Thesis

Fast Decoding of Functional Architecture in Large Neuronal Networks over Small Time Scales

Motolani Olarinre

December 2024
CMU-ML-24-115

Machine Learning Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Statistics and Data Science
Dietrich College of Humanities and Social Sciences
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee

Robert Kass (Chair)
Valerie Ventura
Neil Spencer (UConn)
Gonzalo Mena

Submitted in partial fulfillment of the requirements
for the Degree of Doctor of Philosophy

Keywords: Computational Neuroscience, Statistical inference, Model-based time series clustering, Dynamic time warping, Network models, Uncertainty quantification, Expectation Maximization, Biological neural networks, Machine learning

Abstract

The brain is in large part a complex network of interacting populations of neurons, whose coordinated activity underlies our ability to perform various cognitive and motor tasks. These neuron populations interact through the ensemble effects of sequences of action potentials, referred to as spike trains, from the member neurons. A population may be defined anatomically, based on the physical location of the neurons in the brain, which can be directly observed in experimentally recorded neural data. A population may also be defined functionally, based on the homogenous spiking patterns of the member neurons. These functional populations are typically unobserved in experimentally collected data and must be inferred from the spike train activity of the individual neurons. Regardless of definition, an understanding of the functional connections, that is, the statistical dependencies between these neuron populations, is crucial in understanding how different parts of the brain communicate and function together to perform its various functions. The need for such an understanding necessitates the development of methods to enable this understanding, which forms the motivation for the work done in this thesis.

The primary challenges involved in this objective can be broadly described as follows: 1. Identifying the interacting populations among a set of spiking neurons, and 2: Quantifying the functional connections between these interacting populations. In this thesis, we develop models and methodologies to identify these interacting populations, and to infer the functional connections among them.

In Part I, we develop a Bayesian hierarchical modeling framework to infer the functional connections between interacting populations of neurons using time-dependent features of ensemble neural activity. We apply our model to electrophysiological data recorded from the visual brain areas of multiple subjects, during visual experiments. Our method is able to reveal consistencies in the activity patterns of several brain regions across subjects.

In Part II, we develop a more general modeling framework, which addresses the two primary challenges outlined previously. In this framework, our objective is formulated as a probabilistic graphical model that defines a joint distribution over our observed data. The nodes and edges are unobserved and must be inferred from the observed data. We apply our modeling framework to experimentally recorded data from the visual cortex of a mouse, where the nodes of the graphical model represent the interacting populations, and the edges represent the functional connection between these populations. We discuss the interpretations of our findings on this data.

Finally, we conclude by discussing the future extensions of this work, and how it can be applied in various research domains.

Acknowledgements

I would like to begin by thanking my Amazing mentors over the years: Horacio Rotstein for fostering my interest in mathematics neuroscientific research, as well as Daniel Sword and Brian at Intel Corporation, for helping me navigate my brief sojourn in Industry.

Over the course of my graduate degree, I consider myself fortunate to have been advised by Robert Kass. My journey towards completing a PhD would not have been possible without your guidance and faith in me. I deeply appreciate you pushing me to be the absolute best I could be, even during times when I struggled to see that potential in myself. I am also especially grateful to Neil Spencer, who transitioned from a fellow lab mate in the Neurostats group to the role of a co-advisor over the course of my PhD. From tutoring me in my first year, as I navigated challenging PhD classes, to providing guidance on my final thesis chapter, you have been integral to my success.

I am thankful to my thesis committee members, Valerie Ventura and Gonzalo Mena, for volunteering their valuable time to help me reach this milestone. I would also like to thank my collaborators, especially Yu Chen and Josh Seigle, for their valuable insights when addressing research problems. To my fellow lab mates, Qi Xin and Konrad Urban, thank you for making every lab meeting a learning experience. My experience in the Machine Learning Department at CMU would not have been as enriching without the incredible people I met over the years. Jacob Tyo, Mariya Toneva, Anthony Platanios, Youngseog Chung, Connor Igoe, Ritesh Noothigattu, Robin Schmucker, Renato Negrinho, and many others—I appreciate the long hours we spent figuring out homework problems together, just as much as I value the time we spent playing Super Smash Bros and ping pong in the PhD lounge. Special thanks to Arish Alreja for the constant encouragement and words of wisdom over the years.

I would also like to express my gratitude to the staff of both the Machine Learning and Statistics and Data Science departments. Thank you, Diane Stidle, for your tireless efforts in making MLD an amazing place during our short time there. Similarly, thank you, Danielle Hamilton and Rebecca Nugent, for your contributions to the SDS department.

To the many friends I made around CMU and Pittsburgh—my tribe, my village, and my AfroGSA community (too many to list here)—thank you all for your encouragement and support in countless ways that helped me reach this point.

I could not have dreamed of being where I am today without the love and support of my family. I am deeply grateful to my aunts, uncles, and cousins for their unwavering support over the years. To my siblings, you have both been a source of joy and light, especially during times when things felt bleak. To my parents, you have been a shining example of determination and perseverance. Thank you for instilling in me the will to never give up and for your prayers, which sustained me through this journey. To my grandmother, your faith in me has always been unshakable, and it brings me great joy to make you proud.

Finally, to my amazing partner, you have been my rock and my source of peace through it all—through turbulence and stillness alike. For that, I will always be grateful.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The brain's ability to perform cognitive and motor tasks is driven by the coordinated activity of interconnected regions, which consist of sub-populations of neurons that communicate through sequences of action potentials, known as spike trains. Spike trains encode information about sensory stimuli, motor commands, and cognitive processes, and the temporal patterns of these spike trains are crucial for understanding how the brain processes information [Aljadeff et al., 2016, Reich, 1997]. The ensemble spiking activity of these neurons across multiple populations drives interactions between brain regions [Abbott and Dayan, 1999, Averbeck et al., 2006, Cohen and Kohn, 2011, Nirenberg and Latham, 2003, Shadlen and Newsome, 1998, Zohary et al., 1994], although this population activity is subject to Poisson-like variation, which can easily obscure the functional connections, that is, the statistical dependencies between them, when analyzing data [Chen et al., 2022]. Further considerations include inhomogeneity of the neuron firing patterns within a population, time-varying dynamics, and trial-to-trial variations under the same experimental conditions [Behseta et al., 2009, Cohen and Kohn, 2011, Jia et al., 2020, Lee et al., 2010, Ventura et al., 2005]. We must, therefore, rely on statistical techniques that account for the various sources of variability, in order to uncover these dependencies from neural population spiking data [Kass et al., 2023]. An important statistical tool in modeling a sequence of discrete random events is the point process, a modeling framework that assumes in theory, that at most one random event can occur at any given point in continuous space or time, with the probability of each random event given by a firing rate function [Kass et al., 2014, 2023]. This framework has a long and exhaustive history of applications for modeling spike trains, where it is applied to spiking data recorded in small discrete time bins, such that at most one spike can occur in each bin, resulting in a discrete binary time series [Brillinger, 1988, Brown et al., 2004, Kass and Ventura, 2001, Kass et al., 2005, Pillow et al., 2008, Truccolo et al., 2005]. Concurrently, electrophysiological recording techniques have improved dramatically, including most notably, Neuropixels probes [Jun et al., 2017, Steinmetz et al., 2018, 2021], which can simultaneously record spike trains from hundreds of neurons in multiple cortical regions, to millisecond precision, a scale and resolution not previously possible [Jia et al., 2022, Siegle et al., 2021]. Analysis techniques applied to such vast datasets should run efficiently and scale well with the size of the dataset. This thesis will develop two models to analyze populations of spiking neurons recorded from Neuropixels, with the goal of understanding functional connectivity as defined by fine time-scale timing and coupling relationships between different brain areas. In addition, we should that our models are able to run quickly on large datasets, compared to alternatives, while delivering robust results. We will apply our methods to data from areas of interest in the visual cortex of mice, under varying visual stimulus conditions, during a passive visual task.

The onset of a sensory stimulus in the sensory cortex elicits transient bursts of activity in neural populations, which are presumed to convey information about the stimulus to downstream populations [Manita et al., 2015, Sachidhanandam et al., 2013]. The timing of the synchronized activity peaks is highly variable across interconnected brain regions, and across subpopulations of neurons defined by stimulus-specific neural response profiles, but their relative timing across regions may be less variable, especially for regions that are strongly functionally coupled [Chen et al., 2022, Olarinre et al.]. The standard method for measuring the timing of these peaks is to compute a peri-stimulus time histogram (PSTH), which averages the evoked response across trials, but this ignores trial-to-trial variability in peak times, effectively discarding useful information that might give insights into the propagation of spikes through cortical areas. In Chapter 2, we outline a hierarchical Bayesian modeling framework able to obtain precise estimates of population burst times on a trial-by-trial basis, and reveal correlations in the timing of evoked population bursts across visual areas following the onset of stimulus, by accounting for various sources of variability. This framework is simple enough to be used by most practitioners and easily scales to large amounts of data while maintaining a reasonable runtime. Using our approach, we examine the relative

timing of population bursts in large-scale recordings of spiking activity from six cortical visual areas and one visual thalamic nucleus in thirteen experimental subjects, to identify variations in peak times and region-to-region coupling relationships. While this framework produces good results, it and previous work in this direction, require pre-screening steps to account for the observed diversity in stimulus-dependent population response profiles, a consequence of inhomogeneity in neuron responses both within and across stimulus conditions. Pre-screening is thus aimed at filtering out the stimulus-dependent neuron populations that are deemed "not relevant" to functional coupling across brain regions. This heavily depends on a definition of a functionally relevant population as having a dual peaked response to a given stimulus. However, we often observe populations with both single-peaked and dual-peaked responses, all of which may be relevant to functional connectivity. The filtering done by the previous methods tends to eliminate large populations of potentially relevant neurons by applying a very narrow definition of a functionally relevant population. In Chapter 3, we develop a broad modeling framework for inferring functional connectivity that addresses this limitation by modeling all distinct neuron populations using a Probabilistic graphical model, where the various nodes represent the diversity of neuron populations, and the edges represent the interactions between the populations. Using this framework, we are able to model functional connectivity between multiple populations more broadly. Using synthetic data, we demonstrate the ability of our framework to perform model selection by automatically learning the true number of neuron populations present in a large dataset, as well as the interactions between the populations. This eliminates the need to pre-define a functionally relevant population, and thus to prescreen for them, as the model accounts for these automatically.

Part 1

3-Step Method for Identifying Functional Connections

Chapter 2

Relative timing and coupling of neural population bursts in large-scale recordings from multiple neuron populations

This section is based on the preprint [Olarinre et al.], which is currently under revision for the Journal of Neuroscience. It builds on earlier work [Chen et al., 2022]

This work focuses on our efforts to design a simple and scalable method for probabilistic modeling for neuron population spiking data and to determine the strength of the functional association between different visual areas in the visual cortex of mice. The conventional tool for comparing neural activity across brain regions involves aggregating data across trials, producing a Peri-Stimulus Time Histogram (PSTH). These, however, obscure any information about trial-to-trial variability/covariability, which is a useful proxy for functional association [Averbeck et al., 2006, Ben-Shaul et al., 2001, Bondy et al., 2018, Brody, 1999, Cohen and Kohn, 2011, Lee et al., 2016, Ventura et al., 2005]. Typical attempts at quantifying these trial-to-trial variations make use of spike counts in relatively wide time windows of the population PSTH (where the data is aggregated across neurons in a population rather than across trials). This, however, also obscures the functional association at fast time scales [Averbeck et al., 2006, Bondy et al., 2018, Cohen and Kohn, 2011, Gu et al., 2011, Smith and Kohn, 2008, Smith and Sommer, 2013, Vinci et al., 2016]. Here, we apply a method to the PSTHs to identify these fast timescale variations. We obtain strong results on the correlation in the times at which spiking neural populations reach their peak activity, a feature of the population firing rate profile. We also obtain the order in which these peaks are attained for different areas. In [Chen et al., 2022], we conduct this analysis on 3 visual areas in a single mouse, with repetition in a second mouse, and in [Olarinre et al.], we extend this analysis to 7 areas in 13 mice, enabling conclusions about the degree to which different connectivity patterns vary across mice. In addition, we both simplified the procedure and greatly decreased the computation time.

2.1 Background and motivation

Within tens of milliseconds after the onset of a sensory stimulus, spikes are conveyed from the periphery and evoke a transient burst of activity across large populations of neurons in the thalamus and cortex. Responses to stimulus onset in sensory cortex often include two activity peaks, with the earlier peak reflecting the feed-forward propagation of spikes from the periphery, while the second peak (occurring 100-200 ms later) likely reflects feedback from other cortical areas [Manita et al., 2015, Sachidhanandam et al., 2013]. The standard method for measuring the timing of these peaks is to compute a peri-stimulus time histogram (PSTH), which averages the evoked response across trials, but this ignores trial-to-trial variability in peak times, effectively discarding useful information that might give insights into the propagation of spikes through cortical areas [Chen et al., 2022].

Modern electrophysiological recording techniques, such as Neuropixels probes [Jun et al., 2017, Steinmetz et al., 2021], have enabled simultaneous recordings of spike trains from hundreds of neurons in multiple cortical regions, making it possible to observe the timing of evoked responses in greater detail than was previously possible [Jia et al., 2022, Siegle et al., 2021]. The Allen Brain Observatory Neuropixels Visual Coding Dataset [Allen Institute MindScope Program, 2019], an open dataset consisting of electrophysiological recordings from multiple cortical and thalamic visual areas in parallel, is a prime example of what can achieved with these probes (Figure

2.1A-C). The cortical areas recorded in this dataset display a stereotypical dual-peaked response to the onset of a full-field drifting grating stimulus, with the average timing of the first peaks consistent with their relative hierarchical ordering determined by anatomy (Figure 2.1D) [D'Souza et al., 2022, Harris et al., 2019, Siegle et al., 2021]. As each peak results from the synchronous firing of many neurons in a given region, we refer to these peaks as "population bursts." As noted by [Kass et al., 2023], assuming that behaviorally relevant information is transmitted across parts of the brain through such transient bursts of activity in neural populations, their timing on a trial-by-trial basis should reveal coordinated activity.

We followed [Chen et al., 2022] by focusing on the time at which each population firing rate reaches its maximum value ("peak timing") because, by definition, many spikes occur around the time of the maximal value, so it should be well determined, statistically. Chen et al. [Chen et al., 2022] noticed that the simple ("näive") method of determining peak timing, namely smoothing the population PSTH and finding the time of its maximal value, in fact, produced poor estimates, and they identified three sources of difficulty. First, for any given condition, only a subset of neurons responded similarly to produce the two-peak population profile, while the other neurons effectively diluted the signal by issuing irrelevant noisy spike times. Second, the shape of the peak was condition-specific; the usual population PSTH, as shown in Figure 2.1, is a blurred average across conditions. Third, the Poisson-like noise in the spike trains (which is typical in many brain areas of behaving mammals), from a limited number of condition-relevant neurons, contributed substantially to the inaccuracy in peak timing estimates derived from smoothed population PSTHs. We devised an analytical strategy to overcome these problems, and the new method is much simpler and more computationally efficient than that proposed by Chen *et al.*.

Not only do excessively noisy estimates of timing make it difficult to establish sequential activity across areas, they can also greatly decrease correlations of the timing across two areas, such as V1 and LM. This phenomenon is well known and easy to prove mathematically [Kass et al., 2014, Section 12.4.4]. It is also intuitive: if two measurements tend to move up and down together but independent noise is added to them, the extent to which they move together will be thrown off by the noise, and their correlation will thus be diminished. In the statistics literature, an improved correlation estimate (often discussed under the heading of "errors in variables"), is typically called a "correction for attenuation" [Kass et al., 2014, Section 12,4.4 and references therein]. To be clear, our corrections for attenuation aim to do a better job of estimating the results that would have been obtained had the entire condition-relevant population of neurons been recorded. Figure 2.6 provides an illustration for areas V1 and LM, based on the method we describe here.

Chen *et al.* solved the three problems listed above by developing a comprehensive Bayesian hierarchical model, called the Interacting Population Rate Function (IPFR) model. Simulation studies showed their method could obtain accurate estimates of individual trial population burst times and their trial-to-trial correlations across areas. Because it included, together, all elements of the problem, the IPFR model was rather complicated, and for large data sets could take an excessively long time to run in standard computing environments. As an alternative, we developed a simplified version by solving each of the three problems, separately, in a 3-step procedure. We demonstrate that the new procedure can replicate, with good accuracy, the results of the previous method while having an 85 to 90% reduction in compute time. We then use the new procedure to examine the relative timing and coupling of population bursts across thirteen mice and to infer the mouse-to-mouse variation in these timing and coupling relationships.

Materials and Methods

Experimental Setup

In this work, we analyzed the publicly available Allen Brain Observatory Visual Coding Neuropixels Dataset [Allen Institute MindScope Program, 2019], which includes spike trains and local field potential recordings from the mouse visual system. In each experiment, six Neuropixels probes were targeted to six areas of visual cortex (Figure 2.1A-C), which were identified via functional retinotopic mapping before the experiment. Spike trains from between 40 and 100 neurons from each area were recorded simultaneously from each subject (after applying standard thresholds to spiking sorting quality metrics, see [Siegle et al., 2021] for details). Thirty mice were head-fixed and passively presented with visual stimuli, which included natural movies, full-field flashes, Gabor patches, and drifting gratings. Here, we focused on drifting gratings because they included many repeated trials for each condition, the trials are relatively long (3 s each), and they drive strong responses in visual cortex. The drifting gratings have 40 conditions that result from combining eight grating orientations (0°, 45°, 90°, 135°,180°, 225°, 270°, 315°) and five temporal frequencies (1, 2, 4, 8, 15 Hz). Each condition is repeated 15 times. Each trial lasts for 3 s, with 2 s stimulus and 1 s blank screen, with all conditions randomly interleaved. We analyzed spike trains from the lateral geniculate nucleus (LGN), the thalamic region that receives inputs from the retina and sends outputs to cortex, and six cortical areas: primary visual cortex (V1), which is the primary target of LGN, and

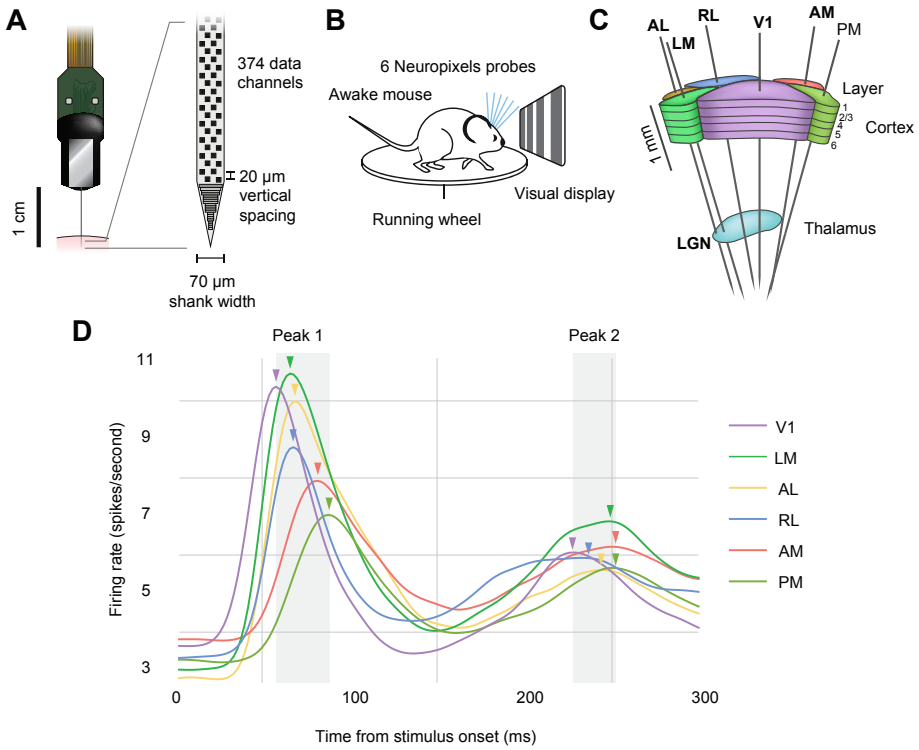

Figure 2.1: **Electrophysiological recordings from seven visual areas in a publicly available dataset. A**, Illustration of a Neuropixels probe used to detect extracellular spiking activity across hundreds of neurons in parallel. **B**, Schematic of the recording configuration. Mice are head-fixed and free to run on a spinning wheel, while passively exposed to visual stimuli. Six Neuropixels probes are targeted to the visual cortex. **C**, In each recording session, probes pass through six visual cortical regions (AL, anterolateral visual area; LM, lateromedial visual area; RL, rostrolateral visual area; V1, primary visual cortex; AM, anteromedial visual area; PM, posteromedial visual area) and one thalamic visual region (LGN, lateral geniculate nucleus). **D**, Overall population response to the onset of a drifting grating stimulus. The population response here is obtained by smoothing PSTHs across neurons and trials for each area. Note the two prominent peaks, which likely result from feedforward and feedback signal propagation, respectively. Arrows indicate the time at (and thus order in) which the firing rate in each area's population reaches its maximal value.

is at the bottom of the visual hierarchy, as well as the rostrolateral (RL); lateromedial (LM); anterolateral (AL); anteromedial (AM); and posteromedial (PM) visual areas, with the last two residing at the top of the anatomically defined visual hierarchy [D'Souza et al., 2022, Harris et al., 2019].

Model Overview and statistical analysis

A high-level sketch of the IPFR model for a single area, under a single stimulus condition, is shown in the left diagrams of Figures 2.2 and 2.3, and details can be found in [Chen et al., 2022]. The three steps of our new procedure correspond to the three problems identified in the introduction. We label these steps (1) interacting population selection, (2) initial peak time and standard error estimation, and (3) peak time denoising using and trial-to-trial correlation. Schematic summaries of this procedure are shown in Figures 2.2 and 2.3.

In step (1) we extracted the subset of the population in each area that responds to a particular condition, which we call the interacting population. As we were interested in the time from stimulus onset at which the intensity function reaches its peak, we filtered out neurons that showed no change in firing rate, or a decrease in firing rate, in response to the stimulus. Then we selected, for each condition, the neurons with a clear peak in the stimulus response profile. We accomplished this selection by fitting a firing rate function to the PSTH for each recorded neuron, and for each stimulus condition, across trials. A neuron's condition-specific PSTH was obtained by first binning the spike train in each trial into 1 ms bins, and then summing these binned spikes trains across all the condition's trials. The firing rate function is modeled non-parametrically using a Poisson Generalized Additive Model (GAM) with a spline basis (Figure 2.4), which is fit to the neuron's PSTH using maximum likelihood [Kass et al., 2014, Chapter 19].

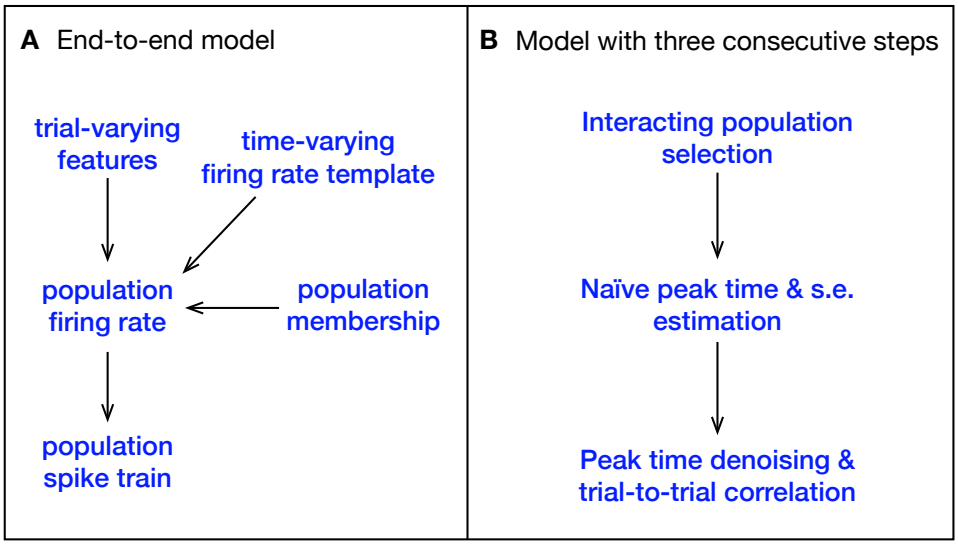

Figure 2.2: **Comparison between the IPFR model and our three stage model for a single stimulus condition.** **A**, The IPFR model. The population spike train on a single trial is driven by its population firing rate, which combines a time-varying firing-rate template with trial-varying features. Only a subset of neurons recorded within the brain area will be used, and this subpopulation is determined by a population membership probability. This is all captured by a single model, with all variables and parameters jointly inferred. **B**, In our model, the estimation procedure is divided into three sequential stages.

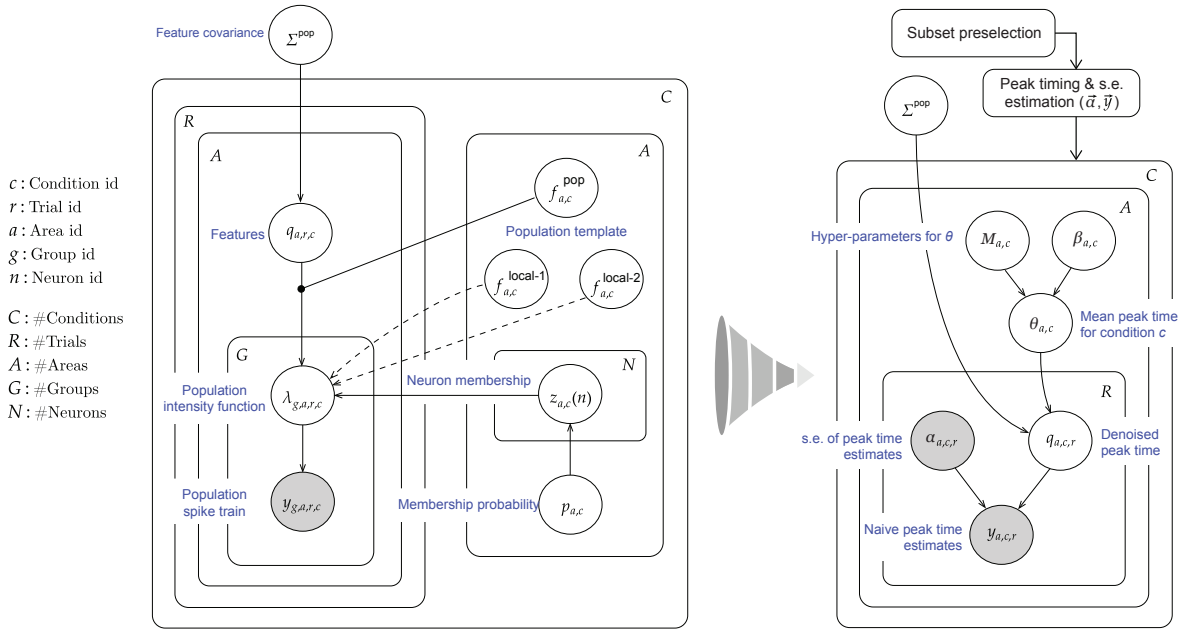

Figure 2.3: **Plate diagram of our simplified multi-step procedure for estimating the timing of population bursts.** Left: Original IPFR model from [Chen et al., 2022]. Right: Simplified model, which divides the estimation task into 3 steps. We are able to obtain comparable results with substantially reduced computation time.

We observed strong orientation tuning in the response profiles of individual neurons, consistent with previous recordings from visual cortex. Examples are shown for three neurons in Figure 2.4. Exploratory analysis revealed that the initial population burst occurs between 30 ms and 160 ms from stimulus onset, and so filtering for the evoked responses typical of an interacting population is done within this window. The filtering criteria for each neuron's firing rate function within the burst window are as follows:

(i) average firing rate is in the top 60% among neurons in the same visual area;

(ii) must have a concave critical point;

(iii) must have maximum slope in the top 60% among neurons in the same visual area; and

(iv) the increase from baseline to peak firing rate in the top 60% among neurons in the same visual area.

