UNIVERSITY OF CALIFORNIA,
IRVINE

Novel Bayesian Methods in Neuroscience

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Statistics

by

Bo Zhou

Dissertation Committee:
Associate Professor Babak Shahbaba, Chair
Professor Hernando Ombao
Associate Professor Zhaoxia Yu

2015

DEDICATION

To Xin Gao, my wife who supports me each step of the way.
To Hongying Li and Guangshun Zhou, my parents and my foundation.
Words can not express how much I love you all.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I am deeply grateful to my supervisor, Babak Shahbaba, whose encouragement, support and supervision enabled me to make this thesis possible. I would like to thank the members of my committee, Hernando Ombao and Zhaoxia Yu, for their extremely patient help in the face of variety of obstacles. I would like to thank Professors, Sam Behseta and David Moorman, for their constant support during the completion of the project. I would like to thank my fellow doctoral students, Shiwei Lan, Zhe Yu, Yuxiao Wang, Yan He and Jie Zhen, for their support, feedbacks and friendship. I greatly appreciate the Dean's Fellowship received towards my PhD from the Donald Bren School of Informatics and Computer Science. The text of this dissertation is a reprint of the material as it appears in Neural Computation and International Conference on Machine Learning. The co-author, Babak Shahbaba, listed in this publication directed and supervised research which forms the basis for the dissertation. Lastly, I offer my regards and blessings to all of those who have supported me in any respect during my academic research.

CURRICULUM VITAE

Bo Zhou

EDUCATION

Doctor of Philosophy in Statistics **2015**
University of California, Irvine *Irvine, California*

Master of Science in Statistics **2011**
University of California, Irvine *Irvine, California*

Bachelor of Science in Applied Mathematics **2009**
Huazhong University of Science and Technology *Wuhan, China*

RESEARCH EXPERIENCE

Graduate Research Assistant **2011–2014**
University of California, Irvine *Irvine, California*

TEACHING EXPERIENCE

Teaching Assistant **2012–2014**
University of California, Irvine *Irvine, California*

SELECTED HONORS AND AWARDS

The Dean's Fellowship **2009–2014**
University of California, Irvine

Graduate Dean's Dissertation Fellowship **2013-2014**
University of California, Irvine

Excellent Academic Performance Fellowship **2007-2008**
Huazhong University of Science and Technology

Excellent Individual in Scientific Research **2006-2007**
Huazhong University of Science and Technology

Outstanding Student Fellowship **2006**
Huazhong University of Science and Technology

Sports Fellowship **2006**
Huazhong University of Science and Technology

REFEREED JOURNAL PUBLICATIONS

A semiparametric Bayesian model for detecting synchrony among multiple neurons **2014**
Neural Computation

A hierarchical modeling approach to data analysis and study design in a multi-site experimental fMRI study **2013**
Psychometrika

REFEREED CONFERENCE PUBLICATIONS

Spherical HMC for constrained target distributions **2014**
International Conference on Machine Learning

INVITED TALKS

A Bayesian model for detecting cross-neuronal interactions **2014**
AI/ML Seminar

A hierarchical Bayesian model for multi-site fMRI study **2012**
Chao Family Comprehensive Cancer Center

PROFESSIONAL MEMBERSHIPS

American Statistical Association (ASA)

ABSTRACT OF THE DISSERTATION

Novel Bayesian Methods in Neuroscience

By

Bo Zhou

Doctor of Philosophy in Statistics

University of California, Irvine, 2015

Associate Professor Babak Shahbaba, Chair

For an individual to successfully complete the task of decision-making, a set of temporally-organized events must occur: stimuli must be detected, potential outcomes must be evaluated, behaviors must be executed or inhibited, and outcomes (such as reward or punishment) must be experienced. Due to the complexity of this process, it is very likely the case that decision-making is encoded by the temporally-precise interactions among a population of neurons. Most existing statistical models, however, are inadequate for analyzing such sophisticated phenomenon as they either analyze a small number of neurons (e.g., pairwise analysis) or only provide an aggregated measure of interactions by assuming a constant dependence structure among neurons over time.

We start by proposing a scalable hierarchical semi-parametric Bayesian model to capture dependencies among multiple neurons by detecting their co-firing (possibly with some lag time). To this end, we model the spike train (sequence of 1's (spike) and 0's (silence)) for each neuron using the logistic function of a continuous latent variable with a Gaussian Process prior. Then we model the joint probability distribution of multiple neurons as a function of their corresponding marginal distribution using a parametric copula model. Our approach provides a flexible framework for modeling the underlying firing rates of each neuron. It also also allows us to make inference regarding both contemporaneous and lagged synchrony.

We evaluate our approach using several simulation studies and apply it to analyze real data collected from an experiment designed for investigating the role of the prefrontal cortex of rats in reward-seeking behaviors.

Next, we propose a non-stationary Bayesian model to capture the dynamic nature of neuronal activity (such as the time-varying strength of the interactions among neurons). Our proposed method yields results that provide new insights into the dynamic nature of population coding in the prefrontal cortex during decision making. In our analysis, we note that while some neurons in the prefrontal cortex do not synchronize their firing activity until the presence of a reward, a different set of neurons synchronize their activity shortly after the onset of stimulus. These differentially synchronizing sub-populations of neurons suggests a continuum of population representation of the reward-seeking task. Our analyses also suggest that the degree of synchronization differs between the rewarded and non-rewarded conditions.

Finally we propose a novel statistical model for detecting neuronal communities involved in decision-making process. Our method characterizes the non-stationary activity of multiple neurons during a basic cognitive task by modeling their joint probability distribution dynamically. Our proposed model can capture the time-varying dependence structure among neurons while allowing the neuronal activity to change over time. This way, we are able to identify time-varying neuronal communities. By identifying communities of neurons that vary under different decisions, we expect our method to provide insights into the decision-making process in particular as well as into a broad range of cognitive functions.

Chapter 1

Introduction

1.1 Background

Cognitive behaviors such as decision-making are complex as they involve organized patterns of cognitive and behavioral subcomponents. For an individual to successfully complete a decision-making task, a number of temporally-organized events must occur: stimuli must be detected, potential outcomes must be evaluated, behaviors must be executed and inhibited, and outcomes (such as reward or punishment) must be experienced. Neural mechanisms involved in the decision-making process reveal principles of cognitive functions in general [73]. Indeed, deficits in decision making underlie multiple psychiatric disorders including, but not limited to, ADHD, bipolar, and compulsive behavior disorders,mood and anxiety disorders, and schizophrenia [61]. Hence, the potential clinical impact of this work is broad. Especially, understanding how populations of neurons function in decision-making process and related behaviors will guide targeting of specific neural systems for development of new treatments for psychiatric disorders.

In recent years, many researchers have studied the relationship between neuronal activities

and decision making [22, 73, 9]. These neurophysiological studies commonly involve recording a sequence of spikes (action potentials) produced by neurons over time, known as a *spike train*, for each neuron (Figure 1.1). However, complex behaviors and cognitive functions (i.e., decision making process) are driven by the interactions between populations of neurons [9] instead of a single neuron. For many years preceding the advent of ensemble recording, neurons were recorded successively and then combined into synthetic populations based upon shared timing. Although this technique continues to produce valuable information [38], investigators are gravitating more and more towards simultaneous recording of multiple neurons [40]. One of the major reasons that multiple-electrode recording technique has been embraced is that it allows us to identify the dynamic activities of populations of neurons simultaneously.

There have been a number of studies investigating how the activity of a single neuron are related to decision-making [22, 73]. Later on, analysis of simultaneously recorded neurons became focused on correlation of activity across pairs of neurons using cross correlation analyses [48] and analyses of changes in correlation over time, i.e., by using a joint peri-stimulus time histogram (JPSTH) [21] or rate correlations [48]. Similar analyses can be also performed in the frequency domain by using coherence analysis of Fourier-transformed neural activity of neuron pairs [6]. These methods attempt to distinguish *exact synchrony* or *lagged synchrony* between a pair of neurons. Subsequently, a class of associated methods were developed for addressing the question of whether synchrony between a pair of neurons is merely due to chance or not. Later, researchers introduced a variety of methods to test the statistical significance of synchrony, such as bootstrap confidence intervals [25]. To detect the presence of conspicuous spike coincidences in multiple neurons, [23] proposed a novel method, where such conspicuous coincidences, called *unitary events*, are defined as joint spike constellations that recur more often than what can be explained by chance alone. In their approach, simultaneous spiking events from N neurons are modeled as a joint process composed of N parallel point processes. To test the significance of unitary events, they

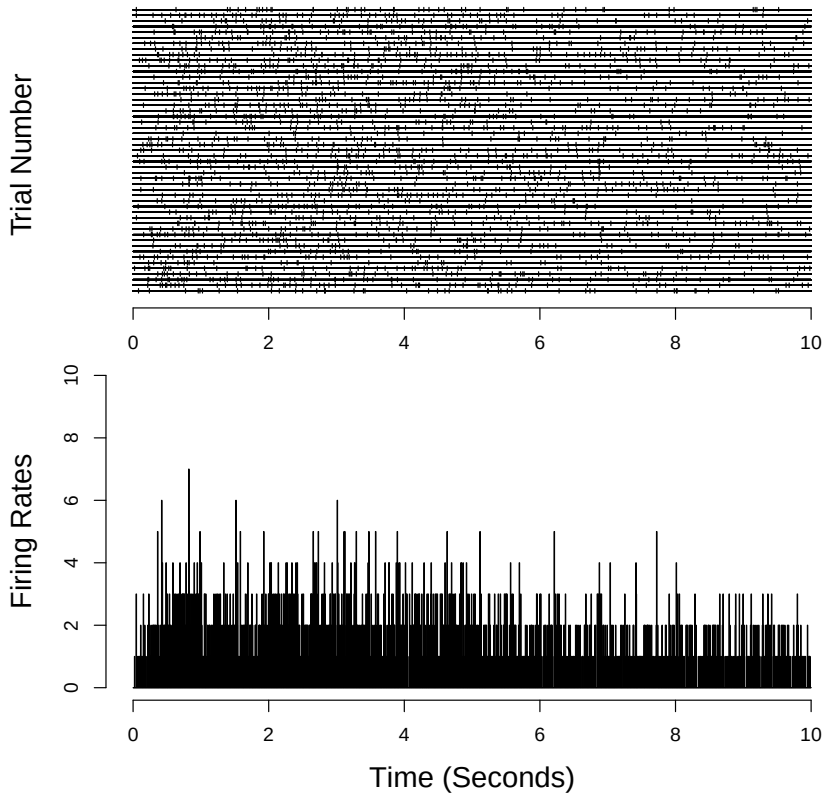


Figure 1.1: The Peri-stimulus time histogram (PSTH) of a single neuron. The upper plot is the continuous representation (point process) of the spike train data. Each line represents a trial and points on the line represent the time when there is a spike. The lower plot is a summary of the spike train data by averaging activity across trials.

developed a new method, called joint-surprise, which measures the cumulative probability of finding the same or even larger number of observed coincidences by chance. [63] investigated how correlated spiking activity in complete neural populations depends on the pattern of visual simulation. They proposed to use a generalized linear model to capture the encoding of stimuli in the spike trains of a neural population. In their approach, a cell's input is presented by a set of linear filters and the summed filter responses are exponentiated to obtain an instantaneous spike rate. The set of filters include a stimulus filter, a post-spike filter (to capture dependencies on history), and a set of coupling filter (to capture dependencies on the recent spiking of other cells). Recent developments in detecting synchrony among

neurons include models that account for trial to trial variability and the evolving intensity of firing rates between multiple trials. For more discussion on analysis of spike trains, refer to [23, 6, 32, 13, 63, 4, 14, 35]

Recently [33] proposed a new method to quantify synchrony between a pair of neurons. They argued that separating stimulus effects from history effects would allow for a more precise estimate of the instantaneous conditional firing rate. Specifically, given the firing history H_t, define $\lambda^A(t|H_t^A)$, $\lambda^B(t|H_t^B)$, and $\lambda^{AB}(t|H_t^{AB})$ to be the conditional firing intensities of neuron A, neuron B, and their synchronous spikes respectively. Independence between the two point processes can be examined by testing the null hypothesis $H_0 : \zeta(t) = 1$, where $\zeta(t) = \frac{\lambda^{AB}(t|H_t^{AB})}{\lambda^A(t|H_t^A)\lambda^B(t|H_t^B)}$. The quantity $[\zeta(t)-1]$ can be interpreted as the deviation of co-firing from what can be predicted by independence (i.e., when $\zeta(t) = 1$ for all t). Note that the marginal probability of firing for each neuron still need to be modeled, respectively. To do this, they assume that the spike train follows a Poisson process, which is the simplest form of point processes. The main limitation of this work is that it assumes that the number of spikes within a particular time frame follows a Poisson distribution. It is, however, very unlikely that actual spike trains follow this assumption [1, 31, 65, 32, 28]. One possible remedy is to use inhomogeneous Poisson process or inhomogeneous gamma-interval process [13] which assumes time-varying firing rates. See [5, 6, 32, 79, 66, 13, 4, 34, 70, 35] for more alternative methods of modeling spike trains. Another limitation is that it assumes static relationship between a pair of neurons which might be unlikely the case during complex decision making process.

A set of multivariate analysis techniques, such as principal components analysis (PCA), independent component analysis (ICA), and k-means clustering, have been also proposed for investigating simultaneously-recorded populations of neurons [20, 10, 56, 6, 48]. There have been also many model-based methods that use Bayesian approach or information theory for inference [See for example, 30, 72]. In addition, Several stationary copula models have

been previously proposed for neuroscience problems when analyzing multiple neurons. [4] proposed a variety of copula models for capturing neural dependencies and develop an efficient maximum likelihood procedure for inference. In a recent work, [77] proposed a unified approach to model multivariate binary data using copulas on partitions. They established a modeling framework which produced likelihood-based inference about the effects of the covariates on marginal success probabilities while accounting for the correlation among multivariate outputs. Our method can be considered as a generalization of this approach from multivariate binary outputs to multivariate binary time series.

These static models discussed above which aggregate cross-neuronal spike-train interactions over time can produce misleading results. Indeed, there have been many dynamic models already developed for modeling brain function and effective connectivity [18, 11, 58, 57, 43, 60, 37]. [18] developed a non-stationary model to detect psychophysiological interactions among neurophysiological measurements while assuming known change points and then apply it to analyze data from an fMRI experiment in which a single female subject was asked to view radially moving dots under two attentional states. Later on [67] developed a new technique to detect change points in voxel-level fMRI studies by assuming a single state-related shift from baseline to activated state and a subsequent return to baseline at a later unknown time. Their approach can be further generalized to more rapid alteration among multiple states by allowing for the possibility of multiple activation onsets and durations though out the course of the time series. The main limitation of these existing dynamic models is that they either assume the change points are known or there are a finite number of unknown change points and hence the model remains approximately static.

In general, the methods discussed so far are not designed to identify sub-networks among neurons. In this dissertation, we address this issue by proposing a novel clustering method for neurons. Our method, presented in Chapter 4, is related to partition probability method (PPM) of [44]. PPM was used for clustering with covariates, i.e., partitioning a set of

experimental units where the probability of any particular partition is allowed to depend on covariates. It is assumed that units with equal or similar covariates should be a priori more likely to co-cluster than others. let $\rho_n = \{S_1, ..., S_{kn}\}$ denote a partition of the n experimental units into kn subsets, Sj. The Partition probability models (see [26, 2, 12]) constructs the probability of a partition, $p(\rho_n)$, by introducing cohesion functions, c(A), for $A \subset \{1, ..., n\}$ that measure how tightly grouped the elements in A are thought to be. The probability model for any partition can then be written as, $p(\rho_n) \propto \sum_{j=1}^{k_n} c(S_j)$. The probability distribution of data conditional on the partition can be modeled as $p(y^n|\rho_n) \propto \sum_{j=1}^{k_n} p_j(y_j^*)$ where y_j^* is the data in the j^{th} partition. To account for the covariates in partitioning, [44] proposed to model the partition as $p(\rho_n) \propto \sum_{j=1}^{k_n} g(x_j^*)c(S_j)$ where $g(x_j^*)$ is a nonnegative function which estimates the similarity of the covariates, x_j^*. Our model for community detection among spike trains can be considered as a generalization of this model in which neurons with high co-firing probability tend to co-cluster than others.

Another approach for detecting neural communities is based on stochastic block models [27]. In the simplest stochastic blockmodels, n vertices are assigned to K blocks, or communities, and undirected edges are placed independently between vertex pairs with probabilities. The number of edges between a pair of vertices is assumed to be independently Poisson distributed. [29] proposed a generalization of blockmodels which shows substantial improvement for community detection in complex networks while accounting for an additional component, variation in vertex degree. In their degree-corrected model, the probability distribution over undirected multigraphs also relies on a new set of parameters, θ_i, which control the expected degrees of vertices i. The main limitation of this work is that the number of communities in networks must be given which might be problematic in real data analysis. Latent threshold models have also been utilized to identify the effective connectivity among multiple time

series (see [46]). A basic latent threshold model can be defined as,

$$y_t = x_t^T b + \epsilon_t, \epsilon_t \sim N(0, \sigma^2) \tag{1.1}$$

$$b_i = \beta_i \mathbb{1}_{\beta_i > d_i}. \tag{1.2}$$

d_i is predefined latent thresholds and the coefficient will be shrunk to zero if it falls below the threshold. This mechanism allows us to obtain a sparse network in terms of effective connectivity among multivariate time series. Those methods discussed above allow us to detect communities in variety of networks. However, the static nature of these methods leads to its inability to detect dynamics of community structures. Some recent works have been focused on generalizing these methods to dynamic versions. [68] recently proposed a hierarchical stochastic block model which extend the stochastic blockmodels to settings where interactions between a group of units are observed at multiple points in time. This model allows the network structures change over time by introducing a hidden Markov model for the parameters of the network. The points in time when structural changes seem to have occurred can also be detected. The community structures of the network are assumed to evolve according to a discrete-time point process and the number of states in the system can be considered as a random variable to be estimated from data. However, this model assume conditional independence between states which might not be supported by the data. More general mechanisms need to be introduced into network models that do not depend on discrete state space models. [47] also developed a dynamic latent threshold models for capturing dynamic network structure among multivariate time series by allowing the coefficient to vary over time. The value of coefficients can be zero for some periods of time and non-zero with time-varying values in other periods. Allowing for temporal changes in relationship among time series, this method provide a framework for detecting sparse dynamic structures in network connectivity by the latent threshold mechanism.

1.2 Contributions

The main contributions of the dissertation are as follows:

- **A semi-parametric Bayesian model** for detecting synchrony among multiple neurons which provides a flexible framework for modeling the time-varying underlying firing rates of single neuron as well as making inference regarding both contemporaneous and lagged synchrony among neurons.
- **A dynamic Bayesian model** for characterizing time-dependent cross-neuronal interactions among multiple neurons simultaneously recorded during decision making process.
- **A non-stationary Bayesian model** for detecting time-varying community structures of the neuronal network.

1.3 Outline

This dissertation is organized as follows. Chapter 2 discusses a semi-parametric Bayesian model for detecting synchrony among multiple neurons. Chapter 3 introduces dynamic Bayesian model for characterizing time-dependent cross-neuronal interactions among multiple neurons. Chapter 4 proposes another dynamic Bayesian model to capture time-varying community structures among population of neurons. The last chapter 5 is devoted to discussion and future directions. .

Chapter 2

A Semiparametric Bayesian Model for Detecting Synchrony Among Multiple Neurons

2.1 Description of the Experiment and Data

In this experiment we recorded the activity of multiple neurons in the prefrontal cortex of rats while they were presented with stimuli that either predicted or did not predict the availability of a reward. Figure 2.1 shows the experiment settings [see 41, for more details]. Rats were tested in an operant box ($\sim 12''$ L $\times$ $9.5''$ W $\times$ $11.5''$ H; Med-Associates) which was equipped with lights, tones, levers, and a receptacle for delivering liquid sucrose reward. During the recording/test sessions, two different stimuli were presented: tone 1 (10 KHz) or tone 2 (5 KHz) individually and in pseudo-random order. At the same time, one of two levers was presented an active-lever, paired with tone 1 (Rewarded-Stimulus - RS) and an inactive-lever paired with tone 2 (Non-rewarded Stimulus - NS). Pressing the active lever resulted in the

offset of tone 1, retraction of the lever, and illumination of the reward receptacle. If the rat then went to the reward receptacle, 0.1 ml of 15% sucrose solution was delivered as a reward. Pressing the inactive lever produced no effect. Tones were played for a maximum of 3 sec, and levers remained extended for a maximum of 10 sec. Animals received a 20-30 sec time-out period following either well entry or lever retraction (after the NS-lever was retracted or if the RS-lever was pressed but the well was not entered). Testing/recording sessions lasted one hour, during which time ~ 50 RS and ~ 50 NS lever/tone complexes were presented. After rats were trained to perform the DS-Sucrose task, we recorded the activity of prefrontal cortical neurons during task performance. After this recording session, we also recorded the activity of the same populations of neurons during four extinction sessions. During these sessions pressing the active lever resulted in illumination of the reward receptacle, but entering the receptacle produced no reward. Over the course of these extinction sessions rats learned not to press the active lever. The results in this report are focused exclusively on data collected during performance of the DS-Sucrose task, and do not include (extinction) data.

