Applications of Point Process Modeling to Spiking Neurons

Yu Chen

January 11, 2022

Neuroscience Institute
Machine Learning Department
Dietrich College of Humanities and Social Sciences
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Robert E. Kass (chair)
Valérie Ventura
Asohan Amarasingham
Joshua H. Siegle

*Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy.*

Abstract

Electrophysiological measurements of neurons recorded individually, collectively, and across brain regions simultaneously have been advancing the study of biophysical properties of individual neurons, information encoding and decoding, and interactions between neuronal ensembles. The recordings are composed of sequences of action potentials, usually referred to as spike trains, which carry signals of neural activity but they are contaminated with ubiquitous Poisson-type noise and the patterns vary from neuron to neuron, time to time, and trial to trial. The discreteness and randomness of the data make point process a suitable tool for understanding the neural signals represented by spike train data.

This thesis provides four applications of point process modeling to spiking neurons. The first project addresses stability of fitted point process regression models. In the second project, we aim to bridge biophysical modeling and statistical modeling by describing how ion channel conductance can affect spike train patterns. The third project studies the covariation of time-dependent population firing rate features among interacting brain regions. The fourth project focuses on the inter-spike dependency between brain areas with weak coupling effects.

Acknowledgments

My Ph.D. life at CMU in the past five years is a great journey and will be precious memory. I would love to acknowledge people who make this thesis possible. First and foremost, I would like to express my extreme gratitude to my advisor Professor Robert E. Kass for his enormous support and invaluable guidance over the past five years. It is incredible honor to be advised by him and joyful to work with him. His enthusiasm, inspiring ideas, and deep insights into statistics, machine learning, and neuroscience have dramatically shaped my mindset. The research experience under his supervision is a tremendous fortune for my career.

Next, I would like to thank my thesis committee members. I do sincerely appreciate Valerie's rigorous training and very detailed instructions on my first project of my Ph.D., which was a solid first step in my early research and I keep benefiting from it. Also many thanks for her being so patient and tolerating my boldness. I am grateful for Han's guidance, many philosophical ideas and suggestions, which are critical for me in formalizing the problem and method. I wish I could meet Han earlier so I would definitely learn much more from him. I am very fortunate to have Josh, one of the main contributors of the Neuropixels dataset and a pioneer in the field, as both my collaborator and committee member. The analysis of the Neuropixels dataset is the foundation of two chapters. I can not imagine it is possible to have this thesis without his support.

It has been a privilege to work with many extraordinary collaborators. Qi and I worked productively together over one summer, and quickly got the first point process paper out. He has also been an amazing friend, lab mate, and roommate. Nate, Nathan, and Alon helped me a lot in understanding biophysical properties of neurons, and opened a new world to me. I thank Tolani, Hannah and Bryan for their endeavors to the feature coupling project. I really enjoy brainstorming and discussing scientific puzzles with Tolani. I appreciate the mentorship of Jie Chen, Wenjie Zhao, Yi-fan Chen, and Yusef Shafi during my internships at AT&T and Google Research.

Finally, I would like to attribute my accomplishment to my parents Qunzhi Chen, Ruiqin Yang, and my girlfriend Ivy Gao. I could not have done this without their encouragement, support, and love.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Electrophysiological measurements of single and multiple neurons play a critical role in studying neurons' biophysical properties, information encoding and decoding, neuron-to-neuron interactions, region-to-region interactions. This fast-growing technique now is able to simultaneously track the activity of up to a thousand of neurons across multiple brain regions [127, 128, 129]. The recorded sequences of action potentials are usually called *spike trains*. The spike trains carry information about external stimuli [111], latent states [31, 123], neuron-to-neuron interactions [81, 111], functional connectivity between neuronal ensembles [32], etc. But these signals are usually accompanied by Poisson-type noise. And the spike train patterns, including the timing of spikes and mean firing rate, vary from neuron to neuron, time to time and trial to trial even under the same experimental condition [12, 32, 73, 95, 135].

Point processes offer a powerful tool for modeling discrete and stochastic spike trains, which has been widely applied in neuroscience for at least 20 years [79], and for early works see [23, 75]. The point process framework models a spike train through the instantaneous firing rate, called *intensity function*, defined as,

$$\lambda(t \mid \mathcal{H}_t) = \lim_{\Delta \to 0} \frac{\mathbb{P}(N(t, t+\Delta] > 0 \mid \mathcal{H}_t)}{\Delta} \tag{1.1}$$

where $N(t, t+\Delta]$ is the spike count over the interval $(t, t+\Delta]$, and $\mathcal{H}_t$ could represent the history of the process up to time t, including not only the spikes, but possibly some other measurements, for example a stimulus [79]. The log-likelihood of a spike train in $[0, T]$ is

$$L = \int_0^T \log \lambda(t \mid \mathcal{H}_t) N(\mathrm{d}t) - \int_0^T \lambda(t \mid \mathcal{H}_t) \mathrm{d}t \tag{1.2}$$

which usually defines as the target estimation equation or loss function for the model. Much of the point process modeling literature considers how to express $\lambda(t \mid \mathcal{H}_t)$ in particular ways favoring certain properties of the data. The point process framework is flexible because the intensity function can be modularized into separate components that are responsible for different factors. Many of these components are assembled in additive form, logarithmic additive form, or logit additive form, which are similar to the generalized linear model:

$$\lambda(t \mid \mathcal{H}_t), \text{ or } \log \lambda(t \mid \mathcal{H}_t), \text{ or logit } p(t \mid \mathcal{H}_t) = \text{factor}_1(t) + ... + \text{factor}_k(t) \tag{1.3}$$

where $p(t \mid \mathcal{H}_t)$ is the probability of having a spike in the time bin at t. Such a model can be designed for individual neurons or multiple neurons. It can incorporate factors such as external stimuli, behaviors, LFP recordings, latent variables, and inter-spike dependency, such as neuron-to-neuron coupling effects or post-spike history effects within a neuron. We categorize some previous works into different combinations of these components. Table 1.1 lists some representative references with short summaries therein.

Single neuron	Stimulus	*in vitro* olfactory mitral cell with injected current [133]; Artificial neuron (Izhikevich model) with simple injected signal [137]; Plasticity of place receptive field with spatial information [43, 47];
	Post-spike history effects	Post-spike filter with refractory effects [133]; Post-spike filter determines firing patterns [137]; Mixture of post-spike filters in Parkinson's disease [37]; Modeling finite number of post spikes [30, 77, 116];
	Latent variables	Brownian motion as the inhomogeneous baseline [124]; Synaptic dynamics and post-synaptic response [4]; Linear dynamic system for evolving receptive field parameters [43, 47];
Multiple neurons	Stimulus	*in vitro* parasol ganglion cells with visual stimulus [111]; 2-D monkey hand movement [134]; LFP oscillation phase [141]; Position encoding in rat CA1 [8, 23, 115]; Motor cortex encoding for arm movement [90];
	Coupling effects	Post-spike filter with refractory, and excitatory or inhibitory coupling filters between neurons [111, 134]; Coupling filters within/between M1 and PMd neurons [130, 134]; Relate the coupling filter to Granger causality [44, 86]; Coupling filters in mouse lumbar spinal cord [44]; Small-world-ness network structure in monkey visual cortex [51]; Coupling filter-based network reconstruction agrees with synaptic connection in crab Stomatogastric ganglion [52]; Coupling filters between rat hippocampal place cells [108]; A theoretical study of firing rate correlations induced by coupling filters and network topology [110]; Coupling filter between rat *in vitro* neurons [116];
	Latent variables	*Gaussian process:* Two Gaussian processes for smooth temporal structure and smooth tuning curves of hippocampal cells decoding [138]; Point process version of GPFA [41, 140]; *Dynamic system:* 1-D latent linear dynamic system for population inhomogeneous baseline [123]; Position decoding from rat CA1 [8, 23, 115]; Motor cortex decoding for arm movement [90, 134]; *Hidden Markov model:* Population with alternating active-quiescent states [31]; *Factor analysis:* Factor analysis using hidden processes [41, 80];

Table 1.1: A list of references categorized according to the types of factors.

1.2 Overview of thesis contributions

	Single neuron	Multiple neurons	External stimuli	Latent variables	Inter-spike dependency
Chapter 2	✓				✓
Chapter 3	✓		✓		✓
Chapter 4		✓	✓	✓	
Chapter 5		✓		✓	✓

Table 1.2: The outline of the thesis.

This thesis provides four applications of point process modeling to spiking neurons. Table 1.2 shows the thesis outline and the main components of each project.

The first project in Chapter 2 addresses stability of fitted point process regression models. The motivation came from a recent paper by Gerhard et al. presented at the SAND8 conference [53, 117]. The authors described a kind of unstable point process, in which a fitted model can generate simulated spike trains with explosive firing rates. We first point out the issue is related to the lack of fit in some situations, and it can be fixed accordingly. Then we propose a simple modification of the post-spike history filter so that the model is only allowed to incorporate a limited number of spikes in the history instead of all spikes in the past. This makes the fitted model more stable while achieving similar goodness-of-fit. The new model can also be more flexible for modeling complicated spike train patterns, such as bursting firing.

In Chapter 3, we aim to bridge biophysical modeling and statistical modeling by describing how ion channel conductance can affect spike train patterns. The stimulus filter and the post-spike history filter in the point process model characterize how a neuron responds to an external stimulus or post-spikes at certain time-lags. The coefficients of those filters are selected as the features of spike train patterns. The goal of the model is to construct the spike train features as functions of the ion channel conductances, so the impact of the channel conductance can be quantified. We find different types of ion channels influence the spike train patterns in different ways. This may reveal their different roles in information encoding.

To better characterize information processing in the brain, it is important to identify situations in which neural activity is coordinated across populations of neurons. In Chapter 4, we analyze covariation of population firing rates within three visual areas, focusing on the timing of peak firing rate. For these data, a naïve method of determining the time of peak firing rate is too noisy: it can not find covariation across areas. Instead, we demonstrate strong cross-area covariation using the point process framework that allows firing rate curves to vary with experimental condition and furthermore allows neurons to participate in population activity under some conditions but not others. Because our approach is multivariate, it has the additional benefit of assessing pairwise covariation conditionally on (after "partialling out") activity of the third area. Results concerning multi-way dependence can constrain theoretical conceptions of circuit operation. Another motivation is to describe functional diversity and specialization of neurons that is relevant to cross-area coordinated activity. We estimate the proportion of recorded neurons that participate in this kind of population activity, and we indicate their cortical depths.

Chapter 5 focuses on the spike-to-spike coupling effect on fine timescale (can be within 20 ms or less), especially between neurons in different regions. We build a flexible, extendable, robust, and computationally efficient tool to quantify the spike-to-spike coupling effect; it can handle hundreds of neurons that are simultaneously recorded by high-density multi-electrode arrays. Our proposed point process regression model can be modularized into two basic components: one part quantifies the coupling effect using a continuous function in a lag range; the other part is responsible for removing the artifacts caused by the background activity. The small number of parameters in the model and optimization-based inference make it efficient for large dataset analysis. We verified the model using many simulation scenarios and some theoretical analysis, then applied the method to the Allen Brain Observatory dataset and discovered trial-to-trial variation of coupling effects on fine timescale.

Chapter 2

Stability of fitted point process spiking neuron models

This is a collaborative work with Qi Xin, Valérie Ventura and Robert E. Kass. It has already been published in [30].

Point process regression models, based on generalized linear model (GLM) technology, have been widely used for spike train analysis, but a recent paper by Gerhard et al. described a kind of instability, in which fitted models can generate simulated spike trains with explosive firing rates [53]. We analyze the problem by extending the methods of Gerhard et al. First, we improve their instability diagnostic and extend it to a wider class of models. Next, we point out some common situations in which instability can be traced to model lack of fit. Finally, we investigate distinctions between models that use a single filter to represent the effects of all spikes prior to any particular time t, as in a 2008 paper by Pillow et al., and those that allow different filters for each spike prior to time t, as in a 2001 paper by Kass and Ventura [77]. We re-analyze the data sets used by Gerhard et al., introduce an additional data set that exhibits bursting, and use a well-known model described by Izhikevich to simulate spike trains from various ground truth scenarios. We conclude that models with multiple filters tend to avoid instability, but there are unlikely to be universal rules. Instead, care in data fitting is required and models need to be assessed for each unique set of data.

2.1 Introduction

Point process regression models based on the framework of generalized linear models (GLMs) have been applied to a wide variety of spiking neuron data ([79], [137], and references therein). These models, which may be considered nonlinear Hawkes processes [28, 44], allow neural firing rates to depend on spiking history. Recently, however, [53] reported that models fitted to real data sets could be unstable in the sense that their firing rates could evolve to become arbitrarily large, generating unrealistic spike trains, even when standard goodness-of-fit tests fail to identify lack of fit (see Figure 2.1 for two examples). In this paper we identify several factors that can lead to this problem, we provide additional analysis for diagnosing it, and we present methods to improve model stability.

In some circumstances, causes of instability are easy to identify and easy to fix. The problem of stability, however, leads naturally to an interesting detail in GLM-type modeling of spike trains. When spike trains are modeled as point processes, the firing rate is defined by the conditional intensity function

$$\lambda(t|H_t) = \lim_{\Delta t \to 0} \frac{\mathbb{P}\big(\Delta N_{(t,t+\Delta t]} = 1|H_t\big)}{\Delta t} \tag{2.1}$$

where H_t is the set of spikes prior to time t, known as the spiking history up to time t, and $\Delta N_{(t,t+\Delta t]}$ is the number of spikes in the interval $(t, t + \Delta t]$. This succinct representation can also incorporate stimulus effects and coupling effects and its implementation can take advantage of a large body of knowledge about generalized regression models [79]. Here we only consider the history effects without external stimulus and coupling neurons. There are many ways to capture the effects of the history H_t on the intensity. Letting t_{j*} be the jth spike time counting backwards prior to time t, a concise and intuitive assumption, for steady-state scenarios (where the baseline rate is constant), takes the intensity to have the form

$$\log \lambda(t|H_t) = \beta_0 + \sum_j h(t - t_{j*}) \tag{2.2}$$

where $h(u)$ is a smooth function, and the summation extends to all spikes that precede time t (within a given trial, if there are trials). This is the form used by [111] and by [53]. [111] referred to $h(u)$ as a post-spike filter. An alternative model, used by [77], instead allows the effects of each previous spike to be different:

$$\log \lambda(t|H_t) = \beta_0 + \sum_{j=1}^{k} h_j(t - t_{j*}). \tag{2.3}$$

If each function h_j involves separate free parameters, then the model in (2.3) would typically have more parameters than the model in (2.2). A main contribution of this paper is to describe situations under which this additional flexibility can be useful. In particular, we suggest that, in realistic scenarios, models of the form (2.3) tend to be stable.

One issue in using (2.3) is that the number of terms k must be selected. In theory the number could be infinite as long as a suitable condition is placed on the functions h_j, such as $\sum M_j < \infty$, where $M_j = \max_u h_j(u)$, but in practice [77] selected k by applying the likelihood ratio test; here, in Section 4, we will suggest another criterion, based on stability. Similarly, in practice, the summation in (2.2) extends over a fixed window of time preceding t, having length we label T_h (at most, the total length of the experiment), leading to the alternative representation

$$\log \lambda(t|H_t) = \beta_0 + \sum_{t_{j*} \in (t - T_h, t]} h(t - t_{j*}). \tag{2.4}$$

We refer to a model described by (2.4) as Fixed Length Filter (FLF), and those described by (2.3) as Fixed Number Filter (FNF), where the fixed number refers to the fixed number of spikes. In our analysis we have found it useful to further categorize FNF models by considering the special

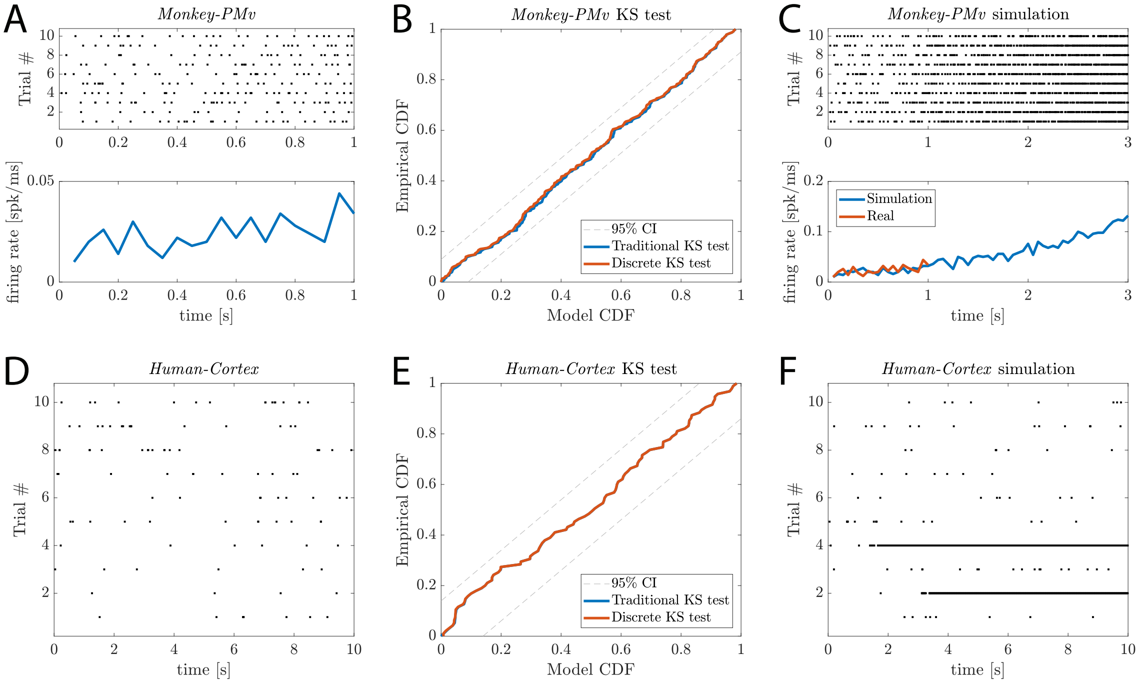


Figure 2.1: Simulation divergence of FLF models (equation 2.4) fitted to data. (A,D) Spike time raster plot and PSTH for datasets *Monkey-PMv* and *Human-Cortex*. (B,E) The fitted FLF models pass the original and discrete KS tests of [24] and [62]; the two tests overlap so they are hard to distinguish. (C,F) Spike time raster plot and PSTH of spike trains simulated from the fitted FLF models, using algorithm 3 in Appendix A.3. (C) If the simulation lasts longer than the training session, the firing rate keeps growing to produce ISIs that are shorter than the refractory period. (F) The simulated spike trains resemble the observed data except for trials 2 and 4, which have many more spikes than the observed spike trains.

case in which $h_j(u) = h_1(u)$, for all j and all u. These models we write these as $\mathrm{FNF_S}$, where S stands for single filter. The more general case we write as $\mathrm{FNF_M}$, with M for multiple. Note that $\mathrm{FNF_S}$ differs from FLF in that the number of spikes is fixed rather than the length of the time interval, but both models use a single filter while $\mathrm{FLF_M}$ uses multiple filters.

We begin, in Section 2, by giving some analytical stability results along the lines of those in [53]. In Section 3 we identify several kinds of model mis-specification that lead to instability, and we note potential solutions. In Section 4 we focus on FNF models and the variation that replaces the constant β_0 with a time-varying function $\beta(t)$. An analytical diagnostic method is then used to select the number of previous spikes to be considered in the model, i.e., the number of terms k in (2.3). In section 5, we compare FLF and FNF models. We close in Section 6 with advice on the use of these models.

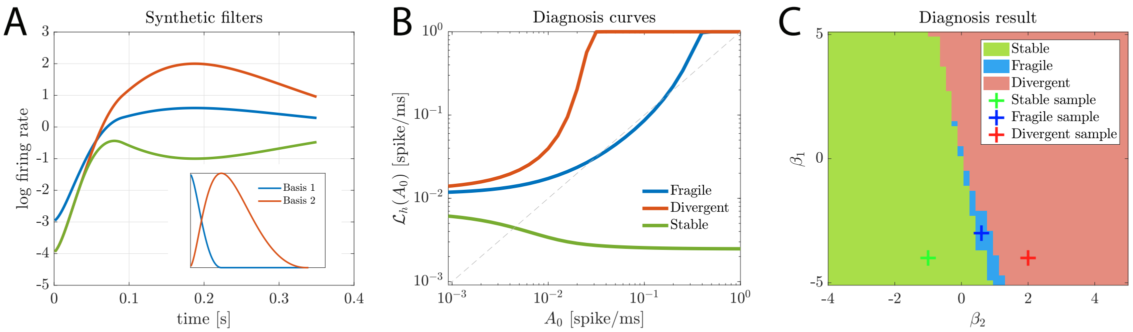


Figure 2.2: (B) Three diagnostic curves corresponding to the three FLF models (equation 4) with baseline rates $\beta_0 = -4$ and filters $h(t)$ shown in (A) in matching colors, where $h(t) = \beta_1 \cdot \mathrm{B}_1(t) + \beta_2 \cdot \mathrm{B}_2(t)$, $\mathrm{B}_1(t) = e^{-t/0.02}$ and $\mathrm{B}_2(t) = e^{-t/0.1}$ are the smooth basis functions used in [111] and shown in the insert in (A), and $T_h = 0.35$ sec. The values of (β_1, β_2) for the three models are marked as crosses in (C). (C) Diagnostic map for the above model as β_1 and β_2 vary.

2.2 Stability analysis

Figure 2.1 shows two examples of unstable simulations from FLF models (equation 2.4) fitted to the *Monkey-PMv* and *Human-Cortex* datasets described in Table A.1, Appendix A, using the smooth basis method of [111]. The fitted FLF models pass the original and discrete KS goodness of fit tests [24, 62] but they are unstable, in the sense that the firing rates of some or all the simulated spike trains evolve to become arbitrarily large, generating unrealistic spike trains. We emphasize that instability is not a matter of extrapolation to unseen data outside the experimental range of time. Rather, if a model is unstable, in simulations it can evolve to producing firing rates far in excess of those seen in real data, which makes it patently unrealistic as a representation of neural physiology.

[53] argue that a reliable diagnostic of model instability can be obtained from the relationship between the firing rate A_0 before the last spike at t_{1*} in an interval, and the firing rate after t_{1*}. They approximate A_0 with the average firing rate in the interval $(t - T_h, t_{1*})$. Then they rewrite the fitted FLF model (Equation 2.4) as

$$\log \lambda(t|H_t) = \beta_0 + h(t - t_{1*}) + \sum_{t_{j*} \in (t - T_h, t_{1*})} h(t - t_{j*}),$$

and approximate the summation by its expectation under the assumption that the point process in the interval $(t - T_h, t_{1*})$ is homogeneous Poisson, yielding

$$\log \lambda(t|H_t) \approx \beta_0 + h(t - t_{1*}) + A_0 \int_{t - t_{1*}}^{T_h} \left(e^{h(u)} - 1\right) \mathrm{d}u. \tag{2.5}$$

[53] then use (2.5) to derive the approximate PDF of the future ISI $t_* - t_{1*}$, where t_* is the time of the next spike after t_{1*}, and calculate the firing rate after t_{1*} as the reciprocal of the mean future ISI:

$$\mathcal{L}_h(A_0) = \frac{1}{\mathbb{E}[t_* - t_{1*}]}. \tag{2.6}$$

Equation 2.6 is a function of A_0 because Equation 2.5 is a function of A_0. The instability diagnostic is obtained by plotting $\mathcal{L}_h(A_0)$ versus A_0, as in Figure 2.2B, for $A_0 \in [0, \lambda_{max}]$, where λ_{max} is the maximum possible firing rate. Without loss of generality, in this paper we do not build a refractory period in the models we consider, except to reproduce [53] Figure 4 (see Appendix A.2 Figure A.1), so λ_{max} is our simulation resolution of 1000 spikes per second. We examine intersections of the diagnostic curve with the secant line $\mathcal{L}_h(A_0) = A_0$, referring to them as *cross points*.

A model is deemed

- **divergent** if all cross points exceed λ_{thr} or $\mathcal{L}_h(A_0)$ is always larger than A_0 (e.g. Figure 2.2B, red curve), where λ_{thr} is a threshold rate judged too high physiologically; here we used $\lambda_{thr} = 0.9 \cdot \lambda_{max}$ spikes/sec;
- **stable** if the number of cross points is odd and they are all below λ_{thr} (green curve);
- **fragile** otherwise (blue curve).

Note that the diagnostic green curve in Figure 2.2B exceeds the first secant for small values of A_0, which is desirable because otherwise the firing rate would eventually decrease down to zero. A divergent model yields unstable simulations, whereas spike trains simulated from a fragile model might first look stable and then degenerate. Therefore the difference between divergent and fragile models is the duration it takes for spike trains to become unstable. Without loss of generality of our results, we do not distinguish between divergent and fragile models, and consider them both unstable.

[53] validated their model stability diagnostic against spike train data: they considered a model family F_θ parametrized by θ, and for each value of θ in a range, they (i) simulated a 10 sec. long spike train from F_θ (e.g. using algorithm 3 in Appendix A.3), and deemed the spike train unstable if the model generated over $0.9 \cdot \lambda_{max}$ spikes in the last second, (ii) produced the diagnostic curve and determined from it if the model was stable, fragile, or divergent and (iii) plotted θ against the outcomes in (i) and (ii). Figure A.1, Appendix A.2, shows these plots for two model families F_θ. Figure A.1A shows the same stability map as in [53] Figure 4, where F_θ is an FLF model (Equation 2.4) with baseline rate $\beta_0 = -5.3$ and filter $h(t) = \beta_1 \cdot \mathrm{B}_1(t) + \beta_2 \cdot \mathrm{B}_2(t) + \text{Dip(t)}$, $\theta = (\beta_1, \beta_2)$, $\mathrm{B}_1(t) = e^{-t/0.02}$ and $\mathrm{B}_2(t) = e^{-t/0.1}$ are the smooth basis functions used in [111] and shown in the insert in Figure 2.2A, Dip(t) is a negative window function modeling a 2 msec. refractory period, and the filter length is $T_h = 0.2$ sec. In Figure A.1C, F_θ is an FLF model with baseline rate $\beta_0 = -4$, filter $h(t) = \beta_1 \cdot \mathrm{B}_1(t) + \beta_2 \cdot \mathrm{B}_2(t)$ with $B_1(t)$ and $B_2(t)$ defined above, and filter length $T_h = 0.35$ sec.

The stability maps in Figure A.1A,C suggest that the diagnostic is mostly reliable, except in small regions of the parameter spaces. This happens because [53] replaced $h(u)$ by the Taylor expansion $(\exp h(u) - 1)$ in (2.5), which is accurate only when $h(u)$ is small. Without this approximation, (2.5) becomes

$$\log \lambda(t|H_t) \approx \beta_0 + h(t - t_{1*}) + A_0 \int_{t-t_{1*}}^{T_h} h(u)\mathrm{d}u, \tag{2.7}$$

and the diagnostic is still tractable, as shown in Appendix D. Figure A.1B,D show the updated

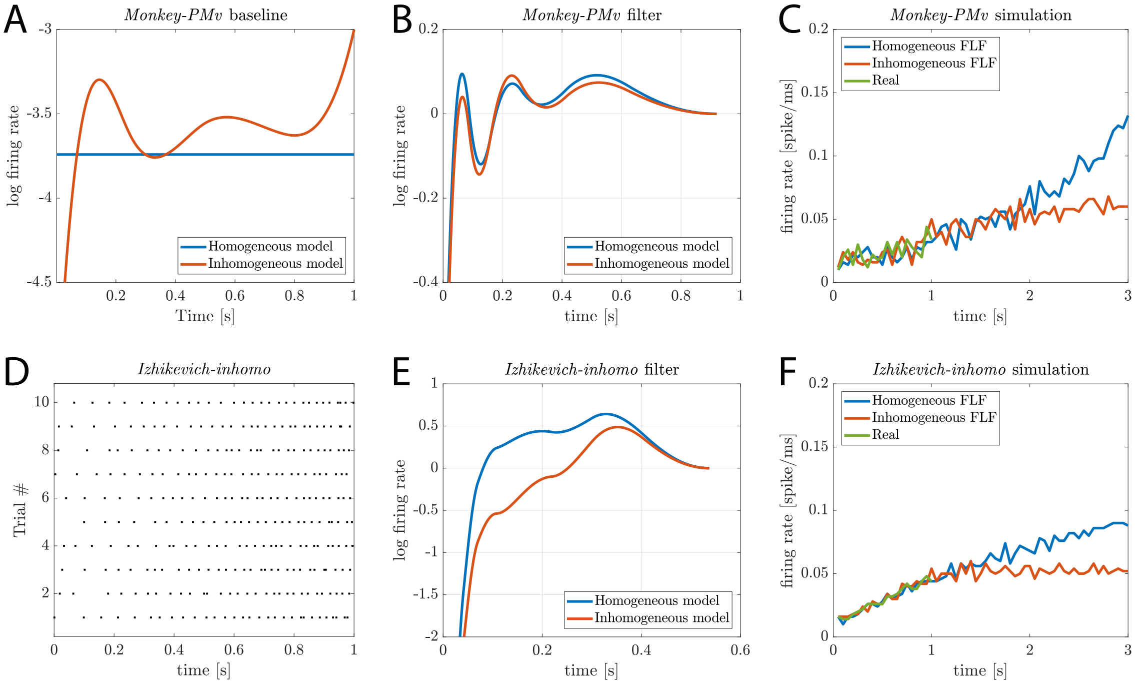


Figure 2.3: Stability of FLF models (equation 2.4) with constant and time varying baseline firing rates. (A) Constant (blue) and time varying (red) baseline rates of FLF models fitted to the *Monkey-PMv* data: the baseline appears to vary. A likelihood ratio test confirms that the time varying baseline model fits the data significantly better (p = 0.046). (B) Fitted filters of the constant and time-varying baseline models. (C) PSTHs of the observed data and of data simulated from the fitted homogeneous and inhomogeneous FLF models: the homogeneous model simulates unstable spike trains; the inhomogeneous model is simulation stable. (D, E, F) Same analysis applied to artificial data generated from an inhomogeneous *Izhikevich* model (algorithm 2 in Appendix A.3). (D) Data raster plot. (E) Fitted filters of the homogeneous and inhomogeneous baseline FLF models: the latter is mostly below the former, which may reduce the chance of unstable simulated spike trains. (F) Indeed, the homogeneous model produces data whose rate diverges; the inhomogeneous model appears to be stable. Note that the inhomogeneous model fits the data significantly better according to a likelihood ratio test ($p \ll 0.001$).

stability maps based on (2.7). The agreement between diagnostic and simulation is very close, and closer than in Figures A.1A,C, so we use the updated diagnostic in the rest of the paper. Figure A.1D is reproduced in Figure 2.2C.

To solve the stability problem when a model is found to be divergent or fragile, [53] suggest stabilizing it by refitting to the data with the constraint that its parameters lie in the stable region of the parameter space. In the next section, we identify three data features that might lead to unstable simulation models, namely small sample size, time varying firing rates, and trial to trial variability or outlier trials, and we provide alternative suggestions for stabilization: collecting more data, fitting inhomeogeneous rate models, and removing outliers, respectively.

2.3 Special cases of FLF model instability

A feature that might lead to unstable models is a small sample size. Indeed fitting a model to a small dataset yields parameter estimates that have large variances and, therefore, that could lie in unstable regions of the parameter space by chance, even if the true parameters lie in stable regions. Collecting more data, if possible, would reduce the variability of parameter estimates and stabilize the model.

Next, consider the *Monkey-PMv*, shown in Figure 2.1A. An FLF model fitted to the data satisfies the KS goodness-of-fit tests (Figure 2.1B), yet simulations from the model diverge (Figure 2.1C). Because the peri-stimulus time histogram (PSTH) in Figure 2.1A appears to increase, we fit a time-varying baseline rate $\beta(t)$ in place of β_0 in (4). That model fits the data somewhat better according to a likelihood ratio test ($p = 0.046$), and data simulated from it do not diverge (Figure 2.3C). (To simulate data past the maximum experimental time of one second, we set $\beta(t) = \beta(1)$ for $t \geq 1$ sec.) Figure 2.3D,E,F shows a similar outcome when we apply the same analysis to synthetic data generated from an inhomogeneous *Izhikevich* model (algorithm 2, Appendix B). Hence, in the presence of a time-varying trial-averaged rate, fitting a constant rate term can produce instability and fitting a time-varying rate can rectify the problem.

Finally, consider the *Human-Cortex* data displayed in Figure 2.1D. An FLF model fitted to the data satisfies the KS goodness-of-fit tests (Figure 2.1E) but two out of ten spike trains simulated from it diverge (Figure 2.1F). Figure 2.4A shows that trials 8, 9, and 10 have rather large spike counts compared to the others, so there might be excess trial-to-trial variability or outlier trials that might cause the instability. To examine the extent to which some trials may be unusually different than others, we compute the distance of each spike train from a central spike train $\overline{ST}$ (defined below) based on a spike train metric devised by [139]. This metric measures the discrepancy between two spike trains by counting the number of spikes in one spike train that can be matched by spikes in the other spike train using a smooth deformation of time, or "time-warping function." If there are N_1 and N_2 spikes in two spike trains ST_1 and ST_2, the distance between the two spike trains is defined as

$$d(ST_1, ST_2) = \inf_{\gamma \in \Gamma} \left(N_1 + N_2 - 2 \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} I_{[t_i = \gamma(s_j)]} + \eta \int_0^T \left(1 - \sqrt{\gamma'(t)}\right)^2 \mathrm{d}t \right), \tag{2.8}$$

where I is the indicator function, t_i and s_j are spike times from ST_1 and ST_2, respectively, and $\Gamma(t)$ is the set of all continuous and piecewise differentiable time warping functions γ such that

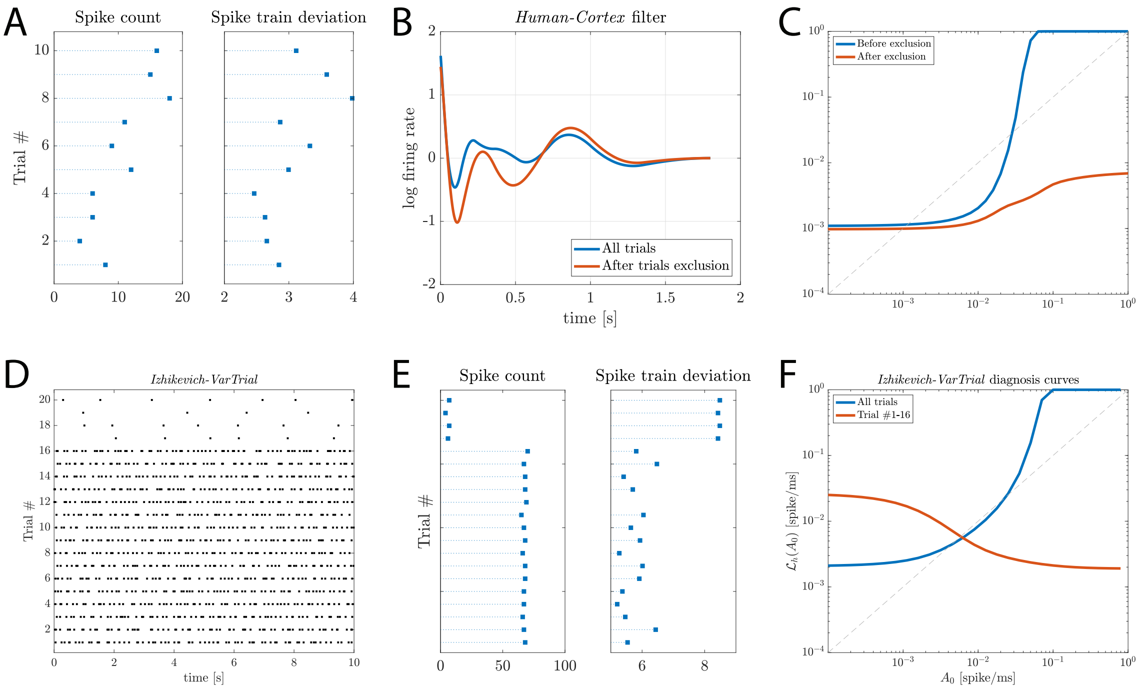


Figure 2.4: Trial-to-trial variability affects simulation stability. (A) Spike train counts and deviations from the central spike train for the *Human-Cortex* dataset. Trials 8 and 9 might be outliers. (B) Filters of FLF models (Equation 2.4) fitted to data with and without suspected outliers. The latter remains mostly below the former for all t, which reduces the possibility of simulating unstable spike trains. (C) Diagnostic curves for models fitted before and after removing the suspected outliers: the model becomes stable after removal. Synthetic spike trains simulated from that more are indeed stable (not shown). (D,E,F) Same analysis applied to spike trains generated from a two-rate *Izhikevich* model. (D) The dataset is composed of 16 spike trains with a high firing rate, and four with a low firing rate. (E) spike counts and deviations from the central spike train clearly identify two groups of spike trains. (F) Diagnostic curves of FLF models fitted to the full datasets (blue), and to the dataset after the four unusual spike trains are removed. The former diagnoses an unstable model, the latter a stable model. Spike trains simulated from the latter model are indeed stable (not shown).

$\gamma(0) = 0$, $\gamma(T) = T$, and $0 < \gamma'(T) < \infty$. In practice γ is approximated by a piecewise linear function from $[0, 0]$ to $[T, T]$ in a discrete grid and the tuning parameter η is set to $(N_1+N_2)\cdot c/2T$, with $5 \leq c \leq 25$ [139]. The choice of c in this range has little impact on results. The first term on the right hand side of (2.8) measures how close ST_1 is to the time wrapped ST_2, and the second penalizes the deviation of the time warping transformation from the identity function $\gamma(t) = t$. The central spike train $\overline{ST}$, is defined as

$$\overline{ST} = \arg\min_{C \in \mathcal{S}} \sum_{i=1}^{n} d(ST_i, C),$$

where $\mathcal{S}$ is the set of all spike trains. We then compute each distance

$$d_i = d(\overline{ST}, ST_i), \quad i = 1, \ldots, n$$

and use d_i to identify unusually discrepant trials.

The deviations d_i for the *Human-Cortex* spike trains are shown in Figure 2.4A. Trials 8 and 9 have the largest values of d_i, and they also have large spike counts. After removing them, the fitted FLF model becomes stable, according to the diagnostic plot in Figure 2.4C. Figure 2.4B shows that the filter fitted after excluding the outliers lies mostly below the initial filter, which reduces the chance of simulation divergence. We note that if we remove only one of these trials the fitted model is again unstable. Furthermore, if we remove any other 2 trials the fitted model is unstable. Figure 2.4D,E,F shows that a similar analysis applied to data generated from *Izhikevich* models with two different firing rates – 16 spike trains have a large firing rate and four have a small firing rate – yields similar conclusions: that is, outlier trials can destabilize models, and careful data pre-processing to remove them might improve stability.