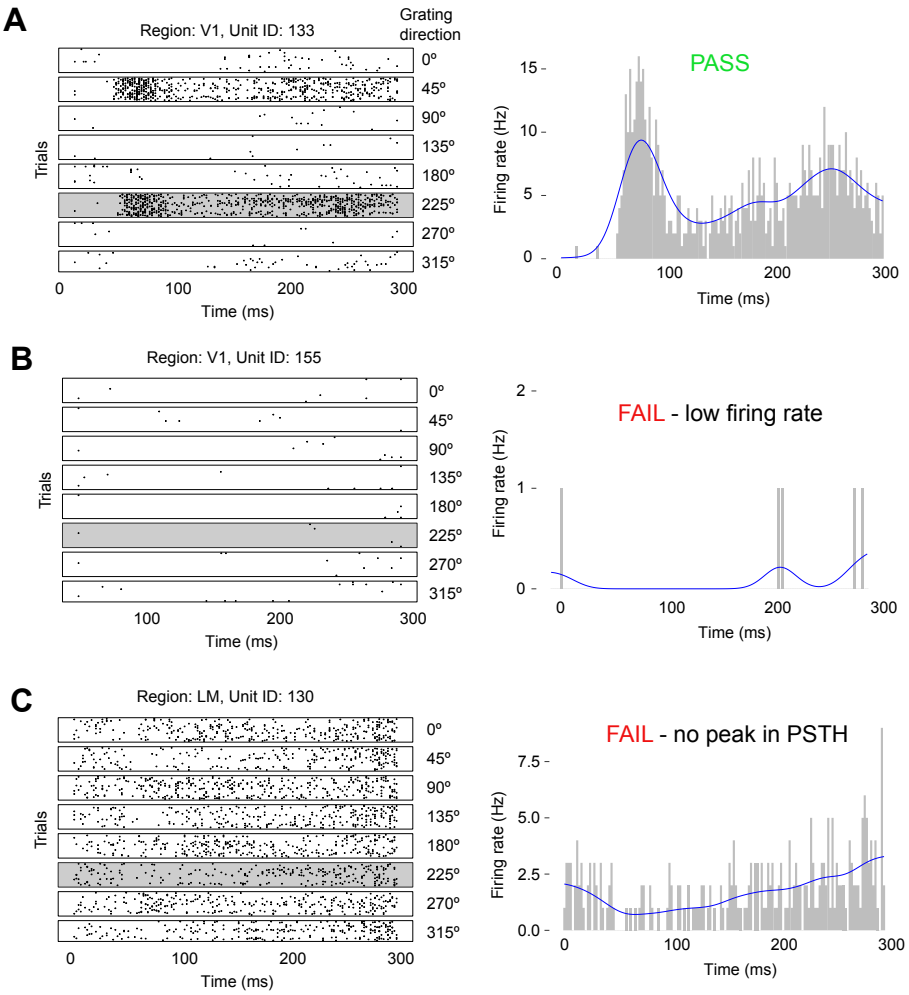

Figure 2.4: **Selection criteria illustrated for three example neurons.** Left: Spike rasters of three different neurons to eight directions of the 1 Hz drifting grating stimulus. Right: PSTH and fitted firing rate function for the 225 degree stimulus condition. **A**, The neuron passes the selection criteria due to its high firing rate and peak in its stimulus response profile. **B**, The neuron fails the selection criteria due to its low firing rate. **C**, This neuron fails because its PSTH lacks a peak (defined as a concave critical point).

These filtering criteria can be easily automated and applied to all mice. Intuitively, they select the neurons within each visual area with a strong, peaked response to a stimulus. The conditions for a strong peaked response were determined from exploratory analysis on a single, randomly selected mouse, to be a high average firing rate (across time) in the peak response time window, a concave peak, and a sharp and noticeable increase in firing rate from its baseline in the peak response window. The presence of these features together is a strong indicator of a peaked response, although the concavity and the increase from baseline matter to a greater extent than the absolute firing rate. The thresholds were determined empirically for each peak separately from an exploratory analysis of data from one mouse (ID 756029989) and validated on a second mouse (ID 760345702) before extending the analysis to the full Allen dataset.

After filtering, we rejected data from any condition in each of the 7 visual areas with an interacting population of less than 10 neurons available for the next stage of the analysis. Following the subset pre-selection step, the resulting data set for each visual area consists of only those neurons that contribute to population activity given each stimulus condition.

In step (2), the population PSTH on a given trial for a particular area is obtained by summing binned spikes trains across neurons. The population firing rate function is estimated in the same manner as with individual neurons (using a Poisson Generalized Additive Model with a spline basis), and the time of maximal firing rate, which is a naive estimate of the "peak time", is obtained from the population firing rate function as the time at which the maximum of this function occurs; its estimation uncertainty is obtained by bootstrap resampling from the population of neurons. As an object of statistical estimation, the peak burst time has the advantage of having, by definition, a relatively large number of spikes occurring near that time.

In step (3), the naive peak times and uncertainties obtained in step (2), which are represented as $y_{c,r}$ and $\alpha^2_{c,r}$ respectively, are inserted into a simple Bayesian hierarchical model, shown visually in figure 2.3 and specified as follows, for trial r under condition c:

$$
\begin{aligned}
y_{c,r}|q_{c,r} &\sim \mathcal{N}(q_{c,r},\ diag(\alpha^2_{c,r})) \\
q_{c,r}|\theta_c,\ diag(\sigma_c),\ \mathbf{R} &\sim \mathcal{N}(\theta_c,\ diag(\sigma_c) * \mathbf{R} * diag(\sigma_c)) \\
\theta_c &\sim \mathcal{N}(M_c,\ diag(\beta^2_c)) \\
[\sigma_c]_i &\sim \text{Half-Cauchy}(1) \\
\mathbf{R} &\sim \text{LKJ}(1)
\end{aligned}
$$

The components of the vectors correspond to the visual areas, indexed by a in figure 2.3. Here, $y_{c,r}$ is a vector of peak time estimates (found in step (2)), which are assumed independent and identically distributed given $q_{c,r}$. The components of the vector $\alpha^2_{c,r}$ (again found in step (2)) are squared standard errors of the peak time estimates. The components of the vector $q_{c,r}$ are denoised peak times (expected values of the components of $y_{c,r}$) while θ_c are mean denoised peak times (the expected values across trials) and σ_c are the corresponding standard deviations, for condition c.

To complete the hierarchical model we adopt prior probability distributions that are commonly used because of their good statistical behavior. We place a half-Cauchy distribution on the standard deviation, with scale parameter 1 [Gelman, 2006, Polson and Scott, 2012]. The symbol $\mathbf{R}$ denotes the matrix of cross-area correlations in the peak-1 times. For its prior we use a probability distribution over the space of (flattened) vectors of product-moment correlations that composes $\mathbf{R}$. This is called the Lewandowski-Kurowicka-Joe (LKJ) distribution [Lewandowski et al., 2009], and is the recommended prior distribution over correlation matrices in popular Bayesian inference software STAN [Stan Development Team, 2024]. This distribution corresponds to one over the manifold of d dimensional positive definite symmetric matrices with unit diagonals and off diagonals between -1 and 1. Note that this manifold has a complex, non-Euclidean geometry because of the constraints imposed by symmetry, unit diagonal, and positive definiteness. A sampling algorithm proposed in [Lewandowski et al., 2009] can generate samples from this manifold, and therefore defines a probability density function over the manifold. The density for a correlation matrix sampled according to this algorithm is proportional to the determinant of the matrix raised to a certain power, which is defined by a free parameter. When this power is 0, the probability density corresponds to a uniform distribution over this manifold. The matrix Σ^{pop}_c in the right-hand side of Figure 2.3 can be expressed as $diag(\sigma_c) * \mathbf{R} * diag(\sigma_c)$, and is the cross area covariance matrix for the peak times $q_{c,r}$. The symbols M_c and β_c are the mean and variance hyper-parameters for the prior on θ_c. We estimate both using maximum likelihood, in a similar way as $y_{c,r}$ and $\alpha_{c,r}$, by summing binned spike trains across both neurons and trials corresponding to stimulus condition c, for each area. This closely approximates fully Bayesian posterior inference, as described in the conditionally independent hierarchical model (CIHM) framework [**?**], also referred to as parametric empirical Bayes (PEB). We use the Rstan package with Hamiltonian Monte Carlo [Stan Development Team, 2024] to obtain posterior samples and then posterior means and variances for $q_{c,r}$, θ_c, σ_c and $\mathbf{R}, r = 1, ..., R, c = 1, ..., C$. In addition, we use the posterior mean peak times $\hat{\theta}_{a,c}$ and posterior variances $\hat{\delta}^2_{a,c}$ to compute, for each area a, estimates of the mean peak time across conditions, $\bar{\theta}_a$. We do this using a weighted mean as the differing degrees of variances in the peak time estimates across areas makes the simple arithmetic mean a higher variance estimator. We do not use the usual formula for a weighted mean (e.g., page 193 of [Kass et al., 2014]) because $\bar{\theta}_a$ involves two sources of variance, the posterior variances $\hat{\delta}^2_{a,c}$ of the condition-dependent estimates and the variance of those estimates across conditions. Thus, the formula we need appears in equation (16.37) on page 461 of [Kass et al., 2014]. We use the following simple iterative algorithm to estimate the appropriate weighted mean and its variance (the maximum likelihood estimate of these two quantities) by alternating between the two, while conditioning on the current value of the other:

- Initialize

$$w_{c,0} = \frac{1}{C}$$

- For k = 1 till convergence, repeat

$$\bar{\theta}_{a,k} = \sum_{c=1}^{C} w_{c,k-1}\ \hat{\theta}_{a,c}$$

$$sd^2_{\bar{\theta}_a,k} = \sum_{c=1}^{C} w_{c,k-1}\ (\hat{\theta}_{a,c} - \bar{\theta}_{a,k})^2$$

$$w_{c,k} = \frac{\left(\hat{\delta}^2_{a,c} + sd^2_{\bar{\theta}_a,k}\right)^{-1}}{\sum_{c=1}^{C}\left(\hat{\delta}^2_{a,c} + sd^2_{\bar{\theta}_a,k}\right)^{-1}}$$

  where $sd^2_{\bar{\theta}_a}$ is an estimate of the variance in the peak times of area a, across stimulus conditions $c = 1, \ldots, C$, around their weighted mean $\bar{\theta}_a$.

We compute the squared standard error of the weighted mean, $se^2_{\bar{\theta}_a}$ as

$$se^2_{\bar{\theta}_a} = \frac{1}{\sum_{c=1}^{C}\left(\hat{\delta}^2_{a,c} + sd^2_{\bar{\theta}_a}\right)^{-1}}.$$

We follow these steps for each mouse to get the weighted means and variances, for all areas. Based on these, for each area, we then aggregate across all mice using the ordinary weighted mean and its standard error; we also compute the standard deviation across mice. We do this analysis for the first and second peaks separately.

Data and code availability

The code for our model is available on GitHub. The data from the Allen Brain Observatory Neuropixels Visual Coding dataset can be accessed via the AllenSDK, the DANDI Archive, and through AWS Registry of Open Data.

Results

In order to establish the estimation accuracy of our model, we first conducted simulations using data generated by the IPFR model. We demonstrate that our three-step method is nearly as accurate as the IPFR model while having greatly reduced computation time.

We then apply our method to data from the Allen Brain Observatory Neuropixels Visual Coding dataset, with the goal of examining mouse-to-mouse variation in relative lead–lag timing and coupling (trial-to-trial correlation) relationships among the different visual areas based on peak 1 and peak 2 timing. Previous studies have shown a temporal ordering in the feedforward propagation of spikes through the visual cortex, with evoked spikes appearing in higher visual areas having longer delays after stimulus onset [Glickfeld and Olsen, 2017, Schmolesky et al., 1998, Siegle et al., 2021]. We expected such an ordering to be reflected in the ordering of peak 1 times across the different areas. Furthermore, any ordering that is functionally relevant should be consistent across subjects, even though the absolute peak times may be subject to a variety of sources of variation having little or no functional relevance. Because feedback propagation has been less well studied, it is unclear what to have expected about the ordering of peak 2 population activity, as peak 2 timing most likely depends on top-down signals coming from other brain regions, as well as the animal's internal state.

In addition to the relative ordering in the peak times across the visual areas, our methods can be used to learn about functional associations between visual areas through marginal and partial trial-to-trial correlations in peak times between pairs of areas. For example, neuroanatomical studies of the mouse thalamocortical pathway have shown there are almost no direct anatomical projections from LGN to higher visual areas [Antonini et al., 1999], which implies communication between the LGN and the higher-order visual areas is mediated through V1. We would therefore expect any marginal correlations between LGN and higher-order areas to be reduced substantially after accounting for the activity of V1, using partial correlation. We report results using these tools.

Performance in estimating ground truth values using simulated data

To compare our proposed model to the IPFR model, we conducted a simulation study using the latter model as the ground truth. Our simulation consisted of two hypothetical brain regions a_1 and a_2, with the number of neurons in each area given by N_1 and N_2 areas, respectively. We specified a stimulus condition s, with proportions p_{a_1} and p_{a_2} of the neurons in the corresponding region belonging to the single peaked response population for s, and the complementary sets of neurons in each region belonging to the flat response population. Given these conditions, as well as the pre-defined peaked and flat response population firing rate templates for each region, we sampled the individual neuron spike trains using a Poisson point process. For a given area, each neuron spike train depends on the neuron's population membership (which follows a categorical distribution), and corresponding population firing rate template. In addition, each trial of the peaked response population firing activity depends on a trial-varying peak time, which follows a bivariate Gaussian distribution (corresponding to each the two regions), with a pre-specified mean μ, variances σ_1 and σ_2 and correlation ρ. We incorporated the trial peak time into the peaked response population firing rate function using time warping. Details of the generative model can be found in [Chen et al., 2022]. We sampled the neuron spike trains in both areas for R trials, and then fit the IPFR model and our three-step model to the resulting data. We also estimated peak times from a Gaussian kernel applied to the population PSTH (we referred to this previously as a naive estimate of the peak times), with bandwidth selected using cross-validation. We repeated this process of simulating and fitting data across 60 repititions, for each of several configurations of ρ, R and lag $(= \mu_2 - \mu_1)$. We fixed N_1, N_2, σ_1, σ_2, p_{a_1}, p_{a_2} at the values 100, 100, 1, 1, 0.8, 0.8, respectively. For each repetition we estimated the parameters of the Gaussian distribution of peak times using each method (for the näive method we estimated correlations from the Pearson correlations of the näive peak time estimates), and then computed the mean and standard error across repetitions. Figure 2.5 shows the fitted firing rate function for each candidate model to area a_1's population PSTH in a single trial of an example simulated dataset, shown for visual comparison.

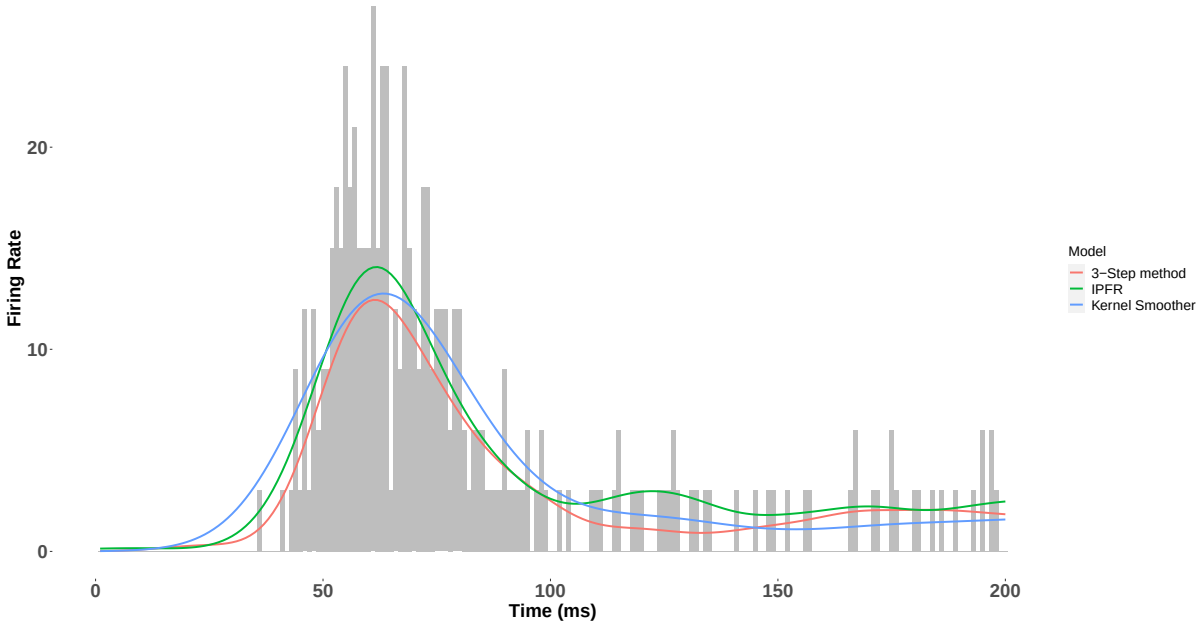

Figure 2.5: **Fitted firing rate function for each candidate model to the population PSTH in a single trial of an example simulated dataset.** For one of the datasets with correlation $\rho = 0.8$ and number of neurons $N_1 = 100$, we show the fitted firing rate function on a single trial, using each method described above. The IPFR and the 3-step method both use a GAM with a log link function to fit the intensity function. However, the 3-step method first filters out those neurons that do not participate in the population burst response, as detailed in the Model Overview and Statistical Analysis section above. Note that the IPFR (green trace) was used as the ground truth to generate the datasets.

The IPFR and the 3-step method both use a GAM with a log link function to fit the intensity function. However, the 3-step method first filters out those neurons that do not participate in the population burst response, as detailed in the Model Overview and Statistical Analysis section above. Tables 2.1 and 2.2 show the outputs, summarized across 60 simulated datasets, for each candidate model. In Table 2.1, the lag times estimates produced by the kernel

smoother are on par with the other two methods. However, the standard error bars are about twice as large as those obtained from the 3 step method, as the kernel smoother is a much noisier estimator of the lag times. Taking the kernel smoother, (which is commonly used in practice to model a firing rate function) as the reference model, Table 2.2 shows the percentage reduction in estimation error obtained from each method when estimating trial to trial correlation. While our method does not have as much of an improvement over the kernel method as the IPFR model on data simulated from the IPFR model, it still considerably improves over the reference, demonstrating the ability of our method to denoise the trial peak times and to more accurately estimate the correlations.

Parameters	Model avg. estimate (s.e)		
lag	IPFR model	3-step method	Kern smooth
8	8 (0.08)	8 (0.1)	7.9 (0.26)
0	0 (0.07)	0 (0.11)	0.02 (0.21)

Table 2.1: **Lag recovery (in milli seconds) from three methods from data simulated from the IPFR model.** In two hypothetical brain areas, and for one stimulus condition, we simulated neuron spiking data, using the IPFR model as the ground truth, for different average lag in peak time between the two areas. We kept ρ, R, σ_1, σ_2, N_1, N_2, p_{a_1} and p_{a_2} fixed at 0.8, 60, 1, 1, 100, 100, 0.8 and 0.8 respectively. We applied the three methods to recover the ground truth lags, and computed the mean estimate for each method across 60 repetitions, as well as the simulation standard errors, shown in parenthesis. We note that in our simulated datasets, the kernel smoother itself produces a decent estimate of the lag time, but has standard errors that are twice as large as the other two methods.

Parameters		Model avg. estimate (s.e)			% error reduction	
ρ	R	IPFR model	3-step method	Kern smooth	IPFR model	3-step method
0.8	60	0.79 (0.05)	0.78 (0.08)	0.32 (0.11)	98	96
0.2	60	0.19 (0.1)	0.18 (0.12)	0.08 (0.19)	92	83
0.8	30	0.79 (0.07)	0.75 (0.16)	0.27 (0.19)	98	91

Table 2.2: **Correlation recovery from three methods from data simulated from the IPFR model.** Same procedure as in Table 2.1, but here we used different combinations of trial-to-trial correlations ρ, and number of trials R, with lag, σ_1, σ_2, N_1, N_2, p_{a_1} and p_{a_2} held fixed at 8ms, 1, 1, 100, 100, 0.8 and 0.8 respectively. We applied our three step method, the IPFR model, and a naïve model based on kernel-smoothed population intensities, to 60 simulated datasets, to recover the ground truth correlations. We compute the mean estimate for each method across repetitions and the simulation standard errors, shown in parenthesis. We also computed the percentage reduction in the estimation error from the naïve kernel smoother achieved by both ours and the IPFR model.

In order to demonstrate the improvement in runtime between the IPFR and the 3-step method, we ran additional simulation studies on the previously described models. First, we report the difference in runtime of both models for number of simulated areas = 2, 4, 6, and 8. For each value, we specify $N_i = 100$ neurons in each area. The results are summarized in Table 2.3. Increasing up the number of areas also increases up the number of neurons to be assigned to a peaked or flat response population, as well as the size of the covariance matrix being estimated. We used a randomly chosen mean vector and covariance matrix for the trial peak time distribution, and we used a peaked response population proportion of $p_i = 0.8$ for each area. From the results in Table 2.3, we observe between 88 and 90% reduction in the runtime of the 3-step method over the IPFR. We also separately investigate the relative runtimes of both models as we vary the number of stimulus conditions, which also varies the number of trials. Table 2.4 summarizes our results for number of stimulus conditions = 1, 5 and 10. Fixing the number of areas at 2, and the number of neurons per area at 100, we use $R_s = 60$ trials for each stimulus condition s_k. Under this configuration, the population membership of the jth neuron in area a_i now also depends on the stimulus condition s_k being considered. We set the proportion of peaked response neurons $p_{i,s} = 0.8$ for each area a_i and stimulus conditions s_k, and the peak time trial to trial correlation and variance as $\rho = 0.8$ and $\sigma = 1$ respectively. In both scaling experiments, we generated 4000 samples of each variable of interest. We obtained between an 85 and 90% reduction in runtime with the 3-step method over the IPFR for problems of the same size. We note that the simulated datasets used in this study are quite small (a single stimulus condition in the area scaling experiment, two areas in the stimulus condition scaling experiment) compared to interesting real-world datasets (5 to 10 areas

and 10s of stimulus conditions on multiple mice), and thus the model time complexity becomes progressively more important on the scales of real-world data.

Model avg. runtime in mins (s.e)			
# Areas	IPFR model	3-step method	% runtime reduction
2	192 (11.3)	23 (4.3)	88
4	278 (14.7)	31 (3.9)	89
6	349 (12.3)	35 (4.7)	90
8	406 (15.8)	44 (3.6)	88

Table 2.3: **Runtime comparison between the IPFR and the 3-step method with varying number of areas.** We simulated neuron spiking data, using the IPFR model as the ground truth, for A = 2, 4, 6, and 8 areas, each with 100 neurons. We had s = 1 stimulus condition, with R = 60 trials. We used a randomly chosen mean vector and covariance matrix for the Gaussian trial peak time distribution. The proportion of neurons in each area belonging to the peaked response population was fixed at 0.8. We applied the IPFR and the 3-step method to 10 simulated datasets, running the algorithm for 4000 iterations in each instance. We measured the average runtime in both models, as well as their standard errors across repetitions, shown in parenthesis. We also computed the percentage reduction in runtime obtained by the 3-step method over the IPFR.

Model avg. runtime in mins (s.e)			
# Stimulus conditions	IPFR model	3-step method	% runtime reduction
1	184 (10.3)	21 (5.3)	89
5	221 (10.8)	31 (5.9)	86
10	307 (9.2)	45 (6.2)	85

Table 2.4: **Runtime comparison between the IPFR and the 3-step method with varying number of stimulus conditions.** Similarly to table 2.3, we simulated neuron spiking data, using the IPFR model as the ground truth, for s = 1, 5 and 10 stimulus conditions, each with 60 trials, for A = 2 areas with a 100 neurons in each. The trial-to-trial correlations ρ, lag, σ_1 and σ_2 were held fixed at 0.8, 8ms, 1 and 1 respectively. The proportion of neurons in each area belonging to the peaked response population was fixed at 0.8 for all stimulus conditions. We applied the IPFR and the 3-step method to 10 simulated datasets, running the algorithm for 4000 iterations in each instance. We measured the average runtime in both models, as well as their standard errors across repetitions, shown in parenthesis. We also computed the percentage reduction in runtime obtained by the 3-step method over the IPFR.

Illustration with real data

We chose one example subject, and two visual areas V1 and LM, to demonstrate the method's correction for attenuation of correlation, and the extent to which this is facilitated by both the sub-population selection and the denoising. Results are shown in Figure 2.6. We estimated the Pearson correlation coefficient for the trial-by-trial peak 1 times in three cases. First (panel A), peak times were obtained by applying a kernel smoother on each trial to the PSTH based on the full population (without the interacting population selection step). The correlation was .06. Second (panel B), we applied the kernel smoother to the PSTH after selecting the interacting population. The correlation increased to 0.2. Third (panel C), the peak times were obtained as posterior means from the Bayesian hierarchical model, according to the complete three-step approach. The correlation obtained further increased substantially to 0.8. The results from our simulation studies suggest that the correlation value of 0.8 were likely attenuated to 0.06 in the näive estimates shown in panel A. See [Behseta et al., 2009] for further depictions of attenuation of correlation in milti-trial spike count data. This example case illustrates the importance of the denoising step in our procedure.

Analysis of data from multiple mice

To understand the functional ordering present in feedforward signal propagation in the visual cortex, we applied our method to data from thirteen mice from the Allen dataset, estimating the average trial-by-trial peak times in

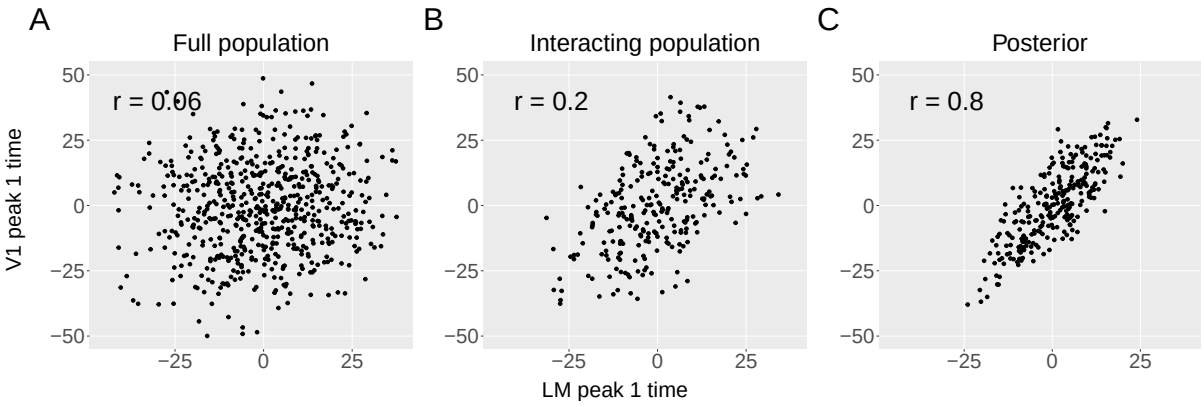

Figure 2.6: **Denoising of peak 1 times for regions V1 and LM in an example mouse. A:** Plot of Estimated peak 1 times using a kernel smoother applied to the condition-specific PSTH based on the full populations of recorded neurons. **B:** Plot of estimated peak 1 times after interacting sub-population selection. **C:** Plot of estimated peak 1 times after applying the full three-step method.

each of seven areas. We then computed the standard deviations of the peak time across mice for each area. Figure 2.7 shows the weighted means, standard errors, and standard deviations for peak (1 and 2) times across mice.

While the ordering of Peak 1 times in Figure 2.7 is consistent with previous results [Siegle et al., 2021], that figure can not indicate the extent to which such timing is or is not consistent across subjects. We examine consistency next.

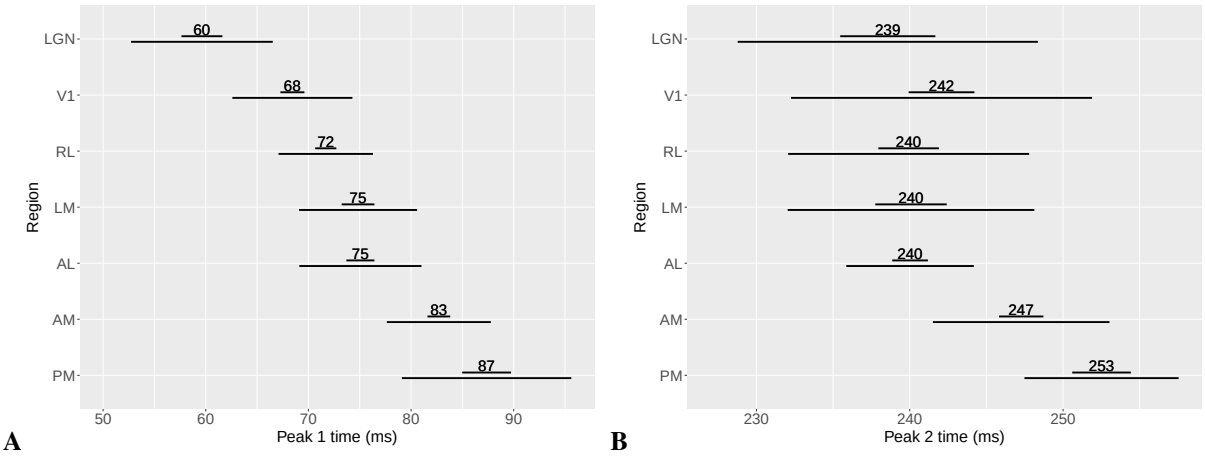

Figure 2.7: **Weighted means, standard errors, and standard deviations across mice for peak 1 time and peak 2 time.** The shorter horizontal bar (top) represents the standard errors, and the longer bars (bottom) represent the standard deviations, **panel A** peak 1 (13 mice), **panel B** peak 2 (11 mice). The ordering in peak 1 times largely disappears in peak 2 times, except that areas AM and PM appear to have somewhat later peak 2 times.