We recorded the activity of multiple single neurons simultaneously by sampling from multiple channels of an implanted 16-wire electrode array. We recorded both action potentials (i.e., spikes– the firing activity of single neurons) and local field potentials (averaged population activity). In this report we focus on action potential data. Arrays were connected to a headstage during recording sessions (20x gain, Plexon) to amplify the recorded neural signals. Signals were passed through a cable (Omnetics Connector) to an electrical commutator (Keyo Electric) to allow free movement. Commutator output was delivered to a Plexon recording system (MAP/16), where signals were amplified (50x), filtered (100 Hz - 8 kHz), and sampled (40 kHz). Action potentials were recorded using RASPUTIN software (Plexon) where gain and thresholds were set to maximize signal-to-noise. Recorded spikes were further sorted offline using Offline Sorter (Plexon) using a combination of template-matching and principal components analyses. By spike-sorting, we were able to isolate action potential waveforms

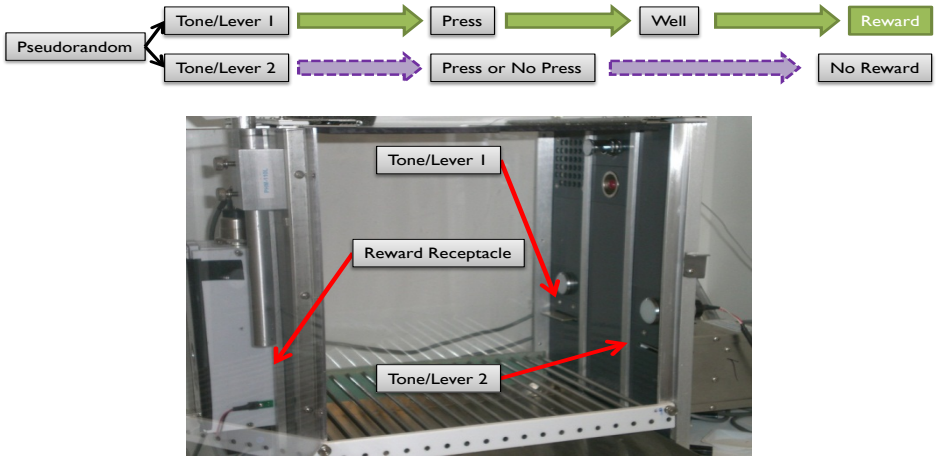


Figure 2.1: A sound-attenuated operant box equipped with 2 retractable levers, a house-light (an overhead light to illuminate the whole box) a tone generator for making tones of different frequencies, and a port on the side of the box opposite the levers where the animal collects its reward (0.1 ml of 15% sucrose solution delivered by a pump outside the box)

corresponding to the activity of each neuron recorded so that we could identify the unique contribution of each neurons activity. Well-isolated (high signal to noise) action potentials from single neurons that fired consistently throughout the recording session were included for analysis. The time stamp of each action potential for each neuron was recorded and time stamps were used for further analysis. Time stamps for behavioral events were also recorded. These were sent from the Med-Associates behavioral control system to the Plexon recording system for use in aligning neural activity to behavior.

2.2 Gaussian Process Modeling of Firing Rates

Capturing synchrony between a pair of neurons partly depends on a better understanding of the firing patterns of each neuron. As seen in Figure 1.1, the nonlinear firing rates can

provide meaningful information regarding the temporal activity of each neuron. To this end, we propose to use a Gaussian process model to model the non-stationary underlying firing rate for a single neuron. First we discretize time so that there is at most one spike within each time interval. The spike trains represented by point processes can then be transformed to binary time series which consist of 1s and 0s, where 1 indicates presence and 0 indicates absence of spikes in each time bin. Define the response variable, y_t, to be the binary time series comprised of 1s (spike) and 0s (silence). The firing rate for each neuron is assumed to depend on an underlying latent variable, $u(t)$, which has a Gaussian process prior. In statistics and machine learning, Gaussian processes are widely used as priors over functions. Similar to the Gaussian distribution, a Gaussian process is defined by its mean (usually set to 0 in prior) and its covariance function C: $f \sim \mathcal{GP}(0, C)$. Here, the function of interest is the underlying latent variable $u(t)$, which is a stochastic process indexed by time t. Hence, the covariance function is defined in terms of t. We use the following covariance form, which includes a wide range of smooth nonlinear functions [64, 51]:

$$\begin{aligned} C_{ij} &= Cov[u(t_i), u(t_j)] && (2.1) \\ &= \lambda^2 + \eta^2 \exp[-\rho^2(t_i - t_j)^2] + \delta_{ij}\sigma_\epsilon^2 && (2.2) \end{aligned}$$

In this setting, ρ^2 and η^2 control smoothness and the height of oscillations respectively. λ, η, ρ and σ are hyperparameters with their own hyperpriors. Throughout this paper, we put $N(0, 3^2)$ prior on the log of these hyperparameters. We specify the spike probability, p_t, within time interval t in terms of $u(t)$ through the following transformation:

$$p_t = \frac{1}{1 + \exp[-u(t)]}, \quad (2.3)$$

As $u(t)$ increases, so does p_t.

The prior autocorrelation imposed by this model allows the firing rate to change smoothly over time. Note that this does not mean that we believe the firing patterns over a single trial are smooth. However, over many trials, our method finds a smooth estimate of the firing rate. The dependence on prior firing patterns is through the term $(t_i - t_j)$ in the covariance function. As this term decreases, the correlation between $u(t_i)$ and $u(t_j)$ increases. This is different from other methods [31, 33] that are based on including an explicit term in the model to capture firing history. For our analysis of experimental data, we discretize the time into 5 ms intervals so there is at most one spike within each interval. Therefore, the temporal correlations in our method are on a slow time scale [25]. When there are R trials (i.e., R spike trains) for each neuron, we model the corresponding spike trains as conditionally independent given the latent variable $u(t)$. Note that we can allow for trial-to-trial variation by including a trial-specific mean parameter such that $[u(t)]^{(r)} \sim \mathcal{GP}(\mu_r, C)$, where $r = 1, \ldots, R$, ($R =$ total number of trials or spike trains).

Figure 2.2 illustrates this method using 40 simulated spike trains for a single neuron. The dashed line shows the true firing rate , $p_t = 5(4 + 3\sin(3\pi t))$, for $t = 0, 0.01, \ldots, 1$, the solid line shows the posterior expectation of the firing rate, and the gray area shows the corresponding 95% probability interval. The plus signs on the horizontal axis represents spikes over 100 time intervals for one of the 40 trials.

Figure 2.3 shows the posterior expectation of firing rate (blue curve) overlaid on the PSTH plot of a single neuron with 5 ms bin intervals from the experimental data (discussed above) recorded over 10 seconds.

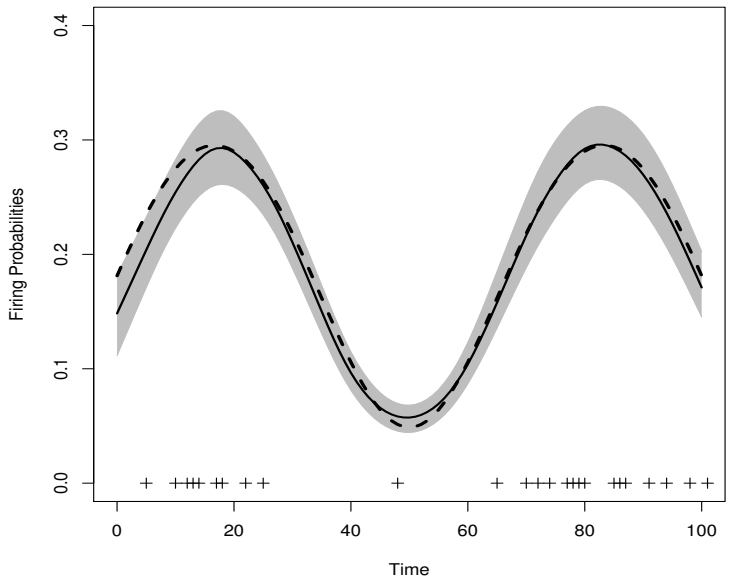


Figure 2.2: An illustrative example for using a Gaussian process model for a neuron with 40 trials. The dashed line shows the true firing rate, the solid line shows the posterior expectation of the firing rate, and the gray area shows the corresponding 95% probability interval. The plus signs on the horizontal axis represents spikes over 100 time intervals for one of the 40 trials.

2.3 Modeling dependencies between a pair of neurons

Based on modeling the firing rates of each single neuron using Gaussian process models discussed above, we can then identify the synchrony between a pair of neurons by jointly modeling of their firing activity. Let y_t and z_t be binary time series indicating presence or absence of spikes within time interval t for two neurons. Denote p_t to be the spike probability at interval t for the first neuron, and q_t to denote the spike probability at the same interval for the second neuron. Given the corresponding latent variables $u(t)$ and $v(t)$ with Gaussian process priors $\mathcal{GP}(0, C_u)$ and $\mathcal{GP}(0, C_v)$ respectively, we model the marginal firing probabilities of the two neurons as $p_t = 1/\{1+\exp[-u(t)]\}$ and $q_t = 1/\{1+\exp[-v(t)]\}$. If the the two neurons are independent, the joint probability of firing at the same time

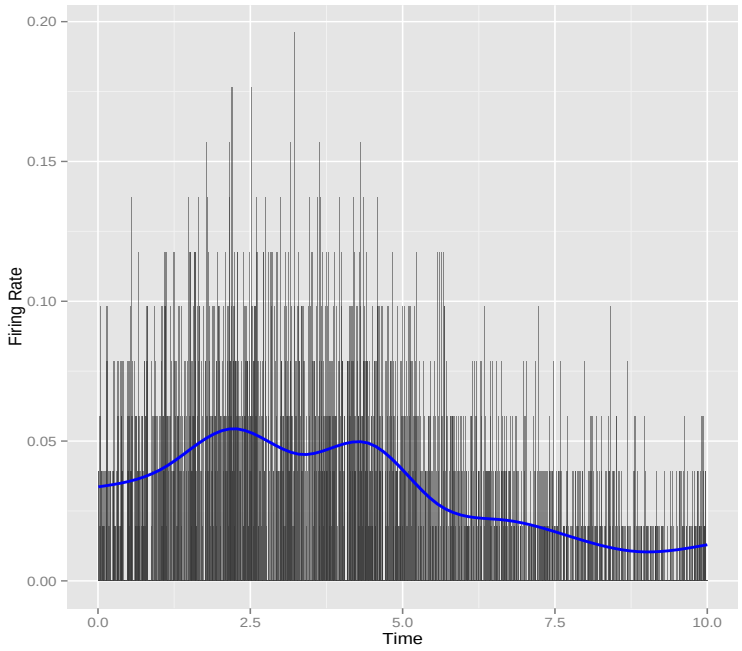


Figure 2.3: Using our Gaussian process model to capture the underlying firing rate of a single neuron from prefrontal cortical areas in rat's brain. There are 51 spike trains recorded over 10 seconds. The PSTH plot is generated by creating 5 ms intervals. The blue curve shows the estimated firing rate (posterior expectation).

is $P(y_t = 1, z_t = 1) = p_t q_t$. In general, however, we can write the probability of firing simultaneously as the product of their individual probabilities multiplied by a factor, $p_t q_t \zeta$, where ζ represents the excess firing rate ($\zeta > 1$) or the suppression firing rate ($\zeta < 1$) due to dependence between two neurons [78, 33]. That is, ζ accounts for the excess joint spiking beyond what is explained by independence. For independent neurons, $\zeta = 1$. Sometimes, the extra firing can occur after some lag time L. That is, in general, $P(y_t = 1, z_{t+L} = 1) = p_t q_{t+L} \zeta$ for some L. Therefore, the marginal and joint probabilities of two neurons can be

written as follows,

$$
\begin{aligned}
P(y_t = 1|p, q, \zeta, L) &= p_t && (2.4)\\
P(y_t = 0|p, q, \zeta, L) &= 1 - p_t && (2.5)\\
P(z_t = 1|p, q, \zeta, L) &= q_t && (2.6)\\
P(z_t = 0|p, q, \zeta, L) &= 1 - q_t && (2.7)\\
P(y_t = 1, z_{t+L} = 1|p, q, \zeta, L) &= p_t q_{t+L} \zeta && (2.8)\\
P(y_t = 1, z_{t+L} = 0|p, q, \zeta, L) &= p_t - p_t q_{t+L} \zeta && (2.9)\\
P(y_t = 0, z_{t+L} = 1|p, q, \zeta, L) &= q_{t+L} - p_t q_{t+L} \zeta && (2.10)\\
P(y_t = 0, z_{t+L} = 0|p, q, \zeta, L) &= 1 - p_t - q_{t+L} + p_t q_{t+L} \zeta && (2.11)
\end{aligned}
$$

where

$$
\frac{\max(p_t + q_{t+L} - 1, 0)}{p_t q_{t+L}} \leq \zeta \leq \frac{\min(p_t, q_{t+L})}{p_t q_{t+L}}. \tag{2.12}
$$

In this setting, the observed data include two neurons with R trials of spike trains (indexed by $r = 1, 2, ..., R$) per neuron. Each trial runs for S seconds. We discretize time into T intervals (indexed by $t = 1, 2, ..., T$) of length S/T such that there are at most 1 spike in within each interval. We assume that the lag L can take a finite set of values from $[-K, K]$

for some biologically meaningful K, and write the likelihood function as follows:

$$\mathcal{L}(\zeta, u, v, L) = \sum_{k=0}^{K} 1_{k=L} \prod_{r=1}^{R} \Big[\prod_{t=1}^{T-k} P(y_t^{(r)}, z_{t+k}^{(r)}) \prod_{t=T-k+1}^{T} P(y_t^{(r)}) \prod_{t=1}^{k} P(z_t^{(r)}) \Big] + \sum_{k=-K}^{-1} 1_{k=L} \prod_{r=1}^{R} \Big[\prod_{t=1}^{T+k} P(y_{t-k}^{(r)}, z_t^{(r)}) \prod_{t=1}^{-k} P(y_t^{(r)}) \prod_{t=T+k+1}^{T} P(z_t^{(r)}) \Big] \tag{2.13}$$

We put uniform priors on ζ and L over the assumed range. As mentioned above, the hyperparameters in the covariance function have weakly informative (i.e., broad) priors: we assume the log of these parameters has a $N(0, 3^2)$ prior. We use Markov Chain Monte Carlo algorithms to simulate samples from the posterior distribution of model parameters given the observed spike trains. See section 2.5 for more details.

2.3.1 Illustrative examples

In this section, we use simulated data to illustrate our method. We consider three scenarios: 1) two independent neurons, 2) two dependent neurons with exact synchrony ($L = 0$), and 3) Two dependent neurons with lagged co-firing. In each scenario, we assume a time-varying firing rate for each neuron and simulate 40 trials of spike trains given the underlying firing rate. For independent neurons, we set $\zeta = 1$, whereas $\zeta > 1$ for dependent neurons.

Two independent neurons. In the first scenario, we consider two independent neurons ($\zeta = 1$). We simulate the spike trains according to our model. The firing probability at time t is set to $0.25 - 0.1\cos(2\pi t)$ for the first neuron and to $0.15 + 0.2t$ for the second neuron. For each neuron, we generated 40 trials of spike trains and divided each trial into 100 time intervals. The left panel of Figure 2.4 shows the corresponding Joint Peri-stimulus

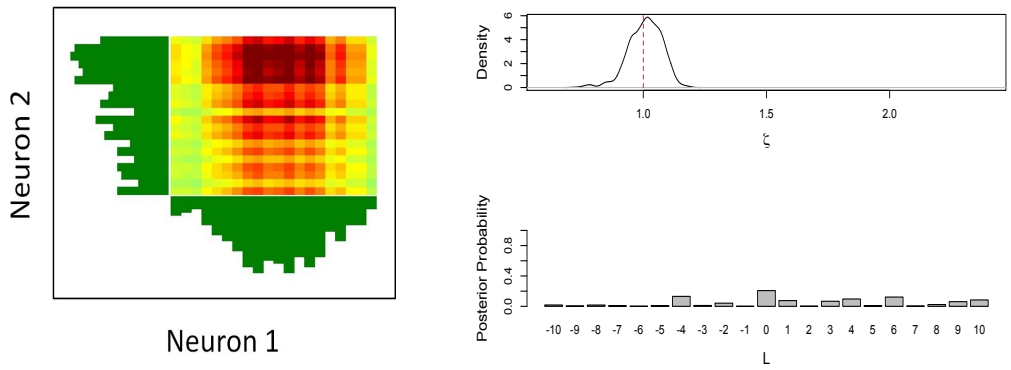


Figure 2.4: **Two independent neurons**– The left panel shows the corresponding Joint Peri-Stimulus Time Histogram (JPSTH). The right panel shows the posterior distributions of ζ and L. Darker cells represent higher frequencies.

Time Histogram (JPSTH). Each cell represents the joint frequency of spikes (darker cells represent higher frequencies) for the two neurons at given times. The marginal distributions of spikes, i.e., Peri-stimulus Time Histogram (PSTH), for the first neuron is shown along the horizontal axis. The second neuron's PSTH is shown along the vertical axis. The right panel of Figure 2.4 shows the posterior distributions of ζ and L. For this example, the posterior distribution of ζ is concentrated around 1 with median and 95% posterior probability interval equal to 1.01 and [0.85,1.12] respectively. This would strongly suggest that the two neurons are independent as expected. Further, the posterior probabilities of all lag values from -10 to 10 are quite small.

Two exact synchronous neurons. For our next example, we simulate data for two dependent neurons with synchrony (i.e., $L = 0$) and we set $\zeta = 1.6$. That is, the probability of co-firing at the same time is 60% higher than that of independent neurons. As before, for each neuron we generate 40 trials of spike trains each discretized into 100 time bins. In this case, the firing probabilities at time t for the two neurons are $0.25 - 0.1\cos(2\pi t)$. Figure 2.5 shows their corresponding JPSTH along with the posterior distributions of ζ and L. The

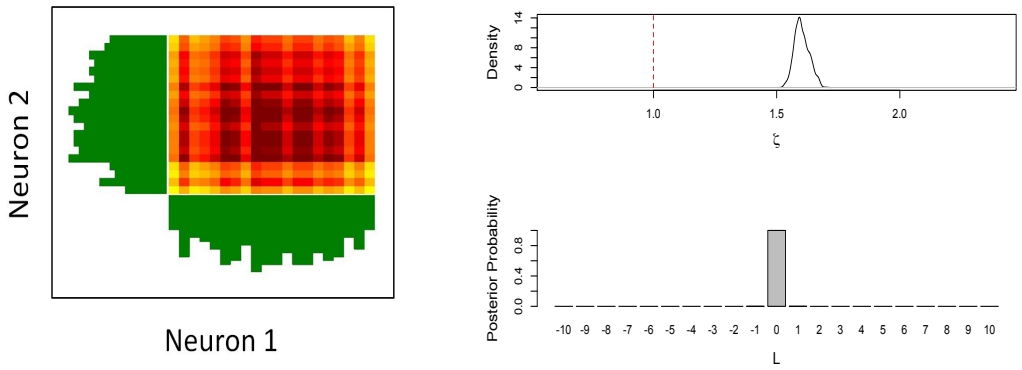


Figure 2.5: **Two dependent neurons in exact synchrony**– The left panel show the frequency of spikes over time. The right panel shows the posterior distribution of ζ and L. Darker cells represent higher frequencies.

posterior median for ζ is 1.598 and the 95% posterior probability interval is [1.548,1.666]. Therefore, ζ identifies the two neurons in exact synchrony with excess co-firing rate than what is expected by independence. Further, the posterior distribution of L shows that the two neurons are in exact synchrony.

Two dependent neurons with lagged co-firing. Similar to the previous example, we set the probability of co-firing to 60% higher than what we obtain by the independence assumption. Similar to the previous two simulations, we generate 40 trials of spike trains each discretized into 100 time bins. The firing probabilities of the first neurons at time t is set to $0.25 + 0.1\sin(2\pi t)$. The second neuron has the same firing probability but at time $t + L$. For different trials, we randomly set L to 3, 4, or 5 with probabilities 0.2, 0.5, and 0.3 respectively. Figure 2.6 shows JPSTH along with the posterior distributions of ζ and L. As before, the posterior distribution of ζ can be used to detect the relationship between the two neurons. For this example, the posterior median and 95% posterior interval for ζ are 1.39 and [1.33,1.44] respectively. Also, our method could identify the three lag values correctly.

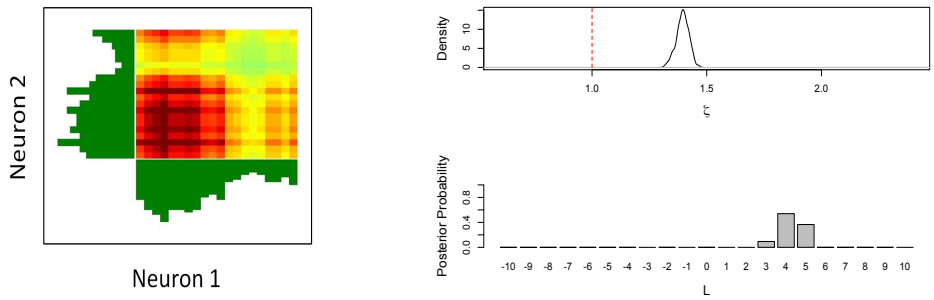


Figure 2.6: **Two dependent neurons in lagged synchrony**– The lag values are set to 3, 4, or 5 with probabilities 0.2, 0.5, and 0.3 respectively. The left panel show the frequency of spikes over time. The right panel shows the posterior distribution of ζ and L. Darker cells represent higher frequencies.

2.3.2 Power analysis

Next, we evaluate the performance of our proposed approach via simulation studies. More specifically, we compare our approach to the method of [30], in terms of statistical power for detecting synchronous neurons. To be precise, given the true value of ζ, we compare the ratio of correctly identifying synchrony between two neurons over a large number of simulated pairs of spike trains. In their approach, [30] find the marginal firing rate of each neuron using natural cubic splines and then evaluate the amount of excess joint spiking using the bootstrap method. Therefore, for our first simulation study, we simulate datasets that conform with the underlying assumptions of both methods. More specifically, we first set the marginal firing rates to $p_t = q_t = 0.2 - 0.1\cos(12\pi t)$, and then generate the spike trains for the two neurons given ζ (i.e., excess joint firing rate). The left panel of Figure 2.7 compares the two methods in terms of statistical power for different values of ζ and different number of trials (20, 30, and 40). Here each trial of spike train has 20 time intervals. For each simulation setting, we generate 240 datasets. In our method, we call the relationship between two neurons significant if the corresponding 95% posterior probability of the ζ does not include 1. For the method proposed by [30], the 95% bootstrap confidence intervals

are used instead. As we can see, our method (solid curve) has substantially higher power compared to the method of [30] (dashed curve). Additionally, our method correctly achieves 0.05 level (dotted line) when $\zeta = 1$ (i.e., the two neurons are independent).

For our second simulation, we generate datasets that do not conform with the underlying assumptions of the two methods. Let $Y = (y_1, \ldots, y_T)$ and $Z = (z_1, \ldots, z_T)$ denote the spike trains for two neurons. We first simulate y_t, i.e., absence or presence of spikes for the first neuron at time t, from Bernoulli(p_t), where $p_t = 0.25 - 0.1\cos(12\pi t)$ for $t \in [0, 0.2]$. Then, we simulate z_t for the second neuron from Bernoulli($b_0 + b_1 y_t$) for given values of b_0 and b_1. We set b_0 (i.e., the baseline probability of firing for the second neuron) to 0.2. When $b_1 = 0$, the two neurons are independent. Positive values of b_1 leads to higher rates of co-firing between the two neurons. When b_1 is negative, the first neuron has an inhibitory effect on the second neuron. For given values of b_1 and number of trials (20, 30, and 40), we generate 240 datasets where each trial has 20 time intervals. The right panel of Figure 2.7 compares the two methods in terms of statistical power under different settings. As before, our method (solid curves) has higher statistical power compared to the method of [30] (dashed curves).

