

# STATISTICAL SIGNAL PROCESSING TOOLS FOR ANALYZING LARGE-SCALE NEURAL ENSEMBLES

By

Mehdi Aghagolzadeh

A DISSERTATION

Submitted to  
Michigan State University  
in partial fulfillment of the requirements  
for the degree of

DOCTOR OF PHILOSOPHY

Electrical Engineering

2012

## ABSTRACT

### STATISTICAL SIGNAL PROCESSING TOOLS FOR ANALYZING LARGE-SCALE NEURAL ENSEMBLES

By Mehdi Aghagolzadeh

Understanding how a neuron processes information and communicates with other neurons is key to unravel the brain's mechanisms underlying perception, learning and motor processing. Key to characterize neurons acting in concert is the ability to simultaneously measure their firing patterns in awake behaving subjects. Microelectrode arrays (MEAs) implanted in the brain allow simultaneous monitoring of the activity of large ensembles of cells while subjects carry out specific tasks. Isolating the spike pattern characterizing each cell's signature firing requires sophisticated signal processing algorithms, particularly to track the nonstationary behavior of these waveforms over extended periods of time. In this thesis, we introduce a compressive spike sorting technique that discriminates spike patterns from individual neurons using a sparse representation of the ensemble raw data. An iterative learning algorithm is introduced to estimate and adapt a set of optimal thresholds that maximize the separability between spike classes while minimizing the average waveform reconstruction error. Once derived, spike trains are then used to infer functional connectivity patterns among the recorded neural ensemble constituents. We introduce two information-theoretic approaches within the realm of graphical models that capture the spatiotemporal dependency between neurons' spiking patterns. Specifically, the class of spatiotemporal maximum entropy models is shown to incorporate higher-order interactions. We demonstrate the richness of these techniques in improving our understanding of neural function and dysfunction at the millisecond and micron resolutions, and their potential to be applied in emerging applications of brain machine interface technology to help improve the lifestyle of people with severe disabilities.

*To my wife, Hamideh Rezaee, for her unconditional support*

## **ACKNOWLEDGEMENTS**

I would like to thank my advisor Dr. Karim Oweiss for his continuous guidance and support since I joined Michigan State University at August 2007. His patience, dedication and encouragement in addition to meeting me at least once every week and sending me to prestigious conferences helped me in advancing my research and writing skills. I would also like to express appreciation to my dissertation committee members, Dr. John Deller, Dr. Haydar Radha, and Dr. Rong Jin. I also like to thank all my previous and current lab mates that I have coauthored papers with: Seif Eldawltly, Fei Zhang, Jianbo Liu, Ki Yong Kwon, Faisal Abunimeh, Ali Mohebi, and John Daly. Finally I would like to thank the National Institute of Health (NIH) for the generous funding and the department of electrical and computer engineering and the graduate school for summer scholarships.

# TABLE OF CONTENTS

List of Tables.....	viii
List of Figures.....	ix
Table of Abbreviations.....	xiii
Introduction.....	1
Overview of Neural Analysis Tools .....	6
2.1. Spike Detection and Sorting.....	8
2.2. Generalized Linear Model .....	12
Neural Activity in the Sparse Representation Domain .....	15
3.1. Spike Trains as Point Processes.....	17
3.2. Sparse Representation .....	20
3.3. Computational Complexity of Spike Sorting in the Sparse Domain.....	24
3.4. Spike Class Generation and Separability .....	26
3.5. Decoding Performance .....	32
3.6. Discussion.....	43
Sorting and Tracking Neural Activity via Simple Thresholding .....	46
4.1. Spike Sorting as a Sparse Optimization Problem.....	47
4.1.1. Optimal Thresholding.....	50
4.1.2. Threshold selection to maximize sorting performance.....	52

4.1.3.	Multilevel Wavelet Decomposition.....	57
4.2.	Collecting Neural Data.....	59
4.2.1.	Generating Surrogate Data with Known Waveform Characteristics.....	59
4.2.2.	Tracking Neurons over Long-Term Recordings.....	62
4.2.3.	Considerations for Practical Implementation.....	67
4.3.	Compressive Spike Sorting Performance.....	69
4.3.1.	Spike Sorting Performance vs. SNR and Compression Rate.....	69
4.3.2.	Wavelet Basis Function vs. Sampling Rate.....	70
4.3.3.	Analysis of Real Data.....	72
4.4.	Discussion.....	82
	Spatiotemporal Graphical Models for Ensemble Spike Train Analysis.....	86
5.1.	Pairwise Maximum Entropy Models.....	87
5.1.1.	Principle of Maximum Entropy.....	89
5.1.2.	Instantaneous Maximum Entropy Model.....	90
5.1.3.	Spatiotemporal Maximum Entropy Model.....	91
5.1.4.	Training Maximum Entropy Models.....	96
5.2.	V1 Population Encoding in Response to Stimulus Grating.....	98
5.2.1.	Single Unit Analysis.....	101
5.2.2.	Multiunit Analysis.....	102
5.3.	Discussion.....	116

Hypergraph Models with Higher Order Interactions Better Explain the Network Dynamics.....	117
6.1. Hypergraph Maximum Entropy Model.....	119
6.2.1. Factoring Multi-information .....	121
6.2.2. Graphical representation.....	123
6.2.3. Cooperative Model.....	125
6.2.4. Minimum Entropy Distance .....	127
6.2. Probabilistic Spiking Model with Synaptic Plasticity .....	129
6.1. Probabilistic Spiking Neurons.....	129
6.2. Task Learning .....	132
6.3. Experimental Results .....	142
6.3.1. Order of the Encoding Model.....	142
6.3.2. Model Consistency .....	144
6.4. Discussion .....	150
Concluding Remarks .....	155
REFERENCES.....	159

## List of Tables

Table 4.1 Pseudo code for determining the optimal thresholds by (a) minimizing the MSE, (b) maximizing the sorting performance.....	55
Table 6.1 Values of the parameters used for the model of equations (6-16), (6-17), (6-18) and (6-19).....	133

## List of Figures

Figure 1.1 Illustration of the two-category data analysis applied to microelectrode array recordings...	2
Figure 2.1 Pattern recognition approach to spike sorting.....	10
Figure 2.2 PCA for feature extraction and EM for unsupervised cluster cutting of 6469 spike event .....	11
Figure 2.3 Schematic of the network structure used to simulate the spike trains.....	14
Figure 3.3 Schematic diagram of a typical data flow in a neuro-motor prosthetic application.....	16
Figure 3.2 Sparse representation of sample events.....	21
Figure 3.3 Five units obtained from recordings in an anesthetized preparation.....	30
Figure 3.4 Unit isolation quality of the data.....	31
Figure 3.5 Schematic of encoding 2D, non goal-directed arm movement: the sample network of neurons is randomly connected with positive and negative connections.....	36
Figure 3.6 Top-left: 400 msec segment of angular direction from a movement trajectory superimposed on tuning “bands” of five representative units.....	38
Figure 3.7 Average mutual information (in bits) between movement direction, $\theta$ , and rate estimators averaged across the two subgroups of neurons in the entire population as a function of decomposition level (i.e. kernel size).....	40
Figure 3.8 Decoding performance of a sample 2D movement trajectory.....	41

Figure 3.9 Computational complexity of PCA/EM/Gaussian kernel and the compressed sensing method.....	42
Figure 4.1 The distribution of spike events in a 2D space obtained using PCA.....	56
Figure 4.2 Schematic diagram for obtaining the sparse representation from the raw recordings using a 4-level wavelet decomposition in a tree structure using the Symlets 2, 4, and 6, at 25 kHz.....	60
Figure 4.3 Distribution of spike PCA features in a 3D space.....	62
Figure 4.4 Tracking neurons for multiday recordings on a single electrode.....	64
Figure 4.5 Tracking neurons for single day recordings.....	66
Figure 4.6 Schematic diagram of the data flow in the implantable wireless system.....	69
Figure 4.7 Average TPR of an FDA classifier trained for different SNR and compression rates.....	71
Figure 4.8 Effect of wavelet basis choice on the average TPR for variable sampling rates between 12 and 28 kHz, and SNRs of 0, 0.5 and 1.5 dB.....	73
Figure 4.9 A 10 second demonstration of in vivo recording from the rat's somatosensory cortex....	74
Figure 4.10 The compression rate and MSE for the iterative simulated annealing learning.....	76
Figure 4.11 Rate-distortion scatter plot. Each dot represents the rate and the distortion for a random threshold drawn from a uniform distribution.....	79
Figure 4.12 Spike sorting performance of the compressive spike sorting (CSS) vs. the conventional PCA sorting for neurons tracked over 1, 2 and 3 days span.....	80
Figure 5.1 Representation of hypothetical instantaneous patterns, $X^{(i)}$ (for $i = 1,2,3$ ).....	93

Figure 5.2 First-order Markov representation of pairwise maximum entropy model.....	95
Figure 5.3 Experimental setup: recording from the V1 area of an anesthetized cat in response to drifting grating stimulus with tetrode microwires.....	99
Figure 5.4 The circular representation demonstrates the 16 different stimuli in the center with the arrows indicating the movement direction of the drifting bars.....	103
Figure 5.5 Predicting spatiotemporal patterns of activity for 1 <sup>st</sup> order maximum entropy model.....	105
Figure 5.6 Predicting spatiotemporal patterns of activity 2 <sup>nd</sup> order maximum entropy model.....	107
Figure 5.7 Independent encoding model.....	110
Figure 5.8 Spatiotemporal maximum entropy encoding model.....	112
Figure 5.9 Actual counted number of patterns in the collective activity of the neural population (solid blue line) and estimated counts by the independent, instantaneous maximum entropy, first and second-order spatiotemporal maximum entropy models.....	115
Figure 6.1 A stimulus entropy representation by a population of 3 neurons.....	122
Figure 6.2 A hypergraph including four factors, representing a Markov network model of a population of six neurons.....	127
Figure 6.3 A simplified version of the network structure.....	134
Figure 6.4 Schematic of the statistical learning rule. Neurons A and B are non-cooperative layer neurons that excite neurons cooperative layer neurons C, D and E.....	136

Figure 6.5 Mean connectivity strength versus trial number for functionally correlated (blue) and uncorrelated (red) cooperative layer neurons.....	137
Figure 6.6 An example of the connectivity between one cooperative layer neuron (neuron 103) and four non-cooperative layer neurons.....	139
Figure 6.7 Probability of number of neurons firing within a 10 ms window for actual data generated from the model and a jittered version of the data for cooperative layer neurons 99 to 108.....	143
Figure 6.8 Sample data from 60 cooperative layer neurons generated by the model during one trial after task learning was completed.....	145
Figure 6.9 Connected information for different orders of the cooperative model for a fixed subpopulation size ( $n = 10$ ) .....	146
Figure 6.10. Inferred factor-graphs for sample targets ‘1’, and ‘8’.....	147
Figure 6.11 Model fit for a sample subpopulation of 10 neurons for: 1- Independent model (blue), 2- Maximum entropy model (red), 3- MinED cooperative model (green).....	148
Figure 6.12 The Kullback-Leibler distance between the true and the single trial distribution of states for each model.....	148
Figure 6.13 The KL-divergence for independent, maximum entropy and cooperative models as a function of the size of the observed population.....	149
Figure 6.14 Two sample inputs of drifting grating stimulus.....	151

## TABLE OF ABBREVIATIONS

MEA	Multielectrode Array
BMI	Brain Machine Interface
GLM	Generalized Linear Model
EEG	Electroencephalogram
fMRI	Functional Magnetic Resonance Imaging
LFP	Local Field Potential
PCA	Principal Component Analysis
ISI	Inter Spike Interval
EM	Expectation Maximization
DWT	Discrete Wavelet Transform
SR	Separability Ratio
GMM	Gaussian Mixture Model
MAP	Maximum a Posteriori
FDA	Fisher Discriminant Analysis
LFSR	Linear Feedback Shift Register
CSS	Compressive Spike Sorting
RLE	Run Length Encoding
KL	Kullback-Leibler
MRF	Markov Random Fields
RGC	Retinal Ganglion Cell
MinED	Minimum Entropy Distance
STDP	Spike Timing Dependent Plasticity
PSTH	Post Stimulus Time Histograms

# 1

## Introduction

Understanding the role of neurons in brain functionality is a cornerstone in neuroscience, psychology and cognitive sciences [1]. Neurons participate in encoding sensory information, motor intentions, and decision-making via excitatory or inhibitory connectivity to thousands of neurons. It is the collective behavior of these connected neuronal elements that form the important aspects of complex behavior [2]. The ability to simultaneously record the activity of an ensemble of neurons in vivo using multielectrode arrays (MEAs) implanted in selected brain areas has made it feasible to study functional connectivity between neurons while awake behaving subjects carry out specific functions [3-6]. The advantage of these arrays is their ability to sample from the neuronal activity at a high temporal and spatial resolution compared to fMRI that lacks temporal resolution, and EEG or ECoG recordings that lack spatial resolution [7].

**The objective of this dissertation is to design statistical signal processing tools to extract the critical information from large-scale recordings of neuronal populations and model the way neurons communicate with each other.** These tools can in general be categorized into two categories, first detecting and sorting action potentials from the background neural activity, and second constructing models among neuronal constituents. Figure 1.1 illustrates the two-category

analysis applied to extracellular recordings of MEAs.

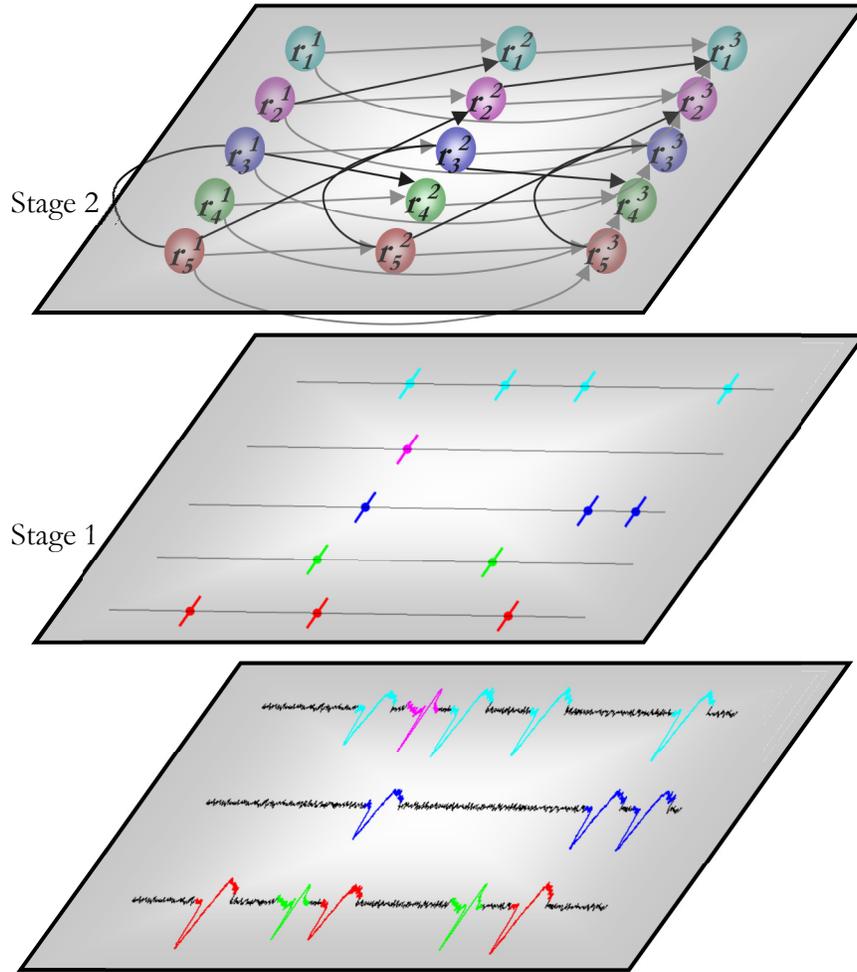


Figure 1.1 Illustration of the two-category data analysis applied to microelectrode array recordings. The bottom plate demonstrates the recorded neural data from three channels and the noisy background activity. Each channel can record from one or more neurons simultaneously, e.g. the first and third rows record two neurons, while the second row records only from one neuron. The middle plate demonstrates the spike trains for the 5 segregated neurons, represented by the exact time stamps of individual events. The top plate demonstrates a spatiotemporal model fit to the collective neural activity of the observed population. The hypothetical interactions among neuronal constituents can be either at instantaneous time instants to represent bidirectional connections or at

different time instants to represent unidirectional connections. For interpretation of the references to color in this and all other figures, the reader is referred to the electronic version of this dissertation.

Chapter 2 provides an overview on some basic neural analysis tools that have been repeatedly used in this dissertation for analyzing extracellular recordings of MEAs. The chapter begins with a general introduction to neurons and the different methods developed for recording its activity. Then, spike detection and sorting algorithms, which are the first step in analyzing any extracellular recordings of a mixture of neural activities, are described. We then explain generalized linear models (GLMs) as the generative model for simulating neural encoding data that has been the centerpiece in evaluating the decoding performance and connectivity inference in this dissertation.

Chapter 3 introduces a novel spike sorting algorithm that operates directly on the sparse representation of the neural recordings, referred to as the compressive spike sorting. In this chapter, we first introduce the state-of-the-art algorithms for detecting, sorting and transmitting neural data from the site of an implantable MEA recording to an external observer. The objective is to automatically detect and sort action potentials in a compressed domain based on exploiting a sparse representation of the extracellular spike waveforms using multilevel wavelet decomposition transform. We will show that by restricting this sparse representation to a subset of projections that simultaneously preserve features of the spike waveforms, we can reasonably approximate the spike trains and the instantaneous firing rates of the recorded neurons. This approach eliminates multiple steps from the typical processing path of neural signals that are customarily performed for instantaneous neural decoding, thus reduces the computational complexity that is essential for an actual implementation of miniaturized implantable brain machine interface (BMI) systems. We compared the computational complexity of the compressive spike sorting with the state-of-the-art

spike sorting and demonstrated the substantial savings in power and hardware resources that is the cornerstone of the design of implantable BMI systems.

Chapter 4 introduces an iterative learning approach for estimating the thresholds for the sparse representation. It was shown in the previous chapter that these thresholds are neuron-specific and permit direct sorting in the compressed domain. The simple thresholding mechanism was demonstrated to discriminate between signals of different neurons, given an efficient multidimensional feature extraction technique. Since thresholding is a binary classifier, we show here that complex nonlinear decision boundaries required for efficient class discrimination can be achieved by fusing a set of weak binary classifiers to optimally represent individual units. The thresholds for these binary classifiers were adaptively estimated through a learning algorithm that maximizes the separability between the sparsely represented signal classes. This approach substantially reduces the computational complexity of extracting, aligning and sorting multiple single unit activity early in the data stream, making it highly suitable for implementation of large scale BMI applications.

Chapter 5 introduces the use of spatiotemporal maximum entropy models for understanding the role of statistical dependence between the output spike trains of cortical neurons in encoding an input stimulus. This dependence may be reminiscent of network state transitions that characterize the dynamics of the stimulus. Pairwise maximum entropy models were suggested to fit instantaneous patterns of activity in small populations that exhibit spatial correlations, but did not account for the temporal correlations that likely represent the effect of spiking history on network state transitions. In this chapter, we introduce an extension to current instantaneous maximum entropy models that accounts for correlations over longer temporal intervals among neurons. We evaluated the performance of our spatiotemporal model in predicting the network states of V1 cortical neurons in

response to drifting grating stimuli, and compared it to that of the instantaneous maximum entropy model, as well as an independent model. We also looked at how stimulus specific are the network representations for either of these models.

Chapter 6 introduces a novel method for inferring connectivity by including higher-order interactions among neuronal constituents and demonstrating this connectivity with hypergraphs. In this chapter, we formulate an information theoretic approach to determine whether cooperation among neurons may constitute a governing mechanism of information processing. Specifically, we show that cooperation among neurons can provide a “message-passing” mechanism that preserves most of the information in the covariates under specific constraints governing their connectivity structure. Using a biologically plausible statistical learning model, we demonstrate the performance of this information theoretic approach in synergistically encoding a motor task using a subset of neurons drawn randomly from a larger population. We demonstrate its superiority in approximating the joint density of the population from limited data compared to the independent and instantaneous maximum entropy models.

Finally, chapter 7 provides concluding remarks and future directions.

# 2

## Overview of Neural Analysis Tools

A typical neuron possesses three distinct parts, the cell body, also referred to as the soma, the axon, and the dendrites [1]. The axon and dendrites are the means by which neurons communicate with each other, and play a part as the neuron's transmitter and receivers, respectively [8]. Neurons are also referred to as excitable cells since they tend to generate an action potential when dendrites receive enough inputs from neighboring neurons and axons, such that the collective activity passes a certain threshold. As a result of neural firing, the generated action potential propagates through the axon to influence the activity of neighboring or distal neurons.

Different techniques developed for recording neural activity can be categorized into noninvasive or invasive methods. While techniques such as electroencephalogram (EEG) [9] and functional magnetic resonance imaging (fMRI) [10] are considered as noninvasive recording methods, patch-clamp recordings of single cells [11] and extracellular recording using multi-electrode arrays (MEAs) [12] are invasive. Although noninvasive techniques due to their harmless nature have been commonly practiced for medical purposes, they basically lack the necessary spatial and temporal resolution required for studying the behavior of neurons in brain functionality [13].

This dissertation mainly focuses on signals collected by invasive recordings techniques using implantable high-density MEAs, where the objective is to study the role of activity of single neurons in a population. These invasive devices have demonstrated to probe high spatial ( $< 50 \mu\text{m}$ ) and temporal ( $< 1$  millisecond) resolutions in the implanted site [14]. The two major kinds of MEAs are wired-based arrays, also referred to as micro-wires [15], and silicon-based arrays [16]. Both micro-wires and silicon-based MEAs have been designed in different shapes and lengths to enable recording from various brain locations of different subjects.

Although MEAs have been designed to record the activity of a subpopulation of neurons, the recorded signal is contaminated by the accumulative activity of millions of surrounding neurons, referred to as the background activity, and noise [17]. We focus explicitly on spikes for a number of reasons: First, their origin – the neurons – is known or can be estimated. Second, spikes fired by a given neuron tend to be highly stereotypical, permitting to model them with probability distributions with relatively few parameters. Third, spike trains –albeit largely variable across multiple repeated trials- carry a lot of information about neuronal tuning characteristics to sensory and motor covariates, thereby permitting adequate modeling of the corresponding neural systems and their dynamics [18]. In contrast, other neural signals, such as Local Field Potentials (LFP) and Electroencephalographic (EEG) signals, are more obscure in terms of the signal sources they originate from, are far more limited in terms of their spatial and temporal resolutions, and their correlation to sensory and/or motor covariates are ill defined. This makes them hard to model and consequently hard to detect or discriminate from pure noise. Therefore, the first step in most neural analysis algorithms is to extract the signal of interest from raw MEA recordings by performing spike detection and sorting. Following, we will explain spike detection and sorting from a general pattern recognition standpoint.

## 2.1. Spike Detection and Sorting

Spike detection and sorting play a crucial role in processing neural signals, largely due to the highly stochastic nature of these signals [17]. The statistical hypothesis testing decides which generative model or hypothesis, among many possible ones, may have generated the observed signals [19]. The degree of complexity of the detection/sorting task can be viewed as directly proportional to the degree of “closeness” of the candidate generative models, i.e., the detection/sorting task gets more complex as the models get closer together in a geometrical sense. In most spike detection/sorting applications, it is often assumed that the signal and noise are independent and therefore uncorrelated.

Spikes are transient stochastic signals that are non-zero only for a subset of samples,  $L$ , within an observation interval. Detecting the transient signal involves knowing the arrival time, the duration and the waveform shape. The most important parameter, however, is the spike arrival time, as it carries all the information about how neurons encode and communicate amongst themselves. Given observations  $\mathbf{x}$ , the detection can be casted as a binary hypothesis testing problem in which the goal is to determine which of the null hypothesis,  $H_0: \mathbf{x} = \mathbf{n}$  or the alternative hypothesis  $H_1: \mathbf{x} = \mathbf{s} + \mathbf{n}$  is in place, where  $\mathbf{s}$  is the transient signal and  $\mathbf{n}$  is noise [20]. Most spike detection algorithms assume the noise and signal are uncorrelated and the binary test can be performed using simple thresholding. The threshold is set as three times the standard deviation of noise [21], and spikes that pass this threshold are cropped and aligned based on a common extremum point, i.e., all spikes are cropped  $L_1$  samples before the local minimum point of the spike and  $L_2$  samples after, where  $L = L_1 + L_2 + 1$ .

Spike sorting refers to the process of classifying the individual spikes extracted after the spike

detection step. The objective is to segregate the activity of each individual neuron to characterize its encoding characteristics. Spike sorting is one of the most extensively studied problems since techniques for extracellular recordings have started to emerge [17]. Since then, the need for more sophisticated spike sorting techniques continues to emerge with the ability to simultaneously sample large populations of neurons. Spike sorting is simply a pattern recognition problem [22]. The approach relies on the sequence of steps illustrated in Figure 2.1. After spikes are detected and extracted, they are aligned relative to a common reference point. This step is important because spikes may be translated and potentially scaled from previous occurrences relative to the detection point especially when the extraction window length  $L$  is empirically chosen by the user.

The feature extraction step finds the most “discriminative” features of the spike waveforms. These features can be viewed as a vector in a  $d$ -dimensional space. The criterion for choosing the features should allow the corresponding pattern vectors from different spike classes to occupy compact and disjoint regions in this  $d$ -dimensional space. The clustering step attempts to separate the extracted features by partitioning the feature space into distinct clusters. This is done by identifying statistical decision boundaries based on perceived similarity among the patterns. This step is difficult because the clusters can have very different shapes and their number is unknown beforehand. The last step is to label the waveforms in each cluster as belonging to a single neuron. These steps constitute the training mode of the pattern classifier. Supervision by the user through the feedback path allows optimizing the preprocessing and feature extraction/selection strategies (e.g. determining the number of clusters). In the test mode, similar preprocessing and feature extraction takes place for the incoming unknown spike. A decision-making step compares the extracted features to the set of stored features and assigns the incoming spike to the closest pattern class using some distance metric.

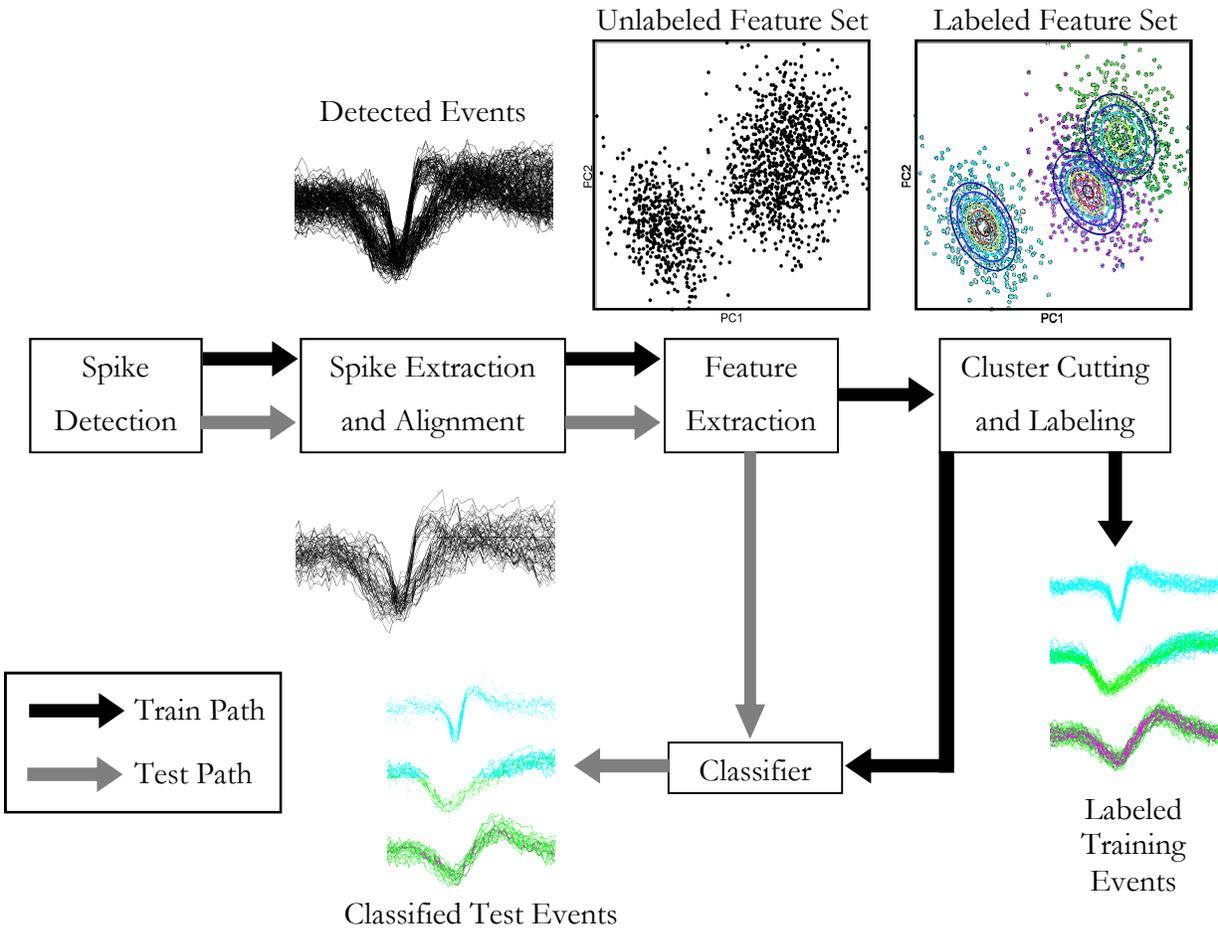


Figure 2.1 Pattern recognition approach to spike sorting. Spike events are extracted after detection and discriminative features are computed and clustered in a 2D or 3D feature space.

The effectiveness of the representation space (feature set) is determined by how well patterns from different classes can be separated, while original details about spike waveform shapes become less important once these patterns are reasonably separable. Therefore, most of the work in spike sorting has focused on the feature extraction and clustering steps. Initial work relied on extracting simple features of the waveform shape such as amplitude and duration [23]. Because these are highly variable features, cluster separability using these methods is often poor unless the SNR is considerably high and the waveforms are stationary. Template matching, on the other hand, relies on

determining the degree of similarity between stored template spike waveforms and incoming spikes [17]. The templates are obtained by averaging many occurrences of labeled spikes in the training mode. The method is computationally demanding because the comparison needs to take place for all possible translations and rotations of the waveforms and is also vulnerable to nonstationarity unless the stored templates are regularly updated.

Principal component analysis (PCA) is a popular method that attempts to alleviate these limitations by representing the spike waveform as a weighted sum of orthonormal eigenvectors. Typically, the PCA feature space uses  $d=2$  or 3 features (depending on how separable the clusters are) to represent each spike. These PCA features are subsequently clustered using algorithms such as Expectation-Maximization clustering [24] or fuzzy c-means clustering [25]. Figure 2.2 illustrates an example of a PCA feature space with two scores representing each spike. In this case, we have 5 clearly isolated clusters that amount to 5 single units in the observed ensemble activity.

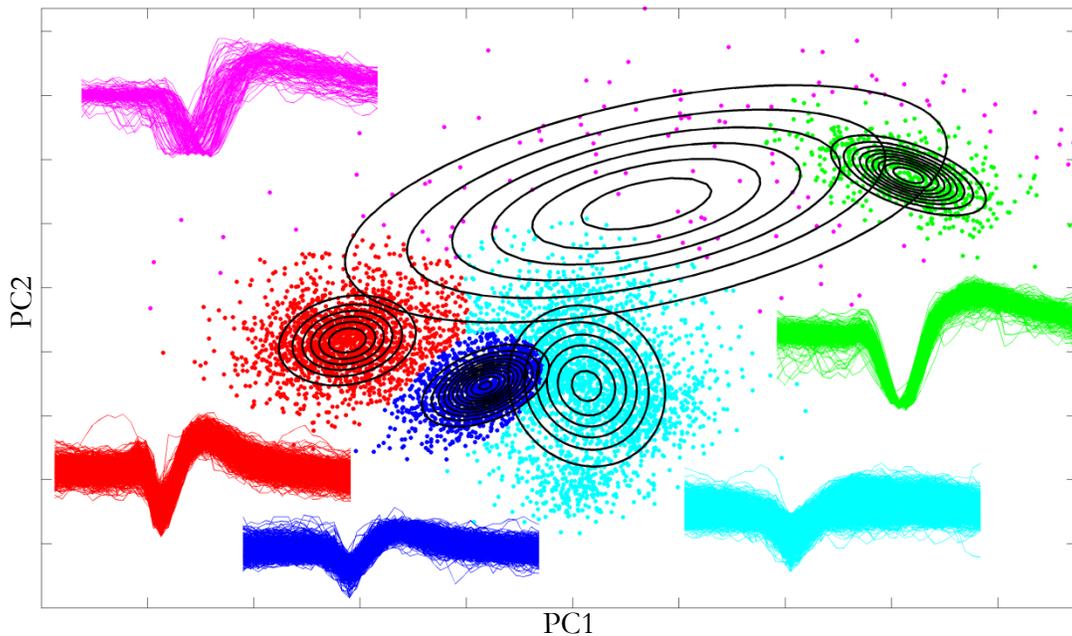


Figure 2.2 PCA for feature extraction and EM for unsupervised cluster cutting of 6469 spike events. Five distinct units can be separated.

## 2.2. Generalized Linear Model

Generalized linear models or GLMs are a family of generative models that have been extensively used to simulate neural encoding models. A GLM models the spike trains of neurons as a binary point process that is governed by the conditional intensity function,  $\lambda(t|F(t))$ , where  $F(t)$  includes all factors that influence the output of a neuron at time  $t$ . In this dissertation, we have used GLMs to simulate a motor encoding model of classical 2D goal-directed arm reach movement in a center-out-reach [18] or a pinball task [26]. The trajectory information was collected from actual repeated arm movements using a computer interface pointing device. The direction of movement was extracted from these trajectories and used as an input to the GLM to modulate the firing pattern of a population of neurons.

We designed the population network in two separate layers, the input layer (also referred to as the non-cooperative layer), and the integration layer (also referred to as the cooperative layer). In the input layer, neurons were only tuned to movement direction, such that the conditional intensity function,  $\lambda_i^{Input}(t|s(t))$ , for neuron  $i$  was given by

$$\lambda_i^{Input}(t|s(t)) = \exp\left(\beta_i + \delta_i \cos\left(\frac{\kappa(t) - \kappa_i}{\omega_i}\right)\right) \quad (2-1)$$

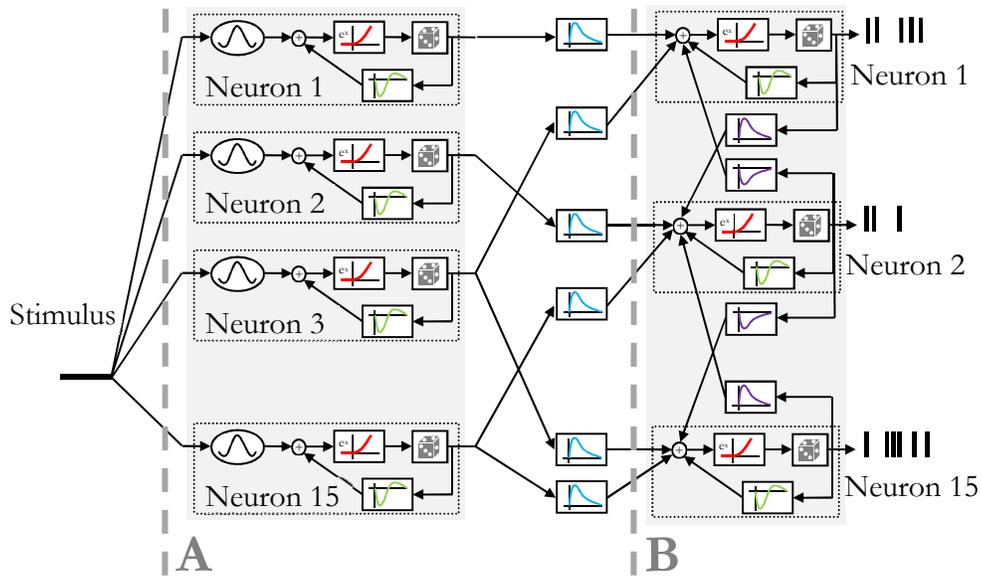
where  $\delta_i$  denotes the modulation depth,  $\kappa(t)$  denotes movement direction,  $\kappa_i$  denotes the neuron's preferred direction (or receptive field in the case of sensory encoding neurons), and  $\omega_i$  denotes the tuning width. This layer was designed such that there were no connections between neurons, and the observed correlations were due to the overlap between their tuning functions. The cosine function was due to the cosine tuning seen in the activity of M1 motor encoding neurons [18].

In the integration layer, neurons were not tuned to any movement parameters. Their encoding resulted from forward connectivity from neurons in the input layer by either excitatory or inhibitory connections. The conditional intensity function of these neurons,  $\lambda_i^{Int}(t|H_i(t))$ , was expressed as

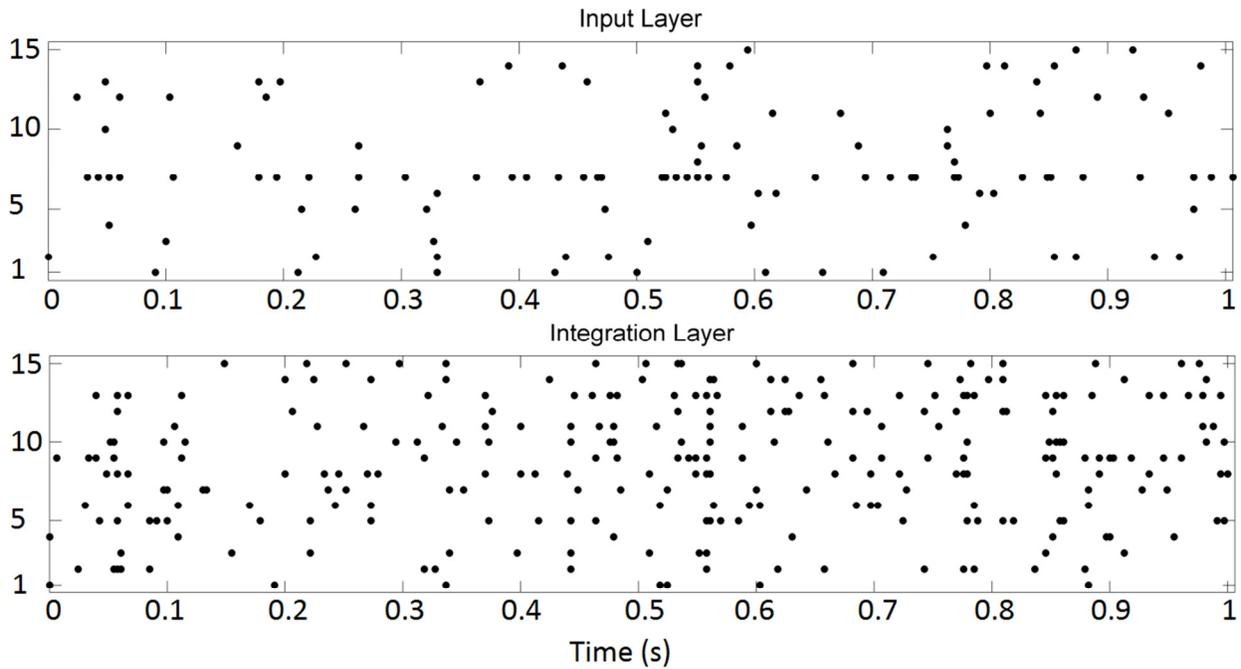
$$\lambda_i^{Int}(t|H_i(t)) = \exp\left(\beta_i + \sum_{j \in \pi(i)} \sum_{m=0}^M \alpha_{ij}^m r_j^{t-m} + \sum_{k \in \tilde{\pi}(i)} \sum_{m=0}^M \gamma_{ij}^m r_k^{t-m}\right) \quad (2-2)$$

where  $H_i(t)$  is the history term for neuron  $i$ ,  $M$  is the length of effective history,  $\pi(i)$  and  $\tilde{\pi}(i)$  are the sets of indices of the pre-synaptic neurons to neuron  $i$  in the input layer and the integration layer, respectively, while  $\alpha_{ij}$  and  $\gamma_{ij}$  represent the forward connections and within integration layer connections, respectively.

Figure 2.3a demonstrates the two-layer GLM used for modeling motor encoding. The blue colored coupling filters are the forward connections between the input and integration layers, expressed as  $\alpha_{ij}^m$ , and the purple colored coupling filters are the connections within the integration layer, expressed as  $\gamma_{ij}^m$  in (2-2). The green filters are known as the intrinsic filters, which are self-inhibitory connections to prevent neurons from bursting activity. They set a refractory period after a spike is fired by a neuron for at least 2 msec in which no spikes are fired. The output of the input and integration layer are in the form of binary spike trains, demonstrated in Figure 2.3b for 15 input layer neurons and 15 integration layer neurons.



(a)



(b)

Figure 2.3 Schematic of the network structure used to simulate the spike trains. (a) The input layer consists of 15 direction-tuned neurons. Integration layer neuron receives forward connections from the input layer. (b) Sample raster plots for one trial during a reach for neurons of the input and integration layers. Every row represents the binary spike train for a neuron, where each dot is a spike event.

# 3

## Neural Activity in the Sparse Representation Domain

Spike trains are the fundamental neural communication mechanism used by cortical neurons to relay, process, and store information in the brain. Decoding the information in spike trains is thus important in neuroscience in order to better understand the complex mechanisms underlying brain functionality. In motor systems, spike trains were demonstrated to carry important information about movement intention and execution [18], and were shown to be useful in the development of Brain Machine Interface (BMI) systems and neuroprosthetics to improve the lifestyle of severely disabled people [27-29]. Cortically-controlled BMI systems depend fundamentally on instantaneous decoding of spike trains from motor cortical neurons recorded during short intervals of around 100-200 milliseconds, often referred to as the movement planning period [29]. The decoding process is typically a cascade of preprocessing steps illustrated in Figure 3.. It features amplification and filtering, followed by spike detection and sorting to segregate single unit activity in the form of binary spike trains. The spike trains are then filtered using a Gaussian kernel to yield a smoothed estimate of the firing rate [30, 31]. These steps have to be performed within the movement preparation period to enable the subject to experience a natural motor behavior.

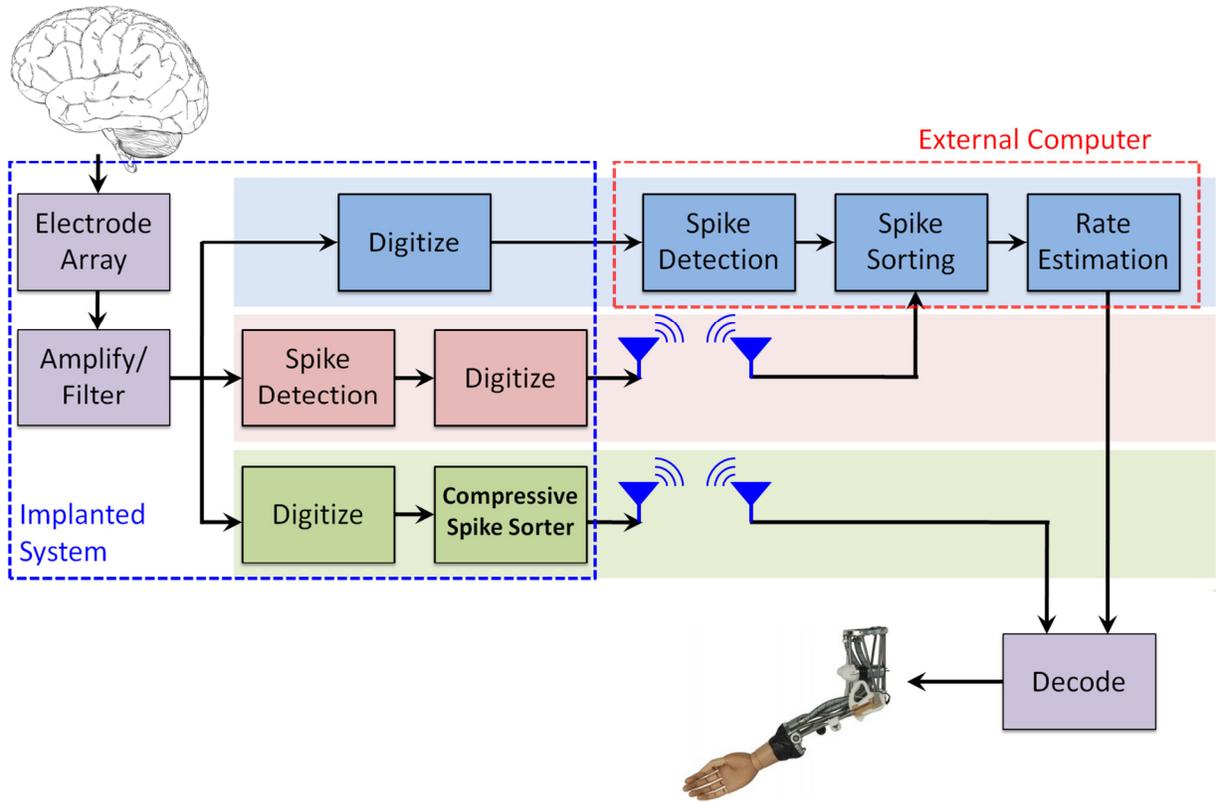


Figure 3.1 Schematic diagram of a typical data flow in a neuro-motor prosthetic application. Ensemble neural recordings are first amplified and filtered prior to telemetry transmission to the outside world. Three data processing paths are considered: (a) Wired systems (top): information is extracted through the cascade of spike detection and sorting followed by rate estimation with a massive computational power, (b) Wireless systems (middle): Telemetry bandwidth is reduced by moving the spike detection block inside the implantable device, (c) Proposed system (bottom): the spike detection, sorting and rate estimation blocks are replaced with one “compressive classifier” block that permits adaptive firing rate estimation in real time to permit instantaneous decoding to take place.

The spike sorting step is arguably the most computationally prohibitive task in Figure 3., requiring two modes of analysis, a training mode and a runtime mode. During training mode, spikes are detected, aligned, and sorted based on certain discriminating features [17], while during runtime,

the features of an observed event are classified as one of the predefined neuronal classes. As a result of the significant amount of computations required to enable this identification/classification process, most of existing systems feature an external computing engine connected through a wire to permit streaming of the high-bandwidth neural data. To eliminate the necessity of the wired connection, it has been suggested to obtain a sparse representation of the recorded data that could be potentially used for denoising and compression [21] as well as spike detection and sorting [32, 33]. In this chapter, we will show that the same sparse representation not only overcomes the severe bandwidth limitations of a wireless implantable system, but also enables adequate estimation of neuronal firing rates without the need to decompress, reconstruct, and sort the spikes on an external device, as illustrated in the bottom of Figure 3.. Following, we will first introduce the model of neuronal firing as a point process and outline the mapping from neural recordings to spike train realizations through sparse representation. Then, the details of the model used to simulate neural activity encoding 2D arm movement is described and preceded by its experimental results.

### 3.1. Spike Trains as Point Processes

In a typical recording experiment, the observations of interest are the times in which events occur from a population of neurons. A spike train is the set of event times that expresses the discharge pattern of an arbitrary neuron  $n$ . This pattern can be modeled as a realization of an underlying point process with a conditional intensity function,  $\lambda_n(t|F)$  also referred to as the firing rate [34]. This intensity function is conditioned on some set,  $F$ , of intrinsic properties of the neuron itself and the neurons connected to it, and the neuron's tuning characteristics to external stimuli [35]. Because many of these properties are hard to measure, the number of events in a given interval,  $N_p$ , is typically random by nature. The sum of  $\lambda_n$  over a finite time interval between times  $T_a$  and  $T_b$

estimates the expected value  $N_p$  within a single trial as [36]

$$E[N_p] = \int_{T_a}^{T_b} \lambda_n(t|F) dt \quad (3-1)$$

Historically, modulation of  $\lambda_n$  from a baseline level  $\beta_n$  has been the fundamental indicator of neuronal involvement in cortical information processing. Estimating  $\lambda_n$  from the set of event times  $\{t_n\}$  is typically achieved by binning the data into time bins of equal width, and counting the number of events occurring within each bin. The resulting spike counts, often referred to as a rate histogram, constitute an instantaneous firing rate estimate  $\hat{\lambda}_n$ . In traditional signal processing, this is equivalent to convolving the spike train with a fixed-width rectangular window. This approach assumes that variations in the rate pattern over the bin width do not carry information that is destroyed if aliasing occurs, for example, when the bin width is not optimally selected to satisfy the Nyquist sampling rate of  $\lambda_n$ .

The binning approach, typically used to obtain the rate histogram, can detect the presence of the type of spike bursts that may exist within the fixed-length bins. However, bursts come in a variety of lengths within a given trial, and can range from very short bursts (3-4 spikes within 2-3 ms [37]) to much longer bursts that can last for more than 2 seconds [38]. This implies that the firing rate of individual neurons is highly non-stationary and that temporal and spectral variations in  $\lambda_n$  are believed to occur over a multitude of time scales that reflect the complex temporal structure of neuronal encoding while subjects carry out similar behavioral tasks [39, 40], or depending on the demands of distinct behavioral tasks [41, 42]. This nonstationarity arises in part because of the dependence of the firing rate on multiple factors such as the degree of tuning (sharp or broad) to behavioral parameters, the behavioral state, the subject's level of attention to the task, level of

fatigue, prior experience with the task, etc. While across-trial averaging of rate histograms helps to reduce this variability, it destroys any information about the dynamics of interaction between neurons that are widely believed to affect the receptive fields of cortical neurons, particularly when plastic changes occur across multiple repeated trials. Typically, a nonparametric kernel smoothing step (e.g. a Parzen window [43]) is needed. The temporal support of the kernel function is known to strongly impact the rate estimator [44]. Moreover, the selection of temporal support is arguably important to determine the type of neural response property sought. For small temporal supports ( $< 2\text{-}3\text{ms}$ ), precise event times can be obtained. As the temporal support approaches the trial length, we obtain the overall average firing rate over that trial. In between these two limits, the temporal support needs to be adaptively selected to capture any nonstationarities in  $\lambda_n$  that may reflect continuously varying degrees of neuronal inhibition and excitation related to the degree of tuning to behavioral parameters.

We ultimately seek to estimate  $\lambda_n$  directly from the recorded raw data. However, two complications arise: First, the events detected are not directly manifested as binary sequence of zeros and ones to permit direct convolution with a kernel to take place, but rather by full Action Potential (AP) waveforms. Second, these events are typically a combination of multiple single unit activity in the form of AP waveforms with generally distinct -but occasionally similar- shapes. This mandates the spike sorting step before the actual firing rate can be estimated.

Let's assume that the actual spike waveforms are uniformly sampled over a period  $T_s$ . Each spike from neuron  $n$  is a vector of length  $N_s$  samples that we will denote by  $\mathbf{g}_n$ . If the event time is taken as the first sample of the spike waveform, but can be generalized to any time index (e.g. that of a detection threshold crossing), then the discrete time series corresponding to the entire activity of neuron  $n$  over a single trial of length  $T$  can be expressed as

$$s_n = \sum_{i \in \{t_n\}} \sum_{k=0}^{N_s-1} g_n[k] \delta[i - k] \quad (3-2)$$

where the time index  $i$  includes all the refractory and rebound effects of the neuron and takes values from the set  $\{t_n\}$ , while  $\delta[\cdot]$  is the Dirac delta function.

