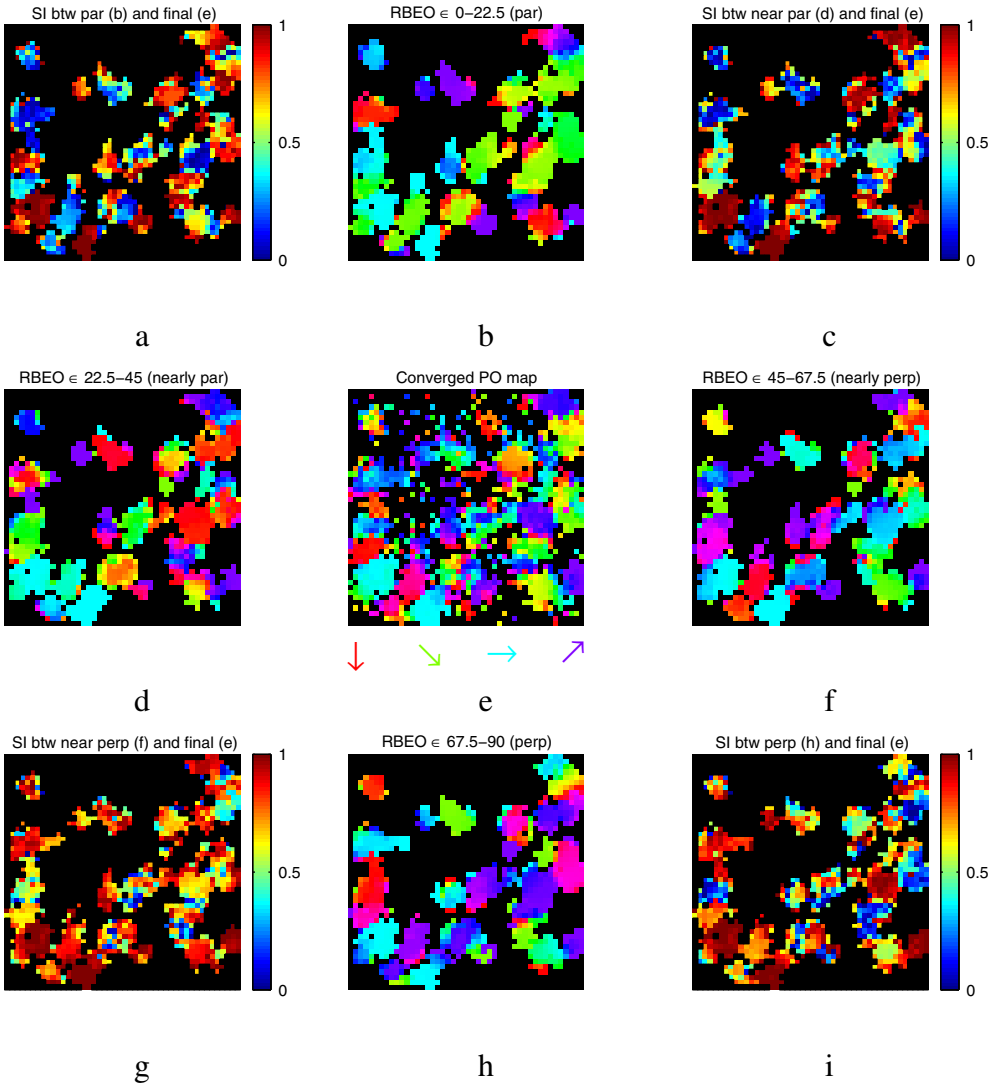
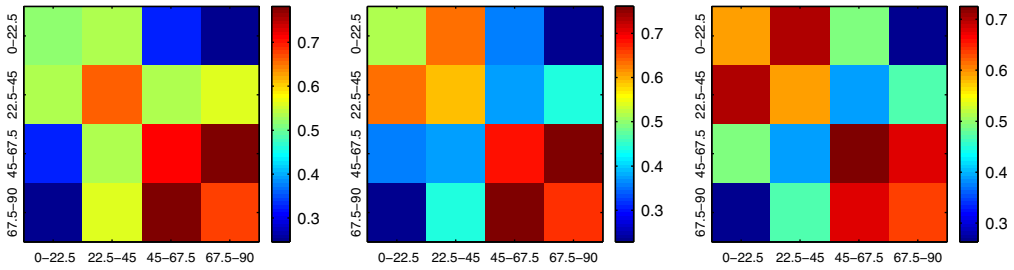