Different kinds of outliers may have to be treated differently. Outlier trials resulting from bad recordings should be removed. But absent such experimental difficulties it remains important to consider unusual features of the data, and to avoid models that fail to account for those features. In the synthetic two-rate *Izhikevich* dataset, for example, four trials are noticeably sparse, making a common rate model fit poorly. Methods based on models that allow for excess trial-to-trial variability are available [135].

2.4 Stability of FNF models

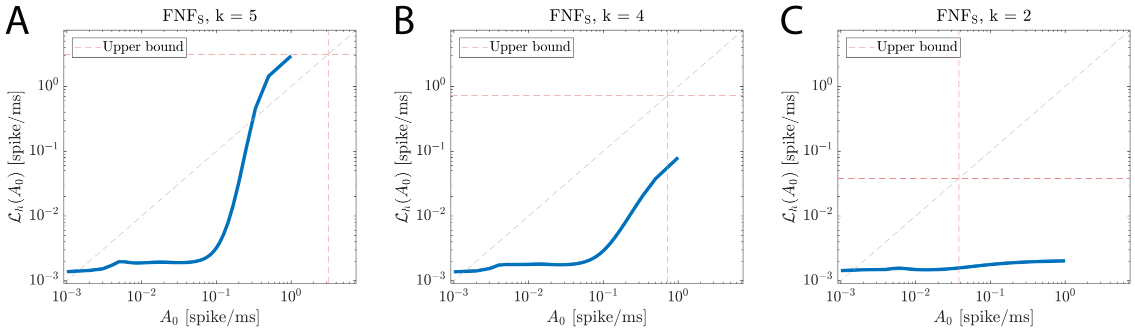


Figure 2.5: Diagnostic curves of FNF$_S$ models with $k = 2, 4, 5$ fitted to the *Human-Cortex* dataset. The dashed red lines are the models' maximum firing rates. The largest model with $k = 5$ is unstable because the diagnostic curve is above the first sequent after it last intersects it. The two other models are stable.

To evaluate the performance of FNF models, we extend the method of [53] to obtain a stability diagnostic, further allowing the baseline rate to be a time-varying function $\beta(t)$, as in [77]. If we approximate the firing rate before the last spike t_{1*} with the reciprocal of the mean ISI, $A_0 = 1/\tau$, and replace the ISIs by their expectation in (2.6), we obtain:

$$\log \lambda(t|H_t) \approx \beta(t) + \sum_{k=1}^{k} h_j(t - t_{1*} + (k-1)\tau). \tag{2.9}$$

As in Section 2, we then use this approximation to derive the approximate PDF of the future ISI $t_* - t_{1*}$, and calculate the firing rate $\mathcal{L}_h(A_0)$ after t_{1*} as the reciprocal of the mean future ISI (see (2.6)). The diagnostic for a fitted model is again based on a plot of $\mathcal{L}_h(A_0)$ against A_0, and its stability determined using the rules in the boxed text in Section 2. For example, Figure 2.5 shows the diagnostics of three FNF$_S$ models fitted to the *Human-Cortex* dataset described in Table A.1, using the smooth basis in [111]. The largest model (panel A) is unstable, in the sense that spike trains generated from that model could have unrealistically large number of spikes; the other two models are stable.

Just as in [53], we can validate our FNF model stability diagnostic against spike train data. For example, Figure 5 shows the stability map for the FNF$_S$ family of models with firing rate $\log \lambda(t|H_t) = \beta_0 + \sum_{j=1}^{5} h(t - t_{j*})$, where $\beta_0 = -4$, $h(t) = \beta_1 \mathrm{B}_1(t) + \beta_2 \mathrm{B}_2(t)$, and $\mathrm{B}_1(t) = e^{-t/0.02}$ and $\mathrm{B}_2(t) = e^{-t/0.1}$ are the basis functions shown in the inset of Figure 2A. For each value of (β_1, β_2), we (i) simulated a 10 sec. long spike train from the model and deemed the model divergent if it generated over 900 spikes in the last second, (ii) produced the diagnostic curve and determined from it if the model was stable, fragile, or divergent, and (iii) plotted (β_1, β_2) against the outcomes in (i) and (ii): the two match, which suggests that our diagnostic is reliable. The many FNF$_M$ models we investigated also suggest that the diagnostic is reliable; we did not provide an example diagnostic map here because all these models were stable across the entire parameter space.

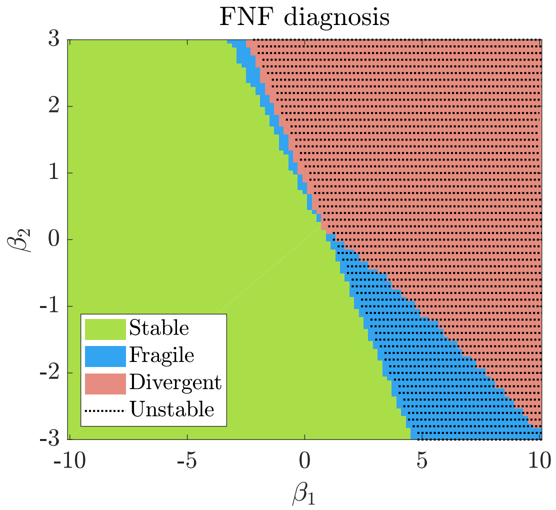


Figure 2.6: Stability map for FNF$_S$ models with firing rates $\log \lambda(t|H_t) = \beta_0 + \sum_{j=1}^{5} h(t - t_{j*})$, where $\beta_0 = -4$, $h(t) = \beta_1 \mathrm{B}_1(t) + \beta_2 \mathrm{B}_2(t)$, $\mathrm{B}_1(t) = e^{-t/0.02}$ and $\mathrm{B}_2(t) = e^{-t/0.1}$. For each value of (β_1, β_2), we (i) simulated a 10 sec. long spike train from the model and deemed the model unstable if it generated over 900 spikes in the last second, (ii) produced the diagnostic curve and determined from it if the model was stable, fragile, or divergent, and (iii) plotted (β_1, β_2) against the outcomes in (i), with unstable simulations indicated by black dots, and (ii), in colors. Our diagnostic is reliable because it matches the simulation well.

Figures 2.5 and 2.7C,F show that all FNF$_S$ models with $k \leq 4$ and all FNF$_M$ models fitted to the *Human-Cortex* dataset are stable. They also all pass the two KS tests so they are not obviously deficient. With many simulation stable models available, it may be desirable to choose one that also fits the data best according to some criterion. Figure 2.7C,F shows the Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) for all these models. Small AIC and BIC values are desirable because AIC is an estimate of prediction risk, and BIC is inversely related to the posterior probability of fitting the correct model, in an asymptotic sense; BIC tends to prefer models with fewer parameters [79]. Both criteria suggest that the FNF$_S$ model with $k = 1$ fits best; it is also simulation stable. Figure 2.7 shows results from two additional examples. Among the stable models fitted to the bursty *Goldfish* dataset, the FNF$_M$ model with $k = 6$ filters fits best based on AIC, and FNF$_S$ model with $k = 9$ fits best based on BIC. The synthetic *Izhikevich-burst* dataset is bursty as well; see Figure A.2A. Its best simulation stable models are FNF$_M$ models with $k = 7$ and $k = 4$ according to AIC and BIC, respectively.

To summarize, our general strategy is to fit FNF models for several values of k and choose a model that is simulation stable and also fits the data well according to criteria such as the KS tests, and AIC or BIC. (Note that because AIC and BIC are obtained from a finite data sample, they have variability, so that similar values should be considered equal.) If no stable model can be found, a model that provides a good fit may be used after constraining its parameters to lie in the parameter subspace corresponding to stability, as [53] suggest.

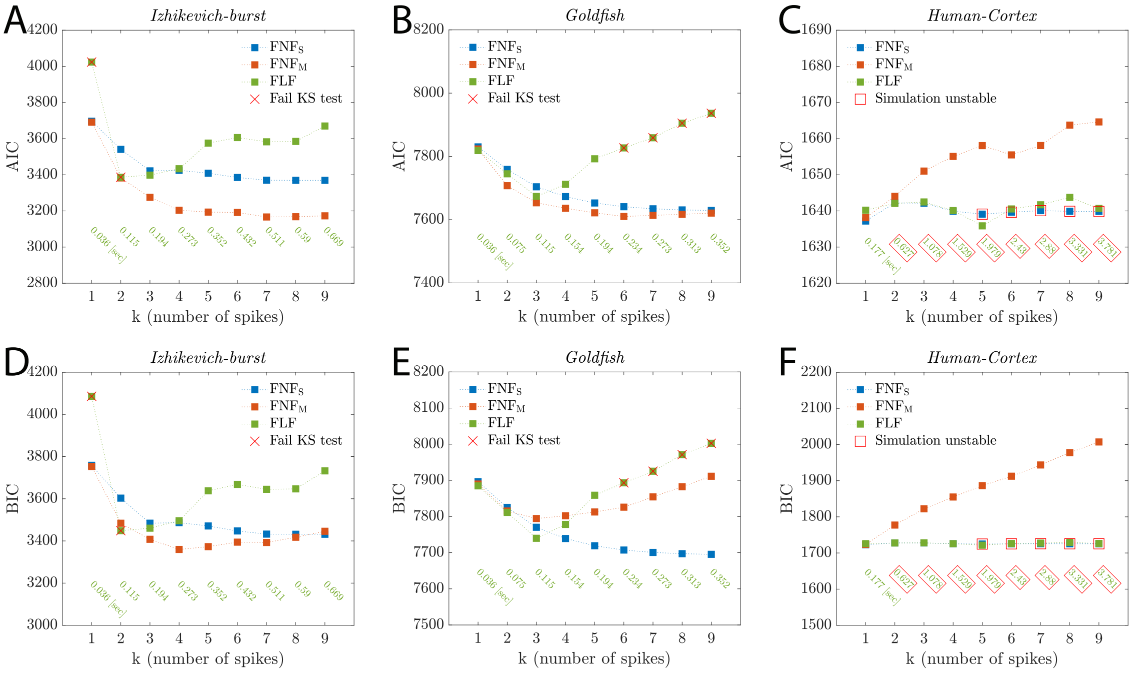


Figure 2.7: (A,B,C) AIC and (D,E,F) BIC values for several FLF, FNF$_S$, and FNF$_M$ models fitted to three datasets. (FNF$_S$ and FLF values are equal in panel F, and the latter mask the former.) Simulation unstable models and models that failed the original and/or the discrete KS test are indicated by red squares and crosses. Many models are simulation stable; the FNF$_M$ models fitted here are all simulation stable. Stable models that achieve a desirable criterion, e.g. low AIC, could be chosen.

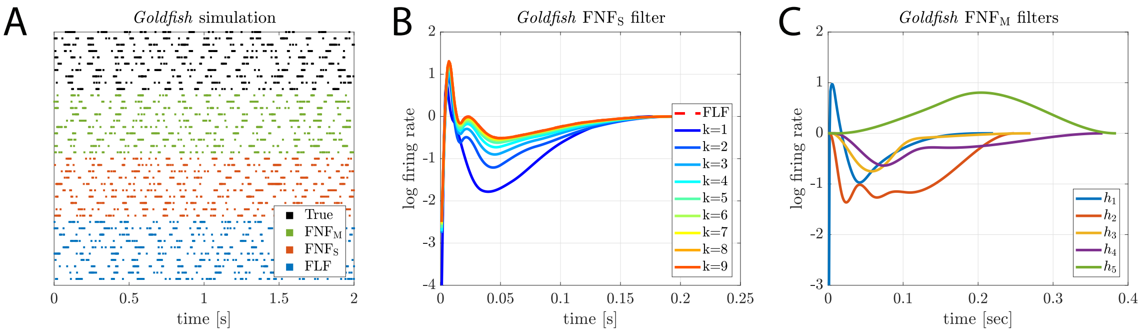


Figure 2.8: (A) *Goldfish* dataset spike trains and simulated spike trains from FLF, FNF$_S$ ($k = 5$), and FNF$_M$ ($k = 5$) models fitted to the dataset: despite modeling the effects of past spiking differently, both type of models can generate busts similar to those in the dataset. (B) Filters of the FNF$_S$ models fitted to the *Goldfish* dataset with past number of spikes $k = 1, ..., 9$. When $k \geq 5$, the filters are almost identical, suggesting that effects of spikes prior to 5 spikes back do not contribute much to the fits. The fitted FLF filter has length $T_h = 0.19$ sec., which contains 4 past spikes on average; it overlaps with the FNF$_S$ filters with $k \geq 5$, suggesting that these models are functionally similar (although FNF$_S$ models are more likely to be stable). (C) Fitted filters of an FNF$_M$ model with $k = 5$: the filters are substantially different, suggesting that burst behavior is captured by differentially weighing the contribution of previous spikes according to their timing and ordering. A likelihood ratio test comparing the FNF$_S$ and FNF$_M$ models with $k = 5$ strongly favors the latter ($p \ll 0.001$).

2.5 Comparison of FNF and FLF models

A key feature of FLF models is that they sum the effect of all spikes in the filter window of length T_h, which puts no limitation on the number of past spikes influencing the firing rate at t, even if T_h is short. Therefore, if a fitted intensity function has a rising trend, an increasing number of spikes could fall within the filter window, and this could increase the firing rate, eventually yielding unstable simulated spike trains. In contrast, FNF models (Equation 2.3) and also the extension to time-varying baseline rates) model history with a fixed number of spikes, k, and if the baseline rate $\beta(t)$ and all of the individual filters are bounded above, the firing rate will be bounded. Furthermore, by allowing multiple filters, $\mathrm{FNF_M}$ models can diminish the effects of multiple spikes that occur, somewhat infrequently, in close temporal proximity. Thus, in principle, FNF$_M$ models tend to be stable, and we did not find any cases in which $\mathrm{FNF_M}$ models were unstable. See, for example, the results in Figure 2.7. However, we have also seen that, in some cases, the fitted FNF$_S$ firing rates can be large enough to become unstable (see Figures 2.5A and panels C and F of Figure 2.7), and for that reason we developed a stability diagnostic and a strategy to stabilize a divergent FNF model in Section 4. We could apply a similar strategy to FLF models, using several filter window lengths T_h in place of several values of k. [53] fitted an FLF model with $T_h = 0.35$ sec to the *Human-Cortex* dataset, which was unstable. Figure 2.7C,F shows the stability status, AIC, and BIC values of fitted FLF models for several values of T_h. (FLN and FNF$_S$ models have similar AIC and BIC values that are hard to distinguish from one another on the plots.) The FLF model with a very short filter length of $T_h = 0.177$ fits the data well (it passes both KS tests and has smallest AIC and BIC) and is stable. Thus, while FNF$_M$ models seem to be inherently less likely to be unstable, we can not make any universal comparative statement about stability, and, importantly, fitting with either type of model requires care. A remaining issue is whether there are interesting cases in which the additional flexibility of FNF$_M$ models is useful. We now present a few additional comparative results.

Figure 2.7 provides a summary of fits for the *Human-Cortex* dataset. We see that FNF$_S$ models have AIC and BIC values similar to, or smaller than FLF models. The FNF$_M$ models have higher AIC, presumably because the additional flexibility of using several filters is not needed to fit the data well yet it increases complexity. On the other hand, the FNF models fitted to the *Izhikevich-burst* data set have smaller AIC and BIC values than the FLF models, and FNF$_M$ models have smaller AIC and nearly all smaller BIC values than FNF$_S$ models, presumably because the data are bursty and thus are not fitted adequately with simpler models. We also used a real data set, labeled *Goldfish*, which consists of recordings from retinal ganglion cells *in vitro* that exhibit bursting firing [96, 132]; see Table 1 and Figure 8A. The AIC and BIC values are again smaller for most FNF models.

These comparisons are substantiated in Figures 2.8 and A.2. Figure 2.8A displays the *Goldfish* data spike trains together with simulated spike trains from a FLF model fitted to the data, having filter length $T_h = 0.19$ seconds containing five past spikes on average, as well as from fitted FNF$_S$ and FNF$_M$ models with $k = 5$: despite modeling the effects of past spiking differently, both type of models can generate busts similar to those observed in the data. Figure 2.8B displays the filters of FNF$_S$ models fitted to the *Goldfish* dataset with past number of spikes $k = 1$ to 9. When $k \geq 5$, the filters are almost identical, suggesting that effects of spikes prior to 5 spikes back do not contribute much to the fits. The overlayed fitted FLF filter with $T_h = 0.19$ sec. overlaps with the FNF$_S$ filters with $k \geq 5$, suggesting that these models are functionally similar (although, based on our previous analysis, FNF$_S$ models are more likely to be stable).

Models with a single filter, as in Figure 2.8B, assume that the effect of any past spike t_{j*} on the firing rate at time t depends only on the elapsed time $t - t_{j*}$ without considering the number of spikes that may have occurred between t_{j*} and t. For the *Goldfish* data, this is a questionable assumption: a likelihood ratio test (LRT) comparing FNF$_S$ and FNF$_M$ models with $k = 5$ strongly favors the latter ($p \ll 0.001$). Furthermore, the fitted filters of the FNF$_M$ model, shown in Figure 2.8C, are substantially different, suggesting that burst behavior is captured better by differentially weighing the contribution of previous spikes according to their timing and ordering. We may interpret these multiple distinct filters by observing several characteristics of the data (see [132]): the average burst length is roughly 25ms, a burst ISI is around 7ms, the median number of spikes in a burst is 4, and bursts occur, on average, roughly every 200 ms. With these in mind, the narrowness and height of the first filter suggests that the effect of the first spike back is strongly influenced by bursting, i.e., when a spike occurs less than 25 ms in the past it is likely that the cell is in a bursting state and the probability of spiking is increased; filters 2 to 4, corresponding to the 2nd to 4th spikes back, diminish the firing rate starting around 25 ms in the past, which presumably signals that when these multiple spikes back are spaced further than 25 ms in the past, the neuron has transitioned to a “down” state; the effect of the 5th spike back is to increase the firing rate after a longer duration, peaking around 200 ms in the past, reflecting an expectation that the neuron has already finished its pause after a burst and has now returned to the bursting state. Figure A.2 in the appendix contains the corresponding plots for the synthetic

Izhikevich-inhomo dataset, from which similar conclusions can be drawn.

In summary, FNF_M models do, sometimes, provide better fits than FLF or FNF_S models, but this is an empirical question that must be answered for each set of data separately. We should also note that models that incorporate hidden burst and non-burst states, as in [132], may provide even better descriptions of bursting spike train data.

2.6 Discussion

We have attempted to provide a thorough analysis of the instability phenomenon identified by [53]. We improved the diagnostic of [53] and extended it to FNF models; we noted that time-varying baseline rates and excess trial-to-trial variability can cause instability of models that do not account for these effects; we introduced a method to detect outlier trials and illustrated its use; and we compared FNF with FLF models in several examples.

It is perhaps worth emphasizing that FNF$_M$ models, with sufficiently large k, were always stable in the examples we investigated, regardless of whether there were stable FLF or FNF$_S$ models. See, for example, Figure 2.7. That figure, together with Figures 8 and 10, also illustrate the way differing variations in spiking behavior may suggest different numbers of filters to use in an FNF model, according to standard model-fitting procedures.

Overall, we concluded that FNF_M models tend to avoid instability, and can provide helpful flexibility in some cases, but we cautioned that selection among the different FLF and FNF models must be done carefully based on the unique characteristics of particular data sets.

Our code is available on `https://github.com/AlbertYuChen/Divergent_Spiketrain_public.git`.

Chapter 3

A biophysical and statistical modeling paradigm for connecting neural physiology and function

This is a collaborative work with Nathan G. Glasgow, Alon Korngreen, Robert E. Kass, and Nathan N. Urban. We will submit the paper to Journal of Computational Neuroscience.

To understand single neuron computation, it is necessary to know how specific physiological parameters affect neural spiking pattern that emerges in response to specific stimuli. Here we present a computational pipeline combining biophysical and statistical models to provide a link between variation in functional ion channel expression and changes in single neuron stimulus encoding. More specifically, we create a mapping from biophysical model parameters to stimulus encoding statistical model parameters. Biophysical models provide mechanistic insight, whereas statistical models can identify associations between spiking patterns and the stimuli they encode. We used public biophysical models of two morphologically and functionally distinct projection neuron cell types: mitral cells (MCs) of the main olfactory bulb, and layer V cortical pyramidal cells (PCs). We first simulated sequences of action potentials according to certain stimuli while scaling individual ion channel conductances. Next, we fit a point process generalized linear model (PP-GLM). A parameter mapping between two types of models can thus be built. In addition, we reconstructed stimulus from the fitted encoding model. This reveals how changes in individual ion channel conductances result in changes in encoding of specific frequency components or stimulus features. This computational pipeline combines models across scales and can be applied as a screen of all channels in any cell type of interest to identify how individual channels influence single neuron computation.

3.1 Introduction

Understanding how the levers that control a cell's physiological properties give rise to single neuron stimulus encoding is a long standing challenge in neuroscience [54]. In this paper, we aim to build a bridge from a cell's biophysical properties to its stimulus encoding properties in a quantitative way.

A cell's physiological and computational properties emerge from biophysical mechanisms such as its membrane properties, ion channel expression and distribution, and morphology. There is considerable variation in both biophysical properties [57, 58, 74, 118, 119] and stimulus encoding properties [5, 6, 58, 109, 119]. This is partially due to variation in ion channel expression [5, 6, 58, 109, 119]. Even in recent patch-seq studies [58, 119], information about what ion channel subtypes are formed and their subcellular distribution is inadequate. This lack of information about functional ion channel expression makes the link to the computational behavior difficult to assess, which is an essential step to understand how variation in observed biophysical building blocks contributes to a diverse and flexible neural code in single cells, circuits, and ultimately behavior.

At present, gathering enough data in experiments to estimate parameters of a detailed biophysical model, ion channel properties and computational model properties are difficult and typically low yield. There are some recent efforts on determining subsets of properties individually through experiments [1, 1, 56, 65, 84, 85], but it is still infeasible to robustly acquire both biophysical and computational properties in the same experiment. So in this study, we employ biophysical simulations using compartmental Hodgkin-Huxley models. This allows full control and interrogation of the underlying mechanisms, as well as an ability to simulate complex responses to arbitrary stimuli. We use these models as an approximation of how a cell would respond to a given stimulus, but with known functional ion channel expression and morphology. We utilize existing templates with rigorous fitting and tuning [1, 84, 85].

Despite their delicate details, biophysical models lack an intuitive interpretation of their computational properties, just like any recording from a real neuron. Statistical models, such as point process generalized linear model (PP-GLM), in contrast, have a simpler set of parameters solely focusing on the encoding process [79, 111, 134]. Nevertheless, statistical models lack mechanistic insight into what drives stimulus encoding patterns. We aim to leverage the strengths of each type of model by mapping their parameters to further our understanding of the link between a cell's biophysical properties to stimulus encoding. A closed-form of such mapping is intractable, so we choose a data-driven strategy. In this paper, we constrained our scope to how variation in individual ion channel conductances relates to stimulus encoding.

In this work, we develop a pipeline and apply it to two morphologically and functionally distinct projection neuron cell types: the mitral cell (MC) of the mammalian main olfactory bulb [14], and the L5 cortical pyramidal cell (PC) [1]. However, this pipeline can be applied to any cell type of interest, given availability of biophysical models, to understand the biophysical mechanisms driving stimulus encoding. How variation in functional ion channel expression impacts computation in a single cell is yet to be determined, let alone how such variation in biophysical properties affects computation across scales in local circuits, between brain regions, and across the brain. Nevertheless, the current pharmacological approach to treating many nervous system disorders is by direct or indirect modulation of biophysical features, namely ion channels. With this pipeline, we aim to fill the gap in understanding how functional ion channel expression contributes to single neuron stimulus encoding and to provide some guidance for experimental studies.

3.2 Methods

The goal of the method is to quantify how ion channels affect stimulus encoding. Biophysical models, like morphologically detailed compartmental Hodgkin-Huxley type models, capture biological mechanisms, but lack clear interpretation of stimulus encoding. Statistical models, like the PP-GLM, represent stimulus response features and incorporate post-spike history in a computationally tractable manner, but lack mechanistic insight [137]. Our method links these two types of models by combining biophysical model output to fit PP-GLM parameters, and then relating PP-GLM parameters to the underlying biophysical parameters. The combined analysis pipeline is depicted in Fig. 3.1. Each portion of the analysis pipeline will be expanded upon in the following sections. We first set up a realistic compartmental Hodgkin-Huxley simulator and proper input signal (Fig. 3.1A). Next, we perform the biophysical simulation to collect the spike trains and repeat the process with different channel conductances (Fig. 3.1B). Last, we jointly train the model using the spikes train with different channel conductances and identify which PP-GLM features are highly influenced by the channel conductances (Fig. 3.1C). Although we have not done so here, the pipeline can be applied on any existing conductance-based biophysical model, and may guide further experimental testing and validation of novel biological insights (Fig. 3.1D).

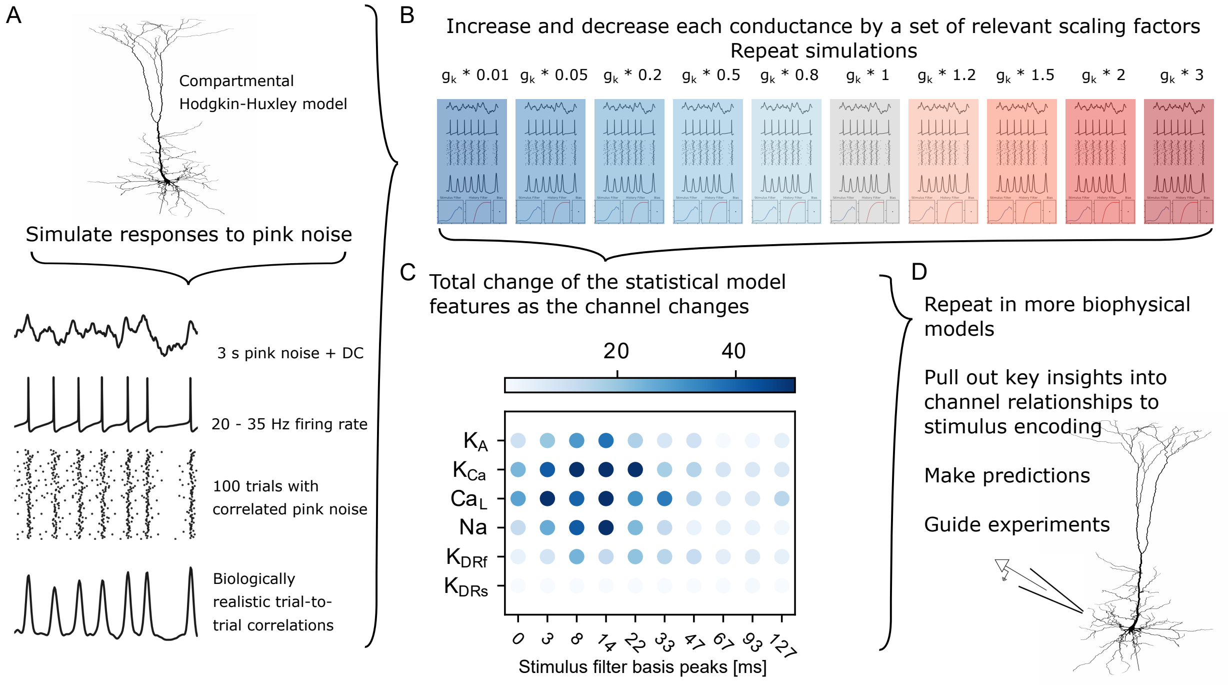


Figure 3.1: The combined biophysical and statistical modeling paradigm. **A** Schematic of biophysical modeling approach to simulate spike trains. See section 3.2.1. **B** Conductance of the kth channel (g_k) in the model multiplied by a scaling factor to globally increase or decrease g_k. Shading indicates g_k increased (red) or decreased (blue) compared to control (gray). See section 3.2.1, Fig. 3.2. **C** A summary of channel conductance influence on stimulus filter. The dot shows the total variance of the stimulus filter features across different channel conductances (defined in Eq. 3.8). Each column indicates a specific stimulus filter feature in a certain time range. Darker color means the feature is more strongly modulated by channel conductance (see section 3.2.3, Fig. 3.4 and 3.5). For example, K_{Ca} channel strongly modulates neural response roughly around 5 to 30 ms post-stimulus. Data are fit with a statistical model PP-GLMs (see section 3.2.2 and Fig. 3.3, 3.4). **D** Examples of next steps for using the pipeline to explore stimulus encoding.

3.2.1 Biophysical model

In order to understand how functional ion channel expression affects stimulus encoding, it is necessary to have confidence in many parameters of ion channel dynamics and distributions. It is experimentally difficult to gather sufficient information about both the cell's functional ion channel expression and the cell's stimulus encoding in a typical whole-cell patch clamp recording due. To overcome these experimental challenges, we instead used detailed biophysical models with necessarily known functional ion channel expression. Then we simulated somatic membrane voltage (V_m) responses to injection of pink noise to evaluate the stimulus encoding properties of a given model. The biophysical modeling portion of the pipeline is shown in Fig. 3.1A-B. We tailored our biophysical model simulations on an idealized version of an actual patch clamp experiment to collect spiking data, later used to fit the statistical model (Fig. 3.1C).

Biophysical model simulations were made in NEURON v7.4 or 7.6 [27] on a personal computer or the Pitt Center for Research Computing cluster. Simulations were performed with fixed time-step integration at 40 kHz. We used two previously published neuron models with code available from ModelDB [66]. These included two distinct cell projection cell types: the rodent olfactory bulb MC [14], and the rodent L5 PC [1]. Each model has detailed 3D morphology based on reconstructions and non-uniformly distributed conductances in the somatic and dendritic compartments which have been constrained to data. Ion channel kinetics were based on Hodgkin-Huxley type models [67]. We assume here that morphology was known, spatial distributions of ion channels were known, and that ion channel kinetics were known. Therefore, we have not varied any of the existing morphological, distribution, or kinetics parameters from their previous implementations.

Our goal is to simulate a whole-cell patch clamp experiment used to ascertain a cell's stimulus encoding properties. Typically this is through the somatic current clamp configuration simultaneously recording somatic V_m and injecting a stimulus with a broad range of frequency components. To exclude any confounding circuit effects, synaptic activity is often blocked pharmacologically, thus our models do not contain any synaptic conductances. All biophysical model simulations were based on the current clamp configuration, with somatic stimulus current injection and somatic V_m recording.

Broadband noise is a rich source of stimuli across a wide range of the frequency spectrum often used to approximate the collection of synaptic events reaching the soma [133]. We used 100 trials of a 3 s stimulus of broadband pink noise riding on a direct current (DC) offset. The stimulus was Gaussian white noise convolved with an alpha function: $\alpha(t) = (t/\tau) * \exp(-t/\tau)$ with $\tau = 3$ ms [48]. In an experiment, the same frozen pink noise stimulus is repeated over many trials leading to some level of trial-to-trial variability in spike time reliability. However, our models are deterministic and have no trial-to-trial variability. To mimic biological trial-to-trial variability we produced sets of correlated pink noise trials that vary from trial-to-trial, as described previously [26]. Each trial's stimulus is a linear combination of a parent noise and a newly generated noise stimulus. The parent noise is shared between all noise trials, but not used as a stimulus itself. For each model, we chose a DC offset, noise standard deviation, and the trial-to-trial noise correlation empirically, by comparing biophysical model outputs to experimental values.

To account for biologically realistic parameter variation, we varied individual ion channel conductances globally by a scaling factor. A set of conductance scaling factors were chosen to represent a biologically realistic parameter variation of about 6-fold [99], while also including nearly complete removal (99%) of a conductance. Although the complete absence of a conductance may not be likely under normal cell-to-cell variation, it may represent a genetic ablation, mutation, or near-fully effective pharmacological block. The scaling factors set included 0.01, 0.05, 0.2, 0.5, 0.8, 1.0, 1.2, 1.5, 2.0, 3.0 (Fig 1b). We simulated V_m in response to the same 100 trials of correlated pink noise for each ion channel and for each scaling factor (Fig. 3.1B). The resulting spiking data were used to fit the PP-GLMs. The spike times were defined as the time when V_m crossed the threshold of 0 mV. Then the spike times were binned into 1 ms intervals. The time bin was small enough that each bin contains at most one spike.

3.2.2 Statistical model

The PP-GLM has been widely applied in electrophysiological recordings to model the patterns of spike trains due to its flexibility, simplicity and versatility [79, 111, 134, 137]. The PP-GLM includes a stimulus filter, a post-spike history filter, a baseline, and a nonlinear link function as shown in Fig. 3.3D. The probability of observing a spike at j'th time bin is $[p_{(i)}]_j$ given the stimulus and the post-spike history up to time bin j (the conditional notation is removed for simplicity). The subscript (i) indicates the quantity for ion channel conductance scaling factor g_i (see section 3.2.1). For one time bin j, the influence of the stimulus is $\sum_{t=0}^{T_k} k(t)s(j-t)$, T_k is the length of the stimulus filter k. s is the vector of the stimulus. The calculation for all time bins is equivalent to convolution, so the notation is simplified to $[k \otimes s]_j$, where $\otimes$ denotes the convolution, $[\cdot]_j$ indicates the data at the j'th time bin. Similarly, the influence of the spikes is $\sum_{t=0}^{T_h} h(t)y(j-t) = [h \otimes y]_j$, T_h is the length of the post-spike history filter h. y is the vector of binary spike trains. $\text{logit}([p_{(i)}]_j)$ is modeled as a linear combination of the variables, which is also known as the logistic regression.

$$\text{logit}([p_{(i)}]_j) = [k_{(i)} \otimes s]_j + \boldsymbol{\beta}_{(i)}^{\text{baseline}} + [h_{(i)} \otimes y_{(i)}]_j \tag{3.1}$$

$$k_{(i)}(t) = \boldsymbol{\beta}_{(i),1}^{K} k_1(t) + ... + \boldsymbol{\beta}_{(i),d_K}^{K} k_{d_K}(t) \tag{3.2}$$

$$h_{(i)}(t) = \boldsymbol{\beta}_{(i),i}^{H} h_1(t) + ... + \boldsymbol{\beta}_{(i),d_H}^{H} h_{d_H}(t) \tag{3.3}$$

$$[x_{(i)}]_j^T := \Big([k_{(i),1} \otimes s]_j, ..., [k_{(i),d_K} \otimes s]_j, 1, [h_{(i),1} \otimes y_{(i)}]_j, ..., [h_{(i),d_H} \otimes y_{(i)}]_j \Big) \tag{3.4}$$

In this PP-GLM, we need to estimate the baseline, the stimulus filter $k_{(i)}(\cdot)$ and the post-spike history filter $h_{(i)}(\cdot)$. Both filters are fitted using with bases K, H. K has d_K bases $\{k_1, ..., k_{d_K}\}$, H has d_H bases $\{h_1, ..., h_{d_H}\}$. $\boldsymbol{\beta}^K$ is the subset for stimulus filter, $\boldsymbol{\beta}^H$ is the subset for the post-spike history filter. $\boldsymbol{\beta}^{\text{baseline}}$ is a scalar representing the baseline. The design of the bases follows [111]. These bases can be seen as manually engineered features of the neuron firing model. As shown in Fig. 3.4B, the bases are bell-shaped curves, each one makes a contribution to the shape of the filter in different lag ranges. The bases are narrower in duration near the spike time (around lag 0 ms), whereas they are wider in duration further from the spike time (larger lag). This corresponds to a neuron's dynamics, which are more complex close to spike initiation and less complex further from the spike initiation. An example of the linear combination of stimulus bases and coefficients to generate a stimulus filter is depicted in Fig. 3.4A-B. The coefficients can be stacked into a vector $\boldsymbol{\beta}(g_i) := \boldsymbol{\beta}_{(i)} \in \mathbb{R}^{d_K+1+d_H}$. The features of PP-GLM in Eq. 3.1 is stacked into $[x_{(i)}]_j$ in Eq. 3.4 as the covariates for regression, so $\text{logit}([p_{(i)}]_j) = [x_{(i)}]_j^T \boldsymbol{\beta}(g_i)$ is in linear form. The log-likelihood of one spike train with T time bins is,

$$\begin{aligned} \ell_{(i)}(\boldsymbol{\beta}(g_i)) =& \sum_{j=1}^{T} \Big([y_{(i)}]_j \log[p_{(i)}]_j + (1 - [y_{(i)}]_j) \log(1 - [p_{(i)}]_j) \Big) \\ =& \sum_{j=1}^{T} \Big([y_{(i)}]_j [x_{(i)}]_j^T \boldsymbol{\beta}(g_i) - \log(1 + \exp\{[x_{(i)}]_j^T \boldsymbol{\beta}(g_i)\}) \Big) \end{aligned} \tag{3.5}$$

The PP-GLM is a powerful model that can capture a rich family of spiking patterns [137]. We applied the PP-GLM to spike trains simulated from each biophysical model where an individual ion channel conductance was scaled differently for each simulation. Thus, for each unique set of ion channel conductances we obtained a set of corresponding PP-GLM coefficients ($\boldsymbol{\beta}(g_i)$) that reflect differences in firing patterns. However, the trend of the changes of coefficients with changing ion channel conductances is typically noisy, making it difficult to determine how changes in coefficients relates to ion channel conductance. The next section discusses a method to overcome this problem by jointly training different $\boldsymbol{\beta}(g_i)$ together.

3.2.3 Linking biophysical and statistical models

To bridge the biophysical model and the statistical model, we create a mapping from the biophysical model parameters to the PP-GLM parameters. We want to study how the PP-GLM features, coefficients $\boldsymbol{\beta}(g)$, change as functions of the ion channel conductance scaling factor g. This mapping can quantify the influence of ion channel conductance on the spike train patterns. The features of the spike trains are not limited to PP-GLM coefficients. We discuss other options in section 3.4.

Spike trains with different ion channel conductances can be fitted separately, but this usually leads to noisy and unstable results. To create a smooth mapping between biophysical model parameters and PP-GLM parameters, we developed the following model in Eq. 3.6. An example can be found in Fig. 3.4, a comparison between a non-smoothed model (Fig. 3.4C,E,G,H) and a smoothed one (Fig. 3.4D,F,K,L). As will be shown later, some changes of the statistical model can be shrunk to zero, meaning the corresponding spike train patter is not modulated by the channel conductance.