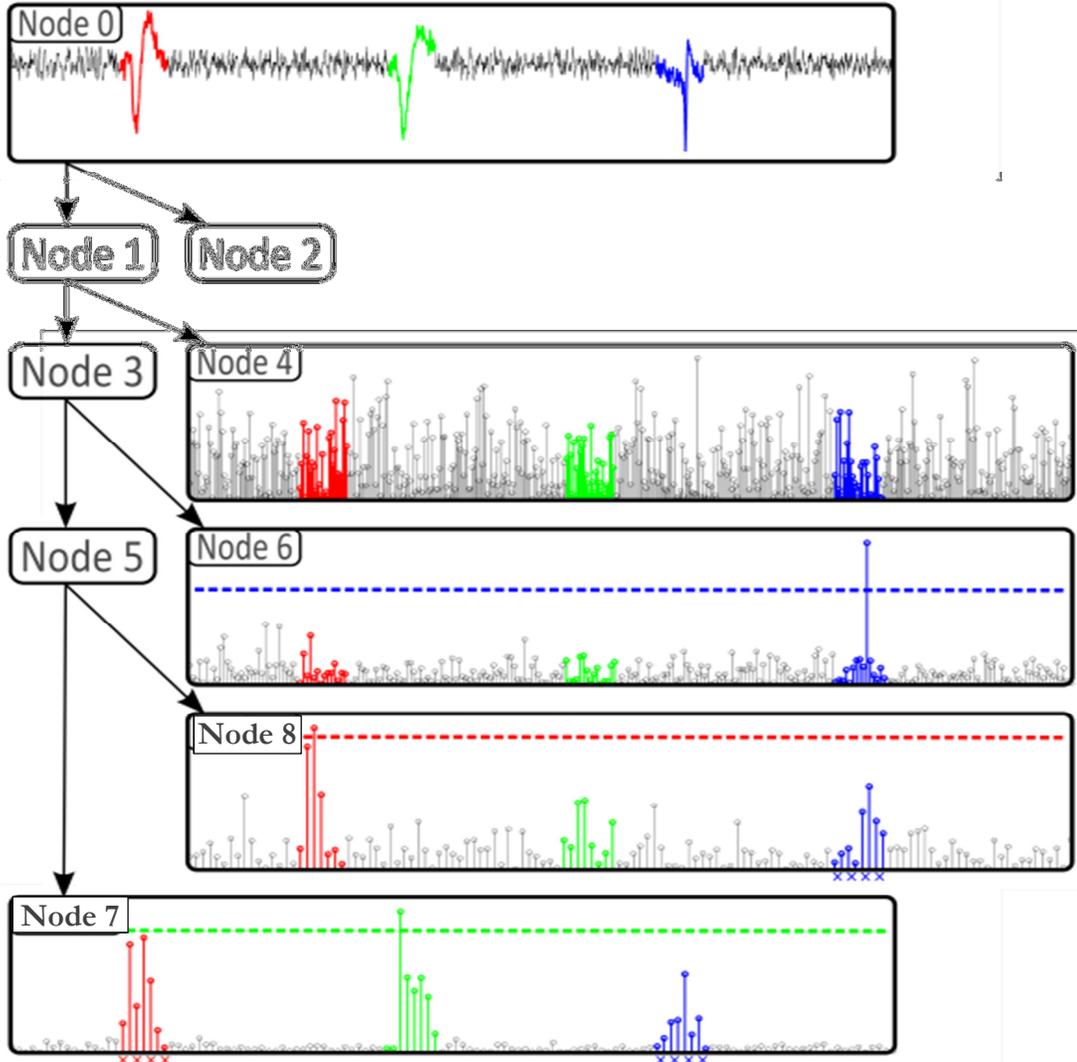
### 3.2. Sparse Representation

For compression purposes, it was shown in [21, 32] that a carefully-chosen sparse transformation operator -such as a wavelet transform- can significantly reduce the number of coefficients representing the spike waveform to some  $N_c \ll N_s$ . This number is determined based on the degree of sparseness  $q$  as  $N_c \approx \varepsilon (q - 2)/2q$  where  $0 < q < 2$  ( $q = 0$  implies no sparseness, while  $q = 2$  implies fully sparse) and  $\varepsilon$  denotes some arbitrarily chosen signal reconstruction error [45]. Mathematically, an observed spike,  $g$ , is represented by the transform coefficients obtained from the inner product  $g^j = \langle g, \omega^j \rangle$ , where  $\omega^j$  is an arbitrary wavelet basis at time scale  $j$ . The wavelet basis should be selected to keep the smallest number of coefficients without significant loss of spike features. In the traditional spike sorting approach, this means performing inverse wavelet transformation followed by spike sorting in the time domain. When multiple units are simultaneously recorded, the spike recordings from the entire population can be expressed as

$$s^j = \sum_{i \in \{t_s\}} \sum_{k=0}^{N_c^j} g^j[k] \delta[i - k] \quad (3-3)$$

where  $N_c^j$  is the number of nonzero transform coefficients at time scale  $j$ , and  $i$  takes values from

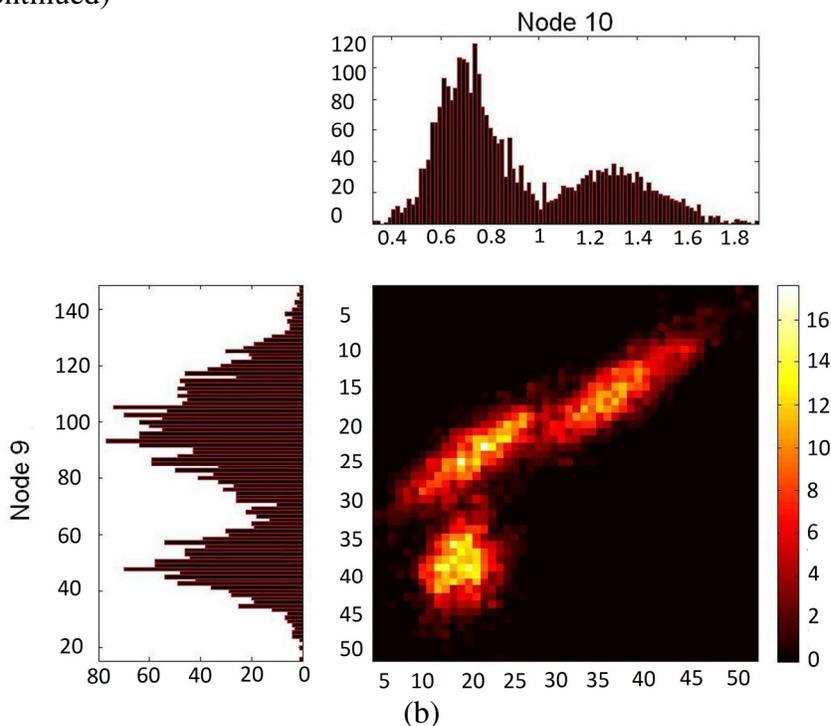
the set of spike times in the whole trial,  $\{t_s\}$ . Note that  $N_c^j \ll N_s$  and the total number of coefficients obtained is  $N_c = \sum_j N_c^j$ .



(a)

Figure 3.2 (a) Sparse representation of sample events from three neurons, red, green and blue in the noisy neural trace for four wavelet decomposition levels indicated by the binary tree. Sensing thresholds are set to allow only one coefficient/event to survive in a given node. For example, node 6, 7 and 8 can be used to mark the blue, green and red events, respectively. (b) 1D and 2D joint distributions of wavelet features for nodes 7 and 8 for the three neurons over many spike occurrences showing 3 distinct clusters. While the 2D projection represents a clear separation between the three neurons, a similar performance can be achieved from the 1D projections by using the two-class discrimination task.

Figure 3.2 (continued)



To minimize the number of the most important coefficients per event, ideally to a single feature, we note that the magnitude of the coefficients  $g^j$  carry information about the degree of correlation of the spike waveforms with the basis  $\omega^j$ . Therefore, this information can be used to single out “the most significant” coefficient via a thresholding process. This process requires defining a sensing threshold for each neuron  $n$  at time scale  $j$ , denoted  $\gamma_n^j$ . This threshold should be selected to preserve the ability to discriminate neuron  $n$ 's events from those belonging to other neurons. Specifically, in every time scale  $j$ , we cast the problem as a binary hypothesis test in which

$$\begin{array}{l} H_1 \\ |g^j[k]| \geq \gamma_n^j \\ H_0 \end{array} \quad \text{for } k = 0, 1, \dots, N_c^j \text{ and } j = 0, 1, \dots, J \quad (3-4)$$

The set of thresholds  $\gamma_n^j$  are selected based on iterative learning approach described in chapter 4.

The outcome of this statistical binary test is a set of time indices per event,  $k^*$ , for which the alternative hypothesis  $H_1$  is in effect. In other words, the sensing threshold in a given time scale should allow only one coefficient to be kept per event, i.e.,  $k^*$  takes only one value. Once this is achieved,  $g^j[k]$  at indices  $k$  where  $H_0$  is in effect are automatically set to zero. Note that this step allows suppressing both noise coefficients as well as those belonging to neurons' other than neuron  $n$ 's. In such case, the thresholded signal can be expressed as

$$\tilde{s}^j = \sum_{k^* \in \{t_n\}} g_n^j[k^*] \delta[i - k^*] \quad (3-5)$$

The outcome of (3-5), after proper normalization, is an estimate,  $\hat{b}_n$ , of the true binary spike train vector  $b_n$ . It can be readily seen that the temporal characteristics of  $\hat{b}_n$  will exactly match that of the binary spike train of neuron  $n$  and consequently preserves all the critical information such as spike counts and inter-spike interval (ISI) statistics allowing rate estimation to be readily implemented [46]. The scaling factor  $g^j[k^*]$  can simply be normalized to obtain a binary sequence exactly matching the spike train of neuron  $n$ . The simple example in Figure 3.2 illustrates this idea. In each wavelet decomposition node, the binary hypothesis test (i.e., the thresholding) is equivalent to a 2-class discrimination task whereby one unit at a time is identified at each decomposition level. The spike class separability (defined below) is compared to that in the time domain and a unit is extracted (i.e. its coefficients removed) from the data set if the unit separability is higher than that of the time domain. This process is repeated until the separability no longer exceeds that of the time domain, or the size of the remaining events is smaller than a minimum cluster size (typically five events), or the maximum number of levels has been reached (typically 4-5 levels).

A fundamental property of the DWT sparse representation suggests that as  $j$  increases,  $\tilde{s}^j$

becomes more representative of the intensity function rather than the temporal details of the spikes themselves, which were eventually captured in finer time scales. This is because the coefficients that survive the sensing threshold will spread their energy across multiple adjacent time indices, thereby performing the same role as the kernel smoothing approach, but at a much less computational overhead as will be shown later. Mathematically, extending the DWT of the vector  $\hat{\mathbf{b}}_n$  to higher level requires convolving it with a wavelet basis kernel with increasing support. This support, denoted  $t_L$  at level  $L$ , is related to the sampling period  $T_s$  as  $t_L = T_s n_w 2^{L-2}$ , where  $n_w$  is the wavelet filter support. For the symmlet4 basis ( $n_w = 8$ ), this temporal support is equivalent to  $\sim 1.2$  ms at level 4 at a 25 kHz sampling rate, which roughly corresponds to one full event duration. Extending the decomposition to level 5 will include refractory and rebound effects of neurons typically observed in the cerebral cortex [40]. Therefore, temporal characteristics of the firing rate will be best characterized starting at level 6 and beyond where the basis support becomes long enough to include two or more consecutive spike events.

### 3.3. Computational Complexity of Spike Sorting in the Sparse Domain

Herein, we compare the cost of estimating the firing rate through the standard time domain spike sorting/kernel smoothing approach and the compressed sensing approach. Both involve calculating the computational cost in two different modes of operation, the “training” mode and the “runtime” mode. In the training mode, features are extracted and the population size is estimated using cluster cutting in the feature space. This should ideally correspond to the number of distinct spike templates in the data. In the runtime mode, the observed waveforms are assigned to any of the existing classes, typically using a Bayesian classifier with equal priors

$$n = \max_n P(g|C_n) \quad (3-6)$$

$$P(g|C_n) = \mathcal{N}(g - \mu_n, \Sigma_n)$$

where  $\mu_n$  and  $\Sigma_n$  are the  $N_s \times 1$  mean vector and  $N_s \times N_s$  temporal covariance matrix for each neuron class  $C_n$ , for  $n = 1, \dots, N$ . The overall computations for the Bayesian classifier are in the order of  $O(N \times N_s^2)$ .

A standard PCA-based spike sorting followed by a Gaussian Kernel rate estimator was used as the benchmark for evaluating the computational cost of the traditional path that appears in the top of Figure 3.1. First, spikes are aligned by searching for a local extreme followed by cropping the waveform symmetrically around that location, which requires computations in the order of  $O(2N_s \times N_p)$ . Finding the eigenvalues and eigenvectors, for example, using a cyclic Jacobi method [47], requires  $O(N_s^3 + N_s^2 \times N_p)$  computations. For projection,  $O(2N_s \times N_p)$  operations are performed to reduce the dimensionality of spike waveforms to a 2-dimensional feature space.

A cluster-cutting algorithm, such as expectation-maximization (EM), is performed on the obtained 2-D feature space. Optimizing EM clustering requires  $O(Nd^2 \times N_p^2)$  computations, where  $N$  here indicates the number of Gaussian models and  $d$  is the dimension of data (here  $d = 2$ ). To detect various spike prototypes, the EM clustering is implemented for different  $N$ 's, and the best fit is selected. The overall computations required for EM clustering for a maximum number of  $N$  units is in the order of  $\sum_{k=1}^N O(4kN_p^2) = O(2N(N+1)N_p^2)$ . Consequently, the overall computations required for training the PCA-based spike sorting is  $O(4N_s \times N_p + N_s^3 + N_s^2 \times N_p + 2N(N+1)N_p^2)$ . In the runtime mode, detected spikes are aligned and projected, and then

classified to one of the predefined units using the Bayesian classifier, requiring computations in the order of  $O(4N_s + 4N)$ .

In contrast, a five-level wavelet decomposition requires operations in the order of  $O(23N_s)$  if classical convolution is used. However, this number can be significantly reduced by using the approach we reported in [48]. Local averaging, typically used to remedy the shift variance property of the DWT, with a node-dependent filter requires computations in the order of  $O(8N_s)$ , since this filter is only applied to nodes 4, 6, 8, 9, and 10 in which spike features are mostly captured. At each node, one unit is discriminated at a time using a 2-class cluster cutting (binary classification). The required computations for this are in the order of  $O(10N_p^2)$ . Consequently, the overall computations required for the training mode is in the order of  $O(31N_sN_p + 10N_p^2)$ . In the runtime mode, every detected event is decomposed, filtered, and classified using a 1-D Bayesian classifier with computations in the order of  $O(31N_s + N)$ . For rate estimation, three methods were considered: the rectangular kernel (rate histogram), the Gaussian kernel and the extended DWT (EDWT) we propose. In EDWT, the firing rate is directly obtained by normalizing the thresholded vectors and extending the decomposition to lower levels (higher frequency resolution). This requires  $O(45N_s n_\omega \sum_{l=5}^{\infty} 2^{-l}) = O(22.5N_s)$ . In the kernel based methods, a kernel function is convolved with the spike train and the rate is estimated by sampling the result. Assuming 45 msec bin width, and 2 msec refractory period, the number of computation required is in the order of  $O(22.5n_W)$ . A Gaussian kernel width of  $n_W = 100$  is typically used to limit the amount of computations.

### 3.4. Spike Class Generation and Separability

Spike waveforms were detected and extracted from spontaneous activity recorded in the primary motor cortex of an anesthetized rat using a 16-channel microelectrode array. All procedures were

approved by the Institutional Animal Care and Use Committee at Michigan State University following NIH guidelines. Details of the experimental procedures to obtain these recordings are described elsewhere [21]. These spikes were manually aligned and sorted using a custom spike sorting algorithm [32]. Out of 24 units recorded, the actual action potential waveforms are shown in Figure 3.3a and 3.3b for five representative units recorded on one electrode. Figure 3.3c shows a scatter plot of the first two principal components of the five representative spike classes in Figure 3.3a. Consider for example unit 4 that appears quite well isolated in the time domain feature space, It is clear that the other classes are poorly isolated. Results of manual, extensive, offline sorting using hierarchical clustering of all the features in the data are displayed in Figure 3.3d. In Figure 3.3e, the clustering result using automated, online PCA/EM cluster-cutting with two principal features is illustrated. Examination of these figures reveals that the lack of separability in the feature space, particularly for units 1, 2, 3 and 5, results in significant differences between the manual, extensive, offline sorting result and the automated online PCA/EM result.

The separability of spike classes in feature space was calculated to determine the sensing thresholds for each neuron at any given time scale  $j$ . Specifically, we used the measure,  $\Gamma(\mathbf{C}) = S_B/S_W$ , where  $S_B$  is the between cluster separability and  $S_W$  is the within cluster separability and  $\mathbf{C} = \{C_n | n = 1, \dots, N\}$  is the set of clusters. The between-cluster separability is defined as [49]

$$S_B = \sum_{n=1}^N \frac{1}{|C_n| \sum_{m \neq n} C_m} \sum_{x \in C_n} \sum_{\substack{y \in C_m \\ m \neq n}} \|x - y\| \quad (3-7)$$

where  $|C_n|$  equals the number of spikes belonging to cluster  $C_n$ ,  $x$  and  $y$  are elements from the set of all spike waveforms and  $\|\cdot\|$  represents the Euclidean distance or L2-norm between two

elements.  $S_B$  provides a factor proportional to the overall separation between clusters. For improved separability, a large  $S_B$  is desired. On the other hand, the within-cluster separability is defined as

$$S_W = \sum_{n=1}^N \frac{1}{|C_n|(|C_n| - 1)} \sum_{x \in C_n} \sum_{y \in C_n} \|x - y\| \quad (3-8)$$

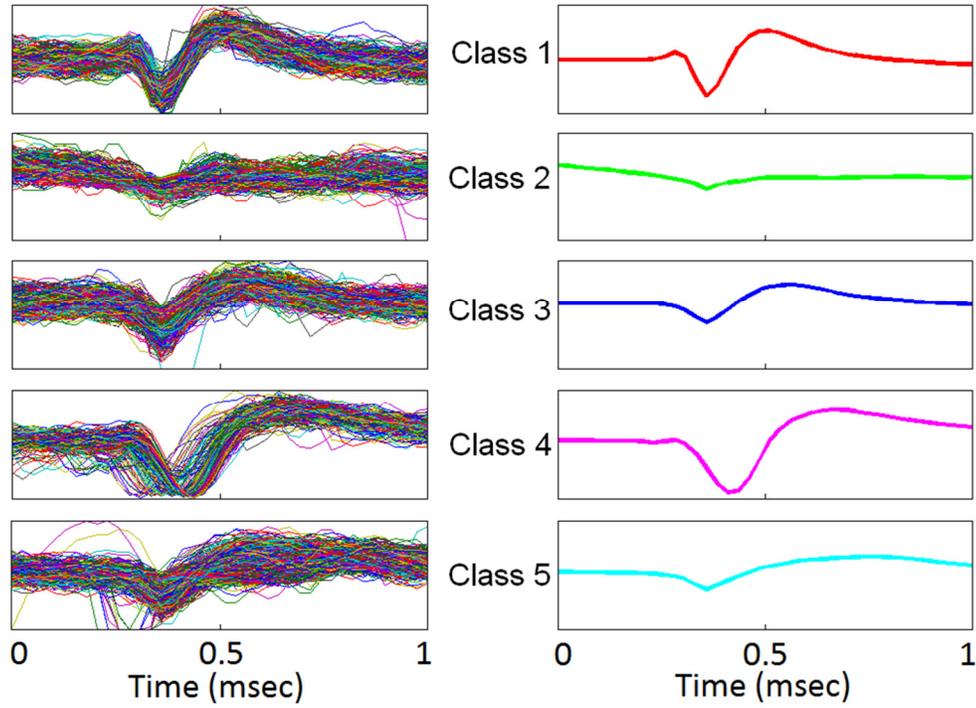
and is proportional to the overall spread within each of the individual clusters. For improved separability, a small  $S_W$  is desired. Therefore, a large  $\Gamma(\mathbf{C})$  indicates a greater overall separability.

We computed a separability ratio (SR) as the ratio between  $\Gamma(\mathbf{C})$  in every node to that in the time domain. Therefore, an SR ratio of 1 indicates equal degree of separability in both domains, while ratios larger than 1 indicate superior separability in the sparse representation domain. This later case implies that at least one unit can be separated in that node's feature space better than the time domain's feature space. This detected unit is subsequently removed from the data and the decomposition process continues until all possible units are detected, or all nodes have been examined on any given electrode. On the other hand, if the same unit can be discriminated in more than one node, the "best node" for discrimination of this unit is the node that provides the largest SR.

Figure 3.4 illustrates that each spike class of Figure 3.3 is separable in at least one node of the sparse representation, in which a two-class situation is considered and one single cluster is isolated in a given node where all other spike classes are lumped together. The different degrees of separability across nodes permit isolating one class at a time, owing to the compactness property of the transform in nodes that are best representative of each class. For example, class 1 appears poorly isolated from class 5 in the time-domain feature space, yet it is well separated from all the other classes in node 6. Since the sensing threshold is chosen to discriminate between spike events and not to minimize the MSE of the reconstructed spike, this selection rule results in thresholds that are

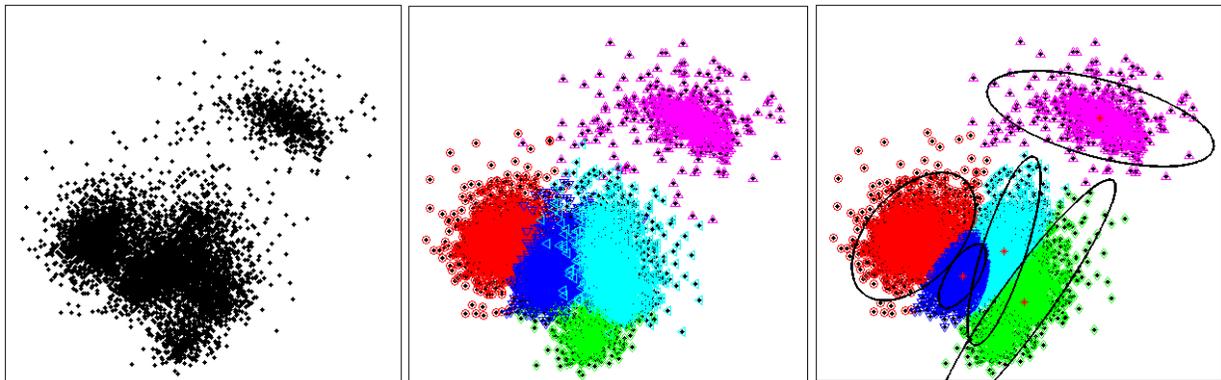
typically higher than those obtained from the universal thresholding rule for denoising and near-optimal signal reconstruction [50].

It can be seen from Figure 3.4a that in most nodes, the SR ratio is larger than 1 (except for nodes 2 and 10). For the 24 units recorded in this data set, the performance of the compressed sensing strategy was  $92.88 \pm 6.33\%$  compared to  $93.49 \pm 6.36\%$  for the PCA-EM. Performance of the sensing threshold selection process was quantified as a function of the number of coefficients retained in Figure 3.4b. As we increase the sensing threshold, the number of retained coefficients logically decreases thereby improving compression. However, the most interesting result is the improved separability by more than 70% compared to time domain separability at roughly 97% compression. This implies that discarding some of the coefficients that may be needed for optimal spike reconstruction and sorting in the time domain in a classical sense does improve the ability to discriminate between spike classes based on their magnitude only. Maximum separability is reached when we retain a few coefficients per event, after which some classes are entirely lost and the performance deteriorates.



(a)

(b)



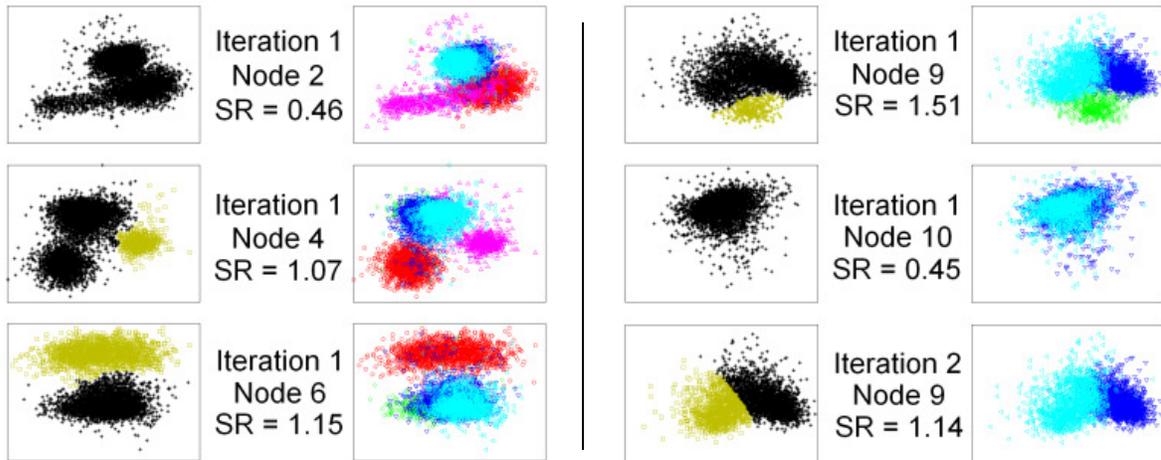
(c)

(d)

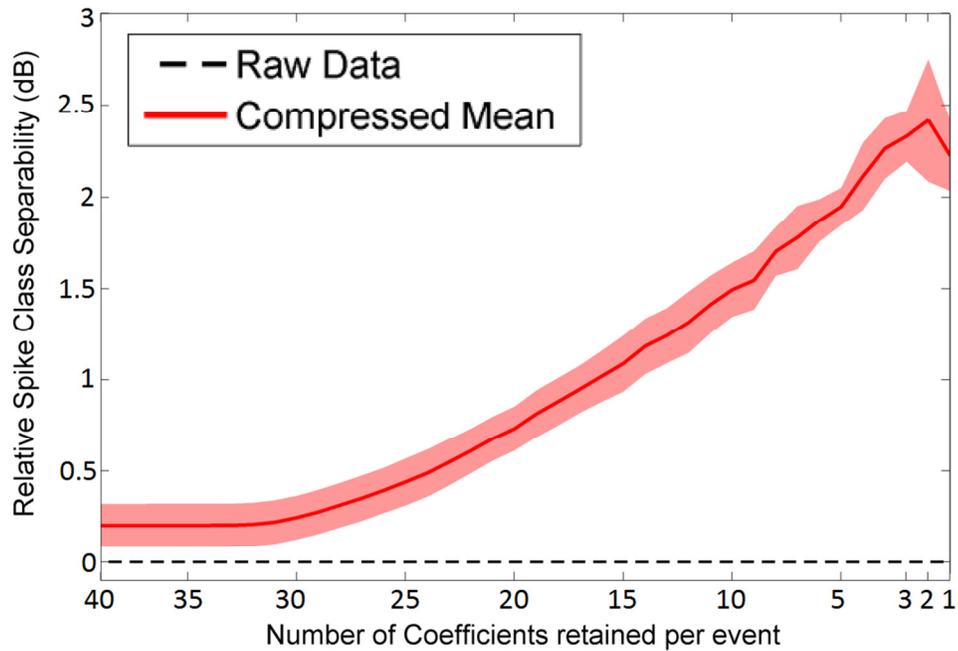
(e)

Figure 3.3 Five units obtained from recordings in an anesthetized preparation. (a) Events from each recorded unit, aligned and superimposed on top of each other for comparison. (b) Corresponding spike templates obtained by averaging all events from each unit on the left panel. (c) PCA 2D feature space. Dimensions represent the projection of spike events onto the two largest principal components. (d) Clustering result of manual, extensive, offline sorting using hierarchical clustering

using all features in the data. (e) Clustering result using the two largest principal components and EM cluster-cutting based on Gaussian Mixture Models (GMM). This is an example of a suboptimal online sorting method with relatively unlimited computational power.



(a)



(b)

Figure 3.4 (a) Unit isolation quality of the data in Fig.3. Each cell in the left side shows the separation (displayed as a 2-D feature space for illustration only) obtained using the compressed

sensing method with the binary hypothesis test in each node. The highest magnitude coefficients that survive the sensing threshold in a given node are considered irregular samples of the underlying unit's firing rate and are marked with the "Gold" symbols in the left panel. The feature space of the sorted spikes using the manual, extensive, offline spike sorting is re-displayed in the right side (illustrated with the same color code as Figure 3) for comparison. If a gold cluster from the left panel matches a single colored cluster from the right panel in any given row, this implies that the corresponding unit is well isolated in this node using the single coefficient/event magnitude alone. The unit is then removed from the coefficient data before subsequent DWT calculation is performed in the next time scale. Using this approach, three out of five units (pink, red, and green) in the original data were isolated during the first iteration in nodes 4, 6, and 9, respectively, leaving out two units to be isolated with one additional iteration on node 9's remaining coefficients. In the first iteration, node 2 shows weak separation ( $SR = 0.45$ ) between units. Unit class 4 has larger separability in node 4 ( $SR=1.07$ ). Units 1 and 2 are separated in nodes 6 and 9 ( $SR=1.15$  and  $1.51$ , respectively). Units 3 and 5 are separated in node 9 afterwards ( $SR = 1.14$ ). (b) Quantitative analysis of spike class separability vs. compression rate (i.e. thresholding) for 24 units recorded in the primary motor cortex of anesthetized rat. A  $\sim 70\%$  improvement can be observed at around 97% compression ratio compared to time domain separability (Raw data).

### **3.5. Decoding Performance**

Because our purpose was to demonstrate the ability to decode movement trajectory directly from neural data using the compressed signal representation, and given that the nature of cortical encoding of movement remains a subject of current debate in the neuroscience community [18, 28, 29, 51], investigation of the methods developed in this paper required generation of neural data with known spike train encoding properties. This section describes in details the methods we used to

model and analyze the data to demonstrate the validity of the approach. Since instantaneous decoding of spike trains is the ultimate goal in this application, we used the decoding performance as a measure of success of this method. To simulate spike trains from motor cortex neurons during movement planning and execution, we used a probabilistic population encoding model of a natural, non-goal directed, 2D arm movement trajectory. The arm movement data were experimentally collected to ensure realistic kinematics. The discrete time representation of the conditional intensity governing each neuron firing rate was modeled as a variant of the cosine tuning model of the neuron's preferred direction  $\theta_n$  (ranging from 0 to  $2\pi$ ) [18]

$$\lambda_n(t_k|x_n) = \exp\left(\beta_n + \delta_n \cos\left(\frac{\theta(t_k) - \theta_n}{\omega_n}\right)\right) \quad (3-9)$$

where  $\beta_n$  denotes the background firing rate,  $\theta(t_k)$  denotes the actual movement direction,  $x_n = [\theta_n, \delta_n, \omega_n]$  is a parameter vector governing the tuning characteristics of neuron  $n$ , where it was assumed that the tuning depth  $\delta_n$  was constant ( $\delta_n$  and  $\beta_n$  were fixed for all neurons and equal to 1 and  $\log(5)$ , respectively), the preferred direction  $\theta_n$  was uniformly distributed, while the tuning width  $\omega_n$  was varied across experiments. Using this model, event times were obtained using an inhomogeneous Poisson process with 2 ms refractory period.

The tuning term in (3-9) incorporates a neuron-dependent tuning width  $\omega_n$ , an important parameter that affects the bin width choice for rate estimation prior to decoding. Variability in this term ( $\omega_n$  ranged from 0.25 to 4 in each experiment) resulted in firing rates that are more stochastic in nature and served to closely approximate the characteristics of cortical neurons' firing patterns [40]. We used the mean squared error between the rate functions obtained from the simulated trajectory data and the estimated rates as

$$MSE = \frac{1}{N} \sum_{n=1}^N (\lambda[n] - \hat{\lambda}_j[n])^2 \quad (3-10)$$

While equation (3-10) provides a simple and obvious measure of performance, it should be noted that in practice the true rate function is unknown. Information theoretic measures are useful in such cases since they assess higher order statistical correlation between the estimators and measurable quantities such as the observed movement and can be useful to determine the time scale that best characterize the information in the instantaneous firing rate. We used a node-dependent mutual information metric between the encoded movement parameter and the rate estimator [52] defined as

$$I_j = \sum_{\theta, \hat{\lambda}_j} P(\theta, \hat{\lambda}_j) \log \frac{P(\theta, \hat{\lambda}_j)}{P(\theta)P(\hat{\lambda}_j)} \quad (3-11)$$

This metric is particularly useful when the instantaneous rate function is not Gaussian distributed.

A sample trajectory, rate functions from neurons with distinct tuning characteristics and their spike train realizations are shown in Figure 3.5. It can be clearly seen that the tuning width has a direct influence on the spike train statistics, particularly the ISI. A broadly tuned neuron exhibits more regular ISI distribution, while a sharply tuned neuron exhibits a more irregular pattern of ISI. Figure 3.5b illustrates the tuning characteristics of a subpopulation of the entire population over a limited range (for clarity) to demonstrate the heterogeneous characteristics of the model we employed. A 9-second raster plot in Figure 3.5c illustrates the stochastic patterns obtained for the trajectory illustrated later.

In Figure 3.6a, a 3 second segment of the movement's angular direction over time is illustrated superimposed on the neuronal tuning range of three representative units with distinct tuning widths.

The resulting firing rates and their estimators using the rate histogram, Gaussian kernel, and extended DWT methods are illustrated for the three units, showing various degrees of estimation quality. As expected, the rate histogram estimate is noisy, while the Gaussian and EDWT methods perform better. In Figure 3.6b, the relation between the wavelet kernel size and the MSE is quantified. As expected, decomposition levels with shorter kernel width (i.e., fine time scales) tend to provide the lowest MSE for neurons that are sharply tuned. In contrast, a global minimum in the MSE is observed for broadly tuned neurons at coarser time scales, suggesting that these decomposition levels are better suited for capturing the time varying-characteristics of the firing rates. Interestingly, the MSE for the EDWT method attains a lower level than both the rectangular and Gaussian kernel methods at the optimal time scale, clearly demonstrating the superiority of our approach. The relation between the tuning width and the kernel size for the entire population is illustrated in Figure 3.6c. As the tuning broadens, larger kernel sizes (i.e. deeper decomposition levels) are required to attain a minimum MSE and thus better performance.

The mutual information between the actual movement trajectory and the rate estimators are shown in Figure 3.7. There is a steady increase in the mutual information versus kernel support until a maximum is reached at the optimal decomposition level that agrees with the minimum MSE performance. This maximum coincides with a rate estimator spectral bandwidth matching that of the underlying movement parameter. Rate estimators beyond the optimal time scale do not carry any additional information about the movement trajectory.

A sample trajectory and the decoded trajectory are shown in Figure 3.8 for four different cases: First, when no spike sorting is required. This is the ideal case in which every electrode records exactly the activity of one unit, but is hard to encounter in practice. Second, when two or more units are recorded on a single electrode but no spike sorting is performed prior to rate estimation. Third, when spike sorting is performed for the latter case using the PCA/EM/Gaussian kernel algorithm.

And fourth, when combined spike sorting and rate estimation are performed using the compressed sensing method. We used a linear filter for decoding in all cases [53]. It is clear that the our method has a decoding error variance that is comparable to the PCA/EM/Gaussian kernel algorithm, suggesting that the performance is as good as, if not superior, to the standard method.

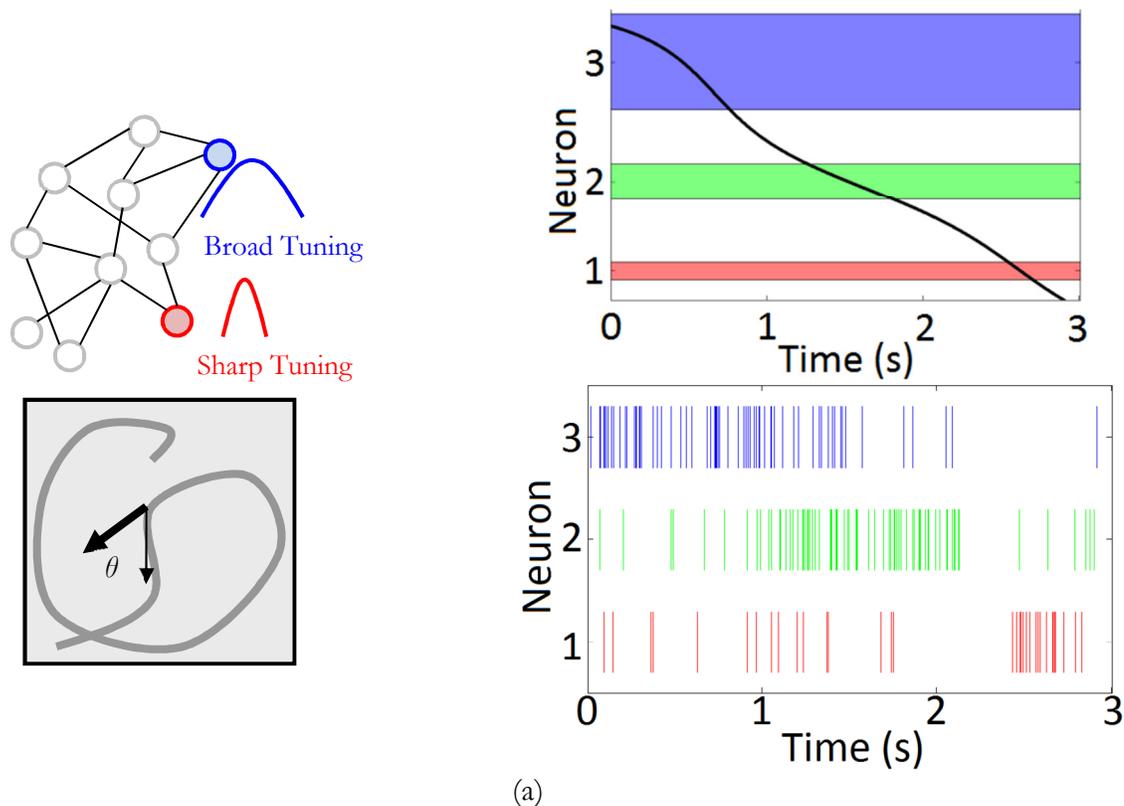
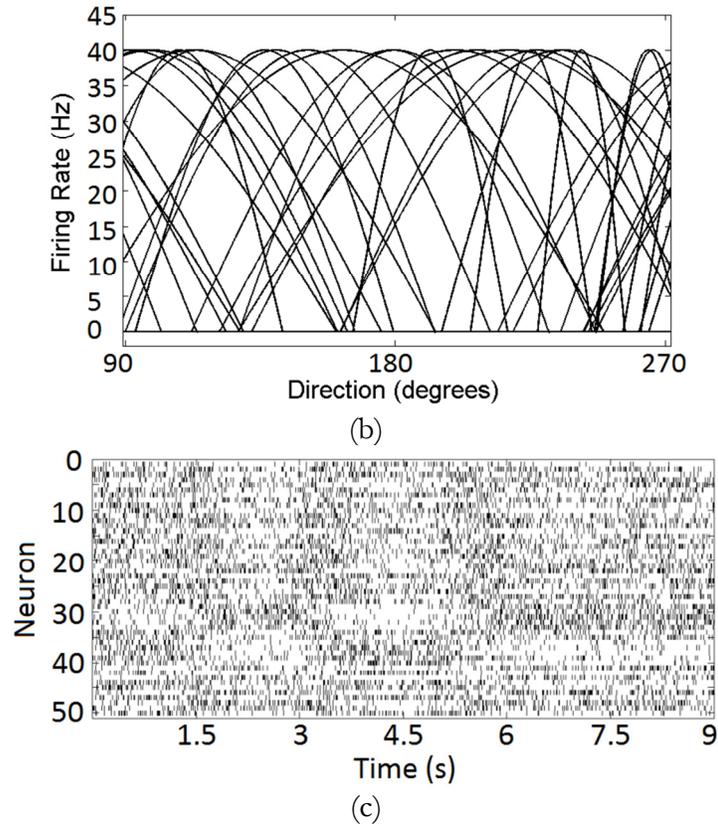
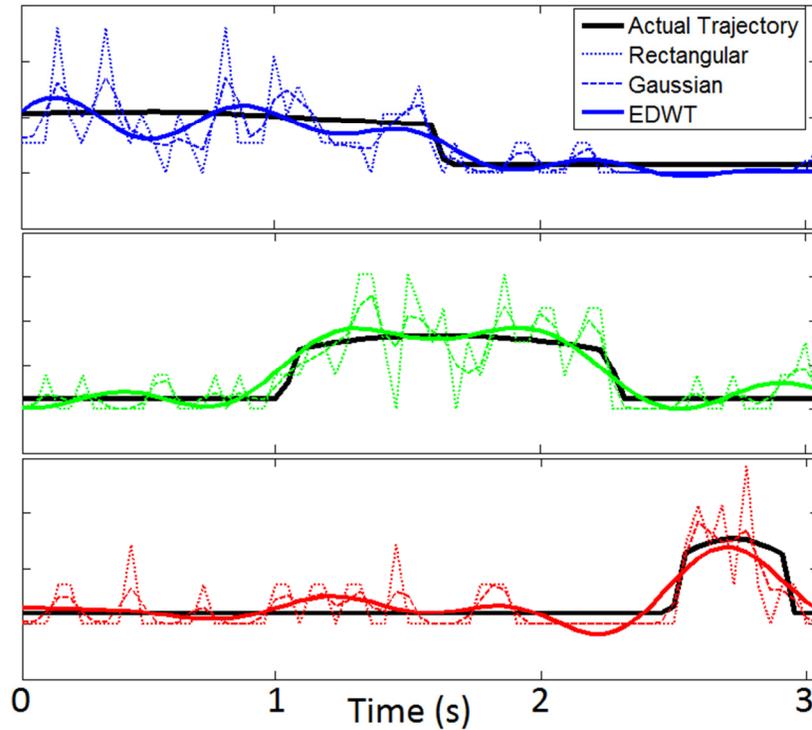


Figure 3.5 (a) Schematic of encoding 2D, non goal-directed arm movement: the sample network of neurons is randomly connected with positive (excitatory), and negative (inhibitory) connections. Right panel demonstrates a symbolic movement trajectory to indicate the movement parameter encoded in the neural population model. Sample firing rates and corresponding spike trains are shown to illustrate the distinct firing patterns that would be obtained with broad and sharp tuning characteristics. (b) Sample tuning characteristics (over a partial range) of a subset of the 50 neurons modeled with randomly chosen directions and widths. (c) Sample 9-sec raster plot of spike trains obtained from the population model.

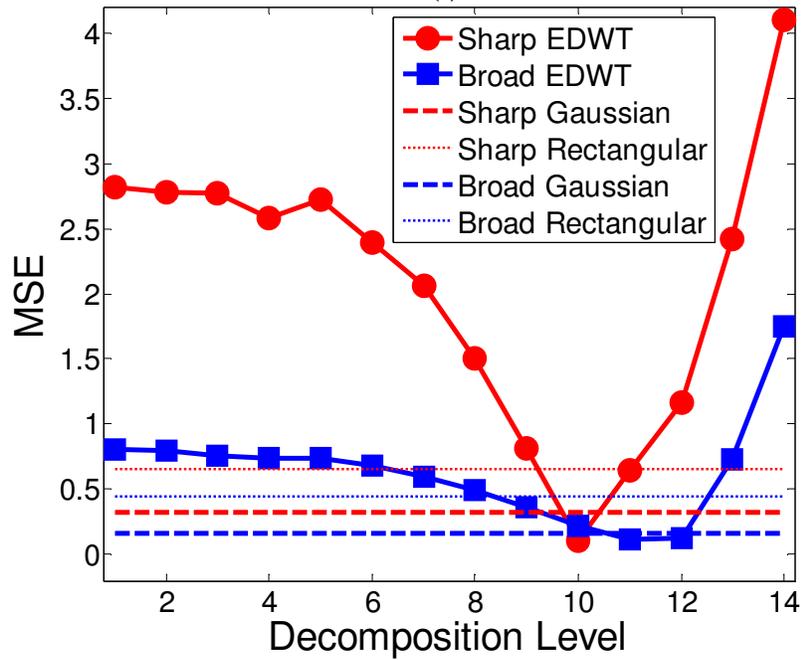
Figure 3.5 (continued)



An important aspect to validate and confirm the superiority of our approach is to compare the computational complexity of the standard PCA/EM/Gaussian kernel rate estimator to the compressed sensing method for different event lengths ( $N_s$ ) and different number of events ( $N_p$ ) per neuron. The results illustrated in Figure 3.9 show that our method requires significantly less computations for training. This is mainly attributed to the complexity in computing the eigenvectors of the spike data every time a new unit is recorded. In contrast, wavelets are universal approximators to a wide variety of transient signals and therefore do not need to be updated with the occurrence of events from new units. In the runtime mode, the computational cost for our method becomes higher when the number of samples/event exceeds 128 samples. At a nominal sampling rate of 40 kHz (lower rates are typically used), this corresponds to a 3.2 msec interval, which is much larger than the typical action potential duration (estimated to be between 1.2-1.5 msec).



(a)

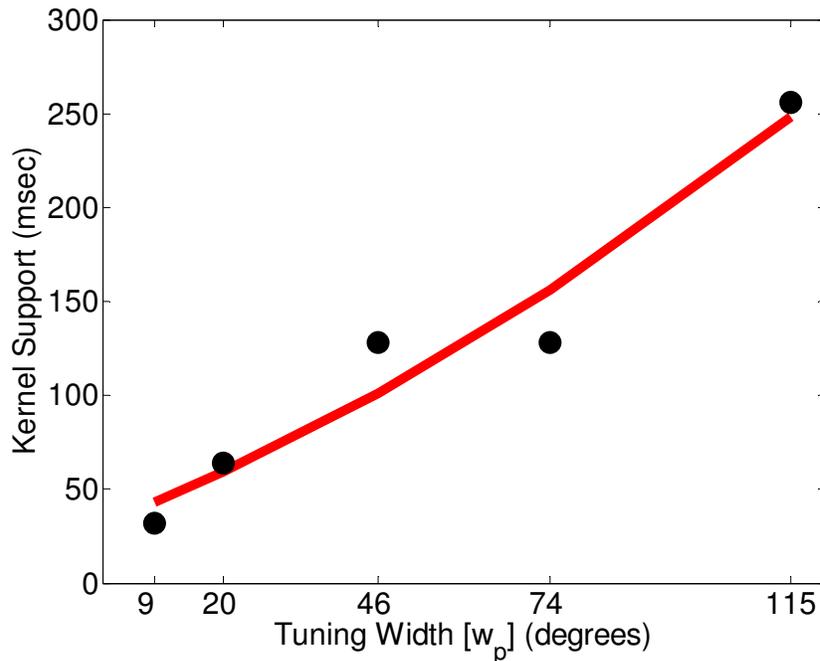


(b)

Figure 3.6 (a) Top-left: 400 msec segment of angular direction from a movement trajectory superimposed on tuning “bands” of five representative units. Top right, middle, and bottom panels: Firing rates obtained from the point process model for five units and their extended DWT (EDWT),

Gaussian, and rectangular kernel estimators. As expected, the rectangular kernel estimator is the noisiest, while the Gaussian and EDWT estimators are closest to the true rates. (b) Mean square error between the actual (solid black line) and the estimated firing rate for each neuron with the three methods. Each pair of dotted and dashed lines is the MSE for rectangular and Gaussian kernel methods, respectively, for the five units in (a). These remain flat as they do not depend on the DWT kernel window length. For the sharply tuned neurons, on average, ten levels of decomposition result in a minimum MSE that is lower than the MSE for rectangular and Gaussian kernel methods. For broadly tuned neurons, 12 levels of decomposition result in optimal performance. (c) Tuning width versus optimal kernel size. As the tuning broadens, larger kernel windows (i.e. coarser time scales) are needed to obtain optimal rate estimators.

Figure 3.6 (continued)



(c)

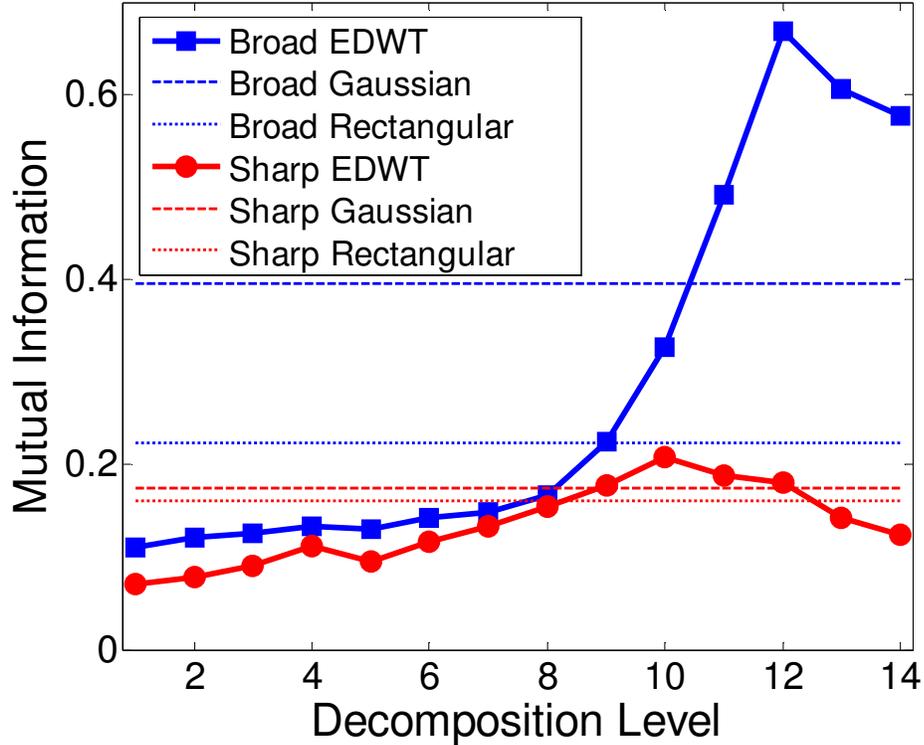


Figure 3.7 Average mutual information (in bits) between movement direction,  $\theta$ , and rate estimators averaged across the two subgroups of neurons in the entire population as a function of decomposition level (i.e. kernel size). Solid lines indicate the performance of the EDWT method (dark for the broad tuning group and gray for the sharp tuning group). The two dashed lines represent the Gaussian kernel method (broad tuning and sharp tuning groups), while the two dotted lines represent the rectangular kernel method in a similar way. As expected, sharply tuned neurons require smaller kernel size to estimate their firing rates. Overall, the EDWT method achieves higher mutual information than either the fixed width Gaussian or rectangular kernels for broadly tuned neurons, while slightly less for sharply tuned neurons owing to the relatively more limited response time these neurons have.