Consistency in ordering of peak times. Despite subject-to-subject variability in actual peak time values, Figure 2.8 shows some consistency in the ordering of peak times across mice. The figure also displays some inconsistencies. For peak 1, across all mice, we observe LGN preceding V1, which precedes higher-order visual areas. For all mice we also observe peak 1 time in both AM and RL preceding PM. However, apart from these relationships, the relative peak 1 timing among higher-order visual areas is specific to each mouse. For peak 2, for all mice LGN and V1 both precede AM and PM, and again RL precedes PM, but all other orderings are inconsistent. For example, LGN precedes V1 sometimes, but not uniformly across mice, and V1 precedes higher-order areas other than AM and PM sometimes, but not for all mice. Again, as with Peak 1, the relative timing among higher-order areas apart from RL and PM is inconsistent.

Peak time correlations among the cortical areas tend to be stronger than those between cortical areas and LGN. We obtained the trial-to-trial correlations in peak times between pairs of areas from the entries of the

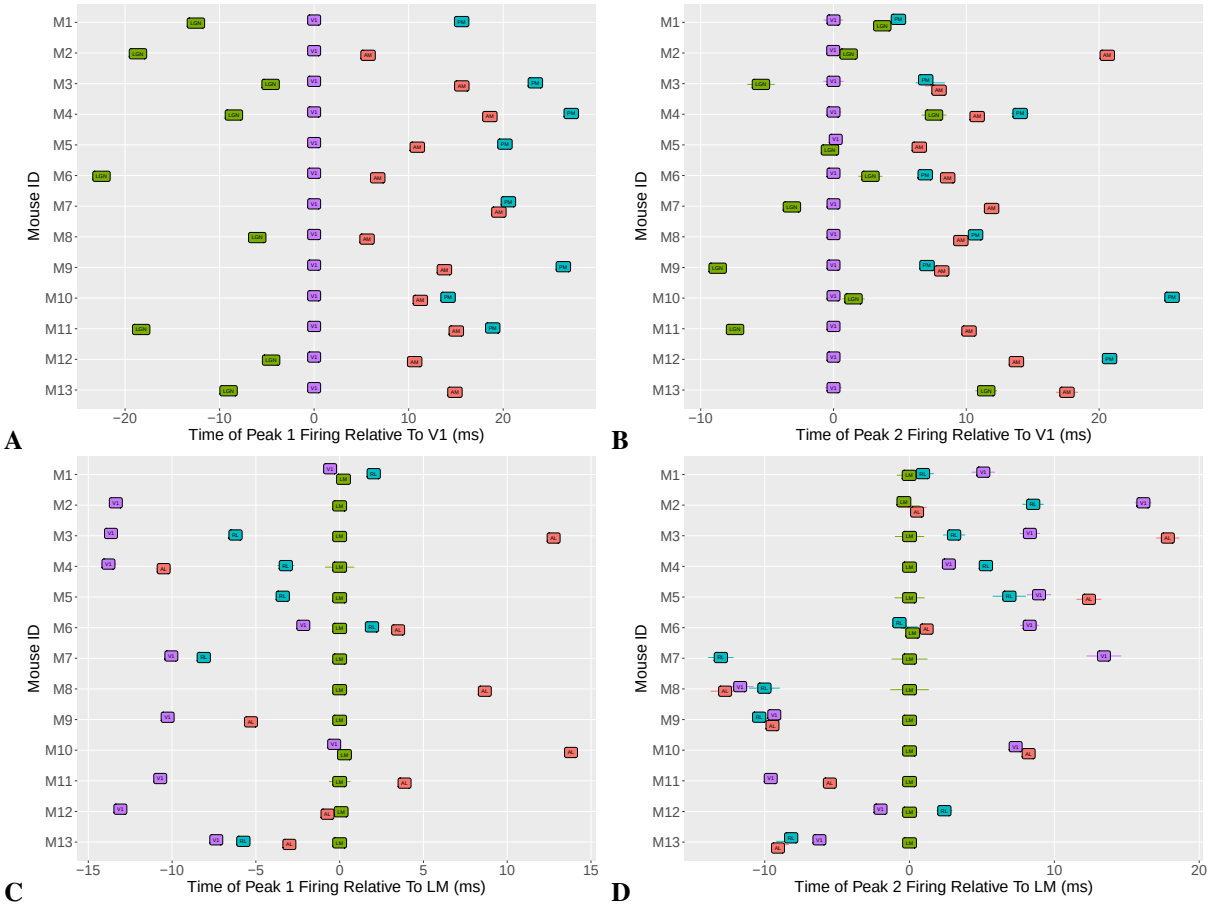

Figure 2.8: **Mouse-to-mouse variability in the time of the initial peak response relative to a reference region.** In panels **A** and **B**, we show Peak 1 and peak 2 (respectively) time estimates for LGN, AM and PM, relative to the corresponding peak time estimate for V1, for the same set of thirteen mice. Panels **C** and **D** show the peak 1 and peak 2 times estimates for V1, RL, LM and AL, relative to the corresponding peak time estimate for LM. In all cases, 1 standard error bar is also shown for the peak time estimates, although many are small enough to be obscured by the region label. We observe a consistent ordering across mice in the peak 1 times of LGN, V1, AM and PM in **A**, suggesting a functionally relevant pathway. We don't observe the same consistency in the peak 2 times for these areas in **B**, although we see LGN and V1 consistently reach their second peak before AM and PM. Among the regions V1, RL, LM and AL, we see that for peak 1 in **C**, V1 tends to reach its peak before the other three, although there appears to be no clear ordering among the three. There is no discernable pattern in the peak 2 times among this set of regions in **D**.

matrix $\mathbf{R}$ described in the modeling section, and aggregated them across mice for both peaks using a weighted mean, where weighting is done using the standard error for each mouse. (As previously stated, this is a lower variance estimator than the simple arithmetic mean.; see the discussion in [Kass et al., 2014, Chapter 8].) As shown in Figure 2.9, the cortico-thalamic correlations tended to be lower on average than the correlations among the cortical areas, although this was more striking for peak 1 than for peak 2.

We also computed the mouse-to-mouse standard deviations of these correlations in each case. Figure 2.10 shows that the correlations between LGN and the cortical areas generally appear more variable across mice than correlations among cortical areas.

Correlations between thalamic and early cortical areas show the largest percentage changes after conditioning on V1. Given the neuroanatomy of the mouse thalamocortical pathway, we sought to quantify the involvement of V1 in the interactions between pairs of areas. We computed the partial correlations between the peak times for each pair of visual areas, conditioned on V1, and compared this to the original (marginal) correlation for the pair. Specifically, we computed the percentage decrease from marginal correlation to partical correlation for each pair of areas, where the partial correlation conditioned on V1. As seen in Figure 2.11, for both peaks, the pair AM–PM has the least decrease in correlation given V1, which is expected given that they are farthest

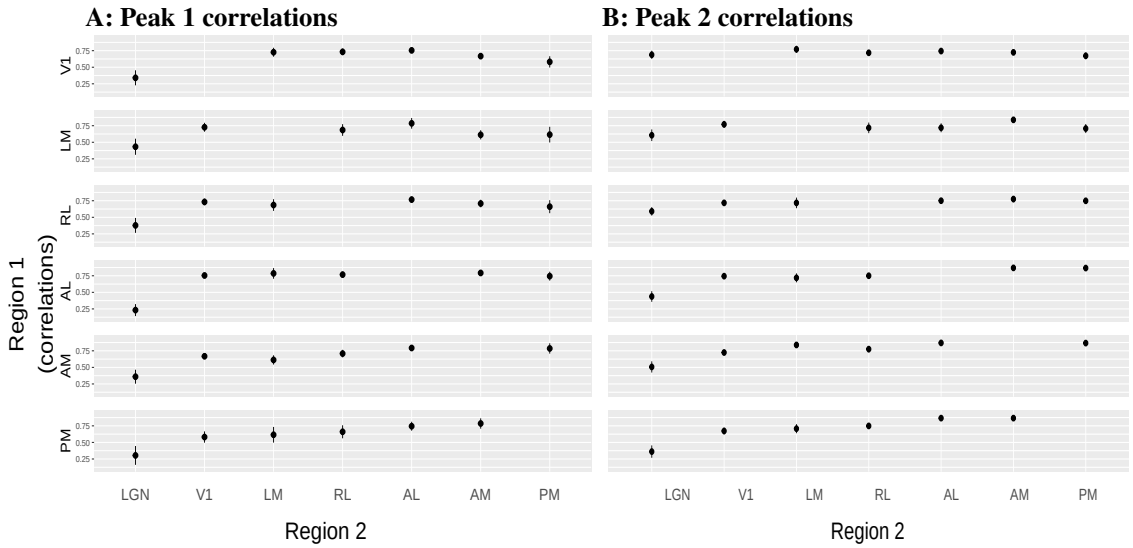

Figure 2.9: **Trial-to-trial correlations in the peak times between pairs of areas.** Each panel shows the weighted mean correlations of peak times between pairs of areas, across $N = 13$ mice, with their one standard error bars. Each row in each panel shows the correlations between a single visual area and all other areas. We observe in **A**, that the correlations in peak 1 times among the cortical areas tend to be stronger than those between cortical areas and LGN. We observe this to a lesser degree in **B**, along with the fact that peak 2 correlations tend to be stronger than their corresponding peak 1 correlations.

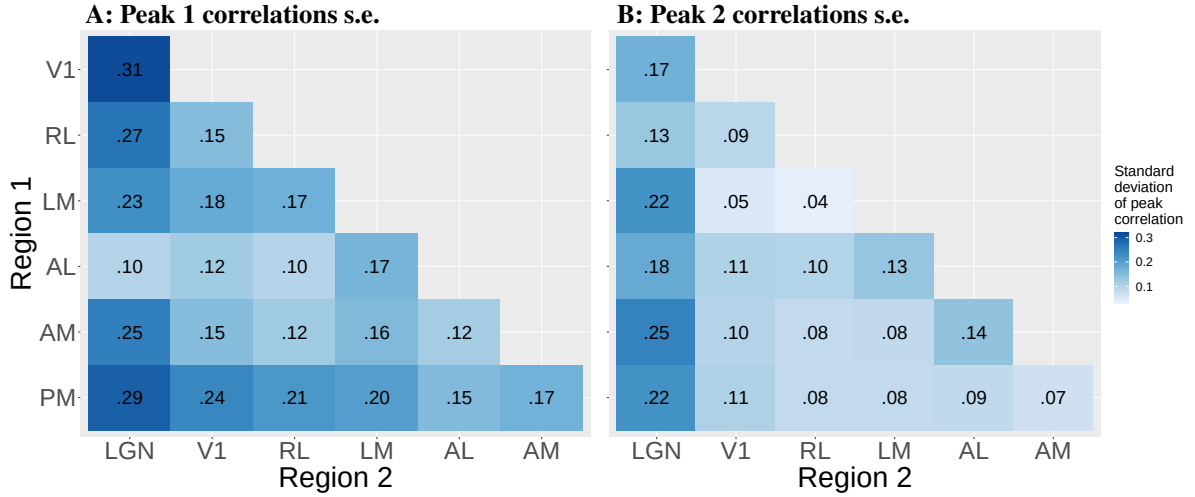

Figure 2.10: **Standard deviations across mice of pairwise correlations between peak times.** Each entry in the heat map represents one standard deviation of the pairwise correlations in peak times between the corresponding pair of regions. the color corresponds to the magnitude of the standard deviation. The figure reveals that the peak 1 correlations tend to be more variable across mice than the peak 2 correlations.

areas from V1 in the visual hierarchy, and there are no expected projections between this region pair through V1 [Harris et al., 2019, Siegle et al., 2021]. Among the cortical regions, according to the results for peak 1, the further downstream the region is from V1, the smaller the drop in its correlations with other regions after conditioning on V1. The biggest drops in correlations are seen in the correlations of LGN and the cortical areas, suggesting strong mediation of these interactions by V1. In particular, the correlation of LGN with RL for peak 1, goes from a positive to a negative correlation after conditioning on V1 (the decrease in correlation is greater than 100%). For peak 2 there are similar large decreases for pairs involving LGN, but the combination of feedback projection with the inconsistent timing results in Figure 2.8 (especially for LGN and V1) suggests the reversal of correlation between LGN and each of AM, AL, and PM after conditioning on V1 may be due to bidirectional connections between V1 and AM, AL, and PM.

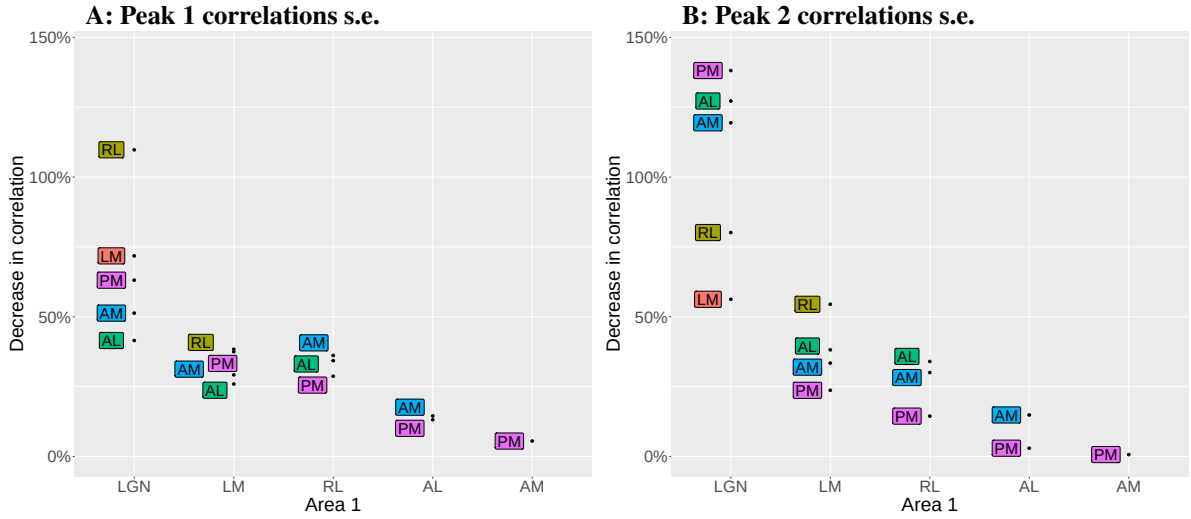

Figure 2.11: **Percentage decrease in the in correlations between pairs of areas after conditioning on V1.** Each labeled point shows the percentage decrease in the peak time correlations for the region pair consisting of the text label region and the corresponding region on the x-axis. For example, after conditioning on V1, the correlation between between the peak 1 times in LGN and AL decreased by about 38%. The standard errors in all cases are $< 10\%$. The previously positive correlation between LGN and RL in peak 1, and between LGN and AM, AL and PM in peak 2 became negative after conditioning V1 (the decrease in correlation is greater than 100%).

Discussion

Studies of sequential timing of activity across brain areas have generally relied on data aggregated across trials. We aimed to develop, assess, and illustrate a relatively simple and computationally efficient method for identifying precise trial-by-trial sequential timing in population activity across areas. Motivated by results of [Chen et al., 2022], we created a straightforward 3-step procedure and found it was nearly as accurate as the more complicated methodology in [Chen et al., 2022] while running about 10 times faster, which enabled our comparative analysis of data from multiple subjects. This powerful method is accessible to the many neuroscientists who could apply it to recordings from large populations of spiking neurons.

Our examination of the variability in sequential timing relationships across 13 mice in the Allen Brain Observatory Visual Coding Neuropixels dataset [Allen Institute MindScope Program, 2019] produced results that are consistent with known anatomy and physiology, while also highlighting the distinction between pathways that are consistent across subjects versus those that are idiosyncratic. In the feedforward case of peak 1, for example, while LGN activity always precedes V1 acticity which always precedes activity in higher-order visual areas, most of the timing relationships among those higher-order visual areas are subject dependent. In the case of peak 2, which involves feedforward, feedback, and inputs from other areas, the relative timing of LGN and V1 is subject dependent.

We note that the subject-to-subject variability we observed may be attributed primarily to differences in animal physiology, differences in the experimental setup used in data collection, or to a combination of both these and other factors. Although inconsistencies across mice remain to be explained in greater detail, the results that were nearly the same across mice are compatible with existing notions of a functionally relevant hierarchical visual pathway for feedforward signal propagation. One interesting set of results that await further exploration involves LGN: the correlations in peak time between LGN and the cortical areas are weaker, on average, and more variable across mice, than those among the visual cortical regions themselves. A potential explanation is that targeting Neuropixels probes to deep brain structures such as LGN, based solely on a map of the visual cortex, is prone to inaccurate placement, resulting in poor representation of relevant LGN neural populations and increased variability across mice. Similarly, partial correlations after conditioning on populations that are incompletely sampled by electrodes must be interpreted carefully. The contrast of the substantial decrease in correlation between LM and AM, after conditioning on V1, versus the small decrease in correlation between PM and AM may seem consistent with notions of visual hierarchy. On the other hand, the modest decrease in correlation between LGN and PM, after conditioning on V1 might be due to the many paths from V1 to PM (which could create variation in timing, as seen in the reduced correlation of V1 with PM compared to other areas) or it could be that key projection neurons may not have been sampled.

To the extent that the substantial mouse-to-mouse variability in functional connectivity observed across partic-

ular visual areas may have physiological sources, they could be genetic, developmental, or experiential factors (or combinations of these). Future studies could explore these distinctions, for example, by comparing response timing in different mouse lines or mice with different types of visual exposure (e.g. dark-reared). Such investigations would be especially useful if combined with causal manipulations.

We have demonstrated the usefulness of this method in estimating peak times and their trial-to-trial correlations for spike train data. Although it is notoriously difficult to tease out informative trial-to-trial fluctuations in continuous data such as EEGs, Klein *et al.* [Klein et al., 2021] decomposed local field potentials (LFPs) into current source densities (CSDs) on a trial-by-trial basis from which they demonstrated cross-population frequency coupling that was not apparent from the original LFPs. A variant of the methodology developed here might, in a similar vein, be useful for establishing timing relationships from CSDs. On the other hand, our analysis of bursts in temporally evolving firing rate functions takes advantage of the substantial information about the timing of their maxima; by definition, that is where the most spikes occur. We also strengthened covariation relationships by confining attention to a single, homogeneous sub-population of neurons in each area. By instead examining multiple sub-populations, future work could investigate the diversity of functional interactions across areas.

Part 2

A generalized approach to inferring Functional Connections

Chapter 3

Learning latent graphs from noisy time-series data

This is a collaborative work with Neil Spencer and Robert Kass. We plan to submit this work to ICML.

Neural functions arise from the interactions between brain networks, where each node in the network is a sub-population of synchronously firing neurons, which form the "functional units" of communication within the brain [Bassett and Sporns, 2017, Bressler and Menon, 2010, Buzsaki and Draguhn, 2004, Sporns et al., 2004]. However, these functional units are not directly observed in neural recordings. For example, using neurophysiological recording devices, we are only able to directly observe the activity of individual neurons within the brain, with no direct visibility of the synchronous activity within, and interactions between, neuron populations. To understand the way in which these functional units communicate, both within and across anatomically mapped brain regions, we require a method to identify the functional units, as well as to quantify the degree of interactions between these units. Furthermore, modern electrophysiological recording devices are able to record from a large number of neurons across multiple brain regions in multiple trials of a single experiment [Jun et al., 2017, Siegle et al., 2021, Steinmetz et al., 2021]. We, therefore also require a method that is able to analyze data from multiple experiments at scale. In this chapter, we present a modeling framework capable of accomplishing both these goals. We demonstrate the features of the model using simulated data and apply our model to experimental data from the Allen Brain Observatory to study functional connectivity in the visual cortex of mice.

3.1 Introduction

3.1.1 Problem statement and motivation

The work outlined in this section is motivated by the limitations of our previous work. While the IPFR and three-step interacting population models previously described produced reasonable connectivity results at small timescales from simulated data, they are able to fit only a subset of the data typically collected in real world neuroscience experiments, as they require the presence of specific bursting patterns in neuron populations. They therefore make use of pre-screening steps to account for the observed diversity in stimulus-dependent population response profiles, a consequence of inhomogeneity in neuron responses both within and across stimulus conditions. Furthermore, they justify the prescreening by making strong assumptions about the response profile of a functionally relevant population, which necessitates filtering out neuron populations that do not conform to these assumptions. The outlined limitation can be understood by considering the neuron response profiles depicted in Figure 3.1. The figure shows 3 examples of the neuron firing patterns found in a typical dataset of spiking neurons. Both the IPFR and the 3 step model prescreening procedures will filter out populations 1 and 2, as they do not display the expected dual peak bursting response that the model expects for a functionally relevant population, despite the fact that the first two populations clearly display a peaked response. Therefore, these prescreening procedures tend to filter out potentially relevant populations of neurons.

Given that multiple population activity is observed in neural recording, we aimed to develop a method to identify homogenous neuron populations, and their interactions, from the observed spiking activity of a collection of neurons. Any suitable model must therefore include the following components: 1. The ability to cluster neurons into homogenous populations based on their observed patterns, and 2. The ability to infer the functional interactions

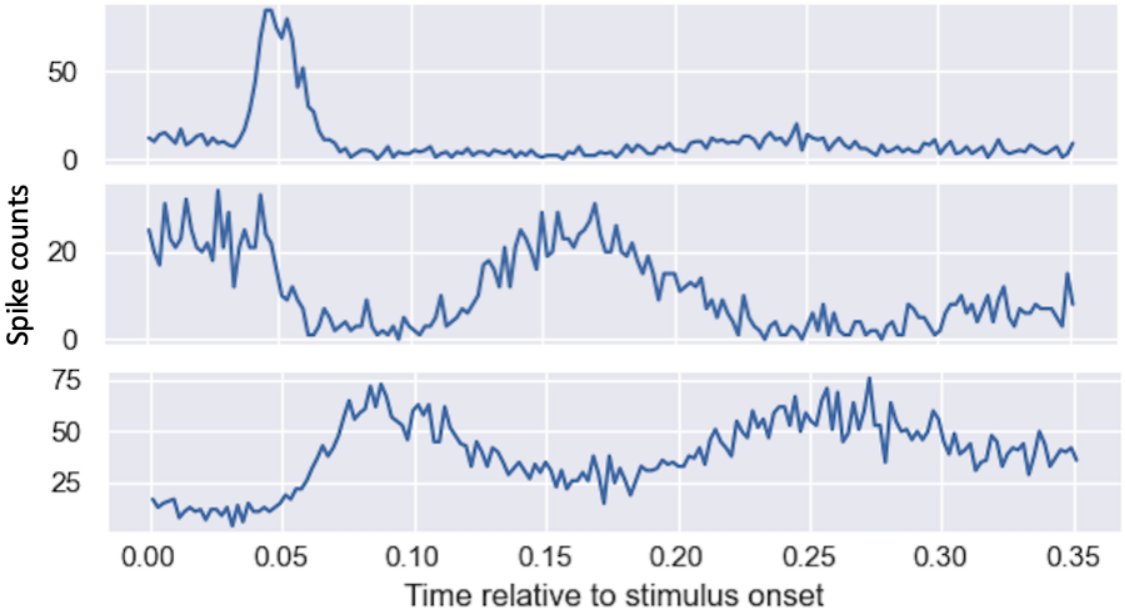

Figure 3.1: The figure shows the population response profile for three neuron populations in a single region of the visual cortex. We see here that only the third population exhibits a characteristic dual peaked response to stimulus. the first 2 populations do, however, show a peaked response, which may be relevant to functional interactions. It would therefore be useful to consider these populations when determining the functional interactions between regions in the brain.

given these populations. In this chapter, we apply a probabilistic graphical modeling framework that combines these two components to infer the dependencies between spiking neurons across functional units. Our framework formalizes this problem as learning the nodes and edges of a probabilistic graph, where the nodes represent the populations of homogenous neurons, and the edges represent the interaction between the populations. We specify a probabilistic graphical model to describe the joint distribution between the interaction populations represented by the nodes. We apply time series clustering to identify the nodes in the model and the edges are inferred from the graphical model. This framework is illustrated in Figure 3.2.

An added source of complexity comes from the fact that the observed data within a population are misaligned in time, making direct clustering ineffective. We address this problem by incorporating a time-warping function during clustering, whose parameters are learned in the model fitting process.

This joint probability distribution, which we refer to as the Mixture of Dependent Poisson Point Processes (MDoP3), is depicted in detail in Figure 3.5. We apply a mixture of time series clustering technique to identify the interacting populations, which we define as the interactions between the corresponding firing rate cluster centers, which correspond to the nodes in Figure 3.2. We also incorporate latent variables representing the neuron population membership, as well as neuron-specific and trial-specific latent firing rate features. The formulation of the mixture of time series framework as a latent variable model facilitates fitting the parameters of the resulting joint likelihood using the Expectation Maximization (EM) algorithm [Dempster et al., 1977]. The optimization problem, when learned using a gradient descent algorithm, also lends itself to the parallel computation of vector operations implemented in frameworks like PyTorch or TensorFlow. These frameworks take advantage of GPU parallelization for much faster inference than is possible with the MCMC methods used previously. The increased processing speed reduces computation bottlenecks, and in turn, promotes scaling the model to larger datasets. It also enables iteration by practitioners by cutting down the time it takes to fit a model.

3.1.2 Related work

Prior work has attempted to address the question of understanding functional connections between brain areas. However, these have been done at the level of the anatomical brain region. For example, Chen et al. [2022] developed a model to study the functional dependencies between functional units with a characteristic stimulus

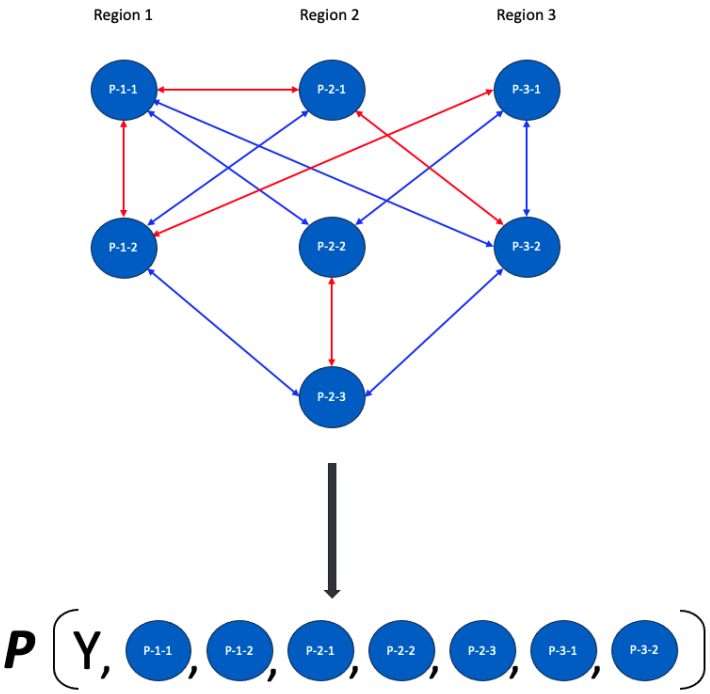

Figure 3.2: The figure shows a conceptual depiction of our modeling framework. Each node in the probabilistic graph, labeled $P - i - j$ represents a homogenous population or cluster of the neurons in the object denoted as Y. In our application, the i in $P - i - j$ represents an anatomically mapped brain region, and the j represents a homogenous subpopulation, or cluster, within the brain regions. We refer to these homogenous subpopulations as "functional units". The bidirectional arrows represent the dependencies between the nodes, which we model as correlations, with red indicating a positive correlation, and blue representing a negative correlation. The graph therefore describes the joint distribution of our observed data Y and the latent interacting populations P.

response profile. The prior work in this thesis itself is aimed at providing a method to answer this question that is scalable and easy for practitioners to implement. It is, however, also primarily limited to studying cross-area functional dependencies and relies on a characteristic functional unit of neurons to measure these dependencies. Bullmore and Sporns [2009] analyzed functional networks with binary edges using graph theoretic measures of network topology. However, they defined the network nodes either anatomically or based on electrodes, which may or may not correspond to individual functional units. Smith et al. [2011] and Biswal et al. [2010] both looked at correlations in BOLD FMRI for various Regions of Interest, but this modality is unable to discover the connectivity patterns over fast time scales. Methods for model-based time series clustering, which can identify functional units in the brain, have also been suggested. Lin et al. [2019] suggests a Bayesian approach to clustering neuron spike trains using a mixture of state space models. Humphries [2011] proposed a method for detecting communities among a set of spiking neurons by running a clustering algorithm over a predefined similarity spike train similarity matrix, such as the van Rossum distance [van Rossum, 2001] and the Victor-Purpura distance [Victor and Purpura, 1996]. Houghton and Sen [2008] and Sotomayor-Gómez et al. [2023] both proposed an extension to the aforementioned distance metrics that allows them to scale to large multi-neuron datasets. The contents of this chapter incorporates both clustering of neuron spike trains and identifying functional connectivity, while delivering fast and efficient computation that scales to large datasets. Specifically, our framework is able to separate out the functional units within each brain region, while explicitly accounting for cross-trial temporal variations in the neuronal spiking profile, making this much more robust than previous time series clustering models. Prior methods have been limited to quantifying functional interaction across anatomical brain regions. Using our proposed method, we are able to identify the strength of the functional interactions, if present, between functional units, both within a single brain region and across brain regions. In addition, our method is highly optimized, and therefore applicable to large datasets, and highly configurable to suit a wide variety of experimental settings.