In the biophysical simulation, the ion channel conductance is scaled with factors $(g_1, g_2, ..., g_B)$ in increasing order (section 3.2.1), and the fitted PP-GLM parameters will change accordingly. We assume this transition between adjacent PP-GLM parameters is smooth. The PP-GLM models with smooth transitions are fitted jointly in Eq. 3.6. The changes of $\boldsymbol{\beta}(g_i)$ are constrained to obtain the smooth trend. This technique is also known as the trend filtering used to smooth the signal in a non-parametric way [87, 113]. As g_i may not be set using equal step sizes due to the experiment settings, the changes of the PP-GLM with larger steps are expected to be larger than those with smaller steps. The term $1/(g_{i+1} - g_i)$ in the penalty is used to normalize the step size. ℓ_1 norm is used in the penalty term so that small changes can be shrunk to zero. If the penalty hyperparameter $\lambda = 0$, it is equivalent to fitting each dataset independently. If $\lambda = \infty$, it is equivalent to fitting each dataset using the same set of coefficients ($\boldsymbol{\beta}(g_1) = ... = \boldsymbol{\beta}(g_B)$). The optimization uses the alternating direction method of multipliers (ADMM) algorithm, see Appendix for implementation details. The algorithm was coded in Matlab R2018a[1].

$$\min_{\boldsymbol{\beta}(g_1),...,\boldsymbol{\beta}(g_B)} \sum_{i=1}^{B} -\ell_{(i)}(\boldsymbol{\beta}(g_i)) + \lambda \sum_{i=1}^{B-1} \frac{1}{g_{i+1} - g_i} \|\boldsymbol{\beta}(g_i) - \boldsymbol{\beta}(g_{i+1})\|_1 \tag{3.6}$$

For the selection of the penalty hyperparameter (λ), there is a rough trade-off between the smoothness of the change $\boldsymbol{\beta}(g)$ as a function of g and goodness-of-fit. When λ is small, the coefficients $\boldsymbol{\beta}(g)$ have large fluctuations. When λ is large, the coefficients $\boldsymbol{\beta}(g)$ change smoothly,

[1] All code is available on: `https://github.com/albertyuchen`.

but int undermines the goodness-of-fit. The tuning parameter is selected from the set using grid-search $\lambda \in \Lambda = \{\lambda_{\max}, \lambda_{\max}\alpha, \lambda_{\max}\alpha^2, ..., \lambda_{\max}\alpha^{k-1}, 0\}$, where $k = 22$ and $\alpha = e^{-1}$. When $\lambda = \lambda_{\max}$, the estimated vector $\boldsymbol{\beta}(g)$ is a constant of g. (See Appendix for details about calculating the $\lambda_{\max}$.) To get the trend as smooth as possible, while maintaining a good fit, λ is selected using the following rule. It selects λ as large as possible, while maintaining a reasonable performance on the test dataset that is as good as the best one.

$$\lambda^* = \arg\max_{\lambda \in \Lambda} \left\{ \lambda : \sum_{i=1}^{B} \ell_{(i)}^{\text{test}}(\boldsymbol{\beta}(g_i, \lambda)) > -\zeta + \max_{\eta \in \Lambda} \sum_{i=1}^{B} \ell_{(i)}^{\text{test}}(\boldsymbol{\beta}(g_i, \eta)) \right\} \tag{3.7}$$

where $\ell_{(i)}^{\text{test}}$ is the log-likelihood on the test dataset. 70% trials were used for training, and 30% trials were used for testing. $\boldsymbol{\beta}(g_i, \lambda)$ is obtained from Eq. 3.6 with respect to g_i under the penalty hyperparameter λ. The likelihood ratio on the test dataset between the one with the largest likelihood value and the one selected with λ^* is at most ζ. $\zeta > 0$ is set as a very small value ($\zeta = \log 1.0005$) so that the difference is not significant. Thus, λ is constrained in range where the log-likelihood is greater than $-\zeta + \max_{\eta \in \Lambda} \; \ell_{(i)}^{\text{test}}(\boldsymbol{\beta}(g_i, \eta))$ to ensure the selected model has satisfactory performance. Then λ is chosen with the largest value among Λ to get the smoothest trend possible of $\boldsymbol{\beta}(g_i)$. In section 3.4, we will show that this selection strategy can achieve a good channel conductance prediction performance as well. The fitted response filters $k(t, g_i) := k_{(i)}(t)$, $h(t, g_i) := h_{(i)}(t)$ and $b(g_i) := \beta_{(i)}^{\text{baseline}}(t)$ obtained under the λ^* show how the channel conductance factor g influence the shapes of the filters. The shapes of the filters reflect the firing patterns and how the neuron responds to the external stimulus and its post-spike history.

3.2.4 Quantifying how ion channel conductance affects the statistical model

The PP-GLM captures the statistical features of spike train patterns. Scaling ion channel conductances can change spike firing patterns, and these changes will be reflected in PP-GLM parameters. To quantify the relationship between PP-GLM parameters and varying ion channel conductances, we define the sum of slopes (SS) for the coefficients $\boldsymbol{\beta}(g_i)$ as follows. The change of the coefficients with changing ion channel conductance represent the change of the corresponding features of the stimulus filter (Eq. 3.2) and post-spike history filter (Eq. 3.3).

$$SS(\lambda)_{[q]} = \sum_{j=1}^{B-1} \frac{1}{g_{j+1} - g_j} |\boldsymbol{\beta}(g_j)_{[q]} - \boldsymbol{\beta}(g_{j+1})_{[q]}| \tag{3.8}$$

The subscript $[q]$ denotes the entry index of a vector. Under a certain penalty hyperparameter λ, some coefficients $\boldsymbol{\beta}(g)$ may become constants of g. However, other coefficients may have a large variance, indicating that these coefficients are more correlated with the ion channel conductance than those that are constant. Coefficients with a large SS indicate features of the PP-GLM that are strongly affected by an ion channel conductance and thus how an ion channel conductance affects a given feature of stimulus encoding. The unit of SS is the unit of $\boldsymbol{\beta}$ divided by the unit of g_i. In our case, the unit of $\boldsymbol{\beta}$ is logit spikes/sec, the unit for g_i is arbitrary as it is the scale of the conductance. We discuss additional methods of quantifying relationships between ion channel conductance and PP-GLM parameters (see section 3.4).

3.2.5 Model Verification

To verify that the method of PP-GLM fitting with trend filtering technique (Eq. 3.6) could recover the trend of the changes defined in Eq. 3.8, we designed the following set of simulations. We used a sequence of PP-GLM models as the true model with smooth transitions, and compared the estimation with the true model. The model performed well in the simulations. The details are in Appendix.

3.3 Results

Here we will demonstrate the entire combined biophysical and statistical modeling pipeline. As this pipeline screens all channels present in each model, we focus only a subset of ion channels to highlight the benefits of this method. We focus primarily on the MC model $K_A$channel, as it demonstrates the utility of the pipeline and as we have previously examined changes information processing from reducing MC $K_A$channel conductance experimentally [109]. Note that while this method can be applied to any ion channel conductance in principle, training PP-GLMs requires spikes. Therefore, if scaling an ion channel conductance results in few or no spikes, then PP-GLMs cannot be trained. The following sections detail the considerations and analyses applied to evaluating the role of given ion channels in stimulus encoding for each step in the pipeline. See the Methods in section 3.2 for detailed implementation instructions.

3.3.1 Biophysical modeling

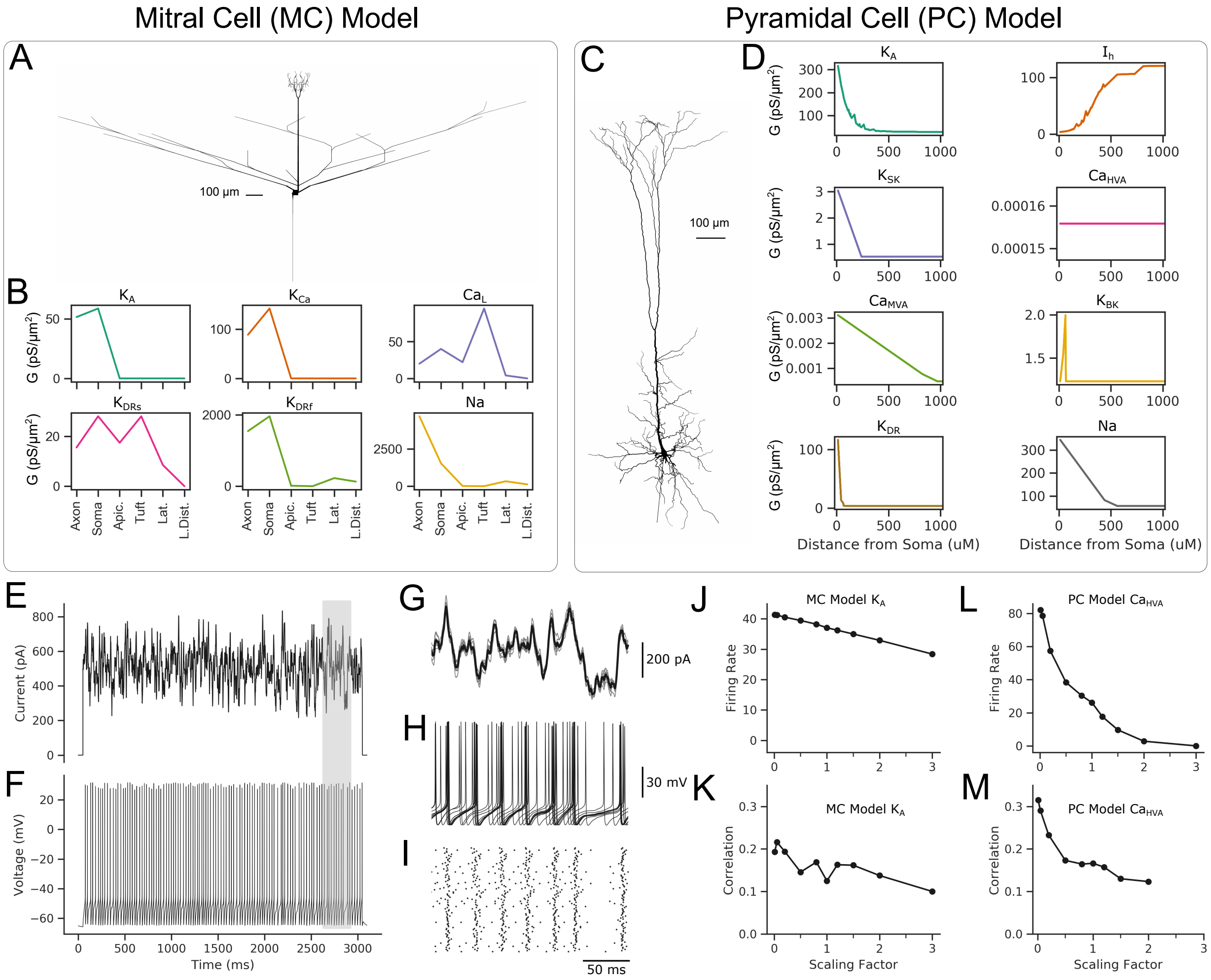


Figure 3.2: Biophysical models. **A** Morphology of the MC. **B** MC channel conductance parameters in subcellular compartments. **C**, Morphology of the PC. **D** PC channel conductance parameters in subcellular compartments as a function of distance from the soma. **E** An example pink noise stimulus injected into the somatic compartment. **F**, The simulated V_m recorded in the somatic compartment resulting from injected pink noise stimulus. **G-I** Detail view of shaded gray region of **E-F** of the mean stimulus (black lines) and 10 individual pink noise stimuli (gray traces) (**G**), with corresponding 10 V_m recordings (**H**) and raster plot of all 100 trials (**I**). **J-M**, Basic statistics of the the simulated trials as a function of ion channel conductance scaling factor for the MC model K_A(**J,K**) and the PC model Ca_{HVA}channel (**L,M**).

We demonstrate the pipeline using two morphologically and functionally distinct projection neuron cell type models, the MC model [14], and the PC model [1]. We chose these biophysical models due to the strict data-driven constraints used to set the morphology and optimize the parameters defining each model's functional ion channel expression. Both biophysical models also contain non-uniform subcellular ion channel distributions including active conductances in dendritic compartments (Fig. 3.2A-D). Although we do not consider dendritic inputs here, these models implicitly capture any effects active dendritic conductances may have on stimulus encoding when driven by somatic spiking. Tuning the parameters of biophysical models is often underconstrained by data and typically many sets of model parameters can fit the data equally well [100, 131]. Both the MC and PC models used here took advantage of varied electrophysiological datasets and a reduced parameter fitting procedure. Subsets of parameters of ion channels are estimated using datasets where ion channels of interest have been isolated. This type of reduced parameter fitting procedure, or parameter peeling procedure, has been shown to greatly reduce the variability of parameter estimates and avoid local minima [85]. The MC model used data collected from multiple cells as an average MC model behavior, whereas the PC model uses data collected from single cells, taking advantage of more robust parameter estimation by using recordings from the somatic and dendritic compartments [84, 85]. Thus, both biophysical models used here have strongly data-driven morphological and functional ion channel expression parameters.

Our goal is to use the biophysical models to simulate an idealized experiment by which we would collect data to fit PP-GLMs, while functional ion channel expression is known. The biophysical models are used to simulate somatic V_m responses to injected pink noise stimulus (Fig. 3.2E-I). The stimulus is broadband and is meant to approximate synaptic input summation at the soma [98] (see section 3.2). Sticking to idealized experimental constrains, we simulate a 3 s stimulus repeated for 100 trials. To generate trial-to-trial variation in spike timing in the deterministic biophysical models, we incorporate correlated noise into the pink noise stimulus (see section 3.2). The stimulus DC offset, standard deviation, and trial-to-trial stimulus correlation are chosen to reflect experimental firing rates and trial-to-trial spike time correlations at control (1.0) scaling factor (Fig. 3.2J-M). We repeat the same biophysical model idealized experimental simulation for every ion channel in a model, while globally scaling the ion channel conductance by a set of scaling factors: 0.01, 0.05, 0.2, 0.5. 0.8, 1.0, 1.2, 1.5, 2.0, 3.0 (see section 3.2; Fig. 3.1B, Fig. 3.3B). Unless otherwise mentioned, through the remainder of the text, black traces correspond to control or scaling of 1.0; blue traces correspond to decreased scaling factors, with the hue darkening with decreasing scaling; and red traces correspond to increased scaling factors, with the hue darkening with increased scaling. We then use this idealized experiment of the simulated spike times in response to the stimulus on each trial as the basis for fitting PP-GLM parameters (Fig. 3.1C; Fig. 3.3D-G).

Focusing on the MC $\mathrm{K_A}$channel and the PC $\mathrm{Ca_{HVA}}$channel shows marked differences in scaling each ion channel conductance on firing rate and trial-to-trial correlations (Fig. 3.2J-M). However, the spike firing dynamics are vastly more complicated than these simple measures can capture. For instance, examining a portion of the stimulus over all trials of all scaling factors for the MC $\mathrm{K_A}$channel, we see profound changes in spike firing patterns within and between trials, even with subtle changes in ion channel conductance scaling factors (Fig. 3.2B-C). When decreasing the MC $\mathrm{K_A}$ion channel conductance from control, spike firing becomes more regular at 0.8 scaling factor, but then loses all trial-to-trial structures at 0.5 scaling, before regaining regular firing when decreasing the ion channel conductance further (Fig. 3.3B). Such changes are also captured as continuous PSTHs (Fig. 3.3C). These types of changes are not well captured by simple measures such as firing rates or trial-to-trial correlations. Therefore, to more accurately and systematically quantify the statistical patterns of the spikes, we introduce the PP-GLM in the following sections (Fig. 3.3D-G). The difference between firing patterns will also be depicted by the PP-GLM, while capturing the stimulus encoding features in a set of PP-GLM parameters. This link between biophysical models with known functional ion channel conductance and statistical models that capture high-dimensional patterns of stimulus encoding is the key advance of this pipeline.

3.3.2 Fitting PP-GLMs

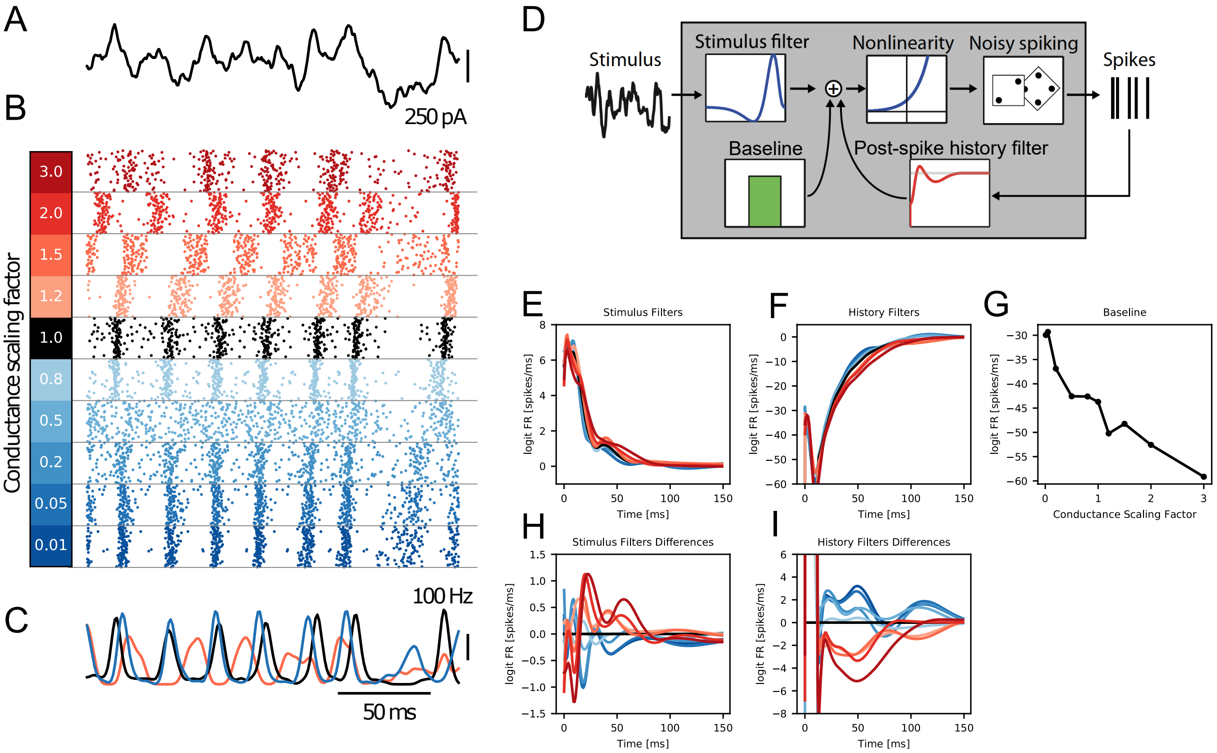


Figure 3.3: PP-GLM – a stimulus encoding model. **A-C** Examples of MC biophysical model simulations of K_A channel in the same section of simulation time in the column: **A**, one stimulus trial. **B** Spike raster plot for all 100 trials for the indicated conductance scaling factor. **C** PSTH for conductance scaling of 1.5, 1, and 0.05. **D** PP-GLM diagram. **E-G** Fitted PP-GLM stimulus filters, post-spike history filters and baselines for different conductances. Colors correspond to the conductance scaling factor legend. **H,I** The differences between stimulus filters and post-spike history filters. The filter with scalar 1 is used as reference shown in dark. The seemingly small difference between filters is critical in the goodness-of-fit as will be shown later.

The stimuli and spike trains from biophysical model simulations described above are used as inputs to fit PP-GLMs (see section 3.2; Fig. 3.3D-G). As discussed above, the spike firing patterns change with scaling the MC $K_A$ion channel conductance (Fig. 3.3A-C). The changes in spike firing patterns are reflected in changes of the PP-GLM parameters for the stimulus filters, the post-spike history filters, and the baseline (Fig. 3.3E-G).

The effect of MC $K_A$channel conductance scaling on the baseline is marked. Increasing the channel conductances significantly inhibits the firing rate which matches well with the conductance dependence of the overall firing rate (Fig. 3.2j; Fig. 3.3G). Fitted stimulus and post-spike history filters are shown in Fig. 3.3E and F. The details of the difference is shown by calculating a simple subtraction of the control scaling factor from all scaling factors (Fig. 3.3H, I). The control scaling factor subtractions reveal how increasing MC $K_A$channel conductance affects different portions of the stimulus filters and post-spike history filters (Fig. 3.3H, I). Some of the changes in PP-GLM filters are seemingly small and noisy. Does $K_A$channel only affect average firing rate (baseline) but not the stimulus response (stimulus filter) or inter-spike dependency (post-spike history filter)? We will show in the next section that some part of the change is due to data noise, even it is large, for example the beginning part of the post-spike history filter. Some part is modulated by channel conductance even the change is relatively small, but it is critical in the goodness-of-fit as will be shown later. Forcing all filters to be the same across different channel conductances leads to very poor fit. Next, we will discover the clear trends in the PP-GLM parameters with changing ion channel conductances.

3.3.3 Fitting PP-GLMs with trend filtering

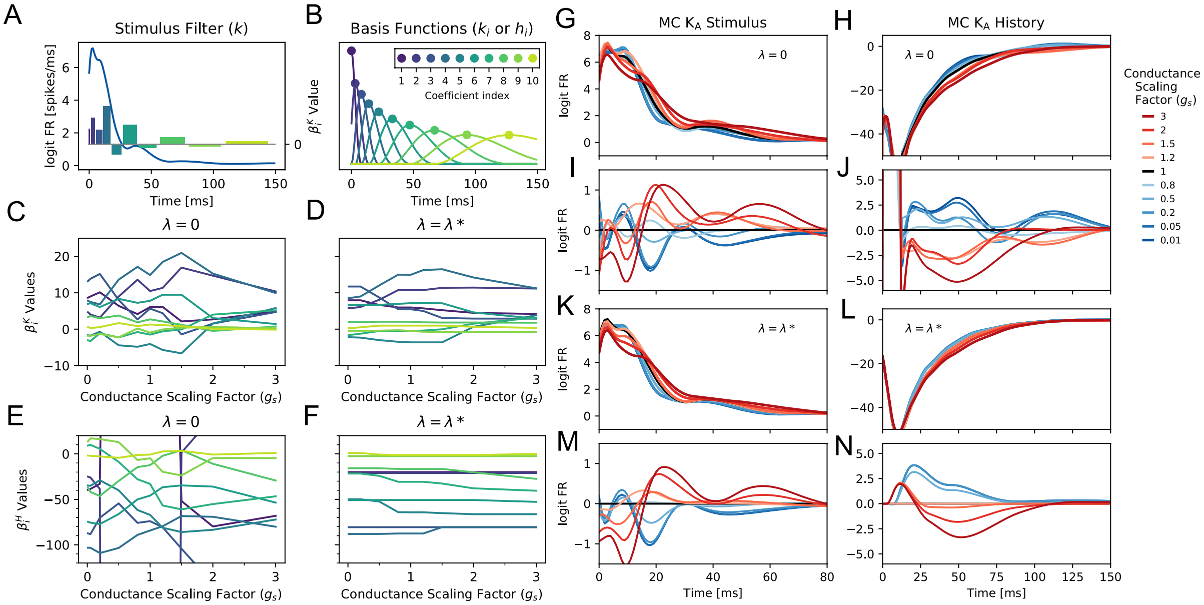


Figure 3.4: Jointly fitted PP-GLM. The example is about the MC model K$_\text{A}$ channel. **A** The stimulus filter (blue trace; k) constructed from Eq. 3.2 (lower), for the K$_\text{A}$ channel with conductance scaling factor (g_i) of 1 and penalty hyperparameter λ of 0. The relative values of coefficients (colored bars; β_i^K) corresponding to the peak times of stimulus bases functions (k_i) as in **B**. **B** The bases functions k_i or h_i in Eq. 3.2 and 3.3 with peaks identified by dots to correspond to the i^th coefficient (colored dots; $\boldsymbol{\beta}_i^K$). The unique set of fitted coefficients combine to generate a stimulus filter as in **A**. **C, D** The values of all stimulus coefficients as a function of g_s with no penalty ($\lambda = 0$; **C**) and the selected penalty hyperparameter $\lambda = \lambda^*$ according to Eq. 3.7. Trace colors correspond to coefficient indices in **B**. The two plots have the same y-axis range. **E, F** the coefficients for post-spike history filters similar to plots **C, D**. The two plots have the same y-axis range. **G, H, K, L** Overlapped stimulus filters and post-spike history filters across channel conductance scaling factors with no penalty and the selected penalty. **I, J, M, N** show the differences between filters by subtracting the filter with scaling factor 1 as the reference.

When the PP-GLMs for an individual ion channel are trained independently across a set of ion channel conductance scaling factors, the changes in the stimulus and post-spike history filter shapes with conductance scaling are often obscured in noise (Fig. 3.3H, I). In this section, we will show how the trend filtering technique smooths such changes. We propose a trend filtering technique (Eq. 3.6 in section 3.2.2), which takes advantage of the smooth changes in ion channel conductance to impose smooth changes on adjacent PP-GLM parameters. The full set of PP-GLMs across conductance scaling factors for an individual ion channel are trained simultaneously. Thus, by jointly training PP-GLMs, we reduce noise and reveal smooth changes in the stimulus and post-spike history filters with changing ion channel conductance.

To illustrate the trend filtering technique, we first expand upon the PP-GLM training procedure. PP-GLMs are trained by optimizing a set of coefficients: 10 coefficients for the stimulus filter, 10 coefficients for the post-spike history filter, and 1 baseline coefficient. The shape of the stimulus filter and post-spike history filter arise from the linear combination of the product of a set of coefficients and a set of corresponding basis functions (see Eq. 3.2, 3.3; Fig. 3.4A, B). The bases have fixed shapes and are spaced specifically in time, which have been designed for fitting spike trains [111](see section 3.2; Fig. 3.4B). An example of how the stimulus filter shape arises from coefficients is shown in Fig. 3.4A, where the vertical bars represent the coefficient values over the time range of its corresponding basis function. Throughout this section, the coefficient indices and corresponding basis functions in time are represented according to the color legend in Fig. 3.4B, and the peak positions are labeled under the figure in Fig. 3.5A,B,D,E,F,G.

The effects of the trend filtering technique are made clear when comparing the changes in the stimulus filter coefficients (Fig. 3.3C, D) and the post-spike history filter coefficients (Fig. 3.4E-F) across the set of ion channel conductance scaling factors. The variation in coefficient values with ion channel conductance scaling is much greater when PP-GLMs are trained independently without any trend filtering penalty ($\lambda = 0$; Fig. 3.4C, E) than when PP-GLMs are trained with the optimal trend filtering penalty hyperparameter ($\lambda = \lambda^*$; Fig. 3.4D, F). Trend filtering penalizes changes in coefficients between adjacent ion channel conductance scaling factors. Therefore, at a moderate penalty, variation in coefficients with ion channel conductance scaling is reduced overall. This reduces variation to near zero for coefficients with small, less meaningful variation, whereas variation in coefficients with substantial, more meaningful variation remain. However, as the penalty hyperparameter increases, trend filtering will eventually impose no variation in any coefficients, which is undesirable for the goodness-of-fit (Fig. 3.5C). Thus, we select an optimal trend filtering penalty hyperparameter λ^* to balance smooth variation in coefficients with ion channel conductance scaling while maintaining goodness-of-fit (Eq. 3.7; Fig. 3.5C). We demonstrate the clarity afforded from the trend filtering technique by comparing the stimulus and post-spike history filters across the set of MC $\mathrm{K_A}$channel conductance scaling factors (Fig. 3.4G-J). Changes in the shapes of the stimulus and post-spike history filters are much more clear, including in the trends from decreasing to increasing MC $\mathrm{K_A}$channel conductances (Fig. 3.4G-J). With trend filtering at the optimal penalty hyperparameter, it is now possible to relate changes in the spike firing patterns (Fig. 3.3B) to the shapes of the stimulus and post-spike history filters (Fig. 3.4I-J). For instance, with increasing MC $\mathrm{K_A}$channel conductance, the post-spike history decreases, corresponding to a longer refractory period. This change is reflected in the widening of spike timing with increasing MC $\mathrm{K_A}$channel conductance (Fig. 3.3B).

3.3.4 Trend filtering reveals important coefficients

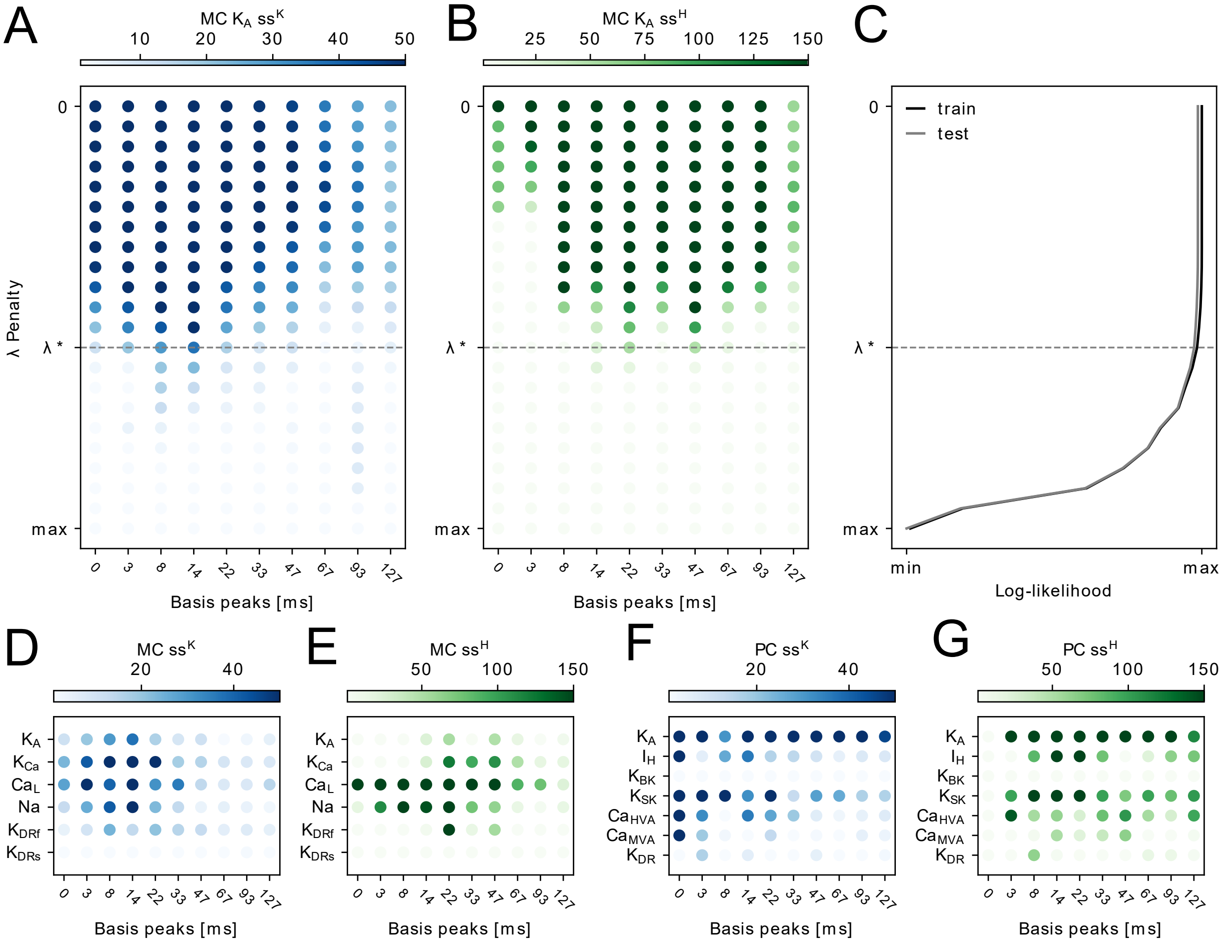


Figure 3.5: The selection of smoothness penalty hyperparameter λ and the results for other channels. **A, B** SS for stimulus filter (**A**) and post-spike history filter (**B**) coefficients with different choices of penalties. SS is defined in Eq. 3.8, describing how large the coefficients change across different channel conductances. The x-axis indicates the peaks of the basis, the order is the same as Fig. 3.4B. The optimal tuning parameter λ^* is indicated by a horizontal line. **C** The log-likelihood for model fits with different penalties. The log-likelihood is divided by the number of trials. **D-G** SS for different channels in the MC model and the PC model with stimulus coefficients in blue and post-spike history coefficients in green. The results all use the optimally selected penalty hyperparameter.

As expected, the qualitative changes in stimulus and post-spike history filters we describe above are reflected in the variation of stimulus and post-spike history coefficients (Fig. 3.4C-F). Trend filtering at the optimal penalty hyperparameter, also reveals the coefficients which are most important for an individual ion channel. For instance, the stimulus coefficients representing the early to mid time range (5-30 ms) basis functions remain after trend filtering at optimal penalty hyperparameter, suggesting that the MC $K_A$channel is particularly important for early to mid time range stimulus encoding (Fig. 3.3D). Similarly, the medium range (20-60 ms) post-spike history coefficients are most important. Here we develop a quantitative measurement of the relative importance of coefficients as revealed by trend filtering.

First, we need a simple quantitative measure to capture the overall variation for each coefficient as a function of ion channel conductance scaling. We assign a single value, the SS, to each coefficient (Eq. 3.8). The SS captures the absolute value of variation in a coefficient with ion channel conductance scaling: a low SS value indicates low coefficient variation as a function of ion channel conductance, whereas a high SS value indicates high coefficient variation as a function of ion channel conductance. SS values do lose information of the direction of changes (positive or negative) and nature of changes (linear or non-monotonic). Coefficient SS values are almost uniformly high when $\lambda = 0$, and SS values decrease non-uniformly to 0 when $\lambda = \lambda_{\max}$ (Fig. 3.4K, L). This corresponds to the changes in coefficient variation from when $\lambda = 0$ to when $\lambda = \lambda^*$ (Fig. 3.4C-F). However, the changes in variation have been compressed to one dimensional vector for easier visualization and analysis (Fig. 3.5A, B).

Our method allows for a low dimensional quantitative representation of how a given ion channel affects specific features of stimulus encoding. We can easily compare how scaling different ion channel conductances affects stimulus encoding (Fig. 3.5D-G). By comparing the effects of different ion channels within the same biophysical model, it is clear how scaling each ion channel conductance affects different features of stimulus encoding. It is an obvious conclusion that different scaling different ion channels affects stimulus encoding in unique ways. The SS measure allows for direct comparisons of specific stimulus encoding parameters. For instance, the MC $\mathrm{K_A}$channel prominently impacts early to medium stimulus coefficients and only weakly impacts post-spike history coefficients (Fig. 3.5D, E). In contrast, the MC $\mathrm{Ca_L}$channel has a greater affect on most post-spike history filter components. This type of difference suggests that the MC $\mathrm{Ca_L}$channel is far more important to encoding post-spike history effects than the MC $\mathrm{K_A}$channel. Similar differences are apparent in the PC model when comparing between the PC $\mathrm{I_H}$and $\mathrm{Ca_{HVA}}$channels (Fig. 3.5F, G). Overall, quantifying coefficient SS after trend filtering provides an accurate and intuitive measure of the roles of different ion channels in stimulus encoding. Furthermore, this low dimensional measure can easily compare how scaling different ion channel conductances affects stimulus encoding.

To verify the method of selecting the optimal trend filtering penalty hyperparameter, we perform a set of simulations based on a known set of PP-GLM parameters and determine whether this method can recover the known values (see section 3.2). Using a set of known PP-GLMs, we simulated 100 spike trains for each of the ion channel conductance scaling factors. Then we used the simulated spike trains to train new PP-GLMs using trend filtering and λ^* selection (Appendix Fig. B.1). We found that our method of trend filtering and λ^* selection found SS values very close to those of the true PP-GLM SS values (Appendix Fig. B.1A, B). We repeated this simulation 100 times to determine the error and variance of our trend filtering and λ^* method. We found that the error and variance between the true PP-GLM parameters and our PP-GLM parameters from simulated spike trains reached a minimum at λ^* (Appendix Fig. B.1D). Importantly, when $\lambda > \lambda^*$ the error and variance increased, supporting our selection of the optimal trend filtering penalty hyperparameter (Appendix Fig. B.1D).

3.4 Discussion

Here we have presented a novel computational and analysis pipeline to combine the strengths of biophysical and statistical models (Fig. 3.1). Our goal is to present this pipeline as a method to be applied to any data-driven detailed biophysical model in order to understand more about how ion channels contribute to stimulus encoding. We have presented examples of our pipeline from two distinct cell types, the MC and PC models, and demonstrated the ability of our analyses to identify how scaling different ion channel conductances affect encoding of different and specific stimulus features (Fig. 3.5D-G; Fig. B.2B, C). We believe the best use of this pipeline will be on larger sets of data-driven models, to extract insights into the biophysical mechanisms underlying stimulus encoding, and then testing these insights experimentally (Fig. 3.1D). In this section, we discuss the considerations and limitations of this combined computational pipeline. We avoid making any strong conclusions or comparisons of any biological insights, as our goal here was to demonstrate the pipeline methodology. The example pipeline models we show here are based on a single biophysical model of each cell type. Due to limitations of biophysical models discussed below, biological insights should be gathered from multiple biophysical models.

Computational considerations. It is feasible to run the entire combined biophysical and statistical modeling pipeline on a standard modern desktop computer. Indeed, although we took advantage of available local compute clusters, many of the tests and preliminary results were generated on desktop computers. Morphologically detailed biophysical models with non-uniform active ion channel conductances throughout the dendritic tree are computationally expensive. However, on modern hardware (Intel Core i7), the full set of biophysical simulations for the MC model could be finished in about nine hours. The PC model contains about three times more compartments and therefore takes around three times as long to complete, but is still feasible to run on a desktop computer. It is possible in principle to use less computationally expensive reduced-complexity biophysical models, but these models will also be less biophysically realistic, which may limit any mechanistic interpretations. Biophysical models are highly parallelizable with simulation time speed nearly linear dependent on the number of cores available. PP-GLMs are less computationally expensive than biophysical models. It takes about an hour to finish the calculation for one ion channel dataset with different penalty hyperparameters in our study. The convergence of the ADMM method slows down when the penalty hyperparameter increases.

The features of the filters are the bases which are the same as the regression bases used for model fitting (See Fig. 3.3B). This is for computation convenience. However, there are many ways to extract features from the stimulus and post-spike history filters. coefficients. As long as the feature extraction function is linear, the calculation is simple. By linear function we mean a mapping from the filter k to a single value (a feature point) as a linear operator. For example, a feature of the stimulus filter can be a point at delay t_0. Given the bases coefficients, the feature point is the following according to Eq. 3.2,

$$k(t_0) = \boldsymbol{\beta}_1^K k_1(t_0) + \boldsymbol{\beta}_2^K k_2(t_0) + ... + \boldsymbol{\beta}_{d_K}^K k_{d_K}(t_0)$$

In this way, we obtain a vector of feature points at additional time delays. We can also calculate the derivative of k at a certain delay t_0, which is,

$$k'(t_0) = \boldsymbol{\beta}_1^K k_1'(t_0) + \boldsymbol{\beta}_2^K k_2'(t_0) + ... + \boldsymbol{\beta}_{d_K}^K k_{d_K}'(t_0)$$

Or

$$\int_{t_0}^{t_1} k dt = \boldsymbol{\beta}_1^K \int_{t_0}^{t_1} k_1 dt + \boldsymbol{\beta}_2^K \int_{t_0}^{t_1} k_2 dt + ... + \boldsymbol{\beta}_{d_K}^K \int_{t_0}^{t_1} k_{d_K} dt$$

Another way to examine the PP-GLM, which is an encoding model, is through decoding. Decoding is the process of estimating a reconstruction of the original stimulus given a spike train and a trained PP-GLM. The MC $\mathrm{K_A}$channel conductance scaling affects encoding of beta frequencies, with only moderate effects on low range, theta frequencies and high range, gamma frequencies. This suggests a role for the MC $\mathrm{K_A}$channel in processing information in beta frequencies.