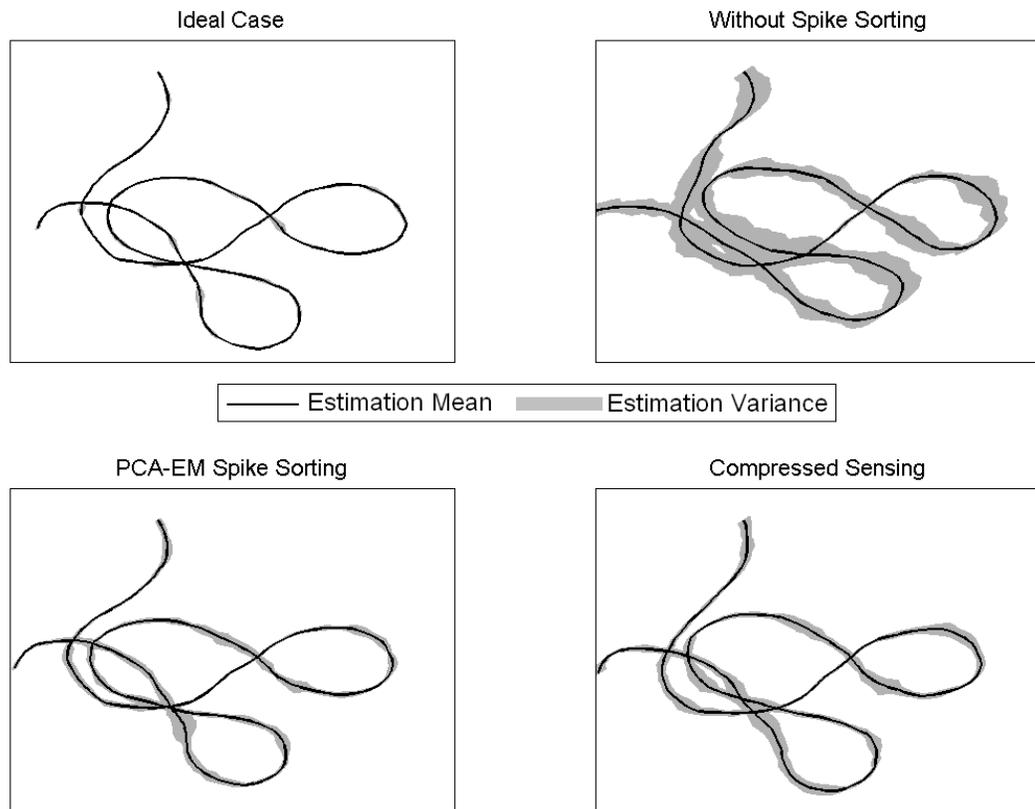
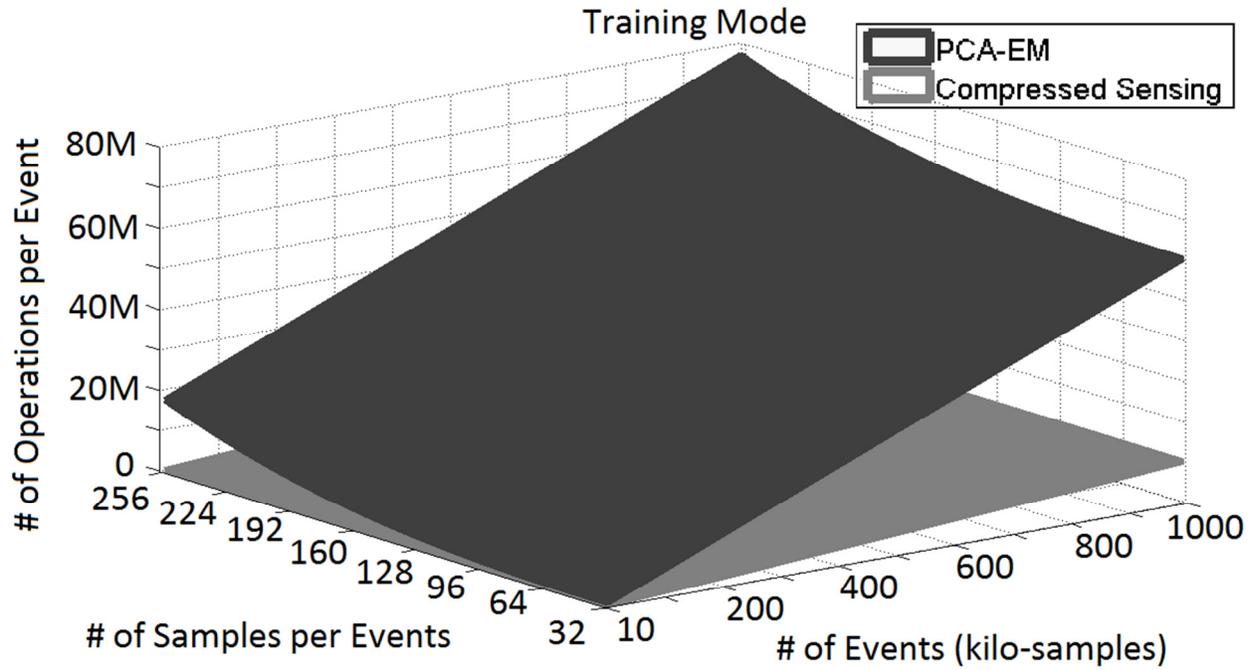
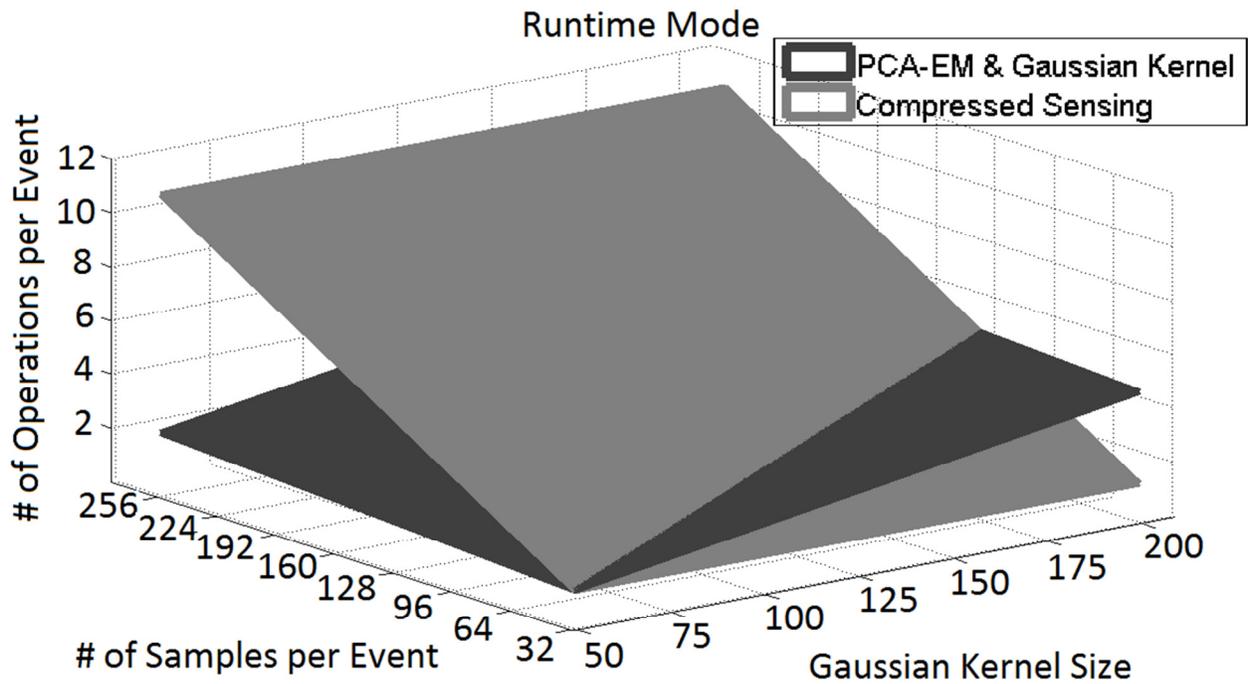


Figure 3.8 Decoding performance of a sample 2D movement trajectory. The black line is the average over 20 trials, while the gray shade around the trajectory represents the estimate variance. Top left: one unit is observed on any given electrode (i.e., neural yield = 1) and therefore no spike sorting is required. The variance observed is due to the network interaction. Top right: every electrode records two units on average (neural yield = 2) and no spike sorting is performed. Bottom left: PCA/EM/Gaussian kernel spike sorting and rate estimation is implemented. Bottom right: Compressed sensing decoding.



(a)



(b)

Figure 3.9 Computational complexity of PCA/EM/Gaussian kernel and the compressed sensing method. (a) Computations per event vs. number of events and number of samples per event in the

training mode. (b) Computations per event vs. number of samples per event and kernel size in the runtime mode. At a sampling rate of 40 KHz and  $\sim 1.2$ - $1.5$  ms event duration (48-60 samples), the compressed sensing method requires less computations than the PCA/EM/Gaussian kernel method. The number of units is assumed fixed in the training mode for both methods ( $P=50$ ).

### **3.6. Discussion**

In this chapter, we have proposed a new approach to directly estimate a critical neuronal response property – the instantaneous firing rate – from a compressed representation of the recorded neural data. The approach has three major benefits: First, the near-optimal denoising and compression allows to efficiently transmit the activity of large populations of neurons while simultaneously maintain features of their individual spike waveforms necessary for spike sorting, if desired. Second, firing rates are estimated across a multitude of timescales, an essential feature to cope with the heterogeneous tuning characteristics of motor cortex neurons. These characteristics are important to consider in long term experiments where plasticity in the ensemble interaction is likely to affect the optimal time scale for rate estimation. Third, as our extensive body of prior work has demonstrated [48, 54], the algorithm can be efficiently implemented in low-power, small size electronics to enable direct decoding of the neural signals to take place without the need for massive computing power. Taken together, these are highly desirable features for real-time adaptive decoding in BMI applications.

We have used a particular model for encoding the 2D hand trajectory for demonstration purposes only. It should be noted, however, that the method is completely independent of that model. What is important to consider is the fact that the sparse representation preserves all the information that needs to be extracted from the recorded neural data to permit faithful decoding to

take place downstream. This includes the features of the spike waveforms as well as the temporal characteristics of the underlying rate functions.

In the tests performed here we have used the same wavelet basis - the symmlet4 - for both spike sorting and rate estimation. This basis was previously demonstrated to be near-optimal for denoising, compression, and hardware implementation. However, the possibility exists to use this basis in the first few levels, and then extend the decomposition from that point on using a different basis that may better represent other features present in the rate functions that were not best approximated by the symmlet4. For example, the “bumps” in the sparse rate estimates in Fig.6 are not as symmetrical in shape as those in the original rate, or those in the Gaussian estimator. For this particular example a more symmetric basis may be better suited.

Estimation of the rate using a fixed bin width may be adequate for certain applications that utilize firing rates as the sole information for decoding cortical responses during instructed behavioral tasks such as goal-directed arm reach tasks [27-29, 55]. These operate over a limited range of behavioral time scales. However, natural motor behavior is characterized by more heterogeneous temporal characteristics that reflect highly-nonstationary sensory feedback mechanisms from the surrounding cortical areas. The firing rates of motor neurons during naturalistic movements are highly stochastic and require a statistically-driven technique that can adapt to the expected variability [40, 56]. This is particularly important given the significant degrees of synchrony typically observed between cortical neurons during movement preparation [57], and also observed during expected and unexpected transitions between behavioral goal representations [58].

While it has been argued that precise spike timing does not carry information about motor encoding [59], one must note that most of the BMI demonstrations to date were carried out in highly-trained subjects performing highly stereotypical, goal-directed behavioral tasks. Very few

studies, if any, have been carried out to characterize naturally occurring movements in naïve subjects. Thus, the potential still exists for new studies that may demonstrate the utility of both neuronal response properties, namely precise spike timing and firing rate, in decoding cortical activity. For that, the sparse representation is able to simultaneously extract these two important elements that are widely believed to be the core of the neural code [60]. Therefore, our approach is the first to offer the solution for extracting both properties within a single computational platform in future generations of BMI systems.

We note that for a fully implantable interface to the cortex to be clinically viable, spike detection, sorting, and instantaneous rate estimation need to be implemented within miniaturized electronics that dissipate very low power in the surrounding brain tissue. More recently, it has been shown that tethering the device to the subject's skull to maintain a wired connection to the implant significantly increases brain tissue adverse reaction, which is believed to negatively affect implant longevity [61]. Therefore, the interface needs to feature wireless telemetry to minimize any potential risk of infection and discomfort to the patient and to elongate the implant's lifespan. We believe that eliminating any of the steps from the signal processing path while preserving the critical information in the neural data will significantly reduce the computational overhead to permit small size, low power electronics to be deployed and accelerate the translation of this promising technology to clinical use.

# 4

## Sorting and Tracking Neural Activity via Simple Thresholding

There has been extensive research on discriminating between spikes from multiple single units in a recorded ensemble [2]. Even when clearly isolated units remain stable over long periods of time, sorting their spike waveforms remains a challenge for a number of reasons. First, action potentials from single cells – while highly stationary when recorded intracellularly – are highly nonstationary when recorded extracellularly, requiring extensive user judgment during the discrimination process. Second, spike waveforms could be substantially similar even though they may have been generated by different cells. Third, the presence of contaminating noise often makes clustering spike waveform features a daunting task; the degree of overlap between putative clusters increases as a function of noise variance. Fourth, the role of individual neurons’ with clearly isolated spike waveforms in encoding a behavioral correlate on one day could substantially change on the next day – a hallmark of brain plasticity [7]. It is therefore imperative to take these factors as well as many others into consideration when devising new methods for discriminating spikes from multiple single units in a recorded neural ensemble.

This chapter introduces a novel approach to discriminate multiple single unit activity in a given neural ensemble recording. The main contribution over the previous chapter [6] is that the approach

is rather simple, yet very efficient compared to classical and state of the art techniques. The approach shares a similar “learning phase” as other techniques in which decision boundaries between clusters of putative single units are first determined. One substantial difference in this phase is that instead of identifying these boundaries using multidimensional search algorithms such as Expectation Maximization (EM) clustering [8], the decision boundaries here are formed by combining simple linear boundaries from a set of binary classifiers. These linear decision boundaries are identified by a set of *optimal thresholds* determined using a simulated annealing search. The testing phase, in turn, results in maintaining the most discriminative features of a given set of spikes and therefore is robust to long-term variability in waveform shapes. In addition, its low computational cost makes it suitable for implementation on ultra-low power miniaturized hardware modules that can be fully implanted in the brain of awake behaving subjects [9].

This chapter is organized as follows. First, we will formulate spike sorting algorithms as a sparse optimization problem and describe the theory behind the simple thresholding. Then, the details of the methods used to collect the neural data and generate surrogate data with known spike waveform characteristics are explained. Finally, we demonstrated the performance of the method for variable signal-to-noise ratios and sampling rates.

#### 4.1. Spike Sorting as a Sparse Optimization Problem

Let’s assume that the extracellular neural voltage trace,  $v(t)$ , recorded at the tip of an electrode can be expressed as a linear superposition of spike waveforms generated by  $N$  spiking neurons as

$$v(t) = \sum_{n=1}^N \sum_{k=1}^{K_n} W_n \left( t - \tau_n^{(k)} \right) + \eta(t) \quad (4-1)$$

where  $W_n(t)$  is a deterministic signal indicating the spike waveform of neuron  $n$ ,  $K_n$  is the number of spikes fired by this neuron within an observation interval of length  $T$ ,  $\tau_n^{(k)}$  are the exact spike times, and  $\eta(t)$  represents all the noise elements, including the background neural noise and the instrumentation noise caused by thermal and electrical effects of the measurement system. Given the prior distribution  $P_\Omega(W_n)$  and the conditional distribution  $P(v(t)|W_n)$ , Bayes' theory can be used to compute the maximum a posteriori (MAP) estimate  $P(v(t)|W_n)P_\Omega(W_n)$  for which the optimal class is the one that maximizes the posterior density [10]

$$n^* = \underset{n}{\operatorname{argmax}} P(v(t)|W_n)P_\Omega(W_n) \quad (4-2)$$

where  $\Omega$  is the set of all spike waveforms,  $\{W_n(t)\}_{n=1}^N \in \Omega$ .

Since the spike waveforms are typically not known, spike sorting involves segmenting the interval around the events surpassing a detection threshold (usually set as three times the standard deviation of the noise), and projecting these segments into a lower-dimensional space using principal components analysis to obtain  $\Omega$  [8]. This set is subsequently clustered into  $N$  independent clusters using algorithms such as  $k$ -means or EM clustering, and the conditional and prior distributions are calculated. The MAP classifier in (4-2) assigns an event to class  $n$  if it maximizes the MAP estimate for that class. In conventional approaches, it is therefore imperative to build the priors and likelihood,  $P_\Omega(W_n)$  and  $P(v(t)|W_n)$ , during the training phase. In spike sorting, however, the average spike waveforms,  $W_n(t)$ , and the number of neurons,  $N$ , are unknown.

For sparse signals, the problem of identifying the signal sources in  $v(t)$  can be formulated as an optimization problem where the Euclidean norm between  $v(t)$  and the observation model in (4-1) is minimized below the noise variance,  $\varepsilon = \|\eta(t)\|_2$ , where  $\|\cdot\|_2$  is the Euclidean norm [11],

$$\min_{N, W_n(t), \vec{\tau}_n} \sum_{n=1}^N K_n \quad s. t. \quad \left\| v(t) - \sum_{n=1}^N \sum_{k=1}^{K_n} W_n(t - \tau_n^{(k)}) \right\|_2 \leq \varepsilon \quad (4-3)$$

Because there are a number of unknowns in (4-3), solving this optimization problem requires dissociating the spike times,  $\vec{\tau}_n$ , from the spike waveforms,  $W_n(t)$ . The first step to achieve this is to sample  $v(t)$  within  $T$  with  $M = \lceil T/\Delta \rceil$  bins, where  $\Delta$  is the bin width (in seconds), to obtain the discrete observations,  $V(t)$ , as

$$V(t) = \sum_{n=1}^N \sum_{m=1}^M \alpha_n^m W_n(t - m\Delta) + \eta(t) \quad (4-4)$$

where  $\alpha_n^m = 1$  indicates that the presence of a spike by neuron  $n$  within the range  $(m-1)\Delta < t \leq m\Delta$ , and  $\alpha_n^m = 0$  indicates otherwise. By replacing (4-4) into (4-3), the optimization problem can be expressed as

$$\min_{N, W_n(t), \vec{\alpha}_n} \sum_n \vec{\alpha}'_n \vec{\alpha}_n \quad s. t. \quad \left\| V(t) - \sum_{n=1}^N \sum_{m=1}^M \alpha_n^m W_n(t - m\Delta) \right\|_2 \leq \varepsilon \quad (4-5)$$

where the vector  $\vec{\alpha}_n = [\alpha_n^m]_{m=1}^M$  represents the binary spike train of neuron  $n$ , and  $A'$  is transpose of  $A$ . The observation model in (4-4) can be expressed as a linear combination of shifted spike waveforms,  $V(t) = \sum_{n=1}^N \vec{\alpha}_n \mathbb{W}_n(t)$ , where the matrix  $\mathbb{W}_n(t) \in R^{m \times M}$  represents  $m$  shifted copies of  $W_n(t)$ , zero-padded up to a length of  $M$ , as

$$\mathbb{W}_n(t) = \begin{bmatrix} W_n(t) & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & W_n(t) & 0 & \cdots & 0 \\ & & & \ddots & & \\ 0 & 0 & 0 & 0 & \cdots & W_n(t) \end{bmatrix}_{m \times M} \quad (4-6)$$

Concatenating all the  $\mathbb{W}_n(t)$  matrices for  $N$  different neurons gives us  $\mathbb{W}(t) = [\mathbb{W}_1(t) | \cdots | \mathbb{W}_N(t)]$ . The observation model can then be represented as  $V(t) = \vec{\alpha}\mathbb{W}(t)$ , where  $\vec{\alpha} = [\vec{\alpha}_1 | \cdots | \vec{\alpha}_N]$ . Considering the fact that spiking activities have a sparse nature, we can expect  $\vec{\alpha}$  to also demonstrate sparsity through a small number of large coefficients distributed among a large number of small coefficients. Thus, the minimization of  $\vec{\alpha}'_n \vec{\alpha}_n$  in (4-5) can be replaced with the following sparse optimization problem as

$$\min_{\vec{\alpha}} \|\vec{\alpha}\|_{L_0} \quad s. t. \quad \|V(t) - \vec{\alpha}\mathbb{W}(t)\|_2 \leq \varepsilon \quad (4-7)$$

where the operator  $\|\cdot\|_{L_0}$  counts the number of nonzero elements in the vector. The problem in (4-7) can be viewed as a classical rate-distortion problem, in which the objective is to reduce the rate - expressed here in terms of the number of non-zero elements in  $\vec{\alpha}$  - while maintaining the distortion - expressed in terms of the Euclidean norm - below a predetermined level [12]. To practically solve (4-7), we replace the  $L_0$ -operator with a thresholding operator. Determining the optimal thresholds is achieved next.

#### 4.1.1. Optimal Thresholding

A threshold,  $\gamma_n$ , is applied to the coefficient vector  $\vec{\alpha}_n$  to provide a sparse coefficient vector,  $\vec{x}_n$ , defined as

$$\vec{x}_n: x_n^m = \begin{cases} \alpha_n^m & |\alpha_n^m| \geq \gamma_n \\ 0 & \text{otherwise} \end{cases} \quad \text{for } m = 1, \dots, M \text{ and } n = 1, \dots, N \quad (4-8)$$

The degree of sparsity in  $\vec{x}_n$  is determined by  $\gamma_n$  such that the sparsity increases with increasing  $\gamma_n$ . This means that the minimization of  $\|\vec{\alpha}\|_{L_0}$  can be approximately replaced by the maximization of the product of  $\gamma_n$ 's, for  $n = 1, \dots, N$ . Thus, the objective function in (4-7) can be replaced by a function that maximizes the product of neuron-specific thresholds for the  $N$  neurons, as

$$\max_{\vec{\gamma} \in \mathbb{R}^N} \prod_{n=1}^N \gamma_n \quad \text{s. t.} \quad \|V(t) - \vec{x}\mathbb{W}(t)\|_2 \leq \varepsilon \quad (4-9)$$

where  $\vec{\gamma} = [\gamma_n]_{n=1}^N$  is the vector comprising these optimal thresholds. This optimization function is non-convex due to the nonlinear thresholding operator in (4-8). Thus, it cannot be solved using regular convex optimization techniques. Instead, iterative methods that perform a generic search such as simulated annealing can be used for finding the optimal thresholds in (4-9). Simulated annealing provides a near-optimal solution without the computational complexity of exhaustive search methods [13]. The unknown parameters in (4-9) are the thresholds,  $\vec{\gamma}$ , and the dictionary  $\mathbb{W}(t)$ . We will reduce the number of these unknown parameters to facilitate the iterative learning by approximating  $\mathbb{W}(t)$  with a predetermined reference dictionary,  $\mathcal{W}(t)$ . This is feasible by considering the fact that spike waveforms generated by different neurons can be faithfully estimated from a linear combination of carefully chosen orthogonal kernels functions [6]. We obtain these kernel functions and subsequently the reference dictionary from a multilevel wavelet transform with  $L$  decomposition levels.

The simulated annealing approach can be summarized as follows: starting from an initial set of arbitrary thresholds,  $\vec{\gamma}^{(0)}$ , we compute the sparse coefficients,  $\vec{x}^{(0)}$ , given  $\mathcal{W}(t)$  and estimate the mean square error (MSE) of the reconstruction,  $MSE^{(0)}$ , by setting the regularization term  $\rho^{(0)} = 1$ . For a maximum of  $K$  iterations, we perturb the thresholds of the previous iteration,  $\vec{\gamma}^{(k-1)}$ , to obtain a new set of thresholds,  $\overline{\vec{\gamma}^{(k)}} = f_p(\vec{\gamma}^{(k-1)})$ . The perturbation function,  $f_p(\cdot)$ , adds a random vector drawn from a zero-mean multivariate Gaussian distribution with a fixed variance  $\delta^2 I$  to  $\vec{\gamma}^{(k-1)}$  to obtain  $\overline{\vec{\gamma}^{(k)}}$ . Given  $\overline{\vec{\gamma}^{(k)}}$  we compute the sparse coefficients,  $\vec{x}^{(k)}$ , and the MSE of the reconstruction,  $MSE^{(k)}$  and  $\rho^{(k)}$  for iteration  $k$ . The perturbed set of thresholds and the updated

regularization term will be accepted if  $MSE^{(k)}$  is less than  $MSE^{(k-1)}$ , otherwise it will be accepted if  $P^{(k)}$  is larger than a random probability,  $\zeta$ , drawn from a uniform distribution between  $[0 \ 1]$ . This probability is estimated as  $P^{(k)} = \kappa_1 - \kappa_2 k/K$ , where constants  $\kappa_1$  and  $\kappa_2$  are set to 0.5 and 0.25, respectively. Table 4.1.a demonstrates the pseudo code for learning the optimal thresholds for discriminating the spike trains while minimizing the reconstruction MSE.

In any given iteration, once we start increasing the thresholds to find the optimal set of thresholds, more coefficients are set to zero. Although this process increases the sparsity of the representation, it also decreases the energy of the reconstructed spike waveforms because a significant number of coefficients is lost. The regularization term  $\rho$  will monotonically increase by increasing the thresholds and thus compensating for the energy loss. Specifically, we adjust  $\rho$  in the  $k^{th}$  iteration to give the lowest mean square error between the observation,  $V(t)$  and  $\rho \times \vec{x}^{(k)}\mathcal{W}(t)$ , in which  $\vec{x}^{(k)}$  are the sparse coefficients obtained by applying the thresholds  $\vec{\gamma}^{(k)}$ .

#### 4.1.2. Threshold selection to maximize sorting performance

So far, the optimization in (4-9) determines the optimal thresholds that minimize the mean square error to provide faithful representation upon reconstruction. These thresholds, however, need not only to provide a faithful representation, but also to maintain a good separability between putative spike waveforms clusters in the sparse domain to achieve good sorting performance. It is noteworthy that using this criterion instead of minimizing MSE may not necessarily result in the same thresholds. This is because a separability criterion accounts for maximal preservation of distance between individual class probabilities in the observed joint density, while a mean square error criterion accounts for maximal separation between signal and noise probabilities and only accounts for bias and variance. Here, the Euclidean norm is replaced with the Fisher information - a

measure of cluster separability - in the sparse domain. For cluster  $n$ , Fisher information is calculated using the within-class separability,  $S_W$ , and the between-class separability,  $S_B$  [10] as

$$S_{W_n} = \sum_{t \in \{\tau_n\}} (\vec{x}(t) - \vec{\mu}_n)(\vec{x}(t) - \vec{\mu}_n)^T \quad (4-10)$$

$$S_{B_n} = K_n(\vec{\mu}_n - \vec{\mu})(\vec{\mu}_n - \vec{\mu})^T$$

$$F_n(S_B, S_W) = \vec{\beta}'_n S_B \vec{\beta}_n / \vec{\beta}'_n S_W \vec{\beta}_n,$$

where  $\vec{\mu}_n = 1/K_n \sum_{t \in \{\tau_n\}} \vec{x}(t)$  is the average spike representation of neuron  $n$ , and  $\vec{\mu} = 1/K \sum \vec{x}(t)$  is the total average across all realizations. For a two-class scenario (neuron  $n$  versus all other neurons) the within-class separability is  $S_W = S_{W_n} + S_{W_{\dot{n}}}$ , and the between-class separability is  $S_B = S_{B_n} + S_{B_{\dot{n}}}$ , where  $\dot{n}$  is the set of cluster indices  $j \in \{1, \dots, N \ \& \ j \neq n\}$ . The vector  $\vec{\beta}_n$  is the optimal linear classifier that separates cluster  $n$  from  $\dot{n}$ . This classifier fuses the outcome of the thresholding blocks,  $\vec{x}(t)$ , to make a decision on whether it is a spike waveform of neuron  $n$  or not. The classifier  $\vec{\beta}_n$  is the eigenvector of the matrix  $S_W^{-1} S_B$  corresponding to the largest eigenvalue. This classifier segregates spike waveforms of neuron  $n$ ,  $W_n(t)$  from  $\Omega$ , for which the projected sparse coefficients of that neuron satisfies  $\vec{\beta}'_n \vec{x}(t)_{\{\vec{x}(t) \in n\}} > \vec{\beta}'_n \vec{x}(t)_{\{\vec{x}(t) \in \dot{n}\}}$ . Note that cluster boundaries for each neuron are first calculated using offline analysis from which the thresholds are subsequently calculated to achieve a similar degree of separability.

The optimization can therefore be stated as simultaneously maximizing the Fisher information and the product of  $\gamma_n$ 's, for  $n = 1, \dots, N$ , as

$$\max_{\vec{\gamma} \in \mathbb{R}^N, \vec{\beta}_n \in \mathbb{R}^N} \frac{\vec{\beta}'_n S_B \vec{\beta}_n}{\vec{\beta}'_n S_W \vec{\beta}_n} \times \prod_{n=1}^N \gamma_n \quad (4-11)$$

where the objective is to find the optimal set of thresholds,  $\vec{\gamma}$ , along with  $N$  optimal binary classifiers,  $\vec{\beta}_n$ , to simultaneously achieve the highest separability and sparsity. Combining the information from  $N$  binary classifiers can also overcome possible labeling errors by individual classifiers, and can detect nonlinear boundaries among classes. It also reduces the complexity of the training by exploiting computationally less expensive classifiers, and increases the reliability of classification decisions [14], [15].

Figure 4.1 provides an example of the performance of both classifiers in sorting two neurons that were initially sorted by a human expert. Each spike waveform is then projected using the two most significant principal components, PC1 and PC2, as the x-axis and y-axis, respectively. The distribution of projected spike waveforms for each neuron is illustrated by ellipsoids to demonstrate the nonlinear decision boundary between the two clusters. These ellipsoids were obtained by fitting the distribution with a Gaussian mixture model (GMM) using at most 10 Gaussian kernels [10]. Given the actual labels, the optimal linear classifier (dashed line), determined by Fisher discriminant analysis (FDA), produces a 91.5% accurate classification [10]. This is while the ensemble classifier that fuses the outcome of two binary classifiers, determined by the optimal thresholds Thr1 and Thr2 as the vertical and horizontal lines, produces a comparable 90.5% accurate classification. Note that the two optimal thresholds are not necessarily the optimal Bayesian thresholds estimated independently from the horizontal or vertical histograms, as demonstrated in Figure 4.1.

Table 4.1 Pseudo code for determining the optimal thresholds by (a) minimizing the MSE, (b) maximizing the sorting performance

**(a) Inputs:**

- Arbitrary thresholds,  $\vec{\gamma}^{(0)}$
- Regularization term,  $\rho^{(0)} = 1$
- Compute sparse coefficients,  $\vec{x}^{(0)}$ , then  $MSE^{(0)} = \|V(t) - \rho^{(0)} \times \vec{x}^{(0)} \mathcal{W}(t)\|_2$

```

1: for iterations  $k = 1, \dots, K$ 
2:    $\overline{\vec{\gamma}^{(k)}} = f_p(\vec{\gamma}^{(k-1)})$ 
3:   Compute  $\vec{x}^{(k)}$ ,  $\rho^{(k)}$  and  $MSE^{(k)}$ 
4:   if  $MSE^{(k)} \leq MSE^{(k-1)}$  or  $P^{(k)} > \varsigma$ 
5:     update  $\vec{\gamma}^{(k)} = \overline{\vec{\gamma}^{(k)}}$ 
6:   end
7: end

```

**(b) Inputs:**

- $\{W_n(t), \{\tau_n\}\}_{n=1}^N$
- Arbitrary thresholds,  $\vec{\gamma}^{(0)}$
- Estimate optimal ensemble classifiers,  $\vec{\beta}_n^{(0)}$  for  $n = 1, \dots, N$ , and  $S^{(0)} = \sum_n F_n(S_B, S_W)$

```

1: for iterations  $k = 1, \dots, K$ 
2:    $\overline{\vec{\gamma}^{(k)}} = f_p(\vec{\gamma}^{(k-1)})$ 
3:   estimate optimal ensemble classifiers  $\vec{\beta}_n^{(k)}$ , for  $n = 1, \dots, N$ , and  $S^{(k)}$ 
4:   if  $S^{(k)} \geq S^{(k-1)}$  or  $P^{(k)} > \varsigma$ 
5:     update  $\vec{\gamma}^{(k)} = \overline{\vec{\gamma}^{(k)}}$ 
6:   end
7: end

```

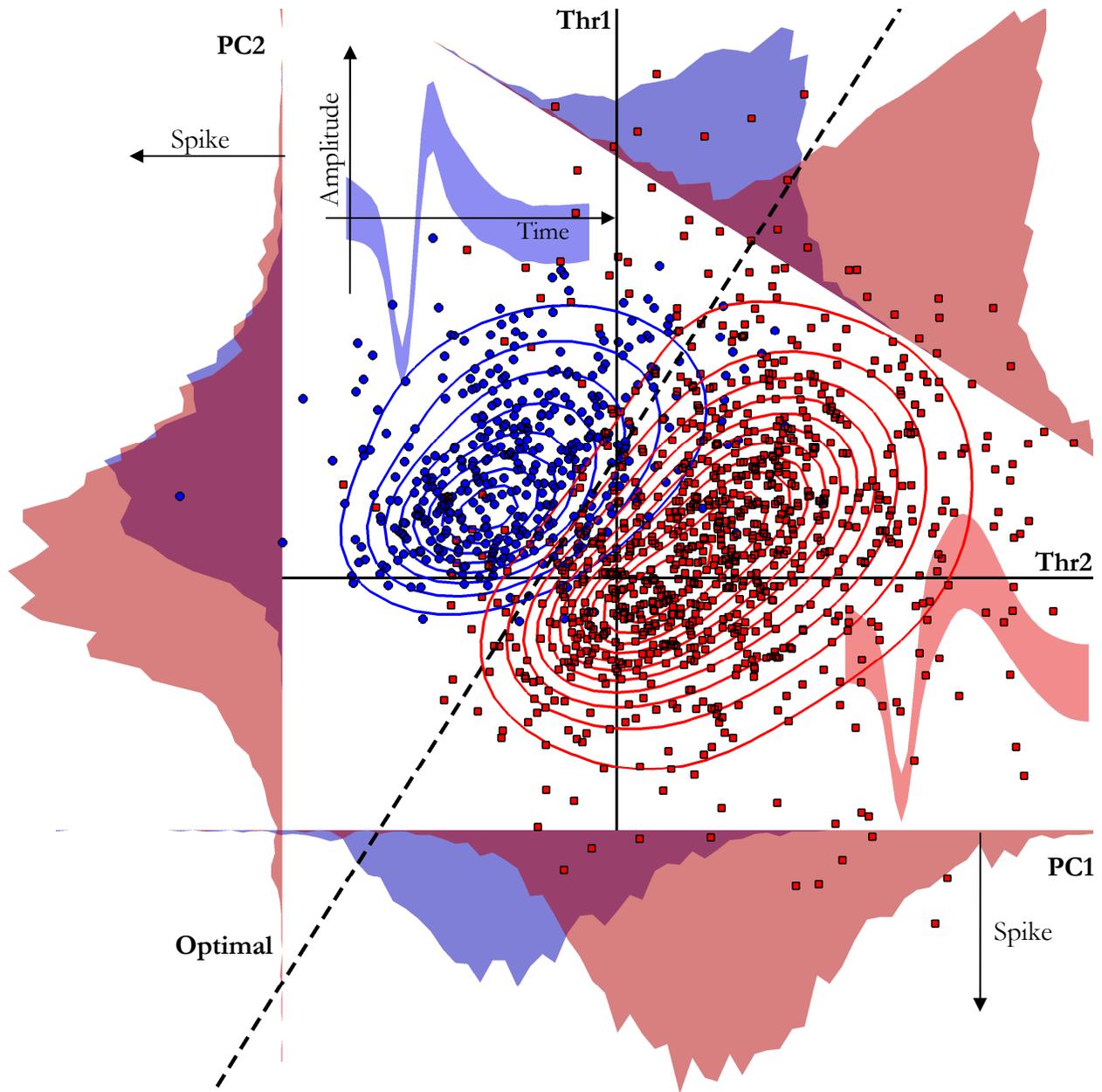


Figure 4.1 The distribution of spike events in a 2D space obtained using PCA. Spike events were initially clustered into two group of sample, and colored to represent two separate neurons. The average spike waveform of each neuron is demonstrated, where the shaded area represents the standard deviation. The nonlinear boundary between the two clusters is illustrated by the ellipsoids that represent the distribution of each cluster. The distribution of each cluster was estimated by fitting at most 10 Gaussian kernels using Gaussian mixture model. Directly applying a linear

classifier (dashed line) introduces a 91.5% accurate classification rate, while optimal thresholding (horizontal and vertical lines) introduces a comparable 90.5% accurate classification rate. To demonstrate that these optimal thresholds are not necessarily the optimal thresholds to independently segregate the two clusters in either the 1D horizontal or vertical representation, we have shown the 1D horizontal and vertical marginal distributions using 1D-histogram representation for the features corresponding to the first and second most significant principal components, PC1 and PC2, respectively.

Maximizing the Fisher's information in (4-11) using simulated annealing can be summarized as follows: the spike waveforms for  $N$  neurons,  $\mathbf{W}_n(\mathbf{t})$ , and their spike times,  $\{\tau_n\}$ , are obtained using an offline spike sorting algorithm. Starting from an arbitrary set of thresholds in iteration (0),  $\vec{\gamma}^{(0)}$ , we find the optimal classifier for neuron  $n$ ,  $\vec{\beta}_n^{(0)}$ , for  $n = 1, \dots, N$ . Let us assume the sum of Fisher's information as  $S^{(0)} = \sum_n F_n(S_B, S_W)$ . For a maximum of  $K$  iterations, we perturb the thresholds found in the previous iteration,  $\vec{\gamma}^{(k-1)}$ , to obtain  $\vec{\gamma}^{(k)}$ . The optimal classifiers,  $\vec{\beta}^{(k)}$ , are determined for the perturbed set of thresholds. If the sum of Fisher's information for the perturbed thresholds,  $S^{(k)}$ , is larger than  $S^{(k-1)}$ , the perturbation will be accepted, otherwise it will be rejected with increasing probability. Table 4.1b demonstrates the pseudo code for this approach.

### 4.1.3. Multilevel Wavelet Decomposition

An important element in the process of finding the optimal thresholds is the approximation of the spike waveforms,  $\mathbf{W}(\mathbf{t})$  through the predetermined reference dictionary,  $\mathcal{W}(\mathbf{t})$ . This dictionary should represent a wide range of spike waveforms, such that most observed spike waveforms can be built using the kernels (rows) of  $\mathcal{W}(\mathbf{t})$ . These kernels can be obtained through orthogonal multilevel wavelet decomposition [17]. Symlets, for example, have shown to be near-optimal for

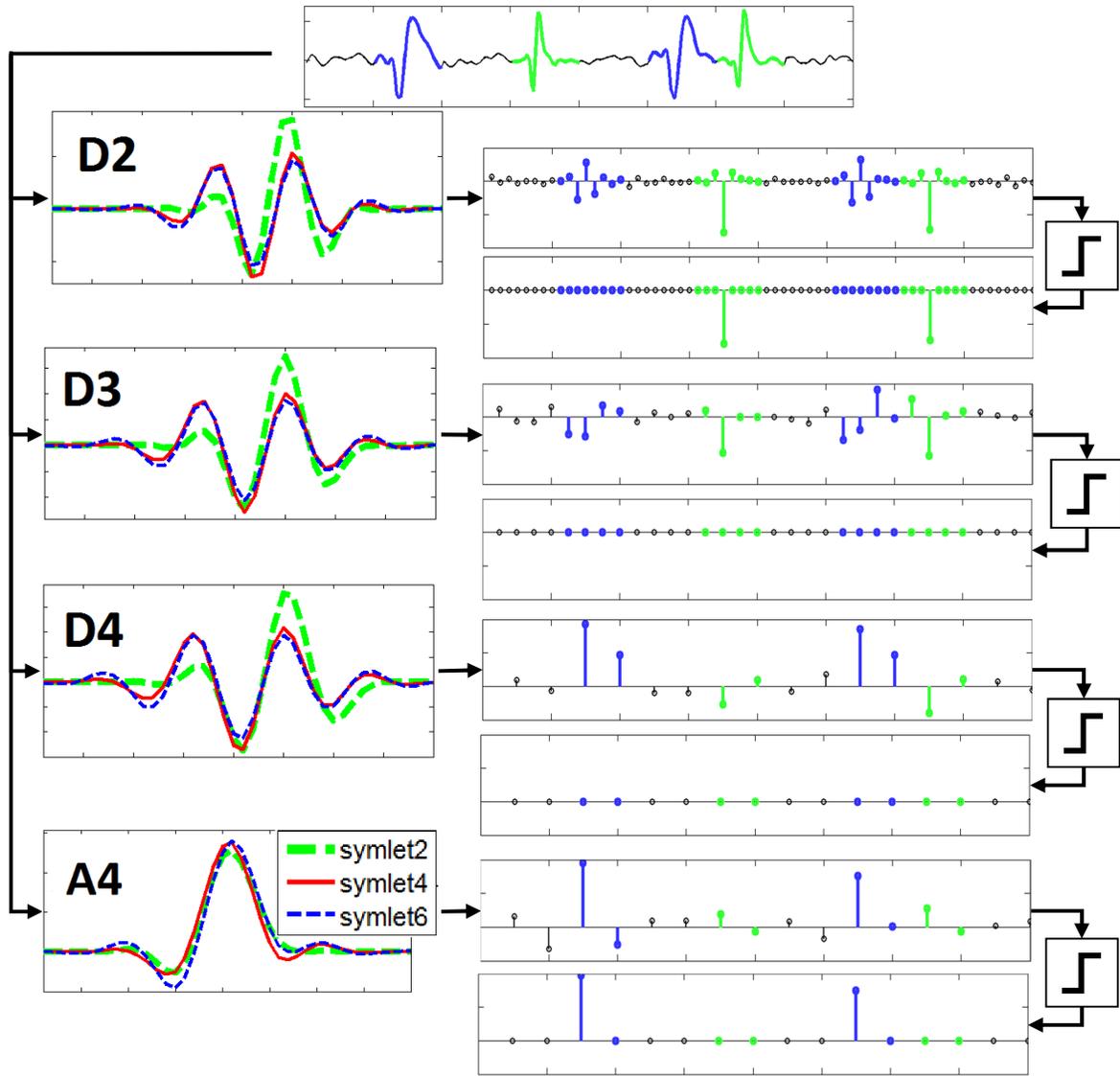
preserving the spike features of a large number of typical spike waveform classes. The left column of Figure 4.2a illustrates four Symlet kernels of the orders 2, 4, and 6 at a sampling rate of 25 kHz that were used to estimate the detail coefficients at decomposition levels 2, 3 and 4 and the approximation coefficients of decomposition level 4. Note that the sampling rate only affects the width of these kernels, but not their form or shape. This figure demonstrates a sample recording,  $v(t)$ , with four spike waveforms, color coded to belong to either the blue or green neuron. The right column demonstrates the coefficients of the detail level 2, 3, 4 and approximation level 4 using Symlet 4 before and after thresholding. The green neuron can be segregated with a neuron-specific threshold at detail level 2, and the blue neuron at approximation level 4. The remaining coefficients are set to zero to increase the overall compression rate.

Since the typical duration of action potentials is around one millisecond [18], wavelet kernels are expected to have a similar temporal support – defined as the width that accounts for more than 99% of the total energy. Figure 4.2b demonstrates the temporal support of Symlet kernels for different sampling rates and decomposition levels. The temporal support increases as the sampling rate decreases or as the decomposition levels increase [19]. A particular Symlet can be selected to provide the best temporal support for a given spike waveform. As an example, Figure 4.2c demonstrates the temporal support for Symlet kernels with four decomposition levels for different sampling rates. For a sampling rate of 25 to 30 kHz, only Symlets 4 and 6 provide the necessary temporal support of 1.2 msec, respectively. Alternatively, Symlet 2 provides this support at a sampling frequency of around 16 KHz. Figure 4.3d demonstrates the temporal support for Symlet kernels at a sampling rate of 25 kHz, but for different decomposition levels. Depending on the sampling rate by which the neural data is recorded and also the number of intended decomposition levels, we can always find the optimal order for the Symlet kernels to obtain the best sparse representation.

## 4.2. Collecting Neural Data

### 4.2.1. Generating Surrogate Data with Known Waveform Characteristics

To evaluate the performance of the optimization methods, synthetic neural recordings for which the ground truth is available is needed. To generate such data, action potentials were detected and extracted from neural recording in the somatosensory cortex of an anesthetized rat using a 32-channel microelectrode array in response to unilateral whisker stimulation. Details of the experimental procedures to obtain these recordings are described elsewhere [16]. All procedures were approved by the Institutional Animal Care and Use Committee at Michigan State University following NIH guidelines. Spike waveforms that passed a threshold set as 3 times the standard deviation of noise signal were detected, aligned and extracted relative to their minimum point and sorted using the classical spike sorting algorithm. We detected 11 clearly distinguishable units on 5 channels of one recording session. To generate synthetic data, multiple spike realizations were drawn from a Gaussian distribution with the mean set as the spike waveform of neuron  $n$ ,  $W_n(t)$ , and the covariance set as  $W_n'(t)W_n(t)$ . Figure 4.3 demonstrates 50 spike realizations for 11 neuron population, in addition to 50 noise samples that were extracted, cropped and aligned from an actual experimental recording with a similar length to the spike realizations. Figure 4.3 demonstrates the distribution of these samples in a 3-dimensional PCA feature space. The signal-to-noise-ratio (SNR) of this data was calculated as  $10 \log_{10} \sigma/\varepsilon$ , where  $\sigma$  is the average root mean square of the spike waveforms. We produced synthetic data with variable SNR by varying the scaling of the noise signal,  $\eta(t)$ .

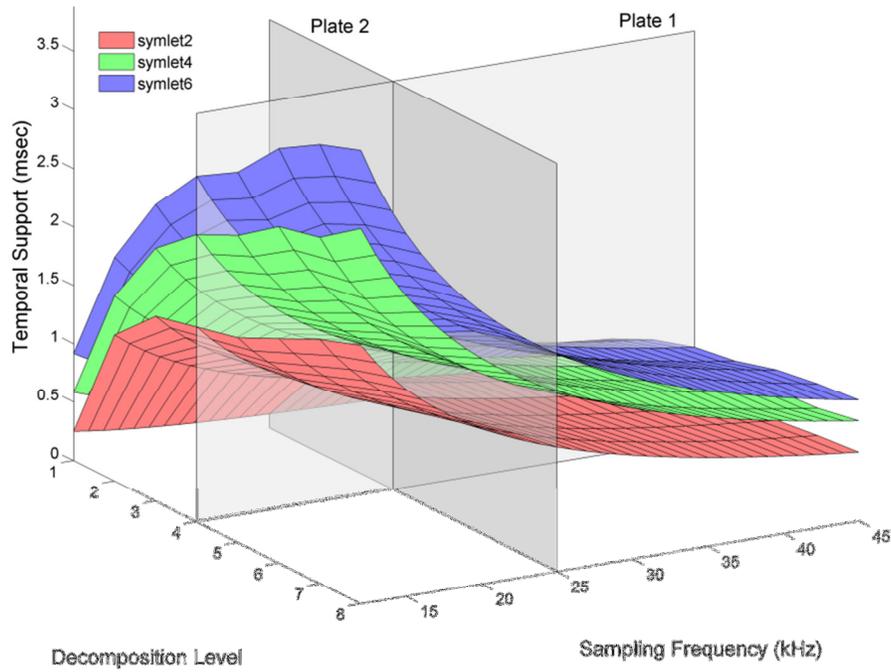


(a)

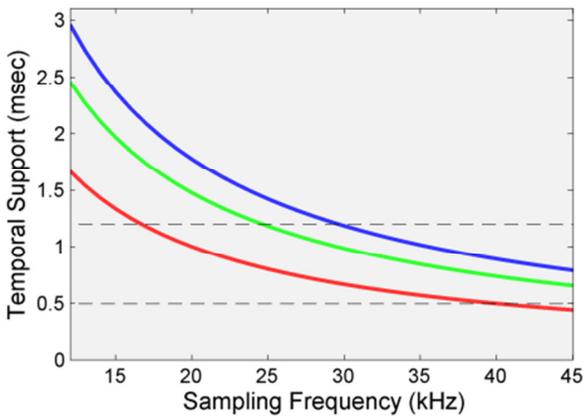
Figure 4.2 (a) Schematic diagram for obtaining the sparse representation from the raw recordings using a 4-level wavelet decomposition in a tree structure using the Symlets 2, 4, and 6, sampled at 25 kHz. The left column in an order demonstrates the orthonormal basis functions for detail levels 2, 3 and 4 and approximation level 4. The right column represents the wavelet coefficients obtained from this decomposition before and after thresholding. The 8msec simulated neural recording demonstrates four spike waveforms generated by the activity of two neurons. We can show for this toy example that a neuron-specific threshold can be estimated to segregate the activity of one

neuron from the rest. The green neuron is segregated at detail level 2, while the blue neuron is segregated at approximation level 4. (b) Temporal support of wavelet basis functions extracted from Symlets 2, 4 and, 6 for different decomposition levels and different sampling rates. (c) Temporal support for different sampling rates at decomposition level 4. (d) Temporal support for different decomposition levels at a 25 kHz sampling rate.

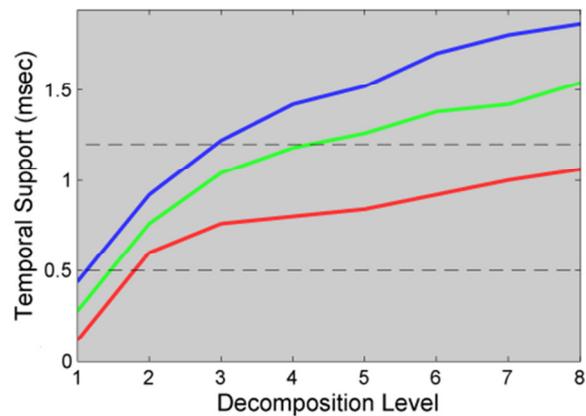
Figure 4.2 (continued)



(b)



(c)



(d)

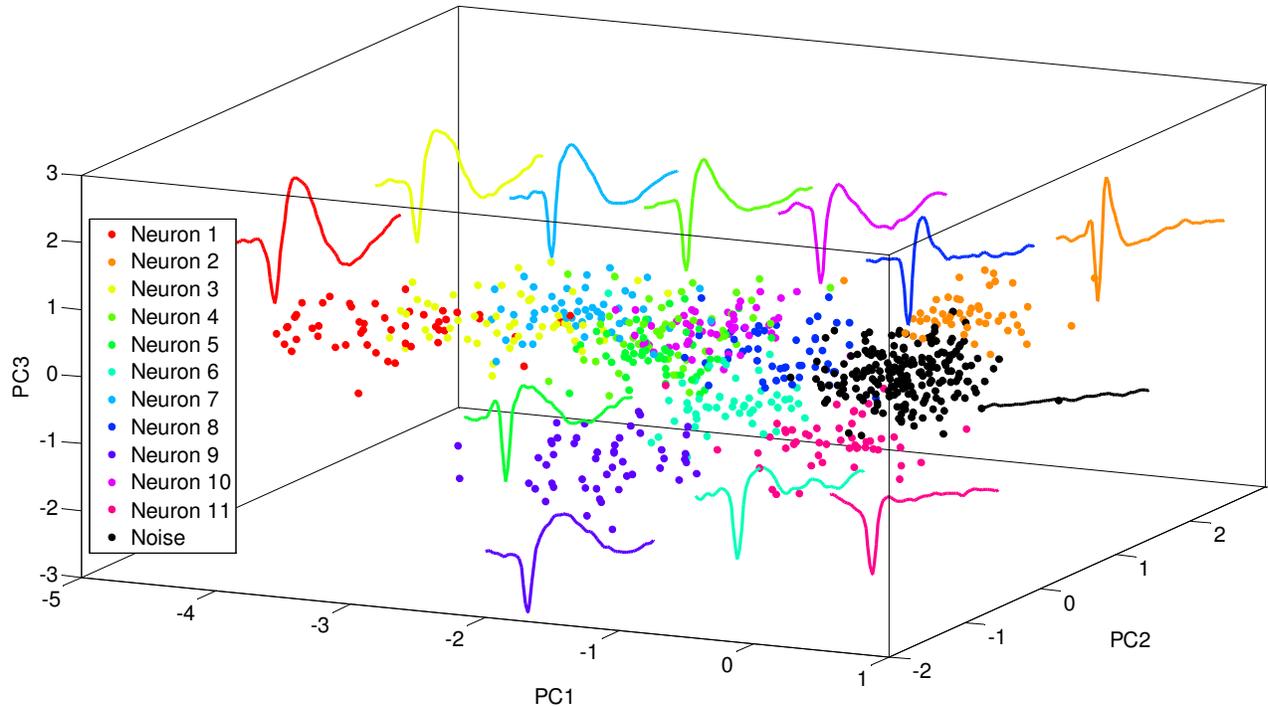
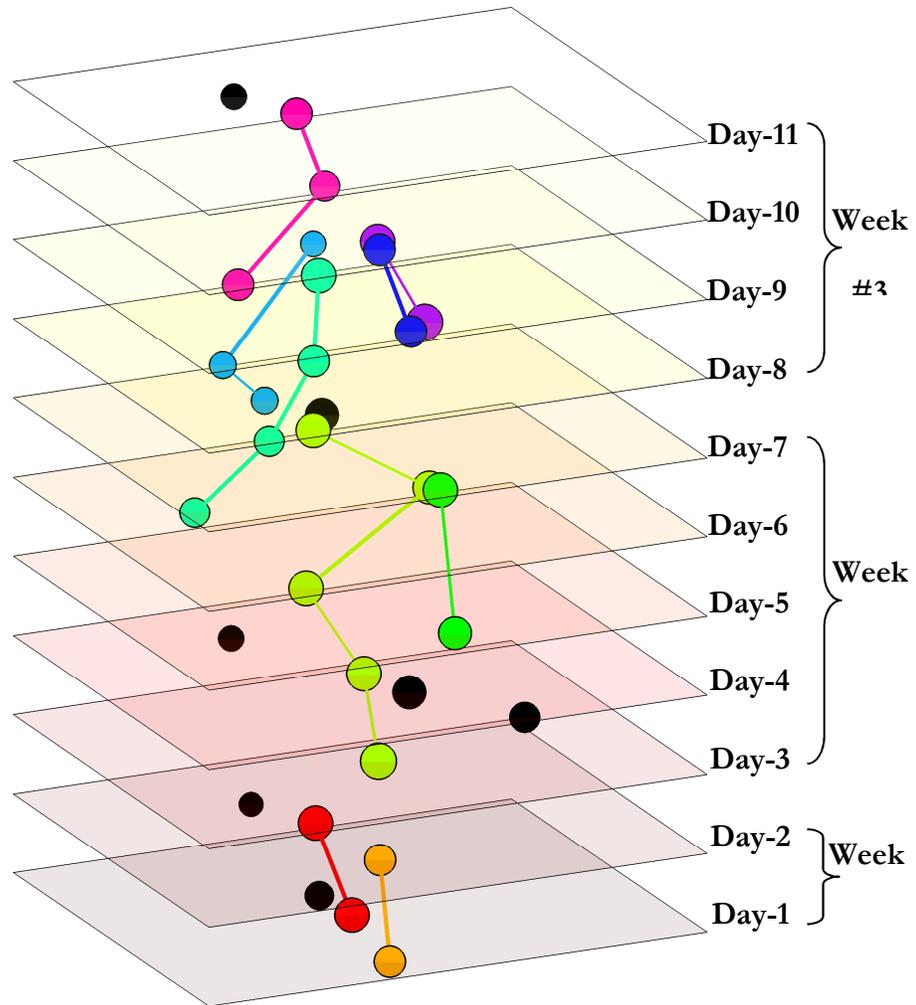


Figure 4.3 Distribution of spike PCA features in a 3D space. The simulated spikes were generated from 11 average spike waveforms of neurons recorded in vivo in the rat’s somatosensory cortex, in addition to noise events (action potentials are average spike waveform for the 11 neurons and noise events). Each sample is color coded to demonstrate the 11 neurons and noise.