2.3.3 Sensitivity analysis for trial-to-trial variability

As mentioned above, our method can be easily extended to allow for trial-to-trial variability. To examine how such variability can affect our current model, we conduct a sensitivity analysis. Similar to the procedure discussed in the previous section, we start by setting the underlying firing probabilities to $p_t = 0.4 + 0.1\cos(12t)$ and $\zeta = 1.2$. For each simulated dataset, we set the number of trials to 20, 30, 40, and 50. We found that shifting the firing rate of each trial by a uniformly sampled constant around the true firing rate does not substantially affect our method's power since the Gaussian process model is still capable of estimating the underlying firing rate by averaging over trials. However, adding independent

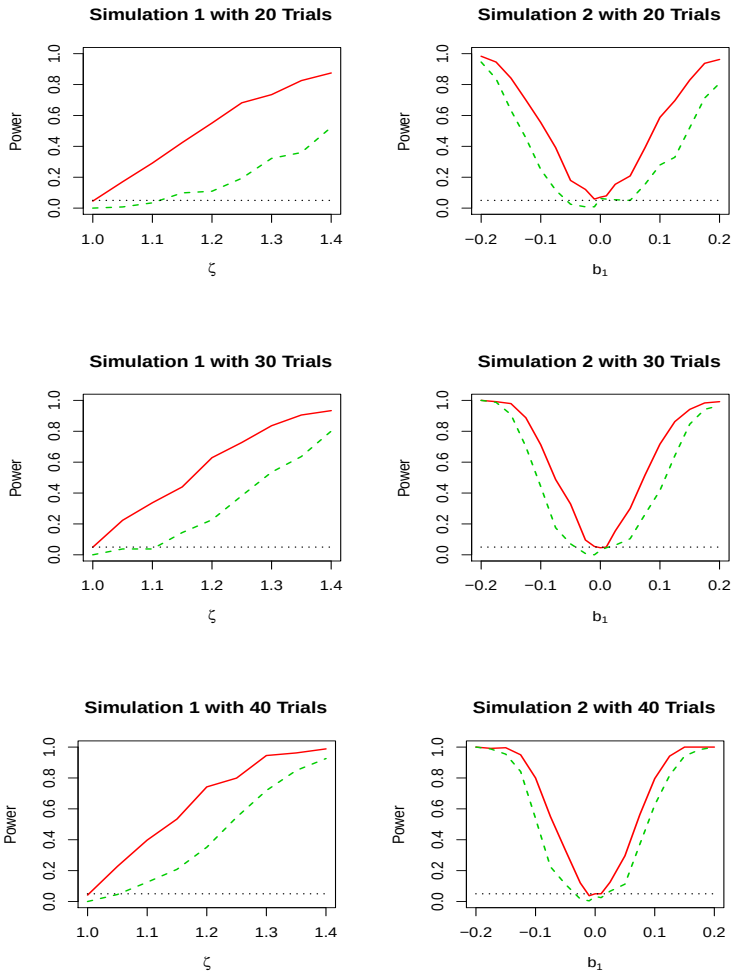


Figure 2.7: **Power analysis**– Comparing our proposed method (solid curves) to the method of [30] (dashed curves) based on statistical power using two simulation studies. Here the dotted lines indicate the 0.05 level.

random noise to each trial (i.e., flipping a fraction of time bins from zero to one or from one to zero) could affect performance, especially if the noise rate (i.e., proportion of flips) is high and the number of trials is low. Figure 2.8 shows the power for different number of trials and varying noise rate from 0 to 10%. As we can see, the power of our method drops slowly as the percentage of noise increases. The drop is more substantial when the number of trials is small (i.e., 20). However, for a reasonable number of trials (e.g., 40 or 50) and a reasonable noise rate (e.g., about 5%) the drop in power is quite small.

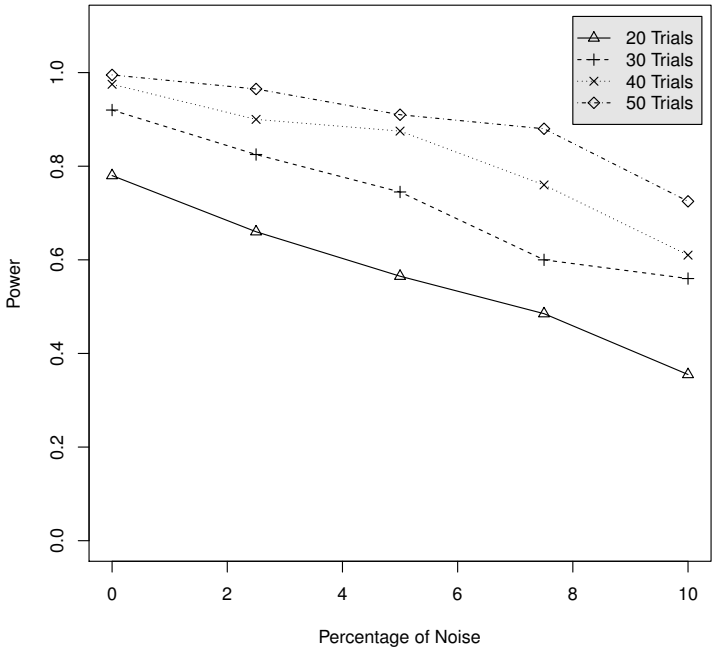


Figure 2.8: **Sensitive analysis for trial-to-trial variability**– Comparing power for varying number of trials and noise rate (i.e., fraction of time bins in a trial flipped from zero to one or from one to zero).

2.3.4 Results for experimental data

We now use our method for analyzing a pair of neurons selected from the experiment discussed above. (We will apply our method to multiple neurons in the next section.) Although we applied our method to several pairs with different patterns, for brevity we present the results for two pair of neurons; for one pair, the relationship changes under different scenarios; for the other pair, the relationship remains the same under both scenarios. Our data include 51 spike trains for each neuron under different scenario (rewarded vs. non-rewarded). Each trail runs for 10 seconds. We discretize the time into 5 ms intervals.

Case 1: Two neurons with synchrony under both scenarios We first present our model's results for a pair of neurons that appear to be in exact synchrony under both

scenarios. Figure 2.9 shows the posterior distributions of ζ and L under different scenarios. As we can see, the posterior distributions of ζ in both cases are away from 1, and $L = 0$ has the highest posterior probability. These results are further confirmed by empirical results, namely, the number of co-firings, correlation coefficients, and the sample estimates of conditional probabilities presented in Figure 2.9. Using the method of [30], the p-values under the two scenarios are $3.2E-11$ and $1.4E-13$ respectively. While both methods provide similar conclusions, their method is limited to detecting exact synchrony only.

Case 2: Two neurons with synchrony under the rewarded scenario only Next, we present our model's results for a pair of neurons appear to be in a moderate synchrony under the rewarded scenario only. Figure 2.10 shows the posterior distributions of ζ and L under different scenarios. In this case, the posterior distributions of ζ is slightly away from 1 in the first scenario; however, under the second scenario, the tail probability of 1 is not negligible. These results are further confirmed by empirical results presented in Figure 2.10: only in the first scenario we observe a moderate difference between the conditional probabilities. Using the method of [30], the p-values under the two scenarios are $2E-4$ and 0.144 respectively.

As discussed above, although for these data the two methods provide similar results in terms of synchrony, our method can be used to make inference regarding possible lag values. Moreover, as we will show in the next section, our method provides a hierarchical Bayesian framework that can be easily extended to multiple neurons.

2.4 Modeling dependencies among multiple neurons

Temporal relationships among neurons, particularly those that change across different contexts, can provide additional information beyond basic firing rates. Because it is possible

to record spike trains from multiple neurons simultaneously, and because network encoding likely spans more than pairs of neurons, we now turn our attention to calculating temporally-related activities among multiple (> 2) simultaneously-recorded neurons.

At lag zero (i.e., $L = 0$), we can rewrite our model for the joint distribution of two neurons in terms of their individual cumulative distributions as follows (we have dropped the index t for simplicity):

$$H(y, z) = \mathcal{C}(F_1(y), F_2(z)) = \left[1 + \beta \prod_{i=1}^{2}(1 - F_i)\right] \prod_{i=1}^{2} F_i \tag{2.14}$$

where $F_1 = F_1(y) = P(Y \leq y)$, $F_2 = F_2(z) = P(Z \leq z)$, and $\beta = \frac{\zeta - 1}{(1-p)(1-q)}$. Note that in this case, $\beta = 0$ indicates that the two neurons are independent. In general, models that couple the joint distribution of two (or more) variables to their individual marginal distributions are called copula models. See [54] for detailed discussion of copula models. Let H be n-dimensional distribution functions with marginals $F_1, ..., F_n$. Then, an n-dimensional copula is a function of the following form:

$$H(y_1, ..., y_n) = \mathcal{C}(F_1(y_1), ..., F_n(y_n)), \text{ for all } y_1, \ldots, y_n \tag{2.15}$$

Here, $\mathcal{C}$ defines the dependence structure between the marginals. Our model for two neurons is in fact a special case of the Farlie-Gumbel-Morgenstern (FGM) copula family [17, 24, 42, 54]. For n random variables $Y_1, Y_2, \ldots, Y_n$, the FGM copula, $\mathcal{C}$, has the following form:

$$\mathcal{C} = \left[1 + \sum_{k=2}^{n} \sum_{1 \leq j_1 < \cdots < j_k \leq n} \beta_{j_1 j_2 \ldots j_k} \prod_{l=1}^{k}(1 - F_{j_l})\right] \prod_{i=1}^{n} F_i \tag{2.16}$$

where $F_i = F_i(y_i)$. Restricting our model to second-order interactions, we can generalize our approach for two neurons to a copula-based model for multiple neurons using the FGM copula family,

$$H(y_1, \ldots, y_n) = \big[1 + \sum_{1 \le j_1 < j_2 \le n} \beta_{j_1 j_2} \prod_{l=1}^{2} (1 - F_{j_l})\big] \prod_{i=1}^{n} F_i \tag{2.17}$$

where $F_i = P(Y_i \le y_i)$. Here, we use $y_1, \ldots, y_n$ to denote the firing status of n neurons at time t; $\beta_{j_1 j_2}$ captures the relationship between the j_1^{th} and j_2^{th} neurons. To ensure that probability distribution functions remain within $[0, 1]$, the following constraints on all $\binom{n}{2}$ parameters $\beta_{j_1 j_2}$ are imposed:

$$1 + \sum_{1 \le j_1 < j_2 \le n} \beta_{j_1 j_2} \prod_{l=1}^{2} \epsilon_{j_l} \ge 0, \quad \epsilon_1, \cdots, \epsilon_n \in \{-1, 1\} \tag{2.18}$$

Considering all possible combinations of ϵ_{j_1} and ϵ_{j_2} in the above condition, there are $n(n-1)$ linear inequalities, which can be combined into the following inequality:

$$\sum_{1 \le j_1 < j_2 \le n} |\beta_{j_1 j_2}| \le 1 \tag{2.19}$$

2.4.1 Illustrative example

To illustrate the model we discussed in the above section, we follow a similar procedure as Section 2.3.4 and simulate spike trains for three neurons such that neurons 1 and 2 are in exact synchrony, but they are independent from neuron 3. Table 2.1 shows the estimated β's along with their corresponding 95% posterior probability intervals using posterior samples from Spherical Hamiltonian Monte Carlo (Spherical HMC). Our method correctly detects the relationship among the neurons: for synchronous neurons, the corresponding β's are

Table 2.1: Estimates of β's along with their 95% posterior probability intervals for simulated data based on our copula-based model. Here, row i and column j shows the estimate of β_{ij}, which captures the relationship between the i^{th} and j^{th} neurons.

β	2	3
1	**0.66 (0.30,0.94)**	0.02 (-0.26,0.27)
2		-0.05 (-0.33,0.19)

significantly larger than 0 (i.e., 95% posterior probability intervals do not include 0), whereas the remaining β's are close to 0 (i.e., 95% posterior probability intervals include 0).

2.4.2 Results for experimental data

We now use our copula-based method for analyzing the experimental data discussed earlier. As mentioned above, during the task the neuronal activity in the prefrontal cortical areas of the rat's brain was recorded under two conditions: rewarded stimulus (lever 1) and non-rewarded stimulus (lever 2). Here, we focus on 5 simultaneously recorded neurons that are active during the task. There are 51 trials per neuron under each scenario. We set the time intervals to 5 ms. Tables 2.2 and 2.3 show the estimates of $\beta_{i,j}$, which capture the association between the i^{th} and j^th neurons, under the two scenarios. Figure 2.11 shows the schematic representation of these results under the two experimental conditions. The solid line indicates significant association between a pair of neurons.

Our results show that neurons recorded simultaneously in the same brain area are correlated in some conditions and not in others. This strongly supports the hypothesis that population coding among neurons (here though correlated activity) is a meaningful way of signaling differences in the environment (rewarded or non-rewarded stimulus) or behavior (going to press the rewarded lever or not pressing) [9]. It also shows that neurons in the same brain region are differentially involved in different tasks, an intuitive perspective but one that

Table 2.2: Estimates of β's along with their 95% probability intervals for the first scenario (Rewarded) based on our copula model.

β	2	3	4	5
1	**0.22(0.07,0.39)**	0.00(-0.07,0.04)	0.03(-0.02,0.15)	0.01(-0.04,0.08)
2		0.03(-0.02,0.18)	0.06(-0.02,0.22)	**0.07(0.00,0.25)**
3			0.08(-0.01,0.26)	**0.21(0.04,0.38)**
4				**0.23(0.09,0.40)**

Table 2.3: Estimates of β's along with their 95% probability intervals for the second scenario (Non-rewarded) based on our copula model.

β	2	3	4	5
1	0.05(-0.02,0.25)	-0.01(-0.09,0.04)	0.15(-0.01,0.37)	0.05(-0.03,0.22)
2		**0.21(0.03,0.41)**	**0.18(0.00,0.37)**	0.03(-0.02,0.19)
3			**0.17(0.00,0.34)**	0.03(-0.02,0.19)
4				0.07(-0.01,0.24)

is neglected by much of behavioral neuroscience. Finally, our results indicate that network correlation is dynamic and that functional pairs– again, even within the same brain area– can appear and disappear depending on the environment or behavior. This suggests (but does not confirm) that correlated activity across separate populations within a single brain region can encode multiple aspects of the task. For example, the pairs that are correlated in reward and not in non-reward could be related to reward-seeking whereas pairs that are correlated in non-reward could be related to response inhibition. Characterizing neural populations within a single brain region based on task-dependent differences in correlated firing is a less-frequently studied phenomenon compared to the frequently pursued goal of identifying the overall function of the brain region based on individual neural firing [76]. While our data only begin to address this important question, the developed model will be critical in application to larger neural populations across multiple tasks in our future research.

2.5 Computation

We use Markov Chain Monte Carlo (MCMC) algorithms to sample from posterior distribution. The typical number of MCMC iterations is 3000 after discarding pre-convergence samples. Algorithm 1 in Appendix shows the overall sampling procedure. We use the slice sampler [52] for the hyperparameters controlling the covariance function of the Gaussian process model. More specifically, we use the "stepping out" procedure to find an interval around the current state, and then the "shrinkage" procedure to sample from this interval. For latent variables with Gaussian process priors, we use the elliptical slice sampling algorithm proposed by [45]. The details are provided in Algorithm 2 in the appendix.

Sampling from the posterior distribution of β's in the copula model is quite challenging. As the number of neurons increases, simulating samples from the posterior distribution these parameters becomes difficult because of the imposed constraints [49, 75, 50, 7, 59]. We have recently developed a new Markov Chain Monte Carlo algorithm for constrained target distributions [36] based on Hamiltonian Monte Carlo (HMC) [15, 53].

In many cases, bounded connected constrained D-dimensional parameter spaces can be bijectively mapped on to the D-dimensional unit ball $\mathbf{B}_0^D(1) := \{\theta \in \mathbb{R}^D : \|\theta\|_2 = \sqrt{\sum_{i=1}^D \theta_i^2} \leq 1\}$, where θ are parameters. Therefore, our method first maps the D-dimensional constrained domain of parameters to the unit ball. We then augment the original D-dimensional parameter θ with an extra auxiliary variable θ_{D+1} to form an extended $(D+1)$-dimensional parameter $\tilde{\theta} = (\theta, \theta_{D+1})$ such that $\|\tilde{\theta}\|_2 = 1$ so $\theta_{D+1} = \pm\sqrt{1 - \|\theta\|_2^2}$. This way, the domain of the target distribution is changed from the unit ball $\mathbf{B}_0^D(1)$ to the D-dimensional sphere, $\mathbf{S}^D := \{\tilde{\theta} \in \mathbb{R}^{D+1} : \|\tilde{\theta}\|_2 = 1\}$, through the following transformation:

$$T_{\mathbf{B}\to\mathbf{S}} : \mathbf{B}_0^D(1) \longrightarrow \mathbf{S}^D,\ \theta \mapsto \tilde{\theta} = (\theta, \pm\sqrt{1 - \|\theta\|_2^2}) \tag{2.20}$$

Note that although θ_{D+1} can be either positive or negative, its sign does not affect our Monte Carlo estimates since after applying the above transformation, we need to adjust our estimates according to the change of variable theorem as follows:

$$\int_{\mathbf{B}_0^D(1)} f(\theta) d\theta_{\mathbf{B}} = \int_{\mathbf{S}_+^D} f(\tilde{\theta}) \left| \frac{d\theta_{\mathbf{B}}}{d\tilde{\theta}_{\mathbf{S}}} \right| d\tilde{\theta}_{\mathbf{S}} \tag{2.21}$$

where $\left| \frac{d\theta_{\mathbf{B}}}{d\tilde{\theta}_{\mathbf{S}}} \right| = |\theta_{D+1}|$. Here, $d\theta_{\mathbf{B}}$ and $d\tilde{\theta}_{\mathbf{S}}$ are under Euclidean measure and spherical measure respectively.

Using the above transformation, we define the dynamics on the sphere. This way, the resulting HMC sampler can move freely on $\mathbf{S}^D$ while implicitly handling the constraints imposed on the original parameters. As illustrated in Figure 2.12, the boundary of the constraint, i.e., $\|\theta\|_2 = 1$, corresponds to the equator on the sphere $\mathbf{S}^D$. Therefore, as the sampler moves on the sphere, passing across the equator from one hemisphere to the other translates to "bouncing back" off the the boundary in the original parameter space.

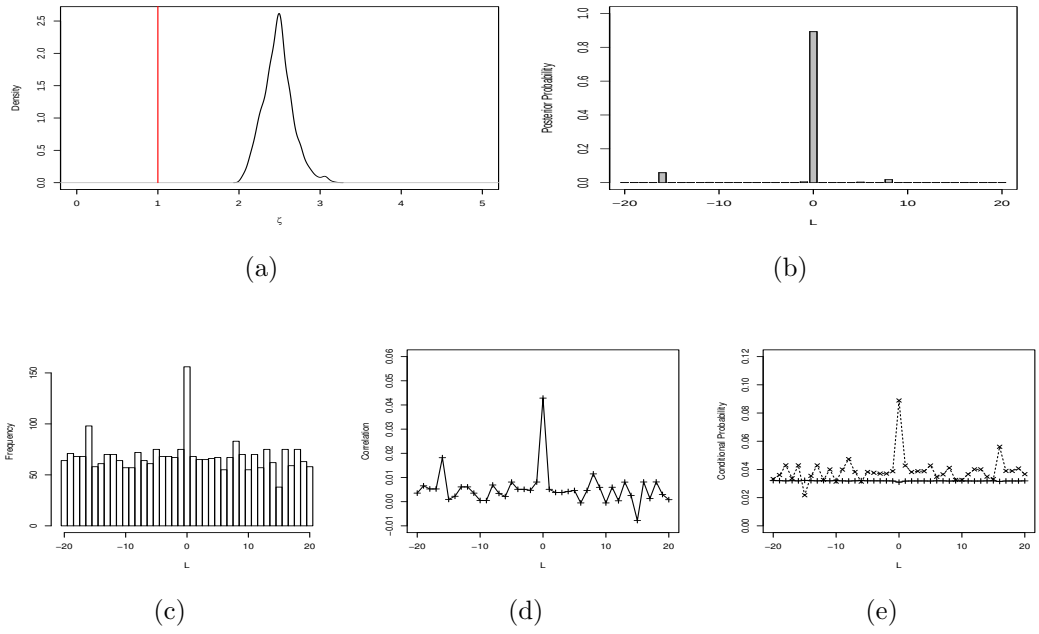
We have shown that by defining HMC on the sphere, besides handling the constraints implicitly, the computational efficiency of the sampling algorithm could be improved since the resulting dynamics has a partial analytical solution (geodesic flow on the sphere). We use this approach, called Spherical HMC, for sampling from the posterior distribution of β's in the copula model. See Algorithm 3 in Appendix for more details.

Using parallelization (i.e., assigning each neuron to a server), our computational method can handle relatively a large number of neurons. The MATLAB implementation of our method runs on a HPC (High Performance Computing) Beowulf cluster with a total of 64 CPUs (CentOS) and equipped with a GPU. For 10 neurons, 20 trials, and 50 time bins, each iteration of MCMC takes 8.4 seconds with acceptance probability of 0.72. For 50 neurons, the time per iteration increases to 24.5 with similar acceptance probability.

While our proposed method takes advantage of several advanced computational techniques, the current implementation of our method is only useful for tens of neurons. This can be improved in future by reducing the computational complexity of our sampling algorithms. For the GP model, the computational complexity is $\mathcal{O}(T^3)$, where T is the number of time bins. This increases linearly with the number of neurons. Note that we can reduce the computational complexity of the GP model to $\mathcal{O}(T)$ by using a Brownian motion instead. To calculate the joint pdf within each time bin of a trial (i.e., using the copula model), the computational complexity increases exponentially by the number of cofiring neurons. The overall complexity increases linearly by the number of trials and the number of time bins.

The space complexity depends on the number of neurons, number of time bins, and number of trials. For a simulation study with 25 neurons, 25 trials, and 50 time bins, the RAM usage is around 4.3GB. Increasing the number of neurons to 50 results in a substantially higher RAM usage close to 7.2GB. If we further increase the number of trials to 50, the RAM usage increases to 10.5GB. If we also increase the number of time bins to 100, the RAM usage increases to 12.2GB.