This pipeline can be applied to other data-driven biophysical models of other cell types of interest. The biggest limitation of our current pipeline is that it only considers the scaling of individual conductances. However, in any cell type of interest, there is substantially more variation and covariation in more than a single ion channel. Understanding how variation and covariation of multiple ion channel conductances combine to affect stimulus encoding is of great interest. Nevertheless, we deemed these types of simulations unfeasible as the computational time from varying just two ion channel conductances together increases exponentially with the number of scaling factors chosen. Furthermore, due to well-known issues of identifiability in tuning biophysical model parameters [100, 131], it is problematic to rely strongly on the parameters of any one model. Instead, we believe a larger-scale, neuroinformatic approach would lessen limitations in our pipeline from restricting analysis to individual ion channels and from model identifiability. Using a large set of data-driven biophysical models (e.g. [56, 101]) alleviates concerns of model identifiability and naturally captures the variation and covariation in ion channel conductances within and between cell types.

Chapter 4

Feature coupling across multiple populations of spiking neurons

This is a collaborative work with Hannah Douglas, Bryan J. Medina, Motolani Olarinre, Joshua H. Siegle, and Robert E. Kass. We have already submitted the draft to PLOS computational biology.

Many studies have described covariation among populations of neurons, and shown how it can be used to characterize neural information processing. Here we report a novel method for assessing covariation among population firing rate curves, based on a point process model for spike trains across multiple interacting neural populations. Our application to spiking data from the Allen Brain Observatory demonstrates the power of this approach to reveal relationships that would otherwise be obscured by noise. We label the procedure "feature coupling" because it identifies trial-to-trial covariation among specific features of the population firing rate curves, in this case the times at which peak firing rates occur. Peaks in population firing rate curves may be considered evoked responses.

The model is hierarchical, in the Bayesian sense, with population firing rate intensity functions depending on time-warped template functions that vary with experimental condition; for each trial, the peak times determine the time-warping of the template function, which is the intensity corresponding to the mean (across trials) of the peak times. Under this conception, trial-to-trial variation changes the speed at which neural activity ramps up and calms down. There is also a neuron-clustering component of the model allowing selection, for each condition, of the neurons that define each interacting population. This approach has the advantage of providing uncertainty assessment for all quantities of interest.

4.1 Introduction

Recent advances in electrophysiological recording technologies have dramatically increased the number of neurons and brain regions that can be recorded in a single experiment [76, 121, 128], offering new opportunities for identifying functional interactions among populations of neurons. Peri-stimulus time histograms (PSTHs) are a simple and useful way to compare the relative timing of neural activity across regions, and are therefore widely used. However, because they aggregate data across trials, PSTHs (or smoothed PSTHs) cannot capture trial-to-trial variation [7, 13, 16, 22, 32, 94, 95, 135]. Studies of trial-to-trial covariability across populations of neurons most often consider spike counts in relatively wide windows, and thus cannot identify relationships at precise timescales [7, 16, 32, 59, 125, 126, 136, 140]. To gain additional understanding of trial-to-trial covariation we developed and studied a method centered on population firing rate curves, which can be estimated with millisecond precision. We report here the resulting multiple-population spiking model, and fitting algorithm, together with analysis of data recorded simultaneously from three areas of the mouse visual system.

The data we have analyzed (available from the Allen Brain Observatory [69]), are from primary visual cortex (V1), the lateral medial visual area (LM), and the anterolateral visual area (AL), three regions that lie at distinct hierarchical processing stages while being tightly interconnected [60, 121]. We focus on the times, relative to the onset of a drifting gratings stimulus, of the two peaks (local maximal firing rates) seen in Fig 4.1A, along with the overall (time-averaged) firing rate.

In Fig 4.1A, the time of the second peak in V1 seems to be only slightly ahead of the time of the second peak in LM, to which V1 is connected anatomically. One might expect there to be a strong correlation, across trials, in the timing of the V1 and LM second peaks. A simple way to examine such covariation would be to smooth the PSTH on each trial, i.e, compute a trial-by-trial version of the curves in Fig 4.1A, and find the time of each peak in those curves. This is not very helpful, however, because, as illustrated in Fig 4.1B, the resulting peak times are very noisy. Fig 4.1B shows no consistent relationship, on a trial-by-trial basis, between the timing of the second peak in V1 and the timing of the second peak in LM, which, if true, would be very surprising. The method we have developed greatly reduces the effects of noise in the population spike trains, leaving behind the much more intuitive trial-to-trial correlation seen in Fig 4.1C.

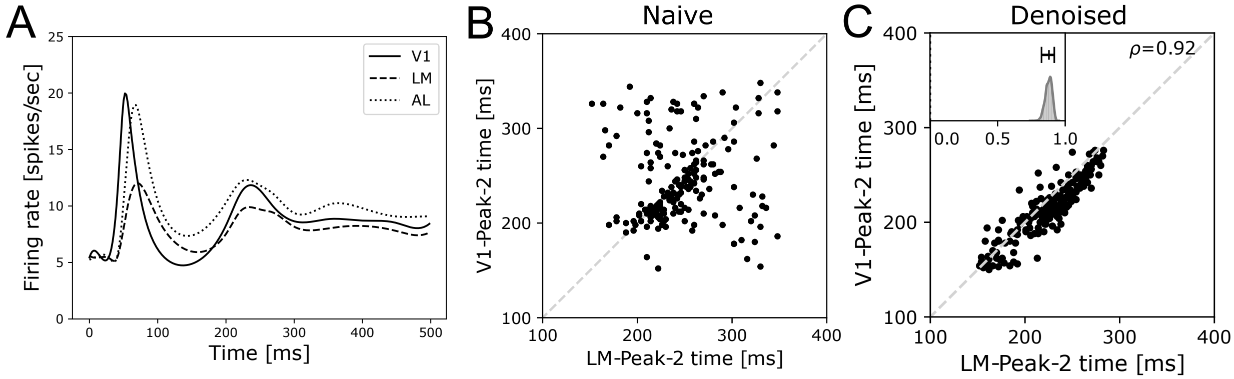


Figure 4.1: **Correlated neural activity features. A** Smoothed population PSTHs for V1, LM, and AL recorded by Neuropixels from a mouse in response to drifting gratings. The three population PSTHs have similar features: the first peak appears at around 60 ms after the onset of the stimulus (time zero), and the second peak appears at around 250 ms after onset. **B,C** Comparison between a naïve method (panel B) and results from our Interacting Population Firng Rate model (panel C) on recovery of Peak-2 position. Each dot, representing one of 195 trials (15 trials in each of 13 experimental conditions), displays the estimated times of the second peak in regions V1 and LM. Both estimates are based on the same subset of active neurons. The naïve method finds the peak of a smoothed population PSTH for each area. No relationship between the peak times in V1 and LM is visible. In panel C, strong covariation appears, and the estimated correlation is .92. The embedded plot in panel C displays estimation uncertainty as a posterior distribution for the correlation. Details about this figure are in 4.4.1.

The method's denoising of correlation and time lag estimation is achieved by (i) introducing a statistical model (a point process model) for the population spike trains, (ii) allowing the population firing rate curves to vary across experimental conditions, and (iii) focusing on condition-specific subsets of neurons that participate in the population. These elements are all incorporated into what we call an Interacting Population Rate Function (IPRF) model. The method has roots in [12, 135], and builds on the large body of work on point process modeling of spike trains [25, 79, 81, 92, 111, 134].

One benefit of denoising is to reveal relationships, like that displayed in Fig 4.1C, which would otherwise be masked. Another is to improve precision of estimates. For example, when we use the naïve method to estimate the lag of LM Peak-2 time behind V1 Peak-2 time, we get a wide 95% CI of $(-5.3, 10.9)$ ms, which is not statistically differentiated from zero; using the method we developed we find this lag to have 95% CI of $(6.2, 13.0)$ ms. In the 4.2.3 section (see Fig 4.4) we give additional estimates of the time lags among activity peaks in the three areas and we provide (see Figs 4.5,C.7 and C.8) disaggregated population firing rate curves that differ from the PSTHs in Fig 4.1A in interesting ways: PSTH representations of firing rate are distorted by blurring across time and averaging across conditions (see also C.9).

In addition to denoising, the method introduced here has two other purposes. First, it is multivariate, which enables pairwise covariation assessment *conditionally* on features of activity in one or more areas. As we show in the 4.2.3 section, this can provide evidence that restricts conceptions of circuit operation. The second further purpose is to describe functional diversity and specialization of neurons that is relevant to cross-area coordinated activity [55, 102, 105]. We estimate the proportion of recorded neurons that participate in this kind of population activity, and we indicate their cortical depths.

4.2 Results

We begin with an overview of the Interacting Population Rate Function (IPRF) model in graphical form followed by a more technical definition. We then describe the algorithms used to fit it, including initialization and uncertainty assessment, our data analysis, and a small simulation study designed to check the likely accuracy of data analytic conclusions.

4.2.1 Model Overview and Specification

The IPRF model describes, for each of several brain areas, a time-varying population firing rate function having key features (here, two dominant peaks), based on an appropriate sub-population of neurons. The model includes both a neuron-selection component and a feature covariation component.

The structure of the IPRF model is illustrated in Fig 4.2. Notice, first, that neurons, indexed by n, come from areas, indexed by a, and repeated trials, indexed by r, are within conditions, indexed by c. Conceptually (beginning at the bottom of the diagram), each neural spike train is a point process determined by a firing rate (intensity) function. The intensity function for a neuron in area a under condition c depends on the whether that neuron, in that condition, participates in cross-area population activity: with probability p_0 it has a baseline firing rate intensity function $f_{a,c}^{pop}$, which we call a template. This is signified by the membership vector z (see Eq 4.2). The template is specific to area a and condition c and is shared in common with all neurons that participate in cross-area interaction (for area a and condition c); these neurons make up the interacting population for area a and condition c. The template $f_{a,c}^{pop}$ is subjected to trial-specific morphing via time-warping, which allows shifts in the times of the two peaks we focus on. These two peak times are two of the features which, together with a gain constant, form the feature vector labeled q. The diagram shows the feature vector being combined with the firing rate template when it becomes an input to the intensity. The features capture trial-to-trial variation, while the template is the intensity corresponding to the mean (across trials) of the features. With probability $1 - p_0$, the neuron is not in the interacting population and, instead, has one of two alternative firing rate function templates, either $f_{a,c}^{local-1}$, which is time-varying or $f_{a,c}^{local-2}$, which constant in time, according to probabilities p_1 and p_2 (where $p_0 + p_1 + p_2 = 1$). In Fig 4.2, these three scalars are collected into a vector labeled *membership probability*. The population and local firing rate template functions are fitted with penalized splines. Finally, the covariation of the 9-dimensional feature vector (3 parameters for each of 3 areas) is summarized in a covariance matrix.

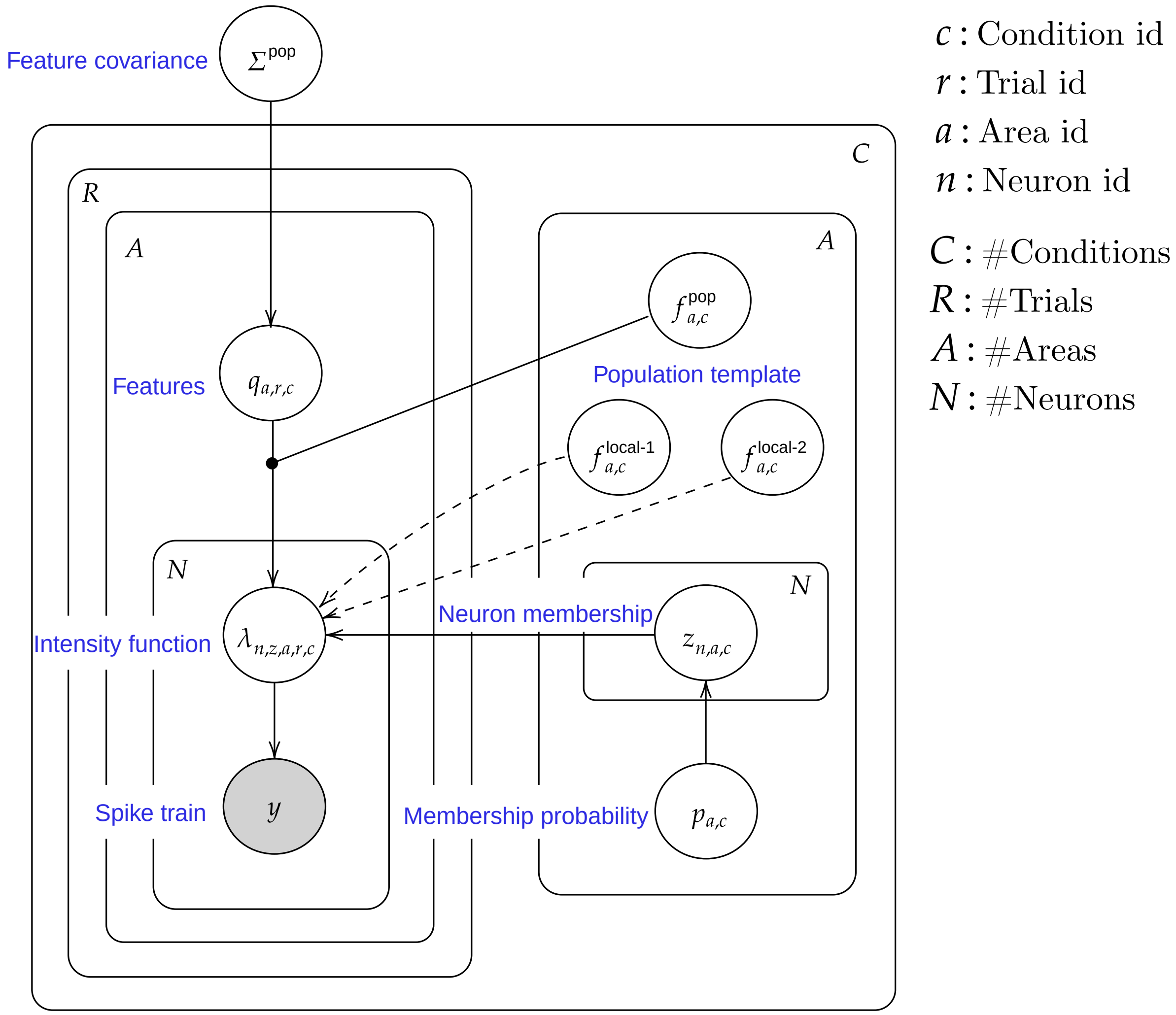


Figure 4.2: **The IPRF model.** As explained in the text, the data y are spike trains modeled as point processes with intensity λ. The covarying features q are combined with the population template f^{pop}, which appears probabilistically in the intensity. The indicators z and probabilities p control whether f^{pop} is part of the intensity or whether, instead, one of the local templates $f^{\text{local-1}}$ and $f^{\text{local-2}}$ is selected.

The more precise specification of the model begins with the population spike trains and corresponding intensity function,

$$y_{n,z,a,r,c}(t) \mid \lambda_{n,z,a,r,c}(t) \sim \text{Poisson}\big(\lambda_{n,z,a,r,c}(t)\big) \tag{4.1}$$

$$\log \lambda_{n,z,a,r,c}(t) \mid q_{a,r,c},, f^{\text{pop}}_{a,c}, f^{\text{local-1}}_{a,c} f^{\text{local-2}}_{a,c}, z_{n,a,c}$$
$$= \begin{cases} f^{\text{pop}}_{a,c}(\varphi^{-1}_{a,r,c}(t) | q_{a,r,c})), & \text{if } z_{n,a,c} = \text{pop} \\ f^{\text{local-1}}_{a,c}(t), & \text{if } z_{n,a,c} = \text{local-1} \\ f^{\text{local-2}}_{a,c}(t), & \text{if } z_{n,a,c} = \text{local-2} \end{cases} \tag{4.2}$$

where φ is the population template time-warping function, specified below in terms of the feature vector $q_{a,r,c}$, and the templates are defined in terms of spline bases B and coefficient vectors β,

$$\begin{aligned} f_{a,c}^{\text{pop}} \mid \beta_{a,c}^{\text{pop}} &= B\beta_{a,c}^{\text{pop}} && (4.3)\\ f_{a,c}^{\text{local-1}} \mid \beta_{a,c}^{\text{local-1}} &= B\beta_{a,c}^{\text{local-1}} && (4.4)\\ f_{a,c}^{\text{local-2}} \mid \beta_{a,c}^{\text{local-2}} &= \mathbf{1}\beta_{a,c}^{\text{local-2}}. && (4.5) \end{aligned}$$

The template group membership indicator z is categorical,

$$z_{n,a,c} \mid p_{a,c} \sim \text{categorical}(p_{a,c}) \tag{4.6}$$

where the three components of each vector $p_{a,c}$ sum to 1. (Categorical(p) is the same as multinomial (n, p) with $n = 1$.)

To complete the model, we define the feature vector, its probability distribution, and the time-warping function. In general, there could be d features for each area. In our data analysis we take $d = 3$ with $q_{a,r,c} = (q_{a,r,c}^{\text{gain}}, q_{a,r,c}^{\text{peak-1}}, q_{a,r,c}^{\text{peak-2}})$, the features being the two peak times (more specifically, the trial-specific deviations of the peak times from those of the trial-invariant template $f_{a,c}^{pop}$) and a gain constant; the gain constant allows the integrated firing rate (integrated across the whole time interval, which controls the expected number of spikes) to vary across areas, conditions, and trials. For a given feature vector, the time-warping function $\varphi_{a,r,c} : [0, T] \mapsto [0, T]$ modifies a template f from a function of time $f(t)$ to the same function of warped time $f(\varphi^{-1}(t))$. Also, in $f_{a,c}^{\text{pop}}$ the constant $q_{a,r,c}^{\text{gain}}$ is added to the time-warped template. We assume the warping function is piecewise linear with the following join-points (also called knots, or landmarks): $(0, 0)$, $(t_{1,L}, t_{1,L})$, $(t_{\text{peak-1}}, t_{\text{peak-1}} + q_{a,r,c}^{\text{peak-1}})$, $(t_{1,R}, t_{1,R})$, $(t_{2,L}, t_{2,L})$, $(t_{\text{peak-1}}, t_{\text{peak-1}} + q_{a,r,c}^{\text{peak-1}})$, $(t_{2,R}, t_{2,R})$, (T, T). Here, the domain of peak-1 is $[t_{1,L}, t_{1,R}]$, meaning that $t_{1,L}$ and $t_{1,R}$ are chosen so that the interval spans (roughly) the times at which the firing rate profile defines the peak. The time-warping function maps the time of peak-1 from $t_{\text{peak-1}}$ to $t_{\text{peak-1}} + q_{a,r,c}^{\text{peak-1}}$. Peak-2 is treated similarly. Piecewise linearity forces linear interpolation of time from the beginning (or end) of the peak range to the peak. The times $t_{1,L}, t_{\text{peak-1}}, t_{1,R}, t_{2,L}, t_{\text{peak-2}}, t_{2,R}$ are fitted at the initialization step, described in the section on Model Fitting and Uncertainty Assessment. Finally, taking the features across areas together, the feature vector has length equal to the product dA (here there will be $dA = 9$ features), and is assumed to follow a normal distribution across trials (and conditions),

$$q_{a,r,c} \mid \Sigma^{\text{pop}} \sim \ N(\mathbf{0}, \Sigma^{\text{pop}}). \tag{4.7}$$

Aside from the template functions $f_{a,c}^{\text{pop}}$ and $f_{a,c}^{\text{local-1}}$ (each having one smoothing parameter determined by cross-validation, as described below), the free parameters are in the matrix Σ^{pop} (which has $dA(dA + 1)/2 = 45$ parameters in our application) and the membership probability vectors $p_{a,c}$ (which has $N \times C = 117$). In addition, the EM and Gibbs sampling algorithms estimate the latent variables $q_{a,r,c}$ and $z_{n,a,c}$.

4.2.2 Model Fitting and Uncertainty Assessment

We have used maximum likelihood for initial fitting, and then Bayesian inference via posterior distributions to assess uncertainty. The hierarchical structure of the model lends itself to application of the EM algorithm and Gibbs sampling. These share a common algorithmic structure, shown in Algorithm 1, with EM and Gibbs sampling each providing their own definitions of the "update." Later in this section we give details of our Gibbs sampling update implementation. We omit EM updating because it is standard, and easily implemented once the structure of the model and resulting likelihood function are understood. In practice, we first complete an initialization step (see below), then we run EM to get maximum likelihood estimates and use those results as initialization for running Gibbs sampling to get posterior distributions based on specified priors (see below).

Algorithm 1: Fitting via Block Structure

```
for $i \leftarrow 1$ to *Max number of iterations* do
    for $c \leftarrow 1$ to $C$ do
        Update $f_{a,c}^{\text{local-1}}, f_{a,c}^{\text{local-2}}$ given the rest;
        Update $f_{a,c}^{\text{pop}}$ given the rest;
        for $r \leftarrow 1$ to $R$ do
            Update $q_{a,r,c}$ given the rest;
        end
        Update $z_{n,a,c}$ given the rest;
        Update $p_{a,c}$ given the rest;
    end
    Update $\Sigma^{\text{pop}}$ given the rest;
end
```

Template functions The template functions $f_{a,c}^{\text{pop}}$ and $f_{a,c}^{\text{local-1}}$ are splines with equally spaced knots (in our data analysis we used 100 knots across the 500 millisecond interval). They are fitted, in the update steps, using penalized likelihood, with second-derivative penalties as in smoothing splines [112]. The smoothing parameter is found using 5-fold cross-validation. In the case of Poisson processes fitting an intensity is equivalent to probability density estimation (see [79, sec. 19.2.2]) and, in density estimation, smoothing splines adapt to varying smoothness [122]. Thus, in this context, we expect penalized splines to fit well (see also related results in [112] and C.2).

The purpose of $f_{a,c}^{\text{local-1}}$ is to improve on the constant firing rate provided by $f_{a,c}^{\text{local-2}}$ in fitting neurons that are irrelevant to cross-population activity. Nothing in the formal specification of the model prohibits similarity of $f_{a,c}^{\text{pop}}$ and $f_{a,c}^{\text{local-1}}$, which creates a model identification problem. In practice, however, confounding of $f_{a,c}^{\text{pop}}$ by $f_{a,c}^{\text{local-1}}$ is precluded by a simple initialization procedure.

Initialization Within each area, each neuron is sorted as having "high," "medium," or "low" firing rate based on spike count during the 500 milliseconds following stimulus onset, after merging across conditions and trials. A PSTH is formed from the high firing rate neurons (for our analysis, the top 25% in firing rates), and from that we obtain the time-warping landmarks $t_{1,L}, t_{\text{peak-1}}, t_{1,R}, t_{2,L}, t_{\text{peak-2}}, t_{2,R}$. The templates $f_{a,c}^{\text{pop}}$ (that is, the basis coefficients) are initialized using penalized splines, with penalty constant chosen through simulation experiments. The neuron membership identifiers $z_{n,a,c}$, and associated probabilities $p_{a,c}$, are initialized using the same sorted spike counts. The feature vectors $q_{a,r,c}$, whose elements represent deviations from the template $f^{\text{pop}}_{a,c}$, are initialized to $(0,0,0)$.

Prior distributions In our implementation we have used the prior distributions

$$\Sigma^{\text{pop}} \mid \Psi_0, \nu_0 \sim \text{IW}(\Psi_0, \nu_0) \tag{4.8}$$

$$p_{a,c} \mid \alpha \sim \text{Dirichlet}(\alpha) \tag{4.9}$$

with hyperparameters Ψ_0, ν_0, α. To make the prior on Σ diffuse, we set $\nu_0 = dA + 1$. We used $\Psi_0 = 2\Phi_0$ where Φ_0 is a diagonal matrix with the square roots of the diagonal elements being equal to the approximate range of q based on data from a different animal. We set $\alpha = 5 \cdot \mathbf{1}$. Additional information may be found in C.3.

Gibbs sampling details Here we provide details of the Gibbs sampling updates in Algorithm 1. Our code is available online at `www.github.com/AlbertYuChen/IPRF`.

1. Updating $f^{\text{pop}}_{a,c}$, $\beta^{\text{pop}}_{a,c}$. The full conditional log posterior is

$$\begin{aligned} &\log p(f^{\text{pop}}_{a,c}|...) = \log p(B\beta^{\text{pop}}_{a,c}|...) \\ =&\sum_{n,r} \ell(y_{n,g,a,r,c}(\varphi_{a,r,c}(t)), \exp\{f^{\text{pop}}_{a,c} + q^{\text{gain}}_{a,r,c}\}) + \eta\mathcal{P}(f^{\text{pop}}_{a,c}) + \text{const} \end{aligned} \tag{4.10}$$

$$\mathcal{P}(f) = -N_{g,a,c} \cdot R_c \cdot \beta^{\text{pop},T}\Omega\beta^{\text{pop}}$$
$$\Omega_{ij} = \int f^{(2)}_i(x) f^{(2)}_j(x)dx$$

where $f^{(2)}_i$ is the second derivative of the ith cubic spline basis element, $N_{g,a,c} \cdot R_c$ is the total number of spike trains, and $\ell(y,\lambda)$ denotes the log-likelihood for a discretized point process with spike train y and intensity λ. The function $f^{\text{pop}}_{a,c}$ is fitted using third-order smoothing spline basis B in Eq (4.3). We used 100 knots are evenly placed in the 500 ms window (so there are 102 basis elements). The sequence of spike counts in 2 millisecond time bins is y. The penalty $\eta\mathcal{P}$ on the coefficients is designed for smoothness in the same spirit as the smoothing spline [63, sec. 5.4]. In Gibbs sampling, the penalty becomes the log prior density, where the prior is normal. The smoothing parameter η is tuned using cross-validation. Details are discussed in C.2. The Gain $q^{\text{gain}}_{a,r,c}$ is set as the offset.

Metropolis-Hastings sampling is embedded in this step of Gibbs sampling. Letting β' be a candidate sample and β^m a sample from the last iteration m, the proposal distribution is

$$\tilde{q}(\beta'|\beta^m) = N(\beta^m, 0.05Q)$$

where Q is the inverse Hessian matrix for β of the posterior at the mode. We estimate Q in the initialization step and hold it fixed subsequently. We follow the suggestion in [49, ch. 12] of setting the scale coefficient of the proposal distribution covariance matrix to be $5.76/d \approx 0.05$, where d is the dimension of the covariance matrix. This balances the trade-off between exploration and rejection. The acceptance ratio on average is kept between 0.2 and 0.8. The acceptance ratio of Metropolis-Hastings sampling method is

$$a = \min\left(1, \frac{p(\beta')\tilde{q}(\beta^m|\beta')}{p(\beta^m)\tilde{q}(\beta'|\beta^m)}\right)$$

2. Updating $f_{a,c}^{\text{local-1}}$, $f_{a,c}^{\text{local-2}}$, $\beta_{a,c}^{\text{local-1}}$ and $\beta_{a,c}^{\text{local-2}}$. The full conditional log posterior is similar to that of $f_{a,c}^{\text{pop}}$. We have

$$\begin{aligned}&\log p(f_{a,c}^{\text{local-1}}|...) = \log p(B\beta_{a,c}^{\text{local-1}}|...)\\ =&\sum_{n,r} \ell(y_{n,z,a,r,c},\ \exp\{f_{a,c}^{\text{local-1}}\}) + \eta\mathcal{P}(f_{a,c}^{\text{local-1}}) + \text{const}\end{aligned} \tag{4.11}$$

   The calculation for $f^{local-2}$ is similar, but simplified because it is a constant times the **1** vector.

3. Updating $q_{a,r,c} = (q_{a,r,c}^{\text{gain}}, q_{a,r,c}^{\text{peak-1}}, q_{a,r,c}^{\text{peak-2}})$. We have

$$\begin{aligned}&\log p(q_{a,r,c}|...)\\ =&\sum_{n,g=g^{\text{pop}}} \ell\Big(\varphi_{a,r,c}(y_{n,z,a,r,c}|q_{a,r,c}^{\text{peak-1}}, q_{a,r,c}^{\text{peak-2}}),\ \exp\{f_{a,c}^{\text{pop}} + q_{a,r,c}^{\text{gain}}\}\Big)\\ &+ \log N(q_{a,r,c}; \mathbf{0}, \Sigma^{\text{pop}}) + \text{const}\end{aligned}$$

   The time-warping function $\varphi_{a,r,c}$ is parameterized by $q_{a,r,c}^{\text{peak-1}}$, $q_{a,r,c}^{\text{peak-2}}$, see 4.2.1. $f_{a,c}^{\text{pop}}$ is given and is treated as the offset. The landmark positions $t_{\text{peak-1}}$ and $t_{\text{peak-2}}$ are determined in the initialization using grid search and are not updated in subsequent fitting.

   Metropolis-Hastings sampling is nested in this step of Gibbs sampling. Letting q' be a candidate sample and q^m the sample from the last iteration at step m, for trial r, condition c and all areas at once (denoted by subscript A) a sample is drawn from the proposal distribution

$$\tilde{q}(q'_{A,r,c}|q^m_{A,r,c}) = N(q^m_{A,r,c} - 0.1 \cdot \overline{q^m_{A,c}}, 0.05Q) \cdot \mathbb{I}_{\text{clip\_region}}(q'_{A,r,c}),$$

   where $\overline{q^m_{A,c}}$ is the mean of $q^m_{A,r,c}$ over all trials and $\mathbb{I}_{\text{clip\_region}}$ truncates so that the first two components of the proposal stay within the domains of peak-1 and peak-2 (see time-warping before Eq (4.7)).

   Subtracting a multiple of $\overline{q^m_{A,c}}$ from $q^m_{A,r,c}$ moves the proposal mean closer to satisfying $\overline{q^m_{A,c}} = 0$, which may be recognized as an identifiability constraint. In principle, the prior on $q_{a,r,c}$ forces identifiability but, because we used a diffuse prior on Σ, samples can drift with increasingly large variances. The proposal we used has lower acceptance rates at this step, but solves the drift problem and produces well-behaved sample trace plots.

4. Updating $z_{n,a,c}$.

$$z_{n,a,c} \sim \text{Categorical}\Big(p(z_{n,a,c} = g|...)\Big),\ g \in \{\text{pop}, \text{local-1}, \text{local-2}\}$$

$z_{n,a,c}$ is drew from categorical distribution. $p(z_{n,a,c} = g|...)$ is the probability of $z_{n,a,c}$ belonging to subpopulation categories $g \in \{\text{pop}, \text{local-1}, \text{local-2}\}$, which is derived as follows,

$$\begin{aligned}
&\log p(z_{n,a,c}|...) = \log p(y_{n,z,a,c}|z_{n,a,c}, q_{a,r,c}, f^{\text{pop}}_{a,c}, f^{\text{local-pop}}_{g,a,c}) \\
&\quad + \log p(z_{n,a,c}|p_{a,c}) + \text{const} \\
=&\log p(z_{n,a,c}|p_{a,c}) + \text{const}+ \\
&\begin{cases} \ell(y_{n,z,a,r,c}(\varphi_{a,r,c}(t)),\ \exp\{f^{\text{pop}}_{a,c} + q^{\text{gain}}_{a,r,c}\}), & \text{if } z_{n,a,c} = \text{pop} \\ \ell(y_{n,z,a,r,c}, \exp\{f^{\text{local-1}}_{a,c}\}), & \text{if } z_{n,a,c} = \text{local-1} \\ \ell(y_{n,z,a,r,c}, \exp\{f^{\text{local-2}}_{a,c}\}), & \text{if } z_{n,a,c} = \text{local-2}. \end{cases}
\end{aligned}$$

5. Updating $p_{a,c}$.

$$p_{a,c}|... \sim \text{Dirichlet}(N_{\text{pop},a,c} + \alpha,\ N_{\text{local-1},a,c} + \alpha,\ N_{\text{local-2},a,c} + \alpha)$$

After conditioning on $z_{n,a,c}$, $p_{a,c}$ becomes independent of the rest variables or data. $N_{\text{pop},a,c}$, $N_{\text{local-1},a,c}$, $N_{\text{local-2},a,c}$ count the total number of neuron memberships $z_{n,a,c}$ in each subgroup in area a in condition c.

6. Updating Σ^{pop}. We draw samples from the Inverse-Wishart distribution

$$\begin{aligned}
p(\Sigma^{\text{pop}}|...) =& \text{IW}(\tilde{\Psi}, \tilde{\nu}) \\
\tilde{\nu} =& \nu_0 + RC \\
\tilde{\Psi} =& \Psi_0 + \mathbf{q}^T\mathbf{q}
\end{aligned}$$

where Ψ_0, ν_0 are hyperparameters. The bold $\mathbf{q} \in \mathbb{R}^{RC\times 3}$ is a stacked matrix of features $q_{a,r,c}$. Each column represents a feature, and each row represents features for a trial. RC is the total number of q. R is the number of trials for each condition, and C is the number of conditions. (We discuss the selection of Ψ_0, ν_0 in C.3.)

Correlation, partial correlation, and regression The estimate of Σ^{pop}, together with samples from its posterior distribution, immediately provide correlations, partial correlations, and regression estimates (also see C.4). More specifically, first, if $V(X) = \Sigma$ for an m-dimensional random vector X, all $\binom{m}{2}$ correlations are obtainable from Σ; second, this means that an estimate of Σ produces estimates of the correlations and samples from the distribution of Σ produce samples from the distributions of those correlations; third, if we partition as $X = (X^{(1)}, X^{(2)}, X^{(3)})^T$ with $X^{(1)}$ being univariate, standard formulas (again using the elements of Σ) also provide estimates of, together with samples from the distribution of, (a) the regression of $X^{(1)}$ on $X^{(2)}$ and (b) when $X^{(2)}$ is also univariate, the partial correlation of $X^{(1)}$ and $X^{(2)}$ conditionally on $X^{(3)}$. Thus, the method provides (immediately) estimates and uncertainties for any regression or partial correlation we wish to examine.

4.2.3 Data Analysis

Implementation of the IPFR model discovers many potentially interesting relationships involving trial-to-trial feature coupling across brain areas, along with relative timing of features, population firing rate profiles, and descriptions of neuron diversity. Here we have chosen to discuss only a few illustrative results. Some others appear in the supplementary material.

Peak-2 correlation and timing We begin with two interestingly different findings that compare correlation with partial correlation. Additional partial correlations are in C.4. Panels A and B of Fig 4.3 displays scatterplots of Peak-2 timing across all trials (as posterior medians): in 4.3A the plot is AL vs LM and in 4.3B it is AL vs V1. Both exhibit high correlation. Consider what we might expect to happen when, in each case, we condition on Peak-2 timing in the third area. If, during the rise phase of Peak-2, the populations of AL and LM neurons were both getting inputs predominantly from V1, then we would expect their partial correlation given V1 to diminish greatly. Fig 4.3C is a scatterplot of residuals after regressing each of AL and LM Peak-2 timing on V1 Peak-2 timing; the correlation of these residuals is the partial correlation of AL and LM Peak-2 timing given V1 Peak-2 timing. The partial correlation (Fig 4.3C) appears to be somewhat smaller than the correlation (Fig 4.3A), but not much smaller. On the other hand, when we examine the Peak-2 timing relationship of AL and V1, conditioning on Peak-2 timing in LM has a dramatic effect: comparing Fig 4.3B with Fig 4.3D, the estimated correlation of .87 (which is highly likely to be above 0.75) drops to an estimated partial correlation of .05 (which is highly likely to be below 0.36).

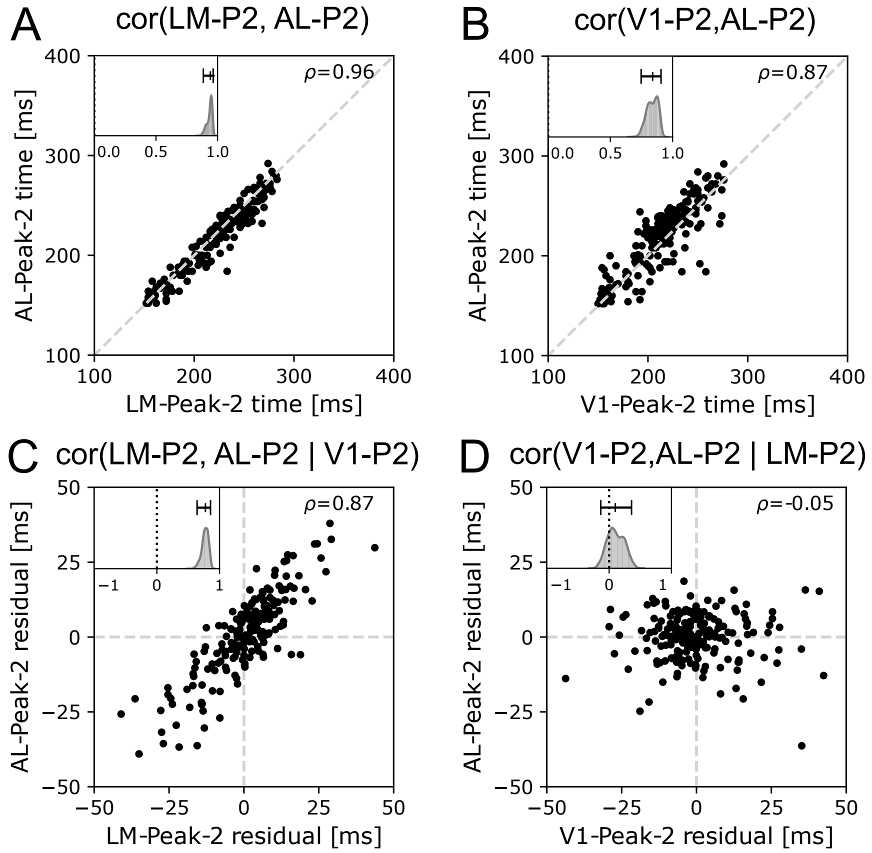


Figure 4.3: **Peak-2 timing: correlation and partial correlation across regions.** Panels A,B display correlations, with each dot representing the estimated time of peak-2 on a given trial. Panels C,D display estimated partial correlations, with each dot representing the residual from a regression on the conditioning variable (the correlation of these residuals being the partial correlation given the conditioning variable). Panels C,D contain partial correlation results, with the areas in C corresponding to those in A and the areas in D corresponding to those in B. All panels use posterior medians as estimates. **A** Despite large variability in peak-2 timing the correlation of LM peak2 time and AL peak-2 time is close to 1. The plot embedded in the upper left corner displays the posterior distribution of the correlation. **B** This plot is for areas V1 and AL, analogous to that in panel A. **C** The residual Peak-2 times for LM and AL are plotted, after regressing on the Peak-2 time of V1 timing. **D** The residual Peak-2 times for V1 and AL are plotted, after regressing on the Peak-2 time of LM. The partial correlation in panel C is somewhat smaller, but not dramatically smaller, than the correlation in panel A. In contrast, the partial correlation in panel D is close to zero, and very much different than the large correlation in panel B.