#### 4.2.2. Tracking Neurons over Long-Term Recordings

To analyze the stability of spike sorting algorithms in detecting isolated units across many sessions of recorded data, we used a dataset containing multiple units collected across multiple consecutive days. Details about the data collection are described elsewhere [20]. In brief, the subject was trained to perform a delayed sensory-guided task and was implanted with a 32 microwire array in mPFC, where extracellular activity was recorded for hour long sessions. Out of multiple sessions acquired, 11 consecutive sessions with appropriate behavioral performance over a span of three

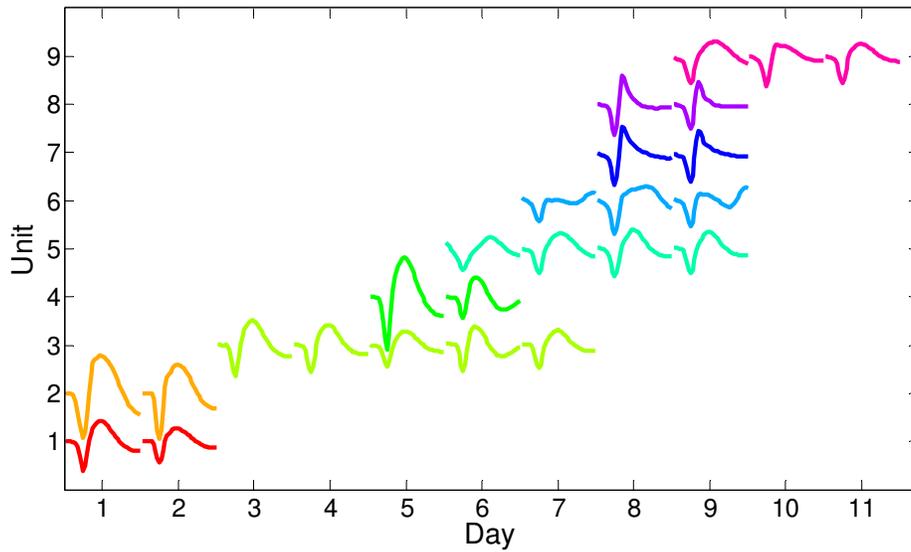
weeks were selected and used in this study to assess the robustness of sorting with respect to changes in the statistical characteristics of the collected signal. The number of neurons in each recording session was determined by a fully automatic spike sorting algorithm that takes into account features extracted from the spike waveforms, in addition to the inter spike interval (ISI) and average firing rate of each neuron [21]. The features include the peak-to-peak amplitude of spike waveforms, the time between the two peaks, and spectral ratio (SR), estimated as the amount of energy between the 2 to 5 kHz sampling range over the total energy. The number of neurons in each session is set as the least amount of neurons that could faithfully explain the variance observed among all spike waveforms. Using a similar combination of spike waveform features and encoding characteristics, neurons from consecutive days are matched to be the same (Figure 4.4a). These neurons show high similarity in both their spike waveforms (Figure 4.4b) and encoding characteristics and can confidently be assumed to be the same neuron, as least from a neural decoding perspective [21]. A total number of 212 neurons were detected from 16 functional electrodes over 11 days, thus providing an average neural yield of  $1.2 \pm 1.3$  units per channel per day with an average SNR of  $5.75 \pm 1.67$  dB.



(a)

Figure 4.4 Tracking neurons for multiday recordings on a single electrode: (a) Each plate represents one day of recording, for a total of 11 days over a period of three weeks. Detected neurons in each day are illustrated by projecting their average spike waveform using PCA, demonstrated as a dot, in which the size of this dot is proportional to the neuron's average firing rate. Neurons that have been tracked over multiple days are demonstrated by similar colors, while other ones are black. A total of nine neurons have been detected to be stable for at least two consecutive days. (b) The average spike waveform of the nine stable neurons over the 11 days color-coded to match (a).

Figure 4.4 (continued)



(b)

Variations in the spike waveforms fired by a single neuron have not only been observed for a span of daily recordings, but also for a span of minutes, thus requiring robust sorting for even short term recordings. Figure 4.5 illustrates these variations in the spike waveforms fired by neurons over hour long sessions. Figure 4.5a demonstrates the 2D representation of spike waveforms of 4 different neurons using PCA. Some of these neurons show a significant change in their energy as the session progresses. Figure 4.5b demonstrates the changes in the spike waveforms more clearly. These variations are demonstrated to be changes in the peak-to-peak amplitude (Figure 4.5c), while the peak-to-peak time and spectral ratio seem to be more robust. Any spike sorting algorithm that uses PCA feature will be sensitive to these variations and cannot appropriately detect and track these neurons over recording sessions.

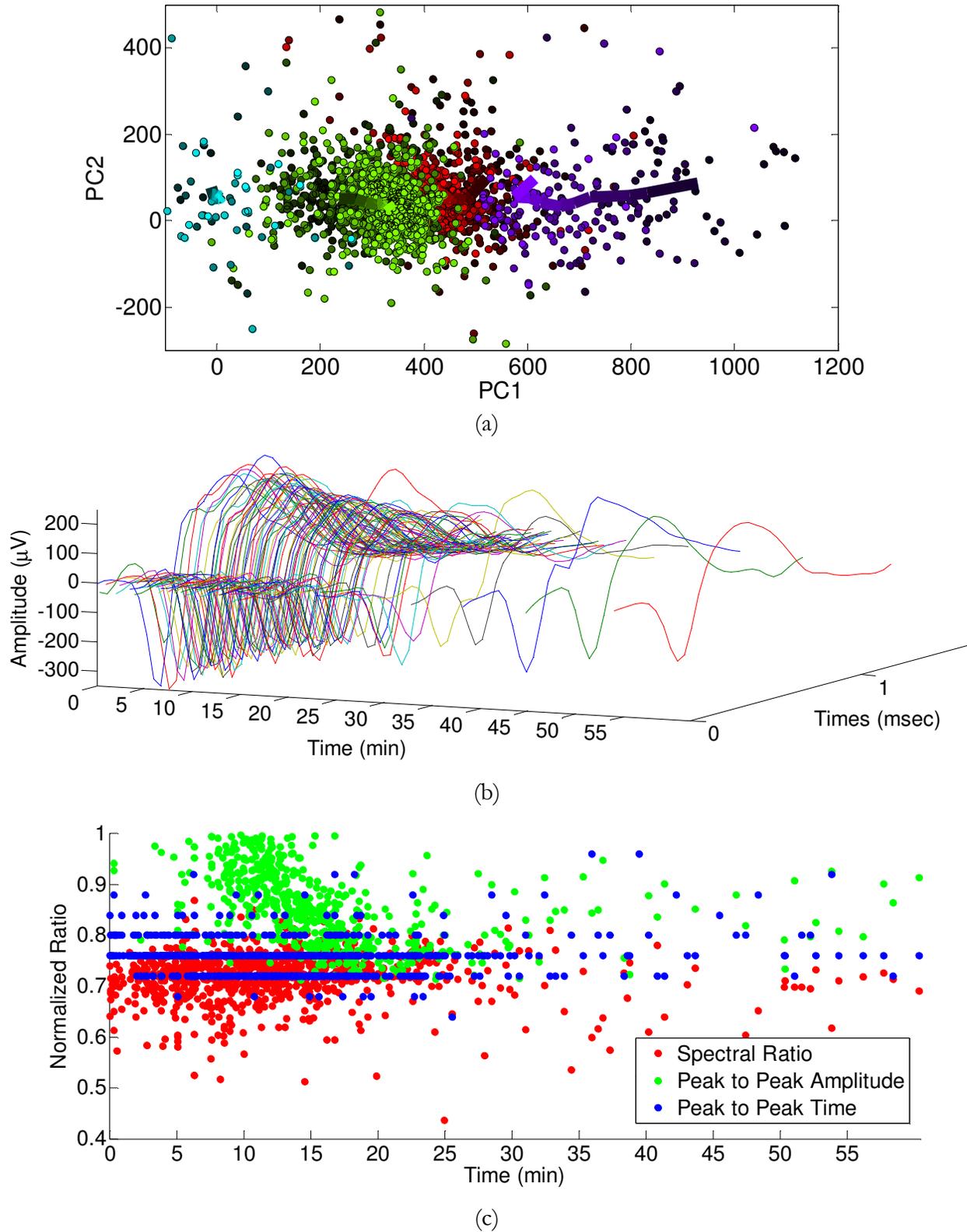


Figure 4.5 Tracking neurons for single day recordings: (a) 2D representation of spike waveforms detected in a single day recording using PCA (4 neurons were detected). Each dot represents the

projection of a spike waveform, color coded to match one of the 4 neurons, where the intensity of its color determines the time by which they were detected (higher intensity mean later times). The average 2D representation of these dots is demonstrated by a line with a similar coloring pattern. A line that extends the longest in the 2D representation demonstrates the largest changes in the average energy of the spike waveforms fired by the corresponding neuron (here the purple-colored neuron). (b) The spike waveforms of the purple-colored neuron. The x-axis demonstrates time in minutes, by which each spike waveform was fired, and the y-axis demonstrates the time in milliseconds by which a spike waveform is recorded. (c) Demonstrates three key features of the spike waveforms generated by the purple-colored neuron, the peak-to-peak, the time span between peaks, and the spectral ratio (SR). While the peak-to-peak time and SR features seem to be constant for this neuron, the peak-to-peak rapidly decreases in the first 30 minutes.

### **4.2.3. Considerations for Practical Implementation**

In our previous work [6], we have demonstrated how a thresholding strategy could be useful to fulfill the design constraints of a fully implantable neural interface that performs instantaneous spike detection and sorting to cope with a limited wireless telemetry bandwidth and minimize the overall latency in BMI applications [9], [17], [22], [23]. We have proposed to equip the system with three ‘operational’ modes, namely a monitoring, a compression and a sensing mode. The optimal thresholds for the compression and sensing modes are determined external to the implantable system using raw data collected during the monitor mode [9]. Here, we refine this design to only include the compression and sensing modes, thereby removing the need for the monitoring mode, which is time consuming due to the bandwidth constraints of the wireless telemetry. Figure 4.6 demonstrates a schematic diagram of this alternative design. It features an additional block that determines the optimal threshold in the compression mode by continuously calculating the MSE

through the regularization terms indicated in Table 4.1a. Because clustering is needed before a sensing mode can be implemented, the optimal thresholds and classifiers for sorting are done externally (as in Table 4.1b) using the data collected during the compression mode. The important advantage of this design over the previous design is the *self-evaluation* capability that permits continuous adjustments of the optimal compression thresholds to cope with possible SNR degradation during long term recordings.

The computational complexity logically becomes an important evaluation metric. Let us assume that a 32 channel neural recording is quantized into *8bits* samples - which have been shown previously to maintain desired information [9]. Using the lifting method to implement the multilevel wavelet decomposition[22], 8 multipliers and 8 adders are required per sample, thus for a spike waveform of length  $L$  samples,  $224 \times L$  full adder operations are required for each recording channel, considering that an *8bit* adder requires 8 full-adders, and a *8bit* multiplier requires 20 full-adders [24]. A similar number is required for the inverse multilevel wavelet decomposition. To apply the optimal thresholds and regularization terms, four *8bit* adders and four *8bit* multipliers are required, therefore requiring a total of  $560 \times L$  full adder operations. To estimate the MSE for a length of  $T$  samples, we need  $1 + \lceil \log_8 T \rceil$  bytes of memory, in addition to approximately 100 fixed bytes of memory to store the residuals of the lifting method in the multilevel wavelet decomposition, the optimal thresholds and regularization terms for each recording channel [9]. It can be seen that the memory size scales sub-linearly - while the number of full adders scales linearly - with the number of samples. The perturbation of thresholds and regularization terms for every  $T$  sample is performed by the central controller using a linear feedback shift register (LFSR) using XOR as the linear function [25]. The LFSR for each threshold requires an *8bit* register and an *8bit* full adder.

Accepted perturbed thresholds are stored in the predetermined registers before beginning the next round of perturbations.

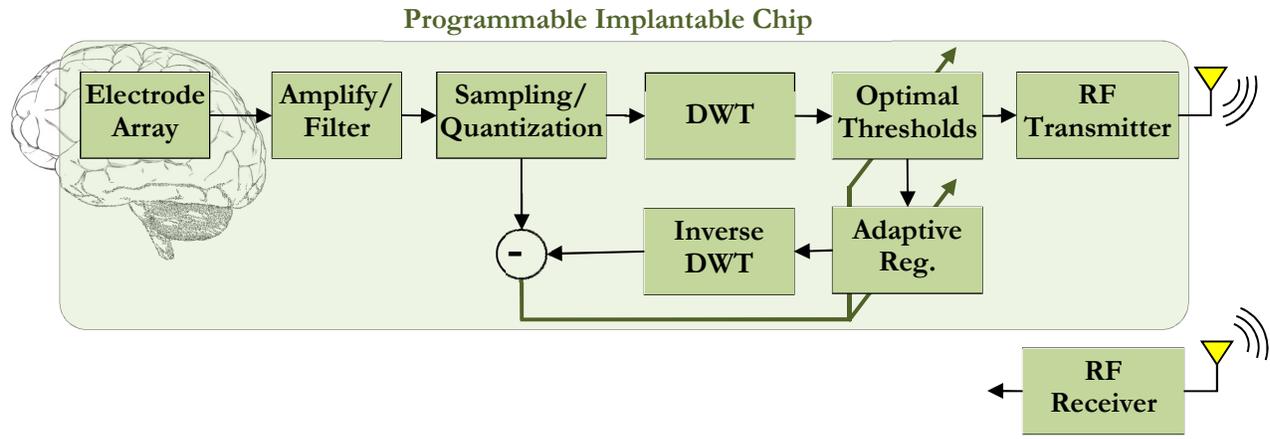


Figure 4.6 Schematic diagram of the data flow in the implantable wireless system. Neural recordings are initially amplified and filtered in the analog front-end prior to sampling/quantization and transmission over the wireless channel, where the raw data is projected into the sparse representation with the discrete wavelet transform (DWT) block, followed by the optimal thresholding block, to reduce the telemetry bandwidth. This diagram also features the adaptive regularization and inverse DWT blocks to project back the sparse representation into the original representation to continuously estimate the MSE. The green arrows demonstrate the iterative learning that takes place by perturbing the set of thresholds and regularization terms for the compression mode.

### 4.3. Compressive Spike Sorting Performance

#### 4.3.1. Spike Sorting Performance vs. SNR and Compression Rate

We evaluated the performance of our approach for both compressing and spike sorting as a function of the SNR and sampling rates and compared it to the classical spike sorting. The compression metric is important to evaluate strategies for telemetry bandwidth reduction during wireless neural recording using implantable systems [17]. The compression rate was estimated as the

number of zero coefficients over the total number of coefficients. In the classical PCA-based spike sorting, the lower dimensional PCA representation can be viewed as one form of signal-dependent compression of the neural data. The compression rate in such case would be the percentage of variance accounted for using a subset of the eigenvectors over the total number of eigenvectors. Figure 4.7 demonstrates the average performance using two neurons, collected at different SNR between 0 to 5 dB. The sorting performance in (a) is expressed as the average true positive rate (TPR), where TPR is estimated as the number of accurately sorted spike waveforms over the total number of spike waveforms from each neuron. The sorting performance in (b) is expressed in terms of the average false positive rate (FPR), where FPR is estimated as the number of noise events over the total number of non-spike events that pass the detection threshold of each neuron. For the 2-neuron mixture, we sampled all possible combinations and produced  $\binom{11}{2} = 55$  synthetic data sets using spikes from the 11-neuron population recorded in one session. An FDA classifier (Figure 4.1) was trained given the spike trains of each neuron, and a test data set was used for calculating the TPR and FPR. The average TPR and FPR for our spike sorting approach are plotted as a blue trace in Figures 4.7a and 4.7b, respectively. We also plotted the projection of this trace on the TPR-compression plane, SNR-compression and TPR-SNR planes for clarity. It can be seen that better spike sorting performance and compression rates are achieved compared to PCA-based approach. Unlike the classical approach, our approach is highly robust against SNR variations as measured by the high TPR. A decrease in SNR – as expected – results in an increase in FPR and consequently a decrease in the compression rate.

### 4.3.2. Wavelet Basis Function vs. Sampling Rate

The sampling rate is an important element in the selection of the wavelet basis function to sparsify the signals. Figure 4.8 demonstrates the effect of choosing Symlet 2 or Symlet 4 on the

average TPR for different SNRs as a function of the sampling rate. The sorting performance for the Symlet 4 basis function dominates the one of Symlet 2 as the sampling rate increases. For sampling rates close to Nyquist (around 15 kHz), Symlet 2 provides a temporal support close to action potentials, while for sampling rates around 25 kHz, Symlet 4 provides a similar support. The difference in performance between the two basis kernels becomes negligible at high SNRs.

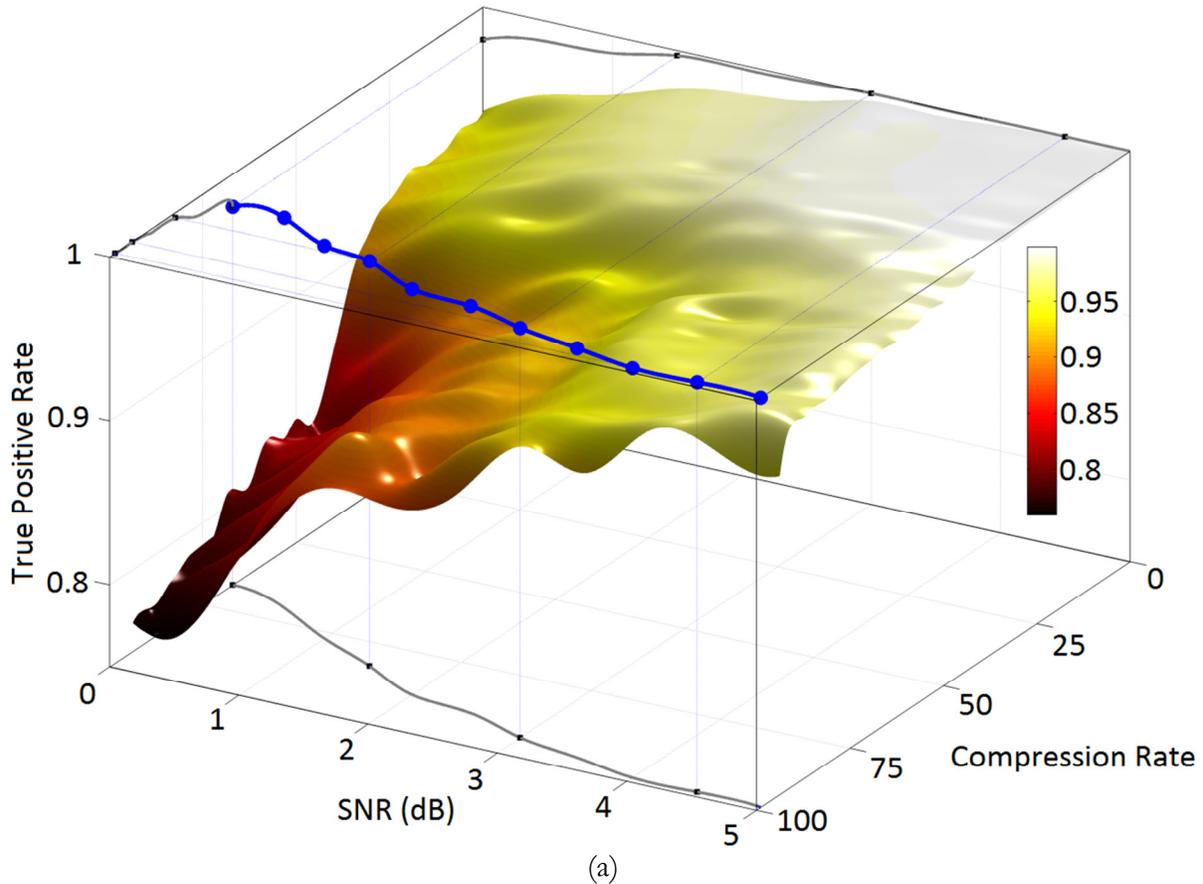
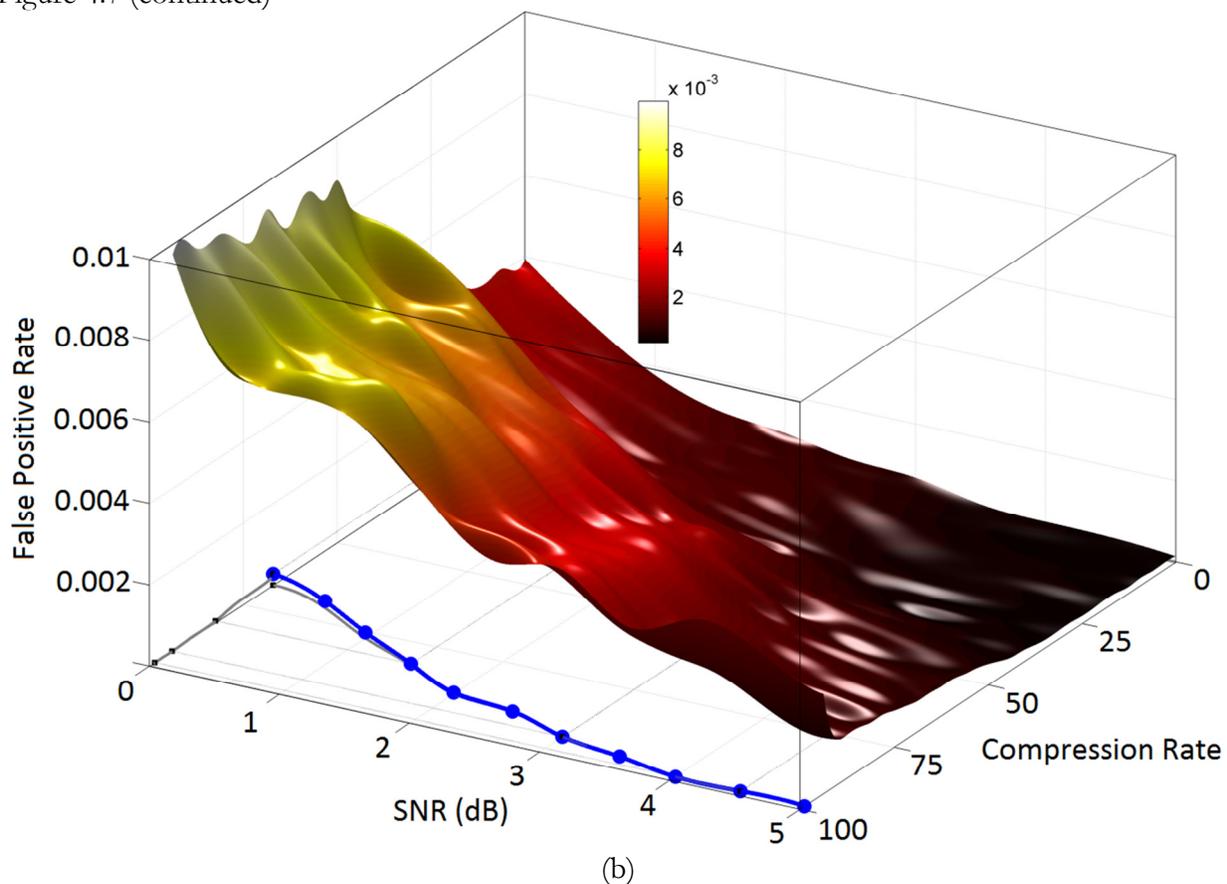


Figure 4.7 (a) Average TPR of an FDA classifier trained for different SNR and compression rates. The compression is achieved by projecting the raw recordings using PCA. A combination of 55 recording segments - each includes the activity of two distinct neurons - was used. The average TPR of our spike sorting approach is demonstrated as the nonlinear blue trace in the 3D space. The projection of this trace is plotted in three planes for better visibility. (b) Average FPR of the FDA classifier and our spike sorting approaches versus variable SNR and compression rates.

Figure 4.7 (continued)



### 4.3.3. Analysis of Real Data

Figure 4.9a demonstrates 10 seconds out of a 27-minute neural recording at a sampling rate of 25 kHz obtained from the rat’s somatosensory cortex during an optogenetic stimulation experiment. Optogenetics were used here to evoke precisely timed spikes from genetically targeted neurons, which helped in faithfully estimating the prior density of spiking compared to spontaneous or natural stimulus-driven activity. Figure 4.9b demonstrates 3D PCA feature space of all events that passed the detection threshold and aligned relative to their minimum point. We clustered the dataset into 5 classes, and illustrated the average spike waveform and its standard deviation in figure 4.8b [21]. The cluster colored as black illustrates putative noise events that passed the detection threshold.

Using simulated annealing (Table 4.1a), we calculated the optimal thresholds for minimizing the MSE –termed “compressed mode”. After 300 iterations, we found the optimal set of thresholds that

provided 90% compression rate. The reconstruction using the remaining 10% of coefficients is demonstrated in Figure 4.9a superimposed on the original data trace for comparison. It can be seen that the form of spike waveforms are faithfully preserved, while the noise has been significantly suppressed. In the compressed mode, the simulated annealing (Table 4.1b) is used to find the optimal thresholds that maximize separability – labeled “sensed mode”. After 300 iterations, these thresholds amounted to a compression rate of more than 99%. Figure 4.10a demonstrates the compression rate and normalized MSE over the two learning phases, and Figure 4.10b demonstrates the progression of the thresholds’ values over the same phases.

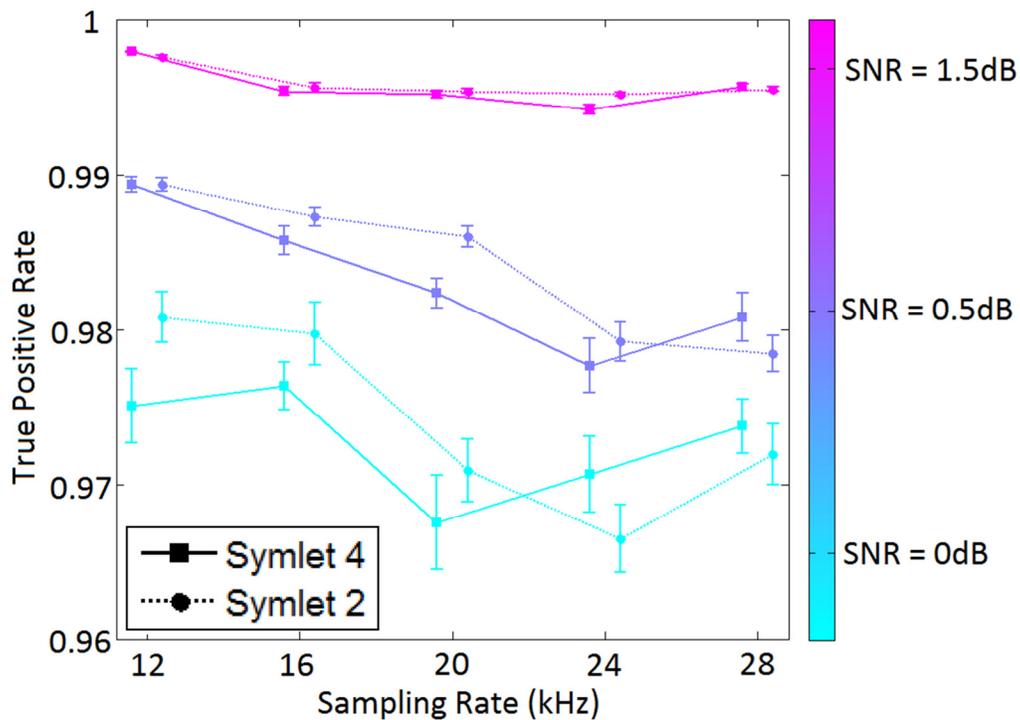


Figure 4.8 Effect of wavelet basis choice on the average TPR for variable sampling rates between 12 and 28 kHz, and SNRs of 0, 0.5 and 1.5 dB. Every dot represents the average TPR for the data in Figure 4.6 and errorbars are variance.

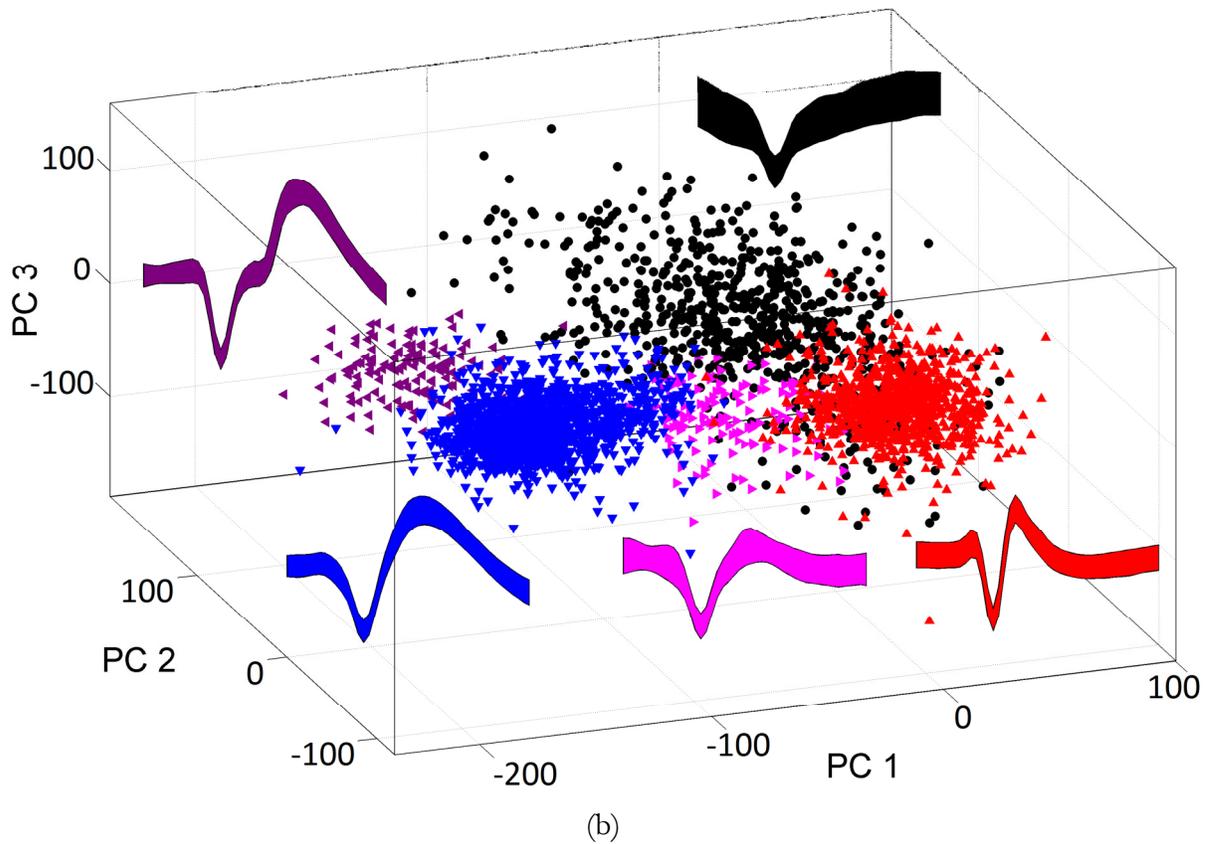
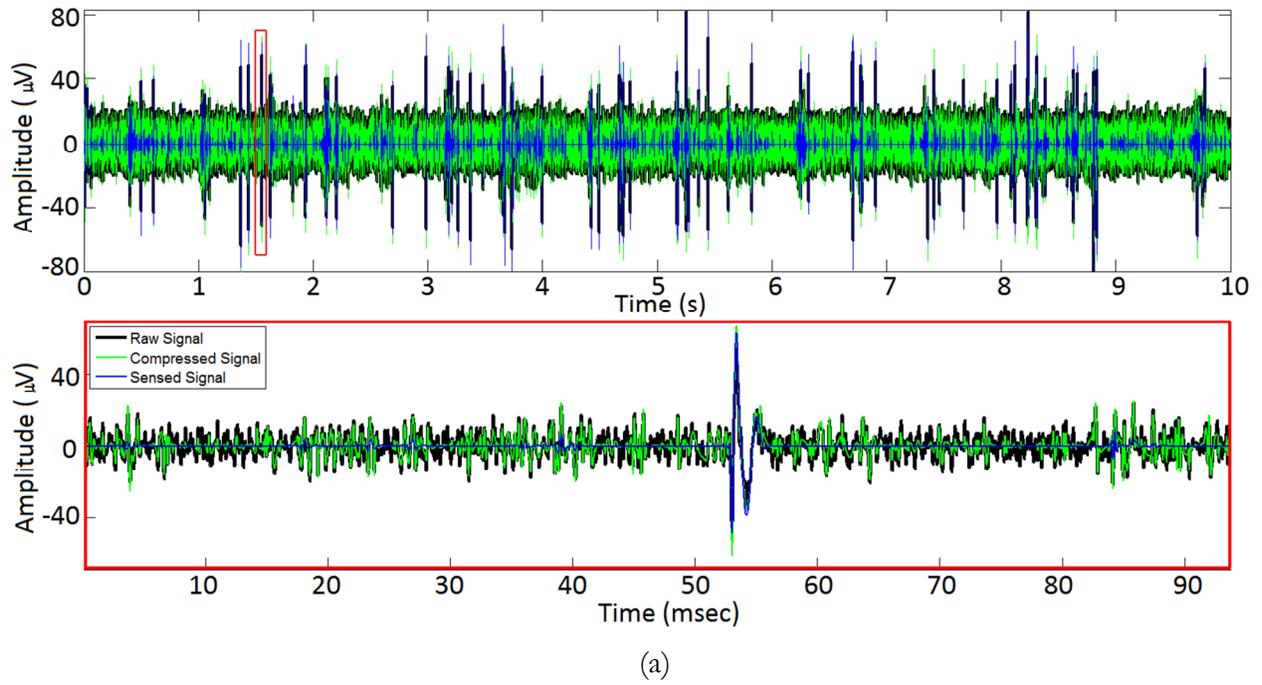
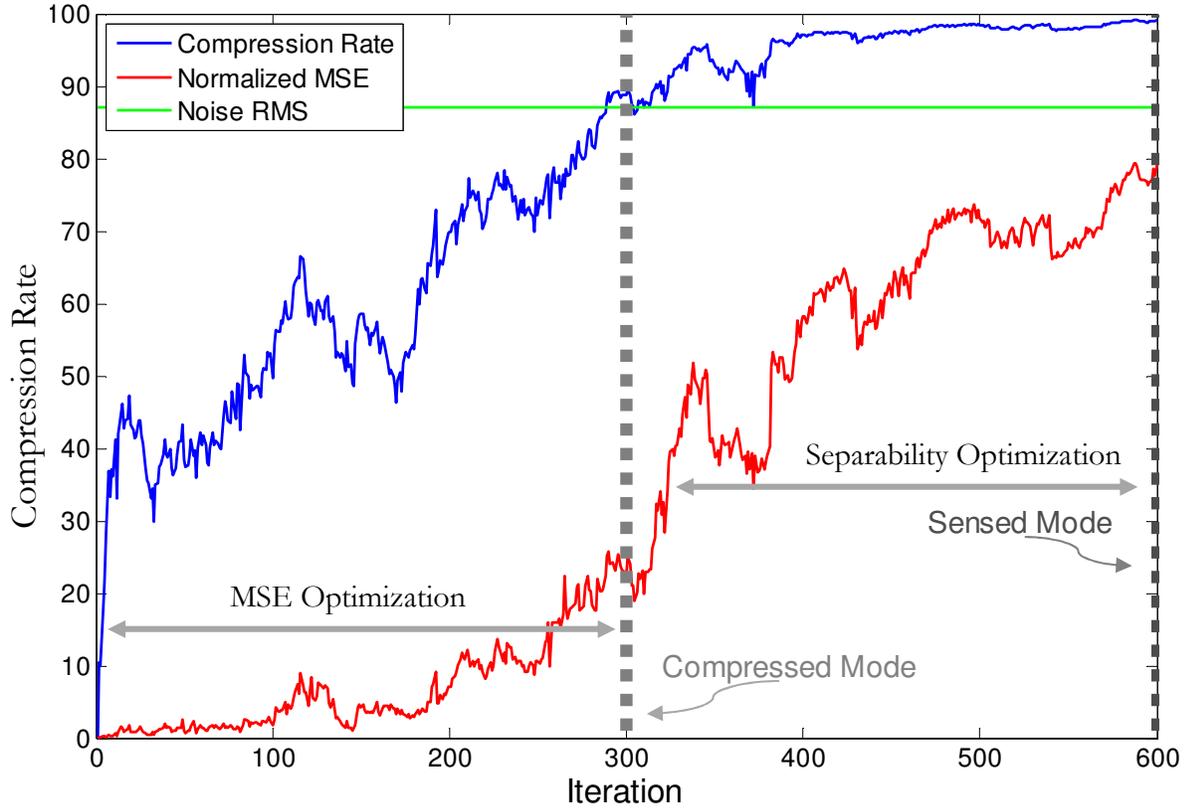


Figure 4.9 (a) A 10 second demonstration of in vivo recording from the rat's somatosensory cortex. The second row is a magnification of the red box illustrated in the first row. The black line

represents the raw neural recording, the green line represents the reconstruction in the compressed mode, and the blue line represents the reconstruction in the sensed mode. (b) Demonstration of all detected events in a 3D space obtained using PCA. Each dot represents the projection of a detected event in the 3D space. Dots are color coded to demonstrate 5 clusters that were obtained using the spike sorting algorithm explained in methods.

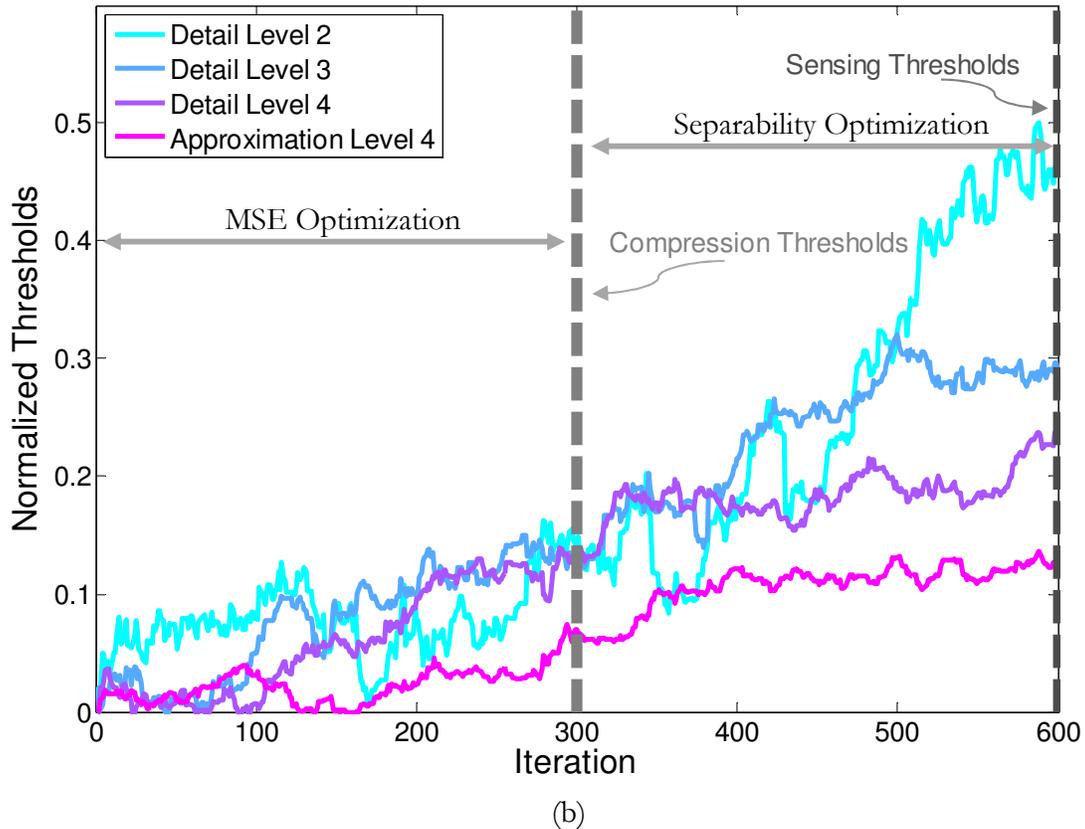
Figure 4.11a demonstrates the rate-distortion tradeoff obtained using random sets of thresholds drawn from a uniform distribution. The learning of the thresholds for the compressed and sensed modes is demonstrated by the trace approximating the lower bound of the rate-distortion tradeoff. While the compression mode threshold is the closest to the origin, the sensed mode threshold achieves the highest compression (i.e., lowest rate) at highest distortion. Although the MSE has increased significantly as a result of the learning phase in the compressed and sensed modes, the degree of separability in both modes remained qualitatively comparable to the raw data separability, as demonstrated in Figure 4.11b. It also demonstrates quantitatively the changes in MSE and rate as well as the degree of separability as the training proceeds for the compressed and sensed modes.



(a)

Figure 4.10 (a) The compression rate and MSE for the iterative simulated annealing learning. The first 300 iterations correspond to finding the optimal thresholds for the compressed mode in which the MSE is minimized. For the compressed mode, a compression rate of 90% was reached. In the next 300 iterations, the optimal thresholds for the sensed mode were estimated using the iterative learning, starting from where the compressed mode stopped. After another 300 trials, a compression rate of 99% was achieved. (b) Threshold variations as learning progresses for the compressed and sensed modes.

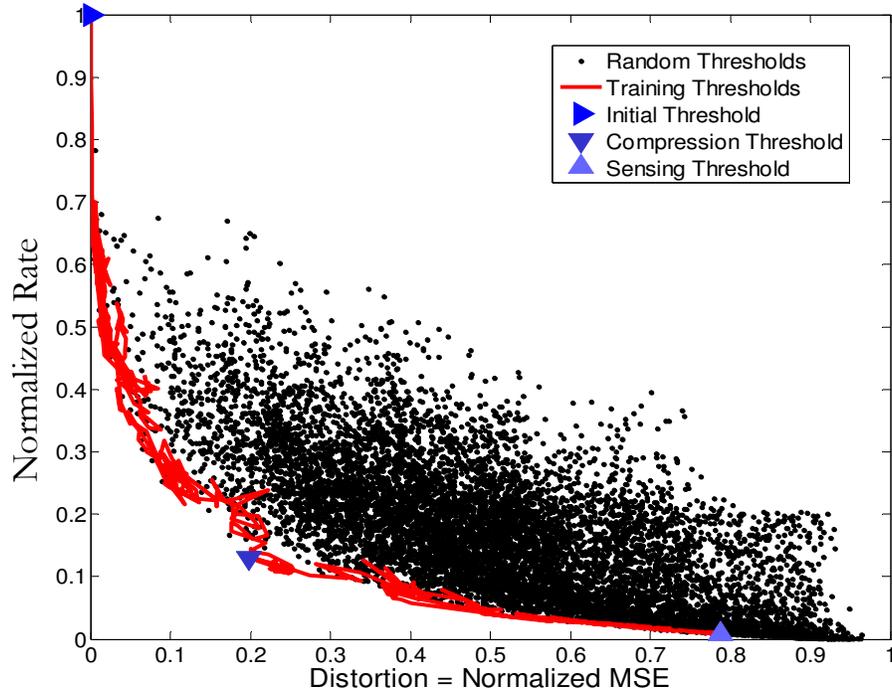
Figure 4.10 (continued)



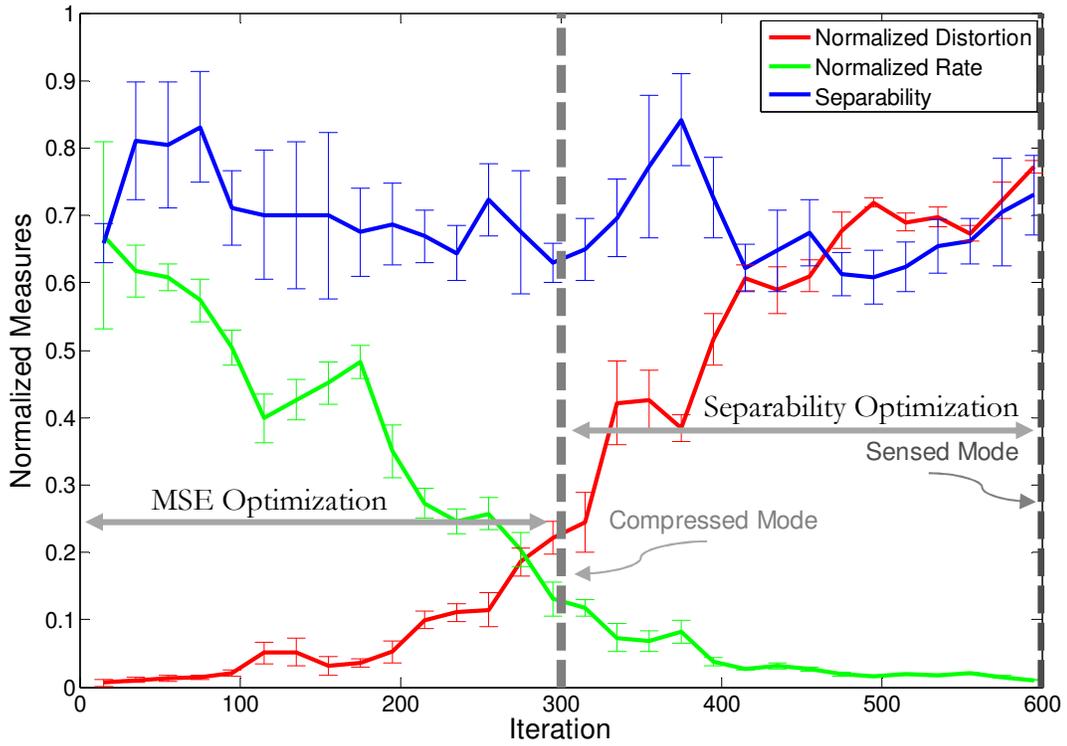
We have compared the sorting performance of our thresholding algorithm, which we refer to as the compressive spike sorting (CSS), with PCA based algorithms for recording sessions of at least an hour long. For each session, we initially obtained the number of neurons and their firing times offline using [21]. Then we learned the optimal thresholds (Table 4.1b) and classifiers for the both CSS and PCA spike sorting algorithms. In the runtime, we had both algorithms classify the detected events that pass a universal threshold as one of the given neurons. Figure 4.12a demonstrates and compares the sorting performance of the CSS versus the PCA algorithm for a  $m$  day span, where initial learning happened at a certain day and the runtime happened  $m - 1$  days afterward on the same electrode. So for a 1 day span, the initial learning and runtime happen at the same day, while there is a day difference for the 2 day span. Figure 4.12a demonstrates the average and standard deviation sorting performance for the total number of 212 neurons detected over the 16 electrodes

and 11 days for the 1 day span. For the 2 and 3 day spans, we only report on the neurons that have been tracked over 2 or 3 consecutive days, which are 15 and 2 cases, respectively. While the sorting performance of the PCA and CSS algorithms was very close for the 1 day span, the sorting performance of the CSS algorithm is statistically significant to the PCA algorithm ( $p < 0.001$ ). Thus, the CSS algorithm could potentially be used for faithfully tracking neurons over consecutive days of recording. Note that the higher sorting performance for the 3 day span compared to the 1 day span is due to typically tracking neurons that demonstrate higher SNRs in their spike waveforms, where the average sorting performance for 1 day span is affected by low SNR neurons that are not tracked over consecutive days.

We have demonstrated the performance of compressive sorting to be comparable and in some cases superior to the conventional spike sorting algorithms that rely on PCA features. Although, PCA features inherently introduce a significant compression rate on the raw neural recordings, through thresholding, CSS provides additional compression due to the elimination of a large number of wavelet coefficients. Figure 4.12a demonstrates a schematic diagram comparing the compression by the two methods. Using  $n$  principal components, the compression rate of PCA is  $(L - n)/L$ , where  $L$  is the number of samples to faithfully represent spike waveforms as the signal of interest. Assuming a  $n$ -level wavelet decomposition and holding on to the largest coefficients in magnitude for the detail levels of 2 to  $n$  an approximation level  $n$  ( $n$  coefficients), we will obtain a similar compression rate. Thresholding, however, to obtain a sparse representation will set a large portion of these coefficients to zero. Using run length encoding (RLE), an additional compression rate can be obtained, that has been reported as the compression rate of the CSS algorithm.



(a)



(b)

Figure 4.11 (a) Rate-distortion scatter plot. Each dot represents the rate and the distortion for a random threshold drawn from a uniform distribution. The red trace demonstrates the same rate-

distortion for the threshold variations for the training in the compressed and sensed modes. The compression thresholds are in close proximity to the origin, while the sensing thresholds are at extremely low rates. The normalized MSE or distortion is estimated as the  $(E_{Raw} - E_{Recon})/E_{Raw}$ , where  $E_{Raw}$  is the root mean square (RMS) of the raw recordings, and  $E_{Recon}$  is the RMS of the reconstruction signal. The normalized rate is estimated as the number of zero coefficients over total number of coefficients. (b) Rate, distortion and the degree of separability as learning progresses through the 600 iterations. The separability was calculated as the normalized inverse of classification errors. The error bars demonstrate the standard deviation.

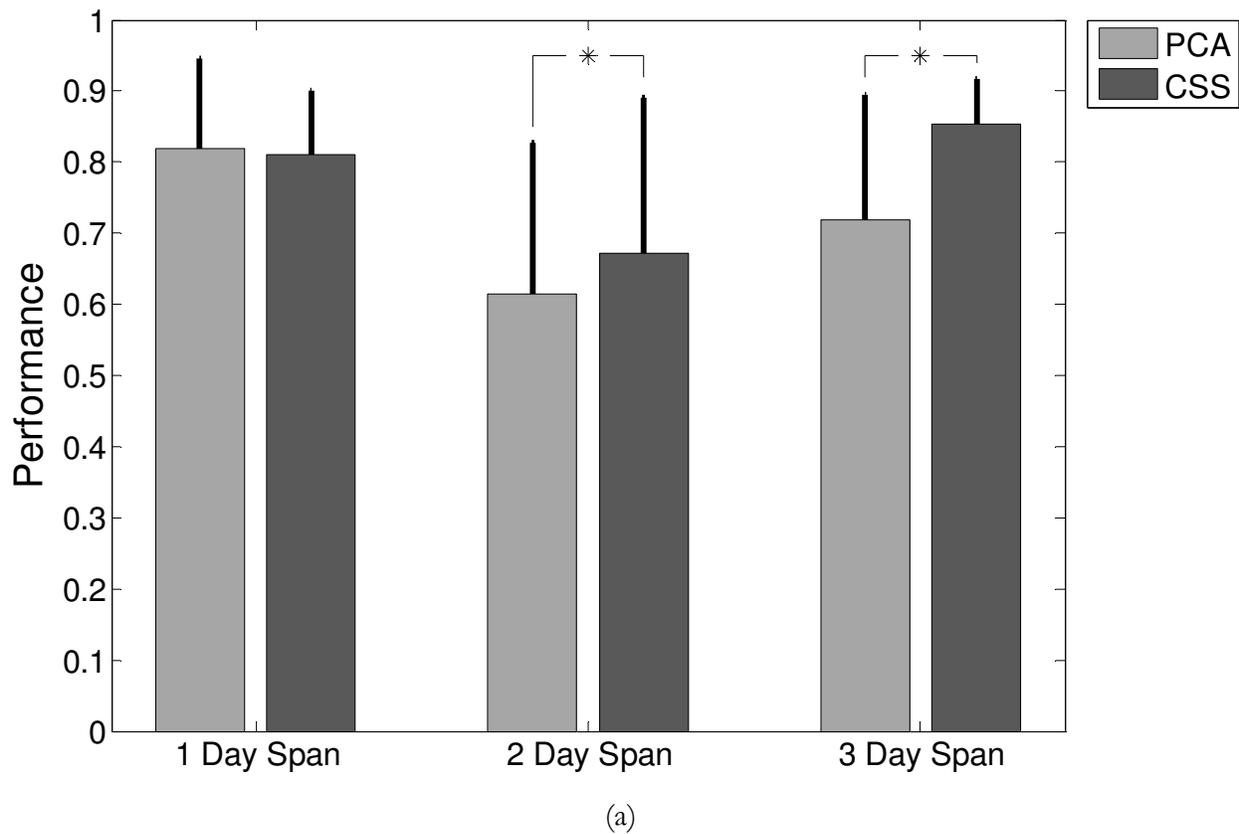
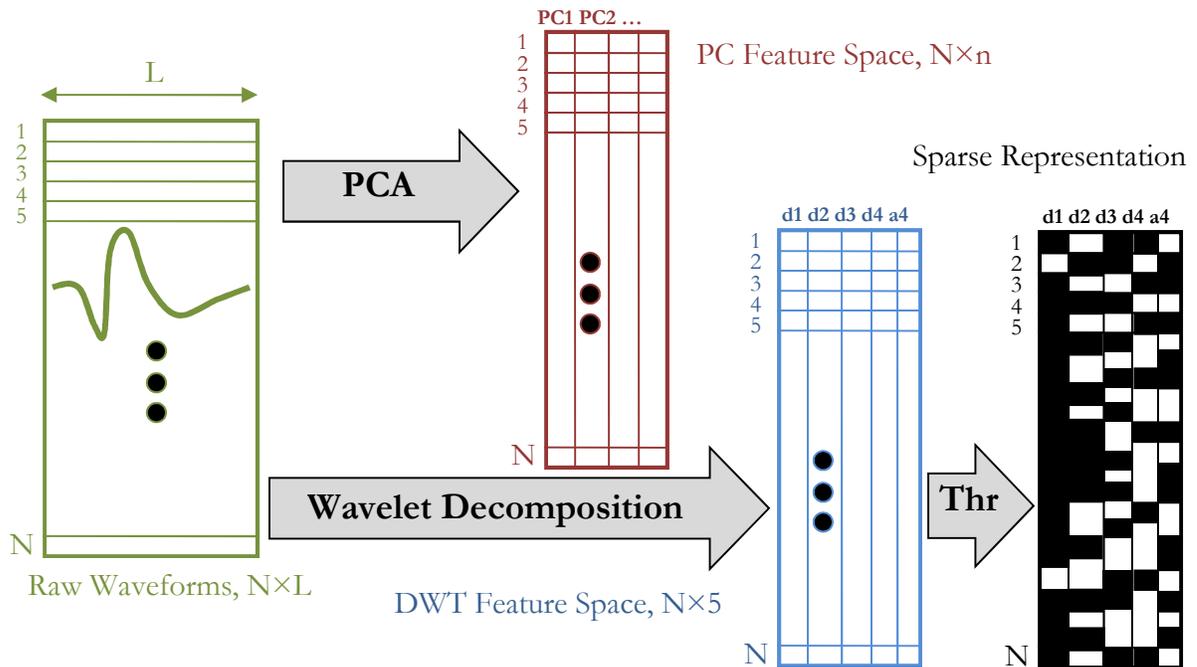


Figure 4.12 (a) Spike sorting performance of the compressive spike sorting (CSS) vs. the conventional PCA sorting for neurons tracked over 1, 2 and 3 days span. The bars demonstrate the average sorting performance and errorbars are the standard deviation. The performance of the CSS

is statistically significant compared to the PCA algorithm for neurons tracked over a span of 2 or 3 days ( $p < 0.001$ ). (b) Schematic diagram for comparing the compression of spike waveforms using PCA vs. sparse representation. The size of the green matrix on left that includes all detected events, i.e. spike waveforms and noise that pass the detection threshold, is  $N \times L$ , where  $N$  is the number of detected events and  $L$  is the number of samples in each event. Using the  $n$  principal components, events are projected into a  $n$ -dimensional space, therefore the projected matrix is  $N \times n$ , and the compression rate is  $(N - n)/N$ . In sparse representation using a 4-level wavelet decomposition, each event is projected into a 5D space, given that only the maximum coefficient survives at each decomposition level, thus introducing a  $(N - 5)/N$  compression rate. Additional compression is obtained by thresholding out a large portion of these coefficients, therefore replacing them with zeros. Run length encoding (RLE) can provide us with an additional  $k/(N \times n)$  compression rate, where  $k$  is the number of zero coefficients.