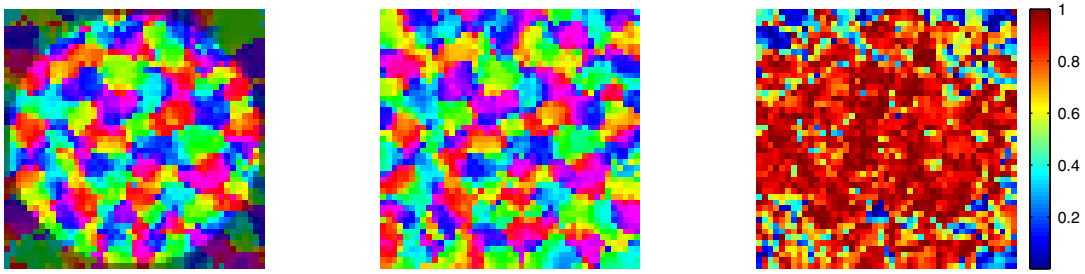
Figure 6.11: Deconstructed PO maps for state 4



a b c

Figure 6.12: Summary of SI between deconstructed PO maps

Pairwise comparisons (average SI) between PO maps created from RBEO subsets for the states 29 (a), 15 (b) and 4, (c). The diagonal entries represent the comparison with the PO map created from the entire set of bumps.



a b c

Figure 6.13: Removing edge effects

(a) PO map obtained by presenting gratings masked with a circular aperture. This map is compared with a PO map obtained when the entire array is stimulated (b); the comparison is shown in (c). This clearly indicates that the square edges of the array do not influence the layout of the map.

Chapter 7

Conclusion

> "As machines become more and more efficient and perfect, so it will become clear that imperfection is the greatness of man."
>
> —Ernst Fischer

In this project, we have presented the bump chip, a silicon model of the primary visual cortex that accounts for the emergence of an orientation map. Two traits set our chip apart from every other man-made system to date: 1) its functional architecture (orientation selectivity) is not specifically engineered *a priori*, and 2) the scaffold for this architecture is innate to the chip itself and exists as an indelible imprint. In other words, our chip attains all of the benefits of a self-organizing learning system *without* having to go through the tedious process of learning. We propose that cortical maps observed in biology are built using a similar design principle. This idea is supported experimentally; orientation maps in kittens are remarkably robust to experimental manipulations, and appear without the need for visual experience.

Our design is based on the simple but powerful idea of opportunism. Building on the work from Ernst et al., our chip uses a recurrent network with local excitatory and distal inhibitory connections to generate periodic patterns of neural activity (bumps). The bumps in our network are pinned to particular locations, determined by transistor mismatch introduced during chip fabrication. Because these pinned bumps respond selectively when driven with different orientations, different groups of neurons are preferentially activated for each stimulus. The resulting map produced by the bump chip reveals that preferences change smoothly across space (due to the spatial extent of bumps), and these preferences repeat at regular intervals (due to bump periodicity) — capturing the essential characteristics of animal maps. The seemingly complicated task of assigning similar selectivities to nearby neurons is cleverly achieved by simply building an imprecise, recurrent network.

Our implementation improved on the previous software model [51] in two respects: First, we added biologically realistic inputs by including separate ON and OFF pathways; the previous model only considered a single luminance channel. To ensure that both ON-driven and OFF-driven orientation maps were aligned, we built our network out of sign-independent cells that are excited by both pathways; this design choice guarantees that bumps interact with both light and dark stimuli in a repeatable manner since both pathways are subjected to the same unique connectivity profile. We made an analogy between our sign-independent cells and 'complex' cells originally described by Hubel and Wiesel based on the observation that neither have identifiable ON–OFF subregions. However, unlike the traditional notion of complex cells that derive their selectivity from orientation-selective simple cells, our sign-independent cells determine their own selectivity through recurrent interactions.