Fig 3 demonstrates the potential power of examining multivariate coupling relationships among features of population firing rate functions. The results in Fig 3D also indicate some circuit mechanism subtlety, especially in conjunction with Fig 4D, which shows that, for Peak-2, on average (across trials), V1 leads AL by about 5.5 ms (95% CI (3.6,8.8) ms), and AL leads LM by about 4.6 ms (95% CI (-0.2,7.3) ms). That is, it is very unlikely that the selected population of LM neurons reaches its Peak-2 maximal firing rate before that of the AL population. Furthermore, for Peak-2, as shown in Fig C.5, we do not see dramatic reduction of the correlation of V1 and LM after conditioning on AL. Perhaps LM provides important feedback input to AL that enhances its Peak-2 rise time; perhaps other areas provide such input to both AL and LM; and, to understand these results, it may be necessary to consider both excitatory and inhibitory contributions. We are not offering evidence for specific scientific explanations. Our purpose here is show that investigations of feature coupling across brain areas can generate potentially interesting findings. Other intriguing patterns may be found in C.3, C.4, C.5 and C.2.

The time lags in Fig 4.4D are also interesting in the context of the rest of Fig 4.4. Fig 4.4C displays large uncertainties in Peak-2 timing, reflecting large trial-to-trial variation (visible in Fig 4.3), and they are much larger than those of Peak-1 times shown in Fig 4.4A. Also, for Peak-2, V1 leads LM by only about half the time that its does for Peak-1 and, as shown in Fig 4.4B, in contrast to Peak-2, Peak-1 occurs at about the same for LM and AL.

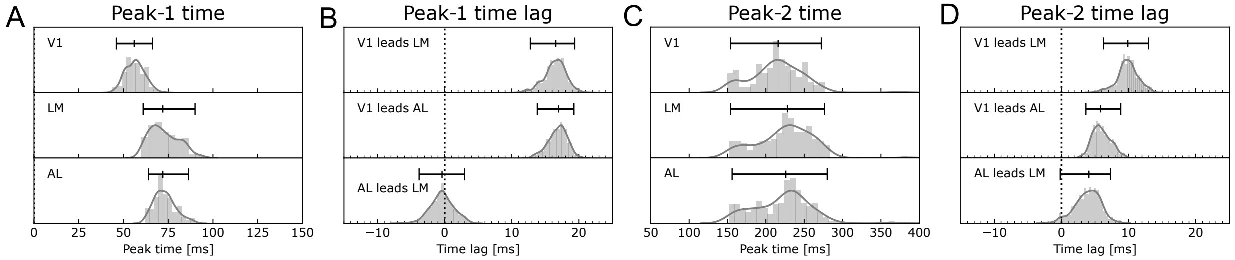


Figure 4.4: **Peak timing across areas.** Posterior distributions are shown together with median and 95% CI (.025 and .975 posterior quantiles). **A** Mean Peak-1 times for V1, LM, and AL are 57 (95% CI (46,66)) ms post stimulus onset; 68 (60,90); and 70 (64,86). **B** Peak-1 time lags between areas: V1 leads LM by 17 (3,19) ms; V1 leads AL by 17 (14, 19) ms; AL and LM are roughly simultaneous, with AL leading LM by -0.4 (-3.8,2.9) ms. **C** Peak-2 times are highly uncertain. For V1, LM, and AL they are 216 (154,272) ms; 232 (154,277) ms; 233 (156,278) ms. **D** The Peak-2 time lags between areas are much more precise than the times themselves: V1 leads LM by 9.6 (6.2,13.0) ms; V1 leads AL by 5.5 (3.6,8.8) ms; AL leads LM by 4.6 (-0.2,7.3) ms.

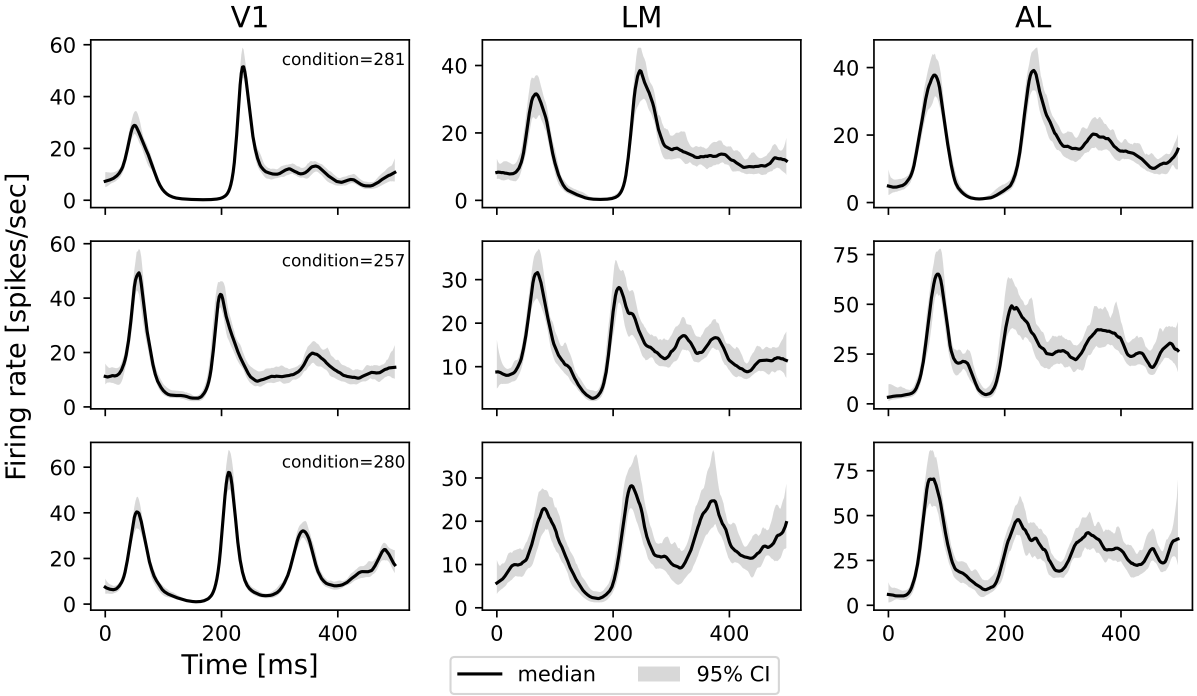
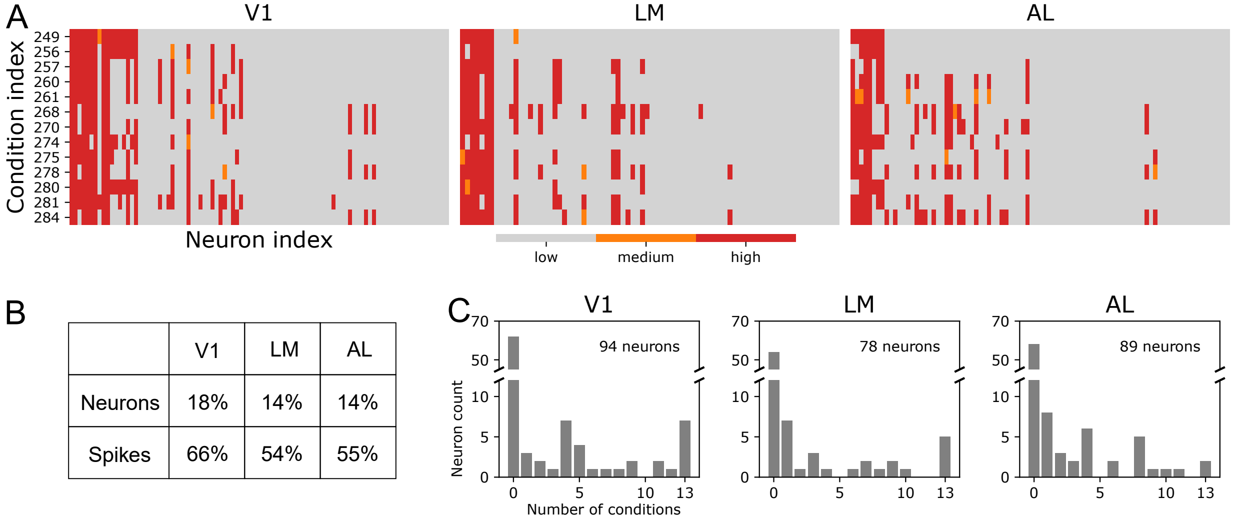


Figure 4.5: **Fitted population firing rate templates** $\exp\{f^{\mathbf{pop}}_{a,c}\}$**.** The figure shows $\exp\{f^{\text{pop}}_{a,c}\}$ in three different conditions `281`, `257`, and `280`. In condition `281` the gratings drifted at 15 Hz with orientation $315°$, in condition `257` at 8 Hz and $315°$, and in condition `281` at 8 Hz and $270°$. Note that the population firing rate function is $\exp\{n_{a,c} f^{\text{pop}}_{a,c}\}$ where $n_{a,c}$ is the number of neurons for which $z_{n,a,c} = \text{pop}$. Each row represents one condition. The solid curves and grey bands are the medians and 95% CIs from the posteriors. There are striking distinctions between these firing rate curves and those in Fig 1A. The complete set of curves is in C.7.

Population templates Fig 4.5 shows the fitted population firing rate templates $\exp\{f^{\text{pop}}_{a,c}\}$ in three different conditions. The V1 templates are very precisely determined (they have narrow posterior bands) and LM and AL templates are well determined. Our use of distinct templates for different conditions reveals both strong similarities and noticeable distinctions in the population responses across conditions.

There is an important deviation in the $f^{\text{pop}}_{a,c}$ templates from the PSTH curves shown in Fig 4.1A: Peak-2 is taller and narrower than it appears in Fig 4.1A. This is largely due to the trial-to-trial variation in peak time, which dampens Peak-2 in the PSTH. In addition, the much higher firing rates in Fig 4.5 are also affected by the more refined selection of neurons in our model than in the process leading to Fig 4.1A. More details can be found in C.9. The complete set of templates, and results for the local-1 templates, are shown in C.7.



	V1	LM	AL
Neurons	18%	14%	14%
Spikes	66%	54%	55%

Figure 4.6: **Neuron subpopulations. A** Probability that a neuron is a member of the population having firing rate template $f^{\text{pop}}_{a,c}$, for all areas and conditions. Columns are distinct neurons, rows are different conditions: red indicates the probability is high (posterior median greater than .9); gray indicates the probability is low (posterior median less than .1); orange indicates an intermediate probability. **B** Proportion of activity contributing to the population represented by $f^{\text{pop}}_{a,c}$, averaged across conditions (based on the posterior medians in A). For example, on average, 18% of V1 neurons were part of the interacting population (with template $f^{\text{pop}}_{a,c}$), but they generated 66% of the spikes. **C** Histograms that summarize the plots in panel A. For each neuron, in each area a, we count the number of conditions c having posterior median of $p_{a,c;0}$ greater than .9 (i.e., the number of red bars in the column corresponding to that neuron in panel A). The histograms display the number of neurons with count x, for $x = 0, 1, 2, \ldots, 13$. In each area, somewhat over half of the neurons never participated ($x = 0$) and those that did participate often participated in only a few conditions; in V1, 7 of the 94 neurons participated in all conditions.

Diversity of neurons Fig 4.6 summarizes membership of neurons in the communicating populations that have templates f^{pop}. As seen in panel A, for nearly all neurons it is clear whether they are fit better using the f^{pop} template or using one of $f^{\text{local-1}}$ or $f^{\text{local-2}}$. Only 22 out of 3393 neuron-condition combinations fail to have at least 90% probability (from the posterior median) of one or the other. Panel B elaborates by showing how many neurons contribute, and what proportion of spikes they generate. Panel C indicates the diversity of condition-dependent neural responses. The large variation in individual-neuron responses is perhaps unsurprising, but it underscores the importance of allowing for such diversity in statistical modeling efforts. The portion of neurons in each subgroup ($p_{a,c}$ in Eq (4.9)) is shown in C.6. Fig 4.7 shows that there is no strong spatial pattern to neuron membership in the communicating population (see also C.10).

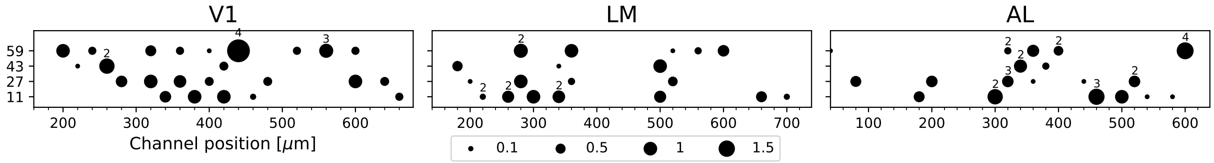


Figure 4.7: **Spatial arrangement of neurons participating in the population having template $f^{\mathbf{pop}}$.** For each area, the x and y axes label the relative positions of the electrodes, with depth along the x axis, relative to the most superficial unit. Each dot indicates, for a given location, that at least one neuron in at least one condition participated (based on posterior medians in Fig 4.6A). The size of the dot indicates spatial frequency, defined as the number of neuron-condition participation events divided by 13 (the number of conditions). When more than one neuron is recorded on an electrode, the number above the dots gives the number recorded. The locations of all subgroups in all conditions are in Fig C.10.

4.2.4 Simulations

We performed simulations to assess performance using sample sizes relevant to our data analysis, first assuming the model structure is correct as described in equations (1)-(7) and then after varying the structure in a particular way. Simulation details are in C.1. We report here results for correlations between features, which are the main interest (e.g., in Fig 3).

In our initial simulation we found small bias and root mean squared error (RMSE). The bias values of the estimated correlations are in the range $[-0.041, 0.040]$ (the mean of all pairs is -0.001). The posterior CIs are mostly close to their putative 95% coverage probability, though a few entries in the table of results are somewhat smaller (see C.3). The RMSE values are in range $[0.022, 0.090]$ (the mean of all pairs is 0.072). The range of error is slightly larger than the lower bound of the estimation. See C.1 for details. We also verified that simulation standard errors of CI end points were small relative to the RMSE values. The range of the values is $[0.0016, 0.0091]$ and the mean standard error across all end points is 0.0056.

Our implementation of the IPFR model assumes that, for the purpose of peak timing identification, trial-to-trial variability in the population firing rate intensity function can be summarized as involving only peak timing. When time warping compensates for trial-varying changes in shape, peak timing identification can be affected. One might expect such effects to be small, but we did run a simple simulation study to check. We did this by injecting noise into each neuron, in the form of intensity shapes that varied across neurons, which changes the population firing rate intensity function (see C.1). Such inhomogeneity could also affect neuron clustering, and thus potentially degrade the performance of the IPFR model. We chose the amount of injected neuron gain to roughly correspond to the real data. See the 4.4 section. (As a check, we also tried a large noise setting with 20 times the variance.) We found that the covariation estimates are not very sensitive to this type of deviation from assumptions (see details in C.1, C.4 and C.5).

4.3 Discussion

The motivating idea behind the IPFR model is to identify trial-to-trial covariation of activity across brain areas by considering covariation of population firing rate intensities. There are many ways these intensities might covary, and the IPFR model assumes a small number of features are of interest. We focused on the timing of the two dominant peaks seen in Fig 1A. Analysis of peak timing is reminiscent of ERP analysis. Although evoked potentials are based on trial averaging because they are too noisy to be useful in single trials, the IPFR model is able to estimate peak timing on a trial-by-trial basis, which makes possible assessment of trial-by-trial covariation. As with phase coupling [88], the presence of feature coupling indicates coordination of activity: it says that whatever creates variation across trials, also creates covariation across those coupled areas. We showed how the IPFR model can produce effective denoising while also revealing multivariate relationships and describing neural diversity in the participating populations.

Several clear scientific caveats should be emphasized. Feature coupling identifies a form of what is usually called functional connectivity. When we say that feature coupling reveals coordinated activity, we do not mean that it must be purposeful in a mechanistic sense. For one thing, only a few brain areas are being analyzed, and in any setting there are bound to be complicated anatomical connectivity patterns, as there are for the mouse visual system [60, 121]. But even if we were able to observe all areas that are relevant to a given task, causal relationships could not be established without causal experiments. Furthermore, while features such as peak timing are useful, they offer a limited description of interaction dynamics. In addition, despite the new capabilities provided by recording arrays such as Neuropixels, the neurons recorded from any area do not constitute a random sample of all relevant neurons. With current experimental data the extent to which recording creates important biases remains unknown.

Statistically, we have assumed that trial-to-trial variation in peak timing can be captured accurately with a model in which peak timing is the *only* source of trial-to-trial variation. Surely there are other ways that firing rate intensities vary across trials. For the accuracy of our conclusions, what matters is whether such other effects produce substantially different times at which the peaks occur and substantially different correlations of peak timing. We also assumed that a single population of neurons was most relevant for both Peak-1 and Peak-2. It is possible that treating them separately would be advantageous.

The implementation we reported here involves a comprehensive Bayesian hierarchical model. It would be possible to decompose some of the components in our IPFR model while also accounting for important sources of variation in different steps [35]. As we described, however, it was straightforward to implement EM and Gibbs sampling for the comprehensive model, which has the advantage of making it easy to get an assessment of uncertainty for any model-based quantity we wish to estimate. We hope our work will stimulate further efforts to harness the power of point process modeling for investigating timing relationships among neural populations and their coordination across brain areas.

4.4 Methods

4.4.1 Materials and pre-processing

We applied our method to the Neuropixels dataset collected by Allen institute [121]. It uses multiple high-density extracellular electrophysiology probes to simultaneously record spiking activities from a wide variety of areas in the mouse brain, especially the visual cortices. The animals were head-fixed and were passively presented with visual stimuli. The details of the experiment setup can be found in [69, 121]. One experiment contains a mixture of many stimulus types, such as natural movies, flashes, Gabor filters, drifting gratings and etc. Our paper uses drifting gratings because it has many repeated trials, the stimuli are simple, the trials are long and it can strongly elicit neural responses. The drifting gratings (type `drifting_gratings` in the dataset) have 40 conditions which are combinations of 8 different orientations (0°, 45°, 90°, 135°,180°, 225°, 270°, 315°, clockwise from 0° = right-to-left) and 5 different temporal frequencies (1, 2, 4, 8, 15 Hz). The spatial frequency is 0.04 cycles/deg and the contrast is 80%. The stimulus for each condition is repeated 15 times. A trial lasts for 3 s with 2 s stimulus and 1 s blank screen. The sequence of the conditions is randomly ordered. The baseline condition has 30 trials with a grey screen. The number of neurons in visual cortical areas recorded by one probe ranges, roughly, from 40 to 100. Usually, 6 probes are recorded at the same time. The dataset assigns unique identities for all properties. For example, a condition is labeled by `stimulus_condition_id`, a trial is labeled by `stimulus_presentation_id`, one experiment session is labeled by `ecephys_session_id`. In this paper, we refer to those identities directly.

We screened the animals and conditions based on whether the regions of interest were recorded and whether the neurons have strong responses, as the time-warping features depended on the peaks of the curves. For example, session `754829445` did not record LM and AL regions. In session `746083955`, the activity of AL was too weak and was almost the same as baseline activities (trials without visual stimulus), so AL was not able to provide any useful information. The preprocessing was done in three steps:

1. Check if the target regions are recorded by any probes.
2. Select the top 50% neurons with the largest spike counts.
3. Select the conditions with strong fluctuating responses. We first calculate the total variation [17, sec. 6.3.3] of kernel-smoothed (Gaussian kernel with standard deviation 10 ms) group PSTH for each condition and each region using the neurons in step 2. Next, sort the conditions by the sum of total variation of all regions in descending order. Then select the top conditions. The cutoff is done by visual check where the last condition has clear two-peak patterns.

We analyzed mouse session `798911424` using 13 drifting gratings conditions. Each condition had 15 repeated trials. The details of the conditions are listed in C.1. The results for V1, LM, and AL, include 94, 89 and 78 neurons respectively. Spike trains were binned in 2 ms.

In Fig 4.1A, each curve represents activity aggregated across 600 trials: there were 15 trials in each combination of 8 orientations and 5 frequencies. Only the most active 50% of neurons with large spike counts are selected. The curves are fitted using regression with splines.

4.4.2 Goodness-of-fit assessment

The goodness-of-fit test is determined by the Kolmogorov–Smirnov (KS) test based on the time-rescaling theorem [24, 62]. The KS test is a little biased as the inter-spike intervals are limited by the trial length, so only short intervals can be observed. We correct this issue by adjusting the null distribution of transformed inter-spike intervals according to the trial length. The result is shown in C.12.

Chapter 5

Interactions across populations on fine timescale

This project is co-advised by Asohan Amarasingham, and Robert E. Kass.

With the advance of the fast-growing high-density electrophysiological recording technology, hundreds of neurons from multiple brain regions can be recorded simultaneously. This offers opportunities to further investigate the interactions between brain areas, especially the spike-to-spike coupling effect on fine timescale between regions (for example within 20 ms or less), which is complementary to many studies based on the mean firing rate. The analysis of the massive spike train recordings is challenged by enormous noise, large neuronal diversity, potential artifacts caused by unobservable activities, and heavy computational tasks. These prompt us to build a flexible, extendable, robust, and computationally efficient tool to quantify the spike-to-spike coupling effect for hundreds of neurons, and to be prepared for thousands of neurons in the foreseeable near future. Our proposed point process regression model can be modularized into two basic components, which can also be extended for more: one part quantifies the coupling effect using a continuous function in a lag range; the other part is responsible for removing the artifacts caused by the background activity. The small number of parameters in the model and optimization-based inference make it efficient for large dataset analysis. We verified the model using many simulation scenarios and some theoretical analysis, then applied the method to the Allen Brain Observatory dataset and discovered trial-to-trial variation of the coupling effect on fine timescale.

5.1 Introduction

Recent advance in high-density electrophysiological recording technologies provides opportunities to explore functional interactions among populations of neurons between brain regions [121]. A recent work [73] presents evidence of spike-to-spike interactions between distant areas on fine timescale in mouse visual cortex [3]. Detecting the spike-to-spike coupling effect between two neurons is contaminated by the artifacts triggered by the shared background activities. The shared input drives two neurons in the same way so it can introduce temporal dependency between the spike trains that is not directly elicited by the spikes from the source neuron. Removing the background artifacts is challenging as they are usually unobserved and can randomly vary from trial to trial [3].

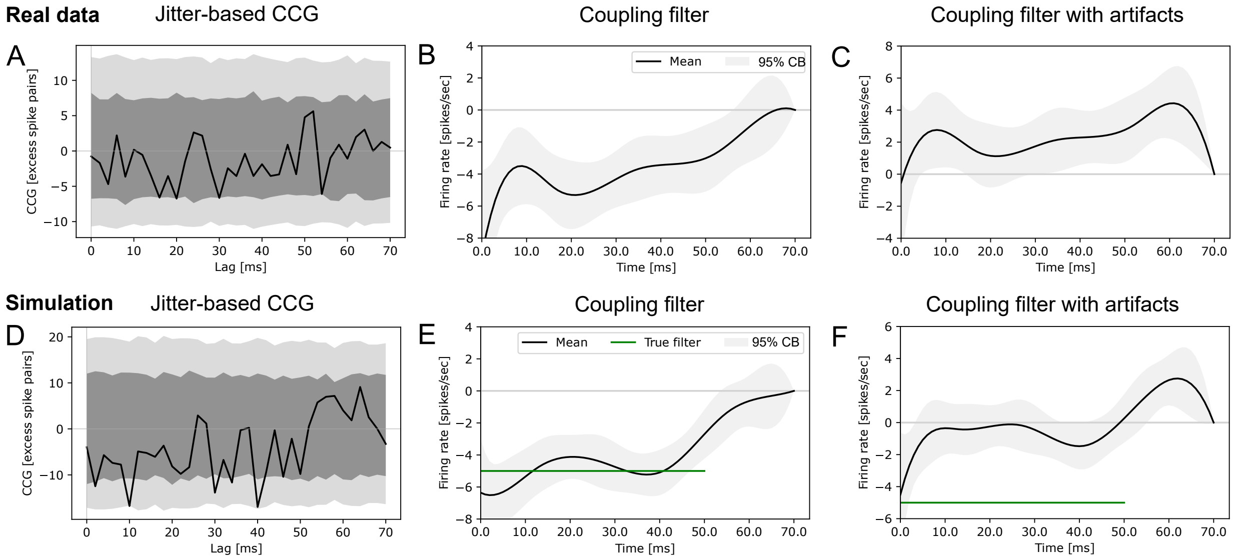


Figure 5.1: **Examples of detecting weak spike-to-spike coupling effect in a time window.** The details of this figure are in Appendix D.3.2. Here we only summarize the key conclusions. The first row shows the result of real data (Allen Brain Observatory Visual Coding Neuropixels dataset [121]) about coupling effect from one neuron in the primary visual cortex to one neuron in the lateral medial visual area. The second row replicates the scenario using a simulation with shared background activity. **A, D** In both cases, the jitter-based cross-correlogram method yields noisy output, which can detect some significant inhibitory effect at certain lags but not in a continuous range. The dark grey band is pointwise 95% confidence band, the light grey band is simultaneous 95% confidence band. **B, E** Coupling filters of the point process regression model. Our method describes the coupling effect using *coupling filters*, which shows the influence of a spike from one neuron to the firing rate of the other neuron in a lag window. Our method can detect the weak inhibitory effect. **C, F** Modeling the coupling effects assuming constant background. By comparing E and F, this assumption leads to positive bias; and the conclusion can be totally different, which detecting inhibitory effect as nearly no effect. So the regression method using a coupling filter needs careful removal of the background artifacts. This explains why the estimated coupling filter in C is above the one in B. More similar examples of real data are in Supplementary D.4.13.

The jitter-based method is designed to handle such problems that can eliminate slow-rate background artifacts without explicitly estimating the shared input between neurons. Fig 5.1 shows an example of fine time interaction from one neuron in the primary visual cortex (V1) to one neuron in the lateral medial visual area (LM) in Allen Brain Observatory Visual Coding Neuropixels dataset [121]. Fig 5.1A is the result of the jitter-based cross-correlogram (CCG). At certain time lags (V1 neuron leads LM neuron), such as 16 ms or 20 ms, the CCG curve is tangent to the bottom 95% pointwise CI, and it becomes non-significant after multiple comparison adjustment, meaning weak inhibitory influence. Although most part of the CCG curve stays within the CI band, it is attempting to hypothesize that the weak inhibitory interaction exists in the whole window from lag 0 ms to 50 ms since the majority part of the CCG curve stays below zero. Similar non-significant outcomes are very common in the dataset especially between neurons in distant regions and they are easily overlooked. Fig 5.1B is the result of our point process regression method. Our method shows significant inhibitory relation between the pair of neurons. The interaction is modeled using a continuous function called *coupling filter*, similar to the widely used point process generalized linear regression model (PP-GLM) [79, 111, 134]. The coupling filter describes how much a spike from one neuron influences the firing rate of the other neuron in a certain lag period. We replicate a similar scenario using a simulation shown in Fig 5.1D,E,F. Our method is able to detect the weak coupling effect and it is close to the true model. It is unsurprising to see the improvement: the power comes from aggregating all the information in the lag period to estimate a much simpler model. Analog to the jitter-based method, ours still needs careful removal of the background artifacts, otherwise, the result can be biased as shown in Fig 5.1F where the coupling filter is close to zero, resulting in a wrong conclusion that there is no significant coupling interaction. Fig 5.1C has similar issue. If the background artifacts are not removed, the inhibitory coupling effect can be detected as no effect or a small positive coupling effect. More examples can be found in Supplementary D.4.13.

Inspired by the jitter-based correction, our regression method eliminates the background artifacts using coarsened spike trains by adding only one covariate (nuisance variable) through all trials unlike the typical PP-GLM, which needs many parameters to model the fluctuating background [89]. If the background varies from trial to trial, the number of parameters grows with sample size which may suffer from Neyman-Scott-type issue (the bias does not vanish as the sample size grows) [114, example 3.5] and large computational burden. Synchrony detection is a special case of discovering spike-to-spike coupling effect that only focuses on zero lag events [61, 78, 83, 120, 141], while we would like to aggregate more lag information to detect weak signals. We point out that the purpose and usage of our method are slightly different from the jitter-based method though. The latter one is usually used for hypothesis testing, emphasizing the existence of the coupling effects, for example, given the timescale of the background activity (fixed in the null), if there are coupling effects at a specific lag. While our method aims to quantify the coupling effects in a time range as a continuous function taking the advantage of data aggregation, as already shown in Fig 5.1. Our method needs to find the optimal timescale of the shared activity instead of fixing that as a part of the null assumption. Determining the timescale of the background activity is part of the inference procedure. Another difference is that our model does not assume the timescale of the shared activity is larger than the timescale of the spike-to-spike coupling effect. In other words, if the shared activity changes as fast as the coupling interaction, the model is still able to eliminate the artifacts. In addition, the proposed regression model is in continuous-time (no need for time-binning), only needs very small memory space, and is robust to multiple variants. The analysis of the new tool will be presented in section 5.2.

Besides simulations, the real data we have used (available from the Allen Brain Observatory [121]) include the primary visual cortex (V1), the lateral medial visual area (LM), and the anterolateral visual area (AL), three regions that lie at different hierarchical processing stages while being highly interconnected [60]. We analyzed the coupling filters among V1→LM and LM→AL. We found the coupling filter type of a pair of neurons (excitatory, inhibitory, no effect, or others) might vary from trial to trial.

5.2 Methods

5.2.1 Point process regression model

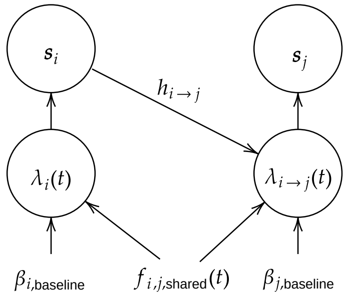

s_i
s_j
$h_{i\to j}$
$\lambda_i(t)$
$\lambda_{i\to j}(t)$
$\beta_{i,\text{baseline}}$
$f_{i,j,\text{shared}}(t)$
$\beta_{j,\text{baseline}}$

Figure 5.2: **Coupling neurons model diagram.** Two neurons i, j are driven by the same input signal $f_{i,j}$. λ_i, λ_j represent the intensity functions. $\mathbf{s}_i, \mathbf{s}_j$ are spike trains. The spikes from neuron i influences the activity of neuron j through the coupling filter $h_{i\to j}$. The goal of the model is to estimate $h_{i\to j}$ but $f_{i,j}$ challenges the estimation.

Consider a pair of coupling neurons $i \to j$ that follow the point process as shown in Fig 5.2. The intensity functions for neurons i, j are[1],

$$\begin{aligned} \lambda_j(t|\mathcal{H}_t) &= \alpha_j + f_{i,j}(t) + \int_0^t h_{i\to j}(t-\tau) N_i(\mathrm{d}\tau) \\ \lambda_i(t|\mathcal{H}_t) &= \alpha_i + f_{i,j}(t) \end{aligned} \tag{5.1}$$

$h_{i\to j}$ is the *coupling filter*, which captures the spike-to-spike interactions between neurons $i \to j$ on fine time scale, say about 30 ms. So the model is history-dependent, $\mathcal{H}_t$ represents the spikes before time t. Commonly, neurons share the same input, according to a random function of time $f_{i,j}(t)$. $N_i(\cdot), N_j(\cdot)$ are counting processes (spike count measures) for neuron i, j respectively. $N_i([t, t+\Delta]) = 1$ if there is one spike in a small interval $[t, t+\Delta]$. α_i and α_j are constant baselines. The goal is to estimate the coupling filter while eliminating the shared input artifacts.

[1] Some works model the intensity functions using log-linear form, for example, the intensity function for neuron i becomes $\log \lambda_i(t|\mathcal{H}_t) = \alpha_i + f_{i,j}(t)$. We do not think this will make a big difference if the coupling effect is small, but the linear form has great computation convenience, see Appendix D.1. Consider at a certain time, the intensity is modeled in two forms, $\lambda = \beta_0 + \Delta h = \exp\{\beta_0' + \Delta h'\}$. β_0, β_0' represent the baselines of two types of models, and $\Delta h, \Delta h'$ represent the contribution of the spike-to-spike coupling effect. If $\Delta h, \Delta h' \ll \beta_0, \beta_0'$, we approximately have $\frac{\Delta h}{\beta_0} + \log \beta_0 \approx \Delta h' + \beta_0'$. So $\Delta h, \Delta h'$ have linear relation. The sign of the coupling effect will be same; if Δh increases, $\Delta h'$ will also increase. The distinction is, approximately, a matter of scale. However, in some specific setups, the additive and multiplicative formats might have noticeable differences [70, Fig 4b with negative interdependence].

Our estimator is designed as follows,

$$\hat{h}_{i\to j} = \arg\min_{h_{i\to j}} L(h_{i\to j}) \tag{5.2}$$

$$L(h_{i\to j}) := \min_{\beta_j,\beta_w} \left\{ -\sum_{s\in N_j} \log \tilde{\lambda}_j(s) + \int_0^T \tilde{\lambda}_j(s)\mathrm{d}s \right\} \tag{5.3}$$

$$\tilde{\lambda}_j(t) := \beta_j + \beta_w\, \overline{\mathbf{s}_i}(t) + \int_0^t h_{i\to j}(t-\tau)N_i(\mathrm{d}\tau) \tag{5.4}$$

$$\overline{\mathbf{s}_i}(t) = \int_0^T W(t-s)N_i(\mathrm{d}s) \tag{5.5}$$

The shared activity $f_{i,j}$ is simply represented by just one nuisance variable $\overline{\mathbf{s}_i}$. L is the negative profile log-likelihood function. The estimator $\hat{h}_{i\to j}$ can be in a simple form, such as a square window, or can be assumed smooth and fitted using nonparametric method such as regression splines. $\overline{\mathbf{s}_i}(t)$ is the coarsened spike train smoothed by kernel W. We use Gaussian kernel with scale σ_w. Apparently, the true model (5.1) is not necessarily in the parametric family of the estimator shown in Eq (5.2)-(5.5), so the maximum likelihood estimator (MLE) in Eq (5.2) is not guaranteed to be consistent. We put tilde over $\tilde{\lambda}_j$ in Eq (5.4) to emphasize the special parametric form that might be different from the form of the true intensity (5.1).

The optimization algorithm is in Appendix section D.1. Our model treats σ_w as a tuning parameter and we do not evaluate its uncertainty due to the optimization difficulty, because it is not tractable to calculate the derivative of Eq (5.3) over σ_w. However, in some situations, for example sampling-based inference, it is possible to regard σ_w as a random variable, see Supplementary D.4.4. In the next section, we will show that with proper selection of the kernel scale σ_w, the MLE is still able to achieve some good properties such as small risk and nearly zero bias. The asymptotic Normality of the estimator is discussed in Lemma D.2.5 and verified numerically in Supplementary D.4.7. This property can be used for model inference, such as hypothesis testing, see Supplementary D.4.8. It is possible that neurons are driven by exclusive fluctuating activities f_i and f_j besides the shared one $f_{i,j}$. We introduce this model in Supplementary D.4.2. Unlike many other point process models, our method is in continuous-time, which does not need to specify the time resolution of the spike trains [42, 46], also see [34, example 8.5(a)]. The memory space thus becomes very small, which is proportional to the number of spikes instead of number of time bins.

5.2.2 Simulation and theoretical study

In this section, we study the behavior of the proposed estimator in Eq (5.2) through simulations and provide corresponding theoretical analysis. We focus on a case where the fluctuating background activity $f_{i,j}$ is a second-order stationary stochastic process, meaning $\mathbb{E}[f_{i,j}(t)f_{i,j}(t+u)]$ only depends on u but not t (the formal description is in Lemma D.2.1). A special case of the the second-order stationary process is the *cluster point process* or *linear Cox process*, which is widely used in point process study [11, 38] and [34, sec. 6.3]. We add the second-order stationary condition only to make theoretical derivations easier. The supplementary material will provide more variants.

We first generate random shared activity $f_{i,j}$, then generate spike trains. Let $\phi_{\sigma_I}(\cdot)$ be a Normal window function with center zero and scale σ_I. t_i^c are the time points of the center process determining the positions of Normal windows, which is generated by a homogeneous Poisson process with intensity ρ.

$$f_{i,j}(t) = \sum_i \phi_{\sigma_I}\left(t - t_i^c\right) \tag{5.6}$$

For simplicity, we first consider the coupling filter in form

$$h_{i\to j}(t) = \alpha_h \cdot \mathbb{I}_{[0,\sigma_h]}(t) \tag{5.7}$$

where α_h is the amplitude, and the filter length is σ_h. σ_I controls the timescale of activity. If σ_I is smaller, then $f_{i,j}$ changes faster. σ_h controls the timescale of the spike-to-spike coupling effect. If σ_h is smaller, then neuron i influences neuron j in a shorter time range. The coupling filter estimator has form $\hat{h}_{i\to j} = \hat{\beta}_h \cdot \mathbb{I}_{[0,\sigma_h]}(t)$ with just one parameter $\hat{\beta}_h$ and the timescale σ_h is known. We use the thinning method to generate continuous-time spike trains [107].