3.2 Methods and materials

3.2.1 Model

Let k be the index of a neuron, and let l be the index of a homogenous neuron sub-population, otherwise called a functional unit, and $s_k = \{s_1, \ldots, s_{N_k}\}$ be the observed neuron spike times for the N_k observed spikes of neuron k. Then, the event times sequence s_k is modeled according to a Poisson point process with the joint pdf given as:

$$f_{S_k}(s_k) = \exp\Big(- \int_0^T \lambda_k(t)dt \Big) \prod_{i=1}^{N_k} \lambda_k(s_i)$$

Where $\lambda_k(t)$ is the intensity function for neuron k on the interval $(0, T]$. In our model, $\lambda_k(t)$ takes the following functional form:

$$\lambda_k(t) = \exp\Big(d_k + \tau_l(t)\Big) \tag{3.1}$$

where

$$\int_0^T \lambda_k(t)dt = \int_0^T \exp\Big(d_k + \tau_l(t)\Big)dt = \exp(d_k) \int_0^T \exp\big(\tau_l(t)\big)dt = E_k \tag{3.2}$$

d_k is the log mean spike count on the interval $(0, T]$, τ_l is the log of the population intensity function for the homogenous subpopulation l to which k belongs, $\exp(d_k) = E_k$ is the mean spike count for the neuron k on the interval $(0, T]$, and by definition in equation 3.2, $\int_0^T \exp\big(\tau_l(t)\big)dt = 1$.

Returning to equation 3.1, and letting $\exp\big(\tau_l(t)\big) = p_l(t)$, we have that

$$\lambda_k(t) = E_k \; p_l(t) \tag{3.3}$$

and so

$$f_{S_k}(s_k) = \exp\Big(- \int_0^T E_k \; p_l(t)dt \Big) \prod_{i=1}^{N_k} E_k \; p_l(s_i) = \frac{E_k^{N_k} e^{-E_k}}{N_k!} N_k! \prod_{i=1}^{N_k} p_l(s_i)$$

The point process joint distribution decomposes into a Poisson distribution over the number of spikes, and a density over the time of the spikes. The $N_k!$ term denotes the number of ways to order the spikes $\{s_1, \ldots, s_{N_k}\}$. The discrete-time analog can be expressed in the following way: $\mathbf{Y}_k = [y_1, \ldots, y_T] \in \mathbb{R}^T$, is the observed binary spike train vector for neuron k for T time bins. The discrete-time joint distribution is given as

$$f_{Y_k}(Y_k) = \frac{E_k^{\sum_t y_t} e^{-E_k}}{\sum_t y_t!} \sum_t y_t! \prod_{t=1}^{T} p_{l,t}^{y_t}$$

The full derivation for the discrete-time point process is given in appendix A.
In order to satisfy the constraint on (the discrete analog of) $p_l(t)$ in eqn 3.3, $\sum_t p_{l,t} = 1$, we normalize $p_{l,t}$ using the following parametrization:

$$p_{l,t} = \frac{\exp(\beta_{l,t})}{\sum_t \exp(\beta_{l,t})}$$

We also include a prior distribution over the mean spike count E_k, by assuming $E_k \sim \text{Gamma}(\alpha_l, \theta_l)$. The choice of the Gamma prior is motivated by the fact that if $Y|\lambda \sim \text{Poisson}(\lambda)$ and $\lambda \sim \text{Gamma}(\alpha, \theta)$ then $Y \sim \text{NB}\Big(\alpha, \frac{\theta}{1+\theta}\Big)$ This means that inferring the subpopulation intensity functions in this model corresponds to doing negative binomial regression on the population PSTH of all the neurons in the corresponding functional unit. This makes this model well-suited to deal with overdispersed data.

3.2.2 Time-Warping

One of the stated goals of our model is to enable inference about the functional interactions between functional subpopulations within and across brain regions. In the previous chapter, we approach this problem by assuming that behaviorally relevant information is transmitted across parts of the brain through transient bursts of activity in the subpopulations of neurons relevant to a given stimulus. In this framework, the timing of these bursts, as measured by the timing of their peaks on a trial-by-trial basis, should reveal coordinated activity. To capture this coordinated activity, we decompose the peak bursting time for a given subpopulation into the following:

1. The average population peak burst time across all trials and all stimulus conditions, which we denote as τ^*
2. The average peak burst time across all trials within a stimulus condition, which we denote as τ_c
3. The peak burst time on a given trial of stimulus condition c, which we denote as τ_{r_c}

Given this, we define the following offset variables: $q_c = \tau_c - \tau^*$, $s_{r_c} = \tau_{r_c} - \tau_c$, that is, the stimulus conditions and trial peak burst time shift, respectively. As we are interested, in the shifts on a trial-by-trial basis, we model the coordinated activity across functional units as a correlation between these trial-to-trial shifts. In particular, we model $s_{r_c} \sim \mathrm{N}(0, \Sigma)$, where each dimension of s_{r_c} contains the peak time offset for a functional subpopulation. The peak time offsets are incorporated into the population intensity functions $p_{l,t}$ by time-warping $p_{l,t}$ using a piecewise linear function $\phi(t, s_{r_c}, q_c)$ [Williams et al., 2020]. $\phi(t, s_{r_c}, q_c)$ is parametrized by landmarks τ_l and τ_r, which respectively define the left and right boundaries of the time-warping window. Time warping therefore aligns the observed population spike trains on each trial of each stimulus with the population intensity function $p_{l,t}$ at their peaks. We define the time warping function $\phi(t, s_{r_c}, q_c)$ in the following way:

$$\phi(t, s_{r_c}, q_c) = \begin{cases} t, & 0 \le t < \tau_l \\ \phi_1(t) = (t - \tau_l)\frac{\tau^* - \tau_l}{(\tau^* + s_{r_c} + q_c) - \tau_l} + \tau_l, & \tau_l \le t < \tau^* + s_{r_c} + q_c \\ \phi_2(t) = (t - (\tau^* + s_{r_c} + q_c))\frac{\tau_r - \tau^*}{\tau_r - (\tau^* + s_{r_c} + q_c)} + \tau^*, & \tau^* + s_{r_c} + q_c \le t < \tau_r \\ t, & \tau_r \le t < T \end{cases}$$

In our application, the function operates by taking as input a time point t in discrete time, as well as a stimulus and config offset s_{r_c}, q_c, and returning as output a new time point $\phi(t, s_{r_c}, q_c)$ in continuous time. The value of the function $p_{l,t}$ is then updated to $p_{l,\phi(t,s_{r_c},q_c)}$ by linearly interpolating between the closest time pinots $t_i \le \phi(t, s_{r_c}, q_c)$ and $t_{i+1} \ge \phi(t, s_{r_c}, q_c)$. Figure 3.3 provides a visual illustration of the action of the time-warping function on the population intensity near the peak times.

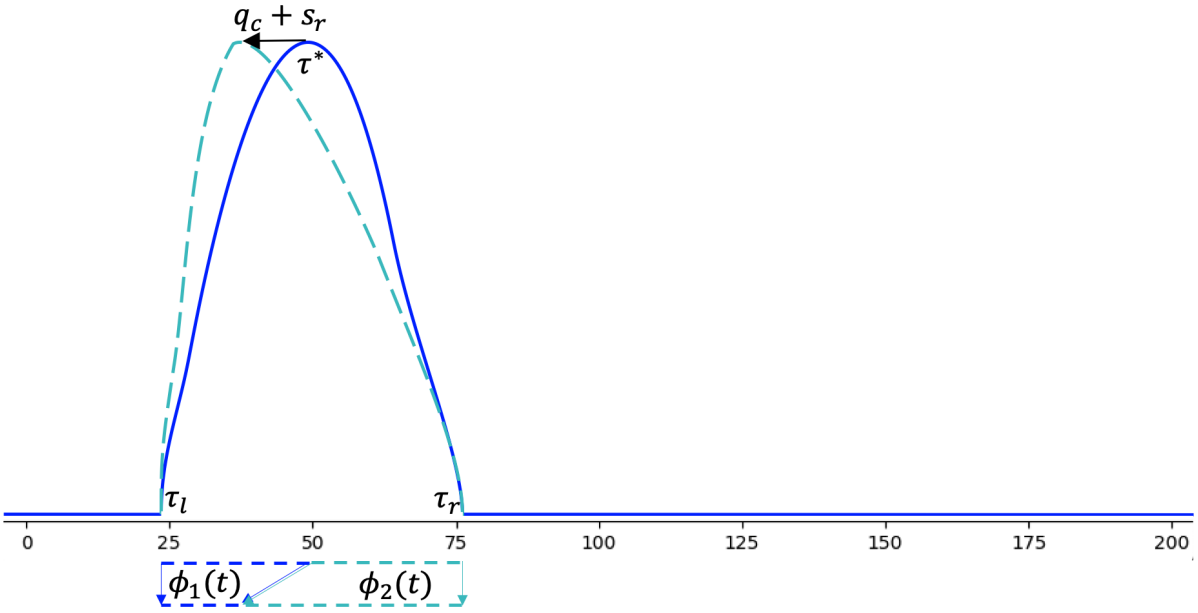

Figure 3.3: Visual illustration of the action of the time warping function on the population intensity near the peak times. The function operates by taking as input a time point t, and then returning as output a new time point $\phi(t)$. The intensity function is then given as $\beta_l(\phi(t))$ warped function would correspond to at time t.

In order to ensure that the trial peak times remain bounded within the time-warping window $[\tau_l, \tau_r]$, we use a piecewise linear scaling function, shown in Figure 3.4, to constrain the peak times. We also penalize the magnitude of the stimulus configuration-specific offset, q_c, as well as the variances of the trial offsets, s_r, which also encourages small peak offset times.

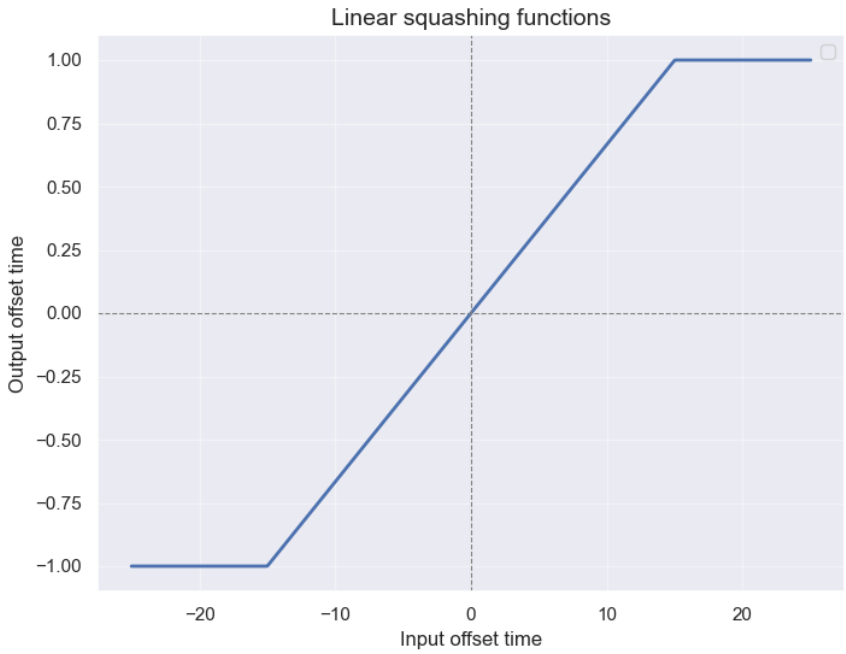

Figure 3.4: We used a piecewise linear squashing function over a broad range to constrain the trial peak time offsets within the time-warping window. We chose a linear squashing function because we are interested in linear trial to trial correlations of the trial peak time offsets.

3.2.3 Inference

The complete likelihood is given by

$$P\Big(\{t_{k,r,t}\}_{K,R,T}, \{E_{k_c}\}_K, \{g_{k_c}\}_K, \{s_{r_c}\}_R;\ \mathbf{\Phi}\Big) =$$
$$\prod_c \Bigg[\prod_{k_c} \prod_{r_c} \prod_t \Bigg(\frac{1}{y_{k,r,t}!} \lambda(E_{k_c}, g_{k_c}, s_{r_c}, t)^{y_{k,r,t}} \exp\big(-\lambda(E_{k_c}, g_{k_c}, s_{r_c}, t)\big) \Bigg)$$
$$\prod_{k_c} \Big(P(E_{k_c}|g_{k_c}) \Big) \prod_{k_c} \Big(P(g_{k_c}|\pi_a) \Big) \prod_{r_c} \Big(P(s_{r_c}) \Big) \Bigg] \tag{3.4}$$

$P\big(E_{k_c}|g_{k_c} = l\big) = \text{Gamma}(\alpha_l,\ \theta_l)$
$P\big(g_{k_c}|\pi_a\big) = \text{Categorical}(\pi_a), \quad \pi = [\pi_1, \ldots, \pi_A] \in \mathbb{R}^L$
$P\big(s_r\big) = \mathcal{N}(0,\ \Sigma_s)$
$P\big(q_c\big) = \|q_c\|_2^2 \quad \text{(Ridge penalty)}$
$P\big(\Sigma_s\big) = \|\Sigma_s^{-1}\|_1 - \|\text{diag}(\Sigma_s^{-1})\|_1 \quad \text{(Sparsity prior/penalty)}$
$P(\beta_l) = \left\| \frac{d}{dt}\beta_l \right\|_2^2 \quad \text{(Roughness prior/penalty)}$

Where

- $\mathbf{\Phi}$ represents the parameters of the joint likelihood
- λ is the conditional intensity function
- E_k is the trial firing rate for the k^{th} neuron
- g_k is the membership for the k^{th} neuron. It indicates which population the neuron belongs to

- $\beta_l = [\beta_{l,1}, \ldots, \beta_{l,T}]$ represents the latent intensity function for the l^{th} neuron population. It defines the behavior of a corresponding population of neurons over the time interval $[0, T]$
- s_{r_c} is the population peak time offset from the overall average on r^{th} trial of condition c (trial-specific feature)
- q_c is the peak time offset from the overall average peak time of the neuron population for stimulus condition c (condition-specific feature)
- Σ_s is the covariance matrix for the trial-to-trial peak time offsets

As was previously discussed, the model has a clustering component, which uses a mixture model (in this case, a mixture of Poisson processes) to identify the functional units within a set of neuron spike trains. The connectivity component is incorporated through the peak time shifts in the bursting behavior of the functional units. The full covariance structure of these peak times is parametrized by Σ_s. The stimulus condition and trial peak time shifts (s_{r_c} and q_c) are incorporated into the latent intensity functions $p_{l,t}$ by time-warping, as described in subsection 3.2.2. Figure 3.5 shows a plate diagram of the specified probabilistic graphical model, as well as a description of the parameters. The observed data are the neuron spike trains $y_{a,c,k,r}$, observed for neuron k in area a on trial r for stimulus configuration c. A neuron's spike train $y_{a,c,k,r}$ depends on its functional unit membership, which is determined by the latent variable $g_{a,c,k} \in \{1, \ldots, L\}$. The members of a functional unit share a normalized population firing rate function $\beta_{a,l}$ in area a. This is combined with the neuron's mean spike count over the course of each trial $E_{a,c,k}$, the peak offsets $q_{a,c}$ for condition c in area a, and the covarying trial peak times, $s_{a,c,r}$ for trial r, to determine the neuron's specific intensity function. The neuron population membership and neuron firing rate latent variables depend on the following generative distribution parameters: The categorical population membership probability ($\pi_{a,l}$) and the gamma firing rate parameters ($\alpha_{a,l}, \theta_{a,l}$), respectively.

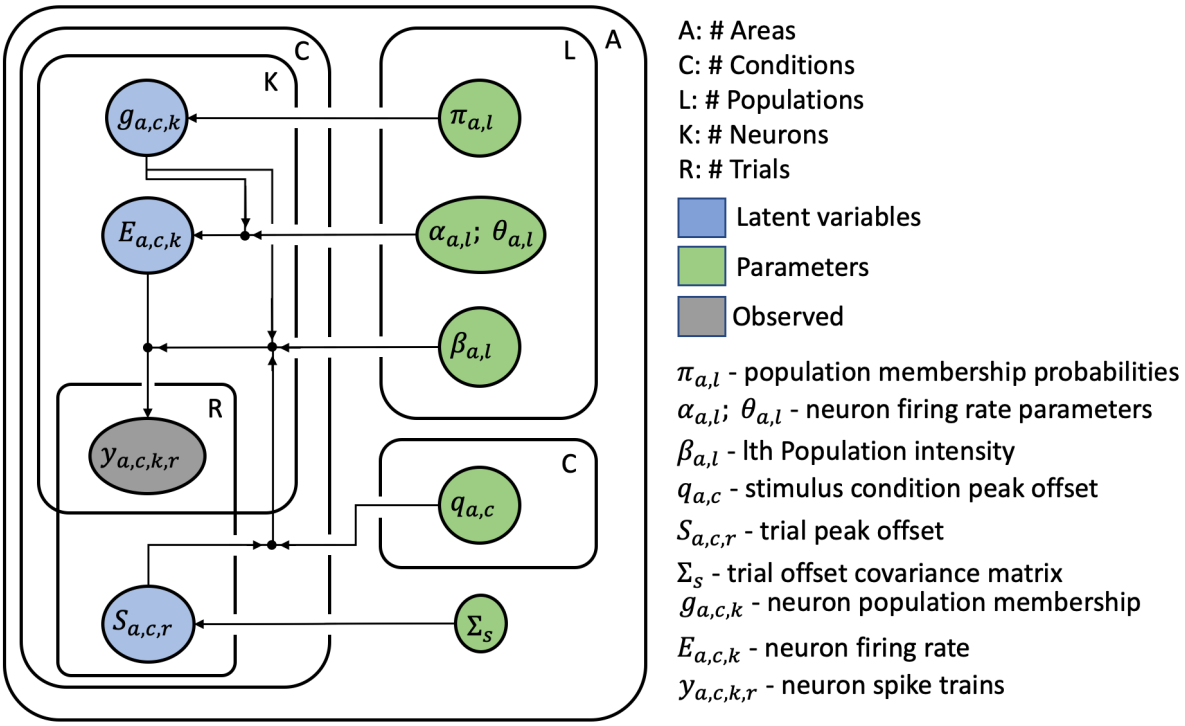

Figure 3.5: Graphical display of the Mixture of Dependent Poisson Point Processes (MDoP3). The observed data are the neuron spike trains $y_{a,c,k,r}$, observed for neuron k in area a on trial r for stimulus configuration c. A neuron's spike train $y_{a,c,k,r}$ depends on its intensity function, which is determined by its membership, $g_{a,c,k} \in \{1, \ldots, L\}$ to one of L template population firing rates $\beta_{a,l}$ in area a. This is combined with the neuron's firing rating rate over the course of each trial $E_{a,c,k}$, the peak offsets $q_{a,c}$ for condition c in area a, and the value of the covarying features $s_{a,c,r}$ for trial r, to determine the neuron's intensity function.

While direct Maximum Likelihood Estimation (MLE) is possible, computing the derivatives of the log-likelihood with respect to the parameters leads to a complicated expression that is analytically intractable, thus requiring

all parameters to be estimated numerically. On the other hand, a fully Bayesian approach for a model and for datasets of this size will require a significant amount of compute time to fit [Olarinre et al.]. We therefore opt for Expectation Maximization (EM) [Dempster et al., 1977] as our inference algorithm. As with direct MLE, the first step is to compute a marginal likelihood over the latent variables, which are the subpopulation membership for each neuron, the average spike count for each neuron over a single trial, and the peak burst times for a single trial ($g_k, E_k,$ and S_{r_c} respectively in equation 3.4). However, instead of maximizing the log marginal probability, as is done in direct MLE, EM maximizes a surrogate objective, the marginal log probability, where the marginalization is over the posterior distribution of the latent variables. The algorithm operates by iterating between an E step, where the posterior distributions of the latent variables are computed while keeping the current parameter estimates fixed, and an M step, where the marginal log probability over the previously computed posterior distribution is maximized over the likelihood parameters while keeping the posterior parameters fixed. We repeat these iterations until convergence. The EM objective is lower bound to the MLE objective, and under mild regularity conditions, will converge to the value of the true objective at the MLE estimates [McLachlan et al., 2004].

E-Step: Following EM convention, we derive the E step by computing the posterior density of the latent variables as follows: Let $\mathbf{Z}$ represent the set of latent variables, which are the average spike count for a neuron $\{E_k\}$, the neuron subpopulation membership $\{g_k\}$ and the trial peak burst time offsets $\{s_{r_c}\}$. Let $\mathbf{D}$ represent the spike train data $\{t_{k_c,r_c}\}$, K_c and R_c represent the number of neurons and trials respectively in condition c, and $\mathbf{\Phi}$ represent the distribution parameters, we have the following expression for posterior density:

$$P\left(\mathbf{Z}|\mathbf{D};\mathbf{\Phi}\right) =$$
$$\prod_c P\Big(\{E_{k_c}\}_{K_c}, \{g_{k_c}\}_{K_c}, \{s_{r_c}\}_{R_c} \big| \{t_{k_c,r_c}\}_{K_c,R_c}\Big) =$$
$$\prod_c \left[\prod_{k_c} P\Big(E_{k_c}|g_{k_c}, \{s_{r_c}, t_{k_c,r_c}\}_{R_c}\Big) \prod_{k_c} P\Big(g_{k_c}|\{s_{r_c}, t_{k_c,r_c}\}_{R_c}\Big) \prod_{r_c} P\Big(s_{r_c}|\{t_{k_c,r_c}\}_{K_c}\Big)\right]$$

with the individual posterior expressions given by

$$\begin{aligned} P\Big(E_{k_c}|g_{k_c}, \{t_{k_c,r_c}\}_{R_c}, \dots\Big) &= \frac{P\big(\{t_{k_c,r_c}\}_{R_c}|E_{k_c}, g_{k_c}, \dots\big)\ P\big(E_{k_c}|g_{k_c}\big)}{\mathbb{E}_{E_{k_c}|g_{k_c}}[P(\{t_{k_c,r_c}\}_{R_c}|E_{k_c}, g_{k_c}, \dots)]} \\ &= \big(R_c+\theta_l\big)^{\sum_{r,t} y_{k,r,t}+\alpha_l} \frac{E_{k_c}^{(\sum_{r,t} y_{k,r,t}+\alpha_l)-1}}{\Gamma(\sum_{r,t} y_{k,r,t}+\alpha_l)} \exp\big(-E_{k_c}(R_c+\theta_l)\big) \end{aligned}$$

which is the pdf of Gamma$\Big(\sum_{r,t} y_{k,r,t}+\alpha_l,\ \ R_c+\theta_l\Big)$, and

$$\begin{aligned} P\Big(g_{k_c}\big|\{t_{k_c,r_c}\}_{R_c}, \dots\Big) &= \frac{\mathbb{E}_{E_{k_c}|g_{k_c}}[P(\{t_{k_c,r_c}\}_{R_c}|E_{k_c}, g_{k_c}, \dots)]\ P(g_{k_c})}{\mathbb{E}_{g_{k_c}}\big[\mathbb{E}_{E_{k_c}|g_{k_c}}[P(\{t_{k_c,r_c}\}_{R_c}|E_{k_c}, g_{k_c}, \dots)]\big]} \\ &= \frac{p_{k_c,l}}{\prod_{r_c}\prod_t y_{k,r,t}!} \frac{\theta_l^{\alpha_l} \prod_{i=1}^{\sum_{r,t} y_{k,r,t}} (\alpha_l+i-1)}{\big(R_c+\theta_l\big)^{\sum_{r,t} y_{k,r,t}+\alpha_l}}\ \pi_l \left[\sum_l \frac{p_{k_c,l}}{\prod_{r_c}\prod_t y_{k,r,t}!} \frac{\theta_l^{\alpha_l} \prod_{i=1}^{\sum_{r,t} y_{k,r,t}} (\alpha_l+i-1)}{\big(R_c+\theta_l\big)^{\sum_{r,t} y_{k,r,t}+\alpha_l}}\ \pi_l\right]^{-1} \end{aligned}$$

which is a probability density over possible functional unit membership. The detailed derivations for these posteriors are included in appendix B. The posterior for the peak time offset terms s_{r_c} terms are analytically intractable, but we can approximate them using a Gaussian family for the posterior distribution of these latent variables. In addition, the marginal log-likelihood over this term is also analytically intractable. To address these, we apply stochastic variational inference techniques [Blei et al., 2017, Jordan et al., 1999] to jointly learn the likelihood parameters as well as the variational posterior parameters. Under the stochastic variational inference variant of the EM algorithm, the parameters of the variational Gaussian posterior are also maximized in each M step of the algorithm, as opposed to keeping them fixed in the M step as with classical EM.

M-step: For the M step of the algorithm, we compute the expected log-likelihood for the joint distribution model

as follows:

$$\sum_c \sum_{k_c} \sum_{r_c} \sum_t \mathbb{E}_{E_{k_c}, g_{k_c}, s_{r_c} | \{t_{k_c, r_c}\}_{K_c, R_c}} \log P\big(\{t_{k_c, r_c}\} | \lambda(E_{k_c}, g_{k_c}, s_{r_c}, t)\big) +$$
$$\sum_c \sum_{k_c} \mathbb{E}_{E_{k_c}, g_{k_c}, s_{r_c} | \{t_{k_c, r_c}\}_{K_c, R_c}} \log P\big(E_{k_c} | g_{k_c}\big) +$$
$$\sum_c \sum_{k_c} \mathbb{E}_{E_{k_c}, g_{k_c}, s_{r_c} | \{t_{k_c, r_c}\}_{K_c, R_c}} \log P\big(g_{k_c}\big) +$$
$$\sum_c \sum_{r_c} \mathbb{E}_{s_{r_c} | \{t_{k_c, r_c}\}_{K_c, R_c}} \log \frac{P\big(s_{r_c}\big)}{Q\big(s_{r_c}\big)}$$

We include the full derivation in the appendix B. After dropping all the terms that are constant in the required parameters, as well as the terms which have closed form solutions, we obtain the following expression as the objective we are maximizing:

$$\mathcal{L} = \frac{1}{N} \sum_c \sum_{k_c} \sum_{r_c} \sum_l \sum_n w_{k_c, l}$$
$$\Bigg(\sum_t y_{k,r,t} \; \beta^+_{l, s^{(n)}_{r_c}, q^+_c, t} - \big(\sum_t y_{k,r,t} \big) \log \sum_t \exp(\beta^+_{l, s^{(n)}_{r_c}, q^+_c, t}) +$$
$$\frac{1}{R_c} \Big[\alpha^+_l (\log \theta_l + b_{k_c, l}) - \log \Gamma(\alpha^+_l) \Big] -$$
$$\frac{1}{K_c} \left[\frac{1}{2} \Big(\log \det(\Sigma^+) + s^{(n)\intercal}_{r_c} \Sigma^{+-1} s^{(n)}_{r_c} \Big) - \frac{1}{2} \sum_d \Big(\log \sigma^{+2}_d + \big(\frac{s^{(n)}_{r_c, d} - \mu_{r_c}}{\sigma_d} \big)^2 \Big) \right] \Bigg)$$

where $w_{k_c, l}$ is the membership probability of neuron k_c to functional unit l. As previously mentioned, we alternate between the E step and the M step, taking a single gradient step in each iteration of the M step until convergence.