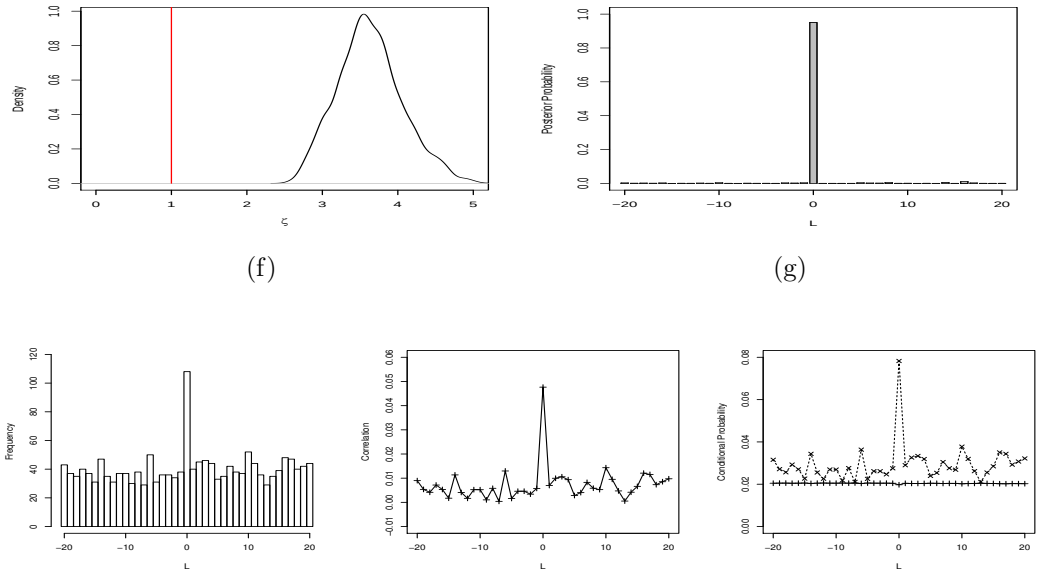
All computer programs and simulated data sets discussed in this paper are available online at `http://www.ics.uci.edu/~babaks/Site/Codes.html`.



(a) (b)

(c) (d) (e)

Case 1–Rewarded

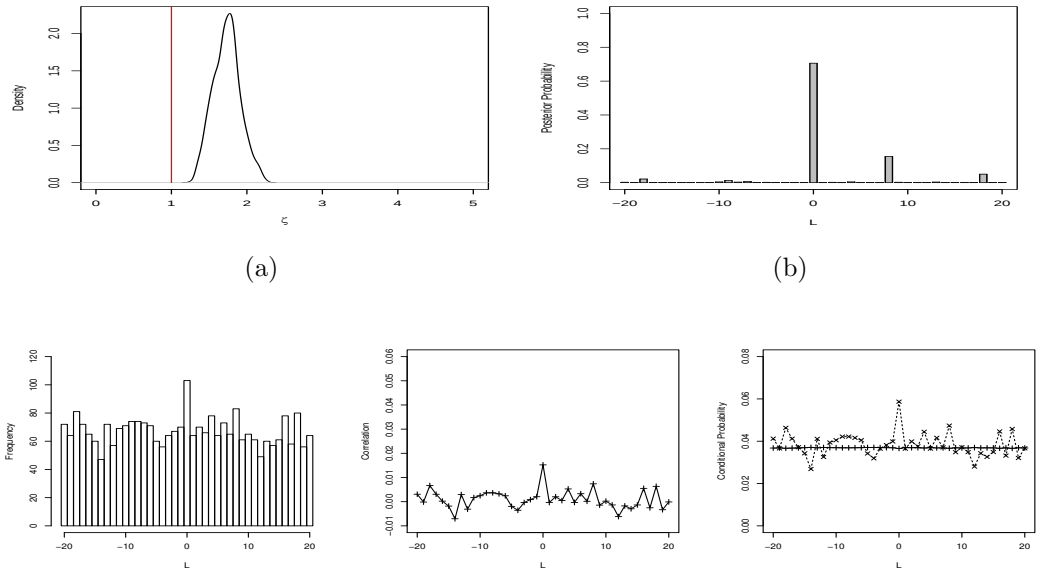


(f) (g)

(h) (i) (j)

Case 1–Non-rewarded

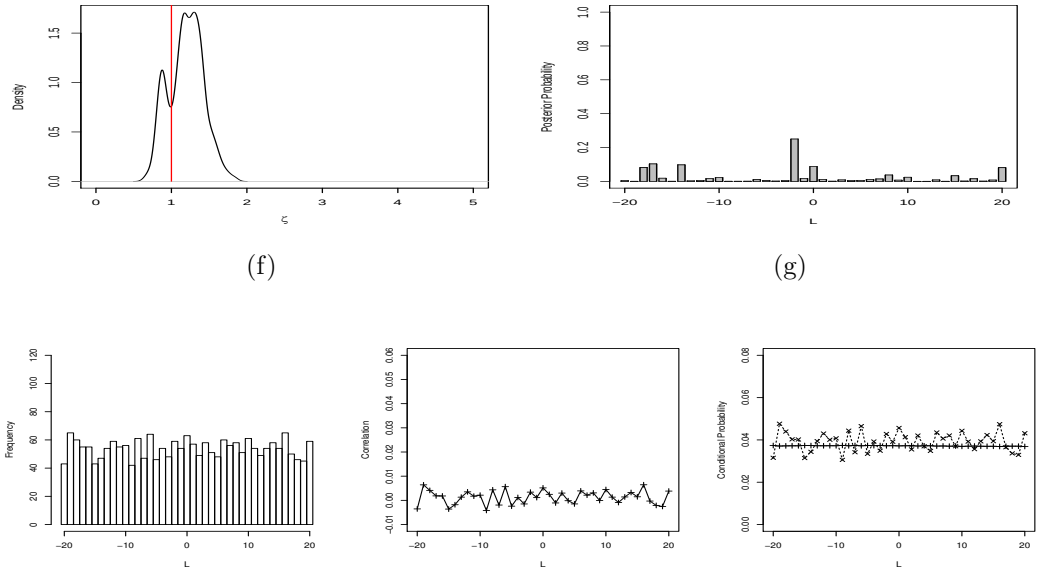
Figure 2.9: **Case 1**: a) Posterior distribution of ζ, b) posterior distribution of lag, c) co-firing frequencies, d) correlation coefficients, and e) estimated conditional probabilities of firing for the second neuron given the firing status (0: solid line, 1: dashed line) of the first neuron over different lag values for the rewarded scenario; (f)-(j) are the corresponding plots for the non-rewarded scenario.



(a) (b)

(c) (d) (e)

Case 2– Rewarded



(f) (g)

(h) (i) (j)

Case 2– Non-rewarded

Figure 2.10: **Case 2**: a) Posterior distribution of ζ, b) posterior distribution of lag, c) co-firing frequencies, d) correlation coefficients, and e) estimated conditional probabilities of firing for the second neuron given the firing status (0: solid line, 1: dashed line) of the first neuron over different lag values for the rewarded scenario; (f)-(j) are the corresponding plots for the non-rewarded scenario.

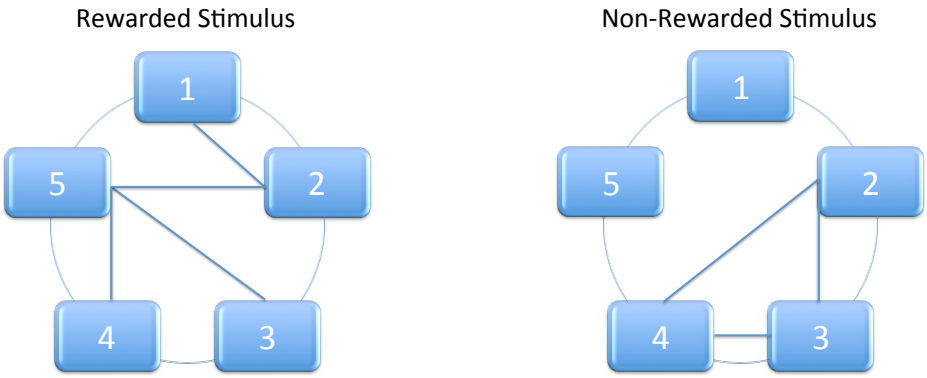


Figure 2.11: A schematic representation of connections between five neurons under two experimental conditions. The solid line indicates significant association.

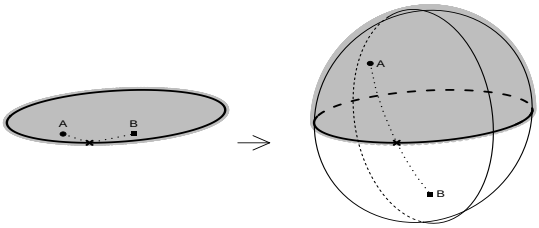


Figure 2.12: Transforming unit ball $\mathbf{B}_0^D(1)$ to sphere $\mathbf{S}^D$.

Chapter 3

A Dynamic Bayesian Model for Characterizing Cross Neuronal Interactions

The model discussed in the above chapter has two main limitations: 1) little power in identifying synchrony among a large number of neurons because of the simplex constrains on the parameters; 2) Unable to identify change in connection structure over time. The following work will be mainly focused on addressing the two limitations.

3.1 Stationary copula model for detecting dependence structure among population of neurons

Again, to model the cross-dependence structure among multiple spike trains, we first discretize the time into small bins. Here, we set the intervals to 100ms. It is possible to use smaller intervals (e.g., 5ms, for example, as we used in the previous chapter) for the station-

ary model discussed in this section; however, our proposed non-stationary model requires longer intervals to be effective and computationally efficient. The spike trains represented by point processes are still transformed to binary time series as before. Let $\mathbf{Y_i} = (Y_{i1}, ..., Y_{iT})$ be the transformed spike train of the i^{th} neuron. Alternatively, We model the joint firing probability of multiple spike trains at time t, $\mathrm{P}_r(Y_{1t} = y_{1t}, ..., Y_{nt} = y_{nt})$, as a function of the marginal probabilities, $\mathrm{P}_r(Y_{it} = y_{it})$. If the neurons are independent, then the joint probability is equal to the product of the marginal probabilities. Note that the joint probability has the following simplex constraint:

$$\sum_{(y_{1t},...,y_{nt})\in(0,1)^n} \mathrm{P}_r(Y_{1t} = y_{1t}, ..., Y_{nt} = y_{nt}) = 1, \text{ for each } t = 1, 2, ..., T. \tag{3.1}$$

To preserve the above constraint, we use a continuous latent variable u_{it} and a threshold τ_{it} corresponding to each Y_{it}, and model the observed spike trains as follows:

$$Y_{it} = \mathbb{1}_{(-\infty,\tau_{it}]}(u_{it}) = \begin{cases} 1, & \text{if } u_{it} \leq \tau_{it} \\ 0, & \text{otherwise.} \end{cases} \tag{3.2}$$

It is worth mentioning that here τ_{it} represents an aggregate of those external conditions that lead to neuronal spiking. As such, τ_{it} may vary as a function of time, as it will be affected by the biological processes leading to action potential, the current state of the network (resulting from the communication among the neurons in the current state of the network), and the external stimuli associated with the experiment. Here 1/0 indicates the presence/absence of spikes within that time bin. For any time t, we assume that $(u_{1t}, ..., u_{nt})$ follows a multivariate Gaussian distribution with mean zero,

$$(u_{1t}, ..., u_{nt})^T \sim N(0, \Sigma), \tag{3.3}$$

For the moment, we assume that Σ does not vary over time so u_{it} is a stationary copula

process. This will be generalized in the dynamic setting. Moreover, the support of the joint distribution of the latent variables is the Cartesian cross of the real lines which can be partitioned into 2^n quadrants by intersecting thresholds, τ_{it}. There are 2^n combination of outputs from n neurons; hence, there are 2^n joint probabilities. Our model guarantees the simplex constraint by mapping the point mass in the 2^n quadrants (which sum to 1) to the 2^n joint probabilities of n neurons. For example, the probability that all the neurons are firing equals to the point mass of the corresponding quadrant as follows,

$$\mathrm{P}_r(Y_{1t} = 1, ..., Y_{nt} = 1) = \mathrm{P}_r\{u_{it} \leq \tau_{it}, \text{for i} = 1, 2, ..., n\}. \tag{3.4}$$

Finally, we specify the the covariance matrix,

$$\Sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{12} & \sigma_{22} & \cdots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{1n} & \sigma_{2n} & \cdots & \sigma_{nn} \end{bmatrix}, \tag{3.5}$$

by decomposing it as follows:

$$\Sigma = \begin{bmatrix} \sigma_{11}^{\frac{1}{2}} & 0 & \cdots & 0 \\ 0 & \sigma_{22}^{\frac{1}{2}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_{nn}^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} 1 & \rho_{12} & \cdots & \rho_{1n} \\ \rho_{12} & 1 & \cdots & \rho_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{1n} & \rho_{2n} & \cdots & 1 \end{bmatrix} \begin{bmatrix} \sigma_{11}^{\frac{1}{2}} & 0 & \cdots & 0 \\ 0 & \sigma_{22}^{\frac{1}{2}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_{nn}^{\frac{1}{2}} \end{bmatrix}. \tag{3.6}$$

3.1.1 On the cross-dependence of the spike trains and the latent processes

In our method, we use ρ_{ij} to capture the relationship between neurons i and j. To show the validity of this approach, we examine the pairwise correlation among multiple spike trains using the observed data,

$$\text{corr}[Y_{it}, Y_{jt}] = \frac{\text{E}[Y_{it}Y_{jt}] - \text{E}[Y_{it}]\text{E}[Y_{jt}]}{\sqrt{\text{Var}[Y_{it}]\text{Var}[Y_{jt}]}} \tag{3.7}$$

where

$$\begin{aligned}
\text{E}[Y_{mt}] &= \text{P}_r(u_{mt} \leq \tau_{mt}) = \text{P}_r(\frac{u_{mt}}{\sigma_{mm}} \leq \frac{\tau_{mt}}{\sigma_{mm}}) = \Phi(\frac{\tau_{mt}}{\sigma_{mm}}) && (3.8)\\
\text{Var}[Y_{mt}] &= \Phi(\frac{\tau_{mt}}{\sigma_{mm}})(1 - \Phi(\frac{\tau_{mt}}{\sigma_{mm}})) && (3.9)\\
\text{E}[Y_{it}Y_{jt}] &= \text{P}_r(u_{it} \leq \tau_{it}, u_{jt} \leq \tau_{jt}) && (3.10)
\end{aligned}$$

$$\begin{aligned}
&= \int_{-\infty}^{\tau_{it}} \int_{-\infty}^{\tau_{jt}} \frac{e^{-\frac{1}{2(1-\rho_{ij}^2)}(\frac{u_{it}^2}{\sigma_{ii}} - 2\frac{\rho_{ij}u_{it}u_{jt}}{\sqrt{\sigma_{ii}\sigma_{jj}}} + \frac{u_{jt}^2}{\sigma_{jj}})}}{2\pi\sqrt{(1-\rho_{ij}^2)\sigma_{ii}\sigma_{jj}}} du_{it}du_{jt} && (3.11)\\
&\equiv f(\frac{\tau_{it}}{\sqrt{\sigma_{ii}}}, \frac{\tau_{jt}}{\sqrt{\sigma_{jj}}}, \rho_{ij}). && (3.12)
\end{aligned}$$

Here Φ is the cumulative distribution function of standard normal distribution. [55] showed that given $\frac{\tau_{it}}{\sqrt{\sigma_{ii}}}$ and $\frac{\tau_{jt}}{\sqrt{\sigma_{jj}}}$, the cumulative distribution of bivariate normal distribution, $f(\frac{\tau_{it}}{\sqrt{\sigma_{ii}}}, \frac{\tau_{jt}}{\sqrt{\sigma_{jj}}}, \rho_{ij})$,

provides comprehensive concordance ordering with respect to ρ_{ij},

$$f(,,-1) \leq f(,,\rho) \leq f(,,\rho') \leq f(,,1) \tag{3.13}$$

where $-1 \leq \rho \leq \rho' \leq 1$. Hence, $\text{corr}[Y_{lt}, Y_{mt}]$ is a non-decreasing function with respect to ρ_{lm}. In addition, when $\rho_{lm} = 0$, $\text{corr}[Y_{lt}, Y_{mt}] = 0$.

Remark. The above derivation suggests that the spike trains maintain the dependence structure as the latent variables. In other words, if the latent variables are positively correlated, negatively correlated, or independent, so are the spike trains. Further, the stronger the correlation between the latent variables, the stronger the correlation between the spike trains. This allows us to make inference about the cross-dependence among spike trains in terms of the correlation parameters ρ_{ij} for the latent variables.

3.1.2 Gaussian Process Prior on the Thresholds

Note that the joint probabilities of spike trains depend on both the latent variables, u_{it}, and thresholds, τ_{it}. While the latent variables specify the dependence structure as discussed above, the thresholds determine the marginal probabilities of firing for each neuron. Specifically, the marginal firing probabilities, $\text{P}_r(Y_{it} = 1)$, depends on the thresholds, τ_{it}, as follows:

$$\text{P}_r(Y_{it} = 1) = \text{P}_r(u_{it} \leq \tau_{it}) = \Phi(\frac{\tau_{it}}{\sigma_{ii}}). \tag{3.14}$$

The marginal probabilities usually follow nonlinear patterns over time. We can accommodate this in our model using the thresholds. That is, our model adjusts τ_{it} in a smooth way to capture the firing probability of neuron i at time t using the observed spike trains. To this

end, we assume τ_{it} has a Gaussian process prior as follows,

$$(\tau_{i1}, ..., \tau_{iT}) \sim \text{GP}(0, C_i), \text{ for i} = 1, 2, ..., n. \tag{3.15}$$

We can still use the same covariance function as discussed before,

$$\begin{aligned} C_i|_{j,k} &= \text{Cov}(\tau_{ij}, \tau_{ik}) \\ &= \lambda_i^2 + \eta_i^2 \exp[-\rho_i^2 (t_i - t_j)^2] + \delta_{jk}\sigma_i^2. \end{aligned} \tag{3.16}$$

in which ρ_i^2 and η_i^2 control smoothness and the height of oscillations respectively. The noise parameter, σ_i^2 (also called the jitter), is essential to improving computation. $\lambda_i^2, \eta_i^2, \rho_i^2$ and σ_i^2 are hyperparameters with their own hyperpriors.

However, when number of time bins, T, increases, computing the likelihood of Gaussian Process models becomes time-consuming as it involves inversion of large-scale covariance matrices. To mitigate the computational cost, we use a Brownian motion prior instead. A Brownian motion is also defined by its mean and covariance; However, its covariance has a simpler form:

$$C_i|_{j,k} = \text{Cov}(\tau_{ij}, \tau_{ik}) = \theta_i \min(t_j, t_k). \tag{3.17}$$

This way, we can reduce the computational complexity of the GP model from $\mathcal{O}(T^3)$ to $\mathcal{O}(T)$.

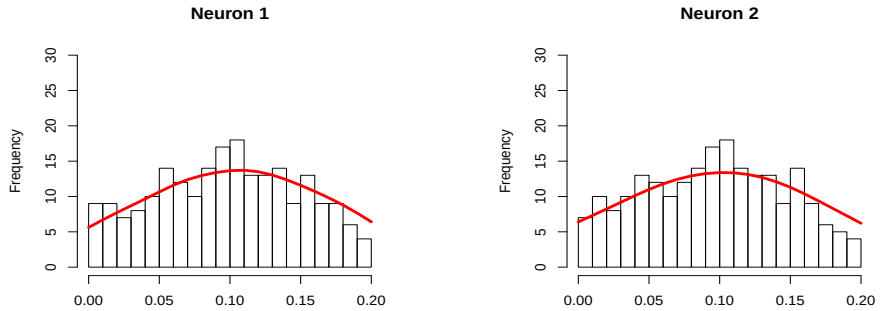
3.1.3 Summary on the stationary model

Our proposed model in this section uses a set of latent variables and thresholds to model the joint probability of multiple spike trains. The latent variables are modeled by a stationary copula process and the thresholds are modeled by Gaussian process models. The copula process model connects the marginal time-varying firing probabilities to the joint probabilities of multiple spike trains. The thresholds control the time-varying marginal firing probabilities, while the latent variables determine the cross-dependence among neurons. Overall, our model involves four types of parameters: 1) latent variables, 2) copula parameters, Σ, 3) thresholds, τ_{it}, and 4) hyperparameters of the Gaussian process model (i.e., the parameters that define C). In our numerical experiments, we used weakly informative priors for all hyperparameters and used Markov Chain Monte Carlo (MCMC) to simulate samples from posterior distributions. Details of our sampling algorithm are provided in Appendix.

3.1.4 An Illustration of the stationary model

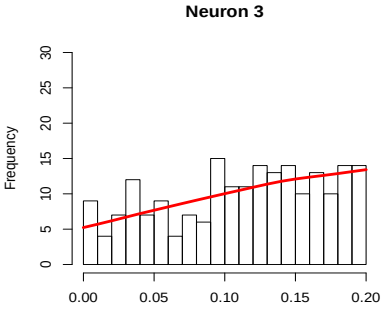
To illustrate the stationary model discussed in the above section, we considered a simple numerical experiment. First, we generated the spike trains of four neurons, where the first pair of neurons were positively correlated, the second pair of neurons were negatively correlated, and that the two pairs are independent of each other. To this end, we generated the time-varying marginal firing probabilities for each neuron and then calculate the joint probabilities of the two neurons by multiplying the product of marginal firing probabilities by an extra term, ζ so that $\zeta = 1$ indicates independence while $\zeta < 1$ and $\zeta > 1$ indicate, respectively, negative correlation and positive correlation between the two neurons. Note that we are using a data generating mechanism that is different from our model in order to test for the robustness of our approach.

In this simulation study, we set $\zeta = 1.3$ for the first pair of neurons and $\zeta = 0.7$ for the

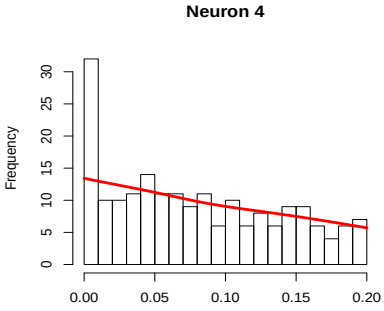


(a) Neuron 1

(b) Neuron 2



(c) Neuron 3



(d) Neuron 4

Figure 3.1: The Peri-Stimulus Time Histogram (PSTH) of 4 neurons with their estimated number of spikes. The red solid line is the estimated number of spikes, which is calculated by multiplying the estimated marginal firing probabilities by number of trials.

second pair of neurons. Table 3.1 shows the posterior estimates of the correlation parameters ρ_{ij}'s with their corresponding 95% posterior intervals. Figure 3.1 shows the peri-stimulus time histogram (PSTH) of these neurons with their estimated number of spikes according to our model. The results presented in Table 3.1 show that our model correctly detects the dependence among the two pairs of neurons while providing a good estimate of the time-varying marginal firing probabilities.