Figure 4.12 (continued)



(b)

#### 4.4. Discussion

The CSS algorithm directly sorts spike waveforms from the sparse representation without the need to reconstruct the original neural recording, yet provides comparable sorting performances to conventional and commercial spike sorting algorithms that rely on the PCA features. The advantage of mutually compressing and sorting the neural data comes from the adaptive search for optimal set of thresholds for the sparse representation that maintain the sorting performance. This search, however, is performed at two separate modes, compressed and sensing modes. Initially MSE is minimized in the compressed mode to provide an offline spike sorting algorithm with a faithful reconstruction of the neural recording that is used to estimate the number of neurons and their firing times. Given these information, a new set of thresholds are learned by maximizing the separability between the neurons in the sparse representation. We have demonstrated these two separate stages of learning on the rate-distortion plot in Figure 4.11a, where we have shown that at the expense of higher distortion, much lower rates can be achieved without compromising the sorting performance. This perspective to implantable data acquisition and processing systems can have direct implication for the future generation of BMI in which the information of several thousand neurons has to be transferred instantaneously to an external actuator.

To make sure the compromise made in CSS in not reconstructing the original neural recording does not affect the overall sorting performance in comparison with conventional PCA spike sorting algorithms that heavily rely on a faithful reconstruction, we provided two different set of comparisons. First, we compared the CSS with PCA sorting for different SNRs and different compression rates. Using synthetically generated data, in which the ground truth is known, we have demonstrated the superior performance of the CSS especially for cases of low SNR while providing a compression rate far beyond the compression obtained from the PCA approach. Second, we compared the sorting performance of the CSS to the PCA algorithm in handling non-stationarities

observed in the neural data particularly over long-term recordings. We have shown using multi-electrode multiday recordings that the CSS is more robust to the variabilities in the spike waveforms, especially as the span of recordings increase from an hour to over multiple days. Unlike the PCA algorithm that is sensitive to variations in the energy of the spike waveforms, the CSS takes into account the distribution of this energy over different sub-bands. This provides a valuable advantage for CSS over PCA sorting algorithms in tracking neurons over long-term recordings. For BMI systems, this implies that a neural decoder initiated using stable units provided by the CSS does not require everyday training.

Key to the CSS is the ability to sparsely represent the neural waveforms using a dictionary of basis kernels that preserve the discriminative features of the spike waveforms. The optimal thresholds are estimated using a learning algorithm that minimizes the mean square error while maximizing separability between putative signal classes. We evaluated the performance for variable SNRs, and compared it with classical spike sorting techniques. We also demonstrated the importance of choosing the appropriate basis kernel on the sorting performance as a function of the sampling rate.

It is important to note that the compression here by our approach is measured by the number of zeroed coefficients over the total number of coefficients. Accordingly, more zeroed coefficients post thresholding result in more compression. Note that this definition should not to be confused with the traditional definition of compression from information theory in which the compression rate is measured in bits using entropy coding techniques [62]. In our case, we use run length encoding - a variant of entropy coding for binary sequences - to estimate the compressed output stream for the implantable BMI configuration (Figure 4.6) [63].

It's noteworthy that the compression in our approach closely resembles the idea in the compressed sensing framework [64, 65]. Since the extracellular unit activity is transient in nature, compressed sensing can also be used to compress the signal by using a dictionary of orthogonal random samplers [66]. It is important, however, to note that we have replaced the random sampler matrix with a known dictionary extracted from multilevel wavelet decomposition based on the prior information we have about the general form of spike waveforms. The wavelet dictionary has also the advantage of requiring less memory to be implemented in an implantable BMI system compared to a random matrix. Changing the L0 optimization in equation (4-7) of Chapter 4 to an L1 optimization had minimal effect on the remainder of the CSS algorithm performance since the optimization is eventually approximated by maximizing the product of thresholds in equation (4-9).

The simple, yet efficient CSS algorithm based on simple thresholding has its own tradeoffs. First, as mentioned before, it provides a poor reconstruction of the raw neural recording obtained from the sparse representation in the sensed mode. Second, like many other spike sorting algorithms, the CSS also depends on an offline spike sorter to provide it with the number of neurons and their firing times that is used to learn the optimal set of thresholds.

An ensemble classifier was used to fuse the classification results from multiple binary classifiers to improve the overall performance and approximate the typical nonlinear boundary between classes with multiple linear classifiers. There are some theoretical and practical reasons to select an ensemble classifier. First, sophisticated classifiers that can detect the nonlinear boundaries between spike clusters are required. By integrating the classification result of multiple simple thresholding blocks, the performance becomes comparable to nonlinear classifiers. Second, spike sorting becomes challenging at low SNRs, eventually resulting in a more complex nonlinear decision boundary between classes. A combination of binary classifiers, however, provides a solution that can be used to approximate the nonlinear boundary. Third, since the coefficients obtained from the sparse

decomposition represent different bands of the frequency spectrum, each thresholding block will extract different information, which can be integrated in the ensemble classifier to increase the performance. Fourth, due to hardware limitations, simple binary classifiers, such as thresholds, are highly desirable.

Our approach identifies the independent neural channels in the raw ensemble recordings while maintaining low computational complexity, satisfying the necessary power and telemetry bandwidth constraints of a fully implantable hardware. Unlike state-of-the-art systems, this approach reduces the continued reliance on an external computing site to monitor changes in signal quality over prolonged recording periods – a problem often encountered in BMI applications that rely on long term use of implantable recording electrodes. This translates into substantial savings in the computational, communication and human supervision costs of adjusting parameters for the implanted system. Consequently, we anticipate this will lead to further improvements in the performance and potential use of these systems in clinical applications.

# 5

## Spatiotemporal Graphical Models for Ensemble Spike Train Analysis

Spike trains are known to carry important information about the response of cortical neurons to external stimuli that mediate perception, learning and motor processing. Spike trains, simultaneously recorded from neurons within a local population, exhibit variable degrees of statistical dependency. A large number of studies have attempted to identify the role of this dependency and whether it improves the encoding of a jointly represented stimulus attribute [67-71], or it simply expresses redundancy to that stimulus [59, 72-75]. To assess the optimality of any population code, the first step is to model this dependency in the population response.

For a better understanding of the exact role of statistical dependence between the spike trains of cortical neurons, one has to capture this dependence by fitting the spatiotemporal patterns of neural activity with a statistical model. Maximum entropy models have been suggested to fit the simultaneously observed patterns of neuronal spike trains, although they predict the probability of instantaneous patterns and thereby ignoring the temporal correlations in the response. Using higher order Markov processes to represent the sequence of instantaneous patterns, we introduce a spatiotemporal maximum entropy model that takes into consideration of the temporal correlations

as well as the spatial ones. We provide an iterative method for learning the parameters of such model, in which the parameters are truncated to obtain sparse graphs that represent the most important interactions in a population network.

## 5.1. Pairwise Maximum Entropy Models

Let us assume a population of  $N$ -simultaneously observed neuronal spike trains, in which an population vector,  $X^t$ , represents the number of spikes observed from these spike trains at time  $t$  in response to an external stimulus,  $s$ . The population vector has  $N$  elements as  $X^t = [X_1^t, \dots, X_N^t]$ , where  $X_n^t$  is the number of spikes fired by neuron  $n$  within time interval  $t$ . Let us also assume binary spike trains, where a zero represents an observation of no spikes in the observed time interval, and one otherwise. In response to some constantly variable stimulus  $s$  within  $T$  time bins,  $s^{1:T}$ , we observe a series of nonstationary population vectors, expressed as  $X^{1:T} = [X^1, \dots, X^T]$ . To fully characterize the nonstationary encoding system one has to initially describe the conditional distribution,  $P(X^{1:T} | s^{1:T})$ . The main challenge in estimating this distribution is the sample size. For a population of  $N$  neurons, the maximum number of different binary population vectors is  $2^N$ , though the actual number is usually smaller.

The conditional distribution,  $P(X^{1:T} | s^{1:T})$ , tries to explain the dynamics of population responses given a particular stimulus, hence is used to explain the causal relationships, or effective connectivity, between the variables that can greatly improve our understanding of the encoding mechanism. However, other methods with simplifying assumptions have also been used to explain some of the variability observed in population responses. One is independently estimating the conditional distribution of neurons, also referred to as the tuning function, as a direct yet simple

method to understand the encoding mechanism for the entire population of neurons. The tuning function expresses the average firing rate of neuron  $n$  in response to different values of  $\mathbf{s}$ , and is expressed as  $P(X_n|\mathbf{s})$ . Under the independence assumption, the joint conditional distribution can be expressed as the normalized product of individual tuning functions,  $P_{ind}(X|\mathbf{s}) = \prod_{n=1}^N P(X_n|\mathbf{s})/Z$ , where  $Z$  is the normalization factor or partition function estimated as  $Z = \sum_{\mathbf{X}} \prod_{n=1}^N P(X_n|\mathbf{s})$ . The independence assumption can be also extended for different time intervals, therefore, the conditional distribution can be expressed as  $P(X^{1:T}|\mathbf{s}^{1:T}) = \prod_{t=1}^T P_{ind}(X^t|\mathbf{s}^t)/\dot{Z}$ , where  $\dot{Z}$  is the product of all  $T$  normalization factors. This representation assumes independence between all neuronal constituents at different times, thus ignores both the spatial and temporal correlations among neuronal constituents. Although the independent model is less sensitive to the sample size problem and is less affected by the increase in population size [76], more complex models are required to capture the dynamics of nonstationary population vectors.

By including the statistical dependency among neuronal pairs, a pairwise model can be constructed to express the conditional distribution as,  $P_{pair}(X|\mathbf{s}) = \prod_{n=1}^N p(X_n, X_{\pi(n)}|\mathbf{s})/Z$ , where  $\pi(n)$  is the set of neurons connected to neuron  $n$  and  $Z$  is the partition function. Note that the connection here does not necessarily indicate an anatomical connection between neurons, but rather a statistical dependence on each other's neural activity. To extend the pairwise model for population vectors of different time intervals,  $t = 1, \dots, T$ , the set  $\pi(n)$  can include any neuronal constituents of time intervals up to time  $t$  that affect the activity of neuron  $n$ . The conditional distribution for the pairwise model can be expressed as  $P_{pair}(X^{1:T}|\mathbf{s}^{1:T}) = \prod_{t=1}^T P_{pair}(X^t, X_{\pi(n)}|\mathbf{s}^t)/\dot{Z}$ . This model not only explains the instantaneous dependencies within a time interval, but also explains dependencies between neuronal constituents at different time intervals, thus including both the spatial and temporal correlations. In this chapter, we will address the question of finding the set of

parent neurons,  $\pi(n)$ , and building a spatiotemporal pairwise model that incorporates both spatial and temporal correlations. Out of the many approaches that can be used for fitting a conditional distribution, we used maximum entropy models to provide us the models with maximum uncertainty.

### 5.1.1. Principle of Maximum Entropy

Using the principle of maximum entropy, we will construct a pairwise model that matches the expectation of certain moments of neuronal constituents,  $f_\alpha(X): X \rightarrow \mathbb{R}$ , with analogous empirical estimates obtained from the true conditional distribution of observed population vectors,  $P(X|s)$  [77]. The empirical expectations of those certain moments is estimated as,

$$E_P[f_\alpha] = \frac{1}{T} \sum_{t=1}^T f_\alpha(X^t) \tag{5-1}$$

where here  $T$  is the total number of observed population vectors. Now let's assume an estimation of the conditional distribution,  $P_\theta(X|s)$ , where  $\theta$  are some unknown parameters of this distribution. The expectation of the certain moments can be estimated given  $P_\theta(X|s)$  as

$$E_\theta[f_\alpha] = \sum_X P(s) P_\theta(X|s) f_\alpha(X, s) \tag{5-2}$$

where  $P(s)$  is the true distribution of the stimulus. The goal is to estimate  $P_\theta(X|s)$  such that the model and empirical expectations match,  $E_\theta[f_\alpha] = E_P[f_\alpha]$ . This problem, however, is underdetermined, meaning that there are infinite number of distributions that conform to this condition. To choose among the many combinations, the principle of maximum entropy selects the

one that maximizes the Shannon's measure of entropy,  $H(X) = -\sum_X P(X) \log P(X)$ , as

$$P^* = \max_{P \in \mathcal{P}} H(P) \quad \text{subject to} \quad E_\theta[f_\alpha] = E_P[f_\alpha] \quad (5-3)$$

where  $\mathcal{P}$  is set of all distributions that satisfy the condition. The solution to this determined problem represents the distribution with the maximum uncertainty. It has be shown that all maximum entropy distributions conform to the family of exponential functions, such as Gaussian, Poisson, and binomial distributions, and are expressed as  $P_\theta(X|s) = \exp(\sum_\alpha \theta_\alpha f_\alpha(X))/Z$ , where  $\theta_\alpha$  is the parameter associated with  $f_\alpha(X)$ .

### 5.1.2. Instantaneous Maximum Entropy Model

If we define the certain moments  $f_\alpha(X)$  as the first order moments,  $X_i^t$ , and second order moments,  $X_i^t X_j^t$ , within the same time interval  $t$ , the maximum entropy model can be expressed as

$$P_\theta(X^t|s) = \frac{1}{Z} \exp\left(-\sum_{i=1}^N h_i X_i^t - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N J_{ij} X_i^t X_j^t\right) \quad (5-4)$$

where  $h_i$  and  $J_{ij}$  are the parameters associated with the first and second order moments, respectively, and  $Z$  is the partition function to ensure  $\sum_X P_\theta(X^t|s) = 1$ . This model is referred to as the instantaneous maximum entropy model, since  $X^t$  is the pattern of activity only at time  $t$ . The conditional distribution of a sequence of population vectors for the instantaneous model can be expressed as the product of individual instantaneous models at different times,  $P_\theta(X^{1:T}) = \prod_{t=1}^T P_\theta(X^t)/Z$ . This model was shown to be effective in representing almost 90 percent of the variance seen in the response of a population less than 20 neurons [78, 79]. The main disadvantage

of this model, however, is that it ignores the temporal correlation that represents the effect of the history of spiking activity on the current population state. Extending the width of neural patterns has been proposed as a method to incorporate spiking history. It destroys, however, the information in precise spike timings [80]. Thus, the instantaneous maximum entropy model does not consider interactions between neurons at different times that are believed to play an important role in the encoding mechanism.

### 5.1.3. Spatiotemporal Maximum Entropy Model

Given the observed population vectors up to time  $t$ ,  $X^{1:t}$ , one can estimate the probability of observing the population vector of time  $t + 1$  as  $P(X^{t+1}|X^{1:t})$ . Assuming a  $k^{th}$ -order Markov model, the conditional probability can be reduced to  $P(X^{t+1}|X^{t-k:t})$ . Assuming a  $k^{th}$ -order Markov model, the joint density of a sequence of population vectors can be expressed as

$$P(X^{1:T}) = P(X^1) \prod_{t=2}^T P(X^t|X^{t-k:t-1}) \quad (5-5)$$

For a first-order Markov model, the transitional probabilities can be expressed as  $P_{nm} = P(X^t = X^{(n)}|X^{t-1} = X^{(m)})$ , where  $X^{(n)}$  and  $X^{(m)}$  are two sample population vectors. The transitional probabilities governing the transition between population vectors in a neural recording are nonsymmetrical,  $P_{nm} \neq P_{mn}$ . To better demonstrate this, we constructed a toy example in which a population network generates one out three possible population vectors,  $X^{(1)}$ ,  $X^{(2)}$ , and  $X^{(3)}$ . Using a first-order Markov model, we illustrate in Figure 5.1a the transitional probabilities that govern the transition between the three population vectors. Let's assume that the transitional probabilities are stationary. Given a particular sequence of population vectors, i.e.

$\{X^{(1)}, X^{(2)}, X^{(2)}, X^{(3)}\}$ , the probability of observing this sequence can be estimated as  $P(X^{(1)})P_{21}P_{22}P_{32}$ , where  $P(X^{(1)})$  is the a priori of observing  $X^{(1)}$ .

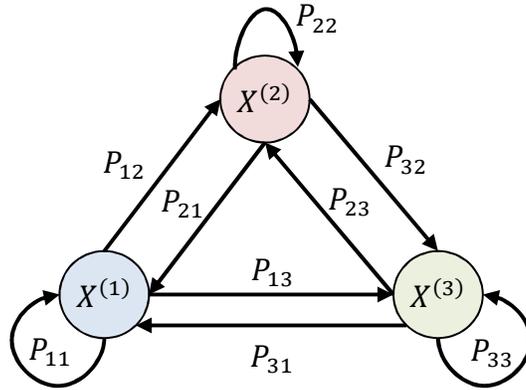
By replacing the pairwise maximum entropy model in (5-4) in the transitional probabilities of a first-order Markov model in (5-5), we can express these probabilities based on the neuronal constituents as

$$P_{\theta}(X^t|X^{t-1}) = \frac{1}{Z} \prod_{i=1}^N \prod_{j=1}^N \exp(J_{ij}^{t-1,t} X_j^{t-1} X_i^t) = \frac{1}{Z} \exp\left(\sum_{i=1}^N \sum_{j=1}^N J_{ij}^{t-1,t} X_i^t X_j^{t-1}\right) \quad (5-6)$$

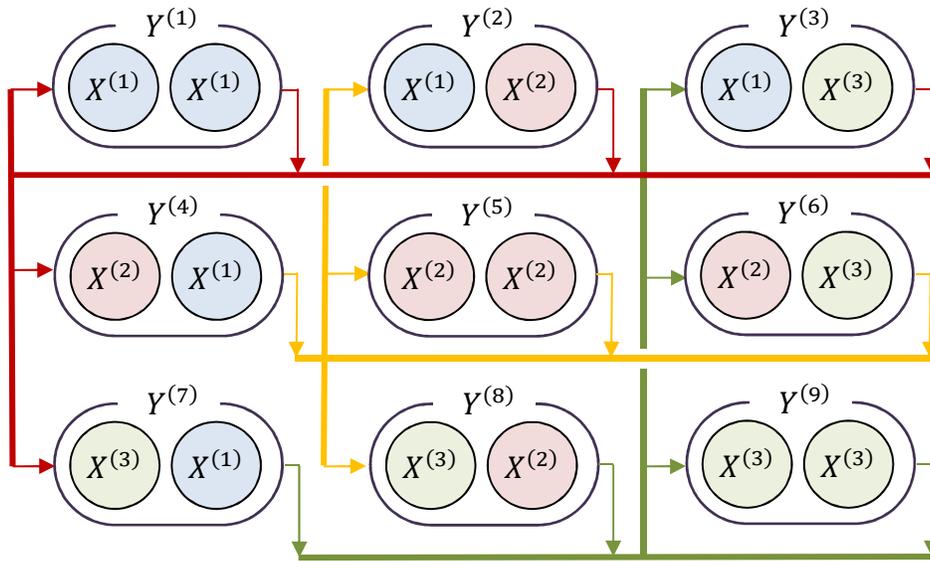
where  $J_{ij}^{t-1,t}$  are the transitional couplings, and  $Z$  is the partition function. By including all elements of the pairwise maximum entropy model, the joint distribution for the first-order Markov model can be expressed as

$$P_{\theta}(X^{t-1}, X^t) = \frac{1}{Z} \exp\left(-\sum_{i=1}^N h_i^{t-1} X_i^{t-1} - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N J_{ij}^{t-1,t-1} X_i^{t-1} X_j^{t-1} - \sum_{i=1}^N \sum_{j=1}^N J_{ij}^{t-1,t} X_i^{t-1} X_j^t - \sum_{i=1}^N h_i^t X_i^t - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N J_{ij}^{t,t} X_i^t X_j^t\right) \quad (5-7)$$

where  $h_i^t$  and  $h_i^{t-1}$  are the individual, and  $J_{ij}^{t,t}$ ,  $J_{ij}^{t,t-1}$  and  $J_{ij}^{t-1,t-1}$  are the coupling terms. This spatiotemporal model, which we refer to as the spatiotemporal maximum entropy model, is more general than the representation introduced in Marre et al to account for history terms [81], but at the expense of increasing the number of parameters to be trained.



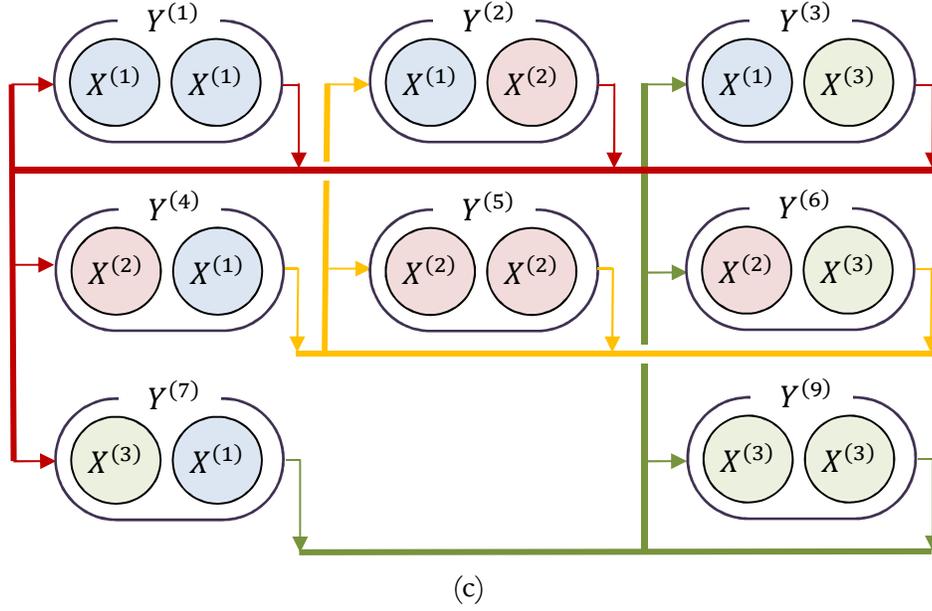
(a)



(b)

Figure 5.1 (a) Representation of hypothetical instantaneous patterns,  $X^{(i)}$  (for  $i = 1, 2, 3$ ), generated by a neural. (b) The first-order Markov representation of state transfers for Fig.1. Each state can either transfer to itself or one of the two other states, thus generating a total of 9 transient states,  $Y^{(k)}$  (for  $k = 1, \dots, 9$ ). (c) Another representation of the transient states,  $Y^{(k)}$ , with  $Y^{(8)}$  being unobserved.

Figure 5.1 (continued)



We simplify (5-7) as  $p_{\theta}(Y^t) = Z^{-1} \exp(-\sum_i \hat{h}_i y_i^t - \sum_{i,j} \hat{J}_{ij} y_i^t y_j^t)$ , where  $Y^t = [X^t, X^{t-1}]^T$  includes the  $X^t$  and  $X^{t-1}$  population vectors,  $\hat{h}_i = [h_i^t, h_i^{t-1}]^T$  is the parameters associated with the first order moments,  $\hat{J}_{ij} = [J_{ij}^{t,t}, J_{ij}^{t-1,t}; J_{ij}^{t,t-1}, J_{ij}^{t-1,t-1}]$  is the parameters associated with the second order moments, and  $T$  is transpose. This representation can be easily extracted for higher order Markov models too. Figure 5.2a and 5.2b demonstrates the model parameters  $J_{ij}^{t-1,t-1}$ ,  $J_{ij}^{t,t}$ , and  $J_{ij}^{t-1,t}$  as blue, red and black connecting lines, respectively.

To better understand the necessity of having both  $h_i^{t-1}$  and  $h_i^t$  or  $J_{ij}^{t-1,t-1}$  and  $J_{ij}^{t,t}$ , Figure 5.1b demonstrates all possible transitions between the three hypothetical population vectors,  $X^{(1)}$ ,  $X^{(2)}$ , and  $X^{(3)}$ , as joint patterns  $Y^{(k)}$  for  $k = 1, \dots, 9$ . If we assume that the possibility of observing any  $Y^{(k)}$  is equal, thus,  $h_i^{t-1}$  will be equivalent to  $h_i^t$ . In this configuration, we have assumed that the possibility of transiting to either of patterns  $X^{(1)}$ ,  $X^{(2)}$ , and  $X^{(3)}$  from the current pattern is equal to  $1/3$ , thus making the possibility of observing  $Y^{(k)}$  equal to  $1/9$  for all  $k$ . Now let's assume a

different configuration in Figure 5.1c, in which  $X^{(3)}$  can only transit to  $X^{(1)}$  or  $X^{(3)}$ , thus making the probability of observing  $Y^{(8)}$  equal to 0. In this case,  $h_i^{t-1}$  is different than  $h_i^t$  since it is estimated using three  $X^{(1)}$ , three  $X^{(2)}$  and two  $X^{(3)}$ , while  $h_i^t$  is estimated using three  $X^{(1)}$ , two  $X^{(2)}$  and three  $X^{(3)}$ . A similar explanation can be given for the difference between  $J_{ij}^{t-1,t-1}$  and  $J_{ij}^{t,t}$  for the configuration in Figure 5.1c.

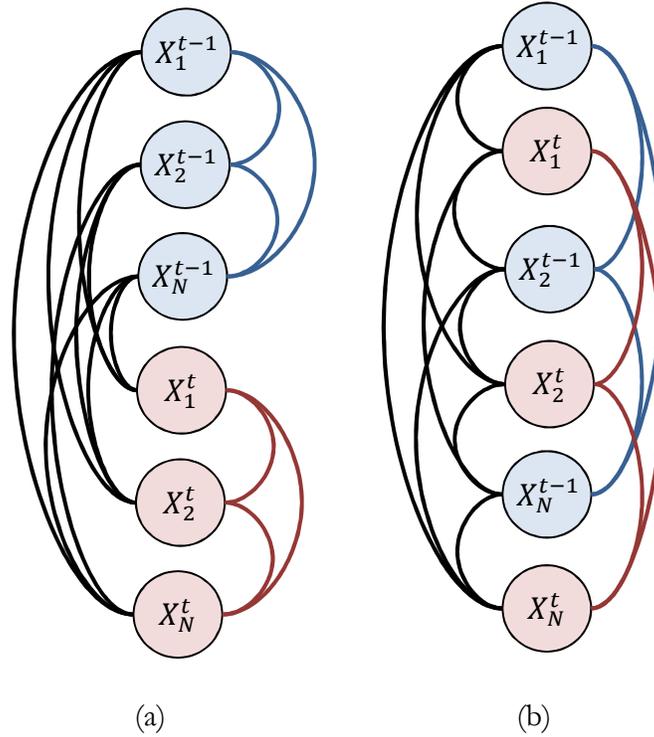


Figure 5.2 First-order Markov representation of pairwise maximum entropy model. The model parameters  $J_{ij}^{t-1,t-1}$ ,  $J_{ij}^{t,t}$ , and  $J_{ij}^{t-1,t}$  are demonstrated as blue, red and black connecting lines, respectively.

The first-order Markov model can be simply extended for higher order Markov representations. For a second-order Markov representation for the maximum entropy model, we define three population vectors jointly as one pattern,  $Z^t = [X^{t-1} \ X^t \ X^{t+1}]$ . The instantaneous maximum

entropy model can therefore be expressed as  $P_\theta(Z^t) = \frac{1}{Z} \exp\left(-\sum \check{h}_i Z_i^t - \frac{1}{2} \sum \sum \check{J}_{ij} Z_i^t Z_j^t\right)$  in which  $\check{h}_i$  and  $\check{J}_{ij}$  are the its parameters and are defined as

$$\check{h}_i = \begin{bmatrix} h_i^{t-1} \\ h_i^t \\ h_i^{t+1} \end{bmatrix} \quad \text{and} \quad \check{J}_{ij} = \begin{bmatrix} J_{ij}^{t-1,t-1} & J_{ij}^{t-1,t} & J_{ij}^{t-1,t+1} \\ J_{ij}^{t-1,t} & J_{ij}^{t,t} & J_{ij}^{t,t+1} \\ J_{ij}^{t-1,t+1} & J_{ij}^{t,t+1} & J_{ij}^{t+1,t+1} \end{bmatrix} \quad (5-8)$$

#### 5.1.4. Training Maximum Entropy Models

Two iterative methods, the gradient descent [82] and the iterative scaling algorithm [79], are commonly used to find the unknown parameters of the spatiotemporal maximum entropy model. Here we will discuss how these methods were obtained in the first place, and introduce a quadratic estimation that learns faster compared to the gradient descent and iterative scaling algorithms. The objective is to estimate the model parameters such that model estimations of the probability of population vectors provide the maximum entropy as

$$\theta^* = \min_{\theta} L(\tilde{p}|\alpha) = \min_{\theta} \sum_t P_\theta(X^t) \log(P(X^t)) \quad (5-9)$$

Where  $L(P_\theta|\theta)$  is the likelihood function. The model parameters are learned using an iterative learning approach, in which the updates for parameter  $k$ ,  $\delta_k$ , are estimated as follows. First, we estimate variations in the likelihood,  $\Delta L$ , given the parameter updates as

$$\Delta L = L(P_\theta|\theta + \delta) - L(P_\theta|\theta) = \sum_t \sum_k P_\theta(X^t) \delta_k f_k(X^t) - \log\left(\frac{\dot{Z}}{Z}\right) \quad (5-10)$$

where  $\dot{Z} = \sum_t \exp(\sum_k (\theta_k + \delta_k) f_k(X^t))$ . By applying the inequality  $-\log(x) \geq 1 - x, \forall x > 0$ , we

will find a lower bound for  $\Delta L$  as

$$\Delta L \geq \sum_t \sum_k P_\theta(X^t) \delta_k f_k(X^t) + 1 - \sum_t P(X^t) \exp\left(\sum_k \delta_k f_k(X^t)\right) \quad (5-11)$$

By applying Jensen's inequality for a probability distribution  $p(x)$ ,  $e^{\sum_x p(x)q(x)} \leq \sum_x p(x)e^{q(x)}$ , and defining  $F(X) = \sum_k |f_k(X)|$ , and decomposing  $f_k(X) = |f_k(X)|s_k(X)$ , where  $s_k(X) = \text{sign}(f_k(X))$ , the lower bound can be written as

$$\begin{aligned} \Delta L \geq \sum_t \sum_k P_\theta(X^t) \delta_k f_k(X^t) + 1 \\ - \sum_t P(X^t) \sum_k \frac{|f_k(X^t)|}{F(X^t)} \exp(\delta_k s_k(X^t) F(X^t)) \end{aligned} \quad (5-12)$$

We can find the optimal parameter updates by solving  $\partial \Delta L / \partial \hat{\theta} = 0$  to have

$$\sum_t P_\theta(X^t) f_k(X^t) = \sum_t P(X^t) f_k(X^t) \exp(\delta_k s_k(X^t) F(X^t)) \quad (5-13)$$

To obtain the iterative scaling algorithm for learning the unknown parameters, we can approximate the right hand side of (5-13) as  $\sum_t P(X^t) f_k(X^t) \times \exp(\delta_k \sum_t s_k(X^t) F(X^t))$ , thus, the updates are estimated as

$$\delta_k = \frac{1}{\sum_t s_k(X^t) F(X^t)} \log\left(\frac{\sum_t P_\theta(X^t) f_k(X^t)}{\sum_t P(X^t) f_k(X^t)}\right) \quad (5-14)$$

To obtain the gradient descent algorithm for learning the unknown parameters, we can approximate the exponential on the right hand side of (5-13) as  $\exp(x) = 1 + x$ , thus, the updates

are estimated as

$$\delta_k = \frac{\sum_t P_\theta(X^t) f_k(X^t) - \sum_t P(X^t) f_k(X^t)}{\sum_t P(X^t) |f_k(X^t)| F(X^t)} \quad (5-15)$$

Here we approximated the exponential on the right hand side of (5-13) as  $\exp(x) = 1 + x + 0.5x^2$ , and thus update the unknown parameters as

$$A = 2 \frac{\sum_t P_\theta(X^t) |f_k(X^t)| F(X^t)}{\sum_t P(X^t) f_k(X^t) F^2(X^t)} \quad (5-16)$$

$$\delta_k = -0.5 \left( A \pm \sqrt{A^2 - 8 \frac{\sum_t P(X^t) f_k(X^t) - \sum_t P_\theta(X^t) f_k(X^t)}{\sum_t P(X^t) f_k(X^t) F^2(X^t)}} \right)$$

which we refer to as the quadratic iterative learning algorithm.

## 5.2. V1 Population Encoding in Response to Stimulus Grating

We evaluated the performance of the different encoding model in predicting the activity pattern of cat V1 neurons in response to drifting grating stimuli, as illustrated in Figure 5.3a [83]. In drifting grating stimuli, a set of dark bars oriented and drifting in one out of 16 possible degrees, from 0 to  $15\pi/8$  degrees, with steps of  $\pi/8$ , were displayed to an anesthetized cat. A stimulus is randomly selected and presented for 2.5 seconds with less than 0.5 seconds of blank screen before presenting the next stimulus. Figure 5.3b demonstrates 9 seconds of an hour long session, in which three different stimuli out of 16 stimuli were randomly presented to the subject. During the blank screen, the spiking activity significantly reduces compared to when one of the stimulus is presented.

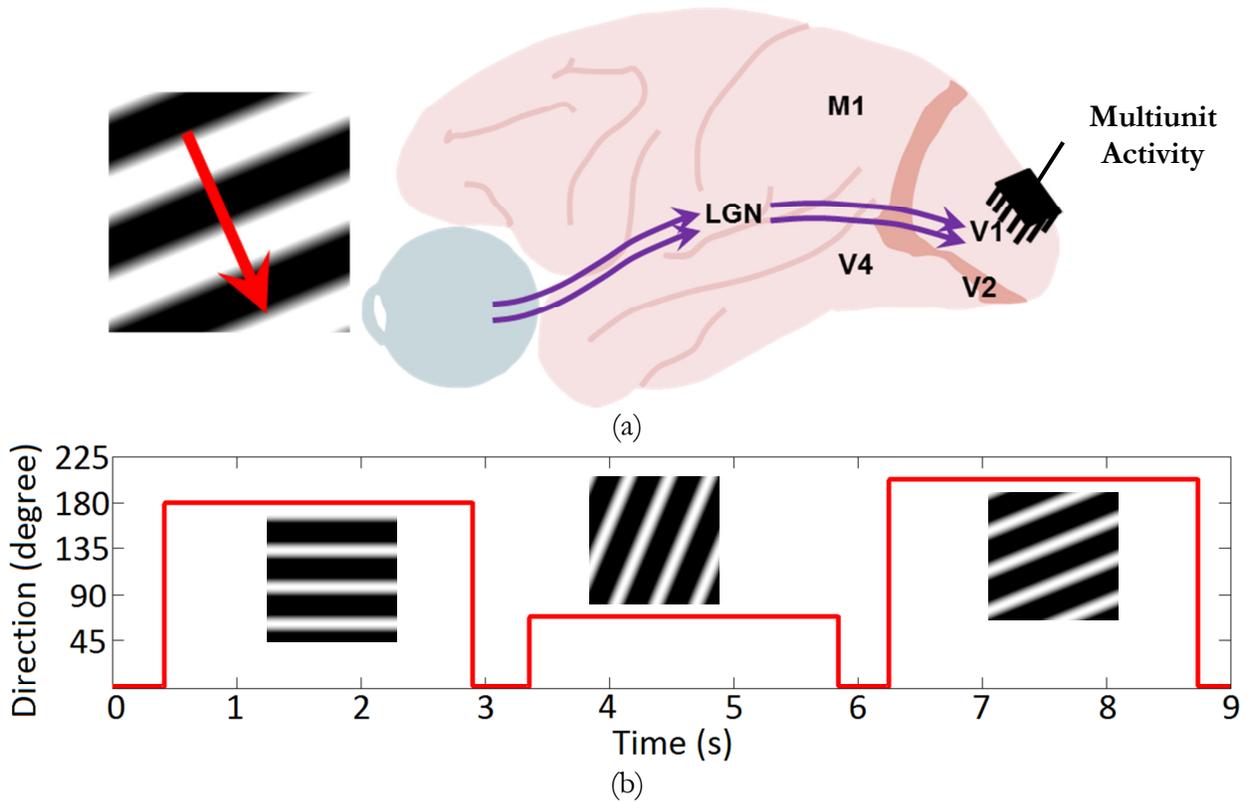
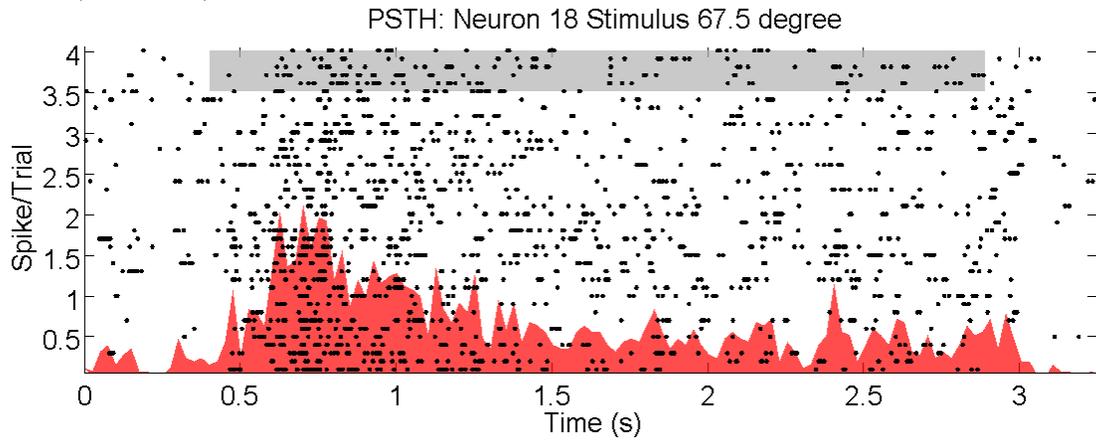
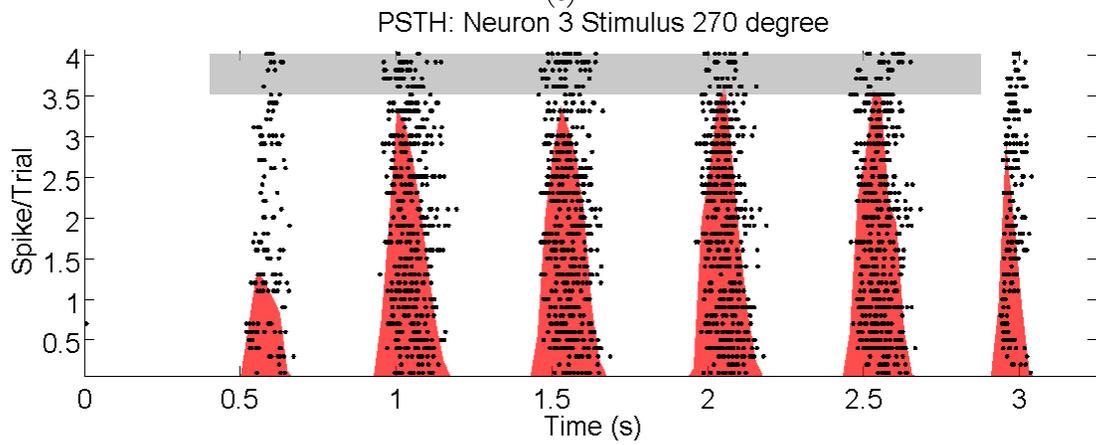


Figure 5.3 (a) Experimental setup: recording from the V1 area of an anesthetized cat in response to drifting grating stimulus with tetrode microwires. The bars in the image presented to the subject moves in forward direction. (b) The sequence of stimuli for an interval of 9 seconds, in which randomly selected stimuli are presented to the subject. Each stimulus is presented for approximately 2.5 seconds, with less than 0.5 seconds of blank screen afterwards. (c) & (d) Single unit activity: the peristimulus time histogram (PSTH) was demonstrated for neurons 18 and 3 for two different stimuli, respectively [84]. Every dotted row demonstrates the spiking activity of a neuron to a trial of the same stimulus. The red plot is the average firing rate of that neuron over all trials and is represented in average spike per second. The gray bar demonstrates when the stimulus was presented. (e) Multiunit activity: demonstrates the spike raster plots for the neural population for 5 seconds. Every row represents the spike train of a neuron, and each bar indicates a spike event.

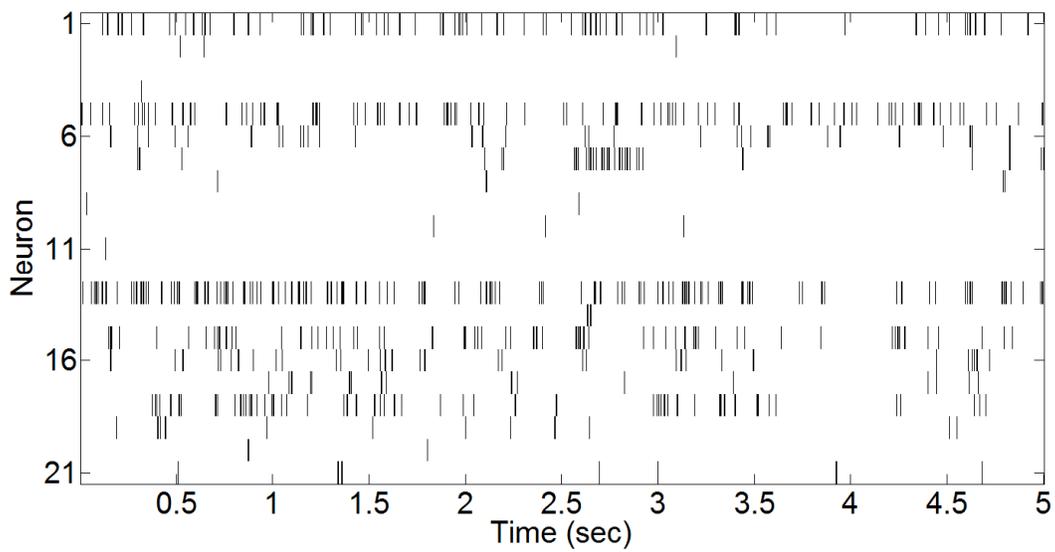
Figure 5.3 (continued)



(c)



(d)



(e)

### 5.2.1. Single Unit Analysis

Using tetrode microwires, a population of 21 neurons was obtained using spike sorting. In a tetrode array, 4 wires are bundled together, thus, the spatial distribution of signal propagation can be used to improve the performance of spike sorting. The spike trains were sampled in 5 msec time intervals, and intervals that contained one or more than one spike were set as '1' and '0' for no spike. Depending on the encoding characteristics of single unit activity, these V1 neurons can be categorized as either movement-selective or orientation-selective neuron. Movement-selective neurons, as demonstrated in Figure 5.3c, are sensitive to the movement direction of the drifting bars. Orientation-selective neurons, on the other hand, respond to the intensity level of the bars, as demonstrated in Figure 5.3d. Unlike the movement-selective neurons, the activity of orientation-selective neurons is periodic and time locked to the frequency of the drifting bars.

Figure 5.4 demonstrates the encoding mechanism of movement-selective and orientation-selective neurons in the V1 using a circular representation of all 16 stimuli. In Figure 5.4a, the preferred movement direction of a movement-selective neuron is  $\bar{\varphi} = 7\pi/8$  degrees. As a result, this neuron fires constantly within the 2.5 seconds interval with a higher average firing rate if the stimulus,  $\varphi$ , is close to  $\bar{\varphi}$ . The tuning function in this case resembles the cosine tuning in motor encoding to preferred velocity, such that the average firing rate is directly proportional to  $\cos(\varphi - \bar{\varphi})$ . As a result, this neuron is responsive to movement directions of  $3\pi/4$  and  $\pi$  degrees. The further  $\varphi$  is from  $\bar{\varphi}$ , the less is the total average firing rate, demonstrated by size of the stimulus icon in center.

The orientation-selective neuron demonstrated in Figure 5.4b, on the other hand, is sensitive to vertical oriented bars, so it will be equally responsive to the bars drifting with  $\bar{\varphi} = 3\pi/2$  or  $\pi/2$

degrees. This neuron generates similar firing rates for stimuli positioned across each other with  $\pi$  degrees difference, thus it will generate higher average firing rates if  $\varphi$  is close to  $\bar{\varphi}$  or  $\bar{\varphi} - \pi$ . The tuning function in this case can be expressed as  $\cos(2(\varphi - \bar{\varphi}))$ . Out of the 21 neurons, 14 were orientation-selective neurons, and 7 were movement-selective neurons. The important question here is whether the multiunit activity of the population of 21 neurons, as demonstrated in Figure 5.3e, can be predicted given only the tuning functions of each neuron, or is it essential to include also the spatial and temporal correlations among neuronal constituents to improve the predictive power.

### 5.2.2. Multiunit Analysis

Although the encoding mechanism of neuron  $n$  can be fully characterized by its tuning function,  $P(X_n|s)$ , its role in the population network is not yet explained. Here, we evaluate and compare the performance of the independent, instantaneous maximum entropy, and spatiotemporal maximum entropy models in explaining the network encoding mechanism. We have looked at the performance of these encoding models from two perspectives. First, we evaluated the predictive power of these models in predicting the observation a particular sequence of population vectors for up to three consecutive vectors. Second, we evaluated the separability of network representations for different stimuli, and how this separability relates to single unit tuning functions.

Figure 5.5 demonstrates the predictive power of the three different encoding models in predicting the probability of observing a sequence of population vectors,  $\{X^{(i)}, X^{(j)}\}$ . The joint distribution,  $P(X^{(i)}, X^{(j)})$ , can be expressed for the independent model as  $P_{ind}(X^{(i)})P_{ind}(X^{(j)})$ , and for the instantaneous maximum entropy model as  $P_{MaxEnt}(X^{(i)})P_{MaxEnt}(X^{(j)})$ , and for the spatiotemporal maximum entropy model as  $P_{MaxEnt}([X^{(i)}, X^{(j)}])$ . The quantile-quantile plot in

Figure 5.5a demonstrates the model estimations versus true empirical probabilities for all observed combinations of  $\{X^{(i)}, X^{(j)}\}$ , each demonstrated as a dot. The encoding model that provides the distribution of dots close to the equality line is better. In this case, the spatiotemporal model was better than the instantaneous model and that was better than the independent model. Figure 5.5b demonstrates the sum of all estimated probabilities with a similar number of '1' in their population vectors, and the closest plot to the empirical plot determines the best model fit.