3.2.4 Initialization

As with mixture models in general, the MDoP3 suffers from inherent non-identifiability of the mixture component parameters, making the objective multimodal and therefore non-convex, having multiple local optima (see Murphy [2012] section 11.3). The starting point is, therefore, important in determining which solution the EM algorithm converges to, with better quality initialization leading to better solutions. To increase the probability of converging to a good local maximum, we ran the algorithm multiple times, and over a variety of initialization techniques. To obtain the best outcomes, we initialized the parameters as follows:

Neuron Memberships, (g_{k_c}) and Population Intensity Functions (β_l):
We compare 4 possible methods of initializing the functional unit intensity function, as well as the neuron memberships.

1. Method of moments (MOM) with random noise: We initialize every population intensity function within each area to the population PSTH for the entire area. We then add Poisson noise to each intensity function to introduce differences among them. As mentioned previously, running the algorithm from multiple starting locations increases its chances of finding a good local maximum, and the random noise enables multiple starts from different starting points across multiple runs of the algorithm. The neurons in each area are given uniform weight (probabilities) for each intensity function.

2. Dynamic time-warping (DTW) clustering: We apply the dynamic time-warping clustering algorithm [Bemdt, 1994, Wang et al., 2018] to the neuron PSTHs in each area. The algorithm operates similarly to k means clustering, where a random initial centroid is selected, and the neurons in each region are iteratively assigned to a centroid. It differs from K means in that the distance metric used is the dynamic time warping distance matric. Using this, we obtain in initial value for the population intensity functions, as well as an initial neuron assignment.

3. Fully random initialization: The population intensity function for each functional unit, as well as the initial neuron assignments, are chosen randomly.

4. Zero initialization: The population intensity function is initialized to zero, and the neurons in each area are given uniform weight for each intensity function.

For each of these methods, Figure 3.6 shows the distribution of log-likelihoods for 20 random initializations (in the case of MOM, DTW, and random initialization). We see that the method of moments and dynamic time-warping initializations generally do better than the zero and random initializations. In general, the model seems to be fairly robust to the choice of initialization given enough training time.

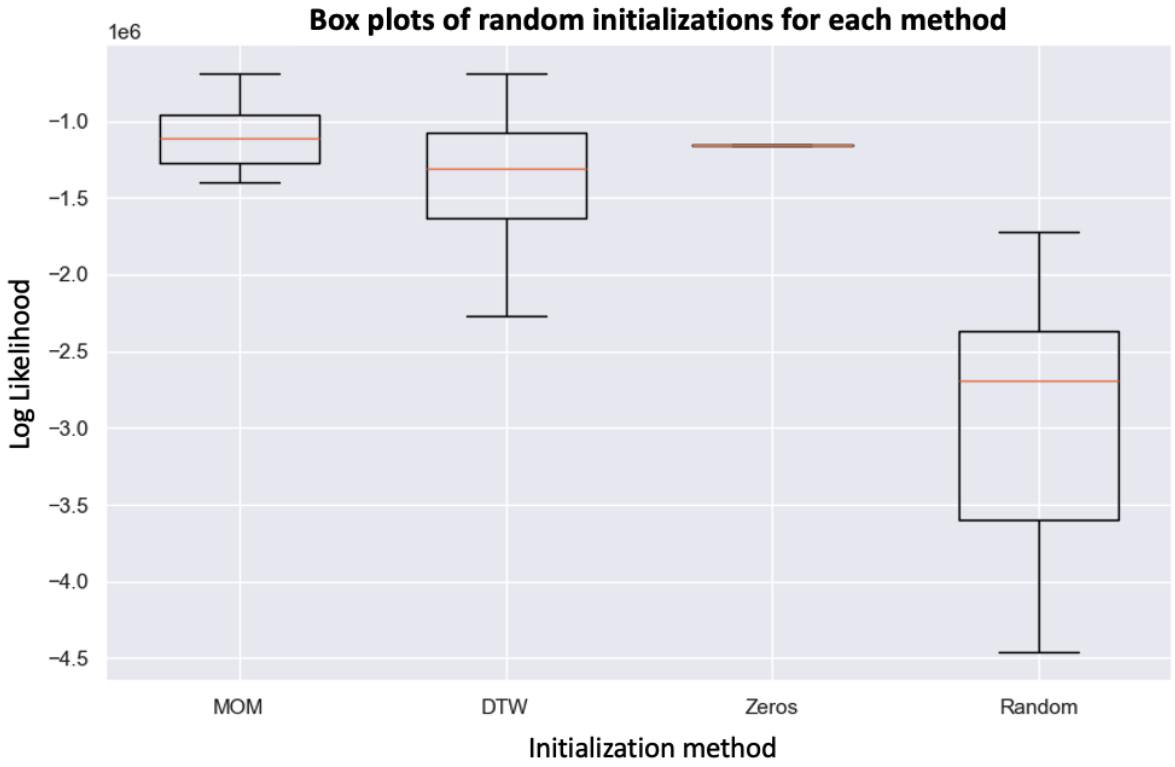

Figure 3.6: The figure compares the log-likelihood obtained by the various methods of initialization of the population intensity function and the neuron membership described earlier. They are Method Of Moments (MOM), Dynamic Time Warping (DTW), Zero initialization and fully random initialization. For each method, we initialized our desired parameters over 20 random initializations. We ran the zero and fully random initializations for 40,000 gradient steps. The zero initialization involves no randomness, and so only has one entry. We ran the MOM and DTW for 10,000 gradient steps, as they appeared to converge much faster than the other two methods of initialization. Results show that the fully random initialization tends to perform worse than the other three, but the model is fairly robust to the choice of initialization, although MOM and DTW tend to converge faster than zero and random initializations

Negative Binomial parameters dispersion (α) and rate (θ) parameter:
Having initialized the populations belonging to the various functional units, we then initialize the parameters of the observed spike count distribution, α and θ, using the method of moments. For each functional unit, we compute the sample average spike count, as well as their sample variance, and initialize the dispersion and rate parameters as follows:

$$\hat{\alpha}_{init} = \frac{\bar{X}^2}{V - \bar{X}}, \quad \hat{\theta}_{init} = \frac{\hat{\alpha}_{init}}{\bar{X}}$$

Stimulus configuration peak burst time offsets (q_c), trial peak burst time offsets (s_{r_c}), and trial peak burst time offsets covariance (Σ_s):
We initialize both the stimulus configuration and trial peak offsets to zeros. We initialize the trial offset covariance matrix to a diagonal matrix, where the diagonal entries are determined by the width of the time-warping window.

This initialization strategy produces good results on the simulated data. When applied to real data, we use multiple

random restarts, and evaluate the log likelihood function for the different initializations, in order to avoid local minima.

3.2.5 Software Implementation

As part of the work done in this thesis, we implemented the previously described model in Python. Considering that the model was developed to be applied to large datasets consisting of thousands of neurons, we implemented the model to minimize the runtime required to process such large datasets. We were able to greatly reduce the model's runtime by:

- **Leveraging PyTorch for GPU Acceleration:** We implemented the model using PyTorch, so as to take advantage of its efficient tensor operations and automatic differentiation capabilities on both CPUs and GPUs. Offloading computationally intensive tasks, such as gradient calculations and matrix operations, to a GPU resulted in significant performance gains.
- **Vectorized Operations:** We vectorized all core computations, so as to exploit PyTorch's highly optimized backend for tensor operations. This ensured that the model performed efficiently by avoiding explicit Python loops.
- **Sparse Matrix Optimization:** We made use of sparse representations for operations involving spike trains and precision matrices, minimizing memory usage and computational overhead. PyTorch's support for sparse tensors was particularly useful in this context.
- **Profiling and Optimization:** Our implementation was extensively profiled using tools such as PyTorch's Profiler and NVIDIA Nsight Systems. Identified bottlenecks were optimized to ensure efficient usage of computational resources. For example, we replaced computationally expensive for-loops with optimized library functions.
- **Logging and Checkpointing:** We created detailed plots and logs at intervals throughout the training process. This facilitated an in-depth view of the model performance and mode of operation, not only at the end of the training process but throughout the process. It also proved indispensable for debugging. Furthermore, we also saved snapshots of the model at intervals, allowing for continued training, or probing of the model at any point during the training process.
- **Parallel Processing:** We utilized Python's multiprocessing library to parallelize parts of the model fitting process, particularly the logging, checkpointing, and other non-core computations.
- **Automatic Differentiation:** To compute gradients required for the M step of the EM algorithm efficiently, we leveraged PyTorch's automatic differentiation.
- **Customizable Model Configuration:** The implementation includes options to tune hyperparameters, select initialization strategies, and adjust the number of functional units (L) dynamically, based on the dataset's size and complexity.

Our model implementation is well suited for large datasets and can run in a practical timeframe while maintaining numerical stability and accuracy. For example, our model is able to the same three areas analyzed in [Chen et al., 2022], using the IPFR, and obtained similar results. However, where the IPFR takes ten hours to fit the three areas, MDoP3 took 20 minutes on GPU for the same dataset. The source code and documentation for this implementation are publicly available at the repository: https://github.com/Tolani-O/MDoP3.

3.3 Results

3.3.1 Model performance on simulated data

In order to verify the ability of the model to recover the desired parameters, we simulated, according to the model described above, data for A = 3 brain areas, with C = 40 stimulus conditions with R = 15 trials for each stimulus condition. For each area, we simulated neuron spike trains of length T = 200, for all neurons and trials, using predefined population intensity functions, each representing a distinct homogeneous population of neurons. Different numbers of unique functions were used in each area: 1, 2, and 3 unique functions in the corresponding area. The predefined population intensity functions used in our simulations, as well as the functions recovered by the model are shown in Figure 3.7. We fit the model to 50 simulated datasets as described above. Figure 3.8,

shows the log-likelihood trajectories for each simulation during training, as the model estimates converge to the ground truth parameters. Each trajectory is vertically shifted by its true log-likelihood. The vertical shift translates all trajectories to have a ground truth at 0. The grey traces correspond to the individual simulations, and the black trace represents the average of all the trajectories.

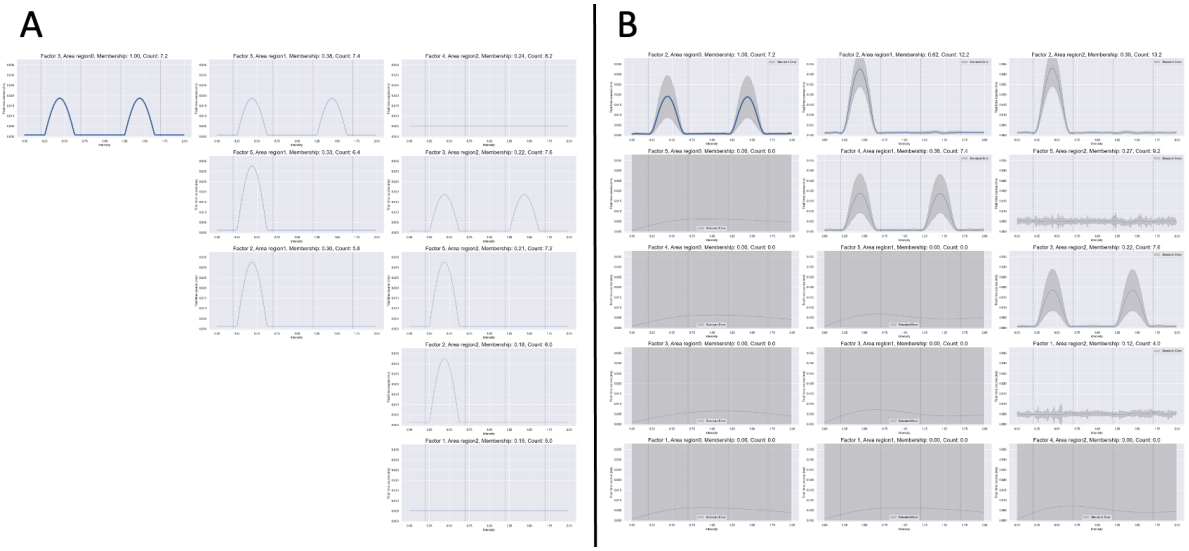

Figure 3.7: The MDoP3 model is able to identify the minimum number of factors needed to fit a particular dataset, and it discards the rest. In this figure, each column represents a different simulated region. **A**: The ground truth functional unit intensity functions used to simulate the data. The second and third regions (labeled 1 and 2) have 1 and 2, respectively, pairs of redundant factors. **B**: The model is able to recover the ground truth factors, and only utilizes the number it needs to avoid redundancy. The exception is when there is data from a constant intensity function, in which case the model assigns no cost for redundancy. **B** also shows the 95% confidence interval for the population intensity functions recovered by the model.

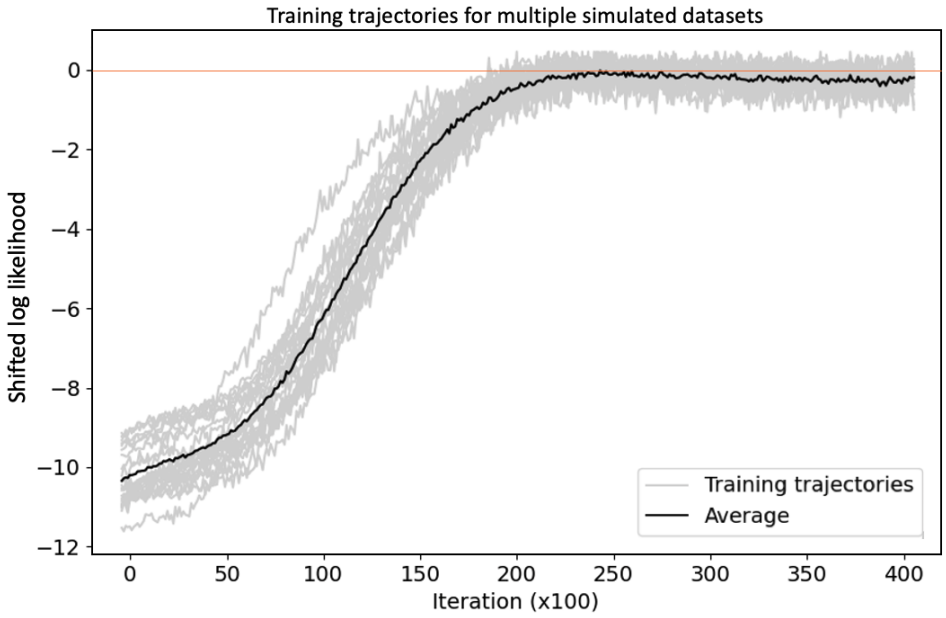

Figure 3.8: Training trajectories for the log-likelihood, evaluated over 50 data simulations, vertically shifted by its true log-likelihood. The vertical shift translates all trajectories to have a ground truth (population) log-likelihood at 0, depicted by the orange horizontal line. The time courses for simulations are represented by the gray traces, the average time course is represented by the black trace. The x-axis shows the number of gradient steps taken (inner iterations) across all EM iterations.

3.3.2 Robustness

We investigate the robustness of the model by fitting it under a variety of assumptions with respect to data generation. As previously mentioned, the model is robust to over-specification of the number of functional units to discover, as it can automatically identify how many functional units are needed for the problem and fit that many (assuming distinct functional unit firing rate profiles). Furthermore, we tested the ability of the model to recover the ground truth parameters under the following assumptions:

- Overspecification to the number of functional units to be learned
- Overdispersion/underdispersion of data
- Varying levels of sparsity on the peak time offsets precision matrix

Overspecification to the number of functional units to be learned: We have shown in Figure 3.7 that the MDoP3 model is able to accomplish model selection by determining the minimum number of functional units needed to fit a given dataset, even when the number of functional units is different for each region. This substitutes the requirement to pre-specify the exact number of clusters you expect to learn with the more relaxed requirement of specifying the maximum number of clusters you might expect to learn, barring any computational constraints, and mitigates the concern for underspecification. As, one might expect, the effect of underspecification of the number of functional units is learning population firing rate functions that combine two or more populations. This is illustrated in Figure C.1.

Overdispersion of data: As mentioned in section 3.2.1, our specified model assumes a Negative Binomial distribution of the population PSTH. It is therefore well suited to cases where the variance of the counts are higher than the mean. We verified this assumption on the datasets to which we applied this model (the Allen Institute Observatory dataset), and we indeed observed that in all areas of all mice, the average population spike count on trial is in fact less than the variance of the counts. In Figure 3.9, we show the ratio of the variance to the mean number of spikes per trial in all areas in an example mouse. We see here that the values are all positive, which validates our assumptions. Although the model will also fit the data in the under-dispersed scenario, it is not able to consistently recover the ground truth parameters used in simulations to generate the data.

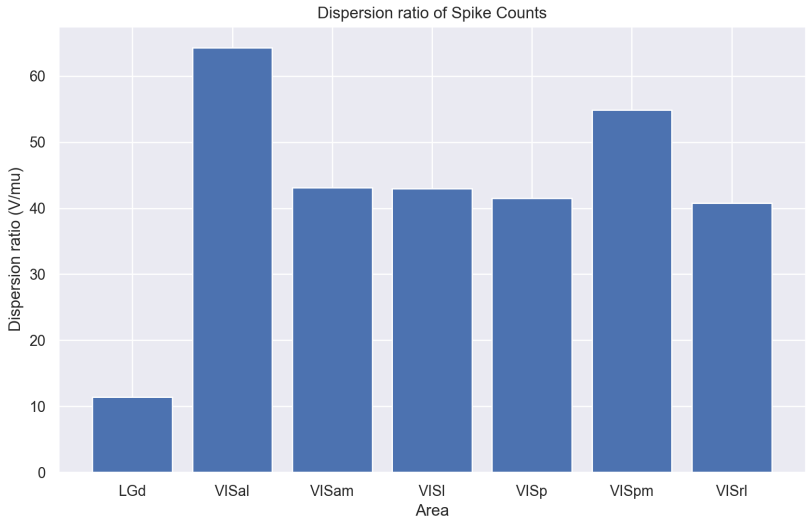

Figure 3.9: Ratio of the variance to mean of the spike count distribution in each area for a single mouse in the Allen Institute dataset. All values are greater than 1 (with some many times greater than 1) indicating that, within each region, the spike counts are overdispersed

Varying levels of sparsity on the peak time offsets precision matrix: As stated in section 3.2.2, the strength of functional connectivity (edge) between pairs of functional units (nodes) is quantified using the trial to trial correlations in peak burst times among the functional units. Assuming these peak times across all functional units are described by a multivariate Gaussian random variable, our model learns the inverse covariance or precision matrix for this joint distribution. We therefore investigate the ability of the model to recover the ground truth

precision matrix under different assumptions about the conditional independence between functional units, that is, assuming varying degrees of sparsity in the Gaussian precision matrix. The results of this investigation are shown in Figure 3.10. We assumed 20%, 60% and 80% sparsity, and in each case, we generated 20 random precision matrixes (by generating a dense matrix and randomly zeroing out a percentage of the entries). We correspondingly generated 20 random spike train datasets, and fit our model to each one. We logged the mean square error (MSE) trajectories of the learned precision matrix and the ground truth matrix for each, and finally computed their pointwise standard errors, shown in the figure. We see that under each sparsity assumption, the mean square errors converge to zero, demonstrating that the model learns an approximation to the true matrix.

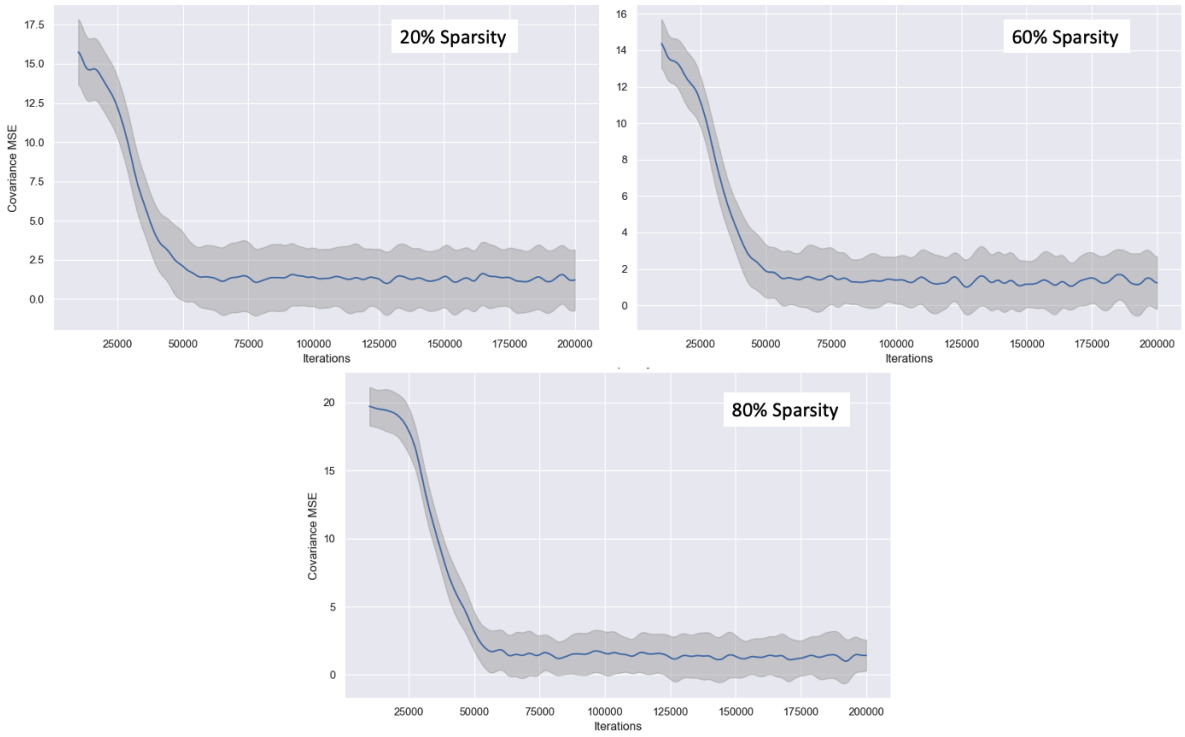

Figure 3.10: Mean Squared Error (MSE) between the model recovered precision matrix and the ground truth precision for 3 different degrees of sparsity in the precision matrix. Sparcity in the precision matrix denotes conditional independence between the functional units (nodes). For each percent sparsity, we randomly generated 20 precision matrices, while keeping track of the MSE trajectory over the course of training. We then averaged the MSE trajectories for the 20 datasets, as well as the pointwise standard errors. We see here that in each case, the MSE approaches zero. Exact convergence to zero requires large amounts of data, and our simulation was on the scale of the experimentally collected data (see 2.1). We anticipate closer to 60% sparsity in our application, since not every functional unit is conditionally correlated with every other unit.

3.3.3 Performance on real data

As stated earlier, we developed this method with the aim of applying it to real data, in order to infer the functional units within each brain area, as well as the strength of the interactions between them. The model is able to identify connections between functional units with a diversity of firing rate response profiles, by contrast to prior methods, such as the IPFR and 3-step method, which are only able to identify connections between specific functional units. The model is also able to run very fast on large amounts of data. We applied our method to the Allen Institute Brain Observatory dataset described in section 2.1. The dataset contains recordings from multiple brain areas of mice presented with a variety of visual stimuli. Specifically, we analyzed data for 40 configurations of the drifting gratings stimulus, from six recorded cortical visual areas (V1, LM RL, AL, AM, PM), and one thalamic nucleus (LGd). The output from fitting this model is shown Figure C.2. Figure 3.11 shows the population intensity functions that were learned for 3 visual areas: V1, LM and AL. We find some of these intensity functions exhibit the double-peaked bursting response we expect from neurons reacting to visual stimulus, while others exhibit a single peaked response, or no bursting response at all. The functional subpopulations also have some degree of marginal correlations between them, as seen in Figure 3.12A. The partial correlation matrix is, however, much

more sparse (Figure 3.12B), indicating that some functional units are conditionally independent, given the other units. We observe from this figure that the functional interaction in region AL (VISal, first column) is primarily due to the first and third functional units, in LM (VISL, second column) is primarily due to the second and fifth functional units, and in V1 (VISp, third column) is primarily due to the third, fourth and fifth functional units.

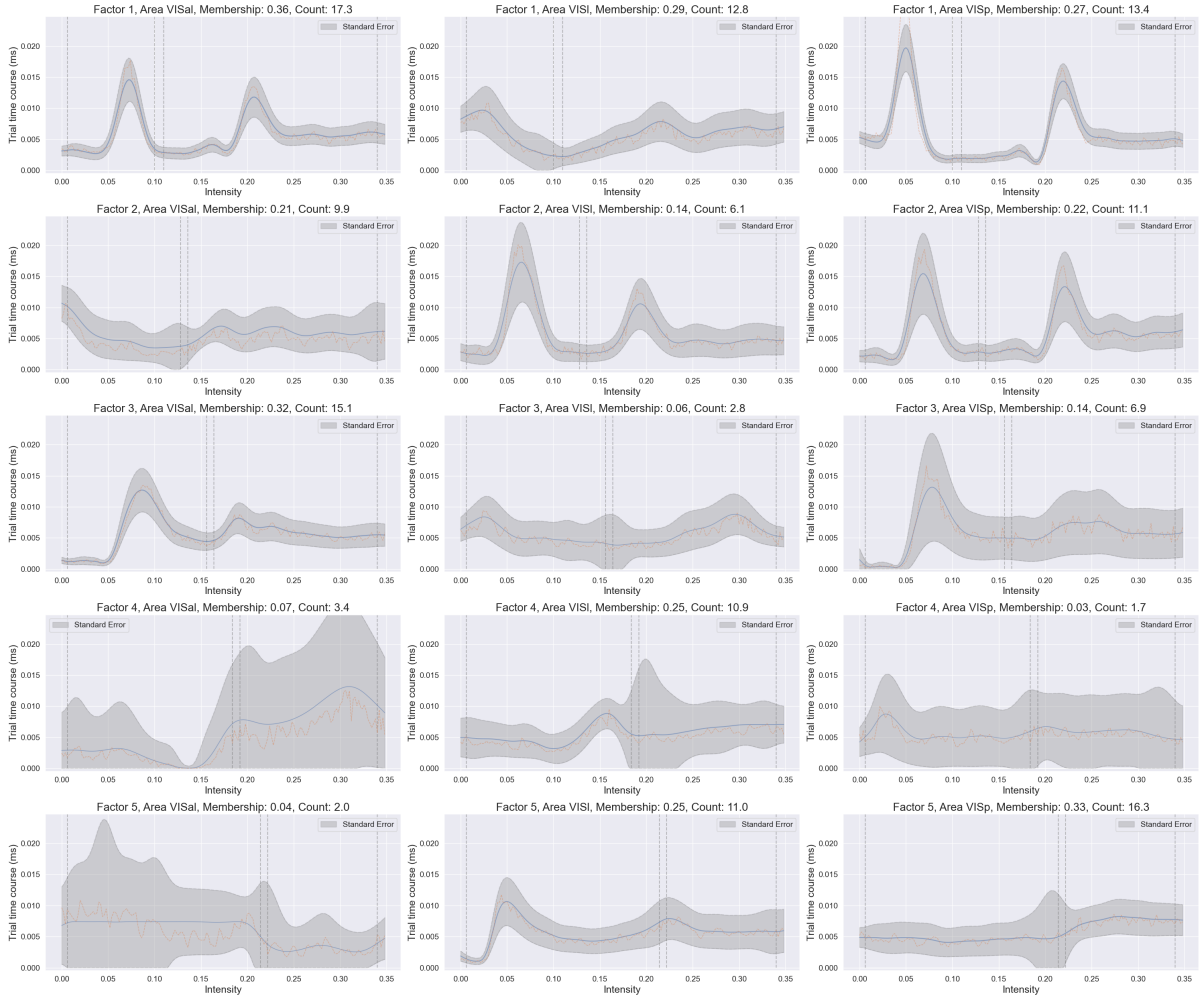

Figure 3.11: The figure shows the population intensity functions learned by the MDoP3, along with their 95% confidence interval. We observe the characteristic stimulus-evoked dual peaked response in several of the learned populations.