Table 3.1: The posterior estimates of correlation parameters for simulated data with their 95% posterior intervals.

Parameter	**Posterior Estimate**
ρ_{12}	0.75 (0.66,0.84)
ρ_{13}	0.02 (-0.12,0.15)
ρ_{14}	0.12 (-0.01,0.25)
ρ_{23}	-0.05 (-0.19,0.12)
ρ_{24}	0.03 (-0.13,0.17)
ρ_{34}	-0.46 (-0.61,-0.32)

3.2 The Proposed Non-Stationary Model

As mentioned above, one of the main limitations of the method discussed in Chapter 2 is the assumption of constant dependence structure among neurons over time. Hence the main objective of our proposed model framework in this section is to develop a flexible non-stationary method that allows for interactions among neuronal spike trains to vary across time. To this end, we can use a time-varying covariance matrix for the latent variables.

$$(u_{1t}, u_{2t}, ..., u_{nt})^T \sim N(0, \Sigma_t), \tag{3.18}$$

where

$$\Sigma_t = \begin{bmatrix} \sigma_{11}^t & \sigma_{12}^t & \cdots & \sigma_{1n}^t \\ \sigma_{12}^t & \sigma_{22}^t & \cdots & \sigma_{2n}^t \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{1n}^t & \sigma_{2n}^t & \cdots & \sigma_{nn}^t \end{bmatrix}. \tag{3.19}$$

The time-dependent covariance matrix implies a dynamic dependence structure among spike trains. Without imposing a structure, however, the above covariance matrix would become

too complex, especially when n is large. We impose such a structure by introducing a series of time-dependent binary indicators in the stationary copula process model discussed in the previous section,

$$\Sigma_t^* = \begin{bmatrix} \sigma_{11} & \sigma_{12}I_1^tI_2^t & \cdots & \sigma_{1n}I_1^tI_n^t \\ \sigma_{12}I_2^tI_1^t & \sigma_{22} & \cdots & \sigma_{2n}I_2^tI_n^t \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{1n}I_n^tI_1^t & \sigma_{2n}I_n^tI_2^t & \cdots & \sigma_{nn} \end{bmatrix} \tag{3.20}$$

$$= \begin{bmatrix} I_1^t & 0 & \cdots & 0 \\ 0 & I_2^t & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & I_n^t \end{bmatrix} \begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{12} & \sigma_{22} & \cdots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{1n} & \sigma_{2n} & \cdots & \sigma_{nn} \end{bmatrix} \begin{bmatrix} I_1^t & 0 & \cdots & 0 \\ 0 & I_2^t & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & I_n^t \end{bmatrix} \tag{3.21}$$

$$+ \begin{bmatrix} \sigma_{11}(1-I_1^t) & 0 & \cdots & 0 \\ 0 & \sigma_{22}(1-I_2^t) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_{nn}(1-I_n^t) \end{bmatrix}. \tag{3.22}$$

3.2.1 On the indicator I_i^t

Here, $I_i^t = 0$ indicates that the i-th neuron is not correlated with any other neurons at time t. These time-dependent indicators identify neurons that are involved in the network and impose a structure on the covariance matrix. We assume $I_k^t \sim Bernoulli(p_k^t)$. To ensure $0 < p_k^t < 1$, we use the probit link function such that $p_k^t = \Phi(q_k^t)$. A Brownian motion prior is assumed on the hyperparameters q_k^t,

$$(q_k^1, q_k^2, ..., q_k^T) \sim GP(\eta, C_k'), \tag{3.23}$$

where $C'_k|_{i,j} = cov(q_k^i, q_k^j) = \beta_k \min(t_i, t_j)$. Note that the mean of the Brownian motion is not necessarily zero. In general, we could treat η as a hyperparameter. For simplicity, however, we fix it to small number (here, $\eta = -2$) in order to encourage sparsity in the network.

By introducing time-dependent binary indicators in to the copula process, our model is capable of identifying time-varying interactions among multiple neurons. Note that in this setting, the correlation between two neurons becomes $\rho_{ij}^t = \rho_{ij} I_i^t I_j^t$. The two neurons are correlated if both I_i^t and I_j^t are non-zero; ρ_{ij} determines the overall strength of the correlation.

Remark. The time-varying correlation matrix Σ_t^* is positive definite. Note that for any vector x, we have $x^T \Sigma_t^* x \geq 0$ as long as the following matrix is positive definite:

$$\Sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{12} & \sigma_{22} & \cdots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{1n} & \sigma_{2n} & \cdots & \sigma_{nn} \end{bmatrix}, \tag{3.24}$$

The equality holds when

$$\begin{bmatrix} I_1^t & 0 & \cdots & 0 \\ 0 & I_2^t & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & I_n^t \end{bmatrix} x = \mathbf{0} \tag{3.25}$$

and

$$\begin{bmatrix} 1-I_1^t & 0 & \cdots & 0 \\ 0 & 1-I_2^t & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & 1-I_n^t \end{bmatrix} x = \mathbf{0} \tag{3.26}$$

This means that the equality holds only when $x = \mathbf{0}$. Therefore, the time-varying version of correlation matrix, Σ_t^*, is positive definite if Σ is positive definite.

3.2.2 An illustrative example

To illustrate the method discussed in the above section, we used a similar simulation study as the one we presented in the previous section. Here, we generalize ζ_t from the previous example so the dependencies can change over time. For the first pair of neurons, ζ_t is shown in Figure 3.2). For all other pairs, we set $\zeta_t = 1$. Figure 3.3 shows the posterior estimates (along with 95% probability intervals) for $\rho_{ij}^t = \rho_{ij} I_{i,t} I_{j,t}$ over time. The results show that our model correctly identifies the only pair of correlated neurons.

3.2.3 An illustration to highlight the limitations of models that assume stationarity

Next, we use a slightly different setting to highlight the problems with stationary models (and thus emphasize the need for non-stationary approaches such as the one being proposed). In this section, we demonstrate that our model can identify time-varying dependencies while methods based on static models such as [30] and our own previous work [74] provide an

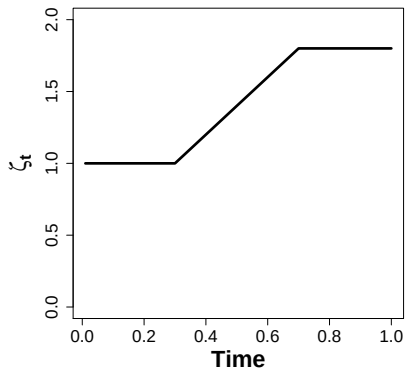


Figure 3.2: The time-dependent ζ_t. At the beginning, $\zeta_t = 1$ suggests that the two neurons are independent. ζ_t starts to increase after then and eventually ends with 1.8 which suggests that the two neurons are positively correlated.

incomplete picture of neuronal activity and, moreover, could give misleading results. To compare our proposed approach to those in [30] and [74], we first generated spike trains for a pair of neurons for which ζ_t follows the pattern shown in the left panel of Figure 3.4. Due to the assumption that the correlation structure is stationary, the method of [30] reports a p-value of 0.555 for the test of $H_0 : \zeta = 1$ (independence) vs. $H_a : \zeta \neq 1$. The method of [74] reports a 95% credible interval of $[0.875, 1.144]$ for ζ which covers 1. No significant evidence of dependence between the neuronal spike trains can be found by both methods. Hence both methods tend to be incapable of detecting the synchrony between the two neurons when this synchrony is brief relative to the period of non-synchrony. We also applied the static version of our proposed model (described in Section 3.1) to the simulated data. The 95% credible interval for the correlation parameter is $[-0.039, 0.066]$ which covers 0 and thus yields no significant evidence for dependence between the two neurons. In contrast to these stationary approaches, the proposed dynamic model captures the time-varying pattern of synchrony between the two neurons as shown in Figure 3.4 (Right). This illustration serves as a reminder that when the true process follows a dependence structure that is time-varying (which is the more likely scenario compared to the simplistic assumption of stationarity),

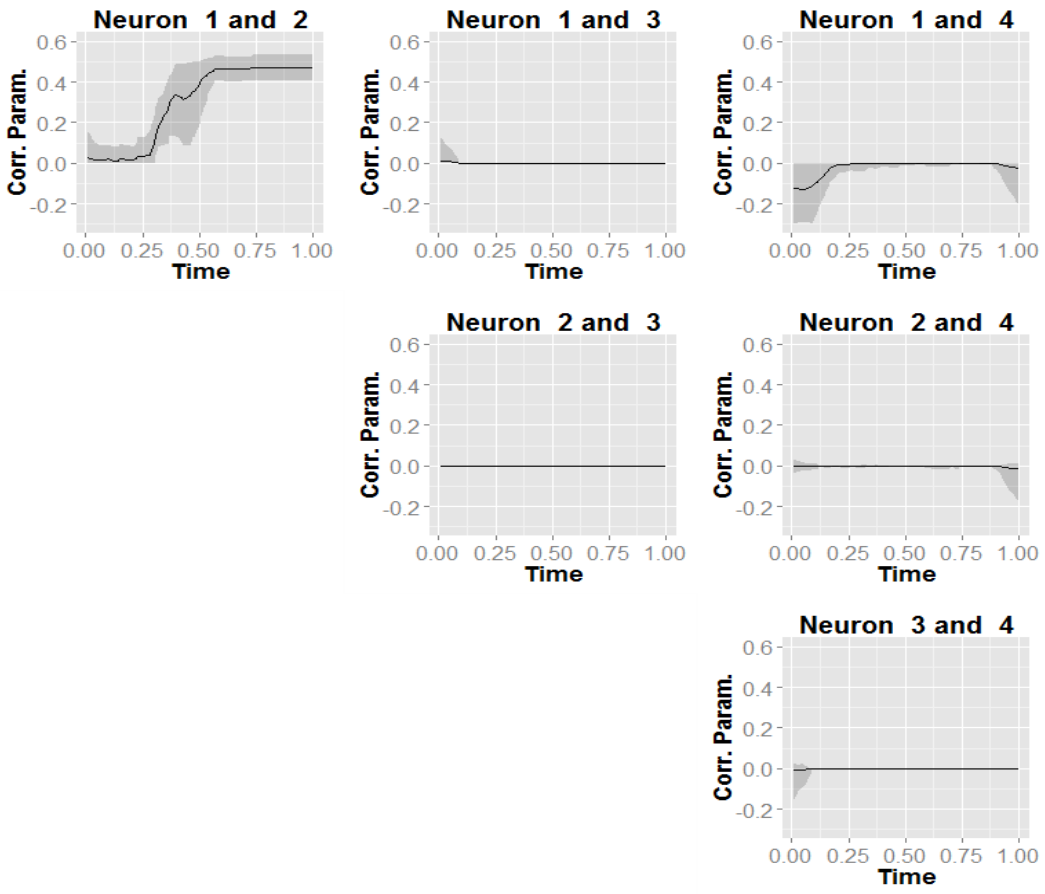


Figure 3.3: The posterior estimates of correlation parameters, $\rho_{ij}^t = \rho_{ij} I_{i,t} I_{j,t}$, with 95% credible intervals. The solid lines are the posterior estimates and the gray areas are the 95% credible intervals.

then it would be more appropriate to fit a non-stationary model instead of a static model.

To further illustrate the advantage of our dynamic method, we conducted two simulation studies to compare our dynamic model with the method in [30] which is the state-of-the-art approach for studying cross-neuronal interactions. For the first simulation study, we compared the two methods in terms of their ability to correctly detect the correlation between two neurons. To this end, we simulated data according to the approach we discussed above. Namely, the two neurons are independent ($\zeta = 1$) for the first 80% of time and dependent ($\zeta \geq 1$) for the last 20% of time. We conducted simulations for 5 scenarios where we

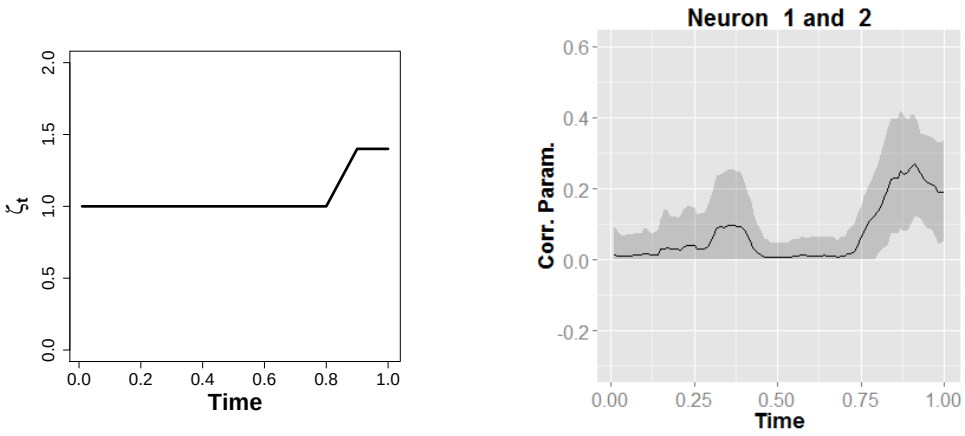
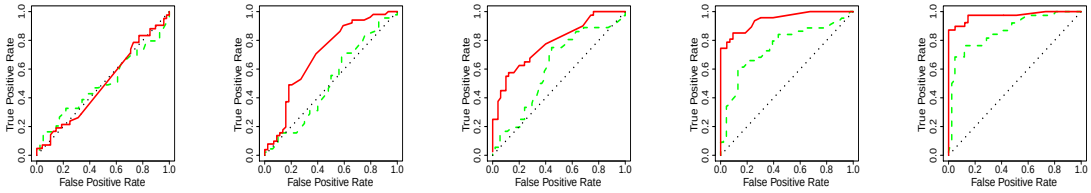


Figure 3.4: **Left:** The time-dependent ζ_t. For the first 80% of time, the two neurons are independent ($\zeta_t = 1$). For the remaining 20% of time, the two neurons become correlated ($\zeta_t > 1$). **Right:** The posterior estimates of correlation parameters, $\rho_{ij}^t = \rho_{ij} I_{i,t} I_{j,t}$, with 95% credible intervals. The solid lines are the posterior estimates and the gray areas are the 95% credible intervals. Note that in our model $\rho = 0$ corresponds to $\zeta = 1$.

set the extra term ζ for the last 20% of time to 1.0, 1.2, 1.4, 1.6, and 1.8 respectively. Corresponding to each pair (which we call "Pair1") under each of the scenarios, we also simulates an independent pair (which we call "Pair2") where $\zeta = 1$ over the entire time. We then apply both models to each pair of neurons and make inference about their dependence. For scenarios 2-5, the models are expected to identify "Pair1" as dependent, while identifying "Pair2" as independent. The first scenario was used as the control where both pairs should have been identified pairs as independent. We repeated each scenario 100 times and use ROC curves to assess models' performance. Note that for the first scenario, we expected the ROC curve to be diagonal. For the remaining scenarios, the higher the area under the ROC curve, the better the model. Figure 3.5 shows the ROC curves, which illustrates the performance of the two models (our proposed in red; [30] in green) in detecting dependence between neurons. The results show that as the strength of correlation increases (ζ increases from 1.0 to 1.8), both methods exhibit an increase in power. Moreover, the results demonstrate that our proposed dynamic model always outperforms the method in [30].



(a) $\zeta = 1.0$ (b) $\zeta = 1.2$ (c) $\zeta = 1.4$ (d) $\zeta = 1.6$ (e) $\zeta = 1.8$

Figure 3.5: ROC curves which illustrate the performance of the two models in terms of detecting dependence between pair of neurons. Our proposed dynamic model is color red; the method in [30] is color green.

For the second simulation study, we compared the prediction power of the two methods. We still simulated the datasets under 5 scenarios as discussed above. This time, however, we randomly selected two-thirds of the data to train both models (i.e., estimate parameters), and then test the models on the remaining one-third of the data by using the firing status of one neuron to estimate the firing probability for the other neuron. We derived the estimated firing probability, $\hat{p}_{2k}$ for the k^{th} observation in the test set for the second neuron using parameters estimated from the training data. Based on the estimated model-based firing probability $\hat{p}_{2k}$, model performance was evaluated in terms of predictive log-likelihood (PL),

$$\text{PL} = \sum_{k \in \text{test set}} y_{2k} \log(\hat{p}_{2k}) + (1 - y_{2k}) \log(1 - \hat{p}_{2k}).$$

Thus, the PL is a valid measure for assessing the capability of a model (with parameters estimated from the training data) to predict values in the testing data. The model with a larger PL value is considered to have greater predictive power. As before, we repeated each scenario 100 times. Figure 3.6 shows the 2.5−th and 97.5−th percentiles of the distribution of PL values for each scenario using our proposed model (shown as green intervals) and the method in [30] (shown as red intervals). Again, the results show that both models display

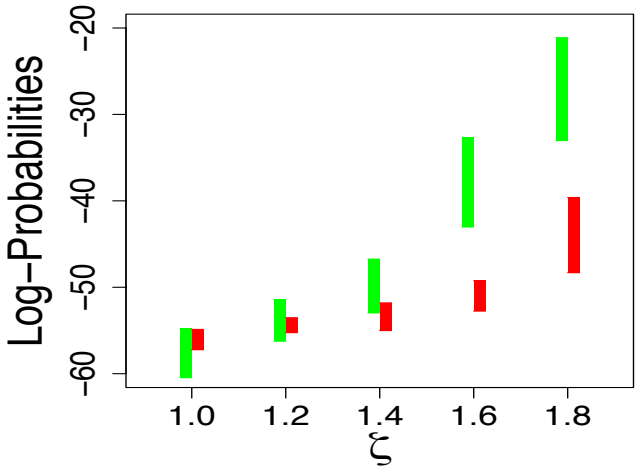


Figure 3.6: 95% intervals of the predictive log-likelihood (PL) of the test sets using our dynamic model (in green) and the method in [30] (in red) for different values of ζ over the last 20% of time interval. Results are based on 100 simulated datasets.

an increasing predictive power as the strength of correlation between two neurons increases. Moreover, consistent with the previous results, our proposed dynamic model outperforms the method in [30].

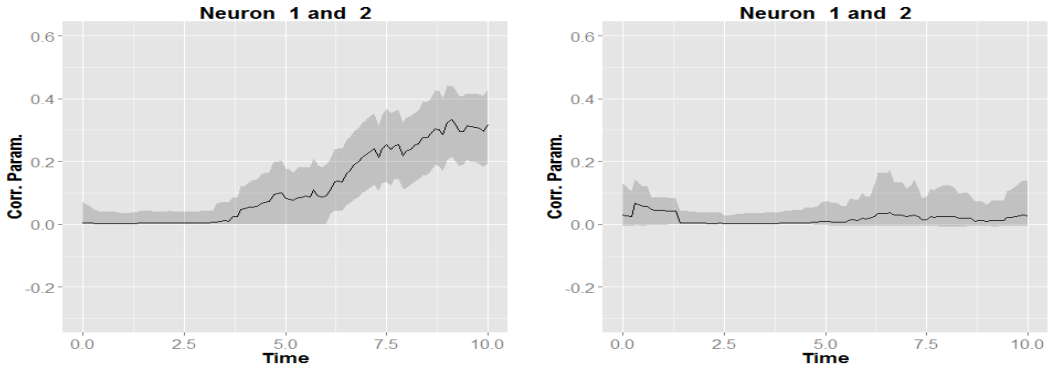
3.2.4 Computation

As mentioned above, we use MCMC to simulate samples from posterior distributions. The current implementation of our method can be used for a moderate (tens) to large (hundreds) number of simultaneously recorded neurons. The bottleneck for our method is sampling from the posterior distribution of latent variables since we need to draw samples from truncated multivariate normal distributions. Currently, we use the R package *tmvtnorm* for this purpose. By introducing the indicator variables into the model, the computation cost of sampling latent variables reduces substantially since it depends on the number of correlat-

ed neurons only (i.e., number of non-zero indicators). For instance, if only 10 out of 100 neurons are in fact involved in the network, the computation cost is mainly dominated by sampling the latent variables for those 10 correlated neurons. Therefore, in many cases, our method can be applied to hundreds of neurons with a sparse network. In addition, the computational cost can be further reduced by more efficient samplers for truncated multivariate normals [e.g., 69]. For the copula parameters and thresholds, we use the elliptical slice sampler, which is especially suited for parameters with Gaussian process priors. Another computationally-intensive part of our algorithm is sampling from the posterior distribution of hyperparameters in the GP model. Using the usual covariance function with the exponential form (3.16), the computational complexity is $O(T^3)$, where T is the number of time bins. However, in our implementation, we reduced this to $O(T)$ by using a Brownian motion model instead (3.17). Overall the computational complexity increases with the number of correlated neurons, number of time bins and number of trials. In our simulation studies with 4 correlated neurons, 100 time bins, and 40 trails, the computational time per iteration is approximately 2.5 seconds.

3.2.5 Analysis of spike train data

As a preliminary step, we first applied the static model to the spike train data of two neurons. The results show that under the rewarded stimulus the 95% credible interval for the correlation parameter was $[0.035, 0.147]$ compared to $[-0.039, 0.049]$ under the non-rewarded stimulus. This suggests that there was some sort of aggregated synchrony between the two neurons under rewarded stimulus but no evidence of aggregated synchrony under the non-rewarded stimulus. The proposed model suggests an interesting result: that the temporal relationship activity between the two neurons differed based on whether the stimulus predicted a rewarded or non-rewarded outcome. As noted, the static analysis above has limitations because it does not portray a complete picture of the time-varying cross-dependence between



(a) Lever 1 (Rewarded) (b) Lever 2 (Non-rewarded)

Figure 3.7: The estimated correlation parameters among the two neurons. The solid line is the posterior estimate and the gray area is the corresponding 95% credible interval.

neurons. As a next step, we applied the dynamic model to the same pair of neurons. Figure 3.7 shows the estimated correlation parameters based on our proposed dynamic model. The results suggest that under the rewarded stimulus, the two neurons started to co-fire after 4 seconds. In contrast, for the unrewarded stimulus, the two neurons remained uncorrelated throughout the experiment. As noted above, the strong cross-dependence between these two neurons was seen during the reward-receipt epoch in rewarded trials. This comparison allowed us to reliably say that the onset of correlation in rewarded trials was not simply a by-product of stimulus presentation (as stimuli are presented in both contexts) but has some relationship between the outcomes or behaviors. Thus, neurons 1 and 2 (from Figure 3.7) are likely members of a population encoding reward or a behavior related to reward-seeking.