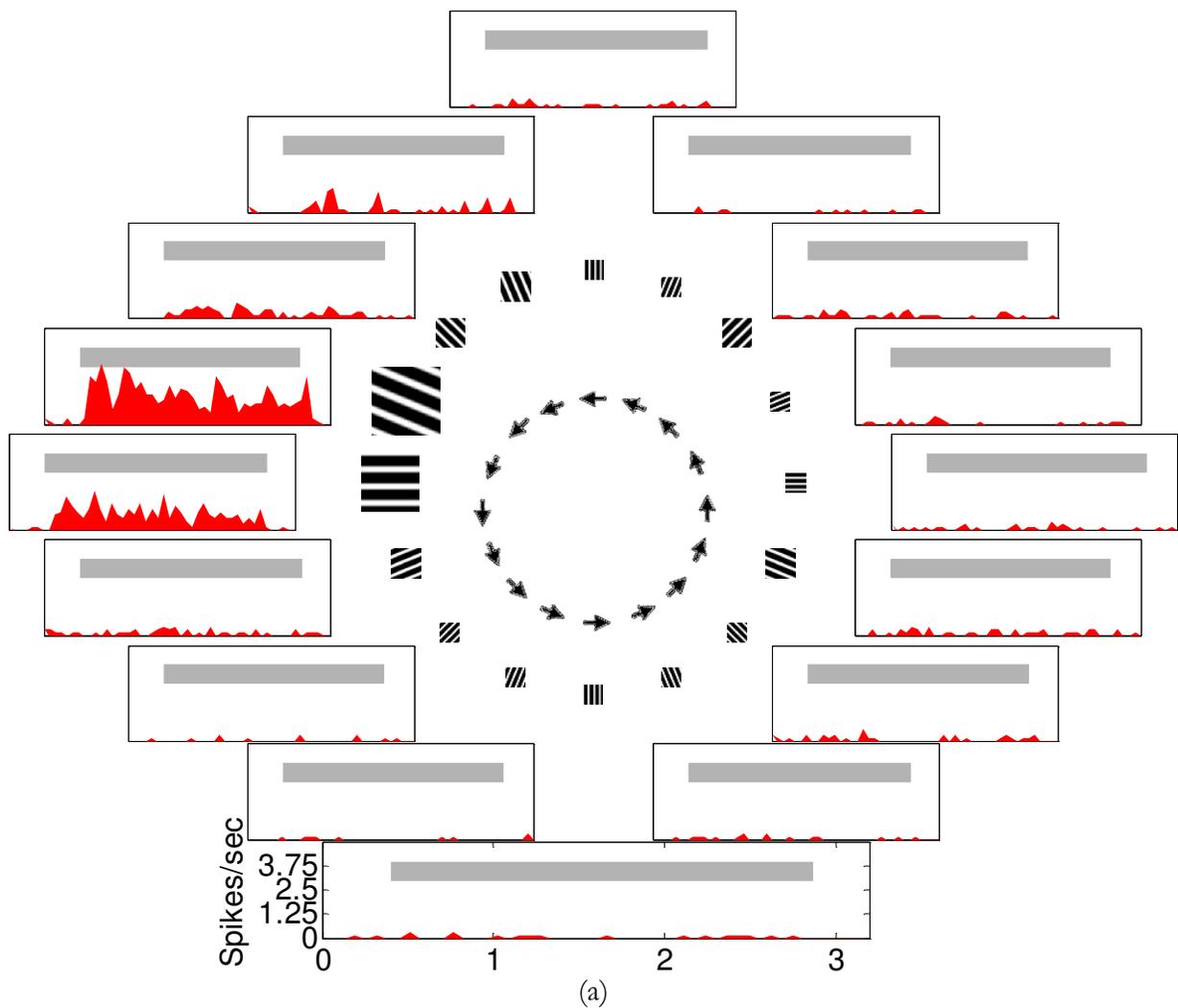
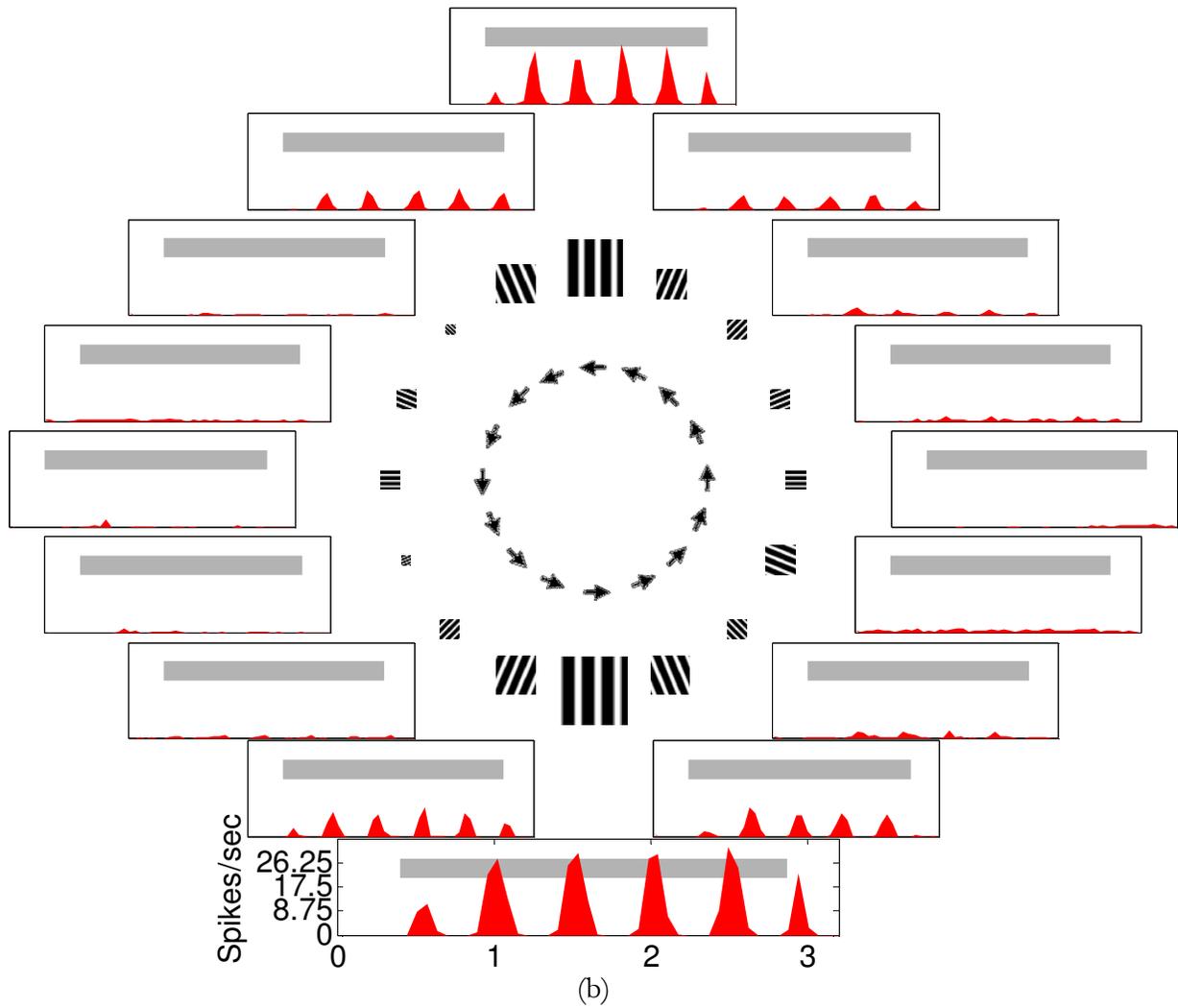
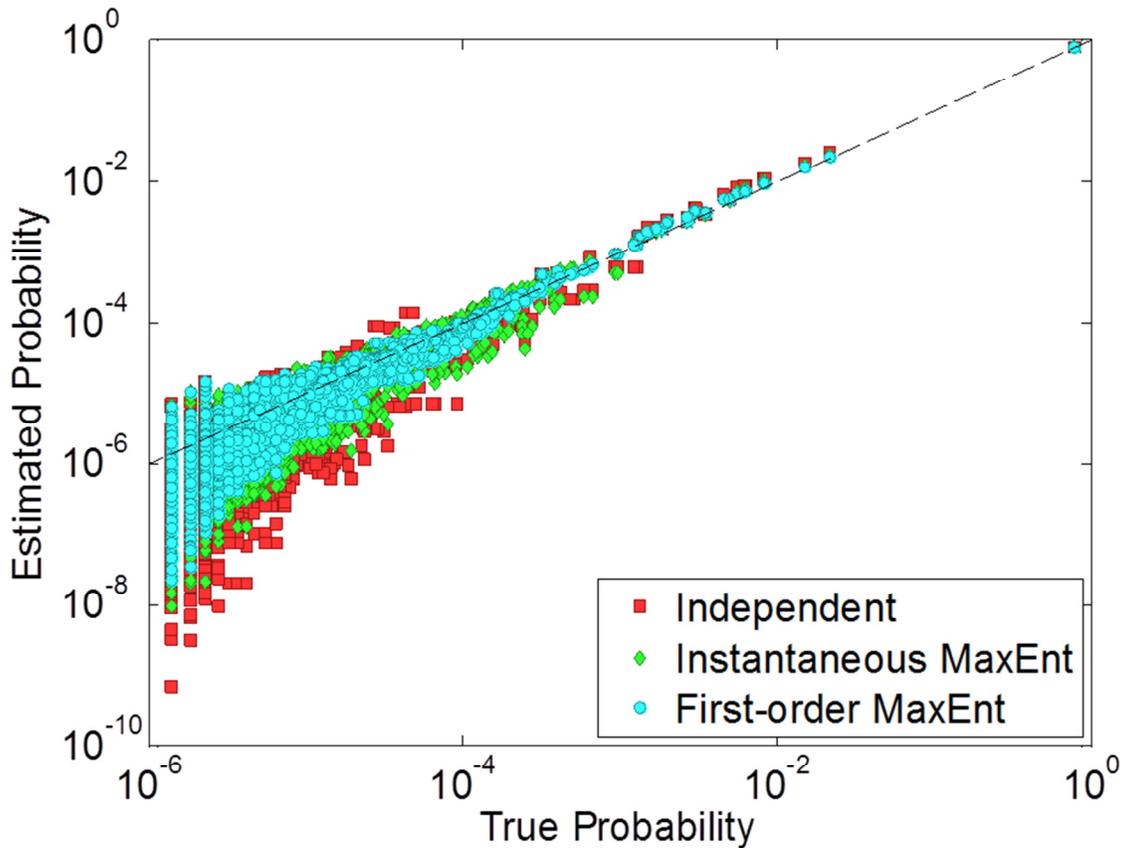


Figure 5.4 The circular representation demonstrates the 16 different stimuli in the center with the arrows indicating the movement direction of the drifting bars. The PSTHs of a neuron in response

to the 16 different stimuli are demonstrated on the outside. The size of the icon representing the stimulus is consistent with the overall average firing rate of that neuron, in which larger sizes demonstrate higher firing rates.

Figure 5.4 (continued)





(a)

Figure 5.5 Predicting spatiotemporal patterns of activity (a) Quantile-quantile plot: the estimated probability of observing population vectors using different encoding models versus the true probability from empirical distribution. Each dot represents a combination of two population vector at times  $t$  and  $t + 1$ . The dotted line represents equality for the estimated and true probabilities. (b) Synchronous pattern estimation: the sum of the estimated probabilities of all population vectors with a similar firing rate is plotted versus the number of '1' in the population vector. The best model is the one closer to the solid black line, which in this case is the first-order maximum entropy model.

Figure 5.6 (continued)

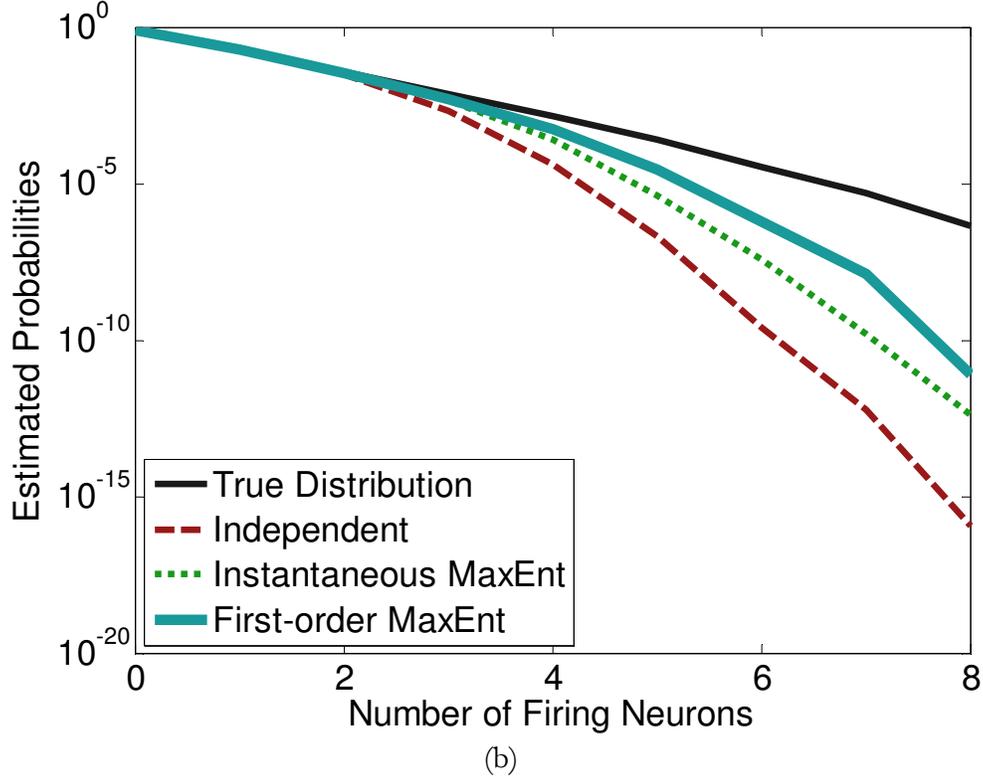


Figure 5.6 demonstrates the predictive power of the four different encoding models in estimating the probability of observing a sequence of population vectors,  $\{X^{(i)}, X^{(j)}, X^{(k)}\}$ . The joint probability distribution,  $P(X^{(i)}, X^{(j)}, X^{(k)})$ , can be expressed for the independent model as  $P_{ind}(X^{(i)})P_{ind}(X^{(j)})P_{ind}(X^{(k)})$ , and for the instantaneous maximum entropy model as  $P_{MaxEnt}(X^{(i)})P_{MaxEnt}(X^{(j)})P_{MaxEnt}(X^{(k)})$ , and for the first-order spatiotemporal maximum entropy model as  $P_{MaxEnt}([X^{(i)}, X^{(j)}])P_{MaxEnt}([X^{(j)}, X^{(k)}])/P_{MaxEnt}(X^{(j)})$ , and for the second-order spatiotemporal maximum entropy model as  $P_{MaxEnt}([X^{(i)}, X^{(j)}, X^{(k)}])$ . Figure 5.6a demonstrates the quantile-quantile plot, in which the second-order spatiotemporal maximum entropy model was better than the first-order spatiotemporal, instantaneous and independent

models. Figure 5.6b demonstrates the sum of all estimated probabilities with a similar number of ‘1’ in their population vectors, and the closest plot to the empirical plot determines the best model fit, which in this case is the second-order spatiotemporal model.

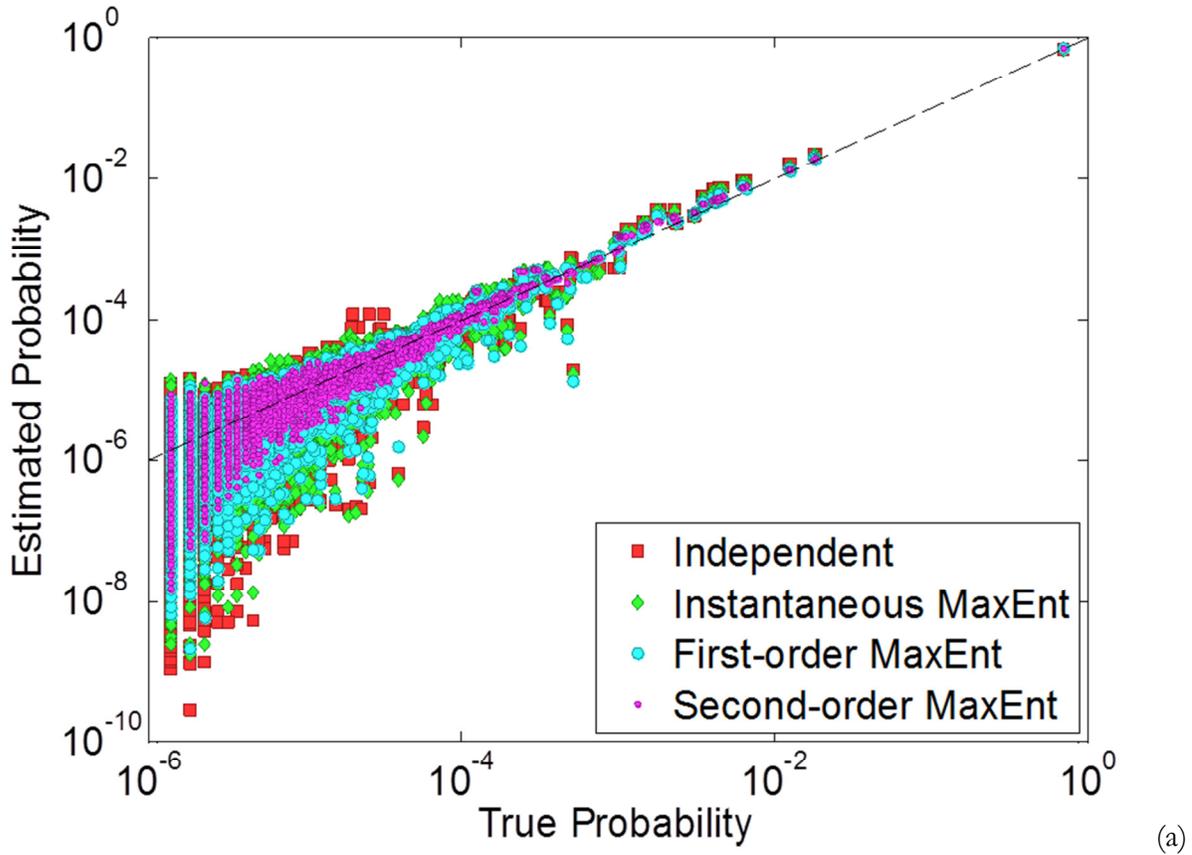
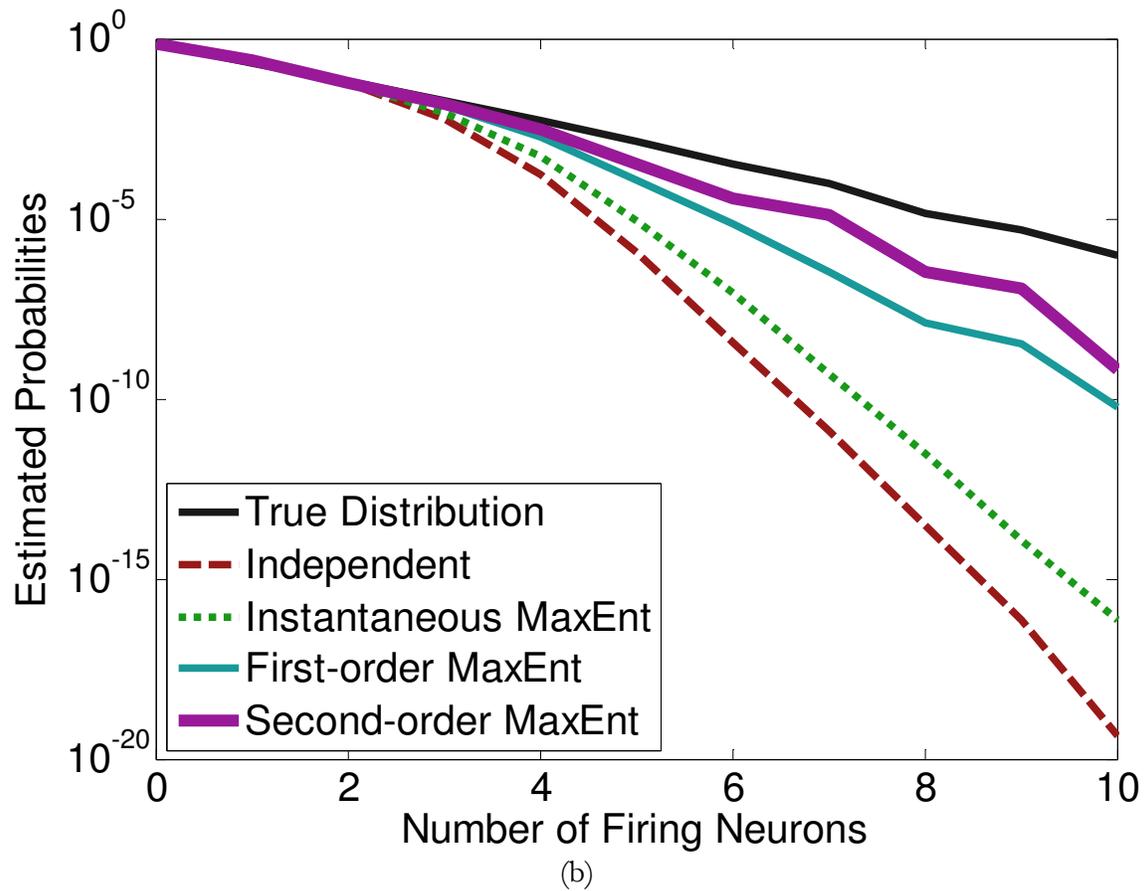


Figure 5.6 Predicting spatiotemporal patterns of activity (a) Quantile-quantile plot: the estimated probability of observing population vectors using different encoding models versus the true probability from empirical distribution. Each dot represents a combination of three population vector at times  $t$ ,  $t + 1$ , and  $t + 2$ . The dotted line represents equality for the estimated and true probabilities. (b) Synchronous pattern estimation: the sum of the estimated probabilities of all population vectors with a similar firing rate is plotted versus the number of ‘1’ in the population vector. The best model is the one closer to the solid black line, which in this case is the second-order maximum entropy model.

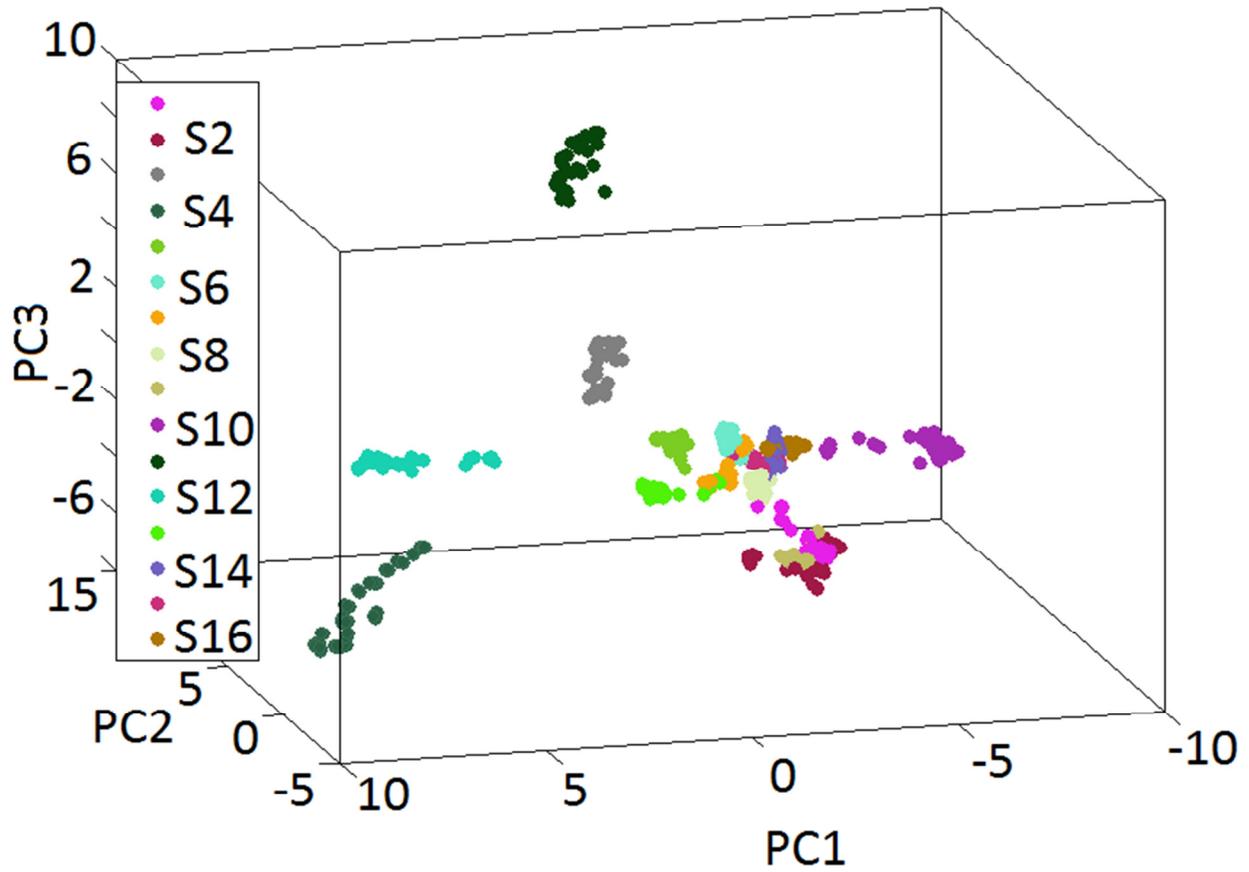
Figure 5.6 (continued)



Another evaluation method is by comparing the separability of network representations of different stimuli. The network representation is obtained from the neural responses to stimulus  $\varphi$ , referred to as trial of stimulus  $\varphi$ , and are expected to be more similar for other trials of stimulus  $\varphi$  and dissimilar to the other network representations. We also expect that the network representation obtained from trials of stimulus  $\varphi$  is somewhat closer to network representation obtained from trials of stimulus  $\varphi \pm \pi$ , referred to as across network, and also somewhat close to network representation from trials of stimulus  $\varphi \pm \pi/8$ , referred to as adjacent networks, and far from the rest of the network representations, referred to as the other networks. The similarity to across and adjacent networks is due to the existence of orientation-selective and movement-selective neurons in the V1 subpopulation, respectively.

The network representation of the independent encoding model is obtained directly from the spike trains using PCA. For that, a response matrix was computed from the binary spike train by summing the number of spike fired within 50 msec time intervals, and was arranged as a vector. Using the three most significant principal components, the vectors for approximately 40 trials of 16 different stimuli were projected into a 3D space, as demonstrated in Figure 5.7a. The level of separability among clusters of network representations for different stimuli was not equal throughout all 16 stimuli.

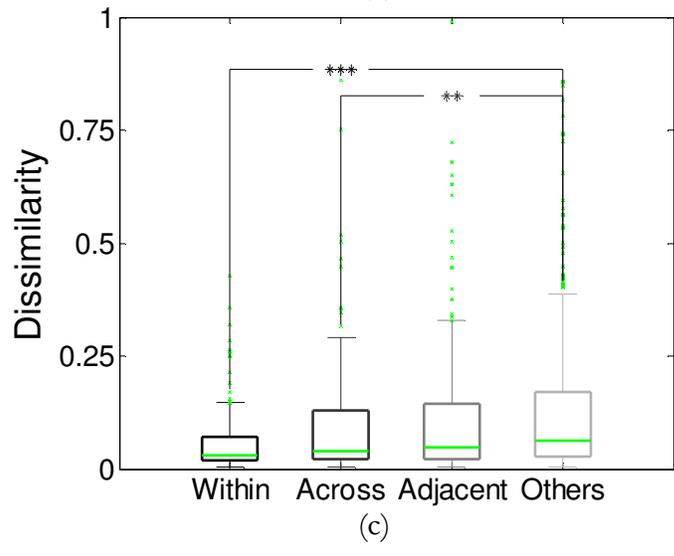
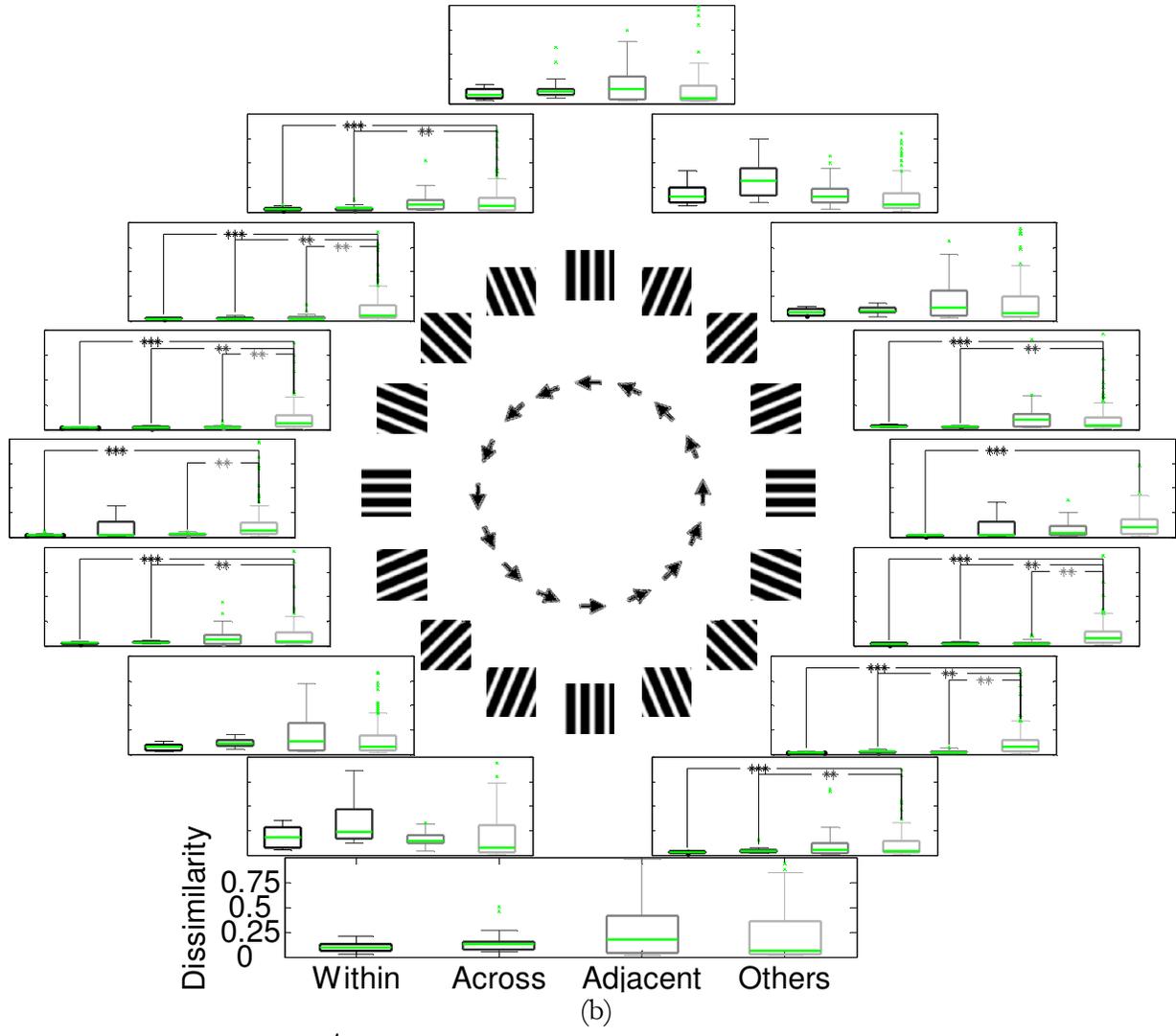
Figure 5.7b demonstrates a circular representation of the Euclidean distance computed between the network representations of within, across, adjacent and other stimuli using boxplots. It can be seen that within distances are typically less than the across and adjacent distances, and they are in general less than the other distances. In this figure, ‘\*\*’ and ‘\*\*\*’ demonstrates that a set of distances is statistically less than another set with a confidence interval of  $P < 0.001$  and  $P < 0.0001$ , respectively. The overall dissimilarity between different network representations is demonstrated in Figure 5.7c, in which the within and across are statistically less than the other networks, but no other conclusive argument can be made for the adjacent vs. other, and the within vs. across or adjacent for this independent encoding model.

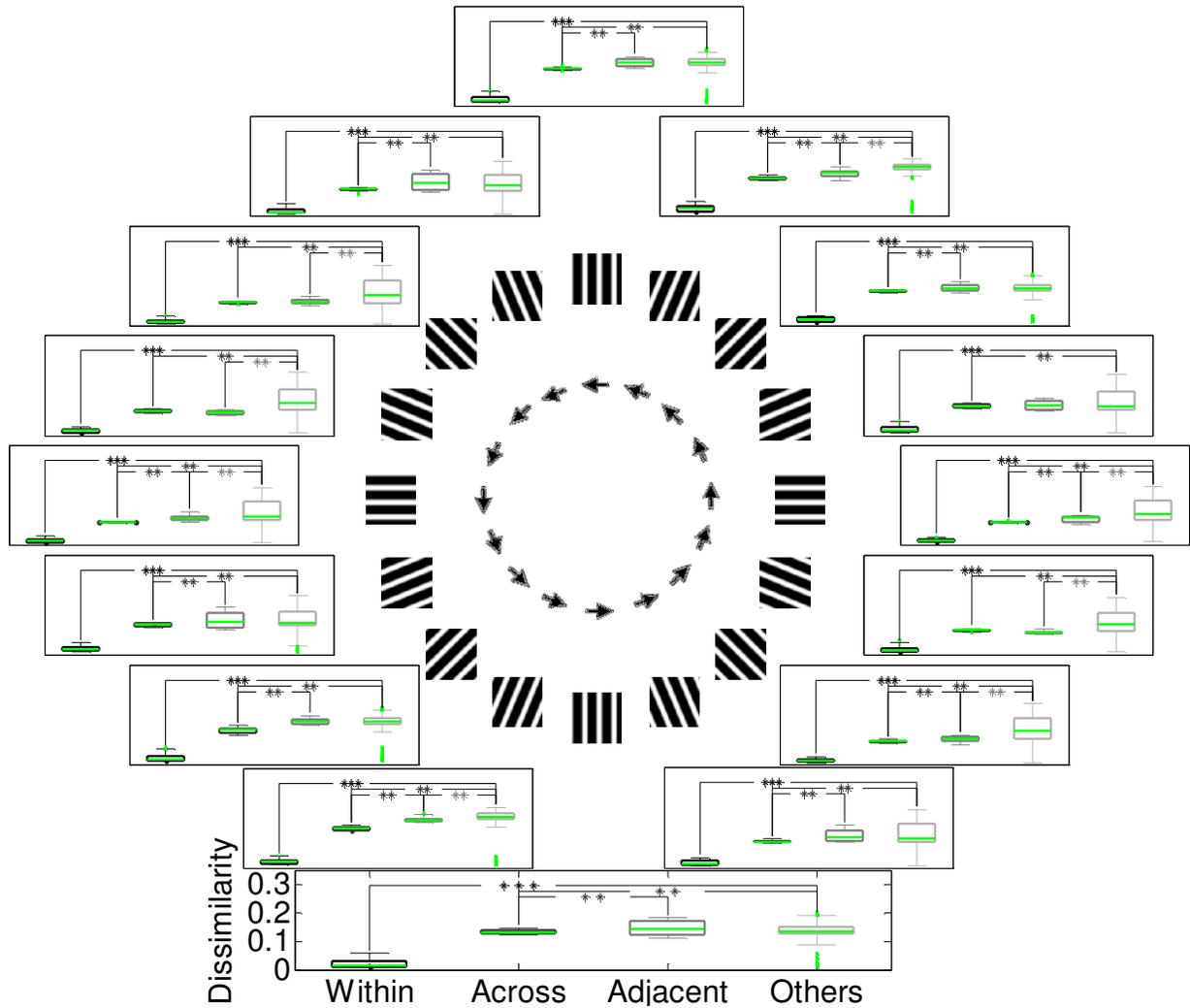


(a)

Figure 5.7 Independent encoding model (a) 3D projection of individual network representations of trials from the 16 stimuli using PCA. (b) Circular representation of the Euclidean distance between the network representations of one particular stimulus with within, across, adjacent and other network representations using boxplots. In the boxplots, the central line demonstrated as the green line is the median or second quartile, the edges of the box are the first and third quartiles, the whiskers extend to the most extreme samples that are not considered as outliers, and outliers are individually demonstrated as green dots. (c) The overall distances for all network representations of different stimuli

Figure 5.7 (continued)





(a)

Figure 5.8 Spatiotemporal maximum entropy encoding model (a) Circular representation of the Euclidean distance between the network representations of one particular stimulus with within, across, adjacent and other network representations using boxplots. (b) The overall distances or dissimilarity measure for all network representations of different stimuli

Figure 5.8 (continued)

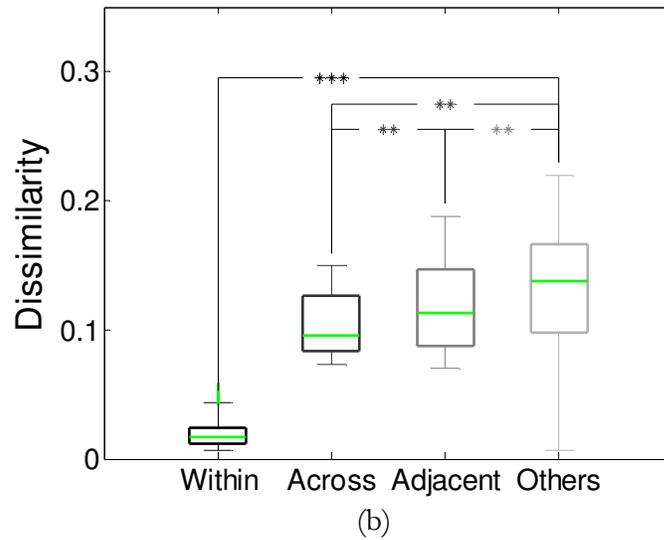


Figure 5.8a demonstrates a circular representation of the Euclidean distance computed between the network representations of the spatiotemporal encoding model for the network of within, across, adjacent and other stimuli using boxplots. It can be seen that within distances are always less than the across and adjacent distances, and they are also less than the other distances. The overall dissimilarity between different network representation is demonstrated in Figure 5.8b, in which the within is statistically less than the across, adjacent and other distances, the across is statistically less than adjacent and other distances, and finally the adjacent is statistically less than the other distances. Compared to the independent encoding model, the spatiotemporal maximum entropy model provides a much more conclusive encoding model with network representations that are not as much affected by noise in the neural response.

To show the importance of accounting for temporal correlations in addition to spatial correlations in predicting the population vectors, we counted the number of patterns that can be observed in an arbitrary network of 21 neurons with different assumptions on their collective activity. In the independent model with no temporal or spatial correlations among the neuronal constituents, neurons in an infinite long session could basically produce  $2^{21}$  population vectors for

the 21 sorted neurons, while this number grows exponentially for a sequence of two population vectors to  $2^{2 \times 21}$ , as demonstrated by the dotted red line in Figure 5.9a. In this figure, a sequence of one, two, three and four population vectors are represented as patterns with 0, 1, 2 and 3, Markov lags, respectively.

The counted number of different population vectors observed over an hour long session was only 748 individual patterns, far less than  $2^{21}$  patterns estimated by the independent model, suggesting that neurons don't fire independently. For the instantaneous maximum entropy model, the model predicts the number of a sequence of two population vectors to be  $748^2$ , demonstrated as the dotted green line, while the actual count is less than 7000 patterns. This inherently suggests using spatiotemporal maximum entropy model that include both temporal and spatial correlations among neuronal constituents. An important factor of this model, however, is to define the order of the Markov model. The solid blue line demonstrates the actual number of a sequence of three population vectors. The number of patterns estimated by the first-order and second-order spatiotemporal models are demonstrated by the dotted cyan and magenta lines, respectively. It can be seen that the estimates by the first-order spatiotemporal model is somewhat off the actual count, but the second-order spatiotemporal model seems to be much closer. This suggests that a second-order spatiotemporal maximum entropy model is sufficient to model the patterns generated by the V1 cortical neurons, and higher-order Markov models seems to be redundant. This suggests that the effective temporal support for the history of population response is around 15 msec.

Figure 5.9b demonstrates the average Kullback-Leibler (KL) divergence between the actual probability distribution of the observed population vectors in response to the 16 stimuli with the ones estimated using different encoding models. Errorbars demonstrate standard deviation. The KL divergence between the true distribution,  $P(X)$ , and its estimation,  $Q(X)$ , is estimated as

$\sum_X P(X) \log(P(X)/Q(X))$ . The KL divergence for the spatiotemporal maximum entropy model is significantly less than the instantaneous maximum entropy model with  $P < 0.01$ , and less than the independent model with  $P < 0.001$ . This is while the instantaneous maximum entropy model does not provide a statistically significant improvement compared to the independent model.

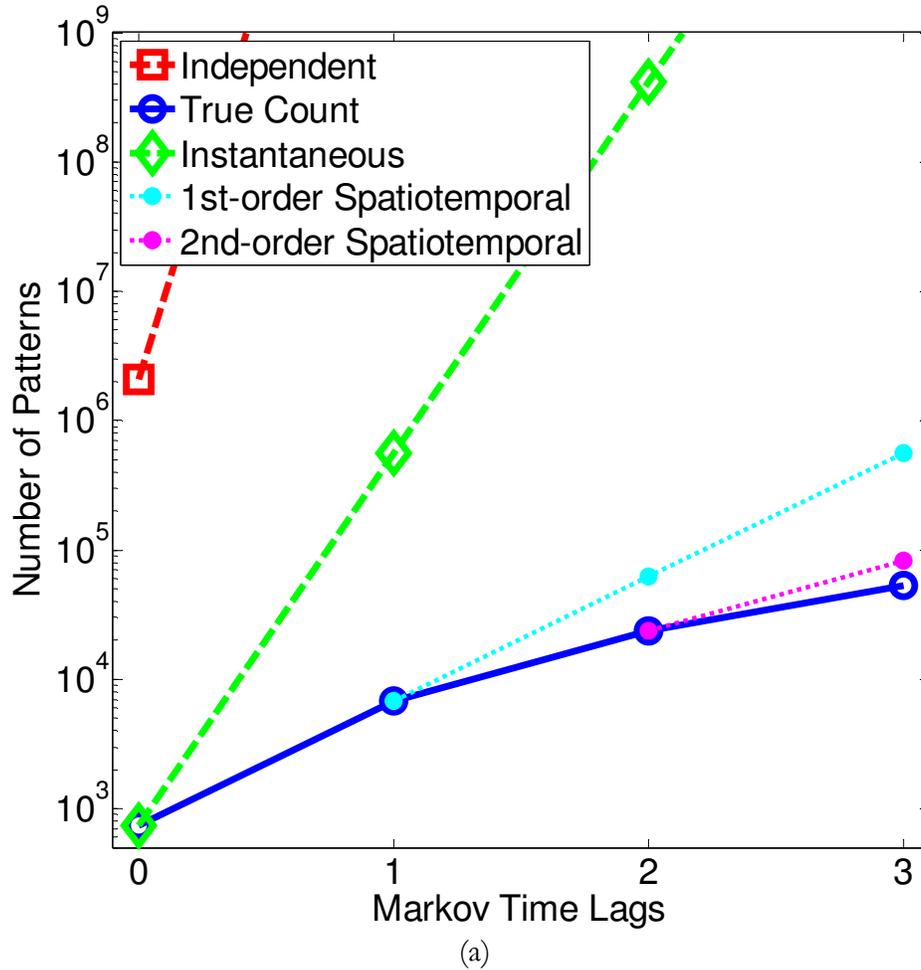
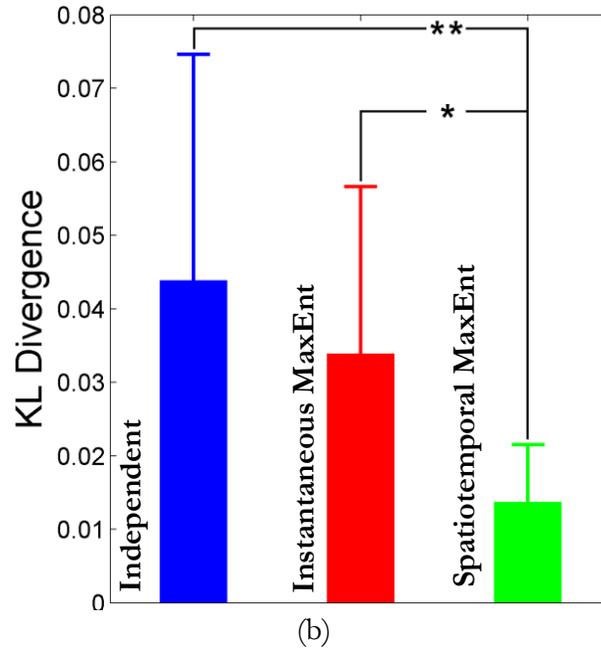


Figure 5.9 (a) the actual counted number of patterns in the collective activity of the neural population (solid blue line), and estimated counts by the independent, instantaneous maximum entropy, first and second-order spatiotemporal maximum entropy models, demonstrated as the dotted red, green, cyan and magenta lines, respectively. (a) Average KL divergence between the true and the estimated distributions over 16 stimuli (error-bars are the standard deviation).

Figure 5.9 (continued)



### 5.3. Discussion

We evaluated the performance of the spatiotemporal maximum entropy model in predicting the activity pattern of cat V1 neurons in response to drifting grating stimuli. A population of 21 neurons was recorded using microwires with tetrode arrays. We compared the prediction power of this model with the instantaneous pairwise maximum entropy model, as well as the independent model. We demonstrated that the extended model converges faster compared to [81], and also reduces the Kullback-Leibler divergence between the estimated and true distributions by 20 percent. Taken together, these results suggest the importance of accounting for temporal correlations in predicting the spatiotemporal patterns.

# 6

## Hypergraph Models with Higher Order Interactions

### Better Explain the Network Dynamics

Cortical neurons have long been hypothesized to encode information about external stimuli using one of two encoding modalities: temporal coding or rate coding [40]. In temporal coding, a neuron fires a spike that is almost always time locked to the external stimulus. Rate coding, on the other hand, presumes that a neuron responds to a stimulus by modulating the number of spikes it fires above or below a baseline level during a fixed time interval. In sensory systems such as visual and auditory cortices, temporal codes are more pronounced [85-88] while in motor systems, rate codes seem to be predominantly present [18, 89, 90].

The mapping relationship between an external stimulus  $s$  and a neural response  $r$  (whether it expresses a temporal code or a rate code) is typically described by the conditional density  $P(r|s)$ . This density completely expresses the “receptive field” of a sensory neuron, or a “tuning function” of a motor neuron [91, 92]. It has been the standard metric of stimulus-driven neural response in neurophysiological experiments for many decades. Recently, correlation among neurons has emerged as another cortical response property that may be playing a role in encoding stimuli [72, 93-95]. This property has naturally emerged as a result of the ability to *simultaneously* record the activity

of an ensemble of neurons in selected brain targets while subjects carry out specific functions [3-6]. Not surprisingly, many important aspects of behavior-neurophysiology relationships stem from the collective behavior of many neuronal elements [2, 96]. The major gain over sequentially recording single-neuron activity is the ability to access the  $n^{th}$ -order conditional density of the observed population,  $P(r_1, r_2, \dots, r_n | s)$ , thereby providing a closer examination of the response space of the underlying biological system that these neurons represent. Clearly, calculation of the joint distribution requires knowledge of all the possible network states, which is practically impossible.

In many cases, the variables  $r_1, r_2, \dots, r_n$  exhibit variable degrees of statistical dependency [73, 97]. Generally speaking, this statistical dependency may either result from a significant overlap in the receptive fields of these neurons, often referred to as *response to a common input*, or from some type of anatomical connectivity between the neurons, or from both [98-100]. To date, it is unclear whether this dependency is part of an efficient neural encoding mechanism that attempts to synergistically and cooperatively represent the stimulus in a network-wide interaction between the neuronal elements, or hinders this encoding mechanism if it were to redundantly express independent neuronal responses to that stimulus.

Numerous information theoretic measures have been proposed to address this synergy-redundancy question [94, 101-106]. In the previous chapter, a maximum entropy model relying on second-order interaction (pairwise correlation) between neurons has been shown to result in a strong cooperative network behavior. The principle of maximum entropy was argued to yield a data distribution with the maximum uncertainty among all possible distributions that could explain the observed data [107]. More recently, a minimum mutual information principle was argued to yield a tighter lower bound on the mutual information between the stimulus and the response of *any* system that explains the data [108]. However, neither approach has directly addressed the following

fundamental questions: how higher order interactions observed in spatiotemporal spiking patterns of cortical neurons contribute to *actual* information transfer through the system and how the joint distribution can be optimally expressed using knowledge of any intrinsic structure in their underlying network connectivity.

In this chapter, we propose an information theoretic metric that helps assess whether a connection-induced dependency in neural firing is part of a synergistic population code. Specifically, we propose to optimally express the conditional neural response in terms of a consistent *hypergraph*. We argue that the stimulus-specific network model associated with each hypergraph attempts to discover how stimulus information governs the way neurons signal one another. We demonstrate that this hypergraph representation yields a minimum entropy distance between the conditional response entropy of a single trial and the “true” response entropy estimated from a concatenation of many trials. We also show that this approach falls under the class of maximum entropy models that seek to find the probability distribution that maximizes the entropy with respect to the known constraints.

The chapter is organized as follows: first we will discuss the theory of the framework behind hypergraph models. Then, we will describe the methods we used to generate the neural data, followed by the obtained results. Discussion is provided about relations to previous work and suggestions for future work.

## 6.1. Hypergraph Maximum Entropy Model

Let’s assume we have a population of  $n$  neurons encoding stimulus  $S$  that can take one out of  $P$  possible values  $\{s_1, s_2, \dots, s_p\}$ . These can be a set of discrete values of a continuous, time-varying

deterministic function, or discrete-valued parameters of a latent stochastic process. For notational convenience, we will denote herein by  $P_S(\cdot)$  quantities conditioned on the presence of  $S$  and make the distinction clear wherever needed. The encoding dictionary is characterized by the conditional probability of the neural response given  $s$ , or the likelihood  $P_S(\mathbf{r})$ , where  $\mathbf{r}$  is a matrix of population responses,  $\mathbf{r} = [r_1, r_2, \dots, r_n]$ . Each neuron's response is expressed in terms of the spike count in a fixed length time bin that is small enough to contain a binary '1' (spike) or a binary '0' (no spike). In the presence of  $s_p$ , we assume that the variables  $r_1, r_2, \dots, r_n$  exhibit some form of stimulus-specific statistical dependency that can be characterized by the  $k^{th}$ -order marginal,  $P_S([r_1, r_2, \dots, r_k])$ , where  $k$  is the number of neurons involved in encoding  $S = s_p$ , and can vary with variable  $p$ . We seek to express the encoding dictionary as a product of all the possible  $k^{th}$ -order conditional marginals,  $P_S^{(k)}(\mathbf{r})$ . Given that a population of  $n$  neurons has  $2^n$  states, expressing the encoding dictionary by searching for the optimum product of marginals is a computationally prohibitive task. Therefore, it is highly desirable to reduce the search space by finding an optimum set of interactions between the variables  $r_1, r_2, \dots, r_n$  for each value of the stimulus  $S$ .

Shannon's mutual information can be used to compute how much information about  $S$  is encoded in  $\mathbf{r}$ . However, it depends on obtaining a good estimate of the *noise entropy* [52], denoted  $H_S(\mathbf{r})$ . When correlation between the variables is present, *multi-information* [109] -also known as *excess entropy* [110]- can be used. It is the relative entropy of the joint density of the variables with respect to the product of their individual densities

$$I_S(\mathbf{r}) = \sum_{\mathbf{r}} P_S(\mathbf{r}) \log \frac{P_S(\mathbf{r})}{\prod_{r_i} P_S(r_i)} = \sum_{r_i} H_S(r_i) - H_S(\mathbf{r}) \quad (6-1)$$

where  $P_s(r_i)$  is the marginal distribution of neuron  $i$  given  $s$ , and  $H_s(r_i)$  represents the conditional entropy of neuron  $i$ . Since the joint entropy cannot exceed the sum of marginal entropies, i.e.,  $\sum H_s(r_i) \geq H_s(\mathbf{r})$ , the multi-information is always positive, and equality holds only when the responses are statistically independent.

### 6.2.1. Factoring Multi-information

Although multi-information accounts for the presence of correlation, it does so in a global context. It is highly desirable to quantify how much of that correlation is intrinsic to independent, pairwise, triplets or higher order interactions among neurons given a particular stimulus. To do this, the  $k^{th}$ -order connected information measure,  $I_s^{(k)}(\mathbf{r})$  [111], also known as co-occurrences [112], computes the distance between the entropy of the  $k^{th}$ -order and the  $(k - 1)^{th}$ -order conditional densities

$$I_s^{(k)}(\mathbf{r}) = H[P_s^{(k-1)}(\mathbf{r})] - H[P_s^{(k)}(\mathbf{r})] \quad (6-2)$$

where  $P_s^{(k)}(\mathbf{r})$  is the  $k^{th}$ -order conditional density and is estimated as a normalized product of marginals up to the  $k^{th}$ -order. It can be easily shown that multi-information is the sum of all  $k^{th}$ -order connected information terms

$$I_s(\mathbf{r}) = \sum_{k=2}^N I_s^{(k)}(\mathbf{r}) \quad (6-3)$$

By computing the ratio of each  $k^{th}$ -order connected information to the multi-information,  $I_s^{(k)}/I_s$ , the contribution of each  $k^{th}$ -order correlation in the encoding dictionary can be obtained.

The order of the encoding model is estimated as the maximum  $k$  with significant connected information, as will be shown in the results section.

We demonstrate the above theoretical analysis with a simple example. In Figure 6.1, we graphically represent the encoding of  $S$  using a population of three neurons  $i$ ,  $j$  and  $k$ . Here we have  $P$  possible stimuli that these three neurons are encoding, where it may also represent different variations of parameters associated with a single time-varying stimulus. The stimulus is encoded independently (e.g. neuron  $k$ ) and jointly (neurons  $i$  and  $j$ ). Since no 3<sup>rd</sup>-order interaction is observed between the variables, the multi-information in this case is equal to the second-order connected information,  $I_S(\mathbf{r}) = I_S^{(2)}(\mathbf{r})$ , and the encoding model order for this population is  $k = 2$  as it fully characterizes the response properties using interactions limited to pairwise correlations. The later can be also measured by the synaptic efficacy [113]. The mutual information between the response and the stimulus is shown by the blue region in Figure 6.1, while the noise entropy,  $H_S(\mathbf{r})$ , is shown by the red region.

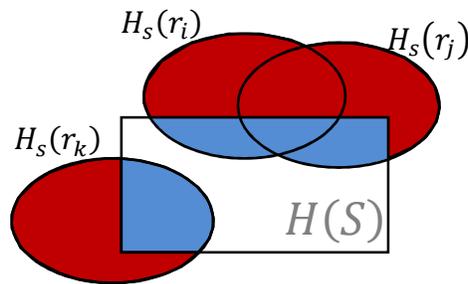


Figure 6.1 A stimulus entropy representation by a population of 3 neurons.  $H(S)$ , expressing the stimulus dictionary. The blue region is the contribution of the three neurons in encoding the stimulus space.  $H_S(\mathbf{r})$  represents the noise entropy and is shown in red. While neuron  $k$  independently encodes the stimulus, neurons  $i$  and  $j$  encode it independently as well as jointly.

## 6.2.2. Graphical representation

A stimulus-induced dependency between the neurons can be graphically represented using Markov Random Fields (MRF) [114]. Correlation can be represented in this graph, denoted  $G$ , as an undirected link between the neurons, or factors. In such case, the encoding dictionary is expressed as

$$P_s(\mathbf{r}) = \frac{1}{Z} \prod_{\pi_i \in G} P_s(r_{\pi_i}) \quad (6-4)$$

$$Z = \sum_i \prod_{\pi_i \in G} P_s(r_{\pi_i})$$

where  $\pi_i$  denotes the set of neurons connected to neuron  $i$ , and  $Z$  is a normalization factor (also known as the partition function).

The edges in a Markov network structure, or *cliques* [115], express the role of second-order interactions between neurons in encoding  $S$ . A Markov network incorporating only first-order cliques assumes neurons to be conditionally independent. This probabilistic model is *non-cooperative* because it does not consider any stimulus-induced correlation among neurons. The encoding dictionary in such case is expressed as

$$P_s(\mathbf{r}) = \frac{1}{Z} \prod_{i=1}^n P_s(r_i) \quad (6-5)$$

On the other hand, when second-order cliques are present, we get the *Ising* model from statistical physics [107, 116]. This model represents a Maximum Entropy model when the density is expressed

as an exponential function of the weighted individual responses and pairwise responses

$$P_s(\mathbf{r}) = \frac{1}{Z} \exp\left(\sum_i \delta_i r_i + \frac{1}{2} \sum_{i \neq j} \gamma_{ij} r_i r_j\right) \quad (6-6)$$

The unknown parameters  $\delta_i$  and  $\gamma_{ij}$  are estimated using an iterative scaling algorithm so that the expected values of the individual responses,  $\langle r_i \rangle$ , and pairwise responses,  $\langle r_i r_j \rangle$ , match the measurable information in the experiment [79]. This information is the mean spike count within a fixed window (i.e. firing rates) of individual neurons and observed pairwise correlations.

Maximum entropy models similar to (6-6) have been shown to faithfully explain the collective stimulus-driven network behavior of ~50-100 retinal Ganglion cells (RGCs) in isolated slices *in vitro* [79, 107, 116-118]. These neurons have well-characterized receptive fields, typically exhibit low firing rates, and were shown to capture ~98% of the multi-information under stimulus presence [116] while only account for ~88% of the multi-information in spontaneous network activity [79]. Cortical neurons observed *in vivo*, on the other hand, are morphologically very different from RGCs, exhibit very heterogeneous firing patterns [39], and therefore are believed to encode stimuli in a more complex fashion than sensory neurons observed *in vitro* such as RGCs [119, 120]. These neurons receive inputs from many cortical and subcortical areas and generate outputs through numerous feedback loops with optimal error-correcting capabilities that are known to play a very important role in optimizing the subject's *experience* with the stimulus over multiple repeated trials [121, 122]. Therefore, their underlying complex structure results in interaction patterns that have much longer spatial and temporal dependence, yielding sequence lengths that are highly nonstationary.