Effects of time warping

As mentioned in section 3.2.2, time warping is required to align the observed population spike trains across trials. It is also the means by which we learn the covariance structure of the trial-to-trial peak time shifts for the population intensity function of each function unit. Figure 3.15 shows the time-warping function, across each trial (left figure) and the distribution of the peak burst times across trials (right figure). We see here that the marginal distribution of the trial peak times is well approximated by a Gaussian distribution. In addition, figure 3.13 shows the difference between the population PSTH without time warping, and with time warping. We see here that when the spike trains are summed up across trials without first aligning the trial, the peak bursting behavior is attenuated by the noise from trial-to-trial peak shifts. By contrast, When the time warping functions are applied before summing, we see the peak burst times clearly visible, which matches up with the learned population intensify function. Finally, figure 3.14 shows an example of the model output when we do not account for the trial-to-trial peak time shifts. We observe the broad bursting behavior, which occurs due to the attempt of the model to account for the peak time shifts in the intensity functions.

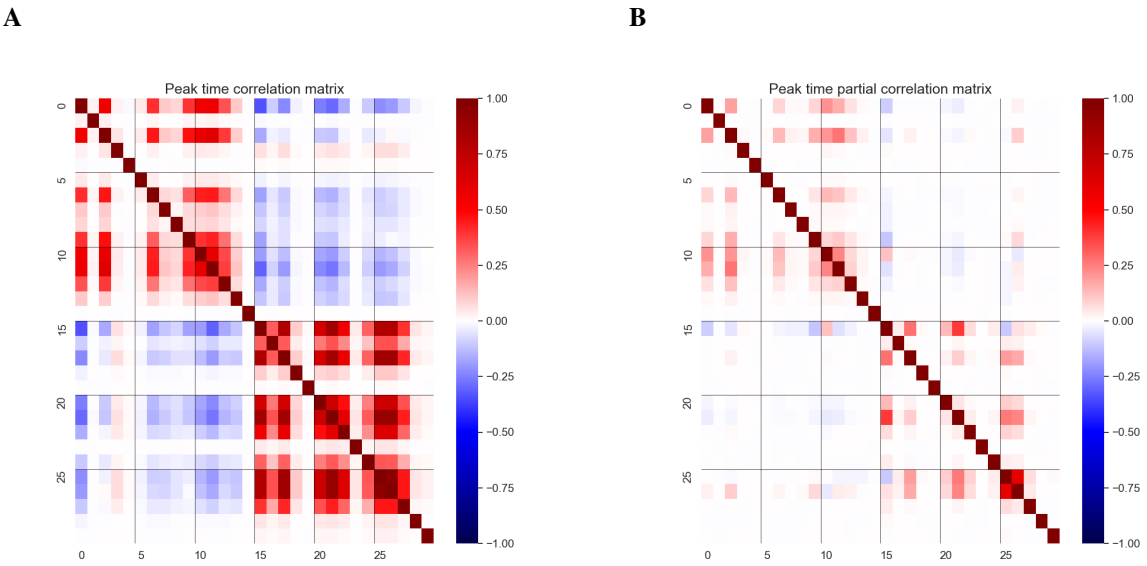

Figure 3.12: The figure shows the marginal and full partial correlations between the functional peak burst times of five functional units, denoted by each pixel, across each of three anatomical brain regions, denoted by each square (AL, LM and V1 respectively), for two burst peaks, denoted by each quadrant. In both matrices, the top left quadrant shows the correlations between the first peaks of all functional units across areas, the top right quadrant shows the correlations of the first and second peaks of the functional units across brain areas, and the bottom right quadrant shows the correlations between the second peaks of the functional units across the three areas. We observe that both peaks tend to have a positive correlation with themselves, and a negative correlation with the other peak. **A** The marginal correlations between both peaks of the functional units. **B** The partial correlations between both peaks of the functional units. The partial correlation matrix is, however, much more sparse, indicating that some functional units are conditionally independent, given the other units. We observe from this figure that the functional interaction in region AL (VISal, first column) is primarily due to the first and third functional units, in LM (VISL, second column) is primarily due to the second and fifth functional units, and in V1 (VISp, third column) is primarily due to the third, fourth and fifth functional units.

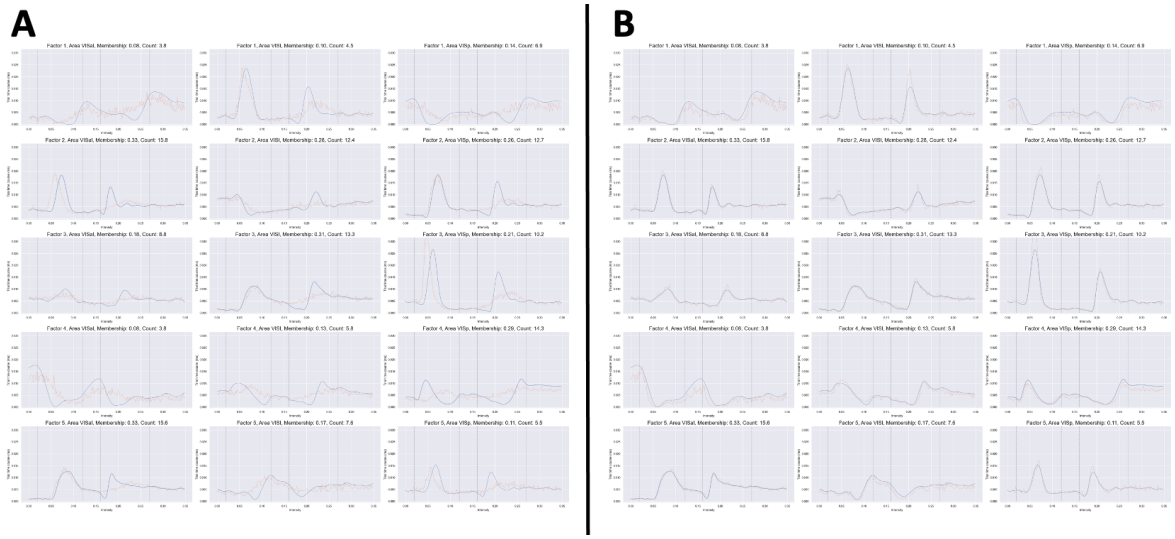

Figure 3.13: Aligning effect of time warping in fitting the population intensity functions to the spiking neuron data. In both figures, we see the model-fitted firing rate functions (blue trace) and the trial-averaged population PSTHs (orange trace). In **A**, we show the data as a trial averaged population PSTH, obtained without the trial-by-trial alignment from time warping. It is clearly misaligned with the firing rate function. In **B**, the PSTH is computed by first aligning the population PSTHs over the trials, and then averaging. In this case, the PSTH and the firing rate function are aligned. The model learns the peak time offsets for each trial, and accounts for these offsets using the time-warping function. Figure 3.14 shows an example of the learned population intensity functions when we do not account for the temporal misalignment.

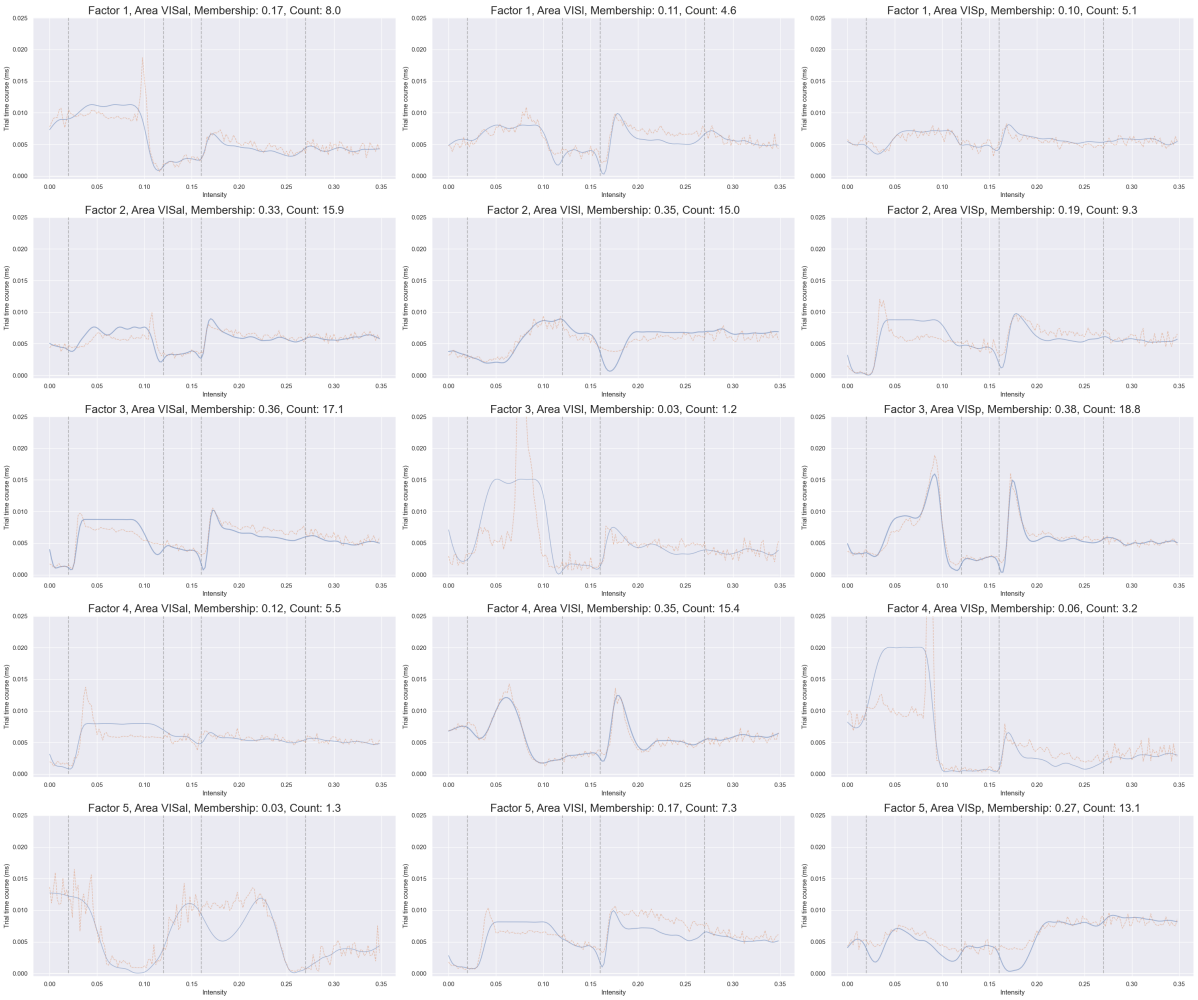

Figure 3.14: The figure shows an example of the learned population intensity functions when we do not account for the temporal misalignment. As a result, the intensity function itself has to account for the misalignment, leading to the flat-topped peak on many of the intensity functions.

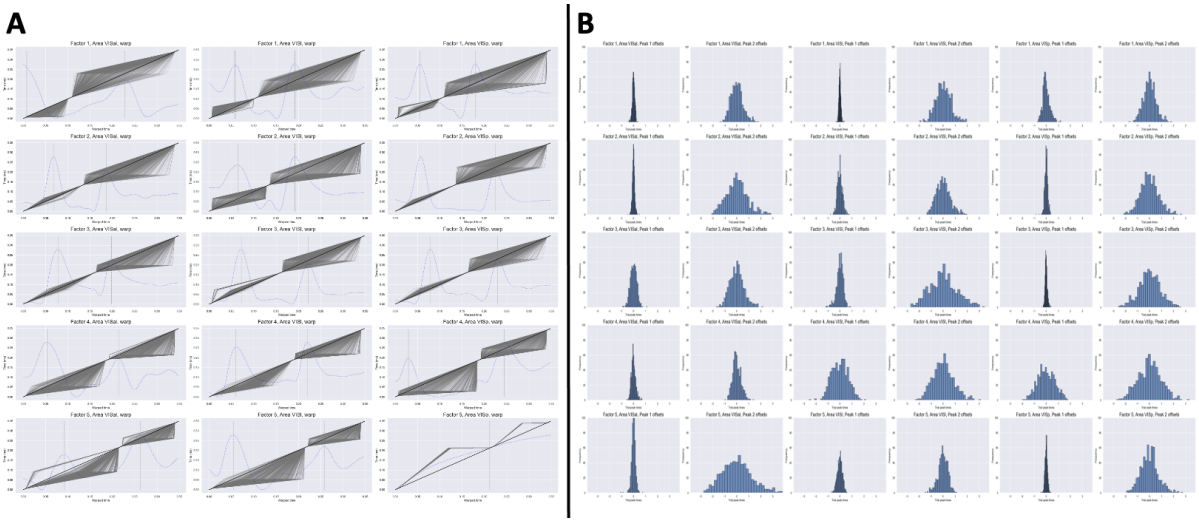

Figure 3.15: **A**: For each region (columns) and each functional unit (rows), we show the piecewise linear time warping function for all trials (grey traces). **B**: The peak time offsets for each individual trial, for the area, functional unit and peak shown in the subplot title. The distribution of the trial-to-trial peak offsets is well approximated by a Gaussian.

3.4 Discussion

In this chapter, we presented a framework for learning latent functional units, as well as their interactions, from noisy neural time-series data. Our proposed model, the Mixture of Dependent Poisson Point Processes (MDoP3), integrates a clustering mechanism for identifying functional units and a probabilistic graphical model to infer their interactions, while accounting for trial-to-trial variability in the population stimulus-response behavior using a time-warping function. We tested our model on simulated data, and thus demonstrated that our model is able to recover the ground truth parameters of the simulated datasets, up to permutations of the mixture model components ions. We also showed that our model is robust to choices of initial starting points, the sparsity structure of the trial peak time precision matrix, and to overspecification of the number of functional units to search for, which we highlight as a strength of the model. The model is optimized for speed and scalability. In this work, we were able to fit the model to a dataset consisting of hundreds of neurons recorded from 7 brain regions, over 15 trials of 40 stimulus presentations. Finally, we applied the MDoP3 model to the Allen Brain Observatory dataset and were able to obtain functional units that exhibited the response and connectivity patterns consistent with those observed in Chen et al. [2022], as well as others not observed by the previous work.

The goal of this work is to advance the study of functional connectivity in computational neuroscience by providing a unified and scalable framework for identifying interacting brain regions, together with their interactions. We accomplish this using a probabilistic framework that identifies the brain areas and captures their interactions and implementing this using modern computational frameworks that enable scaling to large datasets.

Part 3

Conclusion

Chapter 4

Conclusion

The work in the thesis is aimed at advancing our understanding of the functional connections between neuron populations, with a specific focus on interactions at fine temporal scales, while keeping the accompanying computational costs to a minimum. To this end, we advanced two key methods to estimate these functional connections using statistical models: a 3-step method for estimating functional connections, and a novel probabilistic graphical modeling framework, called the Mixture of Dependent Poisson Point Processes (MDoP3). This work addressed critical challenges in analyzing large-scale neuronal datasets, including noise, data variability, and computational efficiency. The findings underscore the utility of these approaches in revealing functional relationships within and across populations of neurons in the brain. Some key contributions of this thesis are:

Scalable and efficient models for functional connectivity: Both the 3-step method and the MDoP3 provide tools to obtain robust estimates of connectivity between brain regions, as measured by correlations in stimulus response latencies. The 3-step method is fast to implement and simple to understand. The MDoP3 is more comprehensive and flexible, to suit a wide range of use cases. Furthermore, it makes use of time-warping techniques to align neuronal spiking data in order to better capture the inherent variability and inter-regional dependencies in neuronal population bursts. Both methods can be scaled to large datasets to obtain results in a reasonable timeframe.
Understanding temporal Dynamics and Connectivity: By applying the 3-step modeling framework to experimentally recorded data, we uncovered consistent patterns in neuronal activity across experimental subjects, highlighting a possible functionally relevant pathway for visual signal processing. This method also unveiled patterns of connectivity between anatomical regions within the visual cortex, as well as between the cortex and the thalamus. This provides a basis for understanding connectivity in the cortex, as well as for answering scientific questions about the relevant functional processes taking place in the brain. The MDoP3 model was able to reveal connections between multiple neuron populations, both within and across brain regions.

In sum, this thesis meets a significant need in the analysis of neuronal population interactions, leveraging useful tools to decode complex temporal dynamics. The resulting insights have far-reaching implications, offering tools and perspectives that advance both theoretical and applied neuroscience. In so doing, this work furthers ongoing exploration into the intricacies of neural activity.

Appendices

Appendix A

Discrete time Poisson point process distribution

Spike train data is recorded in discrete time, at about 1 kHz. They may be analyzed at this resolution, or down-sampled by binning. We use a discrete-time approximation of the continuous-time model in our analysis, resulting in the following formulation:

$$\int_0^T \lambda_k(t)dt = \lim_{\Delta t \to 0} \sum_{t'=0}^{T} \lambda_k(t')\Delta t$$

Where Δt is the size of each time bin, typically about 1 millisecond. For such small values of Δt, the following approximation holds:

$$\int_0^T \lambda_k(t)dt \approx \sum_{t'=0}^{T} \lambda_k(t')\Delta t = \sum_{t'=0}^{T} \lambda_{k,t'}\Delta t = E_k \sum_{t'=0}^{T} \exp(\tau_{l,t'})\Delta t \tag{A.1}$$

The unit integral constraint on $\exp(\tau_l)$, which implies that $\int_0^T \exp\big(\tau_l(t)\big)dt = \lim_{\Delta t \to 0} \sum_{t'=0}^{T} \exp(\tau_{l,t'})\Delta t = 1$ is satisfied by assuming that

$$\exp(\tau_{l,t'}) = \frac{\exp(\beta_{l,t'})}{\Delta t \sum_{t'} \exp(\beta_{l,t'})}$$

Substituting this back into A.1, for a single time step t', we obtain the discretized neuron intensity function as

$$\lambda_{k,t'} = E_k \exp(\tau_{l,t'})\Delta t = E_k \frac{\exp(\beta_{l,t'})}{\sum_{t'} \exp(\beta_{l,t'})} = E_k p_{l,t'}$$

The discretized Poisson process joint pdf for an observed spike train $\mathbf{Y} = [y_1, \ldots, y_T]$, where we assume each bin may contain more than 1 event, is expressed as (we suppress the functional unit index l for brevity):

$$\begin{aligned}
f(y_1, \ldots, y_T) &= \prod_t \frac{1}{y_t!} \lambda_{k,t'}^{y_t} \exp(-\lambda_{k,t'}) \\
&= \prod_t \frac{1}{y_t!} (Ep_t)^{y_t} \exp(-Ep_t) \\
&= E^{\sum_t y_t} \prod_t \frac{1}{y_t!} p_t^{y_t} \exp(-E \sum_t p_t) \\
&= \frac{E^{\sum_t y_t} e^{-E}}{\Big(\sum_t y_t\Big)!} \frac{\Big(\sum_t y_t\Big)!}{\prod_t y_t!} \Big(\prod_t p_t^{y_t}\Big)
\end{aligned}$$

Which for binary time series, has the interpretation of sampling the total number of events from a Poisson distribution, and then sampling the location of the events from a multinomial.

Appendix B

Derivation of the E and M steps of the EM algorithm

E Step:

Deriving the posterior for the latent variables involves the following steps:

$$\prod_c P\Big(\{E_{k_c}\}_{K_c}, \{g_{k_c}\}_{K_c}, \{s_{r_c}\}_{R_c} \big| \{t_{k_c,r_c}\}_{K_c,R_c}\Big) = \prod_c P\Big(\{E_{k_c}\}_{K_c} \big| \{g_{k_c}\}_{K_c}, \{s_{r_c}\}_{R_c}, \{t_{k_c,r_c}\}_{K_c,R_c}\Big)$$

$$\prod_c P\Big(\{g_{k_c}\}_{K_c} \big| \{s_{r_c}\}_{R_c}, \{t_{k_c,r_c}\}_{K_c,R_c}\Big) \prod_c P\Big(\{s_{r_c}\}_{R_c} \big| \{t_{k_c,r_c}\}_{K_c,R_c}\Big)$$

$$= \prod_c \Bigg[\prod_{k_c} P\Big(E_{k_c} \big| g_{k_c}, \{s_{r_c}, t_{k_c,r_c}\}_{R_c}\Big) \prod_{k_c} P\Big(g_{k_c} \big| \{s_{r_c}, t_{k_c,r_c}\}_{R_c}\Big) \prod_{r_c} P\Big(s_{r_c} \big| \{t_{k_c,r_c}\}_{K_c}\Big)\Bigg]$$

$$P\Big(E_{k_c} \big| g_{k_c}, \{t_{k_c,r_c}\}_{R_c}, \dots\Big) = \frac{P\big(\{t_{k_c,r_c}\}_{R_c} | E_{k_c}, g_{k_c}, \dots\big)\ P\big(E_{k_c} | g_{k_c}\big)}{\mathbb{E}_{E_{k_c}|g_{k_c}}[P(\{t_{k_c,r_c}\}_{R_c} | E_{k_c}, g_{k_c}, \dots)]}$$

$$P\Big(g_{k_c} \big| \{t_{k_c,r_c}\}_{R_c}, \dots\Big) = \frac{\mathbb{E}_{E_{k_c}|g_{k_c}}[P(\{t_{k_c,r_c}\}_{R_c} | E_{k_c}, g_{k_c}, \dots)]\ P(g_{k_c})}{\mathbb{E}_{g_{k_c}}\big[\mathbb{E}_{E_{k_c}|g_{k_c}}[P(\{t_{k_c,r_c}\}_{R_c} | E_{k_c}, g_{k_c}, \dots)]\big]}$$

Note that the full marginal distribution $\mathbb{E}_{\{s_{r_c}\}}\Big[\prod_{k_c} \mathbb{E}_{g_{k_c}}\big[\mathbb{E}_{E_{k_c}|g_{k_c}}[P(\{t_{k_c,r_c}\}_{R_c} | E_{k_c}, g_{k_c}, \dots)]\big]\Big]$ is analytically intractable. We therefore resort to using a variational approximation to the posterior, using the Gaussian family of distributions $Q(s_{r_c})$. Even so, we still cannot compute the expected log-likelihood analytically, and thus we need to resort to sampling in order to compute Monte Carlo (empirical) expectations. We make an assumption that can be considered a combination of a mean-field assumption on the conditional distributions of E_{k_c} and g_{k_c} (so that they are independent of the latent variables s_{r_c}), and hard EM on the conditional distributions (so they are instead dependent on the posterior mean/mode of the latent variables s_{r_c}). We do this by replacing $\beta_{l,s_{r_c},q_c,t}$ with $\beta_{l,\mu_{r_c},q_c,t}$.
We need to evaluate the terms:

$$\mathbb{E}_{E_{k_c}|g_{k_c}=l}\big[P\big(\{t_{k_c,r_c}\}_{R_c} | E_{k_c}, g_{k_c}, \dots\big)\big] = \int_{E_{k_c}} \prod_{r_c} \prod_t \Big[\frac{1}{y_{k,r,t}!} \lambda\Big(E_{k_c}, g_{k_c}, t\Big)^{y_{k,r,t}} \exp\Big(-\lambda(E_{k_c}, g_{k_c}, t)\Big)\Big] P\Big(E_{k_c} | g_{k_c}\Big)\ dE_{k_c} =$$

$$\int_{E_{k_c}} \frac{1}{\prod_{r_c}\prod_t y_{k,r,t}!} E_{k_c}^{\sum_{r,t} y_{k,r,t}} \frac{\exp\big(\sum_{r_c}\sum_t \beta_{l,\mu_{r_c},q_c,t} \cdot y_{k,r,t}\big)}{\prod_{r_c}\big[\sum_t \exp(\beta_{l,\mu_{r_c},q_c,t})\big]^{\sum_t y_{k,r,t}}} \exp\Big(-E_{k_c} \sum_{r,t} \frac{\exp(\beta_{l,\mu_{r_c},q_c,t})}{\sum_t \exp(\beta_{l,\mu_{r_c},q_c,t})}\Big) \frac{\theta_l^{\alpha_l}}{\Gamma(\alpha_l)} E_{k_c}^{\alpha_l - 1} \exp(-E_{k_c}\theta_l$$

We let $p_{k_c,l} = \frac{\exp\big(\sum_{r_c}\sum_t \beta_{l,\mu_{r_c},q_c,t} \cdot y_{k,r,t}\big)}{\prod_{r_c}\big[\sum_t \exp(\beta_{l,\mu_{r_c},q_c,t})\big]^{\sum_t y_{k,r,t}}}$, so that

$$= \frac{p_{k_c,l}}{\prod_{r_c}\prod_t y_{k,r,t}!} \frac{\theta_l^{\alpha_l}}{\Gamma(\alpha_l)} \int_{E_{k_c}} E_{k_c}^{\sum_{r,t} y_{k,r,t}} \exp(-E_{k_c}R_c) E_{k_c}^{\alpha_l - 1} \exp(-E_{k_c}\theta_l) dE_{k_c}$$

$$= \frac{p_{k_c,l}}{\prod_{r_c}\prod_t y_{k,r,t}!} \frac{\theta_l^{\alpha_l}}{\Gamma(\alpha_l)} \int_{E_{k_c}} E_{k_c}^{(\sum_{r,t} y_{k,r,t} + \alpha_l) - 1} \exp\big(-E_{k_c}(R_c + \theta_l)\big) dE_{k_c}$$

$$= \frac{p_{k_c,l}}{\prod_{r_c}\prod_t y_{k,r,t}!}\frac{\theta_l^{\alpha_l}}{\Gamma(\alpha_l)}\frac{\Gamma(\sum_{r,t} y_{k,r,t} + \alpha_l)}{\big(R_c+\theta_l\big)^{\sum_{r,t} y_{k,r,t}+\alpha_l}}$$

$$= \frac{p_{k_c,l}}{\prod_{r_c}\prod_t y_{k,r,t}!}\frac{\theta_l^{\alpha_l}\prod_{i=1}^{\sum_{r,t} y_{k,r,t}}(\alpha_l+i-1)}{\big(R_c+\theta_l\big)^{\sum_{r,t} y_{k,r,t}+\alpha_l}}$$

Note that, similar to the case of the Poisson point process corresponding to sampling the number of spikes from a Poisson, and then sampling the bin locations from a multinomial, the marginal likelihood of the spike trains across R_c trials, given a factor and condition corresponds to sampling the number of spike counts $\sum_{r,t} y_{k,r,t}$ from a Negative Binomial, and then sampling the bin locations from a multinomial.
The Negative Binomial has parameters $r = \alpha$, and $p = \frac{\theta}{R_c+\theta}$. It therefore has mean
$\mu_{\text{NB}} = \frac{R_c\alpha}{\theta} = R_c\mu_\Gamma$
and variance
$V_{\text{NB}} = \frac{\mu_{\text{NB}}}{p} = \mu_{\text{NB}} * \frac{R_c+\theta}{\theta} = \frac{R_c\alpha(R_c+\theta)}{\theta^2}$.
The variance can also be expressed as
$V_{\text{NB}} = \mu_{\text{NB}} + \mu_{\text{NB}}\frac{R_c}{\theta} = \mu_{\text{NB}} + (R_c\sigma_\Gamma)^2$.
We can see from here that $V_{\text{NB}} \geq \mu_{\text{NB}}$, $V_{\text{NB}} \to \mu_{\text{NB}}$ as $\sigma_\Gamma \to 0$
Given this, we don't expect the model to handle underdispersion well. Since the method of moments initialization for θ depends on the difference between the empirical variance and the expectation being positive, we might not be able to do method of moments for smaller values of σ_Γ, for which is the difference might be negative.
The marginal over the factor membership is then given as:

$$\mathbb{E}_{g_{k_c}}\big[\mathbb{E}_{E_{k_c}|g_{k_c}=l}[P(\{t_{k_c,r_c}\}_{R_c}|E_{k_c},g_{k_c},\dots)]\big] = \sum_l \mathbb{E}_{E_{k_c}|g_{k_c}}\big[P\big(\{t_{k_c,r_c}\}_{R_c}|E_{k_c},g_{k_c},\dots\big)\big]\; p_{g_{k_c}}$$

$$= \sum_l \frac{p_{k_c,l}}{\prod_{r_c}\prod_t y_{k,r,t}!}\frac{\theta_l^{\alpha_l}\prod_{i=1}^{\sum_{r,t} y_{k,r,t}}(\alpha_l+i-1)}{\big(R_c+\theta_l\big)^{\sum_{r,t} y_{k,r,t}+\alpha_l}}\;\pi_l$$