The second improvement was to use a two-layer excitatory–inhibitory recurrent network as the backbone for our design; the model used by Ernst et al. only considered a

one-layer network with a single analog interaction. Although incorporating a distinct inhibitory layer was expensive in terms of silicon area, including this detail payed off in terms of network behaviors: when recurrent feedback was strong, activity was able to ping-pong between the layers dislodging stuck bumps and promoting dynamics — the end result being faster converging orientation maps. In a single-layer network, on the other hand, strong feedback pins bumps harder to their prefered locations, thereby damping dynamics.

The bump chip, like its software predecessor, was able to generate an orientation map from an imprecise recurrent network. However, it was still unclear how orientation selectivity emerges: Do patterns interact in an orderly fashion with inputs, or do the interactions exhibit complex dynamics? In the next section, we discuss an analytical technique for characterizing spatio-temporal activity, which we developed to gain insight into dynamic pattern formation. Then, using results obtained from our analysis, we are able make predictions about how activity in the developing cortex should behave if neural pattern formation plays a role in development. Finally, we conclude this thesis by describing recent experimental evidence that suggests stochastic phenotype expression might be an important computational motif used throughout ontogenesis.

7.1 Bump-based analysis for describing dynamics

We developed a novel technique for characterizing the dynamics in our chip. Our strategy was to capitalize on the idea that the bumps themselves are the quintessential elements of dynamics (and ultimately of orientation selectivity). Based on this insight, we first reduced neural activity within frames to a collection of elliptical particles, and then estimated how these particles changed between frames. Our method contrasts with standard dimension

reduction techniques (principle component analysis and non-negative matrix factorization) that are often used to quantify dynamics; in particular, the basis vectors in these methods are not constrained to have a physical relationship with the underlying neural activity (bumps). As a consequence, these standard techniques are unable to capture non-obvious relationships about dynamics; a prime example is probing whether bumps behave differently depending on where they form in the network (a result that we discovered in Chapter 6).

Using our bump-based analysis, we estimated the diffusion rates of bumps by computing the mean-squared displacements of their centroids for different time intervals. Similar to atoms vibrating in a crystalline lattice, the mean-squared displacement of bumps had two distinct relationships with time: an initial linear relation corresponding to a random walk, and a later saturation revealing confinement. We also noticed that unlike physical particles, bumps are able to vanish, coalesce, and change shape over time. To quantify these 'non-classical' dynamics, we computed the likelihood a bump vanishes–appears in a predefined area. This analysis revealed that regenerating bumps (i.e., they vanish and then reappear) are repelled from the previously occupied site, suggesting attractive–repulsive interactions across time. Finally, we also showed that bumps teleport (one bump will vanish, and a second will reappear outside of the confinement region); we called this type of motion staccato, reflecting coordinated disjointed movements. Our results confirmed that even though bumps do behave like particles vibrating in a lattice, they can also display diverse non-classical behaviors.

7.2 Testable predictions for biological maps

The model that we have proposed is able to reproduce the characteristic features of biological orientation maps. Furthermore, the bump model's basic algorithm is both simple and robust; the mere fact that this model works at all in silicon attests to its robustness. But does the bump model accurately reflect the process of cortical development, or should we heed the words of the ever cynical H.L. Mencken: *"For every complex problem there is an answer that is clear, simple, and wrong."*?

We highlight three general predictions that our model makes about the establishment of an orientation map (up until a week after eye-opening), all of which can be explored using modern day voltage-sensitive dye (or calcium) imaging techniques and–or single-cell recordings.

The most obvious prediction of the bump model is that recurrent interactions are responsible for orientation selectivity in the developing cortex. Simply put: Do neural activity patterns in V1 act like bumps (in single-condition maps), and if so, are these bumps in any way related to orientation-selective responses? To answer the first question, we can analyze single-frame images of cortical activity (single-condition maps) and test if 'regenerated' bumps experience a repulsive force (as described in Chapter 5), for instance. Repulsive–attractive interactions between self-sustaining regions of neural activity would suggest recurrent interactions are at play.