### 6.2.3. Cooperative Model

Herein, we generalize the Markov structure to edges that can connect any number of vertices to model higher order interactions at a relatively very fine time scale (3 msec). This time scale has been shown to describe excess synchrony typically observed between cortical neurons *in vivo* [90, 123, 124]. This yields a hypergraph  $G$  in which every vertex expresses a marginal distribution and represents a *cooperative* model of the encoding dictionary as [125]

$$P_s(\mathbf{r}) = \frac{1}{Z_G} \prod_{\pi_i \in G} \prod_{\pi_j \subseteq \pi_i} P_s(r_{\pi_j})^{q_j} \quad (6-7)$$

where  $Z_G$  is a normalization factor, and  $P_s(r_{\pi_j})$  are factors defined by the set  $\pi_j$ , which itself is a subset of  $\pi_i$ . The subsets  $\pi_i$  are the edges in the hypergraph  $G$ . The power of the marginals in equation (6-7),  $q_j = \pm 1$ , is defined based on the Möbius inversion property as [126]

$$q_j = -1^{|\pi_j| - |\pi_i|} \quad (6-8)$$

where  $|\cdot|$  is the cardinality of a set. The factorization in (6-7) indicates that all the marginals associated with the subsets of  $\pi_i$  are represented *in addition* to the marginals of set  $\pi_i$ .

The entropy of the cooperative model can be derived from the entropy of all its factors as

$$H_s(\mathbf{r}) = H_{Z_G} + \sum_{\pi_i \in G} \sum_{\pi_j \subseteq \pi_i} q_j H_s(r_{\pi_j}) \quad (6-9)$$

where the entropy  $H_{Z_G}$  represents a constant bias term resulting from the partition function. An example is illustrated in Figure 6.2a. Here we have a hypergraph including four factors that represent the network structure of a population of six neurons. In this hypergraph, circles represent neurons and squares represent factors, respectively. Using (6-7), the encoding dictionary of the cooperative model represented by this hypergraph is factorized as

$$P_s(\mathbf{r}) = \frac{1}{Z_G} \times \frac{P_{123}P_1P_2P_3}{P_{12}P_{23}P_{13}} \times \frac{P_{12}}{P_1P_2} \frac{P_{13}}{P_1P_3} \frac{P_{23}}{P_2P_3} \frac{P_{24}}{P_2P_4} \frac{P_{25}}{P_2P_5} \frac{P_{36}}{P_3P_6} \times P_1P_2P_3P_4P_5P_6 \quad (6-10)$$

where  $P_{ijk}$  is short for  $P_s(r_i, r_j, r_k)$ , and every bracket represents factor sets of different orders. The cooperative model in this example can be further simplified to

$$P_s(\mathbf{r}) = \frac{1}{Z_G} \frac{P_{123}P_{24}P_{25}P_{36}}{P_2^2P_3} \quad (6-11)$$

The  $k^{th}$ -order interactions among neurons illustrated by the hypergraph are shown as ovals, representing the entropy space of each factor. Using (6-9), the entropy estimate of the cooperative model in Figure 6.2 is expressed as

$$H_s(\mathbf{r}) = H_{Z_G} + H_{123} + H_{24} + H_{25} + H_{36} - 2H_2 - H_3 \quad (6-12)$$

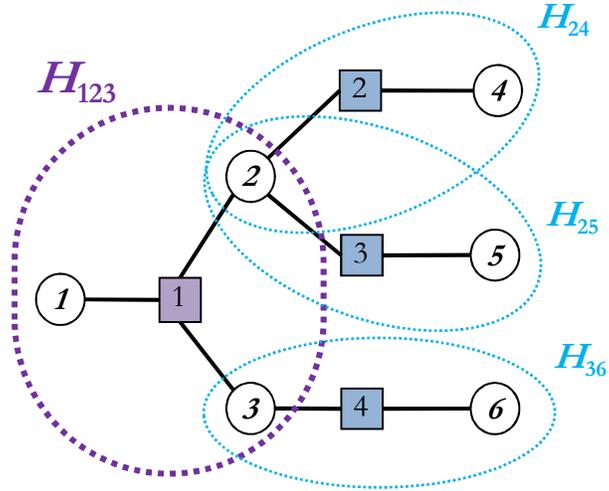


Figure 6.2 A hypergraph including four factors, representing a Markov network model of a population of six neurons. Neurons are illustrated as circles and factors are illustrated as squares. The terms  $H_{123}$ ,  $H_{24}$ ,  $H_{25}$ , and  $H_{36}$  are the entropies of the marginal factors defined by the hypergraph.

#### 6.2.4. Minimum Entropy Distance

In a typical experiment, the stimulus is presented over multiple repeated trials to assess all the possible system states expressed by the output spike trains. The stimulus itself can be an actual information-bearing signal or the output of another neural population, as typical in the layered structure of the cortex [127]. We therefore assume that the “true” entropy of the population given  $\mathcal{S}$  is computed by concatenating a large number of repeated trials for that stimulus. Given the distribution of these data, we seek to find the optimal factorization that results in a model entropy with minimum distance to the true entropy. We refer to this as the minimum entropy distance (MinED) algorithm. This algorithm yields a hypergraph,  $G^*$ , obtained by minimizing the following objective function

$$\text{MinED: } G^* \equiv \min_G |H_{True} - H_G| \quad (6-13)$$

where  $H_{True}$  is the true population entropy,  $H_G$  is the entropy estimate of the population from single trial data.

To initialize the search, we assume the least informative prior expressed by the independent model. The algorithm searches for the best  $k^{th}$ -order marginal out of all  $\binom{N}{k}$   $k^{th}$ -order marginals that minimizes the entropy distance. In each step, only  $k^{th}$ -order marginal factors are accumulated in the hypergraph. Changes to the graph structure are allowed only when the entropy distance decreases. This operation is repeated until convergence occurs. The hypergraph is saved at the end of each iteration, and then continues to higher orders.

The MinED algorithm can also be categorized as a maximum entropy model. One notable difference is that the testable information is the rate of occurrence of network states,  $P_{True}$ , and not the estimates of the individual and pairwise neurons' firing rates,  $\langle r_i \rangle$  and  $\langle r_i r_j \rangle$ , as in [107]. Although we don't have access to the true rate, we have a repertoire of network states collected over multiple repeated trials. Based on this testable information, (6-13) can be rewritten as

$$H_{True} - H_G = \sum_{\mathbf{r}} P_{True} \log P_{True} - \sum_{\mathbf{r}} P_{True} \log P_G \quad (6-14)$$

Based on the maximum entropy principle,  $H_G$  is always less than or equal to  $H_{True}$ , and the maximum  $H_G$  is reached when the MinED objective function is minimized.

## 6.2. Probabilistic Spiking Model with Synaptic Plasticity

Pairwise Maximum entropy models, to our knowledge, have not been applied to stimulus-driven responses of large sized networks of cortical neurons observed *in vivo*. There are multiple reports in the literature suggesting that these models are only suitable for neural systems that are small in size ( $\sim 10$ -20 cells) and operate below a cross over point of the model's predictive power [128]. To gain some insight on whether this is the case, we sought here to simulate an encoding model with the following features:

1. A relatively large size spiking network model to test the generalization capability of the MinED algorithm to larger systems and contrast it to maximum entropy models that perform well in small sized systems.
2. Testable information limited to a small subpopulation of the entire simulated network model so that the true probability distribution can be faithfully approximated across many experiments.
3. Spike trains at the input of the network (i.e. discrete binary representation) rather than continuous inputs (e.g. current injection typically used in experiments *in vitro*) to mimic cortical neurons' input-output mapping.
4. Biologically plausible statistical learning rule that optimizes the system's experience with the stimulus by varying the network structure over multiple repeated trials.
5. Response statistics of the population similar to those observed experimentally.

## 6.1. Probabilistic Spiking Neurons

We simulated a motor task in which a moderately large population of neurons (166 neurons)

encoded a 2D arm trajectory of a subject reaching from a center position to one of 8 peripheral targets [18]. We used an inhomogeneous Poisson model to simulate the spike train data for two interconnected layers of neurons. This model has been shown to fit the firing pattern of motor cortical neurons under a variety of motor tasks [129, 130]. In this model, the firing probability of each neuron at time  $t$  is expressed by the conditional mean intensity function  $\lambda_i(t|g_i(t), h_i(t))$ , where  $g_i(t)$  describes the tuning to the movement parameters (i.e. the stimulus), and  $h_i(t)$  denotes the spiking history of neuron  $i$  (to account for refractory effects) and that of other neurons connected to it up to time  $t$  [34, 35]

$$\lambda_i(t|g_i(t), h_i(t)) = \exp(\beta_i + g_i(t) + h_i(t)) \quad (6-15)$$

where  $\beta_i$  is the background rate of neuron  $i$  that represents membrane voltage fluctuations and

$$h_i(t) = \sum_{j \in \pi_i} \sum_{m=0}^{M_{ij}} \alpha_{ij}(m\Delta) S_j(t - m\Delta) \quad (6-16)$$

where  $\pi_i$  is the set of (input) neurons affecting the firing of (output) neuron  $i$ ,  $M_{ij}$  is the number of history time bins that relate the firing probability of neuron  $i$  to activity from parent neuron  $j$ ,  $S_j(t - m\Delta)$  is the state of neuron  $j$  in bin  $m$  (takes the value of 0 or 1),  $\Delta$  is the bin width. The parameter  $\alpha_{ij}$  models the coupling between neuron  $i$  and neuron  $j$  as [131]

$$\alpha_{ij}(t) = \begin{cases} 0, & t < l_{ij}\Delta \\ A_{ij}(t) \exp(-3000t/M_{ij}), & t \geq l_{ij}\Delta \end{cases} \quad (6-17)$$

where  $A_{ij}(t)$  is the strength of the coupling and can be positive or negative depending on whether

the connection is excitatory or inhibitory, and  $l_{ij}$  is the synaptic latency (in bins) associated with that connection.

Our model consisted of a 2-layer network as depicted in Figure 6.3a. The input layer (66 neurons) was non-cooperative, meaning that the term  $h_i(t)$  was absent from the model (6-15) for these 66 neurons. For this layer, neurons were tuned to two movement parameters: direction and end-point through  $g_i(t)$  as

$$g_i(t) = \eta_i V(t + \zeta) \cos\left(\frac{\theta(t + \zeta) - \theta_i}{\omega_i}\right) - \kappa_i \frac{(\varphi_x - x_i)^2 + (\varphi_y - y_i)^2}{\sigma_i} \quad (6-18)$$

where  $\eta_i$  and  $\kappa_i$  are the weights assigned to the cosine and end-point tuning terms, respectively,  $V(t + \zeta)$  is the movement speed (kept constant) to be reached after  $\zeta$  sec,  $\theta(t + \zeta)$  is the movement direction,  $\theta_i$  is the neuron's preferred direction [18],  $\varphi_x$  and  $\varphi_y$  are the  $x$ - and  $y$ -components of the intended target end-point, respectively, while  $x_i$  and  $y_i$  are the  $x$ - and  $y$ -components of the preferred end-point of neuron  $i$ , respectively. The parameter  $\omega_i$  and  $\sigma_i$  control the direction and end-point tuning widths, respectively. Based on these settings, neural responses would appear to be correlated due to the overlap between the tuning characteristics (i.e. common input), and not due to actual interaction among themselves. In our model, neurons 1 to 40 were each cosine-tuned to a uniformly distributed preferred direction between  $-180^\circ$  and  $180^\circ$  as illustrated in Figure 6.3b. Neurons 31 to 66 were each tuned to a preferred end-point in a 2D space divided into a  $6 \times 6$  grid. The center of each cell in the grid was considered as the end-point as shown in Figure 6.3c. Accordingly, 10 neurons (neurons indexed 31 to 40) were tuned to both direction and endpoint. This mimicked the heterogeneous tuning characteristics frequently observed in motor cortex neurons recording experiments [27, 132].

The cooperative layer (100 neurons), on the other hand, followed the model (6-15) with the term  $g_i(t) = 0$ . Initially, these neurons did not have any tuning preferences and were fully interconnected among themselves with fixed weights, with 80 excitatory neurons and 20 inhibitory neurons similar to the 1:4 ratio typically observed in cortex [133]. For this layer,  $h_i(t)$  was expressed as

$$h_i(t) = \sum_{j \in \pi_i^{nc}} \sum_{m=0}^{M_{ij}} \alpha_{ij}^{nc}(m\Delta) S_j(t - m\Delta) + \sum_{k \in \pi_i^c} \sum_{m=0}^{M_{ik}} \alpha_{ik}^c(m\Delta) S_k(t - m\Delta) \quad (6-19)$$

where  $\pi_i^{nc}$  and  $\pi_i^c$  are the parent sets of neuron  $i$ , while  $\alpha_{ij}^{nc}$  and  $\alpha_{ij}^c$  model the connections from non-cooperative layer and cooperative layer neurons, respectively, as expressed in (6-19). Therefore, the encoding of these neurons to movement parameters resulted from the random connections with subpopulations in the non-cooperative layer. Specifically, each cooperative layer neuron received excitatory connections from 4 randomly selected non-cooperative layer neurons. The synaptic latency  $l_{ij}$  of the connections from the non-cooperative layer neurons was randomly selected from a uniform distribution between 1-3 bins, while for the cooperative layer connections,  $l_{ik}$  was set to 1 bin. Table 6.1 lists the values of the parameters used in (6-16) to (6-19).

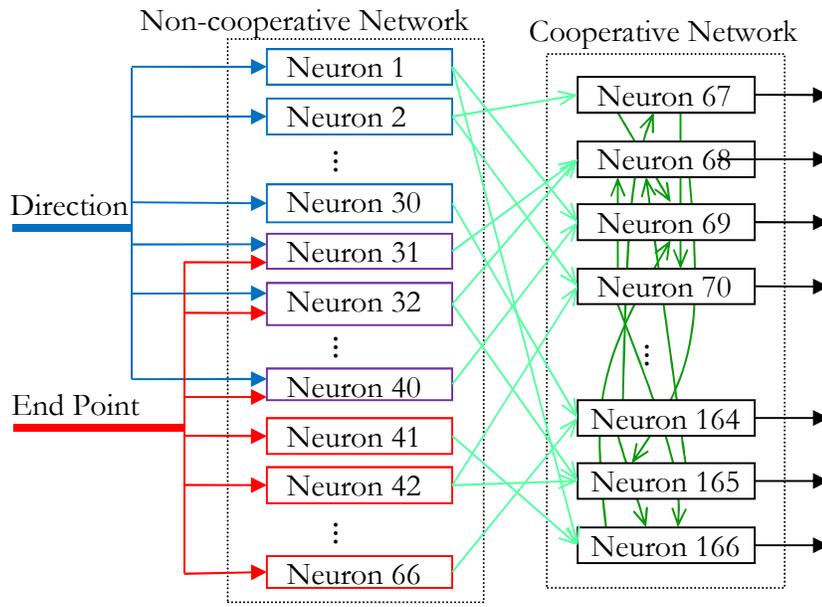
## 6.1. Task Learning

We used a biologically plausible statistical learning rule to optimize the system experience with the task. This rule was governed by a spike timing dependent plasticity (STDP) mechanism. This mechanism has been shown to take place *in vitro* [134] and is believed to underlie cortical plasticity *in vivo* [135-137]. Specifically, under this mechanism, the connection strength between any given pair of interconnected neurons in the cooperative layer was allowed to naturally change if the pair was

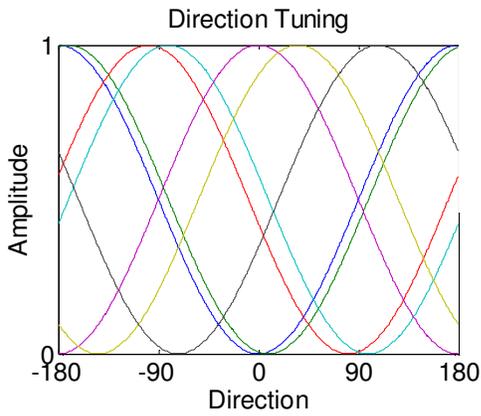
“functionally correlated”. Two neurons in the cooperative layer were considered functionally correlated if they received connections from the same non-cooperative layer neuron with different synaptic latencies or from more than one non-cooperative layer neuron with overlapped tuning. This is illustrated in Figure 6.4a by cooperative layer neurons  $C$  and  $D$ . In this example, only the connection from neuron  $C$  to neuron  $D$  is expected to be strengthened as a result of learning, while all other connections are expected to be weakened if no overlap exists between the tuning of neurons A and B. The details of this learning rule are reported elsewhere [136, 138].

Table 6.1 Values of the parameters used for the model of equations (6-16), (6-17), (6-18) and (6-19). These parameters were fixed for all neurons in the non-cooperative and cooperative layers.  $A_{ik}^{c+}(t)$  and  $A_{ik}^{c-}(t)$  model the weight of the excitatory and inhibitory connections, respectively, between cooperative layer neurons.

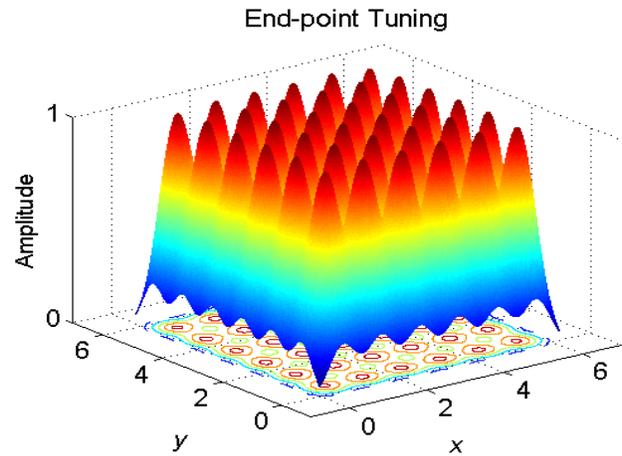
Parameter	Value	Parameter	Value
$\beta_i$	$\log(5)$	$M_{ij}$	40
$\eta_i$	1.5	$M_{ik}$	40
$\omega_i$	1	$A_i^{nc-}(t)$	8
$\kappa_i$	5	$A_{ik}^{c+}(t)$	0.01 (Initial value)
$\sigma_i$	1	$A_{ik}^{c-}(t)$	-8



(a)



(b)



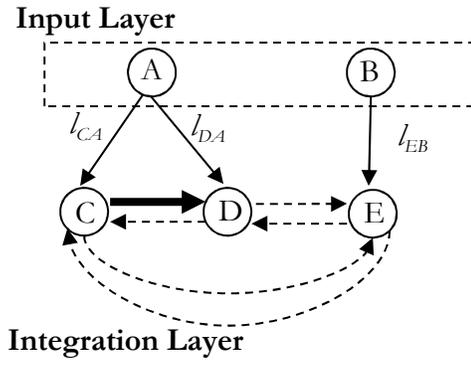
(c)

Figure 6.3 (a) A simplified version of the network structure. The non-cooperative (input) layer consists of 40 direction-tuned neurons and 36 end-point tuned neurons with 10 neurons tuned to both movement parameters. Each cooperative layer neuron receives 4 excitatory connections from

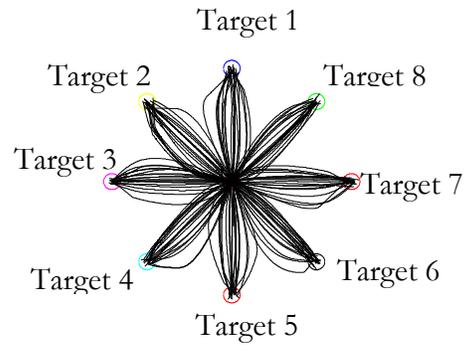
the non-cooperative layer. Connections within the cooperative layer are 80% excitatory and 20% inhibitory. (b) Sample cosine-tuning functions for 7 of the direction tuned neurons. Only 8 are shown out of the 40 neurons for clarity. (c) Gaussian-tuning functions for end-point tuned neurons.

The model was trained with 1000 trials (trajectories) per target where the duration of each trial was 2 sec: 1 sec to reach the target and 1 sec to return back to the center position. Excitatory connections between cooperative layer neurons were the only connections allowed to change as a result of learning. Figure 6.4b shows sample trajectories used during task learning. Every 2 sec, one of the 1000 trajectories was randomly selected from which the movement direction, movement speed, and end-point were extracted. These movement features were then used to modulate the firing of non-cooperative layer neurons according to equation (6-18). Figure 6.4c shows sample spike raster plots obtained in one trial for two adjacent targets 1 and 8 after the training phase. It can be seen that non-cooperative layer neurons exhibit clear task-dependent response modulation reminiscent of their tuning characteristics. In contrast, cooperative layer neurons do not exhibit noticeable changes in their firing rates but significant synchrony seems to be present.

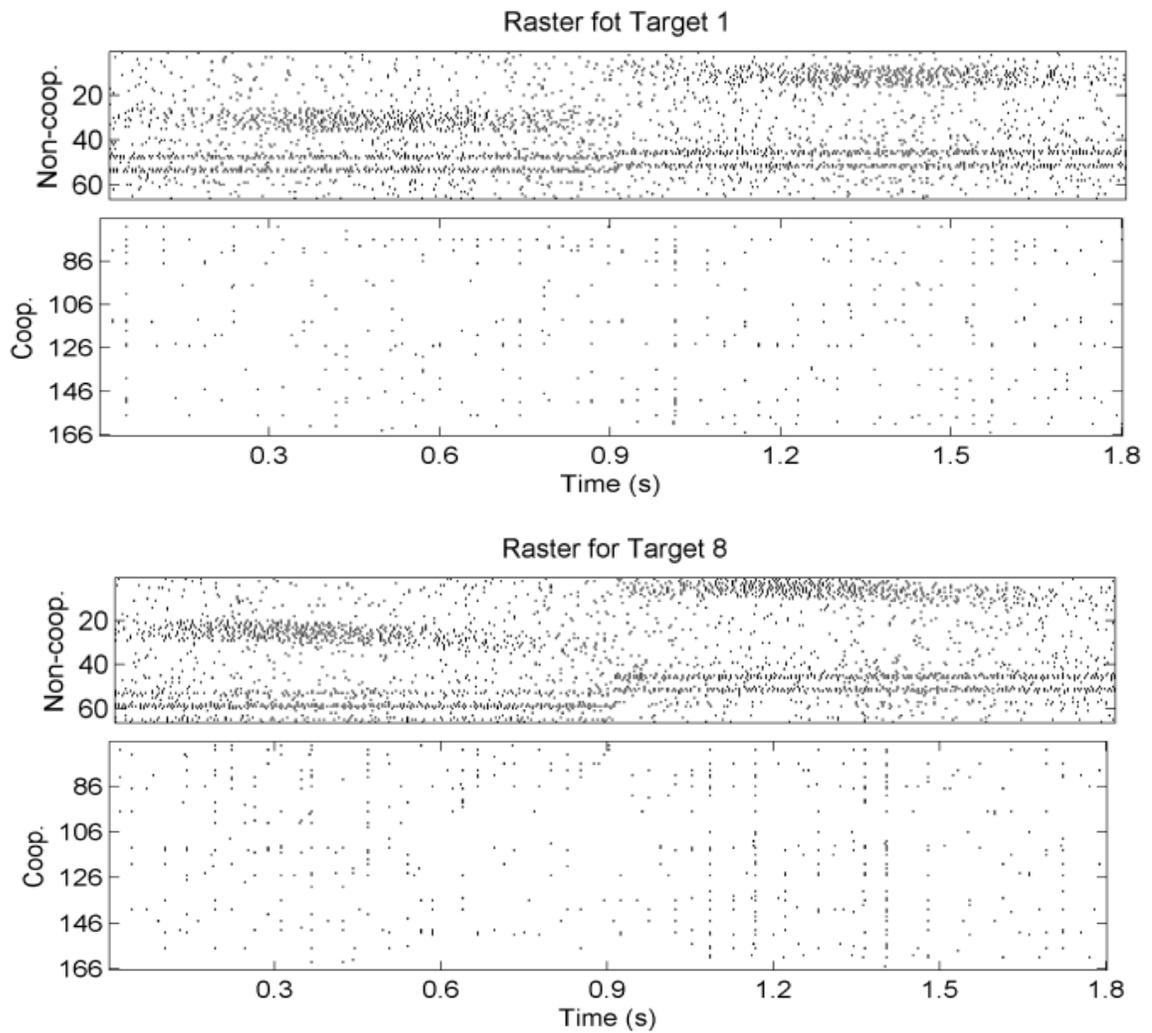
Figure 6.5a shows the average change in connection strength for both functionally correlated and uncorrelated neurons during the learning phase. As can be seen, the strength of the connections between functionally correlated neurons increases above the initial strength (0.01) while those of uncorrelated neurons decrease substantially over multiple training sessions. Figure 6.5b shows the connectivity matrices of cooperative layer neurons before and after learning. Sparse connectivity is apparent indicating that the learning phase *consolidated* the task information in a relatively small number of connections.



(a)



(b)



(c)

Figure 6.4 (a) Schematic of the statistical learning rule. Neurons A and B are non-cooperative layer neurons that excite neurons cooperative layer neurons C, D and E. The parameter  $l_{ik}$  is the synaptic

latency associated with each connection. Assuming  $l_{CA}$  greater than  $l_{DA}$ , neurons  $C$  and  $D$  were considered functionally correlated. Therefore, the only connection that is strengthened is  $C \rightarrow D$ . Dashed connections indicate connections that were expected to weaken over time. (b) Sample trajectories used to simulate the spike trains for the 8 targets. (c) Sample raster plot of the spike trains generated by the model corresponding to trajectories for targets 1 (top) and 8 (bottom) for one trial.

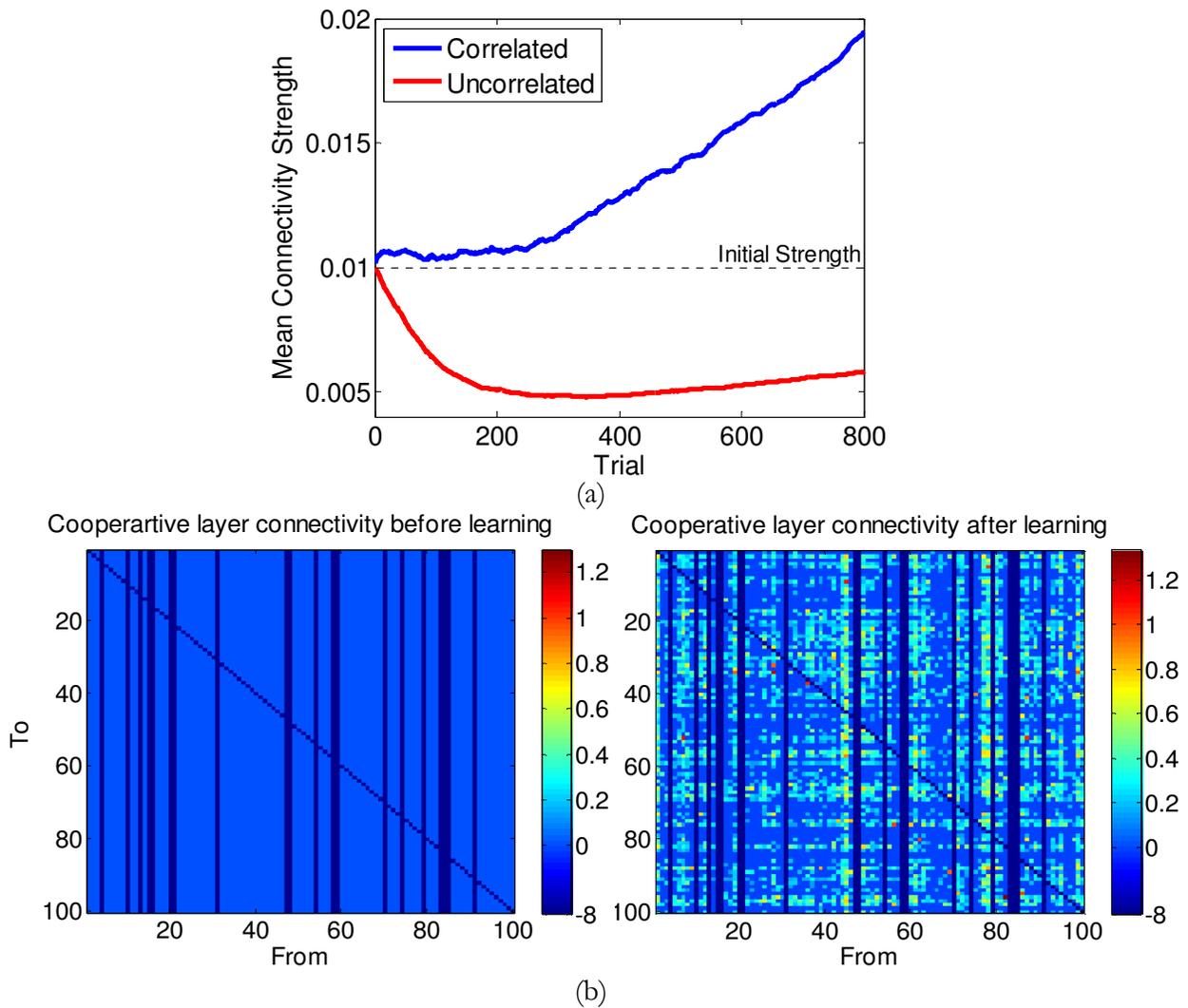


Figure 6.5 (a) Mean connectivity strength versus trial number for functionally correlated (blue) and uncorrelated (red) cooperative layer neurons. (b) Connectivity matrix for the cooperative layer

neurons before training (left) and after training (right). The columns represent pre-synaptic (transmitter) neurons while the rows represent post-synaptic (receiver) neurons.

An example of the statistics of input and output spike trains for sample neurons during the reach portion of the trial are shown in Figure 6.6. Cooperative layer neuron ‘103’ receives input from neurons ‘8’, ‘13’, ‘24’ (direction tuned) and ‘54’ (endpoint tuned). As can be seen in Figure 6.6f, neuron ‘103’ modulated its firing significantly when the movement was to targets ‘5’ or ‘8’ as depicted by the 10 ms bin Post-stimulus Time Histograms (PSTHs) averaged across 1000 trials/target [84]. On the other hand, for these two targets, two out of the four input neurons modulated their firing significantly (neurons ‘8’ and ‘13’ for target ‘5’ and neurons ‘24’ and ‘54’ for target ‘8’) as shown in Figure 6.6b-e. Because target ‘5’ is independently encoded by neurons ‘8’ and ‘13’ and both excite neuron ‘103’, the response modulation of neuron ‘103’ to target ‘5’ is much stronger than its response modulation to targets ‘4’ and ‘6’ combined. The later targets are independently encoded (with similar strength) by the same two neurons ‘8’ and ‘13’, respectively. A similar observation can be made to the response modulation of neuron ‘103’ to target 8 and that of its response to targets 1 and 7 relative to neurons ‘24’ and ‘54’, respectively.

To assess whether the correlation observed between cooperative layer neurons is an actual indicator of stimulus-driven interaction, we created a version of the data in which each spike was randomly jittered around its position over a uniform range of  $[-9, 9]$  msec. This keeps the firing rate of each neuron the same (i.e. similar individual response to the common input) but destroys the time-dependent correlation in the patterns. For a representative population of 10 cooperative layer neurons (neurons ‘99’ to ‘108’), Figure 6.7a shows the probability of the number of neurons firing within a 10ms window for both the actual data and the jittered data. As can be seen, the probability of two or more neurons firing within a 10 msec window for the jittered data becomes smaller than

that for the actual data indicating that the temporal coordination among the neurons is important to maintain the spatiotemporal pattern with higher order interactions and is destroyed by jittering.

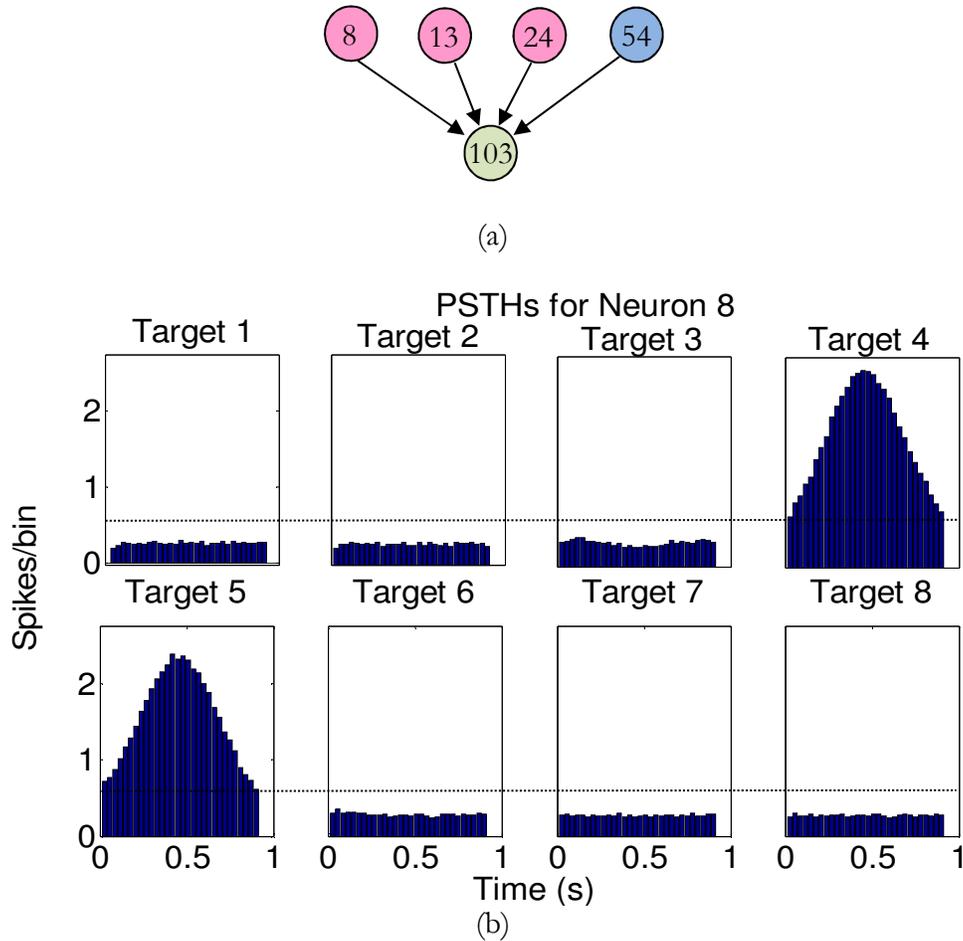
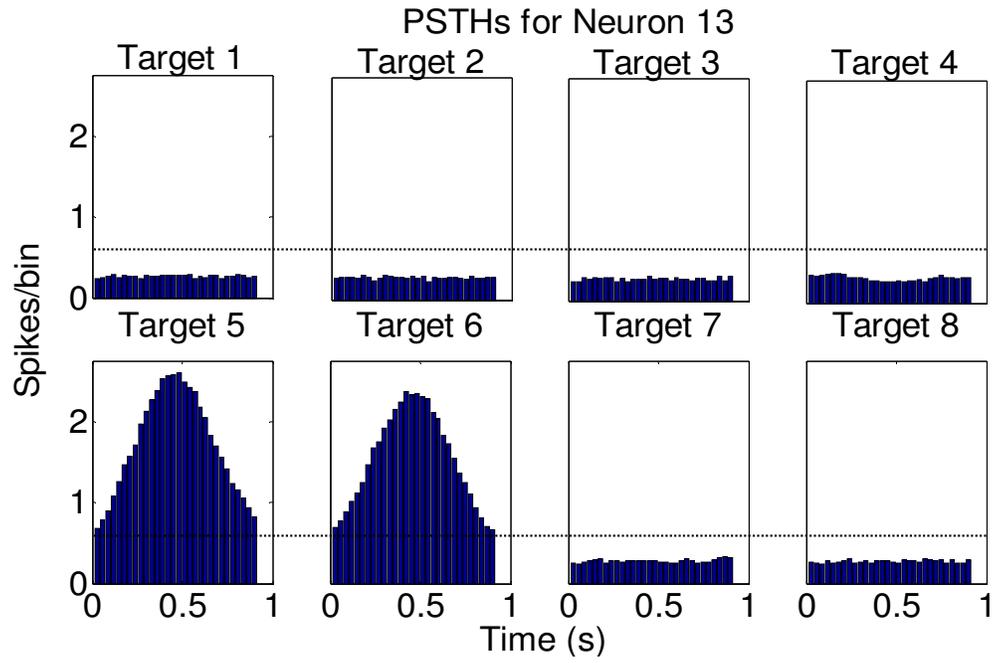
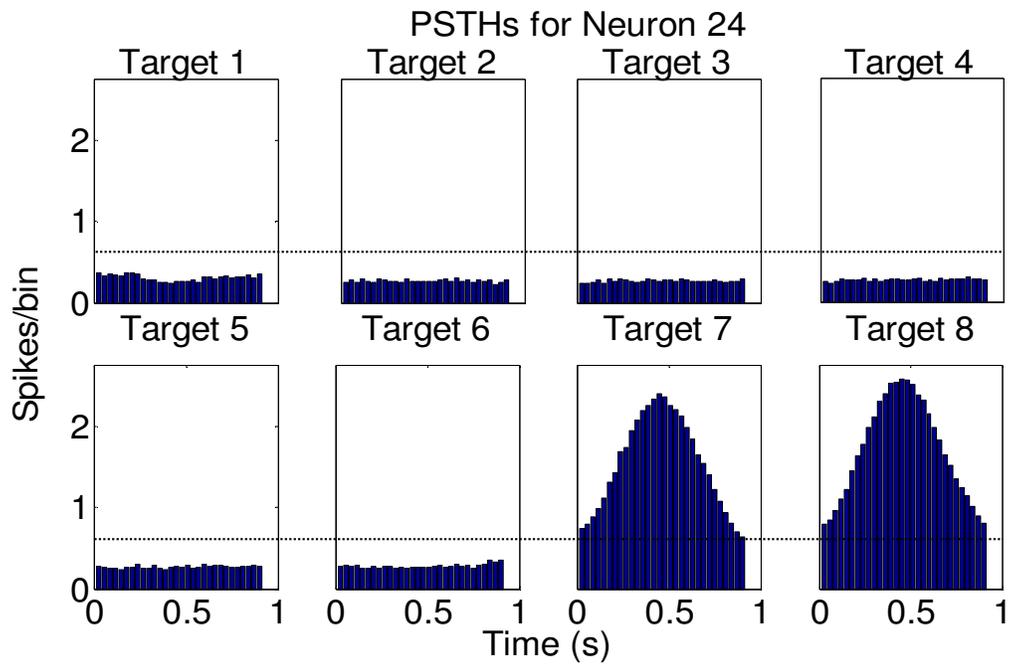


Figure 6.6 (a) An example of the connectivity between one cooperative layer neuron (neuron 103) and four non-cooperative layer neurons (8, 13, 24 and 54). Neurons 8, 13 and 24 are direction tuned while neuron 54 is end-point tuned. (b), (c), (d), (e) and (f): 10 ms bin PSTHs of neurons 8, 13, 24 and 54, and 103, respectively, for each of the 8 targets. Dashed lines represent the significance level computed by averaging the PSTHs of each neuron across all targets and bins.

Figure 6.6 (continued)

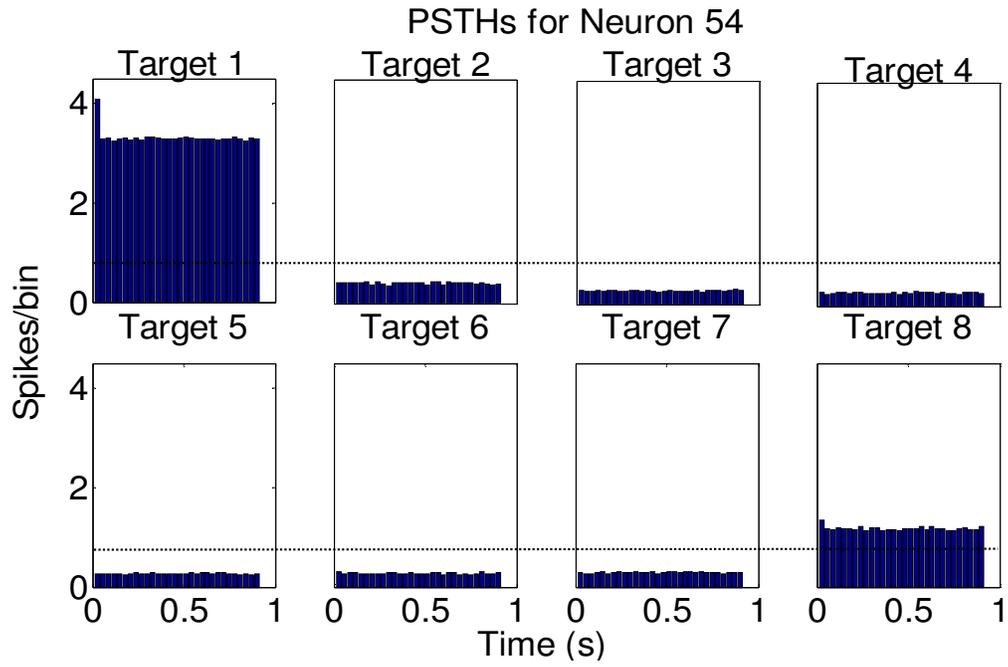


(c)

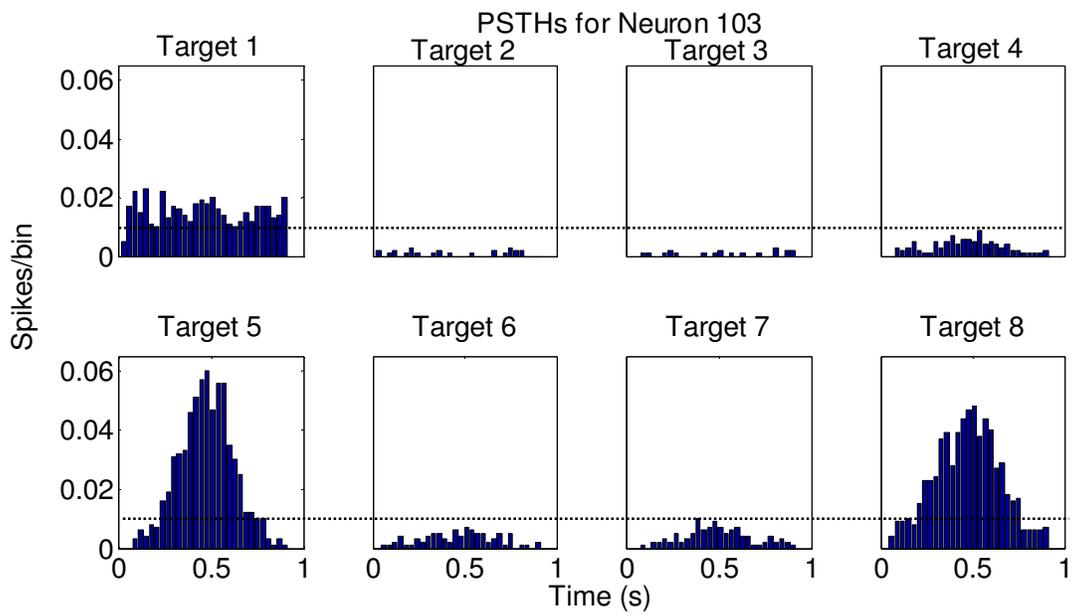


(d)

Figure 6.6 (continued)



(e)



(f)

Figure 6.7b shows the number of times each *specific* pattern of firing occurred in the jittered data versus the actual data. Each point in the figure represents one out of the 1024 possible patterns. The graph indicates that jittering the spike trains decreased the rate of occurrence of most of the patterns compared to the actual data. This confirms the observation in Figure 6.7a that higher order correlation in the data is more than what can be obtained from neurons acting independently.

## **6.3. Experimental Results**

### **6.3.1. Order of the Encoding Model**

After the motor task was learned, no more changes in the connection strengths were permissible and the obtained network topology was kept fixed for the rest of the analysis. 1000 trajectories per target were subsequently presented to the network and the corresponding spike trains were obtained. Figure 6.8a shows sample trial data from target 1, obtained from the cooperative layer neurons. Target-specific trials were then concatenated into a single response matrix,  $\mathbf{r}$ . To estimate the true entropy, the joint density was calculated as the rate of occurrence of distinct patterns. Figure 6.8b shows a sample pattern from a subpopulation of 10 neurons indicated by the box in Figure 6.8a.

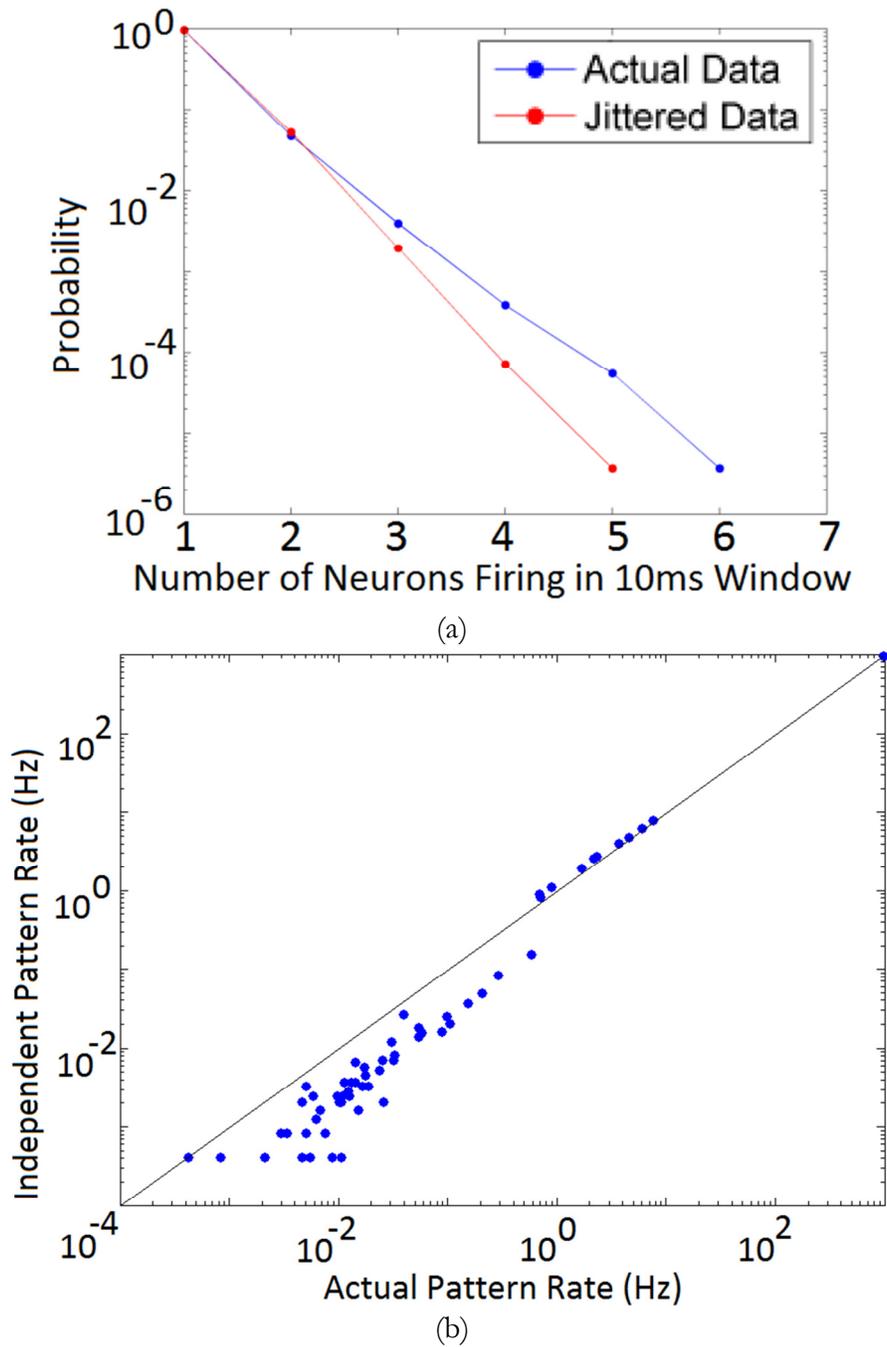


Figure 6.7 (a) Probability of number of neurons firing within a 10 ms window for actual data generated from the model (blue) and a jittered version of the data (red) for cooperative layer neurons 99 to 108. (b) Rate of occurrence of specific firing patterns within a 3 ms window (1 bin) for the jittered data versus the actual data generated from the model.

The order of the encoding model was estimated by determining how much information is intrinsic to each order of interactions. This was done by estimating the  $k^{th}$ -order connected information in (6-2) and comparing it to the multi-information in (6-1). For a subpopulation size of 10 neurons, the  $2^{nd}$ ,  $3^{rd}$ , and  $4^{th}$  order connected information is illustrated in Figure 6.9. It can be seen that the  $k^{th}$ -order connected information for  $k > 3$  is negligible, indicating that information in the model was intrinsic to independent, pairwise and triplet interactions among neurons, but no significant higher-order interactions beyond  $k = 3$  were contributing to the encoding model.

### 6.3.1. Model Consistency

The above analysis was repeated and the underlying network model was inferred for each of the eight targets using the MinED algorithm. Figure 6.10a illustrate two hypergraphs inferred for sample targets 1 and 8 for which the spike train data from a sample trial was shown in Figure 6.4c. It can be seen that a considerable number of second-order interactions exists, confirming the results reported in the literature using *in vitro* data [107]. Furthermore, a considerable portion of third-order interactions is present. Both account for more than 99.9% of the multi-information in the population activity.

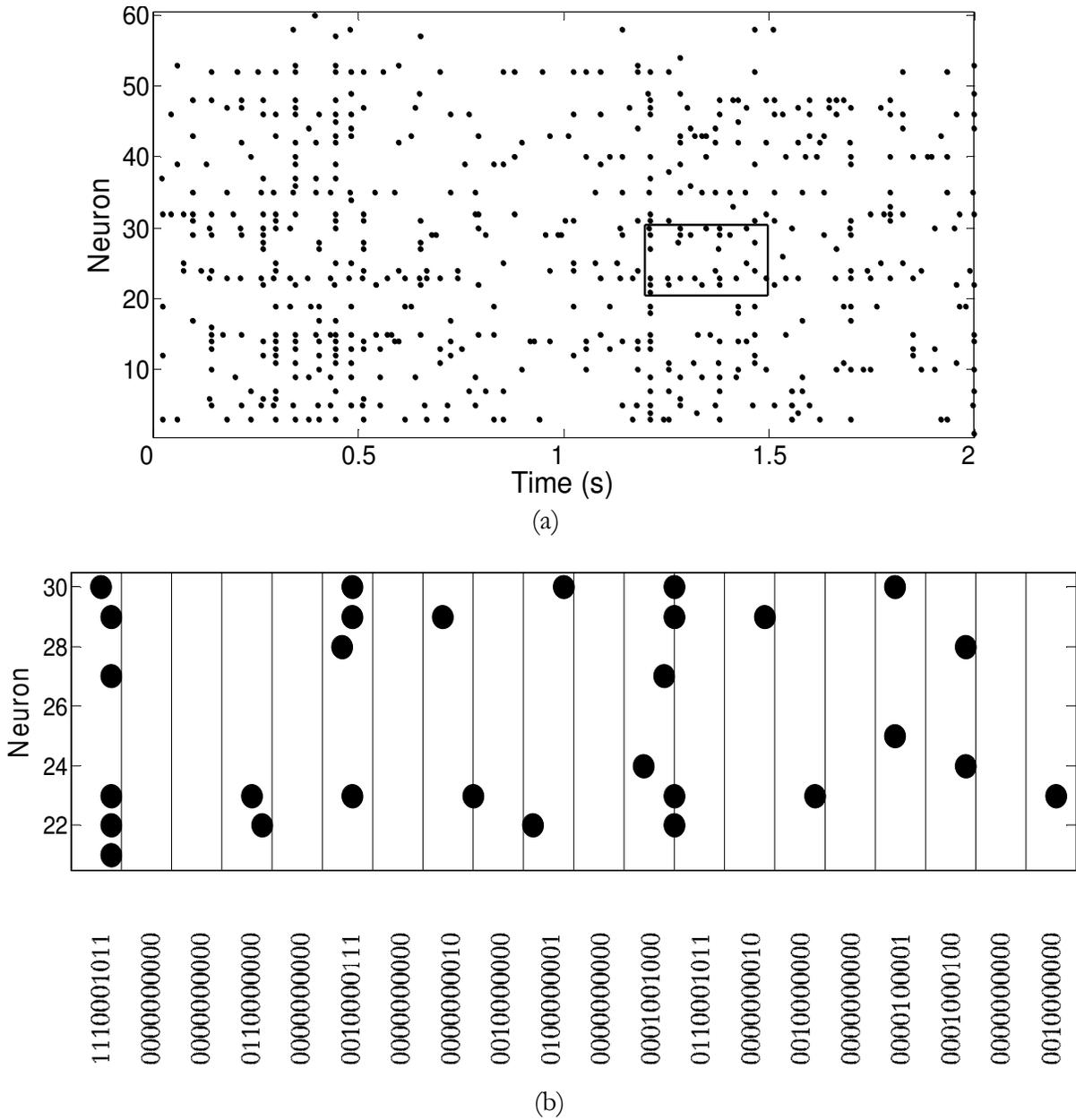


Figure 6.8 (a) Sample data from 60 cooperative layer neurons generated by the model during one trial after task learning was completed. (b) A binary representation of the network states shown by the box in (a). Every state describes the activity pattern of the population in one bin. Some states are repeated more often than others (for e.g., the silent state 000000000 is observed in 7 out of 20 times). Despite that we have  $2^n$  possible states, many states never occur.