Putting these all together, we have

$$P(E_{k_c}\big|g_{k_c},\{t_{k_c,r_c}\}_{R_c},\dots) \propto$$

$$E_{k_c}^{\sum_{r,t} y_{k,r,t}}\frac{p_{k_c,l}}{\prod_{r_c}\prod_t y_{k,r,t}!}\exp\big(-E_{k_c}R_c\big)\frac{\theta_l^{\alpha_l}}{\Gamma(\alpha_l)}E_{k_c}^{\alpha_l-1}\exp(-E_{k_c}\theta_l)$$

$$= \frac{p_{k_c,l}}{\prod_{r_c}\prod_t y_{k,r,t}!}\frac{\theta_l^{\alpha_l}}{\Gamma(\alpha_l)}E_{k_c}^{(\sum_{r,t} y_{k,r,t}+\alpha_l)-1}\exp\big(-E_{k_c}(R_c+\theta_l)\big)$$

and therefore

$$P(E_{k_c}\big|g_{k_c},\{t_{k_c,r_c}\}_{R_c},\dots) =$$

$$\big(R_c+\theta_l\big)^{\sum_{r,t} y_{k,r,t}+\alpha_l}\;\frac{E_{k_c}^{(\sum_{r,t} y_{k,r,t}+\alpha_l)-1}}{\Gamma(\sum_{r,t} y_{k,r,t}+\alpha_l)}\exp\big(-E_{k_c}(R_c+\theta_l)\big)$$

Note that this is a gamma distribution.
Next, we have

$$P(g_{k_c}|\{t_{k_c,r_c}\}_{R_c},\dots) =$$

$$\frac{p_{k_c,l}}{\prod_{r_c}\prod_t y_{k,r,t}!}\frac{\theta_l^{\alpha_l}\prod_{i=1}^{\sum_{r,t} y_{k,r,t}}(\alpha_l+i-1)}{\big(R_c+\theta_l\big)^{\sum_{r,t} y_{k,r,t}+\alpha_l}}\;\pi_l$$

$$\left[\sum_l \frac{p_{k_c,l}}{\prod_{r_c}\prod_t y_{k,r,t}!}\frac{\theta_l^{\alpha_l}\prod_{i=1}^{\sum_{r,t} y_{k,r,t}}(\alpha_l+i-1)}{\big(R_c+\theta_l\big)^{\sum_{r,t} y_{k,r,t}+\alpha_l}}\;\pi_l\right]^{-1}$$

For the peak time latent variable, we have the expression

$$w_{s_{r_c}} = \frac{P(s_{r_c})}{q(s_{r_c})} = \left(\frac{\prod\sigma_i^2}{\det(\Sigma)}\right)^{\frac{1}{2}}\exp\left(-\frac{1}{2}\Big[s_{r_c}^\intercal\,\Sigma^{-1}s_{r_c} - \sum_i\Big(\frac{s_{r_c,i}-\mu_{r_c,i}}{\sigma_i}\Big)^2\Big]\right)$$

M Step:

We can derive the ELBO as:

$$\log P\Big(\{t_{k,r,t}\}_{K,R,T}\Big) =$$

$$\log\left[\int_{\{s_{r_c}\}}\int_{\{g_{k_c}\}}\int_{\{E_{k_c}\}}\prod_c\Big(\prod_{k_c}\prod_{r_c}\prod_t P\big(\{t_{k_c,r_c}\}|\lambda(E_{k_c},g_{k_c},s_{r_c},t)\big)\prod_{k_c}P\big(E_{k_c}|g_{k_c}\big)\prod_{k_c}P\big(g_{k_c}\big)\prod_{r_c}P\big(s_{r_c}\big)\Big)\right]$$

$$=\log\left[\int_{\{s_{r_c}\}}\int_{\{g_{k_c}\}}\int_{\{E_{k_c}\}}\prod_c\Big(\prod_{k_c}\prod_{r_c}\prod_t P\big(\{t_{k_c,r_c}\}|\lambda(E_{k_c},g_{k_c},s_{r_c},t)\big)\prod_{k_c}P\big(E_{k_c}|g_{k_c}\big)\prod_{k_c}P\big(g_{k_c}\big)\prod_{r_c}P\big(s_{r_c}\big)\Big)\frac{P(\dots)}{P(\dots)}d\dots\right.$$

$$=\log\left[\mathbb{E}_{P(\dots)}\prod_c\Big(\prod_{k_c}\prod_{r_c}\prod_t P\big(\{t_{k_c,r_c}\}|\lambda(E_{k_c},g_{k_c},s_{r_c},t)\big)\prod_{k_c}P\big(E_{k_c}|g_{k_c}\big)\prod_{k_c}P\big(g_{k_c}\big)\prod_{r_c}P\big(s_{r_c}\big)\Big)\frac{1}{P(\dots)}\right]$$

$$\geq\mathbb{E}_{P(\dots)}\log\left[\prod_c\Big(\prod_{k_c}\prod_{r_c}\prod_t P\big(\{t_{k_c,r_c}\}|\lambda(E_{k_c},g_{k_c},s_{r_c},t)\big)\prod_{k_c}P\big(E_{k_c}|g_{k_c}\big)\prod_{k_c}P\big(g_{k_c}\big)\prod_{r_c}P\big(s_{r_c}\big)\Big)\frac{1}{P(\dots)}\right]$$

Note that for EM with exact posteriors, the posterior depends on fixed values of the model parameters, and does not factor into the maximization step. For the posterior term of the trial peak time, since we are using a variational posterior, so we keep that posterior term. We are left with the following objective

$$\sum_c\sum_{k_c}\sum_{r_c}\sum_t\mathbb{E}_{E_{k_c},g_{k_c},s_{r_c}|\{t_{k_c,r_c}\}_{K_c,R_c}}\log P\big(\{t_{k_c,r_c}\}|\lambda(E_{k_c},g_{k_c},s_{r_c},t)\big)+\sum_c\sum_{k_c}\mathbb{E}_{E_{k_c},g_{k_c},s_{r_c}|\{t_{k_c,r_c}\}_{K_c,R_c}}\log P\big(E_{k_c}|g_{k_c}\big)$$

$$+\sum_c\sum_{k_c}\mathbb{E}_{E_{k_c},g_{k_c},s_{r_c}|\{t_{k_c,r_c}\}_{K_c,R_c}}\log P\big(g_{k_c}\big)+\sum_c\sum_{r_c}\mathbb{E}_{s_{r_c}|\{t_{k_c,r_c}\}_{K_c,R_c}}\log\frac{P\big(s_{r_c}\big)}{Q\big(s_{r_c}\big)}$$

Where the last term is $-D_{kl}(Q\|P)$, the "Reverse" KL divergence. Evaluating this term by term, we have:

$$\mathbb{E}_{E_{k_c}|\{t_{k_c,r_c}\}_{R_c},g_{k_c},\dots;\theta^-}\log P\big(\{t_{k_c,r_c}\}|\lambda(E_{k_c},g_{k_c},s_{r_c},t)\big)=$$

$$\int_{E_{k_c}}\big[y_{k,r,t}\log\lambda(E_{k_c},g_{k_c},s_{r_c},t)-\lambda(E_{k_c},g_{k_c},s_{r_c},t)-\log y_{k,r,t}!\big]\;P(E_{k_c}|\{t_{k_c,r_c}\}_{R_c},g_{k_c},\dots)\;dE_{k_c}=$$

$$\int_{E_{k_c}}\big[y_{k,r,t}\log E_{k_c}+y_{k,r,t}\big(\beta_{l,s_{r_c},q_c,t}-\log\sum_t\exp(\beta_{l,s_{r_c},q_c,t})\big)-E_{k_c}\frac{\exp(\beta_{l,s_{r_c},q_c,t})}{\sum_t\exp(\beta_{l,s_{r_c},q_c,t})}-\log y_{k,r,t}!\big]$$

$$\big(R_c+\theta_l\big)^{\sum_{r,t}y_{k,r,t}+\alpha_l}\;\frac{E_{k_c}^{(\sum_{r,t}y_{k,r,t}+\alpha_l)-1}}{\Gamma(\sum_{r,t}y_{k,r,t}+\alpha_l)}\exp\big(-E_{k_c}(R_c+\theta_l)\big)\;dE_{k_c}$$

We can integrate the terms individually as follows:

$$y_{k,r,t}\mathbb{E}_{E_{k_c}|\{t_{k_c,r_c}\}_{R_c},g_{k_c},\dots}[\log E_{k_c}]=y_{k,r,t}\Big[\psi\big(\sum_{r,t}y_{k,r,t}+\alpha_l\big)-\log\big(R_c+\theta_l\big)\Big]$$

Where $\psi(\cdot)$ is the digamma function. The second term is constant in E_{k_c} and so is just

$$\mathbb{E}_{E_{k_c}|\{t_{k_c,r_c}\}_{R_c},g_{k_c},\dots}[y_{k,r,t}\big(\beta_{l,s_{r_c},q_c,t}-\log\sum_t\exp(\beta_{l,s_{r_c},q_c,t})\big)-\log y_{k,r,t}!]=y_{k,r,t}\big(\beta_{l,s_{r_c},q_c,t}-\log\sum_t\exp(\beta_{l,s_{r_c},q_c,t})\big)-\log y_{k,r,}$$

And for the third term, we have

$$\frac{\exp(\beta_{l,s_{r_c},q_c,t})}{\sum_t\exp(\beta_{l,s_{r_c},q_c,t})}\mathbb{E}_{E_{k_c}|\{t_{k_c,r_c}\}_{R_c},g_{k_c},\dots}[E_{k_c}]=\frac{\exp(\beta_{l,s_{r_c},q_c,t})}{\sum_t\exp(\beta_{l,s_{r_c},q_c,t})}\frac{\sum_{r,t}y_{k,r,t}+\alpha_l}{R_c+\theta_l}$$

Putting this all together, and dropping terms constant in the parameters, we have:

$$\mathbb{E}_{E_{k_c}|\{t_{k_c,r_c}\}_{R_c},g_{k_c},\dots;\theta^-}\log P\big(\{t_{k_c,r_c}\}|\lambda(E_{k_c},g_{k_c},s_{r_c},t)\big)=$$

$$y_{k,r,t}\Big[\psi\big(\sum_{r,t}y_{k,r,t}+\alpha_l\big)-\log\big(R_c+\theta_l\big)+\beta_{l,s_{r_c},q_c,t}-\log\sum_t\exp(\beta_{l,s_{r_c},q_c,t})\Big]-\frac{\exp(\beta_{l,s_{r_c},q_c,t})}{\sum_t\exp(\beta_{l,s_{r_c},q_c,t})}\frac{\sum_{r,t}y_{k,r,t}+\alpha_l}{R_c+\theta_l}$$

We can then integrate all terms with respect to $P(g_{k_c})$ as follows:

$$\mathbb{E}_{E_{k_c},g_{k_c}|\{t_{k_c,r_c}\}_{R_c},\ldots;\theta^-} \log P\big(\{t_{k_c,r_c}\}|\lambda(E_{k_c},g_{k_c},s_{r_c},t)\big) =$$

$$\left[\sum_l \frac{p_{k_c,l}}{\prod_{r_c}\prod_t y_{k,r,t}!}\frac{\theta_l^{\alpha_l}\prod_{i=1}^{\sum_{r,t} y_{k,r,t}}(\alpha_l+i-1)}{\big(R_c+\theta_l\big)^{\sum_{r,t} y_{k,r,t}+\alpha_l}}\,\pi_l\right]^{-1}\sum_l\left[\frac{p_{k_c,l}}{\prod_{r_c}\prod_t y_{k,r,t}!}\frac{\theta_l^{\alpha_l}\prod_{i=1}^{\sum_{r,t} y_{k,r,t}}(\alpha_l+i-1)}{\big(R_c+\theta_l\big)^{\sum_{r,t} y_{k,r,t}+\alpha_l}}\,\pi_l\right]$$

$$\left[y_{k,r,t}\Big[\psi\big(\sum_{r,t} y_{k,r,t}+\alpha_l\big)-\log\big(R_c+\theta_l\big)+\beta_{l,s_{r_c},q_c,t}-\log\sum_t\exp(\beta_{l,s_{r_c},q_c,t})\Big]-\frac{\exp(\beta_{l,s_{r_c},q_c,t})}{\sum_t\exp(\beta_{l,s_{r_c},q_c,t})}\frac{\sum_{r,t} y_{k,r,t}+\alpha_l}{R_c+\theta_l}\right]$$

We can then integrate all terms with respect to $P(s_{r_c})$ as follows:

$$\mathbb{E}_{E_{k_c},g_{k_c},s_{r_c}|\{t_{k_c,r_c}\}_{R_c,K_c}} \log P\big(\{t_{k_c,r_c}\}|\lambda(E_{k_c},g_{k_c},s_{r_c},t)\big) =$$

$$\frac{1}{N}\left[\sum_l \frac{p_{k_c,l}}{\prod_{r_c}\prod_t y_{k,r,t}!}\frac{\theta_l^{\alpha_l}\prod_{i=1}^{\sum_{r,t} y_{k,r,t}}(\alpha_l+i-1)}{\big(R_c+\theta_l\big)^{\sum_{r,t} y_{k,r,t}+\alpha_l}}\,\pi_l\right]^{-1}\sum_l\left[\frac{p_{k_c,l}}{\prod_{r_c}\prod_t y_{k,r,t}!}\frac{\theta_l^{\alpha_l}\prod_{i=1}^{\sum_{r,t} y_{k,r,t}}(\alpha_l+i-1)}{\big(R_c+\theta_l\big)^{\sum_{r,t} y_{k,r,t}+\alpha_l}}\,\pi_l\right]$$

$$\sum_n\left[y_{k,r,t}\Big[\psi\big(\sum_{r,t} y_{k,r,t}+\alpha_l\big)-\log\big(R_c+\theta_l\big)+\beta_{l,s_{r_c}^{(n)},q_c,t}-\log\sum_t\exp(\beta_{l,s_{r_c}^{(n)},q_c,t})\Big]-\frac{\exp(\beta_{l,s_{r_c}^{(n)},q_c,t})}{\sum_t\exp(\beta_{l,s_{r_c}^{(n)},q_c,t})}\frac{\sum_{r,t} y_{k,r,t}+\alpha_l}{R_c+\theta_l}\right]$$

That concludes the likelihood term. We can now evaluate the first entropy term as:

$$\mathbb{E}_{E_{k_c}|g_{k_c},\{t_{k_c,r_c}\}_{R_c};\theta^-} \log P\big(E_{k_c}|g_{k_c}\big) =$$

$$\int_{E_{k_c}} \log\left[\frac{\theta_l^{\alpha_l}}{\Gamma(\alpha_l)}E_{k_c}^{\alpha_l-1}\exp(-E_{k_c}\theta_l)\right]\,P(E_{k_c}|g_{k_c},\{s_{r_c},t_{k_c,r_c}\}_{R_c})\,dE_{k_c} =$$

$$\int_{E_{k_c}}\Big[\alpha_l\log\theta_l-\log\Gamma(\alpha_l)+(\alpha_l-1)\log E_{k_c}-E_{k_c}\theta_l\Big]\,P(E_{k_c}|g_{k_c},\{s_{r_c},t_{k_c,r_c}\}_{R_c})\,dE_{k_c} =$$

Beginning with the inner integral, we have

$$\int_{E_{k_c}}\Big[\alpha_l\log\theta_l-\log\Gamma(\alpha_l)+(\alpha_l-1)\log E_{k_c}-E_{k_c}\theta_l\Big]$$

$$\big(R_c+\theta_l\big)^{\sum_{r,t} y_{k,r,t}+\alpha_l}\,\frac{E_{k_c}^{(\sum_{r,t} y_{k,r,t}+\alpha_l)-1}}{\Gamma(\sum_{r,t} y_{k,r,t}+\alpha_l)}\exp\big(-E_{k_c}(R_c+\theta_l)\big)dE_{k_c}$$

Integrating each term in turn gives:

$$\mathbb{E}_{E_{k_c}|g_{k_c},\{t_{k_c,r_c}\}_{R_c}} \log P\big(E_{k_c}|g_{k_c}\big) =$$

$$\alpha_l\log\theta_l-\log\Gamma(\alpha_l)+(\alpha_l-1)\Big[\psi\big(\sum_{r,t} y_{k,r,t}+\alpha_l\big)-\log\big(R_c+\theta_l\big)\Big]-\theta_l\frac{\sum_{r,t} y_{k,r,t}+\alpha_l}{R_c+\theta_l}$$

We can then integrate all terms with respect to $P(g_{k_c})$ as follows:

$$\mathbb{E}_{E_{k_c},g_{k_c}|\{t_{k_c,r_c}\}_{R_c};\theta^-} \log P\big(E_{k_c}|g_{k_c}\big) =$$

$$\left[\sum_l \frac{p_{k_c,l}}{\prod_{r_c}\prod_t y_{k,r,t}!}\frac{\theta_l^{\alpha_l}\prod_{i=1}^{\sum_{r,t} y_{k,r,t}}(\alpha_l+i-1)}{\big(R_c+\theta_l\big)^{\sum_{r,t} y_{k,r,t}+\alpha_l}}\,\pi_l\right]^{-1}\sum_l\left[\frac{p_{k_c,l}}{\prod_{r_c}\prod_t y_{k,r,t}!}\frac{\theta_l^{\alpha_l}\prod_{i=1}^{\sum_{r,t} y_{k,r,t}}(\alpha_l+i-1)}{\big(R_c+\theta_l\big)^{\sum_{r,t} y_{k,r,t}+\alpha_l}}\,\pi_l\right]$$

$$\left[\alpha_l\log\theta_l-\log\Gamma(\alpha_l)+(\alpha_l-1)\Big[\psi\big(\sum_{r,t} y_{k,r,t}+\alpha_l\big)-\log\Big(R_c+\theta_l\Big)\Big]-\theta_l\frac{\sum_{r,t} y_{k,r,t}+\alpha_l}{R_c+\theta_l}\right]$$

Due to the mean field assumption, this term is independent of $P(s_{r_c})$. That concludes the first entropy term. We can now evaluate the second entropy term with respect to $P(g_{k_c})$ as:

$$\mathbb{E}_{g_{k_c}|\{t_{k_c,r_c}\}_{R_c};\theta^-} \log P\big(g_{k_c}\big) =$$

$$\left[\sum_l \frac{p_{k_c,l}}{\prod_{r_c}\prod_t y_{k,r,t}!}\frac{\theta_l^{\alpha_l}\prod_{i=1}^{\sum_{r,t}y_{k,r,t}}(\alpha_l+i-1)}{\big(R_c+\theta_l\big)^{\sum_{r,t}y_{k,r,t}+\alpha_l}}\,\pi_l\right]^{-1}\sum_l\left[\frac{p_{k_c,l}}{\prod_{r_c}\prod_t y_{k,r,t}!}\frac{\theta_l^{\alpha_l}\prod_{i=1}^{\sum_{r,t}y_{k,r,t}}(\alpha_l+i-1)}{\big(R_c+\theta_l\big)^{\sum_{r,t}y_{k,r,t}+\alpha_l}}\,\pi_l\log\pi_l\right]$$

That concludes the second entropy term. We can now evaluate the KL term as:

$$\mathbb{E}_{s_{r_c}|\{t_{k_c,r_c}\}_{R_c,K_c}}\log\frac{P\big(s_{r_c}\big)}{Q\big(s_{r_c}\big)}=$$

$$-\frac{1}{N}\sum_n\left[\frac{1}{2}\Big(d\log(2\pi)+\log\det(\Sigma^+)+s_{r_c}^{(n)\intercal}\Sigma^{-1}s_{r_c}^{(n)}\Big)-\frac{1}{2}\Big(d\log(2\pi)+\log\prod_d(\sigma^2)+\sum_d\big(\frac{s_{r_c,d}-\mu_{r_c}}{\sigma_d}\big)^2\Big)\right]=$$

$$-\frac{1}{2N}\sum_n\Big(\log\det(\Sigma^+)-\sum_d\log\sigma_d^2+s_{r_c}^{(n)\intercal}\Sigma^{-1}s_{r_c}^{(n)}-\sum_d\big(\frac{s_{r_c,d}-\mu_{r_c}}{\sigma_d}\big)^2\Big)$$

That concludes the KL term. Letting

$$w_{k_c,l}=\left[\sum_l \frac{p_{k_c,l}}{\prod_{r_c}\prod_t y_{k,r,t}!}\frac{\theta_l^{\alpha_l}\prod_{i=1}^{\sum_{r,t}y_{k,r,t}}(\alpha_l+i-1)}{\big(R_c+\theta_l\big)^{\sum_{r,t}y_{k,r,t}+\alpha_l}}\,\pi_l\right]^{-1}\left[\frac{p_{k_c,l}}{\prod_{r_c}\prod_t y_{k,r,t}!}\frac{\theta_l^{\alpha_l}\prod_{i=1}^{\sum_{r,t}y_{k,r,t}}(\alpha_l+i-1)}{\big(R_c+\theta_l\big)^{\sum_{r,t}y_{k,r,t}+\alpha_l}}\,\pi_l\right]$$

$$a_{k_c,l}=\frac{\sum_{r,t}y_{k,r,t}+\alpha_l}{R_c+\theta_l}\qquad=\mathbb{E}_{E_{k_c}|\{t_{k_c,r_c}\}_{R_c},g_{k_c}=l}\big[E_{k_c}\big]$$

$$b_{k_c,l}=\psi\big(\sum_{r,t}y_{k,r,t}+\alpha_l\big)-\log\big(R_c+\theta_l\big)\qquad=\mathbb{E}_{E_{k_c}|\{t_{k_c,r_c}\}_{R_c},g_{k_c}=l}\big[\log E_{k_c}\big]$$

For the Poisson point process, the intensity function is by definition a measure of the average spike count over an interval (i.e. the Poisson mean). In this model, we have decomposed the neuron's intensity function $\lambda_k(t)=E_k\exp\big(\beta_l(t)\big)$ as it's neuron specific average spike count E_k and the cluster specific locations of the spikes (shared across neurons in the cluster) $\exp\big(\beta_l(t)\big)$. This parametrization implies that $\int\lambda_k(t)=E_k$ and thus $\int\exp\big(\beta_l(t)\big)=1$, i.e. the latent factor cluster centers are unit norm. We Put these all together to get the ELBO as

$$\frac{1}{N}\sum_c\sum_{k_c}\sum_{r_c}\sum_l\sum_n w_{k_c,l}\left(\sum_t\left[y_{k,r,t}\Big[b_{k_c,l}+\beta_{l,s_{r_c}^{(n)},q_c,t}-\log\sum_t\exp(\beta_{l,s_{r_c}^{(n)},q_c,t})\Big]-\frac{\exp(\beta_{l,s_{r_c}^{(n)},q_c,t})}{\sum_t\exp(\beta_{l,s_{r_c}^{(n)},q_c,t})}a_{k_c,l}\right]+\right.$$

$$\left.\frac{1}{R_c}\Big[\alpha_l\log\theta_l-\log\Gamma(\alpha_l)+(\alpha_l-1)b_{k_c,l}-\theta_l a_{k_c,l}+\log\pi_l\Big]-\frac{1}{K_c}\left[\frac{1}{2}\Big(\log\det(\Sigma^+)-\sum_d\log\sigma_d^2+s_{r_c}^{(n)\intercal}\Sigma^{-1}s_{r_c}^{(n)}-\sum_d\big(\frac{s_{r_c,d}-\mu_{r_c}}{\sigma_d}\big)^2\Big)\right]\right)$$

After collecting like terms and dropping terms constant in the parameters, we have

$$\mathcal{L}=\frac{1}{N}\sum_c\sum_{k_c}\sum_{r_c}\sum_l\sum_n w_{k_c,l}$$

$$\left(\sum_t y_{k,r,t}\,\beta^+_{l,s_{r_c}^{(n)},q_c^+,t}-\big(\sum_t y_{k,r,t}\big)\log\sum_t\exp(\beta^+_{l,s_{r_c}^{(n)},q_c^+,t})+\frac{1}{R_c}\Big[\alpha_l^+\big(\log\theta_l^+ +b_{k_c,l}\big)-\log\Gamma(\alpha_l^+)-\theta_l^+ a_{k_c,l}+\log\pi_l^+\Big]-\right.$$

$$\left.\frac{1}{K_c}\left[\frac{1}{2}\Big(\log\det(\Sigma^+)+s_{r_c}^{(n)\intercal}\Sigma^{+-1}s_{r_c}^{(n)}\Big)-\frac{1}{2}\sum_d\Big(\log\sigma_d^{+2}+\big(\frac{s_{r_c,d}-\mu_{r_c}}{\sigma_d}\big)^2\Big)\right]\right)$$

We note here that the maximizer of this objective with respect to θ_l^+ is given by

$$\mathcal{L}(\theta_l^+)=\sum_c\sum_{k_c}w_{k_c,l}\Big[\alpha_l^+\log\theta_l^+-a_{k_c,l}\theta_l^+\Big]$$

$$\frac{d}{d\theta_l^+}\mathcal{L}(\theta_l^+)=\sum_c\sum_{k_c}w_{k_c,l}\Big[\frac{\alpha_l^+}{\theta_l^+}-a_{k_c,l}\Big]=0$$

$$\frac{\alpha_l^+}{\theta_l^+}\sum_c\sum_{k_c}w_{k_c,l}=\sum_c\sum_{k_c}w_{k_c,l}a_{k_c,l}$$

$$\theta_l^+ = \alpha_l^+ \frac{\sum_c \sum_{k_c} w_{k_c,l}}{\sum_c \sum_{k_c} w_{k_c,l} a_{k_c,l}} = \frac{\alpha_l^+}{\mathbb{E}_E[a_{k_c,l}]} = \frac{\alpha_l^+}{\mathbb{E}_E[\mathbb{E}_{E_{k_c}|\{t_{k_c,r_c}\}_{R_c},g_{k_c}=l}[E_{k_c}]]}$$

In addition, using the Lagrangian, we have

$$\mathcal{L}(\pi_l^+) = \sum_c \sum_{k_c} w_{k_c,l} \Big[\log \pi_l^+ \Big] - \lambda(\sum_l \pi_{l:l\in a} - 1)$$

$$\frac{d}{d\pi_l^+} \mathcal{L}(\pi_l^+) = \frac{1}{\pi_l^+} \sum_c \sum_{k_c} w_{k_c,l} - \lambda = 0$$

$$\pi_l^+ = \frac{1}{\sum_c k_{c,a}} \sum_c \sum_{k_c} w_{k_c,l}$$

We get the final expression by plugging $\pi_l \; \forall \; l$ back into the constraint, which gives $\lambda = \sum_c k_{c,a}$. We can interpret this as the sum of fractional assignments of neurons to factor l divided by the total number of neurons, i.e. the average proportion of assignments to factor l. Since these parameters have closed-form maximums, we do not need to maximize over them in the ELBO, and we thus have:

$$\mathcal{L} = \frac{1}{N} \sum_c \sum_{k_c} \sum_{r_c} \sum_l \sum_n w_{k_c,l}$$

$$\Bigg(\sum_t y_{k,r,t} \, \beta^+_{l,s^{(n)}_{r_c},q^+_c,t} - \big(\sum_t y_{k,r,t}\big) \log \sum_t \exp(\beta^+_{l,s^{(n)}_{r_c},q^+_c,t}) + \frac{1}{R_c} \Big[\alpha_l^+ (\log \theta_l + b_{k_c,l}) - \log \Gamma(\alpha_l^+) \Big] -$$

$$\frac{1}{K_c} \left[\frac{1}{2} \Big(\log \det(\Sigma^+) + s^{(n)\intercal}_{r_c} \Sigma^{+-1} s^{(n)}_{r_c} \Big) - \frac{1}{2} \sum_d \Big(\log \sigma_d^{+2} + \big(\frac{s^{(n)}_{r_c,d} - \mu_{r_c}}{\sigma_d}\big)^2 \Big) \right] \Bigg)$$

There are times when no neuron is assigned to a particular factor, and this causes issues with the code. If no neuron is assigned to a factor l, then $\theta_l = 0$ and $\pi_l = 0$, and no gradients can propagate back from θ_l to α_l, thereby stunting the learning. Also, when no neuron is assigned to a particular factor, we cannot learn anything about that factor from data, and end up entirely depending on the prior (in this case the penalty). I get around this by a trick that amounts to moving one neuron from the factor with the highest number of neurons on to the factor with no neuron. That way there is always some learning for every factor and its corresponding α through all iterations.
Note that

$$\frac{p_{k_c,l}}{\prod_{r_c} \prod_t y_{k,r,t}!} \frac{\theta_l^{\alpha_l} \prod_{i=1}^{\sum_{r,t} y_{k,r,t}} (\alpha_l + i - 1)}{\big(R_c + \theta_l\big)^{\sum_{r,t} y_{k,r,t} + \alpha_l}} \, \pi_l = \prod_{r,t} \left[y_{k,r,t}!^{-1} \left[\frac{\exp\big(\beta_{l,\mu_{r_c},q_c,t}\big)}{\sum_t \exp(\beta_{l,\mu_{r_c},q_c,t})} \right]^{y_{k,r,t}} \right] \frac{\theta_l^{\alpha_l} \prod_{i=1}^{\sum_{r,t} y_{k,r,t}} (\alpha_l + i - 1)}{\big(R_c + \theta_l\big)^{\sum_{r,t} y_{k,r,t} + \alpha_l}} \, \pi_l$$

Let

$$\mathbf{U} = \log \left[\prod_{r,t} y_{k,r,t}!^{-1} \left[\frac{\exp\big(\beta_{l,\mu_{r_c},q_c,t}\big)}{\sum_t \exp(\beta_{l,\mu_{r_c},q_c,t})} \right]^{y_{k,r,t}} \frac{\theta_l^{\alpha_l} \prod_{i=1}^{\sum_{r,t} y_{k,r,t}} (\alpha_l + i - 1)}{\big(R_c + \theta_l\big)^{\sum_{r,t} y_{k,r,t} + \alpha_l}} \, \pi_l \right] =$$

$$\sum_{r,t} y_{k,r,t} \Big(\beta_{l,\mu_{r_c},q_c,t} - \log \sum_t \exp(\beta_{l,\mu_{r_c},q_c,t}) \Big) - \sum_{r,t} \log \Gamma(y_{k,r,t} + 1) + \sum_{i=1}^{\sum_{r,t} y_{k,r,t}} \log(\alpha_l + i - 1) +$$

$$\alpha_l \log \theta_l - \Big(\sum_{r,t} y_{k,r,t} + \alpha_l \Big) \log \Big(R_c + \theta_l \Big) + \log \pi_l$$

Thus

$$w_{k_c,l} = \text{softmax}_l(\mathbf{U})$$

We define the time warping function $\phi(t)$ in the following way:

$$\phi(t) = \begin{cases} t, & 0 \le t < \tau_l \\ (t - \tau_l) \frac{\tau^* - \tau_l}{(\tau^* + s_{r_c} + q_c) - \tau_l} + \tau_l, & \tau_l \le t < \tau^* + s_{r_c} + q_c \\ (t - (\tau^* + s_{r_c} + q_c)) \frac{\tau_r - \tau^*}{\tau_r - (\tau^* + s_{r_c} + q_c)} + \tau^*, & \tau^* + s_{r_c} + q_c \le t < \tau_r \\ t, & \tau_r \le t < T \end{cases}$$

Where τ_l and τ_r correspond to the left and right landmarks, and τ^* is the average peak time across all trials and conditions. This corresponds to a piece-wise linear function. Note that the time-warping function is parametrized by the landmarks, and we impose the constraint that the shifted peak times are within the time warping window.This constraint is enforced using the a rescaled sigmoid function, commonly called the tan hyperbolic function. The average peak time is also contained within the landmark window, using a thresholding function. If we define the thresholded average peak time as $\tilde{\tau}^*$, and the constrained trial peak time as $\nu = \nu(\tau^* + s_{r_c} + q_c)$, then the time warping function is

$$\phi(t) = \begin{cases} t, & 0 \leq t < \tau_l \\ (t - \tau_l)\frac{\tilde{\tau}^* - \tau_l}{\nu - \tau_l} + \tau_l, & \tau_l \leq t < \nu \\ (t - \nu)\frac{\tilde{\tau}^* - \tau_r}{\nu - \tau_r} + \tilde{\tau}^*, & \nu \leq t < \tau_r \\ t, & \tau_r \leq t < T \end{cases}$$

We iteratively maximize the ELBO over these parameters $\{\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{q}\}$, each time recomputing the weigh matrices from the current maximizes.
Because of the dependence of both the E step and M step on θ_l, and the dependence of θ_l on the outputs of both steps, we must update θ_l after each step. The algorithm this looks like:
Initialize theta and alpha, then each iteration looks like:
E step update given alpha and theta, theta update given E step update (and alpha), M step update given E step update and theta update, theta update given E step and M step alpha update.
Essentially, the theta variable requires both an E step and an M step update.