To answer the second question, consider imaging the cortex while simultaneously recording from a single cell. The bump model predicts that every time the cell responds selectively to a stimulus, a 'bump' from the imaging data should also be crossing its path. If, however, the cell is selective when a bump is not nearby, we can conclude that something other

than bump interactions determine this cell's selectivity (as predicted by simple feedforward models).

A second prediction of the bump model is that network oscillations are related to bump diffusion, as oscillations provide a mechanism to dislodge stuck bumps, an inevitable consequence of heterogeneity.. This relationship could be directly tested by comparing bump diffusion rates in awake and sleeping animals, which correspond to fast and slow oscillations, respectively.

The final prediction that we highlight, which in our opinion is the most exciting result of this project, is that the way a bump orients itself in relation to the stimulus determines whether that bump contributes to a robust orientation signal (Chapter 6). For example, bumps that orient themselves perpendicular to the stimulus tend to activate cells that are preferentially selective to that stimulus (these bumps tend to land in the 'correct' locations); by contrast, bumps that orient themselves parallel to the stimulus activate cells without any preference (these bumps tend to land at random network locations). If confirmed, this relationship would strongly suggest that the bump model is in fact credible, since no competing model (to our knowledge) relies on a similar mechanism.

7.3 Does biology really use noise?

In the past few years, there has been a growing body of evidence that demonstrates phenotype diversity can be influenced by stochastic variations [77]. In other words, certain attributes of an organism are 'noisy' since they are *not* solely determined by intrinsic factors (genes and molecules). Ostensibly, this noise is undesirable and runs contrary to the law of natural selection since favorable attributes can not be reliably passed along through

the generations; that is, unless random phenotype expression itself offers some advantage to the organism. A recent study suggests that this indeed may be the case.

A group from Berkeley has demonstrated for the first time that stochastic expression contributes to the robustness of a viral infection. The most significant obstacle in eradicating the human immunodeficiency virus type 1 (HIV-1) from patients is long-lived latently infected host cells that do not show any appreciable sign of infection. To the dismay of researchers, neither genetic or cellular factors have been able to account for the observed diversity of latency durations. The Berkeley group discovered why in 2005: stochastic variations in a positive feedback loop drive the phenotype diversity; in contrast, control viruses that lacked this feedback loop exhibited deterministic latencies [77]. It is often cited in evolutionary biology that "chance favors the prepared genome" [78]; in light of this new evidence, perhaps we can also say that "a prepared genome favors chance".

Taking a step back, we can ponder whether this viral example is an isolated case, or if perhaps stochastic phenotype expression is an important computational motif throughout ontogenesis. Neural feedback loops, for example, might also take advantage of random variations. In this project, we have proposed that cortical feature maps, which are putatively the seat for higher cognition in humans, emerge from this same basic principle.

Bibliography

[1] C.A. Mead. *Analog VLSI and neural systems*. Addison Wesley, Reading, Ma, 1989.

[2] K. Zaghloul. *A silicon implementation of a novel model for retinal processing*. PhD thesis, U. Penn, 2001.

[3] K. Boahen. Neuromorphic microchips. *Scientific American*, 292(5):56–63, 2005.

[4] J. Locke. *An essay concerning human understanding.* Edition abridged by Kenneth Winkler., New York: Hackett, 1669.

[5] S. Diamond. Four hundred years of instinct controversy. *Behav Genet*, 4(3):237–52, 1974.

[6] M. Sur and C. A. Leamey. Development and plasticity of cortical areas and networks. *Nat Rev Neurosci*, 2(4):251–62, 2001.

[7] E.R Kandel, J.H. Schwartz, and T.M. Jessel. *Principles of Neural Science*. McGraw-Hill, 4 edition, 2000.

[8] R. Yuste and M. Sur. Development and plasticity of the cerebral cortex: from molecules to maps. *J Neurobiol*, 41(1):1–6, 1999.

[9] T. N. Wiesel. Early explorations of the development and plasticity of the visual cortex: A personal view. *J Neurobiol*, 41(1):7–9, 1999.