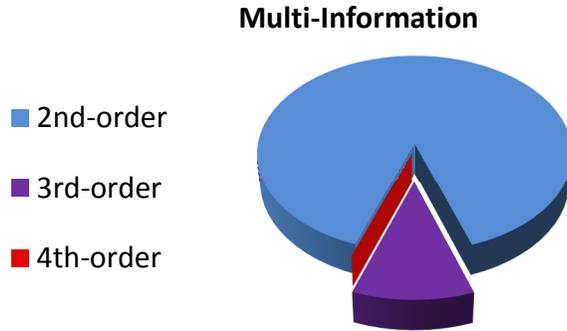


Figure 6.9 Connected information for different orders of the cooperative model for a fixed subpopulation size ( $n = 10$ ). The  $k^{\text{th}}$ -order connected information for  $k > 3$  is negligible compared to the 2<sup>nd</sup> and 3<sup>rd</sup>-orders. The sum of all connected information is equal to the multi-information.

To verify that the observed factors are reminiscent of stimulus-dependent higher order interactions, we computed crosscorrelograms between sample pairwise neurons that were found to have factors in the hypergraphs. Features in the correlograms can be used to determine if serial excitatory or inhibitory synaptic interaction exist between biological neurons [139]. Figures 6.10b and 6.10c show sample crosscorrelograms for neurons pair ('9', '10') with a bin size of 9ms computed from target '1' and target '8' trials, respectively. A single bin central peak at zero lag occurs in the crosscorrelogram for target '1' indicating a serial excitatory interaction [100]. This is not observed between the two neurons for target '8' data and perfectly agrees with the absence of a factor between the two neurons in the hypergraphs in Figure 6.10a. It is also noteworthy that the MinED algorithm extracted many additional pairwise factors that are not detected by the crosscorrelogram method.

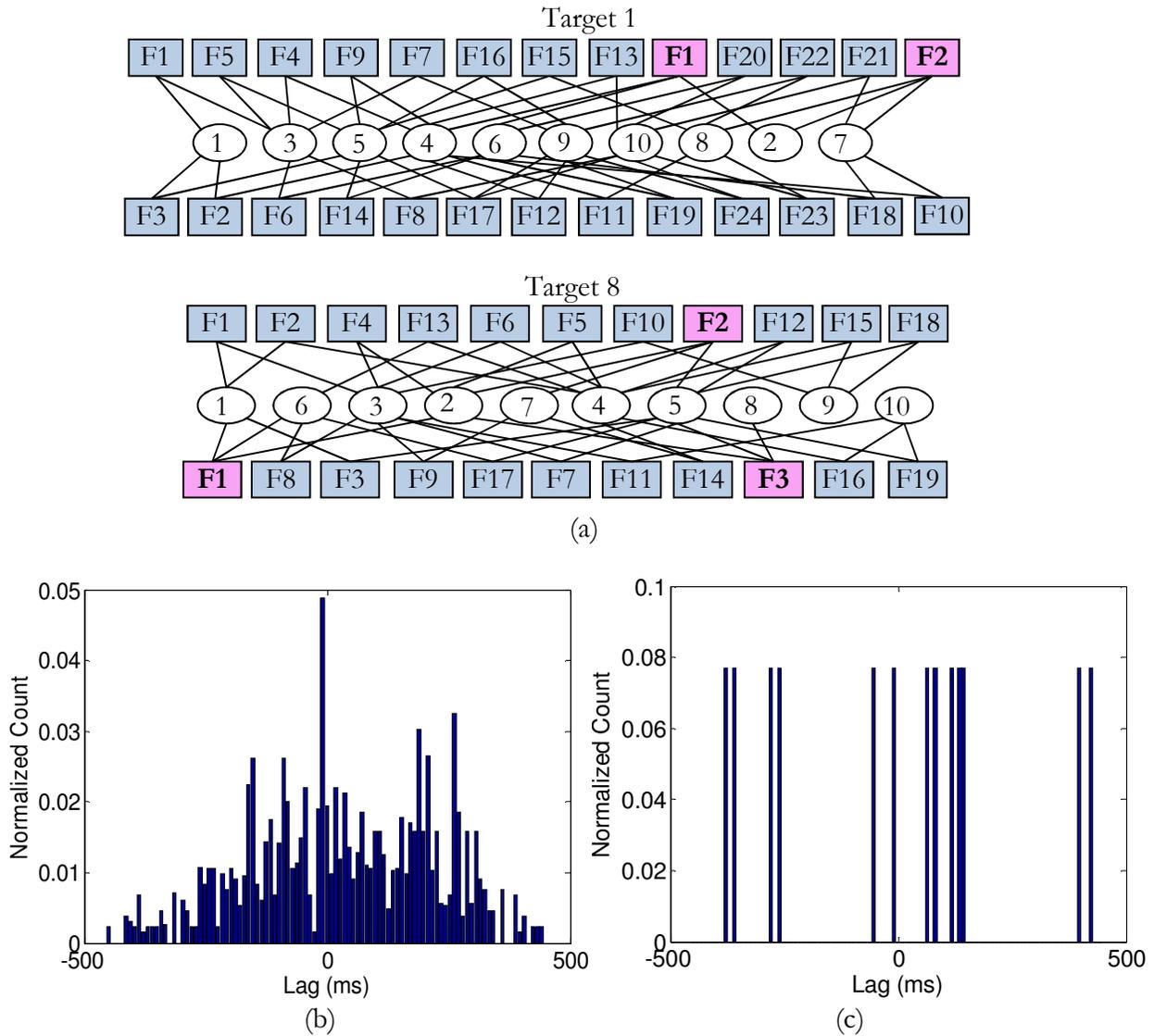


Figure 6.10. (a) Inferred factor-graphs for sample targets ‘1’, and ‘8’, respectively. Every ellipse represents a neurons and the cyan and purple squares represent second and third-order factors, respectively. It can be seen that a considerable number of 2<sup>nd</sup> interactions are present, confirming experimental results *in vitro*. Although the number of 3<sup>rd</sup>-order factors is considerably lower than the number of 2<sup>nd</sup>-order factors, it constitutes a significant portion of the multi information. (b), (c) Crosscorrelograms for neurons ‘9’ and ‘10’ with a bin size of 9ms for target ‘1’ and target ‘8’, respectively. A significant peak occurs in the crosscorrelogram for target ‘1’ indicating an excitatory interaction between the pair that is not observed for target ‘8’.

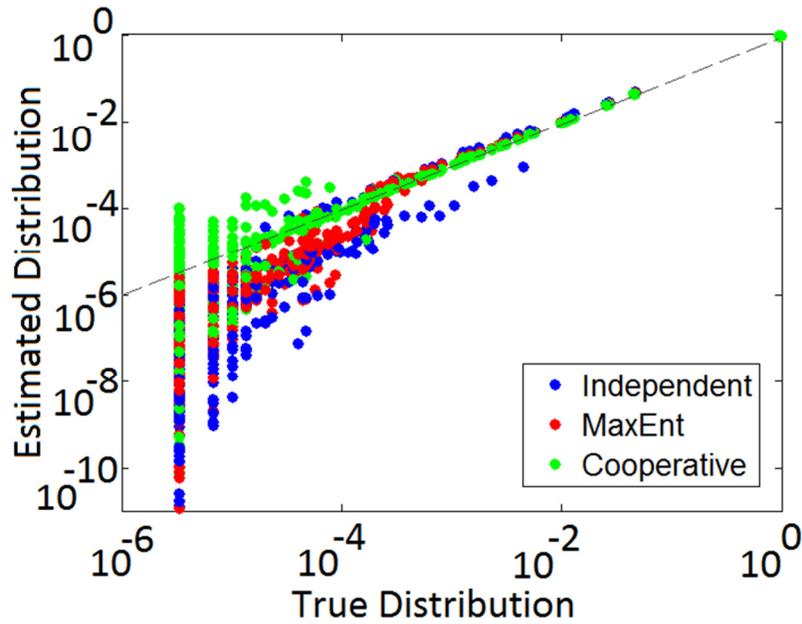


Figure 6.11 Model fit for a sample subpopulation of 10 neurons for: 1- Independent model (blue), 2- Maximum entropy model (red), 3- MinED cooperative model (green). The true density is the measured rate of occurrence for distinct network states while the estimated density is the rate predicted by the probabilistic graphical model.

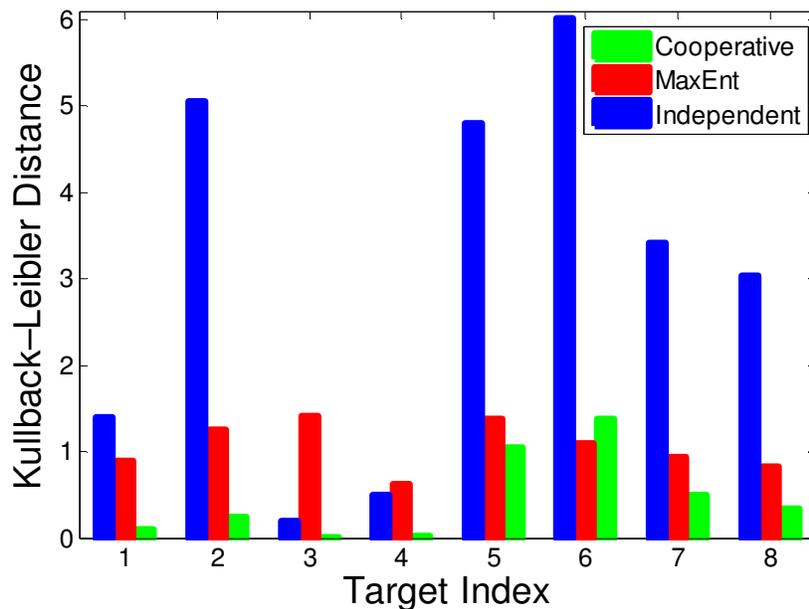


Figure 6.12 The KL distance between the true and the single trial distribution of states for each model. It can be seen that on average the MinED cooperative model results in the smallest distance.

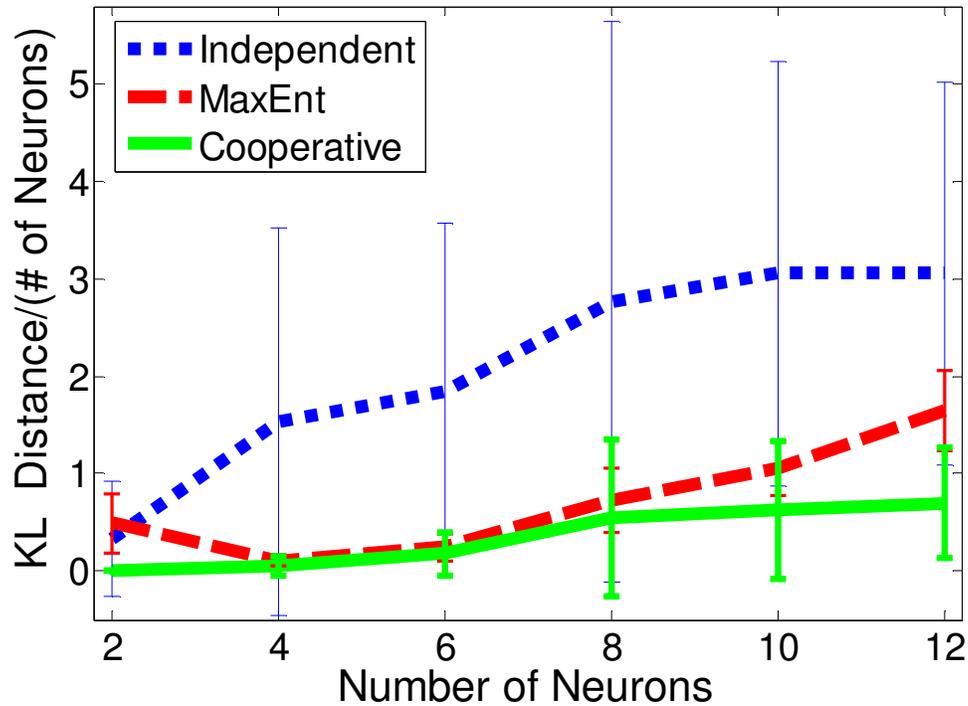


Figure 6.13 The KL-divergence for independent, maximum entropy and cooperative models as a function of the size of the observed population. The distance is normalized by the number of neurons. Error bars represent standard deviation.

Figure 6.11 illustrates the model goodness of fit for the independent, Maximum Entropy and MinED cooperative models. Further departure from the reference line (solid) would indicate the data from the two models were generated from increasingly distinct distributions. According to Figure 6.11, the MinED model is the closest to the reference line, further demonstrating its superiority in capturing the intrinsic higher order interactions between the neurons. This is further demonstrated using the distance between the true and the estimated densities displayed in Figure 6.12 for each of the eight targets.

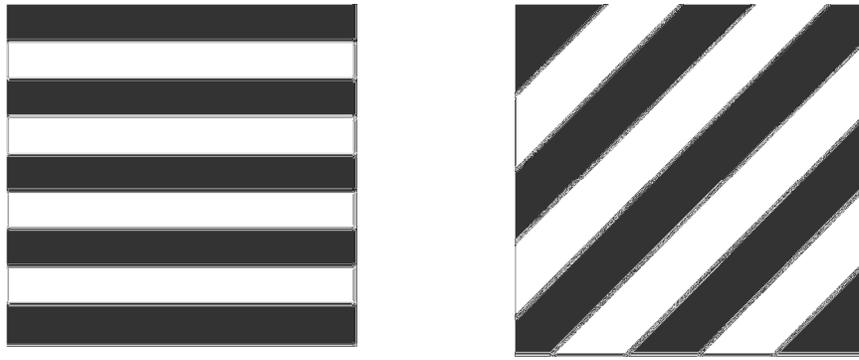
To demonstrate the consistency of the MinED cooperative model as a function of the

population size, Figure 6.13 shows the average normalized Kullback-Leibler (KL) distance for the different models as a function of the observed population size. The error for the cooperative model remains bounded as the number of neurons increase compared to a larger error for the independent and maximum entropy models. One possible explanation is that the probability of observing neurons with significant overlaps in their tuning characteristics increases as the population size increases. This means that the output correlation becomes dominated by the input correlation rather than local interaction between cooperative layer neurons. In such case, the maximum entropy and independent models tend to exhibit some redundancy in their encoding characteristics.

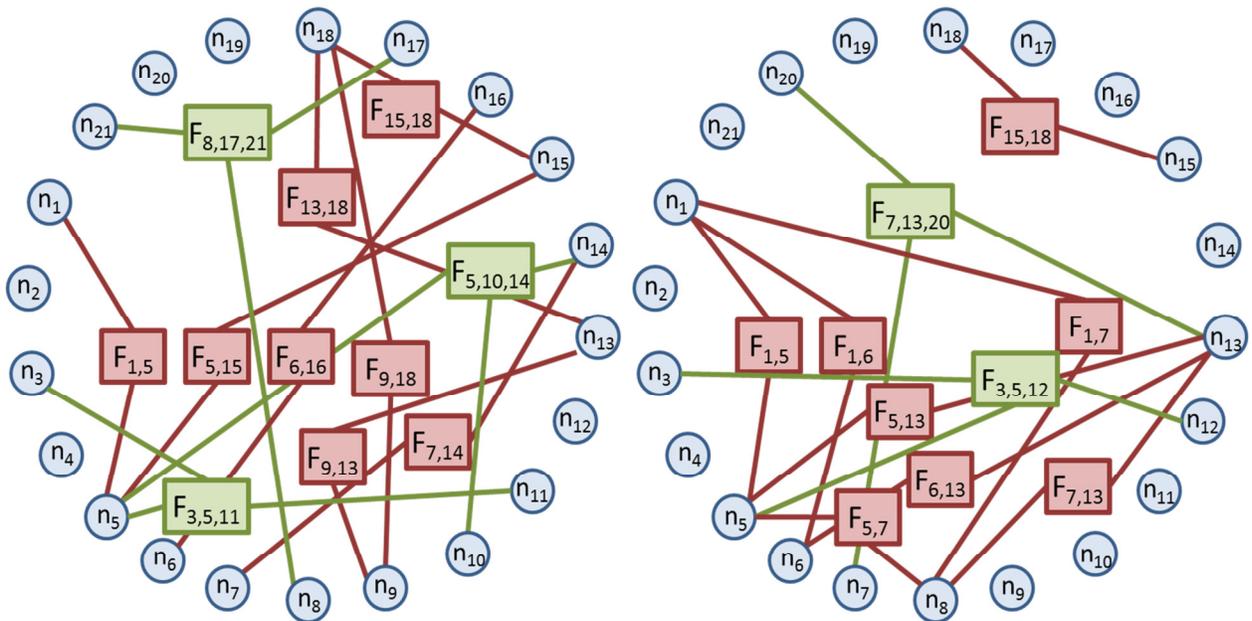
Figure 6.14b demonstrates different hypergraph models obtained for the population responses of V1 data in response to different inputs of drifting grating stimulus (chapter 5), demonstrated in Figure 6.14a. Similar to the hypergraphs of the synthetic motor encoding model in Figure 6.10a, different inputs result in different hypergraphs.

## 6.1. Discussion

As tools for large-scale recording of neural activity in the brain continue to excel, some challenging questions about the nature of the brain's complex information processing naturally arise. Information theory is increasingly becoming popular to address many of these questions. In this chapter, we mainly addressed the problem of how to discover stimulus-dependent spatiotemporal patterns of activity using an information theoretic approach. These patterns may indicate how cooperation among neurons can provide a near-optimal mechanism for encoding external stimuli.



(a)



(b)

Figure 6.14 (a) Two sample inputs of drifting grating stimulus (b) Hypergraph models for the 21 neurons for the two sample inputs in (a). Red squares represent factors of second-order interactions, and green squares represent factors of third-order interactions.

Using factor graphs, we have demonstrated that a network model can best fit single trial data when connection constraints are guided by information available from the concatenation of multiple trials. Generally speaking, limited data constitute a significant challenge [128], particularly for large populations, because the dimension of the response space grows exponentially with the number of

neurons. In such a case, an infinite amount of data would be needed to completely characterize the joint distribution. The density estimates can be severely biased when only limited data is available. Our approach therefore attempts to circumvent this problem by finding the graphical model that explains single trial data and can be simultaneously extrapolated to account for unlimited data size, if available.

Using a model of motor encoding, we demonstrated that task information can be transformed from a non-cooperative network output with independent firing to a cooperative network interaction using a spike timing dependent plasticity learning rule. This biologically plausible mechanism accounts for the ordered set of event occurrences and therefore captures temporal correlations between neuronal states that are caused by delayed synaptic interactions between neurons. Our analysis has revealed that a substantial amount of information in the model is represented in 2nd-order interactions and is efficiently represented by cliques in the graph structures, confirming the results obtained by other groups. However, our results also demonstrate that 3rd-order interactions are present and contribute significantly to the multi-information. These results were not surprising, given the model's dependence on the variable-length history of interaction that contributed to the emergence of temporal patterns of network states. In addition, spatial patterns between neurons have emerged as a result of task learning. This is a component strongly believed to underlie cortical network organization [2].

In our analysis, we have mainly focused on the development of the algorithm and therefore restricted our analysis to spike train data from simulated motor cortex neurons with inhomogeneous Poisson firing characteristics. These models generate neural responses that share a number of features with experimental data. First, it has been shown that cortical neurons can simultaneously encode multiple parameters (e.g. direction and endpoint [27]), reflecting the complex anatomical and

functional organization of the cortical neural circuits. Second, synaptic efficacy has been hypothesized to be mediated by the precise timing of the presynaptic (transmitter) neuron firing and that of the postsynaptic (receiver) neuron [134, 138]. It is widely accepted that synaptic modification plays an important role in learning and memory formation [91]. Third, neurons in our model found to exhibit higher order interaction were driven by discrete neuronal inputs (spike trains). For these reasons, we believe that our encoding model mimics to some extent a biological “information transform” that may take place in an actual spiking neural system. Our model, however, did not include an important element known to guide motor behavior: visual feedback. We have chosen to simulate a task (the center-out-reach task) that minimally relies on this feedback mechanism to guide reach movements (as opposed to, for example, encoding a task with freely moving target which would require continuous adaptation of motor commands).

We also found a strong agreement between the statistics of our encoding model data and related models that have been applied to RGCs in vitro. Specifically, we found that 2nd-order interaction seem to be predominantly present, a result observed by a multitude of studies (e.g. [72, 75, 79, 94, 107]). We also found that our biological learning rule result in network connectivity that is relatively sparse, in agreement with some recent experimental results during direct activation of biological cortical networks [140].

As our results demonstrated, the MinED approach is superior to the maximum entropy model in inferring the underlying model structure, particularly for relatively large size systems. The maximum entropy model finds the network realization that provides the maximum entropy with respect to the testable information (i.e. known constraints such as firing rate and pairwise correlation). This limits its application to small sized networks in which the true distribution can be measured with a high degree of accuracy. Our MinED approach, on the other hand, was developed

to optimally explain single trial data with knowledge of network behavior from a concatenation of a large number of trials. This has the advantage of capturing information about transition between specific sequences of network states, a feature not accounted for by the maximum entropy model. We therefore expect that the MinED approach will yield superior performance in quantifying spatiotemporal patterns of activity that are frequently observed in cortical systems in vivo. Full analysis of real experimental neural data with our approach is currently underway.

We believe this framework can be extremely useful in improving our understanding of brain circuits, and in developing efficient decoders for Brain Machine Interface applications [141]. Despite our focus on spiking neural data analysis, the framework is applicable to many other fields, for example, in gene regulatory networks where the collective role of multiple genes needs to be assessed to characterize an observed phenotype. It can also be useful in the development of sensor network coding strategies in which sensors optimally cooperate to encode signals from the surrounding to optimize their limited resources. In some sense, one can argue that natural brain encoding mechanisms can be extremely insightful in the design of efficient network codes.

# 7

## Concluding Remarks

In this dissertation we have discussed two main steps in analyzing neural data, spike sorting and spike train analysis. Although these are fundamentally separate algorithms, the latter one builds upon the outcome of the first one, and as a result, its success in representing graphical models for population encoding heavily depends on the sorting performance [142]. Falsely detected noise events and missed spikes basically alter the estimations of the empirical moments and the true entropy of the population response, and therefore can affect our estimations on the parameters of the spatiotemporal maximum entropy and hypergraph models.

As discussed early in chapter 3, spike sorting is important but computationally prohibitive in BMI systems. As a result, not all BMI applications feature spike sorting, or simply avoid its extensive computations by classifying events based on their maximum amplitudes via thresholding the raw data. For example, implantable devices designed for transmitting wirelessly the activity of a large number of electrodes only transmit a small interval of action potentials that pass a detection threshold, and rely on each event's maximum amplitude for sorting [143-145]. Although we have shown the effect of poor or lack of spike sorting on the decoding performance in Figure 3.8, some neural decoders rely on higher-level analysis on the encoding characteristics of detected units to compensate for the loss of performance [146]. Therefore, the need for spike sorting is application

dependent, and must be realized considering its computational cost versus the increased performance and reliability gained.

Setting aside the computational cost of spike sorting algorithms, it is still a challenging job with respect to making a decision on the number of units recorded on an electrode and determining the decision boundaries that separate these units. Although, tetrodes that include four closely located electrodes can assist spike sorting algorithms in making a decision on the number of units given the spatial propagation of spike waveforms [147-149], the challenge of determining discriminative borders remains. Including the encoding characteristics of units, if stationary, might help. These encoding characteristics can be the inter-spike interval (ISI) [150], the conditional firing rate [151], or the neuronal tuning function [152]. This additional information can provide reliable performances and can be used for tracking stable events over consecutive days, provided they don't change as a result of neuroplasticity [153].

In this dissertation we have introduced the compressive spike sorting (CSS) to directly sort units in a sparse domain obtained by thresholding multilevel wavelet coefficients. We have also introduced an iterative learning algorithm to find the optimal set of thresholds for the sparse representation. CSS has three major benefits. First, while the discriminative features necessary for a high sorting performance are preserved, significant data compression is obtained. Second, the amount of computations required for sorting directly in the sparse representation is less than conventional spike sorting algorithms, and therefore can be efficiently implemented in low-power, small size electronics. Third, the sorting performance is less sensitive to low SNR recordings and can be used to track units over consecutive days of recording.

For spike train analyses a high performance spike sorting was required, without its

computational power being an issue. Although the term ‘functional connectivity’ was used to refer to the graphical models fit to the conditional distribution of the population response given the stimulus, the connectivity here is different by all means with a physical synaptic connection between neurons. The graphical models have three main applications. One is in the prediction of future population responses given a sequence of observed neural responses. The second is in neural decoding using a maximum a posteriori (MAP) estimator. In this application, the likelihood function is factorized into marginal and pairwise distributions using the graphical model realization, thus providing a more reliable estimation given the limited training data. The third is in linking specific models to certain behavioral events. The spatiotemporal graphical models introduced in this dissertation have proven to be more efficient in all three applications compared to typical independent models, and also instantaneous maximum entropy models that don’t incorporate temporal correlations among neuronal constituents.

While the probability of having a physical connection between two neurons randomly selected from a small localized area of the brain is very small, algorithms introduced in chapters 5 and 6 tend to find numerous statistical interactions among recorded neurons, i.e. the maximum entropy model will find a nonzero value for all unknown parameters. Although there is a small chance that this connectivity is related to an actual physical connection directly or indirectly through hidden neurons, but it is more likely that it originates from neurons responding to a common input, i.e. overlapping receptive fields for sensory neurons or overlapping tuning functions for motor neurons. Whatever the source, we are interested in limiting the number of nonzero elements of the inferred connectivity for the sake of model simplification and avoiding overfitting. This automatically suggests a sparse optimization algorithm for learning the unknown parameters by obtaining graphical models with the minimum number of nonzero elements [154, 155].

## REFERENCES

## REFERENCES

1. Kandel, E., J. Schwartz, and T. Jessell, *Principles of neural science*. 2000, 2000, McGraw-Hill, New York.
2. Hebb, D.O., *The Organization of Behavior; a Neuropsychological Theory*. 1949: Wiley, New York.
3. Wise, K., et al., *Wireless Implantable Microsystems: High-Density Electronic Interfaces to the Nervous System*. Proceedings of the IEEE, 2004. **92-1**: p. 76–97.
4. Normann, R., et al., *A neural interface for a cortical vision prosthesis*. Vision research, 1999. **39** (15): p. 2577-2587.
5. Buzsaki, G., *Large-scale recording of neuronal ensembles*. Nat Neurosci, 2004. **7**(5): p. 446-451.
6. Nicolelis, M.A.L., *Methods for Neural Ensemble Recordings*. 1999: CRC Press.
7. Abbott, L.F. and S.B. Nelson, *Synaptic plasticity: taming the beast*. Nature Neurosci. , 2000. **3** (suppl.): p. 1178-1183.
8. Barnes, A.P. and F. Polleux, *Establishment of axon-dendrite polarity in developing neurons*. Annual review of neuroscience, 2009. **32**: p. 347.
9. Achermann, P. and A. Borbely, *Low-frequency (< 1 Hz) oscillations in the human sleep electroencephalogram*. Neuroscience, 1997. **81**(1): p. 213.
10. Jurkiewicz, M.T., et al., *Sensorimotor Cortical Plasticity During Recovery Following Spinal Cord Injury: A Longitudinal fMRI Study*. Neurorehabilitation and Neural Repair, 2007. **21**(6): p. 527.
11. Armstrong-Gold, C.E. and F. Rieke, *Bandpass Filtering at the Rod to Second-Order Cell Synapse in Salamander (*Ambystoma tigrinum*) Retina*. J. Neurosci., 2003. **23**(9): p. 3796-3806.
12. Stark, E. and M. Abeles, *Predicting Movement from Multiunit Activity*. J. Neurosci., 2007. **27**(31): p. 8387-8394.

13. Frohlich, Z., *Combined spatial and temporal imaging of brain activity during visual selective attention in humans*. Nature, 1994. **372**: p. 8.
14. Normann, R.A., et al., *High-resolution spatio-temporal mapping of visual pathways using multi-electrode arrays*. Vision research, 2001. **41**(10): p. 1261-1275.
15. Palmer, C., *A microwire technique for recording single neurons in unrestrained animals*. Brain research bulletin, 1978. **3**(3): p. 285-289.
16. Najafi, K. and K.D. Wise, *An implantable multielectrode array with on-chip signal processing*. Solid-State Circuits, IEEE Journal of, 1986. **21**(6): p. 1035-1044.
17. Lewicki, M.S., *A review of methods for spike sorting: the detection and classification of neural action potentials*. Network: Computation in Neural Systems, 1998. **9**(4): p. 53-78.
18. Georgopoulos, A.P., A.B. Schwartz, and R.E. Kettner, *Neuronal population coding of movement direction*. Science, 1986. **233**(4771): p. 1416.
19. Oweiss, K.G. and D.J. Anderson, *A new technique for blind source separation using subband subspace analysis in correlated multichannel signal environments*. Acoustics, Speech, and Signal Processing, 2001. Proceedings (ICASSP'01). 2001 IEEE International Conference on, 2001. **5**.
20. Oweiss, K. and M. Aghagolzadeh, *Detection and Classification of Extracellular Action Potential Recordings*. Statistical Signal Processing for Neuroscience and Neurotechnology, KG Oweiss, Ed., Academic Press, Elsevier, 2010: p. 15-74.
21. Oweiss, K.G., *A systems approach for data compression and latency reduction in cortically controlled brain machine interfaces*. IEEE Transactions on Biomedical Engineering, 2006. **53**(7): p. 1364-1377.
22. Bishop, C.M., *Pattern recognition and machine learning*. Vol. 4. 2006: springer New York.
23. Dinning, G.J. and A.C. Sanderson, *Real-time classification of multiunit neural signals using reduced feature sets*. Biomedical Engineering, IEEE Transactions on, 1981(12): p. 804-812.
24. Dempster, A.P., N.M. Laird, and D.B. Rubin, *Maximum likelihood from incomplete data via the EM algorithm*. Journal of the Royal Statistical Society. Series B (Methodological), 1977: p. 1-38.

25. Bezdek, J.C., R. Ehrlich, and W. Full, *FCM: The fuzzy  $c$ -means clustering algorithm*. Computers & Geosciences, 1984. **10**(2): p. 191-203.
26. Suminski, A.J., et al., *Incorporating feedback from multiple sensory modalities enhances brain-machine interface control*. The Journal of Neuroscience, 2010. **30**(50): p. 16777-16787.
27. Carmena, J.M., et al., *Learning to control a brain-machine interface for reaching and grasping by primates*. PLoS Biology, 2003. **1**(2): p. e2.
28. Serruya, M.D., et al., *Brain-machine interface: Instant neural control of a movement signal*. Nature, 2002. **416**(6877): p. 141-142.
29. Taylor, D.M., S.I.H. Tillery, and A.B. Schwartz, *Direct Cortical Control of 3D Neuroprosthetic Devices*. Science, 2002. **296**(5574): p. 1829.
30. Kass, R.E., V. Ventura, and C. Cai, *Statistical smoothing of neuronal data*. Network Computation in Neural Systems, 2003. **14**(1): p. 5-15.
31. Paulin, M.G. and L.F. Hoffman, *Optimal firing rate estimation*. Neural Netw, 2001. **14**(6-7): p. 877-81.
32. Oweiss, K.G., *Multiresolution Analysis of Multichannel Neural Recordings in the Context of Signal Detection, Estimation, Classification and Noise Suppression*, 2002, University of Michigan.
33. Oweiss, K.G. and D.J. Anderson, *Tracking Signal Subspace Invariance for Blind Separation and Classification of Nonorthogonal Sources in Correlated Noise*. EURASIP Journal on Advances in Signal Processing}, 2007. **2007**.
34. Brillinger, D., *The Identification of Point Process Systems*. Annals of Probability, 1975. **3**(6): p. 909-924.
35. Brown, E.N., ed. *Theory of Point Processes for Neural Systems Methods and Models in Neurophysics*, ed. C.C. Chow, et al. 2005, Elsevier: Paris. 691-726.
36. Brillinger, D., *Nerve Cell Spike train Data Analysis*. J. Am. Stat. Assoc., 1992. **87**(418): p. 260-271.
37. Kaneoke, Y. and J.L. Vitek, *Burst and oscillation as disparate neuronal properties*. Journal of Neuroscience Methods, 1996. **68**(2): p. 211-223.

38. Goldberg, J.A., et al., *Enhanced Synchrony among Primary Motor Cortex Neurons in the 1-Methyl-4-Phenyl-1, 2, 3, 6-Tetrahydropyridine Primate Model of Parkinson's Disease*. Journal of Neuroscience, 2002. **22**(11): p. 4639.
39. Churchland, M.M. and K.V. Shenoy, *Temporal Complexity and Heterogeneity of Single-Neuron Activity in Premotor and Motor Cortex*. Journal of Neurophysiology, 2007. **97**(6): p. 4235.
40. Shadlen, M.N. and W.T. Newsome, *The Variable Discharge of Cortical Neurons: Implications for Connectivity, Computation, and Information Coding*. Journal of Neuroscience, 1998. **18**(10): p. 3870-3896.
41. Galan, R.F., G.B. Ermentrout, and N.N. Urban, *Optimal Time Scale for Spike-Time Reliability: Theory, Simulations, and Experiments*. J Neurophysiol, 2008. **99**(1): p. 277-283.
42. Kass, R.E. and V. Ventura, *Spike Count Correlation Increases with Length of Time Interval in the Presence of Trial-to-Trial Variation*. Neural Computation, 2006. **18**(11): p. 2583.
43. Parzen, E., *On estimation of a probability density function and mode*. Annals of Mathematical Statistics, 1962. **33**(3): p. 1065-1076.
44. Cherif, S., K.E. Cullen, and H.L. Galiana, *An improved method for the estimation of firing rate dynamics using an optimal digital filter*. J Neurosci Methods, 2008.
45. Candes, E.J., J. Romberg, and T. Tao, *Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information*. Information Theory, IEEE Transactions on, 2006. **52**(2): p. 489-509.
46. Oweiss, K.G., *Compressed and Distributed Sensing of Multivariate Neural Point Processes*. IEEE International Conference on Acoustics, Speech and Signal Processing, 2007, 2007. **2**.
47. Golub, G.H. and C.F. Van Loan, *Matrix Computations*. 1996: Johns Hopkins University Press.
48. Oweiss, K.G., et al., *A Scalable Wavelet Transform VLSI Architecture for Real-Time Signal Processing in High-Density Intra-Cortical Implants*. IEEE Transactions on Circuits and Systems I, 2007. **54**(6): p. 1266-1278.
49. Tan, P.N., M. Steinbach, and V. Kumar, *Introduction to Data Mining*. 2006: Addison-Wesley.

50. Donoho, D.L., *De-noising by soft-thresholding*. Information Theory, IEEE Transactions on, 1995. **41**(3): p. 613-627.
51. Georgopoulos, A.P., et al., *Reply to Kurtzer and Herter*. Journal of Neurophysiology, 2007. **97**(6): p. 4391-4392.
52. Cover, T.M. and J.A. Thomas, *Elements of Information Theory*. 2006: Wiley-Interscience New York.
53. Warland, D.K., P. Reinagel, and M. Meister, *Decoding Visual Information From a Population of Retinal Ganglion Cells*. Journal of Neurophysiology, 1997. **78**(5): p. 2336-2350.
54. Kamboh, A.M., et al., *Area-Power Efficient Lifting-Based DWT Hardware for Implantable Neuroprosthetics*. Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on, 2007: p. 2371-2374.
55. Srinivasan, L., et al., *A State-Space Analysis for Reconstruction of Goal-Directed Movements Using Neural Signals*. Neural Comp., 2006. **18**(10): p. 2465-2494.
56. Santhanam, G., et al., *A high-performance brain-computer interface*. Nature, 2006. **442**(7099): p. 195–198.
57. Hatsopoulos, N., et al., *At what time scale does the nervous system operate*. Neurocomputing, 2003. **52**(54): p. 25-29.
58. Riehle, A., et al., *Dynamical changes and temporal precision of synchronized spiking activity in monkey motor cortex during movement preparation*. Journal of Physiology (Paris), 2000. **94**: p. 569-82.
59. Oram, M.W., et al., *Excess Synchrony in Motor Cortical Neurons Provides Redundant Direction Information With That From Coarse Temporal Measures*. Journal of Neurophysiology, 2001. **86**(4): p. 1700-1716.
60. Eldawlatly, S., R. Jin, and K. Oweiss, *Identifying Functional Connectivity in Large Scale Neural Ensemble Recordings: A Multiscale Data Mining Approach*. Neural Computation, 2008. **20**: p. 1-28.
61. Biran, R., D.C. Martin, and P.A. Tresco, *The brain tissue response to implanted silicon microelectrode arrays is increased when the device is tethered to the skull*. J Biomed Mater Res A, 2007. **82**(1): p. 169-78.

62. Wu, X. *High-order context modeling and embedded conditional entropy coding of wavelet coefficients for image compression*. in *Signals, Systems & Computers, 1997. Conference Record of the Thirty-First Asilomar Conference on*. 1997. IEEE.
63. Hauck, E.L., *Data compression using run length encoding and statistical encoding*, 1986, Google Patents.
64. Donoho, D.L., *Compressed Sensing*. *Information Theory, IEEE Transactions on*, 2006. **52**(4): p. 1289-1306.
65. Candès, E.J., *The restricted isometry property and its implications for compressed sensing*. *Comptes Rendus Mathematique*, 2008. **346**(9): p. 589-592.
66. Rauhut, H., K. Schnass, and P. Vandergheynst, *Compressed sensing and redundant dictionaries*. *Information Theory, IEEE Transactions on*, 2008. **54**(5): p. 2210-2219.
67. Johnson, K.O., *Sensory discrimination: decision process*. *Journal of Neurophysiology*, 1980a. **43**(6): p. 1771-1792.
68. Johnson, D.H. and W. Ray, *Optimal Stimulus Coding by Neural Populations Using Rate Codes*. *Journal of Computational Neuroscience*, 2004. **16**(2): p. 129-138.
69. Wiener, M.C. and B.J. Richmond, *Using Response Models to Estimate Channel Capacity for Neuronal Classification of Stationary Visual Stimuli Using Temporal Coding*. *J Neurophysiol*, 1999. **82**(6): p. 2861-2875.
70. Abbott, L.F. and P. Dayan, *The effect of correlated variability on the accuracy of a population code*. *Neural Computation*, 1999. **11**(1): p. 91-101.
71. Johnson, K.O., *Sensory discrimination: neural processes preceding discrimination decision*. *Journal of Neurophysiology*, 1980b. **43**(6): p. 1793-1815.
72. Averbeck, B.B., P. Latham, and A. Pouget, *Neural correlations, population coding and computation*. *Nature Reviews Neuroscience*, 2006. **7**: p. 358-366.
73. Zohary, E., M.N. Shadlen, and W.T. Newsome, *Correlated neuronal discharge rate and its implications for psychophysical performance*. *Nature*, 1994. **370**(6485): p. 140-143.
74. Puchalla, J.L., et al., *Redundancy in the Population Code of the Retina*. *Neuron*, 2005. **46**(3): p. 493-504.

75. Latham, P.E. and S. Nirenberg, *Synergy, Redundancy, and Independence in Population Codes, Revisited*. Journal of Neuroscience, 2005. **25**(21): p. 5195.
76. Stevenson, I.H. and K.P. Kording, *How advances in neural recording affect data analysis*. Nature Neuroscience. **14**(2): p. 139-142.
77. Wainwright, M.J. and M.I. Jordan, *Graphical models, exponential families, and variational inference*. Foundations and Trends® in Machine Learning, 2008. **1**(1-2): p. 1-305.
78. Schneidman, E., et al., *Weak pairwise correlations imply strongly correlated network states in a neural population*. Nature, 2006. **440**(7087): p. 1007-1012.
79. Tang, A., et al., *A maximum entropy model applied to spatial and temporal correlations from cortical networks in vitro*. Journal of Neuroscience, 2008. **28**(2): p. 505.
80. Truccolo, W., L.R. Hochberg, and J.P. Donoghue, *Collective dynamics in human and monkey sensorimotor cortex: predicting single neuron spikes*. Nature neuroscience, 2009.
81. Marre, O., et al., *Prediction of spatiotemporal patterns of neural activity from pairwise correlations*. Physical review letters, 2009. **102**(13): p. 138101.
82. Boyd, S.P. and L. Vandenberghe, *Convex optimization*. 2004: Cambridge Univ Pr.
83. Shelley, M. and D. McLaughlin, *Coarse-grained reduction and analysis of a network model of cortical response: I. Drifting grating stimuli*. Journal of Computational Neuroscience, 2002. **12**(2): p. 97-122.
84. Palm, G., A. Aertsen, and G.L. Gerstein, *On the significance of correlations among neuronal spike trains*. Biological cybernetics, 1988. **59**(1): p. 1-11.
85. Gilbert, C.D., *Microcircuitry of the Visual Cortex*. Annual Review of Neuroscience, 1983. **6**(1): p. 217-247.
86. Reich, D.S., et al., *Response Variability and Timing Precision of Neuronal Spike Trains In Vivo*, 1997, Am Physiological Soc. p. 2836-2841.
87. Eggermont, J.J., *Function aspects of synchrony and correlation in the auditory nervous system* Concepts in Neuroscience, 1993. **4**(2): p. 105-129.

88. Dzakpasu, R. and M. Zochowski, *Discriminating different types of synchrony in neural systems*. Physica D., 2005. **208**: p. 115-122.
89. Georgopoulos, A.P., et al., *On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex*. Journal of Neuroscience, 1982. **2**(11): p. 1527-1537.
90. Riehle, A., et al., *Spike Synchronization and Rate Modulation Differentially Involved in Motor Cortical Function*. Science, 1997. **278**(5345): p. 1950.
91. Kandel, E.R., J.H. Schwartz, and T.M. Jessell, *Principles of neural science*. 2000: Appleton & Lange.
92. Rieke, F., *Spikes:: Exploring the Neural Code*. 1997.
93. Montani, F., et al., *The Role of Correlations in Direction and Contrast Coding in the Primary Visual Cortex*. Journal of Neuroscience, 2007. **27**(9): p. 2338.
94. Nirenberg, S. and P.E. Latham, *Decoding neuronal spike trains: How important are correlations?* Proceedings of the National Academy of Sciences, 2003. **100**(12): p. 7348-7353.
95. Jermakowicz, W.J. and V.A. Casagrande, *Neural networks a century after Cajal*. Brain Research Reviews, 2007. **55**(2): p. 264-284.
96. Gourevitch, B. and J.J. Eggermont, *A nonparametric approach for detection of bursts in spike trains*. Journal of Neuroscience Methods, 2007. **160**(2): p. 349-358.
97. Salinas, E. and T.J. Sejnowski, *Correlated neuronal activity and the flow of neural information*. Nat Rev Neurosci, 2001. **2**(8): p. 539-550.
98. Perkel, D.H., G. Gerstein, and G.P. Moore, *Neuronal spike trains and stochastic point processes. II. Simultaneous spike trains*. J. Biophys, 1967. **7**(4): p. 419-40.
99. Gerstein, G.L., D.H. Perkel, and J.E. Dayhoff, *Cooperative firing activity in simultaneously recorded populations of neurons: detection and measurement*. Journal of Neuroscience, 1985. **5**(4): p. 881.
100. Smith, W.S. and E.E. Fetz, *Synaptic Interactions Between Forelimb-Related Motor Cortex Neurons in Behaving Primates*. Journal of Neurophysiology, 2009. **102**(2): p. 1026.

101. Schneidman, E., W. Bialek, and M.J. Berry, *Synergy, redundancy, and independence in population codes*. Journal of Neuroscience, 2003. **23**(37): p. 11539-11553.
102. Globerson, A., et al., *The minimum information principle and its application to neural code analysis*. Proceedings of the National Academy of Sciences, 2009. **106**(9): p. 3490-3495.
103. Brenner, N., et al., *Synergy in a Neural Code*. Neural Comp., 2000. **12**(7): p. 1531-1552.
104. Martignon, L.G., et al., *Neural Coding: Higher-Order Temporal Patterns in the Neurostatistics of Cell Assemblies*. Neural Comp., 2000. **12**: p. 2621-2653
105. Borst, A. and F.E. Theunissen, *Information theory and neural coding*. nature neuroscience, 1999. **2**: p. 947-958.
106. Nelken, I. and G. Chechik, *Information theory in auditory research*. Hearing research, 2007. **229**(1-2): p. 94-105.
107. Schneidman, E., et al., *Weak pairwise correlations imply strongly correlated network states in a neural population*. Nature, 2006. **440**(7087): p. 1007.
108. Globerson, A., et al., *The minimum information principle and its application to neural code analysis*. Proceedings of the National Academy of Sciences, 2009. **106**(9): p. 3490.
109. Studeny, M. and J. Vejnárova, *The multiinformation function as a tool for measuring stochastic dependence*. Learning in graphical models, 1998: p. 261–297.
110. Van Emden, M.H. and A. Universiteit van, *An analysis of complexity*. 1971: Mathematisch Centrum.
111. Schneidman, E., et al., *Network information and connected correlations*. Physical review letters, 2003. **91**(23): p. 238701-238701.
112. Silva, J. and R. Willett, *Hypergraph-Based Anomaly Detection of High-Dimensional Co-Occurrences*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2009. **31**(3): p. 563-569.
113. London, M., et al., *The information efficacy of a synapse*. nature neuroscience, 2002. **5**(4): p. 332-340.

114. Kindermann, R., J.L. Snell, and S. American Mathematical, *Markov random fields and their applications*. 1980: American Mathematical Society Providence, RI.
115. Golumbic, M.C., *Algorithmic graph theory and perfect graphs*. 1980: Academic press.
116. Shlens, J., et al., *The structure of multi-neuron firing patterns in primate retina*. Journal of Neuroscience, 2006. **26**(32): p. 8254.
117. Pillow, J.W., et al., *Spatio-temporal correlations and visual signalling in a complete neuronal population*. Nature, 2008. **454**(7207): p. 995-999.
118. Shlens, J., et al., *The Structure of Large-Scale Synchronized Firing in Primate Retina*. Journal of Neuroscience, 2009. **29**(15): p. 5022-5031.
119. Callaway, E.M., *Local circuits in primary visual cortex of the macaque monkey*. Annu Rev Neurosci, 1998. **21**: p. 47-74.
120. Yoshimura, Y. and E.M. Callaway, *Fine-scale specificity of cortical networks depends on inhibitory cell type and connectivity*. Nat Neurosci, 2005. **8**(11): p. 1552-9.
121. Sanes, J.N. and J.P. Donoghue, *Plasticity and primary motor cortex*. Annual Review of Neuroscience, 2000. **23**(1): p. 393-415.
122. Cohen, M.R. and W.T. Newsome, *Context-dependent changes in functional circuitry in visual area MT*. Neuron, 2008. **60**(1): p. 162-173.
123. Azouz, R. and C.M. Gray, *Dynamic spike threshold reveals a mechanism for synaptic coincidence detection in cortical neurons in vivo*, 2000, National Acad Sciences. p. 8110-8115.
124. Grün, S., M. Diesmann, and A. Aertsen, *Unitary events in multiple single-neuron spiking activity: I. Detection and significance*. Neural Computation, 2002. **14**(1): p. 43-80.
125. Bondy, J.A. and U.S.R. Murty, *Graph theory*. 2007: Springer Verlag.
126. Bell, A.J. *The co-information lattice*. 2003.
127. Crochet, S. and C.C. Petersen, *Cortical dynamics by layers*. Neuron, 2009. **64**(3): p. 298-300.

128. Roudi, Y., S. Nirenberg, and P.E. Latham, *Pairwise maximum entropy models for studying large biological systems: when they can work and when they can't*. Plos Computational Biology, 2009. **5**(5).
129. Truccolo, W., et al., *A Point Process Framework for Relating Neural Spiking Activity to Spiking History, Neural Ensemble, and Extrinsic Covariate Effects*. J Neurophysiol, 2005. **93**(2): p. 1074-1089.
130. Brockwell, A.E., R.E. Kass, and A.B. Schwartz, *Statistical Signal Processing and the Motor Cortex*. Proceedings of the IEEE, 2007. **95**(5): p. 881-898.
131. Sprekeler, H., C. Michaelis, and L. Wiskott, *Slowness: An Objective for Spike-Timing-Dependent Plasticity?* PLoS Comput Biol, 2007. **3**(6): p. e112.
132. Jackson, A., J. Mavoori, and E.E. Fetz, *Long-term motor cortex plasticity induced by an electronic neural implant*. Nature, 2006. **444**: p. 56-60.
133. Hendry, S.H., et al., *Numbers and proportions of GABA-immunoreactive neurons in different areas of monkey cerebral cortex*. J Neurosci, 1987. **7**: p. 1503-1519.
134. Bi, G.Q. and M.M. Poo, *Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type*. Journal of Neuroscience, 1998. **18**: p. 10464-72.
135. Jacob, V., et al., *Spike Timing-Dependent Synaptic Depression in the In Vivo Barrel Cortex of the Rat*. J. Neurosci., 2007. **27**: p. 1271-1284.
136. Froemke, R.C. and Y. Dan, *Spike-timing-dependent synaptic modification induced by natural spike trains*. Nature, 2002. **416**(6879): p. 433-438.
137. Dan, Y. and M.M. Poo, *Spike Timing-Dependent Plasticity: From Synapse to Perception*. Physiological Reviews, 2006. **86**(3): p. 1033.
138. Dan, Y. and M.M. Poo, *Spike timing-dependent plasticity of neural circuits*. Neuron, 2004. **44**: p. 23-30.
139. Perkel, D.H., G.L. Gerstein, and G.P. Moore, *Neuronal spike trains and stochastic point processes II. Simultaneous spike trains*. Biophysical Journal, 1967. **7**(4): p. 419-440.
140. Histed, M.H., A. Pasupathy, and E.K. Miller, *Learning Substrates in the Primate Prefrontal Cortex and Striatum: Sustained Activity Related to Successful Actions*. Neuron, 2009. **63**(2): p. 244-253.

141. Aghagolzadeh, M. and K. Oweiss, *Compressed and Distributed Sensing of Neuronal Activity for Real Time Spike Train Decoding*. IEEE Transactions on Neural Systems and Rehabilitation Engineering, April 2009. **17**(2): p. 116-127.
142. Brown, E.N., R. Kass, and P. Mitra, *Multiple neural spike train data analysis: State-of-the-Art and future challenges*. Nature Neurosc., 2004. **7** (5): p. 456-461.
143. Harrison, R.R., et al., *A Low-Power Integrated Circuit for a Wireless 100-Electrode Neural Recording System*. IEEE JOURNAL OF SOLID-STATE CIRCUITS, 2007. **42**(1): p. 123.
144. Wise, K.D., et al., *Microelectrodes, Microelectronics, and Implantable Neural Microsystems*. Proceedings of the IEEE, 2008. **96**(7): p. 1184-1202.
145. Sodagar, A.M., K.D. Wise, and K. Najafi, *A wireless implantable microsystem for multichannel neural recording*. Microwave Theory and Techniques, IEEE Transactions on, 2009. **57**(10): p. 2565-2573.
146. Ventura, V., *Spike Train Decoding Without Spike Sorting*. Neural Comp., 2007. **20**(4): p. 923-963.
147. Harris, K.D., et al., *Accuracy of tetrode spike separation as determined by simultaneous intracellular and extracellular measurements*. Journal of Neurophysiology, 2000. **84**(1): p. 401-414.
148. Jog, M., et al., *Tetrode technology: advances in implantable hardware, neuroimaging, and data analysis techniques*. Journal of neuroscience methods, 2002. **117**(2): p. 141-152.
149. Mechler, F., et al., *Three-dimensional localization of neurons in cortical tetrode recordings*. Journal of neurophysiology, 2011. **106**(2): p. 828-848.
150. Reich, D.S., et al., *Interspike intervals, receptive fields, and information encoding in primary visual cortex*. The Journal of Neuroscience, 2000. **20**(5): p. 1964-1974.
151. Rad, K.R. and L. Paninski, *Efficient, adaptive estimation of two-dimensional firing rate surfaces via Gaussian process methods*. Network: Computation in Neural Systems, 2010. **21**(3-4): p. 142-168.
152. Zhang, K. and T.J. Sejnowski, *Neuronal tuning: To sharpen or broaden?* Neural Computation, 1999. **11**(1): p. 75-84.
153. M. Aghagolzadeh, A.M.a.K.O., *Sorting Neural Activity via Simple Thresholding*. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 2012: p. to appear.

154. Boyd, S. and L. Vandenberghe, *Convex optimization*. 2004: Cambridge university press.
155. van den Berg, E. and M.P. Friedlander, *Sparse optimization with least-squares constraints*. SIAM Journal on Optimization, 2011. **21**(4): p. 1201-1229.