Appendix C

Underspecification of the number of functional units

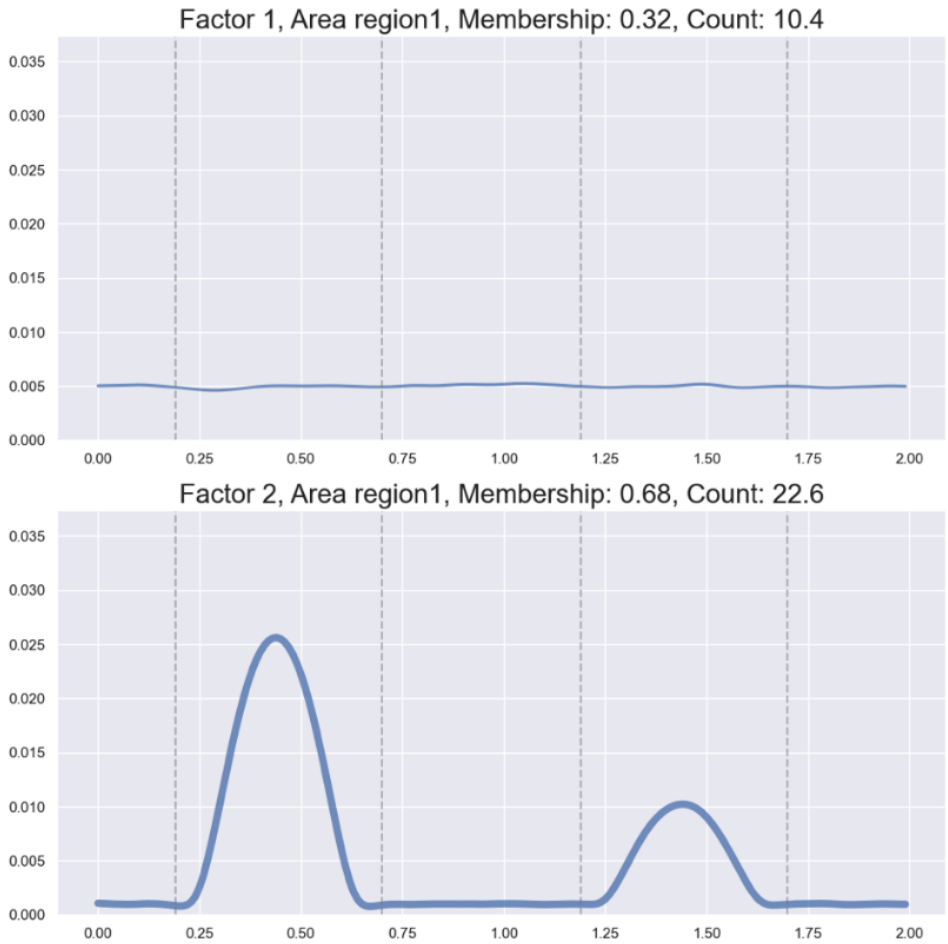

Figure C.1: The figure shows the effect of underspecification of the number of functional units on the learned population intensity functions. The single peak. The ground truth is shown in figure 3.7B (in the third column). When too few functional units are used, the model combines the most similar functional units. In this case, the constant firing rate intensity functions are combined, and the bursting intensity functions are combined.

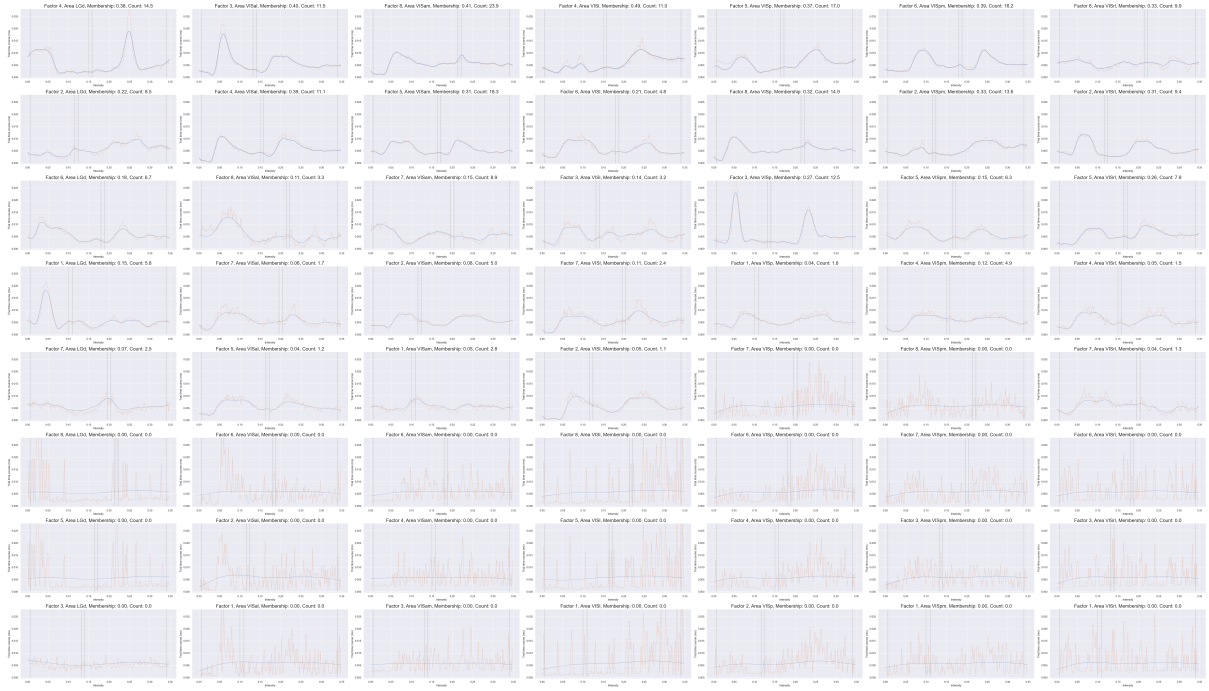

Figure C.2: The figure shows the output obtained by running our model on 7 areas in the Allen Institute Dataset (6 cortical visual areas and one thalamic nucleus.) We include this to demonstrate the ability of our model to scale to very large datasets, as to handle overspecification of the number of functionalunits to fit. We specify 8 functional units, but in most cases, the model only fit 5.

Bibliography

L. F. Abbott and P. Dayan. The effect of correlated variability on the accuracy of a population code. *Neural computation*, 11(1):91–101, 1999.

J. Aljadeff, B. J. Lansdell, A. L. Fairhall, and D. Kleinfeld. Analysis of neuronal spike trains, deconstructed. *Neuron*, 91(2):221–259, 2016. ISSN 0896-6273. doi: https://doi.org/10.1016/j.neuron.2016.05.039. URL https://www.sciencedirect.com/science/article/pii/S0896627316302501.

Allen Institute MindScope Program. Allen Brain Observatory – Neuropixels Visual Coding (Dataset), 2019. URL https://portal.brain-map.org/explore/circuits/visual-coding-neuropixels.

A. Antonini, M. Fagiolini, and M. P. Stryker. Anatomical correlates of functional plasticity in mouse visual cortex. *The Journal of Neuroscience*, 19(11):4388–4406, June 1999. ISSN 0270-6474, 1529-2401. doi: 10.1523/JNEUROSCI.19-11-04388.1999. URL http://www.jneurosci.org/lookup/doi/10.1523/JNEUROSCI.19-11-04388.1999.

B. B. Averbeck, P. E. Latham, and A. Pouget. Neural correlations, population coding and computation. *Nature reviews neuroscience*, 7(5):358–366, 2006.

D. S. Bassett and O. Sporns. Network neuroscience. *Nature neuroscience*, 20(3):353–364, 2017.

S. Behseta, T. Berdyyeva, C. R. Olson, and R. E. Kass. Bayesian correction for attenuation of correlation in multi-trial spike count data. *Journal of neurophysiology*, 101(4):2186–2193, 2009.

D. J. Bemdt. James clifford. 1994.

Y. Ben-Shaul, H. Bergman, Y. Ritov, and M. Abeles. Trial to trial variability in either stimulus or action causes apparent correlation and synchrony in neuronal activity. *Journal of neuroscience methods*, 111(2):99–110, 2001.

B. B. Biswal, M. Mennes, X.-N. Zuo, S. Gohel, C. Kelly, S. M. Smith, C. F. Beckmann, J. S. Adelstein, R. L. Buckner, S. Colcombe, et al. Toward discovery science of human brain function. *Proceedings of the national academy of sciences*, 107(10):4734–4739, 2010.

D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

A. G. Bondy, R. M. Haefner, and B. G. Cumming. Feedback determines the structure of correlated variability in primary visual cortex. *Nature neuroscience*, 21(4):598–606, 2018.

S. L. Bressler and V. Menon. Large-scale brain networks in cognition: emerging methods and principles. *Trends in cognitive sciences*, 14(6):277–290, 2010.

D. R. Brillinger. Maximum likelihood analysis of spike trains of interacting nerve cells. *Biological cybernetics*, 59(3):189–200, 1988.

C. D. Brody. Disambiguating different covariation types. *Neural Computation*, 11(7):1527–1535, 1999.

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.

E. Bullmore and O. Sporns. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature reviews neuroscience*, 10(3):186–198, 2009.

G. Buzsaki and A. Draguhn. Neuronal oscillations in cortical networks. *science*, 304(5679):1926–1929, 2004.

Y. Chen, H. Douglas, B. Medina, M. Olarinre, J. Siegle, and R. Kass. Population burst propagation across interacting areas of the brain. *J Neurophysiology*, 128(6):1578–1592, 2022. doi: 10.1152/jn.00066.2022.

M. R. Cohen and A. Kohn. Measuring and interpreting neuronal correlations. *Nature neuroscience*, 14(7):811–819, 2011.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977. ISSN 00359246. URL http://www.jstor.org/stable/2984875.

R. D. D'Souza, Q. Wang, W. Ji, A. M. Meier, H. Kennedy, K. Knoblauch, and A. Burkhalter. Hierarchical and nonhierarchical features of the mouse visual cortical network. *Nature Communications*, 13(1):503, Dec. 2022. ISSN 2041-1723. doi: 10.1038/s41467-022-28035-y. URL https://www.nature.com/articles/s41467-022-28035-y.

A. Gelman. Prior distributions for variance parameters in hierarchical models (comment on article by Browne and Draper). *Bayesian Analysis*, 1(3):515 – 534, 2006. doi: 10.1214/06-BA117A. URL https://doi.org/10.1214/06-BA117A.

L. L. Glickfeld and S. R. Olsen. Higher-order areas of the mouse visual cortex. *Annual Review of Vision Science*, 3(1):251–273, 2017. doi: 10.1146/annurev-vision-102016-061331. URL https://doi.org/10.1146/annurev-vision-102016-061331. PMID: 28746815.

Y. Gu, S. Liu, C. R. Fetsch, Y. Yang, S. Fok, A. Sunkara, G. C. DeAngelis, and D. E. Angelaki. Perceptual learning reduces interneuronal correlations in macaque visual cortex. *Neuron*, 71(4):750–761, 2011.

J. A. Harris, S. Mihalas, K. E. Hirokawa, J. D. Whitesell, H. Choi, A. Bernard, P. Bohn, S. Caldejon, L. Casal, A. Cho, A. Feiner, D. Feng, N. Gaudreault, C. R. Gerfen, N. Graddis, P. A. Groblewski, A. M. Henry, A. Ho, R. Howard, J. E. Knox, L. Kuan, X. Kuang, J. Lecoq, P. Lesnar, Y. Li, J. Luviano, S. McConoughey, M. T. Mortrud, M. Naeemi, L. Ng, S. W. Oh, B. Ouellette, E. Shen, S. A. Sorensen, W. Wakeman, Q. Wang, Y. Wang, A. Williford, J. W. Phillips, A. R. Jones, C. Koch, and H. Zeng. Hierarchical organization of cortical and thalamic connectivity. *Nature*, 575(7781):195–202, Nov. 2019. ISSN 0028-0836, 1476-4687. doi: 10.1038/s41586-019-1716-z. URL http://www.nature.com/articles/s41586-019-1716-z.

C. Houghton and K. Sen. A new multineuron spike train metric. *Neural computation*, 20(6):1495–1511, 2008.

M. D. Humphries. Spike-train communities: finding groups of similar spike trains. *Journal of Neuroscience*, 31 (6):2321–2336, 2011.

X. Jia, J. H. Siegle, S. Durand, G. Heller, T. Ramirez, and S. R. Olsen. Multi-area functional modules mediate feedforward and recurrent processing in visual cortical hierarchy. *bioRxiv*, 2020.

X. Jia, J. H. Siegle, S. Durand, G. Heller, T. Ramirez, C. Koch, and S. R. Olsen. Multi-regional module-based signal transmission in mouse visual cortex. *Neuron*, 110:1585–1598, 2022.

M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Machine learning*, 37:183–233, 1999.

J. J. Jun, N. A. Steinmetz, J. H. Siegle, D. J. Denman, M. Bauza, B. Barbarits, A. K. Lee, C. Anastassiou, A. Andrei, C. Aydın, M. Barbic, T. J. Blanche, V. Bonin, J. Couto, B. Dutta, S. L. Gratiy, D. A. Gutnisky, M. Häusser, B. Karsh, P. Ledochowitsch, C. M. Lopez, C. C. Mitelut, S. Musa, M. Okun, M. Pachitariu, J. Putzeys, P. D. Rich, C. Rossant, W. lung Sun, K. Svoboda, M. Carandini, K. D. Harris, C. Koch, J. O'Keefe, and T. D. Harris. Fully integrated silicon probes for high-density recording of neural activity. *Nature*, 551: 232–236, 2017. URL https://api.semanticscholar.org/CorpusID:205262002.

R. E. Kass and V. Ventura. A spike-train probability model. *Neural computation*, 13(8):1713–1720, 2001.

R. E. Kass, V. Ventura, and E. N. Brown. Statistical issues in the analysis of neuronal data. *Journal of neurophysiology*, 94(1):8–25, 2005.

R. E. Kass, U. T. Eden, and E. N. Brown. *Analysis of Neural Data*. Springer New York, New York, NY, 2014. ISBN 978-1-4614-9602-1. doi: 10.1007/978-1-4614-9602-1_1. URL https://doi.org/10.1007/978-1-4614-9602-1_1.

R. E. Kass, H. Bong, M. Olarinre, Q. Xin, and K. N. Urban. Identification of interacting neural populations: methods and statistical considerations. *Journal of Neurophysiology*, 130(3):475–496, 2023.

N. Klein, J. H. Siegle, T. Teichert, and R. E. Kass. Cross-population coupling of neural activity based on gaussian process current source densities. *PLOS Computational Biology*, 17:1–24, 11 2021. doi: 10.1371/journal.pcbi.1009601. URL https://doi.org/10.1371/journal.pcbi.1009601.

J. Lee, H. R. Kim, and C. Lee. Trial-to-trial variability of spike response of v1 and saccadic response time. *Journal of neurophysiology*, 104(5):2556–2572, 2010.

J. Lee, M. Joshua, J. F. Medina, and S. G. Lisberger. Signal, noise, and variation in neural and sensory-motor latency. *Neuron*, 90(1):165–176, 2016.

D. Lewandowski, D. Kurowicka, and H. Joe. Generating random correlation matrices based on vines and extended onion method. *Journal of Multivariate Analysis*, 100(9):1989–2001, 2009. ISSN 0047-259X. doi: https://doi.org/10.1016/j.jmva.2009.04.008. URL https://www.sciencedirect.com/science/article/pii/S0047259X09000876.

A. Lin, Y. Zhang, J. Heng, S. A. Allsop, K. M. Tye, P. E. Jacob, and D. Ba. Clustering time series with nonlinear dynamics: A bayesian non-parametric and particle-based approach. pages 2476–2484, 2019. URL http://proceedings.mlr.press/v89/lin19b.html.

S. Manita, T. Suzuki, C. Homma, T. Matsumoto, M. Odagawa, K. Yamada, K. Ota, C. Matsubara, A. Inutsuka, M. Sato, M. Ohkura, A. Yamanaka, Y. Yanagawa, J. Nakai, Y. Hayashi, M. Larkum, and M. Murayama. A top-down cortical circuit for accurate sensory perception. *Neuron*, 86(5):1304–1316, June 2015. ISSN 08966273. doi: 10/f7dp4m. URL https://linkinghub.elsevier.com/retrieve/pii/S0896627315004134.

G. J. McLachlan, T. Krishnan, and S. K. Ng. The em algorithm. Technical report, Papers, 2004.

K. P. Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

S. Nirenberg and P. E. Latham. Decoding neuronal spike trains: how important are correlations? *Proceedings of the National Academy of Sciences*, 100(12):7348–7353, 2003.

M. Olarinre, J. H. Siegle, and R. E. Kass. Relative timing and coupling of multiple population bursts in large scale neural recordings across multiple subjects. *In review at the Journal of Neuroscience*.

J. W. Pillow, J. Shlens, L. Paninski, A. Sher, A. M. Litke, E. Chichilnisky, and E. P. Simoncelli. Spatio-temporal correlations and visual signalling in a complete neuronal population. *Nature*, 454(7207):995–999, 2008.

N. G. Polson and J. G. Scott. On the half-cauchy prior for a global scale parameter. *Bayesian Analysis*, 7(4):887 – 902, 2012. doi: 10.1214/12-BA730. URL https://doi.org/10.1214/12-BA730.

D. Reich. Spikes: Exploring the neural code. *Network: Computation in Neural Systems*, 8(3):008, aug 1997. doi: 10.1088/0954-898X/8/3/008. URL https://dx.doi.org/10.1088/0954-898X/8/3/008.

S. Sachidhanandam, V. Sreenivasan, A. Kyriakatos, Y. Kremer, and C. C. H. Petersen. Membrane potential correlates of sensory perception in mouse barrel cortex. *Nature Neuroscience*, 16(11):1671–1677, Nov. 2013. ISSN 1097-6256, 1546-1726. doi: 10.1038/nn.3532. URL http://www.nature.com/articles/nn.3532.

M. T. Schmolesky, Y. Wang, D. P. Hanes, K. G. Thompson, S. Leutgeb, J. D. Schall, and A. G. Leventhal. Signal timing across the macaque visual system. *Journal of Neurophysiology*, 79(6):3272–3278, June 1998. ISSN 0022-3077, 1522-1598. doi: 10.1152/jn.1998.79.6.3272. URL http://www.physiology.org/doi/10.1152/jn.1998.79.6.3272.

M. N. Shadlen and W. T. Newsome. The variable discharge of cortical neurons: implications for connectivity, computation, and information coding. *Journal of neuroscience*, 18(10):3870–3896, 1998.

J. Siegle, X. Jia, S. Durand, S. Gale, C. Bennett, N. Graddis, G. Heller, T. K. Ramirez, H. Choi, J. A. Luviano, P. A. Groblewski, R. Ahmed, A. Arkhipov, A. Bernard, Y. N. Billeh, D. Brown, M. A. Buice, N. Cain, S. Caldejon, L. Casal, A. Cho, M. Chvilicek, T. C. Cox, K. Dai, D. J. Denman, S. E. J. de Vries, R. Dietzman, L. Esposito, C. Farrell, D. Feng, J. Galbraith, M. Garrett, E. C. Gelfand, N. Hancock, J. A. Harris,

R. Howard, B. Hu, R. Hytnen, R. Iyer, E. Jessett, K. Johnson, I. Kato, J. Kiggins, S. Lambert, J. Lecoq, P. Ledochowitsch, J. H. Lee, A. Leon, Y. Li, E. Liang, F. Long, K. Mace, J. Melchior, D. Millman, T. Mollenkopf, C. Nayan, L. Ng, K. Ngo, T. Nguyen, P. R. Nicovich, K. North, G. K. Ocker, D. Ollerenshaw, M. Oliver, M. Pachitariu, J. Perkins, M. Reding, D. Reid, M. Robertson, K. Ronellenfitch, S. Seid, C. Slaughterbeck, M. Stoecklin, D. Sullivan, B. Sutton, J. Swapp, C. Thompson, K. Turner, W. Wakeman, J. D. Whitesell, D. Williams, A. Williford, R. Young, H. Zeng, S. Naylor, J. W. Phillips, R. C. Reid, S. Mihalas, S. R. Olsen, and C. Koch. Survey of spiking in the mouse visual system reveals functional hierarchy. *Nature*, 592:86–92, Jan. 2021. ISSN 0028-0836, 1476-4687. doi: 10.1038/s41586-020-03171-x. URL http://www.nature.com/articles/s41586-020-03171-x.

M. A. Smith and A. Kohn. Spatial and temporal scales of neuronal correlation in primary visual cortex. *Journal of Neuroscience*, 28(48):12591–12603, 2008.

M. A. Smith and M. A. Sommer. Spatial and temporal scales of neuronal correlation in visual area v4. *Journal of Neuroscience*, 33(12):5422–5432, 2013.

S. M. Smith, K. L. Miller, G. Salimi-Khorshidi, M. Webster, C. F. Beckmann, T. E. Nichols, J. D. Ramsey, and M. W. Woolrich. Network modelling methods for fmri. *Neuroimage*, 54(2):875–891, 2011.

B. Sotomayor-Gómez, F. P. Battaglia, and M. Vinck. Spikeship: A method for fast, unsupervised discovery of high-dimensional neural spiking patterns. *PLOS Computational Biology*, 19(7):e1011335, 2023.

O. Sporns, D. R. Chialvo, M. Kaiser, and C. C. Hilgetag. Organization, development and function of complex brain networks. *Trends in cognitive sciences*, 8(9):418–425, 2004.

Stan Development Team. RStan: the R interface to Stan, 2024. URL https://mc-stan.org/. R package version 2.32.6.

N. A. Steinmetz, C. Koch, K. D. Harris, and M. Carandini. Challenges and opportunities for large-scale electrophysiology with neuropixels probes. *Current Opinion in Neurobiology*, 50:92–100, 2018. ISSN 0959-4388. doi: https://doi.org/10.1016/j.conb.2018.01.009. URL https://www.sciencedirect.com/science/article/pii/S0959438817303161. Neurotechnologies.

N. A. Steinmetz, C. Aydin, A. Lebedeva, M. Okun, M. Pachitariu, M. Bauza, M. Beau, J. Bhagat, C. Böhm, M. Broux, S. Chen, J. Colonell, R. J. Gardner, B. Karsh, F. Kloosterman, D. Kostadinov, C. Mora-Lopez, J. O'Callaghan, J. Park, J. Putzeys, B. Sauerbrei, R. J. J. van Daal, A. Z. Vollan, S. Wang, M. Welkenhuysen, Z. Ye, J. T. Dudman, B. Dutta, A. W. Hantman, K. D. Harris, A. K. Lee, E. I. Moser, J. O'Keefe, A. Renart, K. Svoboda, M. Häusser, S. Haesler, M. Carandini, and T. D. Harris. Neuropixels 2.0: A miniaturized high-density probe for stable, long-term brain recordings. *Science*, 372(6539):eabf4588, Apr. 2021. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.abf4588. URL https://www.science.org/doi/10.1126/science.abf4588.

W. Truccolo, U. T. Eden, M. R. Fellows, J. P. Donoghue, and E. N. Brown. A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects. *Journal of neurophysiology*, 93(2):1074–1089, 2005.

M. C. van Rossum. A novel spike distance. *Neural computation*, 13(4):751–763, 2001.

V. Ventura, C. Cai, and R. E. Kass. Trial-to-trial variability and its effect on time-varying dependency between two neurons. *Journal of neurophysiology*, 94(4):2928–2939, 2005.

J. D. Victor and K. P. Purpura. Nature and precision of temporal coding in visual cortex: a metric-space analysis. *Journal of neurophysiology*, 76(2):1310–1326, 1996.

G. Vinci, V. Ventura, M. A. Smith, and R. E. Kass. Separating spike count correlation from firing rate correlation. *Neural computation*, 28(5):849–881, 2016.

W. Wang, G. Lyu, Y. Shi, and X. Liang. Time series clustering based on dynamic time warping. In *2018 IEEE 9th international conference on software engineering and service science (ICSESS)*, pages 487–490. IEEE, 2018.

A. H. Williams, B. Poole, N. Maheswaranathan, A. K. Dhawale, T. Fisher, C. D. Wilson, D. H. Brann, E. M. Trautmann, S. Ryu, R. Shusterman, D. Rinberg, B. P. Ölveczky, K. V. Shenoy, and S. Ganguli. Discovering precise temporal patterns in large-scale neural recordings through robust and interpretable time warping. *Neuron*, 105(2):246–259.e8, 2020. ISSN 0896-6273. doi: https://doi.org/10.1016/j.neuron.2019.10.020. URL https://www.sciencedirect.com/science/article/pii/S0896627319308943.

E. Zohary, M. N. Shadlen, and W. T. Newsome. Correlated neuronal discharge rate and its implications for psychophysical performance. *Nature*, 370(6485):140–143, 1994.