[10] D. H. Hubel and T. N. Wiesel. Binocular interaction in striate cortex of kittens reared with artificial squint. *J Neurophysiol*, 28(6):1041–59, 1965.

[11] C. Blakemore and R. C. Vansluyters. Innate and environmental factors in development of kittens visual-cortex. *Journal of Physiology-London*, 248(3):663–716, 1975.

[12] Y. Fregnac and M. Imbert. Early development of visual cortical cells in normal and dark-reared kittens: relationship between orientation selectivity and ocular dominance. *J Physiol*, 278:27–44, 1978.

[13] K. Albus and W. Wolf. Early post-natal development of neuronal function in the kitten's visual cortex: a laminar analysis. *J Physiol*, 348:153–85, 1984.

[14] B. O. Braastad and P. Heggelund. Development of spatial receptive-field organization and orientation selectivity in kitten striate cortex. *J Neurophysiol*, 53(5):1158–78, 1985.

[15] K. Krug, C. J. Akerman, and I. D. Thompson. Responses of neurons in neonatal cortex and thalamus to patterned visual stimulation through the naturally closed lids. *J Neurophysiol*, 85(4):1436–43, 2001.

[16] F. Sengpiel and P. C. Kind. The role of activity in development of the visual system. *Curr Biol*, 12(23):R818–26, 2002.

[17] G. D. Mower, D. Berry, J. L. Burchfiel, and F. H. Duffy. Comparison of the effects of dark rearing and binocular suture on development and plasticity of cat visual cortex. *Brain Res*, 220(2):255–67, 1981.

[18] I. Godecke and T. Bonhoeffer. Development of identical orientation maps for two eyes without common visual experience. *Nature*, 379(6562):251–4, 1996.

[19] M. C. Crair, D. C. Gillespie, and M. P. Stryker. The role of visual experience in the development of columns in cat visual cortex. *Science*, 279(5350):566–70, 1998.

[20] L. E. White, D. M. Coppola, and D. Fitzpatrick. The contribution of sensory experience to the maturation of orientation selectivity in ferret visual cortex. *Nature*, 411(6841):1049–52, 2001.

[21] F. Sengpiel, P. Stawinski, and T. Bonhoeffer. Influence of experience on orientation maps in cat visual cortex. *Nat Neurosci*, 2(8):727–32, 1999.

[22] C. S. Furmanski and S. A. Engel. An oblique effect in human primary visual cortex. *Nat Neurosci*, 3(6):535–6, 2000.

[23] D. M. Coppola and L. E. White. Visual experience promotes the isotropic representation of orientation preference. *Vis Neurosci*, 21(1):39–51, 2004.

[24] B. Chapman, M. P. Stryker, and T. Bonhoeffer. Development of orientation preference maps in ferret primary visual cortex. *J Neurosci*, 16(20):6443–53, 1996.

[25] B. Chapman and M. P. Stryker. Development of orientation selectivity in ferret visual cortex and effects of deprivation. *J Neurosci*, 13(12):5251–62, 1993.

[26] T. Bonhoeffer and A. Grinvald. The layout of iso-orientation domains in area 18 of cat visual cortex: optical imaging reveals a pinwheel-like organization. *J Neurosci*, 13(10):4157–80, 1993.

[27] S. LeVay, T. N. Wiesel, and D. H. Hubel. The development of ocular dominance columns in normal and visually deprived monkeys. *J Comp Neurol*, 191(1):1–51, 1980.

[28] M. P. Stryker and S.L. Strickland. Physiological segregation of ocular dominance columns depends on the pattern of afferent electrical activity. *Invest Opthalmol Vis Sci*, 25(278), 1984.

[29] M. Weliky and L. C. Katz. Disruption of orientation tuning in visual cortex by artificially correlated neuronal activity. *Nature*, 386(6626):680–5, 1997.

[30] M. Weliky. Recording and manipulating the in vivo correlational structure of neuronal activity during visual cortical development. *J Neurobiol*, 41(1):25–32, 1999.

[31] M. Meister, R. O. Wong, D. A. Baylor, and C. J. Shatz. Synchronous bursts of action potentials in ganglion cells of the developing mammalian retina. *Science*, 252(5008):939–43, 1991.