There are a number of reasons that explain why neuronal populations may exhibit synchrony. In the periphery, sensory stimuli can simultaneously activate neurons, producing sensory-evoked synchrony. In some cases, synchrony arises from correlated input via, for example, neurons or neural populations that innervate multiple target populations, driving these multiple targets at the same time. For example, in the prefrontal cortex, where neurons studied here were recorded, synchrony is thought to result from strong, correlated, patterned input from the hippocampus and possibly other brain areas [3]. In other cases, notably in cortex,

interactions between inhibitory GABA interneurons and pyramidal neurons are responsible for generating synchrony across populations (see [19]).Correlated activity can also be seen among populations of neurons when activity increases and is maintained, as has been seen, for example, in synchronous activity in the prefrontal cortex underlying working memory (see [71]).

Next, we examined the cross-dependence between all of the 6 neurons in the data. Figures 3.8 and 3.9 show the time-varying correlation parameters among these neurons under different scenarios. For simplicity, in Figures 3.10 and 3.11, we divided the time series into two time intervals: Period 1 covers the first 5 seconds following the stimulation onset (i.e., $[0, 5]$ seconds) while Period 2 covers the last 5 seconds after stimulation onset (i.e., $(5, 10]$ seconds). Within Period 1 of the rewarded scenario, neurons 2, 5, and 6 seemed to be weakly correlated, while others were not at all implicated in the network. During Period 2, neurons 1, 2, 3, and 4 became strongly correlated (Figure 3.10). Under the non-rewarded stimulus, neuron 1, 3, and 4 are correlated throughout both epochs. Towards the end of Period 2 (approximately $7 - 10$ seconds), neurons 5 and 6 join the network with moderate correlations (Figure 3.11). The results suggest that neurons 1, 3, and 4 are involved in the network under both scenarios; whereas, neuron 2 plays a differential role under the two scenarios: it is always involved in the correlation network under rewarded stimulus, while it remains isolated from the network under non-rewarded stimulus. In addition, neurons 5 and 6 vary under different stimuli: they are strongly involved in the network at the beginning following presentation of the rewarded stimulus and weakly involved in the network at the end of session after presentation of the non-rewarded stimulus. The results described here reveal new information about network encoding of different aspects of the behavioral task, in particular providing us two powerful and intersecting insights into population coding of behavior. First, our model reliably measures correlation among groups of neurons over prolonged periods of time. This is an important advance in understanding how events and behaviors can influence synchronous activity in neural populations. Intriguingly, neurons 2

and 5 exhibit synchronous activity in what appears to be the time between lever-press and reward acquisition. Thus by differentially synchronizing different populations of neurons over time, we see a continuum of population representation of the reward-seeking task. Importantly, the degree of synchronization is different in the non-rewarded condition. In this case, neurons 1, 3, and 4 exhibit synchronous activity rapidly after the onset of the non-reward-predicting tone. This synchronization may serve as a response-inhibition signal that allows the animal to withhold responding to the non-rewarded lever.

Brain regions such as the ventral medial prefrontal cortex have frequently been characterized as playing a role in response inhibition: inactivation of these areas increases behavior and stimulation of them suppresses it [62]. The neural signaling underlying this type of executive control has remained elusive however, although some neurons have been shown to be activated during behavioral suppression [39]. However, it is unlikely that the small numbers of neurons showing increased activation in these circumstances represents the mechanism by which populations of neurons encode behavior. Rather, population signaling, reflected in correlated firing, is likely to underscore neural coding of these relatively sophisticated behaviors.

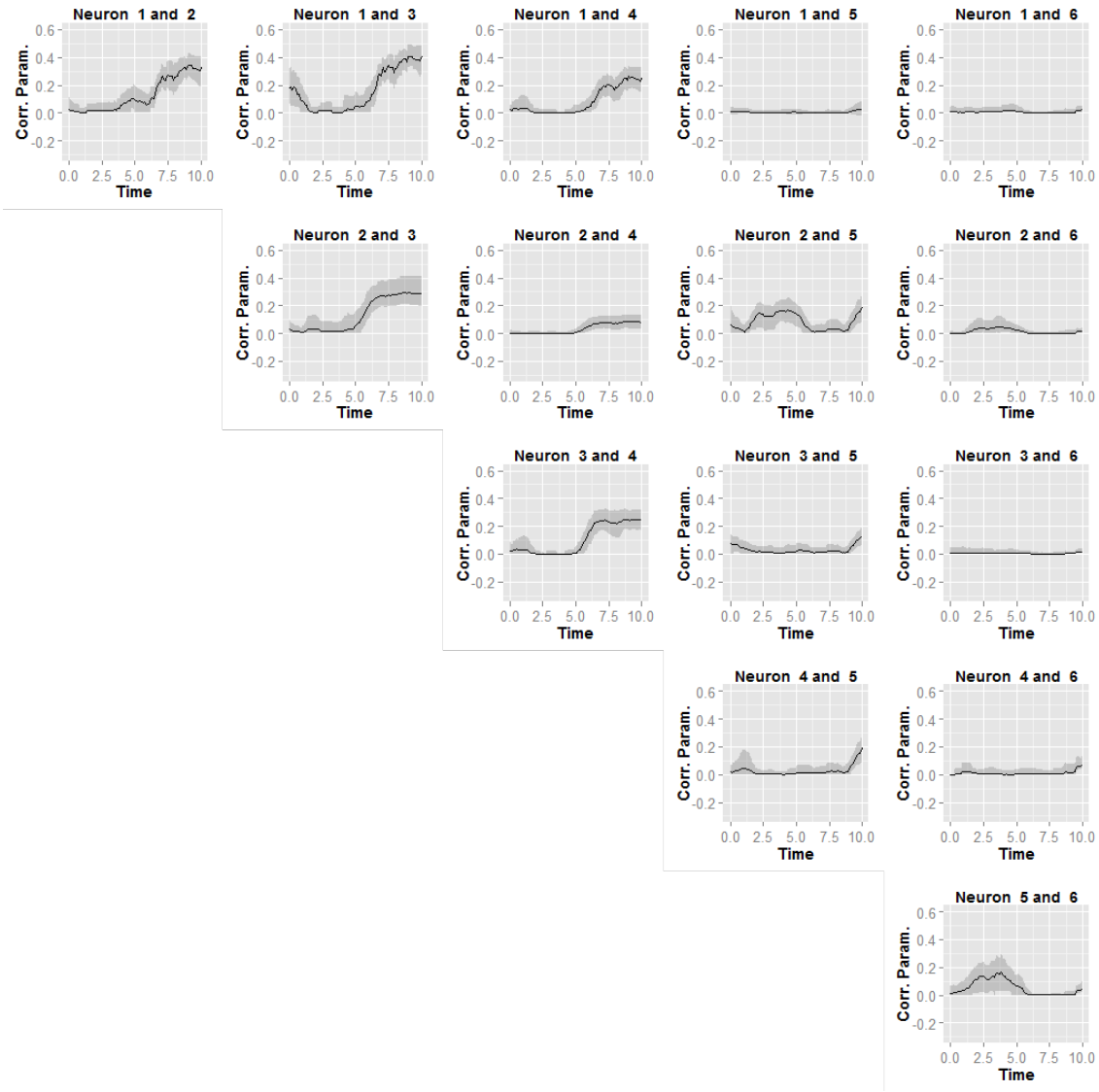


Figure 3.8: The estimated correlation parameters among the 6 neurons under rewarded stimulus. The solid lines are the posterior estimates and the gray areas are the corresponding 95% credible intervals.

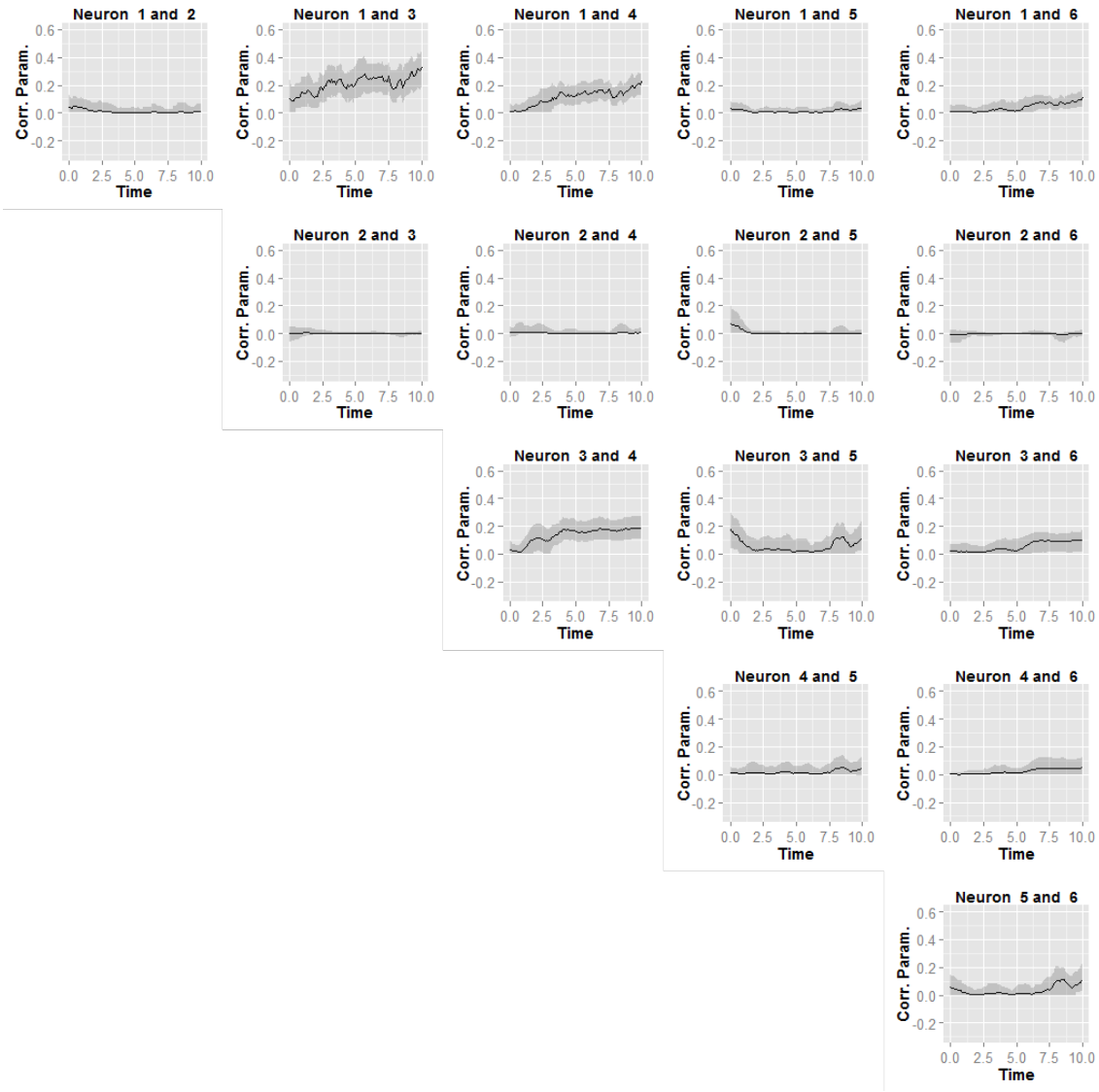
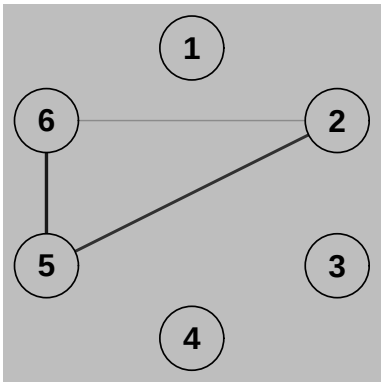
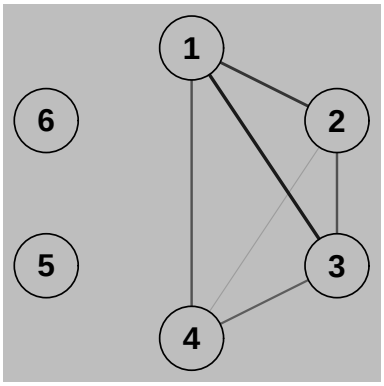


Figure 3.9: The estimated correlation parameters among the 6 neurons under non-rewarded stimulus. The solid lines are the posterior estimates and the gray areas are the corresponding 95% credible intervals.

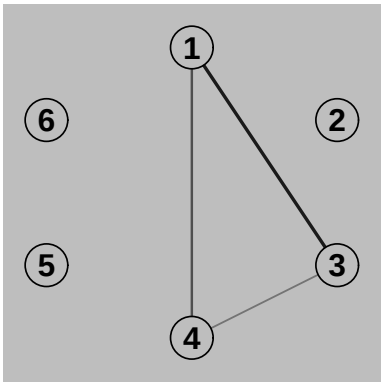


(a) Period 1

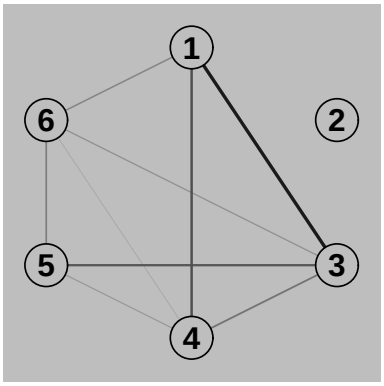


(b) Period 2

Figure 3.10: Graphical representation of correlation network under rewarded stimulus. The nodes are the neurons and the edges indicate the connection between neurons. The grayscale indicates the strength of connection.



(a) Period 1



(b) Period 2

Figure 3.11: Graphical representation of correlation network under non-rewarded stimulus. The nodes are the neurons and the edges indicate the connection between neurons. The grayscale indicates the strength of connection.

Chapter 4

A Dynamic Bayesian Model for Detecting Neuronal Communities

In chapter 3 we discussed a Bayesian model which can capture dynamics of dependence structure among neurons over time. However, this model assumes that there is only one subpopulation of connected neurons. To relax this assumption, we propose a flexible, yet robust semi-parametric Bayesian method for capturing multiple communities of neurons by simultaneous modeling of their spike trains. By identifying communities (subsets) of neurons that distinguish one type of decision from another, we expect our method to provide insights into how neural systems encode decision-making variables as well as other cognitive functions more broadly.

4.1 A Stationary Model for Community Detection

To detect neuronal communities among simultaneously recorded neurons, still we first discretize time to transform their corresponding spike trains, represented by point processes

(Figure 1.1), to binary time series which consist of 1s and 0s, where 1 indicates presence and 0 indicates absence of spikes in each time bin. Here, we still set the time intervals to 100ms due to proposed non-stationary model benefits from longer intervals. Let $\mathbf{Y_i} = (Y_{i1}, ..., Y_{iT})$ be the transformed spike train of the i^{th} neuron. As discussed in the above chapter, we still model the joint firing probability of multiple spike trains at time t, $\mathrm{P}_r(Y_{1t} = y_{1t}, ..., Y_{nt} = y_{nt})$, as a function of the marginal probabilities, $\mathrm{P}_r(Y_{it} = y_{it})$, by introducing the latent variables, u_{it}. The latent variables, however, will follow Gaussian distribution with covariance matrix that is specified differently from the one in Chapter 3. In this section we will be mainly focused on how to define the structure of the covariance matrix so as to capture the community structure of neuronal networks.

In order to detect subsets of correlated neurons, we use Ewens sampling formula [16] to allocate neurons into K partitions without pre-specifying K. We start by assigning the first neuron into an empty subset. Given the previous $j-1$ assignments, we assign the j^{th} neuron either to an existing subset, M, with probability

$$\frac{|M|}{\alpha + j - 1}$$

or to an empty subset (i.e., the neuron starts a new partition) with probability

$$\frac{\alpha}{\alpha + j - 1}$$

Here, $|M|$ is the number of items previously assigned to M, and α is a hyperparameter whose value determines the number of partitions: larger values of α increases the probability of new partitions. The resulting partition distribution is known as Ewens distribution, which defines a prior distribution on any specific partition denoted as $\{z_1, z_2, \ldots, z_n\}$, where z_i assigns the i^{th} neuron to one of the K partitions. Given the partition, we rewrite the correlation matrix

of latent variables, R, as follows:

$$R = \begin{bmatrix} 1 & \rho_{12}\mathbb{1}_{\{z_1=z_2\}} & \cdots & \rho_{1n}\mathbb{1}_{\{z_1=z_n\}} \\ \rho_{12}\mathbb{1}_{\{z_1=z_2\}} & 1 & \cdots & \rho_{2n}\mathbb{1}_{\{z_2=z_n\}} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{1n}\mathbb{1}_{\{z_1=z_n\}} & \rho_{2n}\mathbb{1}_{\{z_2=z_n\}} & \cdots & 1 \end{bmatrix}. \tag{4.1}$$

Here ρ_{ij} can be interpreted as the correlation between the i^{th} and j^{th} latent variables if the i^{th} and j^{th} neurons belong to the same partition (community). The correlation between two neurons is 0 if they belong to different partitions (communities).

4.1.1 A simulation study for the stationary model

We start by evaluating our stationary model based on a simulation study, where we randomly generate spike trains for 10 neurons. There are two subsets of correlated neurons as shown in Figure 4.1. To generate the data, we first generate the latent variables from a multivariate Gaussian distribution and then generate the spike trains by setting a pre-specified threshold on the continuous latent variables. The correlation is set to 0.45 if there is connection between two neurons, and it is set to zero otherwise. We apply our stationary model to the resulting data. The estimated correlation matrix presented in the Table 4.1 shows that our method correctly identify the underlying structure: all independent neurons have non-significant correlation (i.e., the corresponding 95% posterior probability includes zero).

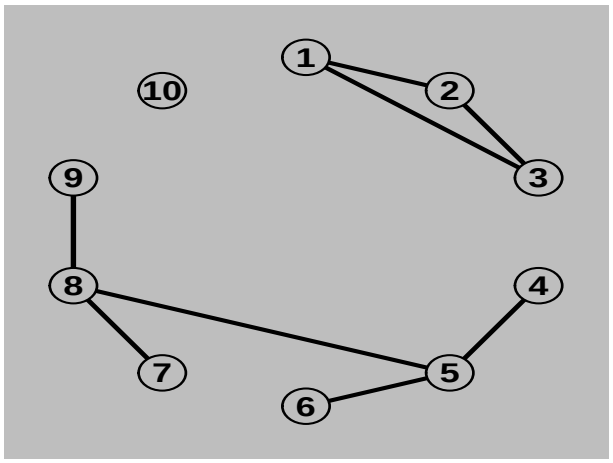


Figure 4.1: The community structure of correlated neurons. Each node represents one neuron and each edge represents the connection between a pair of neurons.

Table 4.1: The estimated pair-wise correlation parameters among the 10 simulated neurons

1	**0.45(0.31,0.59)**	**0.52(0.33,0.69)**	0.02(-0.16,0.19)	-0.01(-0.17,0.14)	0.00(-0.13,0.14)	0.04(-0.10,0.21)	0.00(-0.11,0.12)	0.01(-0.12,0.15)	0.00(-0.10,0.11)
0.45(0.31,0.59)	1	**0.47(0.31,0.64)**	0.03(-0.12,0.16)	0.00(-0.12,0.11)	0.00(-0.13,0.12)	-0.02(-0.17,0.13)	0.01(-0.15,0.16)	0.02(-0.12,0.17)	0.00(-0.12,0.11)
0.52(0.33,0.69)	**0.47(0.31,0.64)**	1	0.00(-0.13,0.14)	0.01(-0.11,0.15)	0.00(-0.15,0.14)	0.02(-0.11,0.16)	0.01(-0.10,0.15)	0.03(-0.11,0.18)	0.01(-0.11,0.16)
0.02(-0.16,0.19)	0.03(-0.12,0.16)	0.00(-0.13,0.14)	1	**0.42(0.27,0.55)**	0.05(-0.0.10,0.22)	-0.03(-0.18,0.11)	0.01(-0.15,0.17)	-0.04(-0.21,0.12)	-0.02(-0.17,0.12)
-0.01(-0.17,0.14)	0.00(-0.12,0.11)	0.01(-0.11,0.15)	**0.42(0.27,0.55)**	1	**0.44(0.30,0.59)**	0.03(-0.12,0.20)	**0.51(0.35,0.66)**	-0.03(-0.18,0.11)	0.00(-0.14,0.15)
0.00(-0.13,0.14)	0.00(-0.13,0.12)	0.00(-0.15,0.14)	0.05(-0.0.10,0.22)	**0.44(0.30,0.59)**	1	0.03(-0.11,0.18)	0.01(-0.14,0.16)	0.04(-0.11,0.20)	0.01(-0.14,0.17)
0.04(-0.10,0.21)	-0.02(-0.17,0.13)	0.02(-0.11,0.16)	-0.03(-0.18,0.11)	0.03(-0.12,0.20)	0.03(-0.11,0.18)	1	**0.46(0.29,0.61)**	0.05(-0.08,0.21)	0.02(-0.13,0.17)
0.00(-0.11,0.12)	0.01(-0.15,0.16)	0.01(-0.10,0.15)	0.01(-0.15,0.17)	**0.51(0.35,0.66)**	0.01(-0.14,0.16)	**0.46(0.29,0.61)**	1	**0.44(0.28,0.60)**	0.00(-0.12,0.13)
0.01(-0.12,0.15)	0.02(-0.12,0.17)	0.03(-0.11,0.18)	-0.04(-0.21,0.12)	-0.03(-0.18,0.11)	0.04(-0.11,0.20)	0.05(-0.08,0.21)	**0.44(0.28,0.60)**	1	0.00(-0.13,0.13)
0.00(-0.10,0.11)	0.00(-0.12,0.11)	0.01(-0.11,0.16)	-0.02(-0.17,0.12)	0.00(-0.14,0.15)	0.01(-0.14,0.17)	0.02(-0.13,0.17)	0.00(-0.12,0.13)	0.00(-0.13,0.13)	1

4.2 Non-stationary model

Based upon the stationary model discussed above, our next step is to develop a flexible *non-stationary* model that can detect time-varying neuronal communities. To this end, we generalize the stationary model presented in the previous section by using a time-varying correlation matrix for the latent variables. More specifically, we allow the partition indicators to change over time, and use z_{it} to denote the corresponding community of the i^{th} neuron

at time t. Therefore, we rewrite the correlation matrix as follows:

$$R_t = \begin{bmatrix} 1 & \rho_{12}\mathbb{1}_{\{z_{1t}=z_{2t}\}} & \cdots & \rho_{1n}\mathbb{1}_{\{z_{1t}=z_{nt}\}} \\ \rho_{12}\mathbb{1}_{\{z_{1t}=z_{2t}\}} & 1 & \cdots & \rho_{2n}\mathbb{1}_{\{z_{2t}=z_{nt}\}} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{1n}\mathbb{1}_{\{z_{1t}=z_{nt}\}} & \rho_{2n}\mathbb{1}_{\{z_{2t}=z_{nt}\}} & \cdots & 1 \end{bmatrix}. \tag{4.2}$$

In prior, we assume that z_{it} follow a dynamic Ewens distribution as follows:

1. At time t=1, $(z_{11}, z_{21}, ..., z_{n1})$ has the standard Ewens distribution.

2. Conditional on the status of neurons at time t, the i^{th} neuron at time $t+1$ stays with the current subset (community) with probability β, moves to an existing subset, M, with probability $(1-\beta)\frac{|M|}{n_{-z_{it}}+\alpha}$, and moves to an empty subset with probability $(1-\beta)\frac{\alpha}{n_{-z_{it}}+\alpha}$. Here, $n_{-z_{it}}$ denotes the number of neurons currently not in the same subset as the i^{th} neuron.