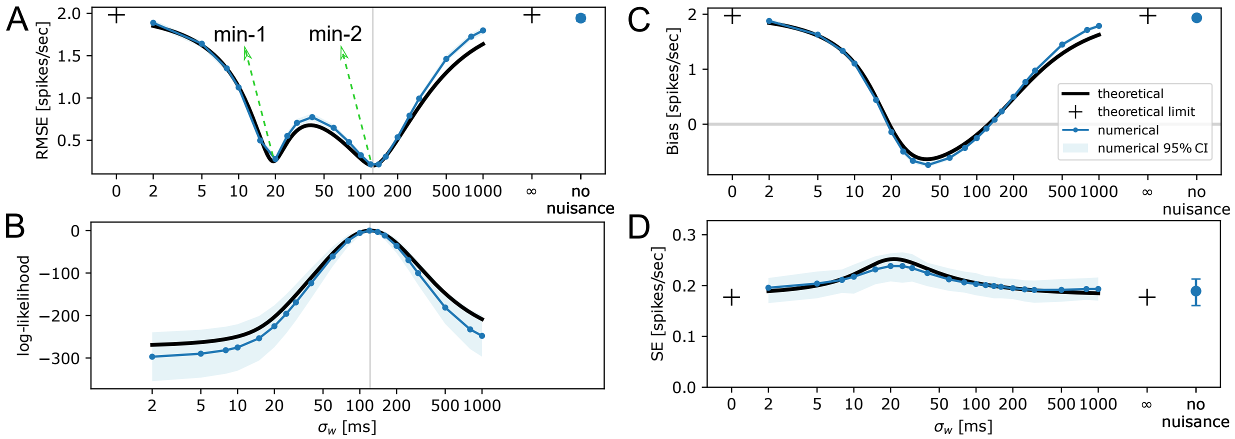


Figure 5.3: **Simulation and theoretical analysis of the estimator $\hat{\beta}_h$.** Simulation details are in the text. We show the properties of $\hat{\beta}_h$ as a function of smoothing kernel scale σ_w of W (as in Eq (5.5)). For numerical cases, we evaluate the properties at different σ_w indicated by the blue dots. The x-axis is in logarithmic scale. The numerical (blue curves) and theoretical results (dark curves) are very close. The pointwise confidence interval for RMSE and SE is calculated using bootstrap (bootstrap the replicated estimators, not the spike train data). The pointwise confidence interval for bias is calculated based on standard deviation. The blue band for the likelihood is 1.96×standard deviation. **A** The estimated risk root mean square error (RMSE) of the estimator $\hat{\beta}_h$. The two local minimums are labeled by "min-1" and "min-2". Our method prefers to select "min-2" indicated by the vertical line. The RMSE can be decomposed into bias (shown in **C**) and standard error (shown in **D**). **B** The maximum log-likelihood as function of σ_w. Since the likelihood functions may have different offsets, we align them by the peak (the maximum value across σ_w) to zero, then calculate the mean and pointwise standard deviation. The vertical line indicates the peak (numerical and theoretical peaks overlap), which matches the position of "min-2" in A. The theoretical extreme cases "0" and "∞" mean the scale σ_w of smoothing window W goes to limit 0 or ∞. The numerical case "no nuisance" represents the model without including the nuisance regressor $\overline{\mathbf{s}_i}$ in Eq (5.5), which becomes a typical Hawkes process model ignoring the fluctuating background activity.

Fig 5.3 shows the simulation results and corresponding theoretical approximations about the properties of the estimator. By theoretical, we mean the properties of the estimator, such as estimated risk or likelihood, are derived based on the second-order stationarity condition, through which we would hope to provide insights about how the timescale of the activity is linked to the behaviors of the estimator. The estimator actually is not sensitive to the second-order stationary assumption (see Supplementary D.4.1 simulations with time-varying timescale activities). The activity $f_{i,j}$ in the true model is set as a cluster process in Eq (5.6) with $\sigma_I = 100$ ms. The square window filter width is $\sigma_h = 30$ ms and $\alpha_h = 2$ spikes/sec in Eq (5.7). The firing rate of the center process $\rho = 30$ spikes/sec. The baselines are $\alpha_j = \alpha_i = 10$ spikes/sec. One simulation case has 200 trials and the length of the trial is 5 sec. Each trial is assigned with an independently generated $f_{i,j}$. The numerical properties were estimated using 100 replicated simulation cases. We have discussions about the situation where σ_h is unknown and it does not match the true coupling filter width, see Supplementary D.4.6. More complicated non-parametric fitting of the coupling filter is presented in Supplementary D.4.5. We consider a similar setting but with Laplacian window function for the cluster process in Supplementary D.4.9.

If we use a constant baseline model (as known as the Hawkes process model, or linear PP-GLM with constant baseline, or removing $\overline{\mathbf{s}_i}$ in Eq (5.4)) without considering the fluctuating background signal, the estimated coupling filter will be positively biased (Fig 5.3C "no nuisance"), as the common input between neurons will contribute to part of the estimated spike-to-spike interactions. This explains why the estimated filter in Fig 5.1F is above the true filter. Constant baseline model is equivalent to model Eq (5.5) with infinitely wide kernel $\sigma_w \to \infty$ (Fig 5.3 "∞" points) or infinitely small kernel $\sigma_w \to 0$ (Fig 5.3 "0" points), because the nuisance variable is not able to capture any shared activity (also see Corollaries D.2.2.1, D.2.2.2, and D.2.2.3). The bias of the estimator in Fig 5.3C is positive if σ_w is too wide or narrow, and the bias becomes negative between $\sigma_w = 20$ ms and $\sigma_w = 125$ ms (the theoretical derivation is in Lemma D.2.2). The gap between the numerical bias and theoretical bias becomes larger when σ_w is larger than 500 ms and this also appears in RMSE. The SE in Fig 5.3D does not change too much as σ_w changes (the theoretical derivation is in Lemma D.2.3).

The estimated risk of the estimator has two local minimums, labeled by "min-1" and "min-2" in the figure, and they are very close to zero bias solutions since SE does not change too much. If we aim to select a model with small risk, then there seem to have two solutions. We prefer to choose the model with σ_w at "min-2" (indicated by the vertical line in Fig 5.3A) for several reasons: 1, the slope of the risk curve around "min-2" is smaller than the slope near "min-1" (the x-axis of the figure is in logarithmic scale), so the model is less sensitive to the selection of σ_w; 2, from a practical point of view, "min-2" can be selected by maximizing the log-likelihood over candidate σ_w (Fig 5.3B), which agrees with the model selection methods based on likelihood, such as AIC or BIC; 3, as will be shown shortly, the position of "min-2" is related to the timescale σ_I of $f_{i,j}$ and it is invariant of coupling filter scale σ_h or amplitude α_h. The optimal $\sigma_w = 125$ ms of the smoothing kernel is closer to the scale of the background activity $\sigma_I = 100$ ms; So the nuisance variable, the coarsened spike train $\overline{\mathbf{s}_i}$ (as in Eq (5.5)), can be interpreted as a approximation of the background activity, and σ_w reflects the timescale of that. 4, if the coupling filter is fitted using non-parametric method, the risk at "min-1" will be much higher than the risk at "min-2". This will be illustrated in Supplementary D.4.5. Next, we explore how these properties are related to the shared activity and the coupling effect.

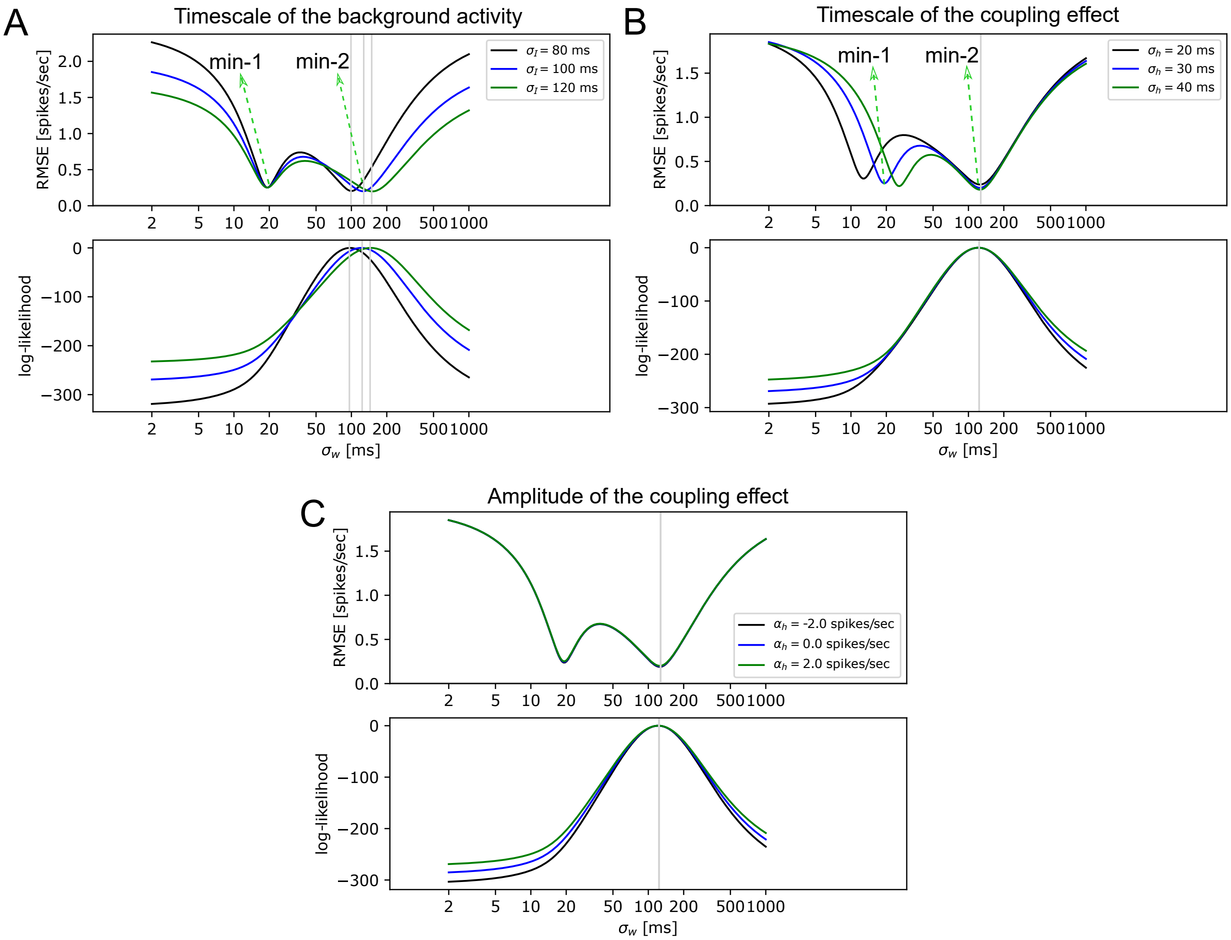


Figure 5.4: **Influences of background activity timescale, coupling filter timescale, and coupling filter amplitude on the estimator $\hat{\beta}_h$.** We show the RMSE and log-likelihood curves as in Fig 5.3. The settings are the same as Fig 5.3 except for different σ_I in A, different σ_h in B and different α_h in C. This figure only shows the theoretical results. The numerical results are very close (data not shown). The log-likelihood functions may have different offsets, we align them by the peak to zero (maximum value across σ_w). The local minimums of the risk are labeled by "min-1" on the left and "min-2" on the right for each case. **A** The RMSE and likelihood curves with different $\sigma_I = 80, 100, 120$ ms. $\sigma_h = 30$ ms and $\alpha_h = 2$ spikes/sec are fixed. If σ_I increases, "min-2" shifts to the right, while "min-1" does not move. The peak of the likelihood function also moves accordingly and it is aligned with "min-2" (indicated by the grey vertical lines). **B** The RMSE and likelihood curves with different $\sigma_h = 20, 30, 40$ ms. $\sigma_I = 100$ ms and $\alpha_h = 2$ spikes/sec are fixed. If σ_h increases, "min-1" shifts toward right, but "min-2" and the peak position of the likelihood function do not change (grey vertical lines). **C** The RMSE and likelihood curves with different $\alpha_h = -2, 0, 2$ spikes/sec. $\sigma_I = 100$ ms and $\sigma_h = 30$ ms are fixed. α_h does not affect the risk of the estimator, but it changes the shape of the likelihood function slightly.

Fig 5.4 shows the relations between the estimator's properties and the timescale of the shared activity (σ_I of $f_{i,j}$ in Eq (5.6)), the timescale of the spike-to-spike coupling activity (σ_h of the coupling filter in Eq (5.7)), and the amplitude of the coupling filter (α_h in Eq (5.7)). The scale σ_I of the shared activity $f_{i,j}$ is related to "min-2" and the peak of the log-likelihood function (Fig D.1A). If σ_I is larger, the optimal σ_w also becomes larger. The scale σ_h of the coupling filter is related to "min-1", but it does not change the peak position of the log-likelihood (Fig D.1B). If σ_h becomes smaller, "min-2" does not change, but "min-1" shifts toward left. The amplitude of the coupling filter (α_h in Eq (5.7)), whether it is positive or negative, does not change the position of the two local minimums (Fig D.1C). Our method does not restrict the timescale of the background is larger than the coupling effect. The "min-2" associated with σ_I is not always on the right side, if σ_I decreases to a very small values where the the background can change as fast as the coupling effect, the positions of "min-1" and "min-2" will be switched, see Supplementary D.4.3. The above properties suggest a second way of choosing σ_w. As the optimal σ_w does not depend on the timescale or amplitude of the coupling effect, we can use a predetermined plug-in estimator for σ_w (Appendix D.1.2).

We also explore many other properties of the estimator. Here we briefly summarize the key conclusions. The details are in the supplementary session. The scenarios include:

1. Supplementary D.4.1. Timescale-varying background. The timescale σ_I of the background is no longer a fixed value (Eq (5.6)), it randomly changes from time to time. In this case, it randomly changes between 80 and 140 ms. $f_{i,j} = \sum_i \phi_{\sigma_{I,i}}(t - t_i^c)$, where every time point of the center process t_i^c is assigned with a different scale $\sigma_{I,i}$. The process changes faster at smaller $\sigma_{I,i}$, and changes slower at larger $\sigma_{I,i}$. The selected kernel width σ_w balances the varying timescale and the selected estimator can still get small risk and low bias as will be shown through simulations. However, if the shared activity is a mixture of very distinct timescales, for example a compound of 100 ms and 5 ms, the selected model with a single smoothing kernel W (Eq (5.5)) can not balance both timescales and some bias still exists. We will further address this in Supplementary D.4.10, where the fast-changing activity is driven by the spike trains of a subpopulation, and the multivariate regression is a remedy for the issue if the driving spike trains can be observed.

2. Supplementary D.4.2. Non-shared fluctuating background. Besides the shared background activity $f_{i,j}$, neurons can be driven by other sources of activities, which are not shared between neurons. This scenario breaks the framework shown in Fig 5.2. The non-shared inputs do not affect the model selection. Surprisingly, the non-shared components help the estimation by decorrelating the background and the coupling effect as it decorrelates the inputs of two neurons. The bias and risk become smaller.

3. Supplementary D.4.3. Fast-changing background. The shared activity $f_{i,j}$ has very small timescale σ_I. If σ_I is set as a small value such as 20 ms or even 5 ms, which is smaller than the spike-to-spike interaction timescale, the model is still able to accurately estimate the coupling effect. This is seen as a significant advantage over the jitter-based method, which cannot handle fast-changing background. The method can split the background effect and coupling effect as the former one is undirectional and the latter one is directional effect between neurons. The jitter-based method or similar bootstrapping methods can break the subtle fine timescale effect.

4. Supplementary D.4.4. Bayesian model. The regression model is built on likelihood and can be adopted for Bayesian inference where the smoothing kernel width σ_w can be treated

as a random variable. σ_w reflects the timescale of the shared activity as already shown in Fig 5.4, but the uncertainty of σ_w is not directly related to the variance of the timescale σ_I of the shared activity (if it varies like Supplementary D.4.1). So σ_w should be interpreted as a representation of the average shared activity timescale but not the whole range of the timescale. Incorporating the uncertainty of smoothing kernel width σ_w does not change the estimation of the coupling effect too much.

5. Supplementary D.4.5. Non-parametric fitting for the coupling filter. The coupling filter is fitted using non-parametric method. This example shows the versatility of the regression method and the connection to the commonly used point process GLM [79, 111, 134].
6. Supplementary D.4.6. Selection of coupling filter length. In practice, usually the range of the coupling effect is unknown. If the coupling filter length σ_h is shorter than the true coupling filter length, then the properties do not change a lot. But if the coupling filter of the estimator is longer than the truth, then the bias becomes larger, and the kernel width selection does not match the optimal risk. So if users are not confident with the coupling filter length, it is recommended to use a shorter coupling filter or non-parametric fitting as in Supplementary D.4.5.
7. Supplementary D.4.7. Asymptotic Normality of the estimator. The Normality of the estimator's distribution is verified using simulations. The property can be used for model inference, for example calculating p-values in hypothesis testing (Supplementary D.4.8).
8. Supplementary D.4.8. Hypothesis testing example. We present examples of hypothesis testing based on the proposed regression model. We verify that the p-value distribution under the null is uniform and demonstrates the power of the estimator using weak coupling effects under a small sample size. The performance of our method is better than the jitter-based CCG.
9. Supplementary D.4.9. Background activity with Laplacian window function. The Gaussian window function in Eq (5.6) used to generate fluctuating background activity is replaced by the Laplacian window function, which has a sharper shape at the center and thicker tails. The performance of the estimator and the conclusion are the same.
10. Supplementary D.4.10. Multivariate regression and partial relation. It is very natural to extend the bivariate model to a multivariate regression model. The multivariate regression model can handle the shared input artifacts with a mixture of very distinct timescales under some circumstances. In this simulation scenario, a subpopulation Z drives two neurons X and Y through coupling filters on a very fast timescale together with slow-changing background activity. The bivariate model Eq 5.2 can not remove the artifacts caused by both slow-changing background and fast-changing spike train-driven input. But if the driving spike trains are observed, the multivariate regression model can eliminate all artifacts.
11. Supplementary D.4.11. Self-coupling effect. This problem comes from real data goodness-of-fit test. The self-coupling effect could lead the basic model in diagram Fig 5.2 to select a smaller smoothing kernel width than the optimal one minimizing the risk. It also undermines the goodness-of-fit. However, the introduced extra bias is small. When replicated the real data analysis in section D.4.15 with different kernel widths, the conclusion did not change.

12. Supplementary D.4.12 Rate coupling and delayed shared input. Besides spike-to-spike coupling, we consider *rate coupling* between neurons. A special case of rate coupling is the delayed shared input. The shared component arrives at two neurons with different delays, showing a fine timescale coupling between the underlying instantaneous firing rates. The coupling between the firing rates does not affect the estimation of the spike-to-spike coupling effect.

5.3 Application to neuroscience data

Neuroscience data are noisy, and usually there are not enough repeated trials to make very detailed estimation of the coupling filters between all pairs. Therefore, we start with a simple model that can identify the basic types of the spike-to-spike interaction, such as excitatory, inhibitory, no coupling effect, or others. We applied our method to the Allen Brain Observatory Visual Coding Neuropixels [121]. Details of the materials and the algorithm are in Appendix D.3. We found the coupling filter type was not always the same and changed from trial to trial.

We use the coupling filter templates to identify the type of all coupling filters on each trial. Fig. 5.5 shows the fitted templates of the coupling filters. Not all filters fit into the three categories, so we add two more groups. It is important to use a powerful tool to identify the coupling filter type properly. Otherwise, it may not be able to detect the weak effect or get a wrong type, see Fig 5.1 and Supplementary D.4.13. Without properly removing the background artifacts, as already shown in Fig 5.3 the bias is positive if the fluctuating background is not considered, type-0 "no effect" can be regarded as type-1 "excitatory", and type-2 "inhibitory" can be regarded as type-0 "no effect" or type-1 "excitatory", see Supplementary D.4.13.

- Type 0: No coupling effect.
- Type 1: Positive square window coupling filter representing the excitatory effect. The filter length is 50 ms. Usually the excitatory effect is weak, which is similar to the case in Fig 5.1, so we just use one variable for the coupling filter.
- Type 2: Negative square window coupling filter representing the inhibitory effect. The filter length is 50 ms. The design is similar to the excitatory coupling filter.
- Type 3 and type 4: Oscillatory-shape filters but with different phases. The tail is small. The filter length is 40 ms.

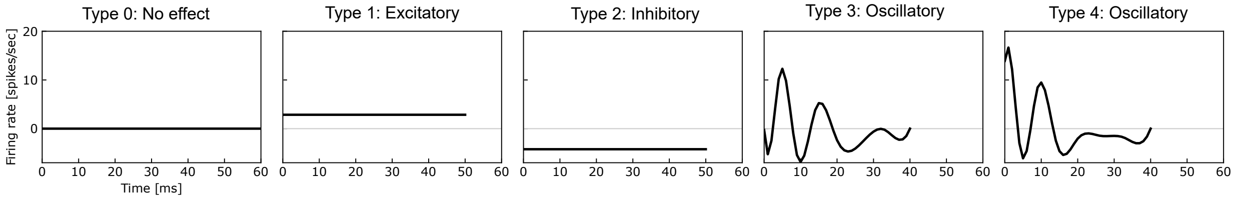


Figure 5.5: **Fitted coupling filter templates.** The first three represent no coupling effect, excitatory, and inhibitory coupling filters. No all filters fit into these three categories, so two more templates are included with oscillatory shapes. The phase between type 3 and type 4 are different.

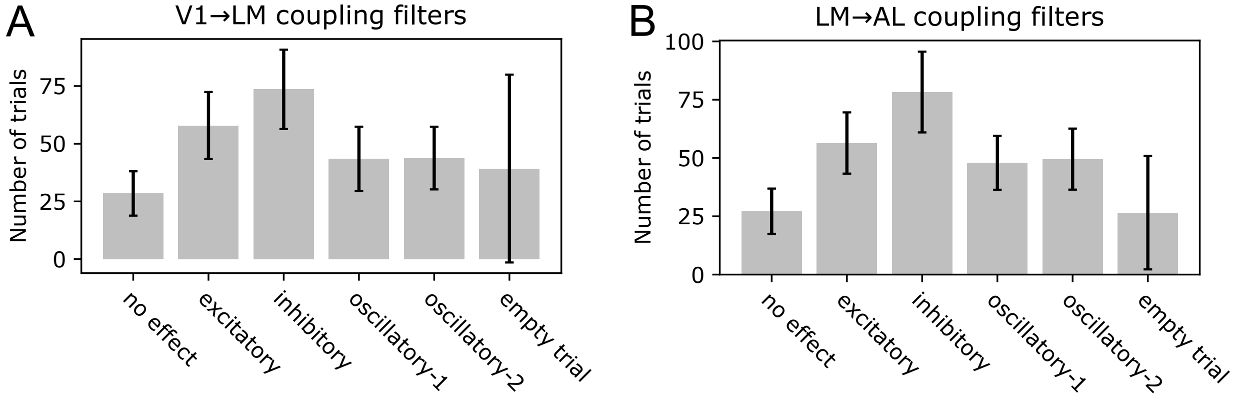


Figure 5.6: **Frequency of coupling filter types of all coupling filters.** Our method identifies the type of a coupling filter on each trial. Totally there are 285 trials for each pair of neurons. The histograms show the mean number of trials of each type across all pairs among V1→LM in **A** and LM→AL in **B**. The error bar is the standard deviation of the number of trials.

Next, we identified the coupling filter types of two pairs of neurons on all trials. Fig 5.6 counts the frequency of each type of all trials. There are cases where the source neuron or the target neuron do not generate spikes in the whole trial window, so we can not tell what the coupling filter type is. The trials without observed spikes are categorized into "empty trial". The coupling filter type is not always the same and it varies from trial to trial. Our analysis totally includes 672 coupling filters among V1→LM, and 648 coupling filters among LM→AL. Fig 5.6 shows the mean and standard deviation of the frequency of each coupling filter type. In both Fig 5.6A and B, inhibitory type has higher frequency. Since the filter shapes are noisy and the filters in types 0, 1, 2 are very close, some clusters might merge into one. To avoid this issue, we fix the group weights similar to the k-means clustering instead of updating the group weights in a typical mixture model, see details in Appendix D.3.3. Fixing the group weights for all trials does not introduce trial-to-trial variance. We tested the method using different weights, see Supplementary D.4.15. The portion of each type is sensitive to the group weights due to the small sample size on each trial.

We assessed the goodness-of-fit using Kolmogorov–Smirnov test based on the time rescaling theorem [24, 62]. The results are shown in Supplementary D.4.16. Most coupling filters have a good fit except for a few pairs possibly due to the self-coupling effect. The self-coupling effect can lead the model to select a smaller smoothing kernel width, see Supplementary D.4.11. We repeated the analysis using a larger and a smaller smoothing kernel width (σ_w of W in Eq (5.5)), the results are very similar, see Supplementary D.4.15. We repeated the analysis using totally different conditions but the same animal. The conclusion is the same, see Supplementary D.4.17.

5.4 Discussion

One motivation for this work is to develop a flexible, robust and computational friendly tool to handle large electrophysiological dataset with multiple regions simultaneously recorded. Larger and more complicated recordings demand that the modeling framework incorporate more factors. Another motivation is about extendable modeling. The purpose of modeling changes from case to case. We avoid designing a tool that is highly constrained by specific assumptions or formats. The framework of the model ought to be easily modified, new components can be freely plugged in, and different components are ideally separated.

These requirements lead to the proposed model in Eq 5.2-5.5, which is modularized into two components through the regression framework: the coupling filter component is the main interest of this work, that is used for inferring spike-to-spike interactions; the kernel smoothed spike train $\overline{\mathbf{s}_i}$ in Eq (5.4) is responsible for removing the artifacts caused by the fluctuating background. The optimization-based inference is more efficient than sampling-based (while the method can still be used to the latter, see Supplementary D.4.4). The model does not need to specify time resolution and the memory complexity is small. Rather than detecting whether the coupling effect is significant or not, we are more interested in quantifying the effect and build uncertainty around that so it can be used for more applications, as opposed to only constructing the null distribution of a statistic (while it can still be used for hypothesis testing, see Supplementary D.4.8, where build CI for the estimated coupling filter instead of having its null distribution). The modularized design is flexible since both parts can be adjusted separately for different purposes in various situations, which also has a connection with the widely used point process GLM[79, 111, 134]. For example, if the timescale of the coupling filter is shrunk into a few milliseconds, the model can also be used for synchrony detection [61, 78, 83, 120, 141] (simulations or data analysis are not shown as detecting the effect at only one time lag is not the main goal of the paper). New components, for example latent variables, can be added to the intensity function in Eq 5.4. But this may lose the convenience of continuous-time modeling if the new component can not be integrated in closed-form (see Appendix D.1). As long as the target equation Eq 5.2 is differentiable over the coefficients, the optimization is not a big problem. Many variants of the dataset or models are presented in the Supplementary. Robustness also comes from modularization. The coupling filter can be adapted to the characteristics of the data without changing the nuisance variable for the background activity (see Fig 5.4B,C). We do not assume the timescale of the background is larger than the coupling effect, or vice versa, see Supplementary D.4.3. For example, as shown in Fig 5.1, if the coupling effect is weak, the coupling filter can be simplified with less parameters so that more information in a lag range can be collected for the estimation, thus it is less sensitive to the noise at one time lag. Leveraging the advantages of kernel smoothing, the nuisance variable does not need customized design for a special type of the background activity. The smoothing kernel width can be selected automatically, and the result is not sensitive to the selection, see Fig 5.3 and Supplementary D.4.4. So the model can be easily applied to massive dataset with minimal manual intervention.

In our work, the model used for inference is not the same as the specified true model. The background activity $\overline{\mathbf{s}_i}$ is not assumed to be in the same format as $f_{i,j}$ in Eq 5.1 and 5.4. If the inference model and the assumed true model were the same, when the true model needed to be modified, possibly it would totally overturn the inference procedure, thus limits the flexibility of the modeling. However, the cost of our model is that the MLE is no longer guaranteed to have nice properties such as consistency or asymptotic Normality, etc. We spend a lot of efforts on justifying that even the true model is not in the special parametric family of the inference model, it can still achieve satisfactory performance.

Next, we would like to point out some drawbacks of the model and analysis. We add the second-order stationary condition for the background activity to make the theoretical derivations simple (Lemma D.2.1, Lemma D.2.7). However, if the timescale of background changes from time to time, it can break this assumption, but the model can still hold the same properties as in a second-order stationary process, see Supplementary D.4.1. This means the discovered properties hold in weaker conditions than the second-order stationarity. However, we have not found such general necessary conditions from either numerical or theoretical point of view. Our theory (Lemma D.2.5) shows that if the estimator of the coupling effect is very close to the true model, then the estimator is asymptotically normal. Numerical evidence shows that even the estimator has non-negligible bias, it can still have normal distribution. We have not provided a theoretical explanation for this. Another shortcoming of our work is that we have neither strictly proved why the estimator selected by maximizing the likelihood can minimize the risk and get low bias, nor provided a bound on the model selection error. Instead, we showed that the estimator maximizing the likelihood and that minimizing the risk agree with each other very well, through theoretical approximation (for example, Fig 5.3 A and B dark curves), and provide numerical simulations to verify the properties of the estimator empirically (for example, Fig 5.3 A and B blue curves). The regression model fails in eliminating the artifacts if the background activity is a mixture of very distinct timescales, for example the timescales are compounds of 5 ms and 100 ms. We partially address this issue using multivariate regression in Supplementary D.4.10. But we think a more general solution can be an extension of the current model by incorporating multiple levels of background artifacts, at least two levels. As shown in Supplementary D.4.10, the bias is significant only when the fast-changing component is very strong. We will leave the research of this type of issue in the future. But we first need to find out if such strong fast-changing background exists in real data. If a coupling filter is fitted by a square window, and it is longer than the true range of coupling effect, it can disturb the model selection. So we recommend shorter square window if users are unsure about the range of the coupling effect, or exploring with non-parametric fitting or CCG first, see Supplementary D.4.6. Our theory has not covered self-coupling effect yet, and we realize that this effect can mislead the model selection and undermines goodness-of-fit in some situations, see Supplementary D.4.11. But the introduced error is minor. We will fix this issue in the next step of our work.

Appendix A

Appendix for Chapter 2

A.1 Datasets

Data	*Monkey*	*Human*	*Goldfish*
Feature	inhomogeneous	trial-to-trail variability	bursty
# Trials	10	10	1
Trial duration (sec)	1	10	30
Mean firing rate (Hz)	24.0	1.0	32.4
Source	Gerhard et al.	Gerhard et al.	Tokdar et al.

Table A.1: Details of the datasets used in this paper. The *Monkey* and *Human* datasets [53] consist of single unit recordings from monkey cortex PMv and M1 areas, and from the neocortex of a person with a pharmacologically intractable focal epilepsy, respectively. The *Goldfish* dataset [96, 132] consists of recordings from retinal ganglion cells in vitro that exhibit bursting firing.

A.2 Diagnostic Maps

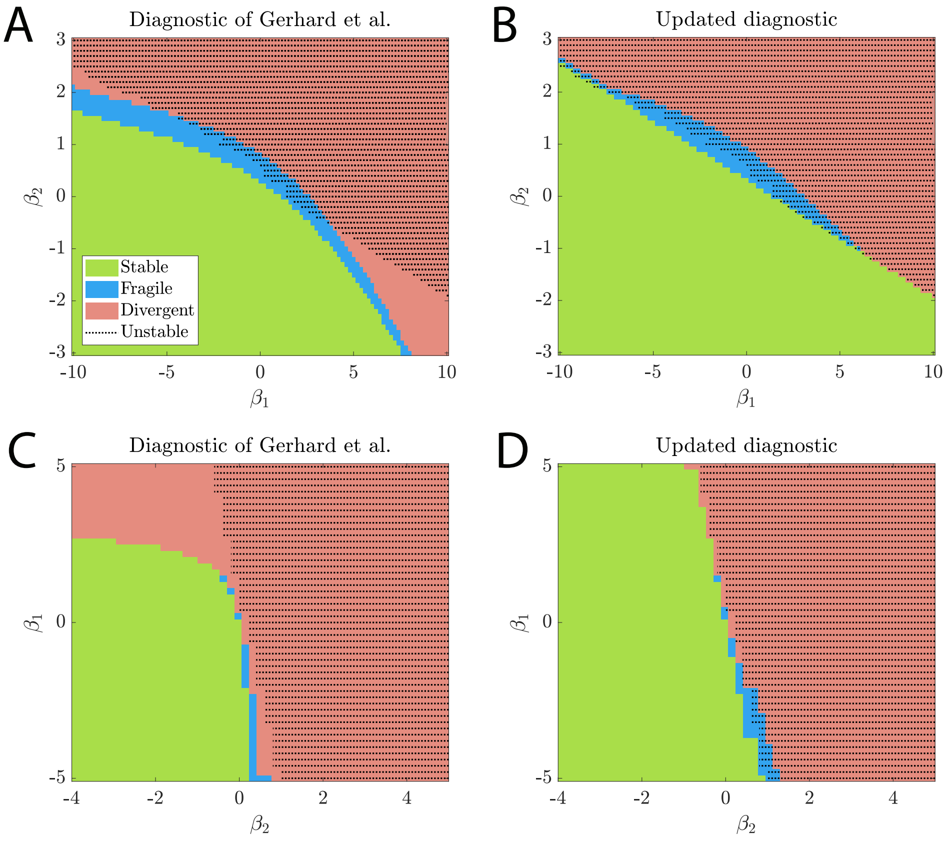


Figure A.1: Stability maps for two FLF models (Equation 2.4) F_θ using the diagnostic of [53] and our updated diagnostic. For each value of θ, we (i) simulated a 10 sec. long spike train from F_θ, and deemed the model unstable if it generated over 900 spikes in the last second, (ii) produced the diagnostic curve and determined from it if the model was stable/fragile/divergent, and (iii) plotted θ against the outcomes in (i) and (ii). (A) Reproduction of the stability map in [53] Figure 4, where F_θ is an FLF model with $\beta_0 = -5.3$ and $h(t) = \beta_1 \cdot \mathrm{B}_1(t) + \beta_2 \cdot \mathrm{B}_2(t) + \mathrm{Dip(t)}$, where $\mathrm{B}_1(t) = e^{-t/0.02}$, $\mathrm{B}_2(t) = e^{-t/0.1}$ and Dip(t) is a negative window function modeling a 2 msec. refractory period, $\theta = (\beta_1, \beta_2)$, and filter length $T_h = 0.2$ sec. The maps suggest that the diagnostic is mostly reliable, except in small regions of the parameter spaces. (B) Our updated diagnostic for the same model matches the simulation better. (C) Stability map using the diagnostic of [53] for F_θ an FLF model with with $\beta_0 = -4$, $h(t) = \beta_1 \cdot \mathrm{B}_1(t) + \beta_2 \cdot \mathrm{B}_2(t)$. Basis $\mathrm{B}_1(t)$ and $\mathrm{B}_2(t)$ are the same as Figure 2.2A. $\theta = (\beta_1, \beta_2)$, and filter length $T_h = 0.35$ sec. (D) Our updated diagnostic for the same model matches the simulation better.

A.3 Simulation algorithms

Algorithms 1, 2, and 3 generate *Izhikevich-xx*, FLF, and FNF datasets, respectively.

Algorithm 2: Izhikevich simulation algorithm. The Izhikevich dynamical model can generate a rich family of biophysically realistic spike patterns [71, 72], including time varying rate pike trains, tonic spikes, bursts, etc. A list of parameters to produce various effects is given in [137].

```
Input: Parameters $a, b, c, d$. Time resolution $\Delta$. Time varying spiking intensity $I(t)$.
Initial: $u(0) = 0$, $v(0) = 0$, $\mathbf{S} = \{0\}$
for $t = 0$ to $T$ do
    $\mathrm{d}v = 0.04v(t)^2 + 5v(t) + 140 - u(t) + I(t)$
    $\mathrm{d}u = a(bv(t) - u(t))$
    $v(t+1) = v(t) + \mathrm{d}v \cdot \Delta$
    $u(t+1) = u(t) + \mathrm{d}u \cdot \Delta$
    if $v(t+1) > 30$ then
        $v(t+1) = c$
        $u(t+1) = u(t) + d$
        $\mathbf{S} = \mathbf{S} \cup \{t+1\}$
    else
        continue
    end
end
Output: S
```

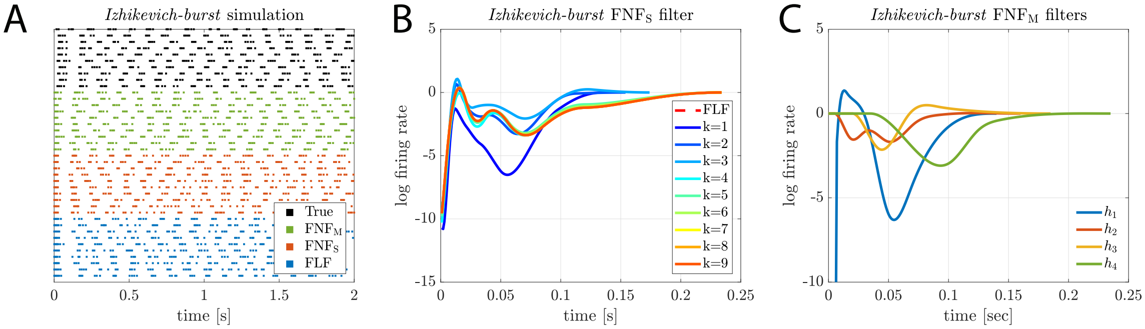


Figure A.2: (A) *Izhikevich-burst* synthetic dataset spike trains and simulated spike trains from FLF, FNF$_S$(k=4), FNF$_M$(k=4). Both type of models can generate busts similar to those in the dataset. (B) Fitted filters of the FNF$_S$ models with number of spikes $k = 1, ..., 9$. When $k > 3$, the filters are very close to each other since further spikes will not make too much contribution to the future firing rate, thus will only affect the filter shape slightly. The fitted FLF filter overlaps with the FNF$_S$ filters with $k \geq 5$, suggesting that these models are functionally similar. (C) Fitted filters of an FNF$_M$ model with $k = 4$: the filters are substantially different, which suggests that past spikes of different order have different effects on the firing rate. A likelihood ratio test comparing the FNF$_S$(k=5) and FNF$_M$(k=5) models favors the FNF$_M$ model ($p \ll 0.001$).