[32] K. D. Miller, E. Erwin, and A. Kayser. Is the development of orientation selectivity instructed by activity? *J Neurobiol*, 41(1):44–57, 1999.

[33] I. Godecke and B. Chapman. Effects of on-center retinal ganglion cell blockade on development of visual cortical responses. *Soc Neurosci Abstr*, 24:1051, 1998.

[34] N. V. Swindale. The development of topography in the visual cortex: a review of models. *Network*, 7(2):161–247, 1996.

[35] R Miikkulainen, J.A. Bednar, Y Choe, and J Sirosh. *Computational Maps in the Visual Cortex*. Springer, Berlin, 2005.

[36] M. B. Elowitz, A. J. Levine, E. D. Siggia, and P. S. Swain. Stochastic gene expression in a single cell. *Science*, 297(5584):1183–1186, 2002.

[37] C. V. Malsburg. Self-organization of orientation sensitive cells in striate cortex. *Kybernetik*, 14(2):85–100, 1973.

[38] V. Braitenberg and C. Braitenberg. Geometry of orientation columns in the visual-cortex. *Biological Cybernetics*, 33(3):179–186, 1979.

[39] D.O. Hebb. *The organization of behavior*. Wiley, New York, 1949.

[40] M. B. Dalva, M. Weliky, and L. C. Katz. Relationships between local synaptic connections and orientation domains in primary visual cortex. *Neuron*, 19(4):871–80, 1997.

[41] R. Linsker. From basic network principles to neural architecture: emergence of orientation columns. *Proc Natl Acad Sci U S A*, 83(22):8779–83, 1986.

[42] K. D. Miller and D. J. C. Mackay. The role of constraints in hebbian learning. *Neural Computation*, 6(1):100–126, 1994.

[43] D. L. Ringach. Haphazard wiring of simple receptive fields and orientation columns in visual cortex. *J Neurophysiol*, 92(1):468–76, 2004.

[44] D. S. Kim and T. Bonhoeffer. Reverse occlusion leads to a precise restoration of orientation preference maps in visual-cortex (vol 370, pg 370, 1994). *Nature*, 372(6502):196–196, 1994.

[45] A. M. Turing. The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London Series B-Biological Sciences*, 237(641):37–72, 1952.

[46] G Turk. Generating textures on arbitrary surfaces using reaction-diffusion. In *Computer Graphics*, volume 25, pages 289–98. ACM, 1991.

[47] R. T. Liu, S. S. Liaw, and P. K. Maini. Two-stage turing model for generating pigment patterns on the leopard and the jaguar. *Phys Rev E Stat Nonlin Soft Matter Phys*, 74(1 Pt 1):011914, 2006.

[48] M. Sur, A. Angelucci, and J. Sharma. Rewiring cortex: the role of patterned activity in development and plasticity of neocortical circuits. *J Neurobiol*, 41(1):33–43, 1999.

[49] H. Meinhardt. Models of biological pattern formation: common mechanism in plant and animal development. *Int J Dev Biol*, 40(1):123–34, 1996.

[50] Y. Chen and A. F. Schier. The zebrafish nodal signal squint functions as a morphogen (vol 411, pg 607, 2001). *Nature*, 412(6846):566–+, 2001.

[51] U. A. Ernst, K. R. Pawelzik, C. Sahar-Pikielny, and M. V. Tsodyks. Intracortical origin of visual maps. *Nature Neuroscience*, 4(4):431–436, 2001.

[52] H. Z. Shouval, D. H. Goldberg, J. P. Jones, M. Beckerman, and L. N. Cooper. Structured long-range connections can provide a scaffold for orientation maps. *Journal of Neuroscience*, 20(3):1119–1128, 2000.

[53] S. Borkar. Designing reliable systems from unreliable components: The challenges of transistor variability and degradation. *IEEE Micro*, 25:10–16, 2005.

[54] B. Taba and K. Boahen. Balancing guidance range and strength optimizes self-organization by silicon growth cones. *Artificial Neural Networks: Formal Models and Their Applications - Icann 2005, Pt 2, Proceedings*, 3697:1027–1034, 2005.