The parameter α control the sparsity of the network, and β controls the dynamics of the network over time. For the experiments described in the following section, we set $\alpha = 0.2$ and $\beta = 0.4$. In general, however, they can be treated as hyperparameters for additionally flexibility.

4.2.1 A simulation study for the dynamic model

Next, we conduct another simulation study where correlation structures among neurons may change over time. In this case we simulate the data according to model by [33]. To this

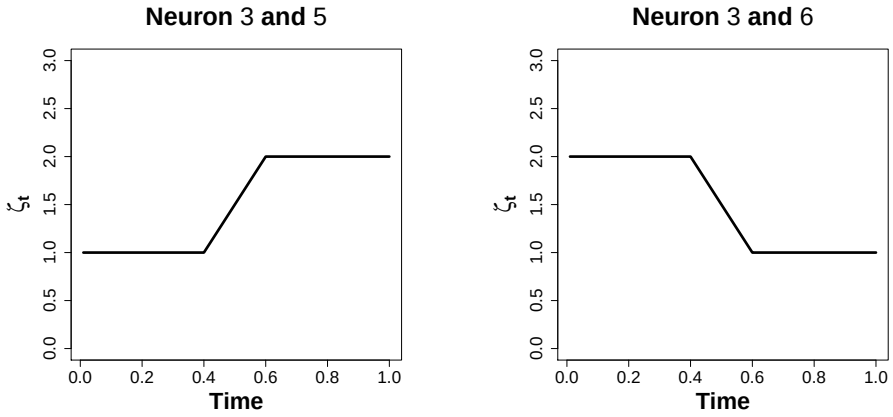


Figure 4.2: Dynamics of the extra term, ζ_t, for the neurons with time-varying correlations. Note that $\zeta_t = 1$ indicates independence $\zeta_t > 1$ indicates positive correlation between each pair of neurons.

end, we first generate time-varying marginal firing probabilities for each neuron and then calculate the joint probabilities of the two neurons by multiplying the product of marginal firing probabilities by an extra term, ζ, such that ζ =1 indicates independence while $\zeta < 1$ and $\zeta > 1$ indicate, respectively, negative correlation and positive correlation between the two neurons. Note that we are using a data generating mechanism that is different from our proposed model in order to test the robustness of our approach. In this simulation study, we have two communities of correlated neurons. For the first community, there are two neurons (indexed by 1 and 2) with constant correlation structure ($\zeta = 2$). For the second community, there are 4 neurons (indexed by 3, 4, 5 and 6), where neuron 3 and 4 have constant correlation ($\zeta = 2$); whereas, the correlations between neuron 3 and neurons 5 and 6 change over time as shown in Figure 4.2. Neurons 7 to 10 are not involved in the network. Figure 4.3 shows the posterior estimates (along with 95% probability intervals) over time for the neurons with strong connections. The results are summarized in Figure 4.4, which shows that our model could correctly identify the underlying dynamic structure of neuronal communities. Note that there is a weak connection between neurons 4 and 6 in the first half and between neurons 4 and 5 in the second half because of the strong connection between

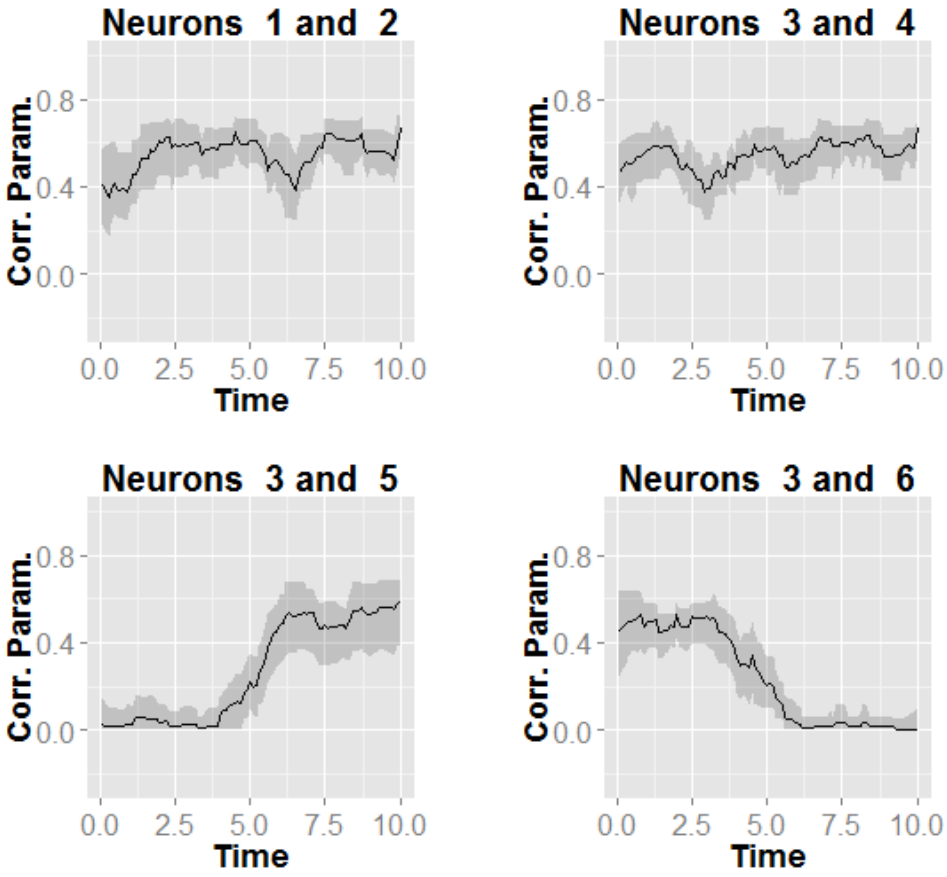
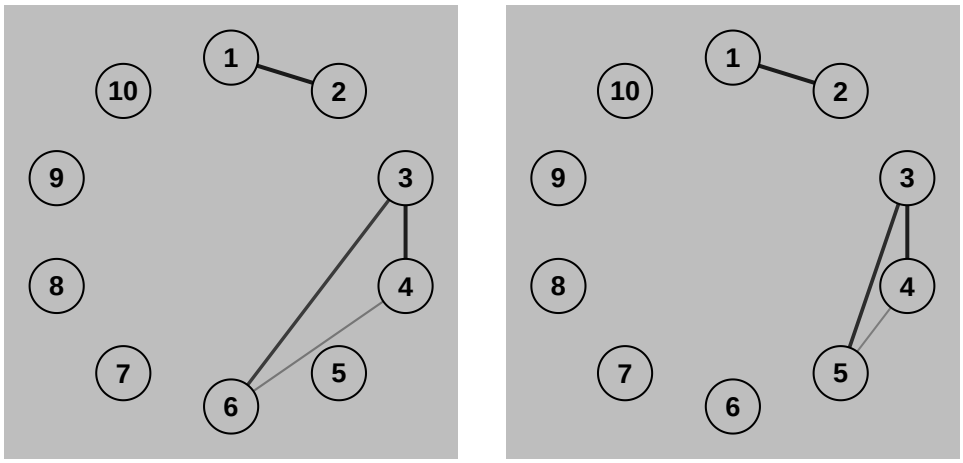


Figure 4.3: The posterior estimates of correlation parameters with 95% credible intervals. The solid lines are the posterior estimates and the gray areas are the 95% credible intervals.

neurons 3 and 4. See the supplementary file for more detailed results.

4.2.2 Results for spike train data

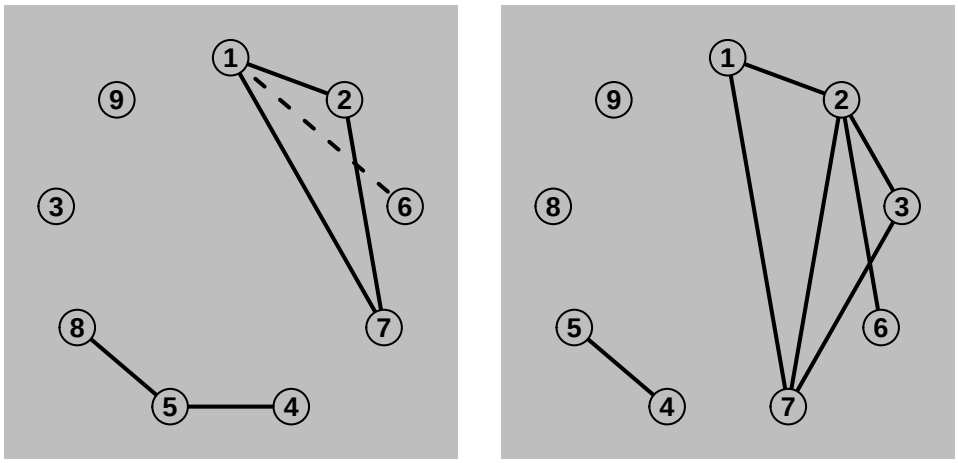
Finally, we apply our method to data collected from the experiment discussed above to examine the community structure of 9 active neurons, which were simultaneously recorded during the experiment under both conditions. Under the rewarded stimulus, we detect two communities of correlated neurons (Panel (a), Figure 4.5). Under the non-rewarded stimulus, we detect two different communities of correlated neurons (Panel (b), Figure 4.5). The



(a) First Half (b) Second Half

Figure 4.4: Time-varying community structure of correlated neurons. For the first half of time, there are two communities (neurons 1, 2 and neurons 3, 4, 6). For the second half of time, neuron 5 joins the second community in stead of neuron 6.

results suggest that neurons 1, 2, 7 and neurons 4, 5 are correlated under both conditions. Neurons 3, 6, and 8, on the other hand, play different roles under different conditions. This indicates that there are specific neural networks in the prefrontal cortex that represent specific variables related to basic reward-based decision-tasks. In both scenarios, while the estimated correlations change over time (see Figure 4.6 and 4.7), the overall structure of networks remains the same throughout the course of experiment.



(a) Rewarded (b) Non-rewarded

Figure 4.5: The community structure of correlated neurons under (a) rewarded and (b) non-rewarded scenarios. Each node represents one neuron and each edge represents the connection between a pair of neurons. Here the solid line represents positive correlation and the dash line represents negative correlation. While the estimated correlations change over time, the overall structure of networks under each scenario remains the same throughout the course of experiment.

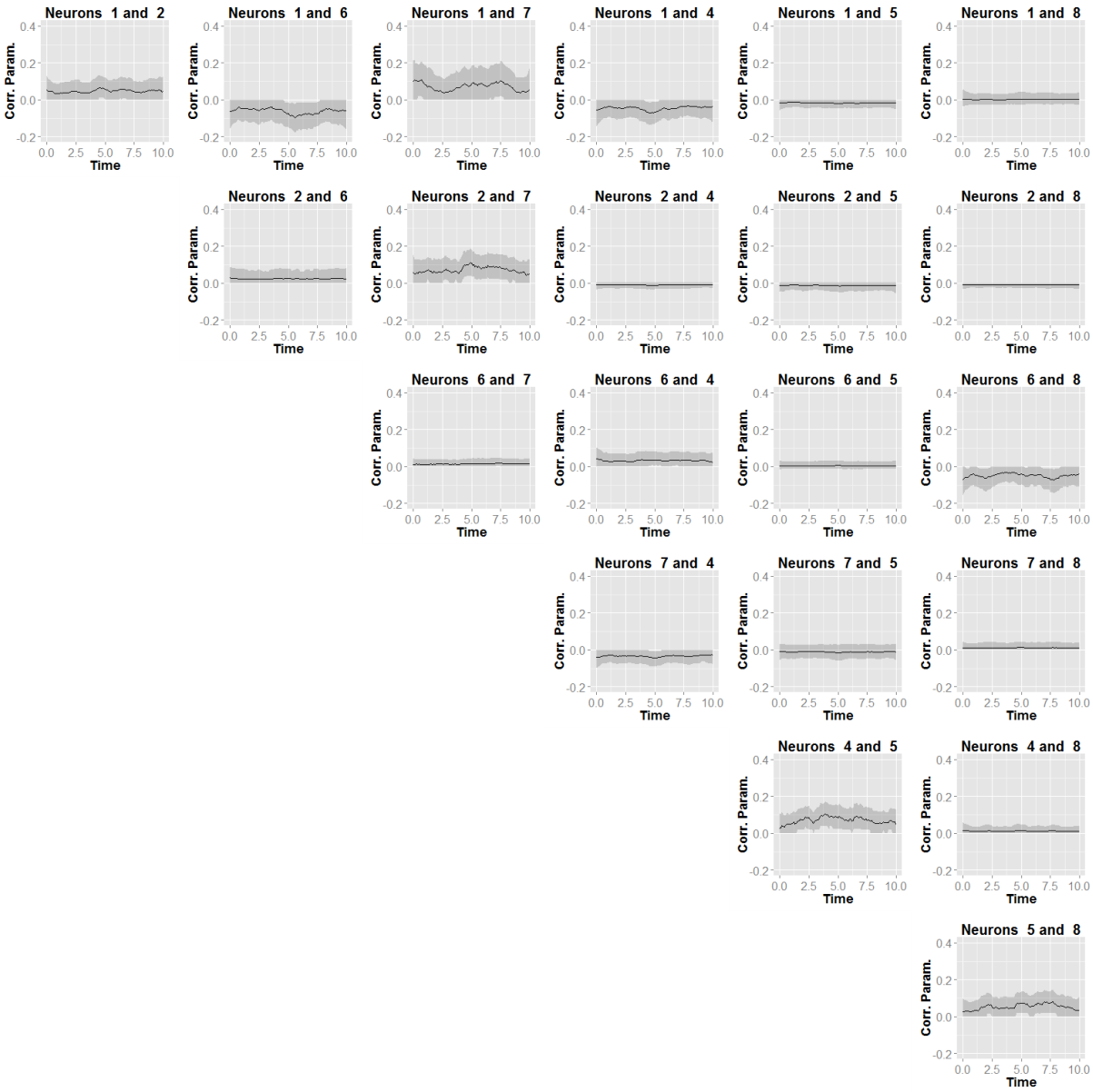


Figure 4.6: **Rewarded:**The posterior estimates of correlation parameters with 95% credible intervals. The solid lines are the posterior estimates and the gray areas are the 95% credible intervals.

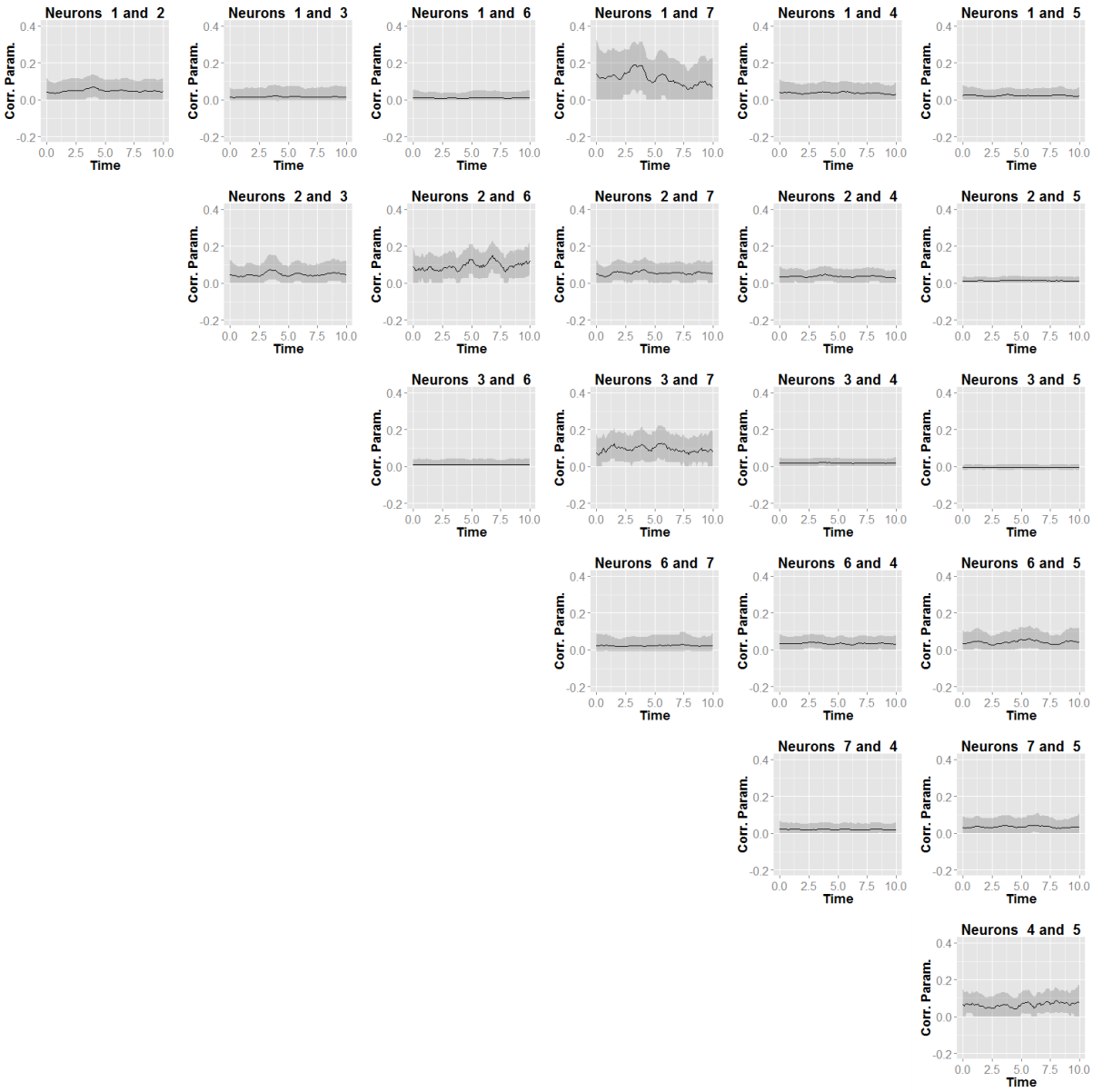


Figure 4.7: **Non-rewarded:**The posterior estimates of correlation parameters with 95% credible intervals. The solid lines are the posterior estimates and the gray areas are the 95% credible intervals.

Chapter 5

Conclusions and Future Directions

5.1 Conclusions

The method we proposed in chapter 2 benefits from multiple aspects, including flexibility, computational efficiency, interpretability, and generalizability. The latter is especially important because the model offered in this work can be adopted for other computationally intensive biological problems. Our proposed method also offers a modeling approach to the problems of identification and calibration of co-firings at various lag times. Consequently, from the statistical inferential perspective, our hierarchical modeling approach would fair better when compared with methods such as cross-correlation analysis, mainly because not only it sheds light on how signals in the network are communicating, but also it informs the scientist of the sharpness of those cross relationships through posterior confidence intervals. Also, note that although it is possible to run other methods, e.g., the method of [30], for different lags and perform multiple hypothesis testing, our method offers a modeling paradigm for measuring lagged and exact synchrony at the same time. This is important because it avoids complex and multistage testing procedures for pairs of neurons. The

sampling algorithm we proposed for detecting synchrony among multiple neurons is also advantageous over other commonly used MCMC techniques such as Metropolis-Hastings. This fact becomes even more salient especially considering that the current technology provides high-dimensional data by allowing the simultaneous recording of hundreds of neurons. The analysis presented offers a number of ways in which examining the temporal relationship of activity among multiple neurons can reveal information about population dynamics of neuronal circuits. These kinds of data are critical in going beyond treating populations as averages of single neurons, as is commonly done in physiological studies. They also allow us to ask the question of whether neuronal firing heterogeneity contributes towards a unified whole [76], or whether separates populations in a brain area are differentially involved in different aspects of behavior [8].

The Bayesian dynamic model we developed in Chapter 3 for analyzing cross-dependence between neurons in a population shows a number of advantages: (1.) It captures the time-dependent dependencies among neurons and thus is applicable for analyzing spike train data recorded while a subject is performing a complex cognitive task; (2.) It allows us to test differences in cross-neuronal correlations in activity under different experimental conditions; (3.) It can be applied to a moderate to large number of neurons, and the computational complexity increases with the number of interacting neurons only; (4.) It can be applied to variety of data from a broad spectrum including continuous-valued time series that have some latent structure; (5.) Finally, the proposed model yielded results that revealed new insight into the dynamic nature of population coding in the prefrontal cortex of rats during decision making.

In addition, the generalized model for proposed in Chapter 4 allows us to implement a more flexible framework for detecting time-varying community structures among neuronal networks. While having the advantages of the models discussed above, this model is even more computationally efficient as computation complexity only increase with community sizes and

hence can handle a larger number of connected neurons. Another advantage of this model is that it allows us to detect multiple subsets or communities of connected neurons during the experiment, rather than only one population of connected neuron discussed above. This is especially important because our proposed approach can capture more complex dependence structures and hence can produce more valuable insights on a variety of cognitive functions including, but not limited to, attention, memory and so forth.

The model produced results that led us to pose new hypotheses related to the prefrontal control of executive function (response execution/inhibition). The large hypothesis resulting from these data is that complex behaviors such as decision-making come about by ensemble coding of discrete behavioral components (stimulus representation, response execution, reward consumption, etc.). These behaviors are encoded by temporally discrete synchronization of specific populations of neurons, although these populations share members. There are a number of key experiments that must be performed to verify this hypothesis and to refine our understanding of encoding along these lines. It is critical to isolate specific subcomponents of behavior that may be encoded. So, for example, studying neural encoding specifically during response execution or specifically during response inhibition (as opposed to in more complex tasks) will allow a precise association between synchronous populations and behavior. In addition, we need to test whether these ensembles reliably form– with the same member neurons– each time a behavior is performed or whether they exhibit subtle differences on a trial-to-trial basis. Further, we need to better understand the temporal aspects of synchronous activation– at what point do synchronous ensembles form and for how long? Finally, we will need to apply the model to larger populations of neurons to determine how sparse population coding in the prefrontal cortex actually is. These are critical experiments and analyses that are feasible with the techniques that we have at hand and which will provide substantially more detail on the precise nature of neuronal ensemble encoding of cognitive function.