Algorithm 3: FLF model (Equation 2) simulation algorithm. This algorithm simulates ISIs from a unit rate exponential distribution and inverts them using the time rescaling theorem [24, 79] to obtain the past spike times.

```
Input: time resolution $\Delta$, baseline $\beta(t), t \in [0, T]$, and post-spike filter $h$ with length $L$;
define $f(s_1, s_2) := \sum_{t=s_1}^{s_2} \lambda(t|H_t)\Delta$, $0 \leq s_1 < s_2 \leq T$, assume $\lambda(t|H_t) \geq 0$;
Initial: $\mathbf{S} = \emptyset$ be the set of spike time points;
$\lambda(t|H_t) = \beta(t)$
$t_{1*} = 0$
while TRUE do
    draw one sample $Z \sim \mathrm{Exp}(1)$;
    if $f(t_{1*}, T) < Z$ then
        return
    else
        $s = \arg\min_{\tau}\{f(t_{1*}, \tau) \geq Z\}$
        $\mathbf{S} = \mathbf{S} \cup \{s\}$
        $t_{1*} = s$
        Update the firing rate function by adding the impact of the new spike to the
          future firing rate: $\log\lambda(\tau + t|H_t) = \log\lambda(\tau + t|H_t) + h(\tau)$, for all
          $\tau \in [0, \min(L, T - t)]$
    end
end
Output: S, $\lambda(t|H_t)$
```

Algorithm 4: FNF model (Equation 3) simulation algorithm.

```
Input: time resolution $\Delta$, baseline $\beta(t), t \in [0, T]$, and $k$ post-spike filter $h_i$ with length $L_i$
define $f(s_1, s_2) := \sum_{t=s_1}^{s_2} \lambda(t|H_t)\Delta$, $0 \leq s_1 < s_2 \leq T$, $\lambda(t|H_t) \geq 0$ is the total firing rate
Initial: $\mathbf{S} = \emptyset$ be the set of spike time points
$\lambda(t|H_t) = \beta(t)$
$t_{1*}, t_{2*}, ..., t_{k*} = 0$
while TRUE do
    draw one sample $Z \sim \mathrm{Exp}(1)$
    Update the firing rate function $\log \lambda(t|H_t) = \beta(t) + \sum_{i=1}^{\min(k,|\mathbf{S}|)} h_i(t - t_{i*})$, if $t > L_i$, $h_i(t) = 0$. $t_{i*}$ are the last i'th spike.
    if $f(t_{1*}, T) < Z$ then
        return
    else
        $s = \arg\min_{\tau}\{f(t_{1*}, \tau) \geq Z\}$
        $\mathbf{S} = \mathbf{S} \cup \{s\}$
        $t_{1*} = s, t_{2*} = t_{1*}, ..., t_{k*} = t_{(k-1)*}$
    end
end
Output: $\mathbf{S}$, $\lambda(t|H_t)$
```

A.4 Misc. results

Derivation of Equation 2.7

$$\sum_{t_{j*}\in(t-T_h,t_{1*})} h(t-t_{j*}) \approx \mathop{\mathbb{E}}_{N}\Big[\int_{t-T_h}^{t_{1*}} h(t-\tau)\mathrm{d}N(\tau)\Big] \tag{A.1}$$

$$= \mathop{\mathbb{E}}_{N_\Delta}\left[\mathop{\mathbb{E}}_{N|N_\Delta}\Big[\int_{t-T_h}^{t_{1*}} h(t-\tau)\mathrm{d}N(\tau)\Big|N_\Delta\Big]\right] \tag{A.2}$$

$$\overset{t-\tau=u}{=} \mathop{\mathbb{E}}_{N_\Delta}\Big[\frac{N_\Delta}{t_{1*}-t+T_h}\int_{t-t_{1*}}^{T_h} h(u)\mathrm{d}u\Big] \tag{A.3}$$

$$= A_0\int_{t-t_{1*}}^{T_h} h(u)\mathrm{d}u \tag{A.4}$$

where $N_\Delta = N_{(t-T_h,t_{1*})}$ is the number of spikes in $(t-T_h, t_{1*})$, and A_0 is the mean firing rate in that time window. In equation A.2, the inner expectation is taken over spike count conditioned on a fixed number of spikes in the interval $(t-T_h, t_{1*})$. If the filter $h(u)$ is estimated by $e^{h(u)}-1$ in [53], the error will be larger if $h(u)$ is not close to 0. Because the point process itself is unknown, the firing rate function is approximated under the assumption that it is a homogeneous Poisson process. For homogeneous Poisson process, if the number of events is fixed, they distribute evenly in the interval, which leads to equation A.3.

The time rescaling theorem Let $Z_i = \int_{t_{i-1}}^{t_i}\lambda_0(t)dt$, where t_i are spike times, Z_i are time integral transformed intervals. Time rescaling theorem states that if $\lambda_0(t)$ is the firing rate of the true model, then Z_i are iid and $Z_i \sim \text{Exp}(1)$. The goodness-of-fit test checks how close the distribution of transformed intervals from estimated model is to the unit exponential distribution.

Appendix B

Appendix for Chapter 3

B.1 Simulation study

First, we created a series of PP-GLMs with coefficients $\boldsymbol{\beta}^0(g), g \in \{g_1, ..., g_B\}$. These parameters corresponded to the models with different ion channel conductance scaling factors. Differences between adjacent models $\boldsymbol{\beta}^0(g_i)$ and $\boldsymbol{\beta}^0(g_{i+1})$ were small and the trend was smooth. The parameters came from a previous fit. Then for each $\boldsymbol{\beta}^0(g_i)$, we simulated 100 3-second spike trains according to Eq. 3.1. The simulated spike trains were then used to fit new PP-GLMs in Eq. 3.6, and the penalty hyperparameter λ was selected as described above in Eq. 3.7. We expected to see that after applying the trend filtering technique, the model could recover the trend of changes despite the Poisson-like noise from the spike trains. We repeated the above procedures 100 times to acquire the mean and the variance of the error. Besides trend recovery simulation, we also checked the goodness-of-fit using KS test based on time rescaling theorem [24, 62]. All fitted models had good performance (data not shown). The results are shown in Fig. B.1.

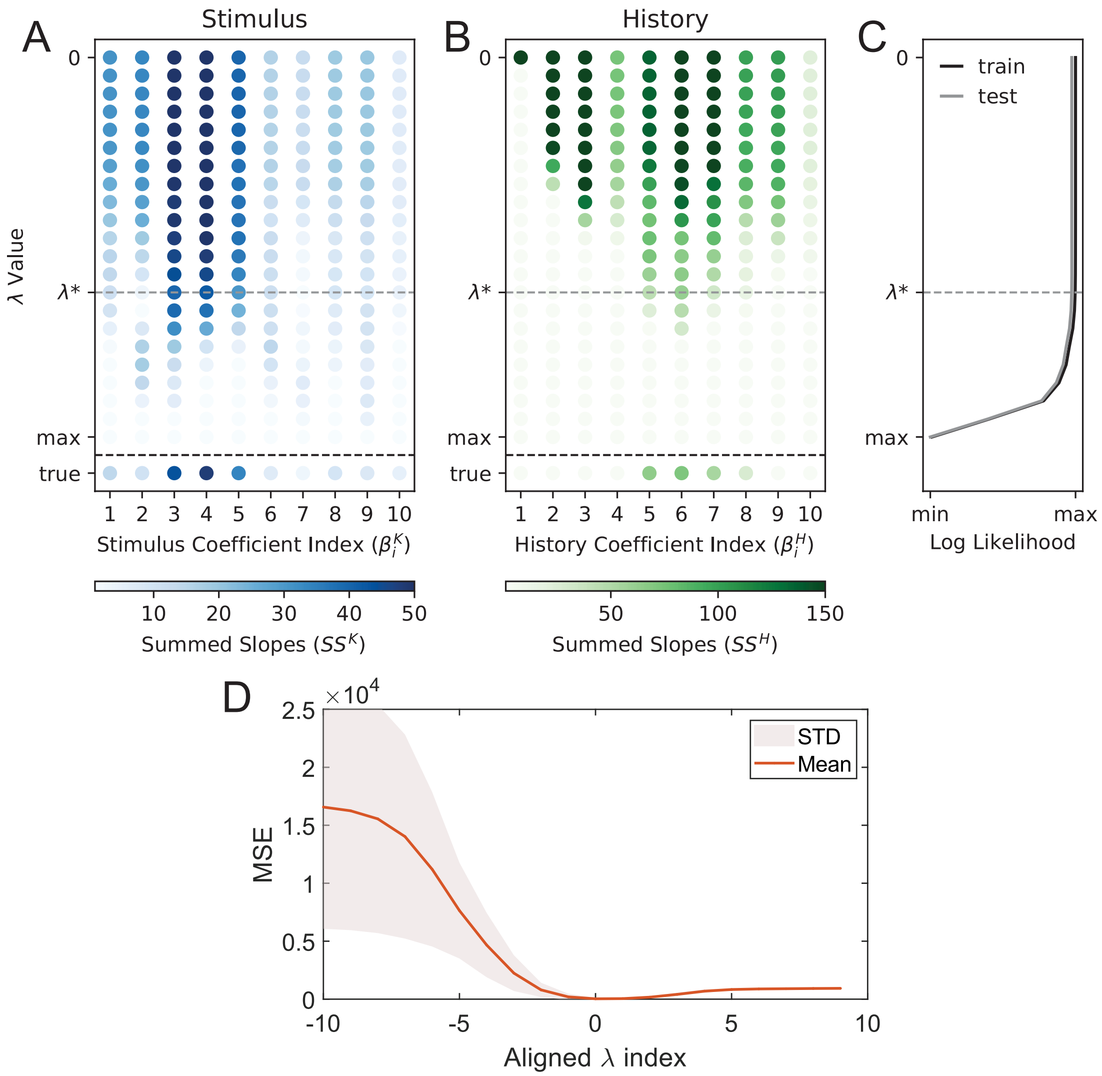


Figure B.1: Simulations verification for the joint training model 3.6. **A, B, and C** provide one example fit. **D** summarizes 100 repeated fits. SS values for PP-GLM fits of simulated spike trains for (**A**) the stimulus and (**B**) post-spike history coefficients, and (**C**) the log-likelihood all as a function of λ. True SS values are shown at the bottom of A and B. Panels are similar to Fig. 3.3K-M, except that the true model refers to a known set of PP-GLMs with coefficients $\boldsymbol{\beta}^0(g), g \in \{g_1, ..., g_B\}$. When $\lambda = \lambda^*$ (gray dashed line), the SS values are very close to the true SS values, thereby validating our trend filtering penalty hyperparameter selection method. **D** SS error between true PP-GLM and 100 separate sets of simulated spike trains from the true PP-GLM as a function of aligned λ index. Since different runs may choose different optimal tuning parameter, so the tuning parameters along the x-axis, the index λ, are aligned to the optimal λ^* at index 0.

B.2 ADMM optimization algorithm for training PP-GLMs with trend filtering

B.2.1 Update rules

Training PP-GLMs with trend filtering (Eq. 3.6) can be optimized using *alternating direction method of multipliers* (ADMM) [18, 113]. It can be rewritten as,

$$\begin{aligned}
& \min_{\boldsymbol{\beta}_{(g_1)},\dots,\boldsymbol{\beta}_{(g_B)}} && \sum_{i=1}^{B} -\ell_{(i)}(\boldsymbol{\beta}(g_i)) + \lambda\|D\boldsymbol{\beta}\|_1 && \text{(B.1)}\\
\iff \quad & \min_{\boldsymbol{\beta}_{(g_1)},\dots,\boldsymbol{\beta}_{(g_B)},\mathbf{z}} && \sum_{i=1}^{B} -\ell_{(i)}(\boldsymbol{\beta}(g_i)) + \lambda\|\mathbf{z}\|_1 && \text{(B.2)}\\
& \text{subject to} && \mathbf{z} - D\boldsymbol{\beta} = 0 && \text{(B.3)}
\end{aligned}$$

Where $\boldsymbol{\beta} = (\boldsymbol{\beta}_{(g_1)}^T, \dots, \boldsymbol{\beta}_{(g_B)}^T)^T$, $\ell_{(i)}(\boldsymbol{\beta}(g_i))$ is defined in Eq. 3.5, D represents the difference operator between blocks of $\boldsymbol{\beta}$, each block has dimension $d \times d$.

$$D = \left(\begin{array}{c:c:c:c:c}
\frac{1}{g_2-g_1} I_{d\times d} & -\frac{1}{g_2-g_1} I_{d\times d} & & & \\ \hdashline
 & \frac{1}{g_3-g_2} I_{d\times d} & -\frac{1}{g_3-g_2} I_{d\times d} & & \\ \hdashline
 & & \cdots & & \\ \hdashline
 & & & \frac{1}{g_B-g_{B-1}} I_{d\times d} & -\frac{1}{g_B-g_{B-1}} I_{d\times d}
\end{array}\right)$$

The augmented Lagrangian is,

$$\begin{aligned}
L_\rho(\boldsymbol{\beta}, \mathbf{z}, \mathbf{w}) &= \sum_{i=1}^{B} -\ell_{(i)}(\boldsymbol{\beta}(g_i)) + \lambda\|\mathbf{z}\|_1 + \frac{\rho}{2}\|\mathbf{z} - D\boldsymbol{\beta} + \mathbf{w}\|^2 - \frac{\rho}{2}\|\mathbf{w}\|^2 \\
&= \sum_{i=1}^{B} -\ell_{(i)}(\boldsymbol{\beta}(g_i)) + \lambda\|\mathbf{z}\|_1 + \frac{\rho}{2}\|\mathbf{z} - \sum_{i=1}^{B} D_{(i)}\boldsymbol{\beta}(g_i) + \mathbf{w}\|^2 - \frac{\rho}{2}\|\mathbf{w}\|^2
\end{aligned}$$

$\mathbf{w}$ is the scaled dual variable (scaled by $1/\rho$). $\boldsymbol{\beta}(g_i) \in \mathbb{R}^d$, $D \in \mathbb{R}^{(B-1)d\times Bd}$, $D_{(i)} \in \mathbb{R}^{(B-1)d\times d}$, $\mathbf{z}, \mathbf{w} \in \mathbb{R}^{(B-1)d}$. The augmented term is introduced to increase the robustness of the calculation by changing the target into a strict convex problem. Note that $\rho = 0$ is equivalent to the standard Lagrangian problem. The ADMM update rules are,

Broadcast

$$\boldsymbol{\beta}(g_i)^{(k+1)} = \operatorname*{arg\,min}_{\boldsymbol{\beta}(g_i)} \quad -\ell_i(\boldsymbol{\beta}(g_i)) + \frac{\rho}{2}\|\mathbf{z}^{(k)} - \sum_{j\in[B]\setminus\{i\}} D_{(j)}\boldsymbol{\beta}(g_j)^{(k)} - D_{(i)}\boldsymbol{\beta}(g_i) + \mathbf{w}^{(k)}\|^2, \tag{B.4}$$

for $i = 1, ..., B$.
Gather

$$\mathbf{z}^{(k+1)} = \arg\min_{\mathbf{z}} \quad \lambda\|\mathbf{z}\|_1 + \frac{\rho}{2}\|\mathbf{z} - \sum_{i=1}^{B} D_{(i)}\boldsymbol{\beta}(g_i)^{(k+1)} + \mathbf{w}^{(k)}\|^2 \tag{B.5}$$

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} + \mathbf{z}^{(k+1)} - \sum_{i=1}^{B} D_{(i)}\boldsymbol{\beta}(g_i)^{(k+1)} \tag{B.6}$$

Eq. B.4 can be calculated using Newton's method. Define the target Eq. B.4 as $R(\boldsymbol{\beta}(g_i))$, $\mu_{(i)} = \frac{1}{1+\exp\{-X_{(i)}\boldsymbol{\beta}(g_i)\}}$. The gradient and the Hessian matrix are the following,

$$\nabla R = X_{(i)}^T(\mu_{(i)} - Y_{(i)}) + \rho D_{(i)}^T \left(\sum_{j \in [B] \setminus \{i\}} D_{(j)}\boldsymbol{\beta}(g_j)^{(k)} + D_{(i)}\boldsymbol{\beta}(g_i) - \mathbf{z}^{(k)} - \mathbf{w}^{(k)} \right) \tag{B.7}$$

$$\nabla^2 R = X_{(i)}^T \text{diag}\left(\mu_{(i)} \odot (1 - \mu_{(i)})\right) X_{(i)} + \rho D_{(i)}^T D_{(i)} \tag{B.8}$$

Eq. B.5 is equivalent to,

$$\mathbf{z}^{(k+1)} = S_{\lambda/\rho}\left(\sum_{i=1}^{B} D_{(i)}\boldsymbol{\beta}(g_i)^{(k+1)} - \mathbf{w}^{(k)}\right)$$

There are other ways to update the equations above in practice. As suggested by [18, sec 3.4.5], the algorithm updates each $\boldsymbol{\beta}(g_i)$ in turn multiple times before performing the dual variable update. If $\boldsymbol{\beta}(g_i)$ are updated in parallel, and $\boldsymbol{\beta}$ and **z** are updated only once, the algorithm may diverge.

B.2.2 Stopping rules

We determine the convergence of the algorithm using primal residuals and dual residuals [18], which stem from the primal feasibility and dual feasibility.

Primal feasibility

$$\mathbf{z}^\star - D\boldsymbol{\beta}^\star = 0$$

Dual feasibility

$$0 \in \partial \sum_{i=1}^{B} -\ell_i(\boldsymbol{\beta}(g_i)^\star) - \rho D^T(\mathbf{u}^\star/\rho), \quad \mathbf{w}\star := \mathbf{u}^\star/\rho$$
$$0 \in \partial\|\mathbf{z}^\star\|_1 + \rho(\mathbf{u}^\star/\rho), \quad \mathbf{w}^\star := \mathbf{u}^\star/\rho$$

Note that we use the rescaled ADMM, **u** is the original dual variable.

Primal residual

$$\mathbf{r}^{(k+1)} := \mathbf{z}^{(k+1)} - D\boldsymbol{\beta}^{(k+1)} \tag{B.9}$$

Here $\boldsymbol{\beta}$ is a stack of $\boldsymbol{\beta}(g_i)$.

Dual residual Since $\mathbf{z}^{(k+1)}$ achieves the minimum value of Eq. B.5, so

$$\begin{aligned}0 \in& \partial\lambda\|\mathbf{z}^{(k+1)}\|_1 + \rho\left(\mathbf{z}^{(k+1)} - D\boldsymbol{\beta}^{(k+1)} + \mathbf{w}^{(k)}\right)\\ =&\partial\lambda\|\mathbf{z}^{(k+1)}\|_1 + \rho\mathbf{w}^{(k+1)}\end{aligned}$$

We can see that $\mathbf{z}^{(k+1)}$ and $\mathbf{w}^{(k+1)}$ always satisfy this part of the dual feasibility. This is also the reason why we set the learning rate as ρ.

As $\boldsymbol{\beta}^{(k+1)}$ achieves the minimum value of Eq. B.4, so

$$\begin{aligned}0 \in& \nabla_{\boldsymbol{\beta}}\sum_{i=1}^{B} -\ell_{(i)}(\boldsymbol{\beta}(g_i)^{(k+1)}) - \rho D^T\left(\mathbf{z}^{(k)} - D\boldsymbol{\beta}^{(k+1)} + \mathbf{w}^{(k)}\right)\\ =&\nabla_{\boldsymbol{\beta}}\sum_{i=1}^{B} -\ell_{(i)}(\boldsymbol{\beta}(g_i)^{(k+1)}) - \rho D^T\left(\mathbf{z}^{(k+1)} - D\boldsymbol{\beta}^{(k+1)} + \mathbf{w}^{(k)}\right) + \rho D^T\left(\mathbf{z}^{(k+1)} - \mathbf{z}^{(k)}\right)\\ =&\nabla_{\boldsymbol{\beta}}\sum_{i=1}^{B} -\ell_{(i)}(\boldsymbol{\beta}(g_i)^{(k+1)}) - \rho D^T\mathbf{w}^{(k+1)} + \rho D^T\left(\mathbf{z}^{(k+1)} - \mathbf{z}^{(k)}\right)\\ \implies \quad & \rho D^T\left(\mathbf{z}^{(k)} - \mathbf{z}^{(k+1)}\right) \in \nabla_{\boldsymbol{\beta}}\sum_{i=1}^{B} -\ell_i(\boldsymbol{\beta}(g_i)^{(k+1)}) - \rho D^T\mathbf{w}^{(k+1)}\end{aligned}$$

This means that, the following can be viewed as the dual residual.

$$\mathbf{s}^{(k+1)} := \rho D^T\left(\mathbf{z}^{(k)} - \mathbf{z}^{(k+1)}\right) \tag{B.10}$$

B.2.3 Warm start

ADMM is notorious for slow convergence, especially when λ and ρ is large. When $\lambda = \lambda_{\max}$, we know the lasso penalty term $\|D\boldsymbol{\beta}\|_1 = 0$ as it shrinks all entries toward zero. So at $\lambda = \lambda_{\max}$ we have,

$$\boldsymbol{\beta}(g_1) = ... = \boldsymbol{\beta}(g_B) = \boldsymbol{\beta}_g^{\star} \tag{B.11}$$

All blocks of $\boldsymbol{\beta}(g_i)$ are unified. And it achieves the minimum value of the target Eq. B.1.

$$\begin{aligned}\boldsymbol{\beta}_g^{\star} &= \arg\min_{\boldsymbol{\beta}_g}\sum_{i=1}^{B} -\ell_{(i)}(\boldsymbol{\beta}_g)\\ \implies \quad \frac{\partial}{\partial\boldsymbol{\beta}_g}&\sum_{i=1}^{B}\ell_{(i)}(\boldsymbol{\beta}_g) = 0\end{aligned} \tag{B.12}$$

At the optimal value, by the stationary condition of $\mathbf{w}^\star$ we also have,

$$\mathbf{z}^\star = D\boldsymbol{\beta}^\star = 0$$

Next we can derive the $\mathbf{w}^\star$ using the stationary condition of $\boldsymbol{\beta}^\star$.

$$\begin{aligned}
&\boldsymbol{\beta}_g^\star = \underset{\boldsymbol{\beta}(g_i)}{\arg\min}\ -\ell_i(\boldsymbol{\beta}(g_i)) + \frac{\rho}{2}\|\mathbf{z}^\star - \sum_{j\in[B]\setminus\{i\}} D_{(j)}\boldsymbol{\beta}_g^\star - D_{(i)}\boldsymbol{\beta}(g_i) + \mathbf{w}^\star\|^2 \\
\implies\ & 0 = \frac{\partial}{\partial\boldsymbol{\beta}(g_i)}\left(-\ell_i(\boldsymbol{\beta}(g_i)) + \frac{\rho}{2}\|\mathbf{z}^\star - \sum_{j\in[B]\setminus\{i\}} D_{(j)}\boldsymbol{\beta}_g^\star - D_{(i)}\boldsymbol{\beta}(g_i) + \mathbf{w}^\star\|^2\right)\Bigg|_{\boldsymbol{\beta}(g_i)=\boldsymbol{\beta}_g^\star} \\
\implies\ & 0 = -\frac{\partial}{\partial\boldsymbol{\beta}(g_i)}\ell_i(\boldsymbol{\beta}(g_i))\Bigg|_{\boldsymbol{\beta}(g_i)=\boldsymbol{\beta}_g^\star} - \rho D_{(i)}^T\left(\mathbf{z}^\star - D\boldsymbol{\beta}^\star + \mathbf{w}^\star\right) \\
\implies\ & 0 = -\frac{\partial}{\partial\boldsymbol{\beta}(g_i)}\ell_i(\boldsymbol{\beta}(g_i))\Bigg|_{\boldsymbol{\beta}(g_i)=\boldsymbol{\beta}_g^\star} - \rho D_{(i)}^T\mathbf{w}^\star
\end{aligned}$$

$\forall i = 1, ..., B$. Now we define,

$$\mathbf{v} = \begin{pmatrix} -\frac{\partial}{\partial\boldsymbol{\beta}(g_1)}\ell_i(\boldsymbol{\beta}(g_1)) \\ \cdots \\ -\frac{\partial}{\partial\boldsymbol{\beta}(g_B)}\ell_i(\boldsymbol{\beta}(g_B)) \end{pmatrix} = \begin{pmatrix} X_{(1)}^T(\mu_{(1)} - Y_{(1)}) \\ \cdots \\ X_{(B)}^T(\mu_{(B)} - Y_{(B)}) \end{pmatrix}$$

where $X_{(i)}, \mu_{(i)}, Y_{(i)}$ are defined the same as Eq. B.7, and the gradient of the PP-GLM log-likelihood function is calculated in the same way. From the stationary condition we know,

$$\begin{aligned}
&\rho D^T\mathbf{w}^\star = \mathbf{v} \\
\implies\ & \mathbf{w}^\star = \frac{1}{\rho}(DD^T)^{-1}D\mathbf{v}
\end{aligned}$$

We also need to consider the stationary condition of Eq. B.5. As

$$\begin{aligned}
&\mathbf{z}^\star = \underset{\mathbf{z}}{\arg\min}\ \lambda\|\mathbf{z}\|_1 + \frac{\rho}{2}\|\mathbf{z} - D\boldsymbol{\beta}^\star + \mathbf{w}^\star\|^2 \\
\implies\ & \mathbf{z}^\star = S_{\lambda/\rho}\left(D\boldsymbol{\beta}^\star - \mathbf{w}^\star\right) = 0 \\
\implies\ & \mathbf{z}^\star = S_{\lambda/\rho}\left(-\mathbf{w}^\star\right) = 0, \quad \lambda = \lambda_{\max}
\end{aligned}$$

The last equality must hold as the definition of $\lambda_{\max}$ in Eq. B.14 guarantees the zero solution. Then we use the optimal solution $\{\boldsymbol{\beta}^\star, \mathbf{z}^\star, \mathbf{w}^\star\}$ as the initial values for the ADMM when λ is large. When $\lambda = \lambda_{\max}$, it takes only one iteration to converge of course.

B.2.4 Model parameters

ρ is an optimization parameter instead of a statistical parameter. Under very general conditions, the ADMM algorithm converges to optimum for any fixed value of ρ [18]. In practice, the rate of convergence and the numerical stability can strongly depend on the choice of ρ [113]. Large ρ values impose a large penalty on violations of primal feasibility in Eq. B.4, so the algorithm favors diminishing the primal residual. Conversely, the definition of $\mathbf{s}^{(k+1)}$ in Eq. B.10 suggests that small ρ values reduce the dual residual [18]. So we adopt an adaptive strategy to balance the primal and dual residuals as the following,

$$\rho(k+1) = \begin{cases} \tau_{\text{incr}}\rho(k), & \text{if } \|\mathbf{r}^{(k)}\|_2 > \mu\|\mathbf{s}^{(k)}\|_2 \\ \frac{1}{\tau_{\text{decr}}}\rho(k), & \text{if } \|\mathbf{s}^{(k)}\|_2 > \mu\|\mathbf{r}^{(k)}\|_2 \\ \rho(k), & \text{otherwise} \end{cases}$$

Since we use a rescaled dual variable, we need to change the $\mathbf{w}$ as well to maintain the same dual variable,

$$\mathbf{w}^{(k+1)} = \begin{cases} \frac{1}{\tau_{\text{incr}}}\mathbf{w}^{(k)}, & \text{if } \|\mathbf{r}^{(k)}\|_2 > \mu\|\mathbf{s}^{(k)}\|_2 \\ \tau_{\text{decr}}\mathbf{w}^{(k)}, & \text{if } \|\mathbf{s}^{(k)}\|_2 > \mu\|\mathbf{r}^{(k)}\|_2 \\ \mathbf{w}^{(k)}, & \text{otherwise} \end{cases}$$

$\lambda_{\max}$ can be derived via KKT conditions [18, 113].

$$\begin{aligned} & 0 \in \frac{\partial}{\partial \boldsymbol{\beta}} \sum_{i=1}^{B} -\ell_{(i)}(\boldsymbol{\beta}(g_i)) + \partial\lambda\|D\boldsymbol{\beta}\|_1 \\ \iff \quad & \frac{\partial}{\partial \boldsymbol{\beta}} \sum_{i=1}^{B} \ell_{(i)}(\boldsymbol{\beta}(g_i)) = \lambda D^T\mathbf{u} \end{aligned}$$

For some $\mathbf{v}$,

$$\mathbf{u}_i \in \begin{cases} \{1\}, & \text{if } (D\boldsymbol{\beta})_i > 0 \\ \{-1\}, & \text{if } (D\boldsymbol{\beta})_i < 0 \\ [-1, 1], & \text{if } (D\boldsymbol{\beta})_i = 0 \end{cases} \tag{B.13}$$

So that we get

$$\lambda_{\max} = \|(DD^T)^{-1}D\mathbf{v}\|_\infty \tag{B.14}$$

where

$$\mathbf{v} := \frac{\partial}{\partial \boldsymbol{\beta}} \sum_{i=1}^{B} \ell_i(\boldsymbol{\beta}(g_i)\Big|_{\boldsymbol{\beta}=\boldsymbol{\beta}^\star})$$

the $\boldsymbol{\beta}^\star$ is obtained in Eq. B.12.

B.3 Stimulus reconstruction

We have presented a method that links channel conductance to specific stimulus filter or post-spike history filter features in time-domain. Next, we will provide a frequency-domain method which is an alternative analysis of how ion channel conductance affects stimulus encoding. We study the frequency properties of the spike-decoded stimulus, which is reconstructed by trained

PP-GLMs [111]. The decoded stimulus presents the information that has been encoded in the spike train. Stimulus reconstruction provides an intuitive method to investigate how ion channel conductances affect stimulus encoding by comparing the reconstructed stimulus to the actual input stimulus. The method follows the steps in [133]. The stimulus is reconstructed using the maximum a posterior (MAP) estimation of the stimulus given a fitted PP-GLM [111, 133], shown as the following,

$$\max_{\mathbf{s}} P(\mathbf{s}|y_{(i)}; \boldsymbol{\beta}(g_i), \theta) = \max_{\mathbf{s}} P(y_{(i)}|s; \boldsymbol{\beta}(g_i))P(\mathbf{s}; \theta) \tag{B.15}$$

where $\mathbf{s}$ is the vector of full stimulus; $y_{(i)}$ is the spike train for the neuron with channel conductance factor g_i; $\boldsymbol{\beta}(g_i)$ are the coefficients of the PP-GLM in Eq. 3.1; $P(y_{(i)}|\mathbf{s}; \boldsymbol{\beta}(g_i))$ is the likelihood function given in Eq. 3.5; and $P(\mathbf{s}; \theta)$ is the prior of the stimulus with parameters θ. As described in section 3.2.1, the stimulus is the white noise convolved with an alpha function as we introduced in section 3.2.1. The white noise has a Normal distribution $N(\mathbf{0}, \sigma^2 I)$. The convolution is a linear transform of the white noise, its corresponding convolution matrix is A. So the prior distribution is $P(\mathbf{s}; \theta) = N(\mathbf{0}, \sigma^2 AA^T)$. We assume the noise variance σ and the alpha function is known.

The optimization of this model is the following. The log posterior of for stimulus reconstruction (Eq. B.15) can be written as,

$$\begin{aligned}
&\log P(y_{(i)}|s; \boldsymbol{\beta}(g_i))P(s; \theta) \\
=&\sum_{j=1}^{T} y_{(i),j}(K_j s + H_j y_{(i)} + \boldsymbol{\beta}^{\text{baseline}}) \\
&- \sum_{j=1}^{T} \log(1 + \exp\{(K_j s + H_j y_{(i)} + \boldsymbol{\beta}^{\text{baseline}})\}) \\
&- \frac{1}{2} s^T (\sigma^2 AA^T)^{-1} s + C \\
=&y_{(i)}^T K s - \mathbf{1}^T \log(1 + \exp\{Ks + Hy_{(i)} + \boldsymbol{\beta}^{\text{baseline}}\}) - \frac{1}{2} s^T (\sigma^2 AA^T)^{-1} s + C
\end{aligned}$$

The above log-posterior is a convex function of s, thus the maximum a posterior (MAP) estimation can be done using Newton's method. The gradient is,

$$\frac{\partial}{\partial s} \log P(y_{(i)}|s; \boldsymbol{\beta}(g_i))P(s; \theta) = K^T y_{(i)} - K^T \sigma\big(Ks + Hy_{(i)} + \boldsymbol{\beta}^{\text{baseline}}\big) - (\sigma^2 AA^T)^{-1} s$$

The Hessian matrix is,

$$\frac{\partial^2}{\partial s \partial s^T} \log P(y_{(i)}|s; \boldsymbol{\beta}(g_i))P(s; \theta) = - K^T \text{diag}\Big(\sigma\big(Ks + Hy_{(i)} + \boldsymbol{\beta}^{\text{baseline}}\big)\Big) K - (\sigma^2 AA^T)^{-1}$$

K, H are the convolution matrices for the stimulus filter and the post-spike history filter. K_j and H_j are the j'th row of the matrices. $\boldsymbol{\beta}^{\text{baseline}}$ is the baseline, it belongs to the parameter $\boldsymbol{\beta}(g_i))$. C is the constant which is not a function of s. $\sigma(\cdot)$ is the sigmoid function. The operators $\sigma(\cdot)$, $\exp(\cdot)$ and $\log(\cdot)$ are element wise, the output has the same dimension as the input. With the gradient and the Hessian matrix, one can use gradient descent method or Newton's method to get the optimal s. The posterior is a convex function of s, thus it is guaranteed to get the globally optimal solution.

The original stimulus and reconstructed stimulus were compared using the spectrum coherence in different frequency bands. The spectrum analysis was implemented with Welch's method [103]. First, the signal was split into overlapping segments. The window length was 256 data points (256 ms width), with 32 data points overlap. Each window was masked with a Bartlett window. Second, the periodogram was calculated for each window using the discrete Fourier transform, then computed the squared magnitude of the output. All the periodograms were then averaged. Third, we estimated the magnitude-squared coherence, which is a function of frequency with values between 0 and 1, indicating how well the input signals x matched to y at each frequency. The estimator for the coherence is the following [91],

$$\hat{C}_{xy}(f) = \frac{|\hat{S}_{xy}(f)|^2}{\hat{S}_{xx}(f)\hat{S}_{yy}(f)} \tag{B.16}$$

where $\hat{S}_{xy}(f)$ is the estimated cross-spectral density between x and y, $\hat{S}_{xx}(f)$ and $\hat{S}_{yy}(f)$ are the estimated auto-spectral density. The estimated spectral densities were estimated by averaging the periodograms of all windows.

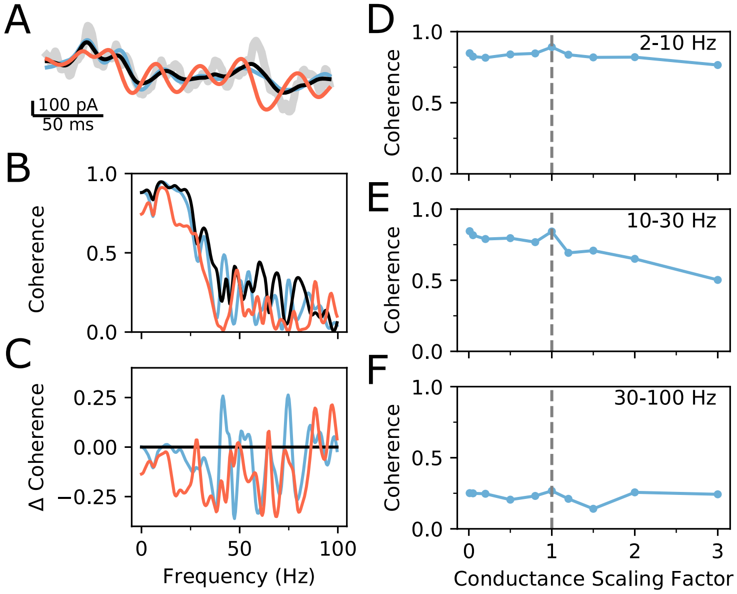


Figure B.2: Stimulus reconstructions and spectral coherence. **A** An example stimulus reconstruction for conductance scaling of 1.5, 1.0, and 0.5 (colored lines) compared to the actual stimuli (gray line) for the MC $K_A$channel. **B** Magnitude squared coherence between the stimulus reconstruction and the mean stimulus for conductance scaling of 1.5, 1.0, and 0.5. **C** The difference in coherence between the conductance scaling and control scaling of 1.0. **D-F** The mean coherence across indicated frequency bands as a function of conductance scaling factor. Gray dotted line represents control scaling factor. All panels are for the MC $K_A$channel.

Another way to examine the PP-GLM, which is an encoding model, is through decoding. Decoding is the process of estimating a reconstruction of the original stimulus given a spike train and a trained PP-GLM (Eq. B.15; Fig. B.2A). We then compare the reconstructed stimulus to the original stimulus by measuring the coherence as a function of the signal frequency (Fig. B.2B, C). We consider only stimulus reconstructions from PP-GLMs trained with optimal trend filtering penalty hyperparameter, as the reconstructed stimuli for PP-GLMs trained without trend filtering were nearly identical (data not shown). This is expected, as the goodness-of-fit is nearly identical between $\lambda = 0$ and $\lambda = \lambda^*$ (Fig. 3.5C). The coherence analysis allows estimation of specific frequency components that are, or are not, encoded when scaling different ion channel conductances (Fig. B.2B). Here we evaluate how ion channel conductance scaling affects the coherence between the reconstructed stimulus and the original stimulus, by measuring the difference between scaled ion channel conductances and the control ion channel conductance (Fig. B.2C). For example, when scaling the MC $\mathrm{K_A}$channel, increasing $\mathrm{K_A}$channel conductance generally reduces coherence across the frequency spectrum, whereas decreasing MC $\mathrm{K_A}$channel conductance shows increased coherence at specific frequencies 35-50 Hz and 70 Hz (Fig. B.2C). Generally, the coherence measures are fairly noisy, which we can smooth by averaging over well characterized frequency bands (Fig. B.2D-F). The MC $\mathrm{K_A}$channel conductance scaling affects encoding of mid range, beta frequencies (Fig. B.2D-F), with only moderate effects on low range, theta frequencies and high range, gamma frequencies. This suggests a prominent role for the MC $\mathrm{K_A}$channel in encoding of mid range, beta frequencies. Overall, the additional approach of examining stimulus reconstructions further reveals how different ion channel conductance scaling affects encoding of specific stimulus features.

Appendix C

Appendix for Chapter 4

We provide supplementary information and analyses in the form of appendices, figures, and tables.

The details of the drifting gratings conditions used in our analysis, such as temporal frequency and orientation, are summarized in C.1. Details about the simulation study are in C.1, C.3, C.4, and C.5. Supplementary analyses about curve fitting are in C.2, additional material about the priors is in C.3, and essential formulas for partial correlation and regression are in C.4. Selected regression results are in C.2.

The remainder of our supplementary material is in a series of figures. The complete set of correlations, that is posterior distributions of the correlations, are in C.3 with corresponding full partial correlations in C.4 (full here means after conditioning on all the rest of the features). The plot of marginal and partial correlation similar to Fig 3B,D for the case of V1 and LM conditioning on AL is shown in C.5. The highly concentrated posterior distributions of p_{ac} are displayed in C.6. The complete set of estimated templates, similar to the plots for selected conditions, are in Fig 4.5. Different plots for the same estimated templates are in C.8, while C.9 shows additional comparisons of the templates to the corresponding PSTHs obtained from the same selected set of neurons. The locations of recorded neurons, together with their classification (based on the posterior median), are shown in C.10. Goodness-of-fit plots for each trial are in C.12.

C.1 Simulation study

The true model in the simulations together with the spiking data were generated according to the process in Eq (4.1)-(4.8). The matrix Σ^{pop} and the average population activities $f_{a,c}^{\text{pop}}$, $f_{a,c}^{\text{local-1}}$, $f_{a,c}^{\text{local-2}}$ came from the fit to the real data. We included 3 features (Gain, Peak-1, Peak-2) for 3 virtual brain areas. The total number of trials was 180. For one simulation dataset, we totally drew 2000 MCMC samples, and dropped first 500 samples. The parameters were initialized with the true values. We totally created 500 repetitions which could make the CI coverage have a standard error around $\sqrt{0.95 \times 0.05/500} \approx 0.01$ with 95% CI. We calculate the CI ends standard error by following [40, sec. 3.2.1]. A loose lower bound of RMSE can be determined by the standard error of Pearson's correlation with the same sample size, while assuming there is no bias and the true feature values are known. The standard error monotonically decreases as correlation absolute value grows. The standard error of Pearson's correlation is between 0.012 (if correlation is 0.9) and 0.072 (if correlation is 0) [45]. The RMSE range of the simulation is slightly larger than this range due to extra uncertainty from other parts of the model and the bias of the estimation.