[55] C. R. Laing and C. C. Chow. Stationary bumps in networks of spiking neurons. *Neural Computation*, 13(7):1473–1494, 2001.

[56] D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J Physiol*, 160:106–54, 1962.

[57] M. Usher, M. Stemmler, and Z. Olami. Dynamic pattern-formation leads to 1/f noise in neural populations. *Physical Review Letters*, 74(2):326–329, 1995.

[58] A. G. Andreou and K. A. Boahen. Translinear circuits in subthreshold mos. *Analog Integrated Circuits and Signal Processing*, 9(2):141–166, 1996.

[59] K. Boahen. *Retinomorphic Vision Systems: Reverse Engineering the Vertebrate Retina*. PhD thesis, California Institute of Technology, 1997.

[60] E. Culurciello, R. Etienne-Cummings, and K. A. Boahen. A biomorphic digital image sensor. *Ieee Journal of Solid-State Circuits*, 38(2):281–294, 2003.

[61] J. Lin. Word-serial address-event transceiver layout compiler. *The Neuromorphic Engineer*, 1(2):2, 2004.

[62] M. Mahowald. *An Analog VLSI System for Stereoscopic Vision*. Kluwer Academic Publisher, Norwell, Ma, 1994.

[63] K. A. Boahen. A burst-mode word-serial address-event link-i transmitter design. *Ieee Transactions on Circuits and Systems I-Regular Papers*, 51(7):1269–1280, 2004.

[64] K. A. Boahen. A burst-mode word-serial address-event link-ii: Receiver design. *Ieee Transactions on Circuits and Systems I-Regular Papers*, 51(7):1281–1291, 2004.

[65] P.A. Merolla, J.V. Arthur, and J Wittig. Usb revolution. *The Neuromorphic Engineer*, 2(2):10–11, 2005.

[66] S.K. Kachigan. *Multivariate Statistical Analysis: A Conceptual Introduction*. Radius Press, 1991.

[67] G. C. DeAngelis, I. Ohzawa, and R. D. Freeman. Spatiotemporal organization of simple-cell receptive fields in the cat's striate cortex. ii. linearity of temporal and spatial summation. *J Neurophysiol*, 69(4):1118–35, 1993.

[68] D. L. Ringach, R. M. Shapley, and M. J. Hawken. Orientation selectivity in macaque v1: diversity and laminar dependence. *J Neurosci*, 22(13):5639–51, 2002.

[69] G. G. Blasdel and G. Salama. Voltage-sensitive dyes reveal a modular organization in monkey striate cortex. *Nature*, 321(6070):579–85, 1986.

[70] P. H. Schiller, B. L. Finlay, and S. F. Volman. Quantitative studies of single-cell properties in monkey striate cortex. v. multivariate statistical analyses and models. *J Neurophysiol*, 39(6):1362–74, 1976.

[71] J. Anderson, I. Lampl, I. Reichova, M. Carandini, and D. Ferster. Stimulus dependence of two-state fluctuations of membrane potential in cat visual cortex. *Nat Neurosci*, 3(6):617–21, 2000.

[72] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948.

[73] A. Borst and F. E. Theunissen. Information theory and neural coding. *Nat Neurosci*, 2(11):947–57, 1999.

[74] F. Rieke, D. Warland, R.R. de Ruyter Van Steveninck, and W. Bialek. *Spikes: Exploring the neural code*. MIT Press, Cambridge, Ma, 1997.

[75] J. D. Victor. Asymptotic bias in information estimates and the exponential (bell) polynomials. *Neural Computation*, 12(12):2797–2804, 2000.

[76] N. Fisher. *Statistical Analysis of Circular Data*. Cambridge Univ. Press, Cambridge, UK, 1993.

[77] L. S. Weinberger, J. C. Burnett, J. E. Toettcher, A. P. Arkin, and D. V. Schaffer. Stochastic gene expression in a lentiviral positive-feedback loop: Hiv-1 tat fluctuations drive phenotypic diversity. *Cell*, 122(2):169–82, 2005.

[78] L. H. Caporale. Chance favors the prepared genome. *Ann N Y Acad Sci*, 870:1–21, 1999.