The conclusions from this analysis are primary and will need to be studied from further experiments on decision-making in our laboratory and different contexts. Nevertheless, the results of the analysis of even this limited data set are already highly intriguing and already begin to reveal new insight into the dynamic nature of population coding in the prefrontal cortex during basic cognitive tasks. In particular, using these models to characterize how populations of neurons synchronize at precise times across behavior will let investigators focus precisely on the relationship between population coding and discrete components of behavior. For example, it is overly-general (i.e., lacking in temporal precision) to state that neural populations exhibit synchronous activity during reward-seeking. Instead, we can specify that particular neural populations are correlated during specific subcomponents of these behaviors (stimulus presentation, lever approach, reward-well entry, reward consumption, etc.). As demonstrated in our analysis, by narrowing down our window of focus with respect to population encoding, we can achieve a previously unconsidered combination of spatial breadth and temporal precision. Typically the focus on neural coding of discrete behavioral intervals is limited to analysis of single-neuron activity.

5.2 Future Directions

The methods proposed in this dissertation can be generalized in several ways. First, We start our research by modeling the firing rates of one neuron using Gaussian process models. In this project, we consider the nonlinear firing rates as a function of time only. In future, one possibility is to account for history effects by modeling the firing probabilities as a function of time and time difference to the last spike. Next we provide a model framework for jointly modeling of multiple neurons by introducing some latent variables which are assumed to be Gaussian distributed. We might also generalize it to student $t's$ distribution to obtain more robust inference regarding the connection among neurons. The proposed non-stationary

model is already a step forward towards dynamic modeling of neuronal networks. However, our current model assumes that if two neurons are related, the relationship is either positive or negative so that the sign remains constant but may vary in strength over time. To relax this assumption, this model can be extended for situations where the direction of the relationship changes (from positive to negative or vice-versa) over time.

In addition, our current approach only model undirected connectivity among multiple neurons. In the future, it is of great interest to incorporate directed connectivity into multiple neurons. The direction in neural network will help us how information are transmitted in brain when the subjects are performing certain types of cognitive tasks. Lagged directional connectivity among a large population of neurons with be even more appealing as it can produce much more insights about how population of neurons encodes complex cognitive behaviours.

Another direction might be to develop more computationally efficient methods. As the advancement in technology, in future hundreds of neurons or even more can be simultaneously recorded during the experiment. The current method might become infeasible when addressing such a large amount of data. For example, in our proposed method, the most computational intensive part is to sample the latent variables which are truncated normal distribution. Developing a fast and efficient algorithm to sample from high-dimension truncated normal distribution with not only benefit our analysis of spike train data but also be widely appreciated in other fields involving sampling of truncated Gaussian distributions.

Bibliography

[1] R. Barbieri, M. C. Quirk, L. M. Frank, M. A. Wilson, and E. N. Brown. Construction and analysis of non-poisson stimulus-response models of neural spiking activity. *Journal of Neuroscience Methods*, 105:25–37, 2001.

[2] D. Barry and J. A. Hartigan. Product partition models for change point problems. *The Annals of Statistics*, 20(1):260–279, 1992.

[3] K. Benchenane, A. Peyrache, M. Khamassi, P. L. Tierney, Y. Gioanni, F. P. Battaglia, and S. I. Wiener. Coherent theta oscillations and reorganization of spike timing in the hippocampal- prefrontal network upon learning. 66(6):921–936, 2010.

[4] P. Berkes, F. Wood, and J. Pillow. Characterizing neural dependencies with copula models. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 129–136. 2009.

[5] D. R. Brillinger. Maximum likelihood analysis of spike trains of interacting nerve cells. *Biological Cybernetics*, 59:189–200, 1988.

[6] E. N. Brown, R. E. Kass, and P. P. Mitra. Multiple neural spike train data analysis: state-of-the-art and future challenges. *Nature Neuroscience*, 7(5):456–461, 2004.

[7] M. A. Brubaker, M. Salzmann, and R. Urtasun. A family of mcmc methods on implicitly defined manifolds. In N. D. Lawrence and M. A. Girolami, editors, *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics (AISTATS-12)*, volume 22, pages 161–172, 2012.

[8] T. J. Buschman, E. L. Denovellis, C. Diogo, D. Bullock, and E. K. Miller. Synchronous oscillatory neural ensembles for rules in the prefrontal cortex. *Neuron*, 76(4):838–46, 2012.

[9] G. Buzsáki. Neural syntax: Cell assemblies, synapsembles, and readers. *Neuron*, 68(3):362–385, 2010.

[10] J. Chapin. *Populatin-level analysis of multi-single neuron recording data: multivariate statistical methods*. In Methods for Neural Ensemble Recordings, M. Nicolelis Editor, CRC Press: Boca Raton (FL), 1999.

[11] I. Cribben, T. Wager, and M. Lindquist. Detecting functional connectivity change points for single-subject fmri data. *Frontiers in Computational Neuroscience*, 7(143), 2013.

[12] E. M. Crowley. Product partition models for normal means. *Journal of the American Statistical Association*, 92(437):192–198, 1997.

[13] J. P. Cunningham, B. M. Yu, K. V. Shenoy, and M. Sahani. Inferring neural firing rates from spike trains using gaussian processes. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *NIPS*, 2007.

[14] C. O. Diekman, P. S. Sastry, and K. P. Unnikrishnan. Statistical significance of sequential firing patterns in multi-neuronal spike trains. *Journal of neuroscience methods*, 182(2):279–284, 2009.

[15] S. Duane, A. Kennedy, B. J. Pendleton, and D. Roweth. Hybrid monte carlo. *Physics Letters B*, 195(2):216 – 222, 1987.

[16] W. J. Ewens. The sampling theory of selectively neutral alleles. *Theor Popul Biol*, 3(1):87–112, 1972.

[17] D. J. G. Farlie. The performance of some correlation coefficients for a general bivariate distribution. *Biometrika*, 47(3/4), 1960.

[18] K. J. Friston, C. Buechel, G. R. Fink, J. Morris, E. Rolls, and R. J. Dolan. Psychophysiological and modulatory interactions in neuroimaging. *NeuroImage*, 6(3):218–229, 1997.

[19] S. Fujisawa, A. Amarasingham, M. T. Harrison, and G. Buzsáki. Behavior-dependent short-term assembly dynamics in the medial prefrontal cortex. *Nature neuroscience*, 11(7):823–833, 2008.

[20] A. P. Georgopoulos, A. B. Schwartz, and R. E. Kettner. Neuronal population coding of movement direction. *Science*, 233(4771):1416–1419, 1986.

[21] G. L. Gerstein and D. H. Perkel. Simultaneously recorded trains of action potentials: Analysis and functional interpretation. *Science*, 164(3881):828–830, 1969.

[22] J. I. Gold and M. N. Shadlen. The neural basis of decision making. *Annu Rev Neurosci*, 30(535-74), 2007.

[23] S. Grün, M. Diesmann, and A. Aertsen. Unitary events in multiple single-neuron spiking activity: I. detection and significance. *Neural Computation*, 14(1):43–80, 2002.

[24] E. J. Gumbel. Bivariate exponential distributions. *Journal of the American Statistical Association*, 55:698–707, 1960.

[25] M. T. Harrison, A. Amarasingham, and R. E. Kass. Statistical identification of synchronous spiking. In P. D. Lorenzo and J. Victor., editors, *Spike Timing: Mechanisms and Function*. Taylor and Francis, 2013.

[26] J. A. Hartigan. Partition models. *Communication in Statistics - Theory and Method*, 19(8):2745–2756, 1990.

[27] P. W. Holland, K. Laskey, and S. Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 5(2):109–137, 1983.

[28] A. L. Jacobs, G. Fridman, R. M. Douglas, N. M. Alam, P. E. Latham, G. T. Prusky, and S. Nirenberg. Ruling out and ruling in neural codes. *Proceedings of the National Academy of Sciences of the United States of America*, 106(14):5936–5941, 2009.

[29] B. Karrer and M. E. J. Newman. Stochastic blockmodels and community structure in networks. *Phys. Rev. E*, 83(1):016–107, 2011.

[30] R. E. Kass, R. C. Kelly, and W.-L. Loh. Assessment of synchrony in multiple neural spike trains using loglinear point process models. *Annals of Applied Statistics*, 5, 2011.

[31] R. E. Kass and V. Ventura. A spike-train probability model. *Neural Computation*, 13:1713–1720, 2001.

[32] R. E. Kass, V. Ventura, and E. N. Brown. Statistical issues in the analysis of neuronal data. *Journal of Neurophysiology*, 94:8–25, 2005.

[33] R. C. Kelly and R. E. Kass. A framework for evaluating pairwise and multiway synchrony among Stimulus-Driven neurons. *Neural Computation*, pages 1–26, 2012.

[34] A. Kottas and S. Behseta. Bayesian nonparametric modeling for comparison of single-neuron firing intensities. *Biometrics*, pages 277–286, 2010.

[35] A. Kottas, S. Behseta, D. E. Moorman, V. Poynor, and C. R. Olson. Bayesian nonparametric analysis of neuronal intensity rates. *Journal of Neuroscience Methods*, 203(1), 2012.

[36] S. Lan, B. Zhou, and B. Shahbaba. Spherical Hamiltonian Monte Carlo for constrained target distributions. In *Proceedings of the 31th International Conference on Machine Learning (ICML)*, (2014).

[37] M. Lindquist, Y. Xu, M. Nebel, and B. Caffo. Evaluating dynamic bivariate correlations in resting-state fmri: A comparison study and a new approach. *NeuroImage*, in press, 2014.

[38] T. Meyer and C. Olson. Statistical learning of visual transitions in monkey inferotemporal cortex. *Proceedings of the National Academy of Sciences of the United States of America*, 108:19401–19406, 2011.

[39] M. R. Milad and G. J. Quirk. Neurons in medial prefrontal cortex signal memory for fear extinction. *Nature*, 420(6911):70–4+, 2002.

[40] E. Miller and M. Wilson. All my circuits: Using multiple electrodes to understand functioning neural networks. *Neuron*, 60(3):483–488, 2008.

[41] D. Moorman and G. Aston-Jones. Orbitofrontal cortical neurons encode expectation-driven initiation of reward-seeking. *Journal of Neuroscience*, 34(31):10234 – 46, 2014.

[42] D. Morgenstern. Einfache beispiele zweidimensionaler verteilungen. *Mitteilungsblatt für Mathematische Statistik*, 8:234–235, 1956.

[43] G. Motta and H. Ombao. Evolutionary factor analysis of replicated time series. *Biometrics*, 68:825–836, 2012.

[44] P. Müller and F. Quintana. Random partition models with regression on covariates. *J Stat Plan Inference*, 140(10):2801–2808, 2010.

[45] I. Murray, R. P. Adams, and D. J. MacKay. Elliptical slice sampling. *JMLR: W&CP*, 9:541–548, 2010.

[46] J. Nakajima and M. West. Bayesian analysis of latent threshold dynamic models. *Journal of Business and Economic Statistics*, 31(2):151–164, 2013.

[47] J. Nakajima and M. West. Dynamic factor volatility modeling: A bayesian latent threshold approach. *Journal of Financial Econometrics*, 11(1):116–153, 2013.

[48] N. S. Narayanan and M. Laubach. Methods for studying functional interactions among neuronal populations. *Methods in molecular biology*, 489:135–165, 2009.

[49] P. Neal and G. O. Roberts. Optimal scaling for random walk metropolis on spherically constrained target densities. *Methodology and Computing in Applied Probability*, Vol.10(No.2):277–297, June 2008.

[50] P. Neal, G. O. Roberts, and W. K. Yuen. Optimal scaling of random walk metropolis algorithms with discontinuous target densities. *Annals of Applied Probability*, Volume 22(Number 5):1880–1927, 2012.

[51] R. M. Neal. Regression and classification using Gaussian process priors. *Bayesian Statistics*, 6:471–501, 1998.

[52] R. M. Neal. Slice sampling. *Annals of Statistics*, 31(3):705–767, 2003.

[53] R. M. Neal. MCMC using Hamiltonian dynamics. In S. Brooks, A. Gelman, G. Jones, and X. L. Meng, editors, *Handbook of Markov Chain Monte Carlo*, pages 113–162. Chapman and Hall/CRC, 2011.

[54] R. B. Nelsen. *An Introduction to Copulas (Lecture Notes in Statistics)*. Springer, 1 edition, 1998.

[55] R. B. Nelsen. An introduction to copulas. Springer, 2006.

[56] M. Nicolelis, C. Stambach, A. Brisben, and M. Laubach. *Methods for simultaneous multisite neural ensemble recordings in behaving primates*. In Methods for Neural Ensemble Recordings, M. Nicolelis Editor, CRC Press: Boca Raton (FL), 1999.

[57] H. Ombao and S. Van Bellegem. Coherence analysis: A linear filtering point of view. *IEEE Transactions on Signal Processing*, 56:2259–2266, 2008.

[58] H. Ombao, R. von Sachs, and W. Guo. Slex analysis of multivariate non-stationary time series. *Journal of the American Statistical Association*, 100:519–531, 2005.

[59] A. Pakman and L. Paninski. Exact Hamiltonian Monte Carlo for Truncated Multivariate Gaussians. *ArXiv e-prints*, Aug. 2012.

[60] T. Park, I. Eckley, and H. Ombao. Estimating the time-evolving partial coherence between signals via multivariate locally stationary wavelet processes. *IEEE Transactions on Signal Processing*, accepted, 2014.

[61] M. P. Paulus. Decision-Making dysfunctions in psychiatry altered homeostatic processing? *Science*, 318(5850):602–606, 2007.

[62] J. Peters, P. W. Kalivas, and G. J. Quirk. Extinction circuits for fear and addiction overlap in prefrontal cortex. *Learning & Memory*, 16(5):279–288, 2009.

[63] J. W. Pillow, J. Shlens, L. Paninski, A. Sher, A. M. Litke, E. J. Chichilnisky, and E. P. Simoncelli. Spatio-temporal correlations and visual signalling in a complete neuronal population. *Nature*, 454(7207):995–9, 2008.

[64] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2nd edition, 2006.

[65] D. S. Reich, J. D. Victor, and B. W. Knight. The power ratio and the interval map: Spiking models and extracellular recordings. *Journal of Neuroscience*, 18(23):10090–10104, 1998.

[66] F. Rigat, M. de Gunst, and J. van Pelt. Bayesian modeling and analysis of spatio-temporal neuronal networks. *Bayesian Analysis*, pages 733–764, 2006.

[67] L. F. Robinson, T. D. Wager, and M. A. Lindquist. Change point estimation in multi-subject fmri studies. *NeuroImage*, 49(2):1581–1592, 2010.

[68] A. Rodriguez. Modeling the dynamics of social networks using bayesian hierarchical blockmodels. *Statistical Analysis and Data Mining*, 5(3):218–234, 2012.

[69] G. Rodriguez-Yam, R. A. Davis, and L. L. Scharf. Efficient gibbs sampling of truncated multivariate normal with application to constrained linear regression, 2004.

[70] L. Sacerdote, M. Tamborrino, and C. Zucca. Detecting dependencies between spike trains of pairs of neurons through copulas. *Brain Res*, 1434:243–56, 2012.

[71] Y. Sakurai, T. Nakazono, S. Ishino, S. Terada, K. Yamaguchi, and S. Takahashi. Diverse synchrony of firing reflects diverse cell-assembly coding in the prefrontal cortex. *Journal of Physiology-Paris*, 107(6):459–470, 2013.

[72] E. Schneidman, M. J. Berry, R. Segev, and W. Bialek. Weak pairwise correlations imply strongly correlated network states in a neural population. *Nature*, 440(7087):1007–1012, 2006.

[73] M. N. Shadlen and R. Kiani. Decision making as a window on cognition. *Neuron*, 80(3):791–806, 2013.

[74] B. Shahbaba, B. Zhou, H. Ombao, D. Moorman, and S. Behseta. A Semiparametric Bayesian Model for Detecting Synchrony Among Multiple Neurons. *Neural Computation*, (to appear), 2014.

[75] C. Sherlock and G. O. Roberts. Optimal scaling of the random walk metropolis on elliptically symmetric unimodal targets. *Bernoulli*, Vol.15(No.3):774–798, August 2009.

[76] M. G. Stokes, M. Kusunoki, N. Sigala, H. Nili, D. Gaffan, and J. Duncan. Dynamic coding for cognitive control in prefrontal cortex. *Neuron*, 78(2):364–375, 2013.

[77] B. Swihart, B. Caffo, and C. Crainiceanu. A unified approach to modeling multivariate binary data using copulas over partitions. *Johns Hopkins University, Dept. of Biostatistics Working Papers*, page 213, 2010.

[78] V. Ventura, C. Cai, and R. E. Kass. Statistical assessment of time-varying dependency between two neurons. *J Neurophysiol*, 94(4):2940–7, 2005.

[79] M. West. Hierarchical mixture models in neurological transmission analysis. *Journal of the American Statistical Association*, 92:587–606, 2007.

Appendices

Appendix A

Algorithm 1 Sampling latent variables, copula parameters, and hyperparameters

Initialize the matrix of latent variables, $U|_{(n,D)}$, where the i^{th} column corresponds to the latent variables of the i^{th} neuron, n is the number of time bins, and D is the number of neurons.

Initialize the hyperparameters, θ, which specify the Gaussian process priors for the latent variables.

Initialize the copula model parameters, β, as a $D(D-1)/2$ vector.

for $i = 1, \ldots, B$ **do**

  Sample $U^{(i+1)}$ from posterior distribution conditional on $U^{(i)}$, $\theta^{(i)}$ and $\beta^{(i)}$, $P(U^{(i+1)}|Y, U^{(i)}, \theta^{(i)}, \beta^{(i)})$, using the elliptical slice sampler (Algorithm 2).

  for $j = 1, \ldots, D$ **do**

    Sample $\theta_j^{(i+1)}$ from the posterior distribution of the hyperparameters of the j^{th} latent variable conditional on the latent variables, $P(\theta_j^{(i+1)}|U_j^{(i+1)}, \theta_j^{(i)})$, using the slice sampler [52].

  end for

  Sample $\beta^{(i+1)}$ from the posterior distribution conditional on the latent variables, $P(\beta^{(i+1)}|Y, U^{(i+1)}, \beta^{(i)})$, using Spherical HMC presented (Algorithm 3).

end for

Algorithm 2 Elliptical slice sampler for latent variables

Let U be the current state of the latent variables.
Sample $U^* \sim N(0, \Sigma)$, where Σ is the covariance matrix of the Gaussian process.
Calculate the log-likelihood threshold for the elliptical slice sampler,
$v \sim \text{Uniform}[0, 1]$
$\log y \leftarrow \log(L(U)) + \log(v)$
Let α be the angle for the slice.
Draw a proposal and define the corresponding bracket,
$\alpha \sim \text{Uniform}[0, 2\pi]$
$(\alpha_{\min}, \alpha_{\max}) \leftarrow (\alpha - 2\pi, \alpha)$
Set $U' \leftarrow U\cos(\alpha) + U^*\sin(\alpha)$
while $\log(L(U')) < \log y$ **do**
  if $\alpha < 0$ **then**
    $\alpha_{\min} \leftarrow \alpha$
  else
    $\alpha_{\max} \leftarrow \alpha$
  end if
  $\alpha \sim \text{Uniform}[\alpha_{\min}, \alpha_{\max}]$
  $U' \leftarrow U\cos(\alpha) + U^*\sin(\alpha)$
end while
Return U' as the new state.

Algorithm 3 Spherical HMC for copula parameters

Initialize the copula parameters, $\beta^{(1)}$, along with their appropriate transformation, $\tilde{\beta}^{(1)}$, at the current state.
Sample a new velocity value $\tilde{v}^{(1)} \sim \mathcal{N}(0, I_{D+1})$.
Define the potential energy, U, as minus log density of $\tilde{\beta}$ and the kinetic energy, K, as minus log density of $\tilde{v}$.
Set $\tilde{v}^{(1)} \leftarrow \tilde{v}^{(1)} - \tilde{\beta}^{(1)}(\tilde{\beta}^{(1)})^T\tilde{v}^{(1)}$
Calculate the Hamiltonian function: $H(\tilde{\beta}^{(1)}, \tilde{v}^{(1)}) = U(\tilde{\beta}^{(1)}) + K(\tilde{v}^{(1)})$
for $\ell = 1$ to L **do**

$$\tilde{v}^{(\ell+\frac{1}{2})} = \tilde{v}^{(\ell)} - \frac{\epsilon}{2}\left(\begin{bmatrix} I_D \\ 0 \end{bmatrix} - \tilde{\beta}^{(\ell)}(\beta^{(\ell)})^T\right)\nabla U(\beta^{(\ell)})$$

$$\tilde{\beta}^{(\ell+1)} = \tilde{\beta}^{(\ell)}\cos(\|\tilde{v}^{(\ell+\frac{1}{2})}\|\epsilon) + \frac{\tilde{v}^{(\ell+\frac{1}{2})}}{\|\tilde{v}^{(\ell+\frac{1}{2})}\|}\sin(\|\tilde{v}^{(\ell+\frac{1}{2})}\|\epsilon)$$

$$\begin{aligned}\tilde{v}^{(\ell+\frac{1}{2})} \leftarrow & -\tilde{\beta}^{(\ell)}\|\tilde{v}^{(\ell+\frac{1}{2})}\|\sin(\|\tilde{v}^{(\ell+\frac{1}{2})}\|\epsilon) \\ & + \tilde{v}^{(\ell+\frac{1}{2})}\cos(\|\tilde{v}^{(\ell+\frac{1}{2})}\|\epsilon)\end{aligned}$$

$$\tilde{v}^{(\ell+1)} = \tilde{v}^{(\ell+\frac{1}{2})} - \frac{\epsilon}{2}\left(\begin{bmatrix} I_D \\ 0 \end{bmatrix} - \tilde{\beta}^{(\ell+1)}(\beta^{(\ell+1)})^T\right)\nabla U(\beta^{(\ell+1)})$$

end for
Calculate $H(\tilde{\beta}^{(L+1)}, \tilde{v}^{(L+1)}) = U(\tilde{\beta}^{(L+1)}) + K(\tilde{v}^{(L+1)})$
Calculate the acceptance probability

$$\alpha = \min\left(1, \exp\{-H(\tilde{\beta}^{(L+1)}, \tilde{v}^{(L+1)}) + H(\tilde{\beta}^{(1)}, \tilde{v}^{(1)})\}\right)$$

Accept or reject the proposal according to α