Our model assumes the neurons within each group have the same property, meaning they share the same intensity functions $f^{\text{pop}}_{a,c}$, $f^{\text{local-1}}_{a,c}$, $f^{\text{local-2}}_{a,c}$. To verify this assumption, we created two extra simulation scenarios by adding neuron-to-neuron variance to the intensity functions. We did not find systematic error in the curve fitting on the real data, so the injected noise was not specified in specific ways. Instead, we leveraged the generative model and created the noise learned from the data. In one case, we added mild neuron-to-neuron variance. We first collected posterior samples of f^{pop}, $f^{\text{local-1}}$, $f^{\text{local-2}}$. Then we randomly assigned a sample to each neuron with respect to its group. In the other case, we added more aggressive neuron-to-neuron variance. We approximated the conditional posterior distribution (Eq (4.10) and (4.11)) of the intensity function coefficients using Laplacian method when the sampling was mixed. This output a multivariate Normal distribution, where the mean was the MAP, and the covariance matrix was the inverse of the Hessian. Next we amplified the covariance matrix by 20 times, and drew samples for each neuron. We replicated the evaluations for these two extra experiments, and the results are similar.

For the mild noise case, the bias values of the estimated correlations are in the range $[-0.039, 0.05]$ (the mean of all pairs of features is 0.0025). The correlation posterior CI coverage is shown in C.4. The RMSE values are in range $[0.025, 0.091]$ (the mean of all pairs of features is 0.074). The simulation standard errors of CI end points were small relative to the RMSE values. The range is in $[0.001, 0.010]$ (the mean of all pairs of features is 0.0057). For the aggressive noise case, the bias values of the estimated correlations are in the range $[-0.033, 0.047]$ (the mean of all pairs of features is 0.001). The correlation posterior CI coverage is shown in C.5. The RMSE values are in range $[0.024, 0.088]$ (the mean of all pairs of features is 0.072). The simulation standard errors of CI end points were small relative to the RMSE values. The range is in $[0.001, 0.0092]$ (the mean of all pairs of features is 0.0056).

C.2 Curve fitting

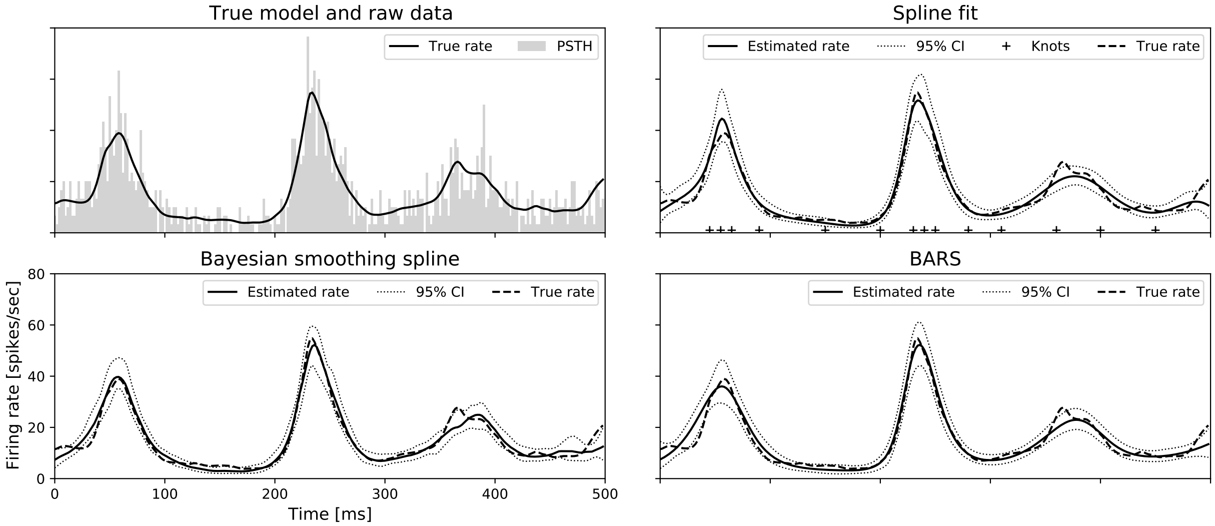


Figure C.1: **A comparison between different curve fitting methods.** The raw data has 150 trials. The knots positions of spline fitting are shown at the bottom of the figure. Our method, the Bayesian smoothing spline, chooses the tuning parameter using 5-fold cross-validation. The BARS uses the default parameters.

The above figure shows a simulation example with two peaks similar to the "pop" activity. We compare our method with two other methods, BARS and spline fitting with manually selected knots. Our method has has 100 knots and 102 bases. The spline fit model has 16 knots, 18 bases. BARS does not have a fixed number of knots. We use the default parameters in the online package [82]. The estimated curves are all very close the true model, and the pointwise CI bands trap the true curve very well. The synthesized data has 150 trials. The CI band of the spline fitting (top right plot) is calculated from the Fisher information. The positions of the knots are shown at the bottom. The tuning parameter of our method (bottom left plot) comes from 5-fold cross-validation [63, sec. 7.10.1]. The spline fit model has 18 free parameters, so the number of degrees of freedom is 18. The BARS MCMC samples on average have 10.3 knots, so the average number of degrees of freedom is 12.3. For the Bayesian smoothing spline method, the number of degrees of freedom is 6.6. Even the number of knots is large, but the effective number of parameters is actually very small. The number of degrees of freedom is obtained by maximizing a posterior and approximating the problem as least squares (in the Reinsch form) [36].

The method needs the hyperparameter η to control the smoothness of the curves. In the real data analysis, we first selected a good initial tuning parameter (the tuning procedure can be run in a few iterations to have a good initial guess, or use simulated data), fitted the model and fixed other parameters except for the curve fitting coefficients. Then we trained the curve fitting by maximizing a posterior (MAP), which is equivalent to penalized maximizing the likelihood or the Tikhonov regression. We selected the tuning parameter using 5-fold cross-validation by splitting the trials into 5 segments in all conditions [63, sec. 10.7.1] while holding the neuron clustering when the algorithm is mixed. We also tested the tuning parameter on simulated data with two-peak patterns, the result was the same. Since the number of neurons or trials used to estimate the curve varies, we multiply η with the number of trials in Eq (C.5), and (C.6) (assuming the trial length is fixed, otherwise the trial length can also be a factor), so that the tuning parameter selection is invariant of or less sensitive to sample size.

C.3 Priors

We first list some properties of inverse-Wishart distribution, which will later be used for designing the prior for Σ^{pop}. The following properties can be derived from the conclusions in [9, 68]. Consider the following distribution for the $D \times D$ covariance matrix Σ. The scale matrix Ψ is diagonal with all positive values. The number of the degrees of freedom is ν.

$$\Sigma \sim \text{inverse-Wishart}(\Psi, \nu) \tag{C.1}$$

The marginal distribution of any correlation entry is the following,

$$\begin{aligned} \rho &= \frac{\Sigma_{i,j}}{\sqrt{\Sigma_{i,j}\Sigma_{j,j}}}, \quad i \neq j \\ p(\rho) &= \frac{1}{2}\text{Beta}\left(\frac{\rho+1}{2}\middle|\frac{\nu-D+1}{2}, \frac{\nu-D+1}{2}\right), \quad \rho \in [-1,1] \end{aligned} \tag{C.2}$$

If $\nu = D+1$, the marginal distribution is uniform. Σ has mode $\Psi/4$, the mean does not exit.

For any partial correlation entry,

$$\begin{aligned} \rho' &= -\frac{\Omega_{i,j}}{\sqrt{\Omega_{i,j}\Omega_{j,j}}}, \quad \Omega = \Sigma^{-1}, \quad i \neq j \\ \rho' &\sim \frac{1}{2}\text{Beta}\left(\frac{\rho'+1}{2}\middle|\frac{\nu-1}{2}, \frac{\nu-1}{2}\right), \quad \rho' \in [-1,1] \end{aligned} \tag{C.3}$$

If $\nu = D+1$, the marginal distribution is a zero-concentrated distribution. As the dimension D increases, it becomes more concentrated. Marginal and partial correlation properties above do not depend on the scale matrix as long as all diagonal elements are positive.

Next, consider the marginal distribution of the diagonal element $\Sigma_{i,i}$,

$$\Sigma_{i,i} \sim \text{Inverse-Gamma}\left(\frac{\nu-D+1}{2}, \frac{\Psi_{i,i}}{2}\right) \tag{C.4}$$

The inverse-Gamma distribution is diffuse enough to capture the uncertainty using its long thick tail. If select $\nu = D+1$, the marginal distribution does not depend on D. Some previous works, such as [2], add extra uncertainty for $\Psi_{i,i}$ in the prior, but we think it is not necessary.

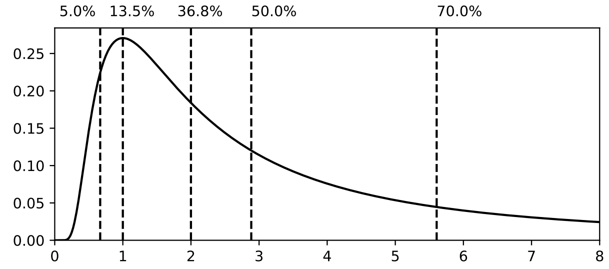


Figure C.2: **An example of variance $\Sigma_{i,i}$ marginal distribution.** $\Psi_{i,i} = 4$. The mode is 1, the corresponding quantile percentage is 13.5%. The median is at 2.89. The quantile at 2 corresponds to 36.8%, which stays between the mode and the median. The 70.0% quantile is 5.61 times of the mode. Note that the ratio between these quantiles is invariant of Ψ and D.

In Eq (4.8), $\Psi_0 = 2\Phi_0$ is a diagonal scale matrix. The number of the degrees of freedom is $\nu_0 = D + 1$, where $D = d \cdot A$ is the dimension of the matrix Σ^{pop} (Eq (4.7)). Based on the properties above, such a design leads to some desired properties: 1, as detailed prior knowledge for the correlation between features is limited, we use uninformative prior for correlations [9]. The marginal distribution of a correlation derived from Σ^{pop} is uniform on $[-1, 1]$ (see Eq (C.2)); 2, the marginal distribution of partial correlations derived from Σ^{pop} concentrate at 0. More specifically, it is the scaled Beta distribution on $[-1, 1]$ with coefficients $(D/2, D/2)$ (see Eq (C.3)). As the dimension D grows, it concentrates more at the center. Conditioning on more relevant or correlated features is more likely to make the partial correlation closer to zero; 3, the marginal variance of a feature i is diffused with inverse-Gamma$(1, [\Phi_0]_{ii})$ (see Eq (C.4)). Φ_0 is a diagonal covariance matrix with the standard deviations being the ranges of the features. The ranges of Peak-1, Peak-2 and Gain are roughly estimated using a different animal. The standard deviations for the Gain, Peak-1 shifting and Peak-2 shifting are 0.15 log spikes/sec, 10 ms, 30 ms. We set $\Psi_0 = 2\Phi_0$ such that $[\Phi_0]_{i,i}$ is on the long tail in the distribution in Eq (C.4) and it stays between the mode and the median (also see the above figure). This can make the prior of the feature variance less informative.

The prior for the neurons' memberships uses Dirichlet distribution (4.9) for its simplicity. The hyperparameter is $\alpha = 5 \cdot \mathbf{1}$ as an uninformative prior which is relatively small comparing to the number of neurons. The membership assignment is insensitive to α in a certain range. We checked the results with $\alpha = \mathbf{1}$ (flat prior) and $\alpha = 10 \cdot \mathbf{1}$. Then, we quantified the distinction using the difference ratio of memberships' modes of all neurons in all conditions. The total count is $N \cdot C$. The model with $\alpha = 5 \cdot \mathbf{1}$ and the model with $\alpha = \mathbf{1}$ have 2.1% membership difference. The model with $\alpha = 5 \cdot \mathbf{1}$ and the model with $\alpha = 10 \cdot \mathbf{1}$ have 1.4% membership difference. In addition, the prior with $\alpha = 5$ plays a role of regularization, which prevents the model from an empty clusters before the samples get mixed, while $\alpha = 1$ does not do so.

The priors related to curve fitting are the follows,

$$\beta^{\text{pop}}_{a,c} \mid \eta, \Omega \sim \frac{1}{\tilde{Z}_0} \exp\{-\eta N_{\text{pop},a,c} R_c (\beta^{\text{pop}}_{a,c})^T \Omega \beta^{\text{pop}}_{a,c}\} \tag{C.5}$$

$$\beta^{\text{local-1}}_{a,c} \mid \eta, \Omega \sim \frac{1}{\tilde{Z}_1} \exp\{-\eta N_{\text{local-1},a,c} R_c (\beta^{\text{local-1}}_{a,c})^T \Omega \beta^{\text{local-1}}_{a,c}\} \tag{C.6}$$

$$p(\beta^{\text{local-2}}_{a,c}) \propto 1 \tag{C.7}$$

$\tilde{Z}_0, \tilde{Z}_1$ are the normalizers of the distributions. The constraint of the coefficients is designed for smoothness in the same spirit as the smoothing spline [63, sec. 5.4]. In Gibbs sampling, the penalty becomes the log prior density, where the prior is normal. The smoothing parameter η is tuned using cross-validation. Details are discussed in C.2. $N_{\text{pop},a,c}$ and $N_{\text{local-1},a,c}$ count the number of neurons in group "pop" and "local-1" respectively. R_c is the number of trials of a condition. Multiplying with N and R makes η less sensitive to varying sample size.

C.4 Properties of multivariate Normal distribution

We derive the marginal correlation, partial correlation and R^2 values directly from estimated covariance matrix. Let X be a random vector following multivariate Normal distribution $X \sim N(\boldsymbol{\mu}, \Sigma)$. Σ is the covariance matrix. The marginal correlation between two entries X_i, X_j is,

$$\rho(X_i, X_j) = \frac{\Sigma_{i,j}}{\sqrt{\Sigma_{i,i} \cdot \Sigma_{j,j}}}$$

The partial correlation is derived from the marginal and conditional distribution. Let us consider the conditional distribution of $X_W|X_Z$, where X_W, X_Z are entries of X. $A = W \cup Z$ is the index set of those entries. Then,

$$(X_W, X_Z)^T \sim N(\mu_A, \Sigma_{A,A})$$

The conditional distribution follows,

$$X_W|X_Z \sim N(\mu_{W|Z}, \Sigma_{W|Z})$$

$\mu_{W|Z} = \mu_W + \Sigma_{W,Z}\Sigma^{-1}_{Z,Z}(X_Z - \mu_Z)$, $\Sigma_{W|Z} = \Sigma_{W,W} - \Sigma_{W,Z}\Sigma^{-1}_{Z,Z}\Sigma_{Z,W}$. Then we can get the partial correlation using the covariance matrix $\Sigma_{W|Z}$. Now set $W = \{i, j\}$. $\Sigma' = \Sigma_{W|Z}$.

$$\rho(X_i, X_j|X_Z) = \frac{\Sigma'_{i,j}}{\sqrt{\Sigma'_{i,i} \cdot \Sigma'_{j,j}}}$$

The regression analysis is also based on the conditional distribution. Let $y = X_W$ be the independent variable, W has a single index. The conditional relation can be rewritten as,

$$y = X_Z\beta + b + \varepsilon$$

where $\beta = \Sigma^{-1}_{Z,Z}\Sigma_{Z,W}$, $b = \mu_W - \Sigma_{W,Z}\Sigma^{-1}_{Z,Z}\mu_Z$, $\varepsilon \sim N(\mathbf{0}, \Sigma_{W|Z})$. So we can derive

$$R^2 = 1 - \frac{\Sigma_{W|Z}}{\Sigma_{W,W}}$$

$\Sigma_{W,W}$ is the variance of the independent variable, $\Sigma_{W|Z}$ is the variance of the regression residual.

C.5 Supplementary figures and tables

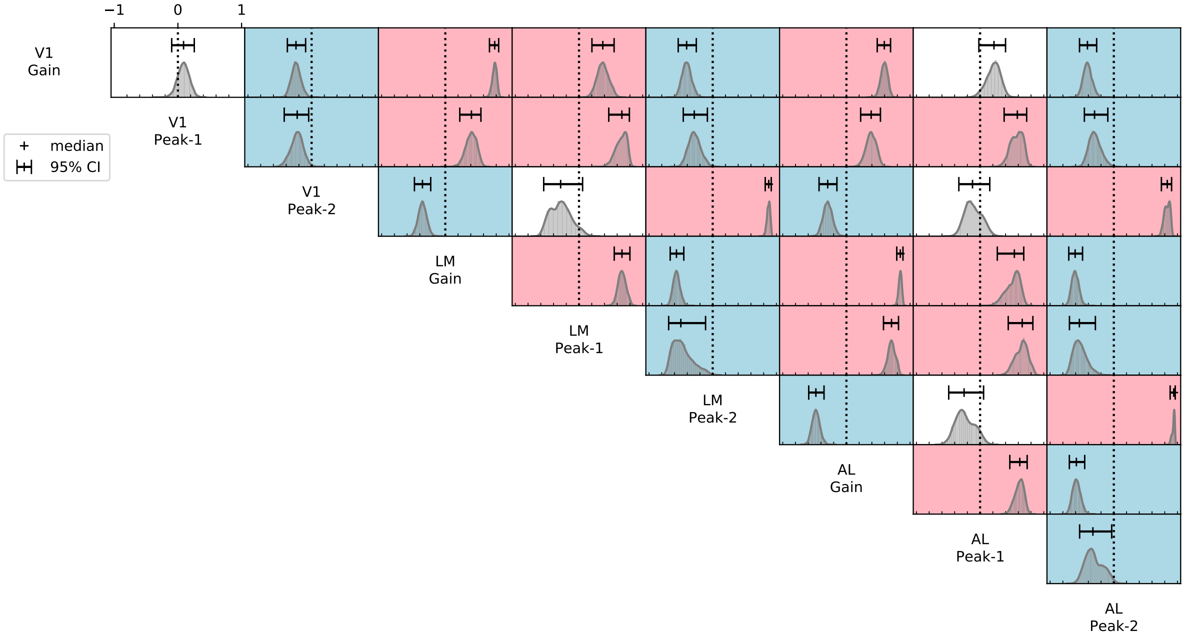


Figure C.3: **Full marginal correlations** Pairwise marginal correlations between all features. Significantly positive values are labeled by red, and significantly negative ones are labeled by blue.

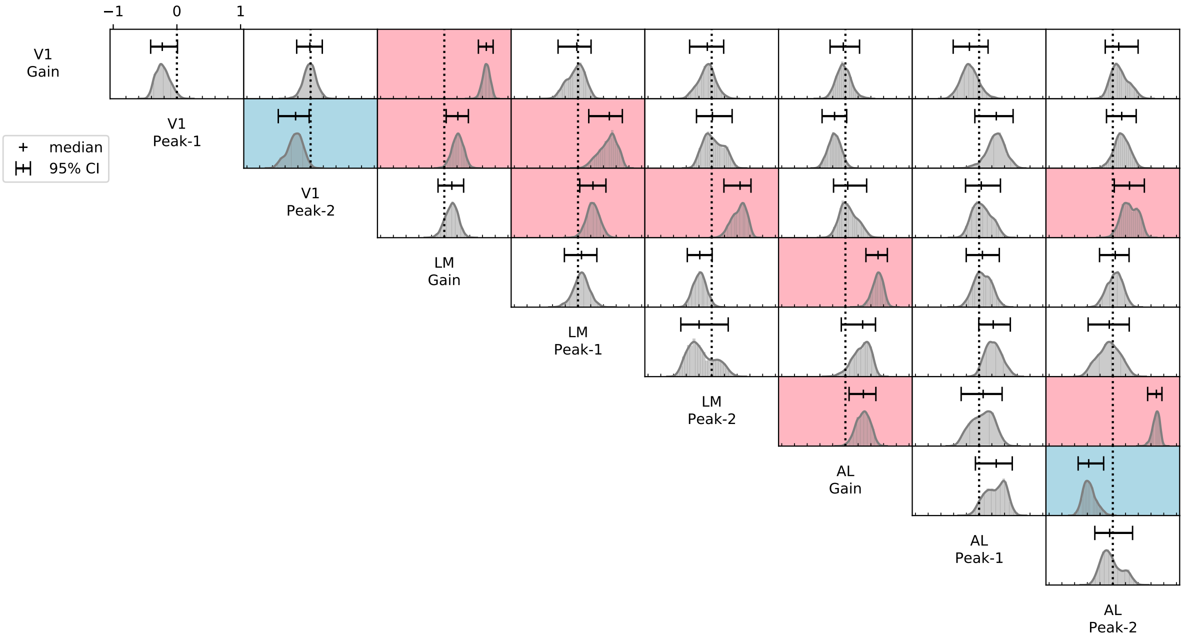


Figure C.4: **Full partial correlations** Each entry shows the correlation between two features conditioning on all the rest. Considering all combinations of variables there are 4572 partial correlations among the 9 features. Here we display the 45 correlations together with the 45 corresponding full conditional partial correlations and we also provide several additional partial correlations. Significantly positive values are labeled by red, and significantly negative ones are labeled by blue.

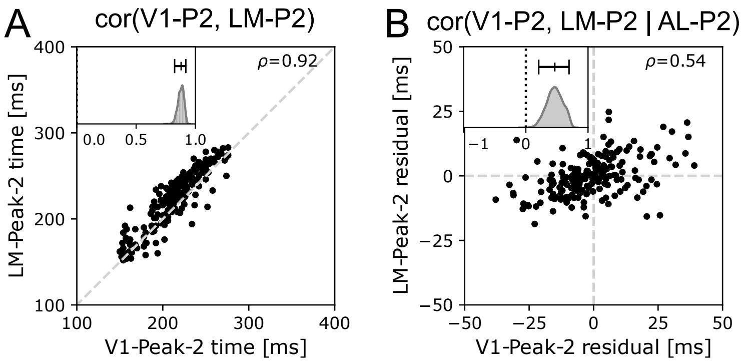


Figure C.5: **Marginal correlation and partial correlation of Peak-2** This figure is similar to Fig 4.3. **A** shows estimated marginal correlation. Each dot represents the estimated time of peak-2 on a given trial. **B** shows estimated partial correlation. Each dot represents the residual from a regression on the conditioning variable (the correlation of these residuals being the partial correlation given the conditioning variable). The embedded plots in the corner are the posterior distribution of the correlations. There is no dramatic reduction of the correlation of V1 and LM after conditioning on AL

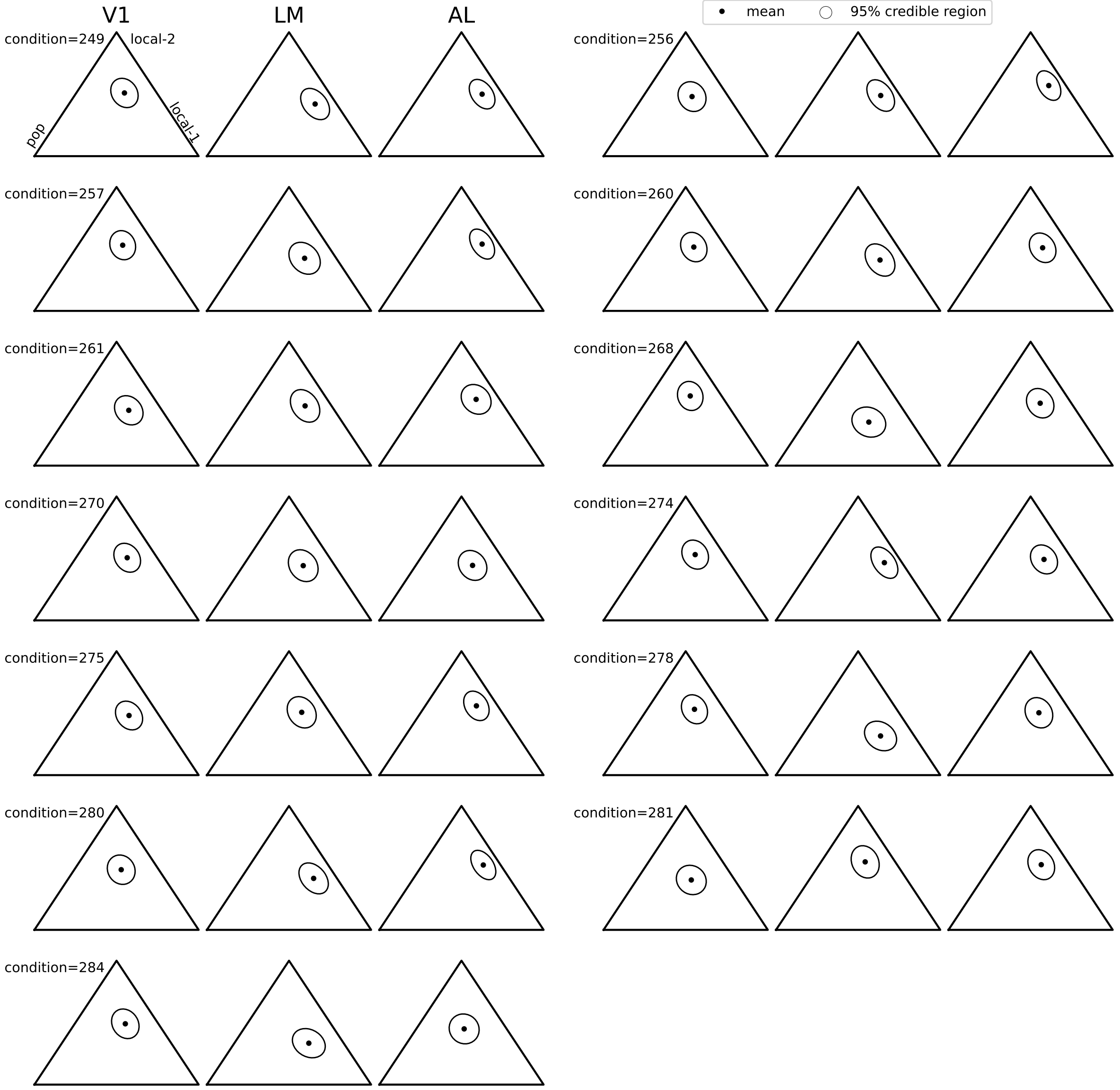


Figure C.6: **The posterior distributions of $p_{a,c}$.** Each condition has a group of 3 triangles for V1, LM and AL. $p_{a,c}$ is a 3-entry vector showing the probability of the subgroup memberships. So $p_{a,c}$ are in 2-simplex and they are mapped to the triangles. The left angle represents the "pop", the right one is the local-1, and the top one is "local-2". If a point is closer to an angle meaning the corresponding component has a larger portion. To simplify the visualization, we approximate the posterior using bivariate Normal distribution. The dot is the mean, and the ellipse is the 95% credible region of the Normal distribution. Most distributions are far from "pop" corner (left angle), which means "pop" takes small portion of neurons. The distributions are close to "local-2" corner (top angle), so "local-2" include a large part of neuron. All the distributions are highly centralized, meaning the uncertainty for neuron clustering is very small. The portions slightly vary from condition to condition, which shows the diversity of neurons' behavior.

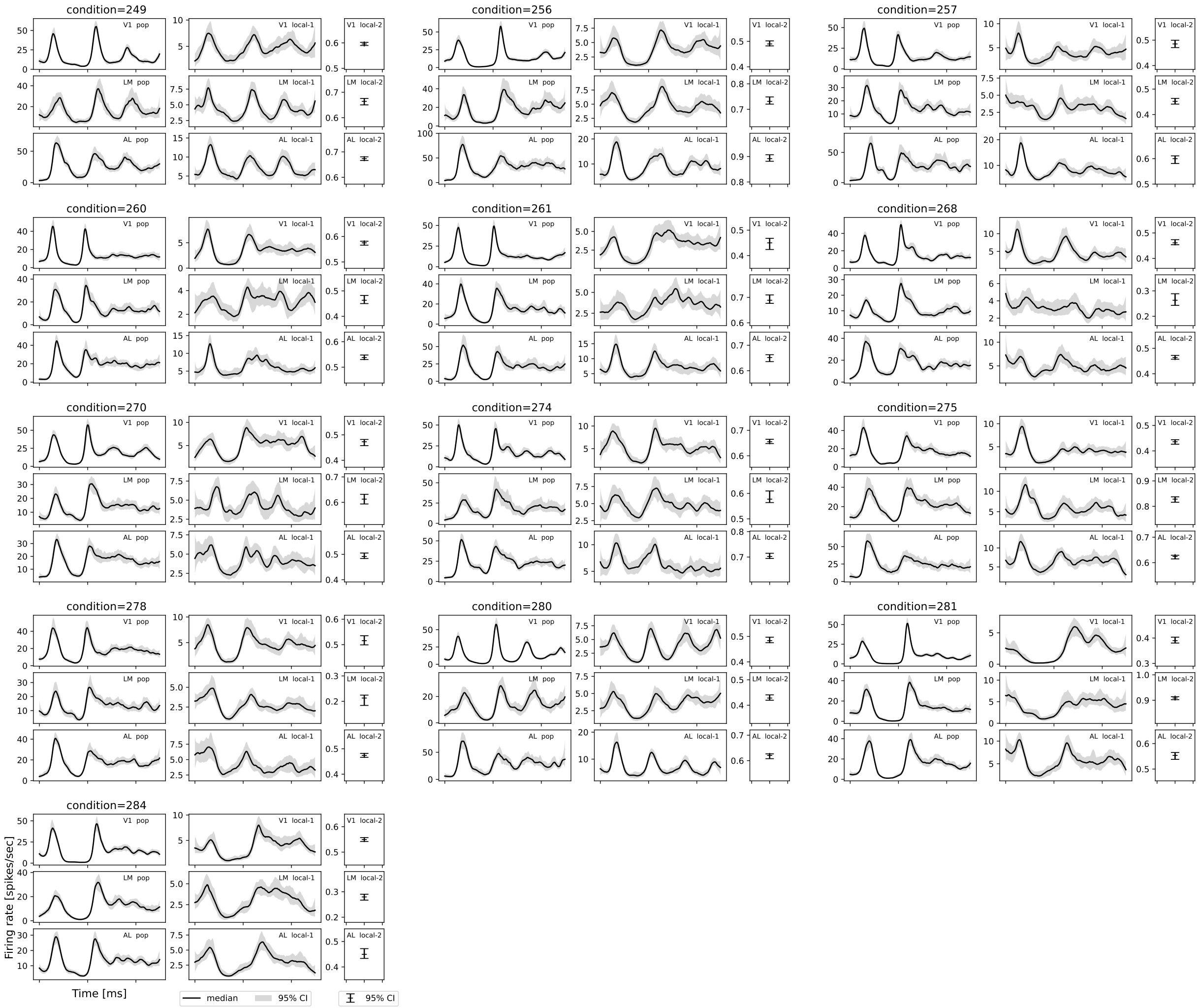


Figure C.7: **Population templates in all conditions.** Similar to Fig 4.5, the figure shows the fitted population templates with the median and the pointwise 95% CI of the posterior. The figure is composed of 3×3 blocks for 13 conditions. In a condition block, the columns are $\exp\{f_{a,c}^{\text{pop}}\}$, $\exp\{f_{a,c}^{\text{local-1}}\}$, and $\exp\{f_{a,c}^{\text{local-2}}\}$. The rows are V1, LM, and AL.

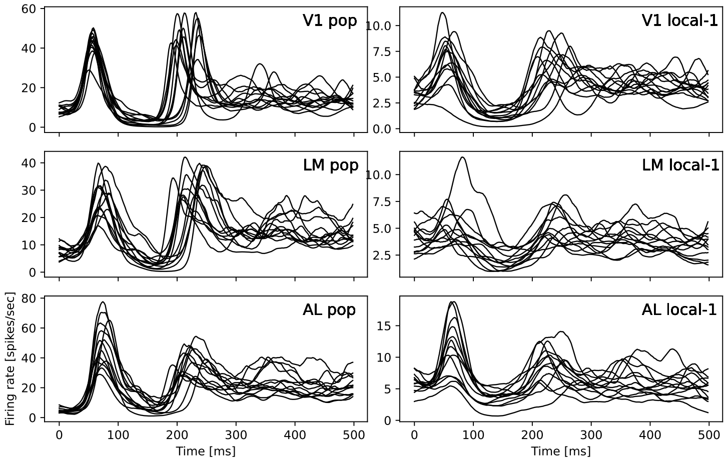


Figure C.8: **Population templates in all conditions.** The curves are the same as those in C.7, which are the medians of the $\exp\{f^{\text{pop}}_{a,c}\}$ and $\exp\{f^{\text{local-1}}_{a,c}\}$. Each curve represents one condition. We overlap the curve across conditions to demonstrate the condition-to-condition variance, which is equivalent or larger than the feature variance, such as Peak-1 or Peak-2 shifting. This suggests disaggregating the data to better capture the subtly varied features.

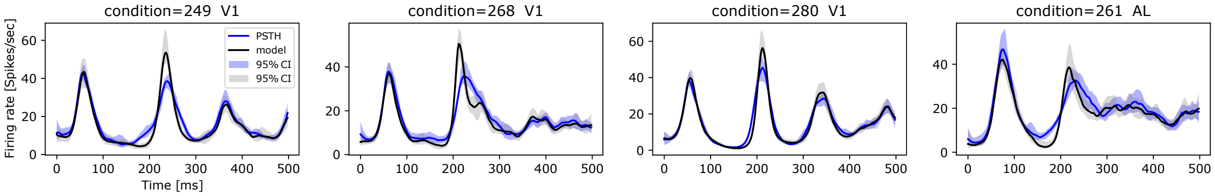


Figure C.9: **Examples of** $\exp\{f^{\mathbf{pop}}_{a,c}\}$ **versus PSTH.** The PSTH and the pointwise CI are estimated using Bayesian smoothing spline similar to Algorithm 1 line 5 without incorporating peaks shifts. The PSTH curves fit the same set of neurons as in the "pop" group (selected from the mode of the posterior). In many conditions, the Peak-2 of $\exp\{f^{\text{pop}}_{a,c}\}$ becomes narrower and higher than the PSTH. The CI bands around the Peak-2 hardly overlap, for example `stimulus_condition_id = 249, 268, 280`. This is because the activity trials are aligned better. However, in some conditions like `261` AL, the Peak-2 of the model is not significantly higher than the PSTH. In most conditions, the Peak-1 of $\exp\{f^{\text{pop}}_{a,c}\}$ does not change too much because the trial-to-trial deviations are small. Another observation is that the variance of the Peak-2 shapes is larger than that of Peak-2. This explains why the Peak-2 in Fig 4.1, the overall average, is wider and lower than Peak-2 due to larger diffusion together with condition-to-condition variance. This also suggests that simply averaging neural activities, even with more trials, may not get more accurate neural responses without considering the time-shifting deviations or without treating different conditions separately.

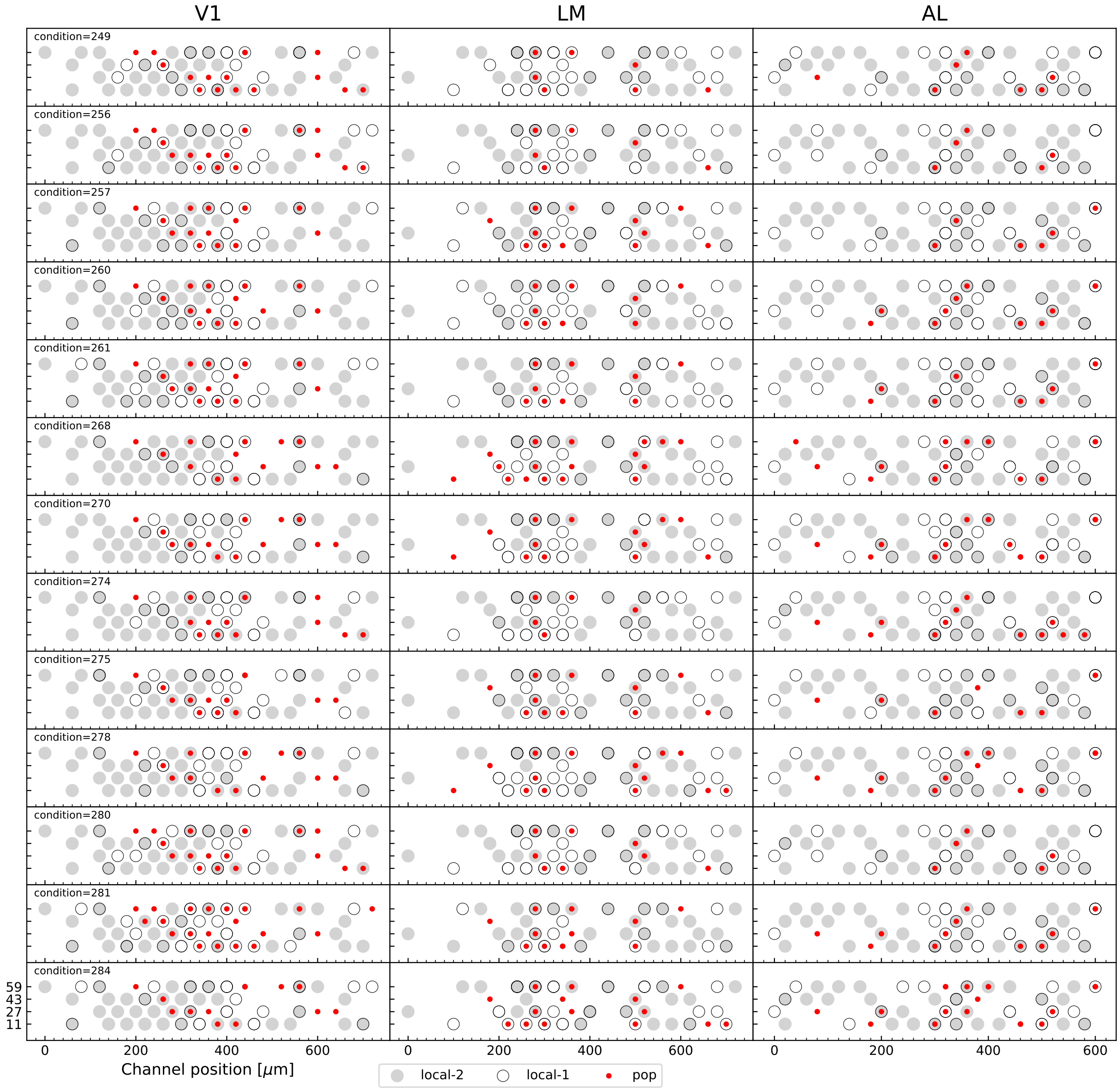
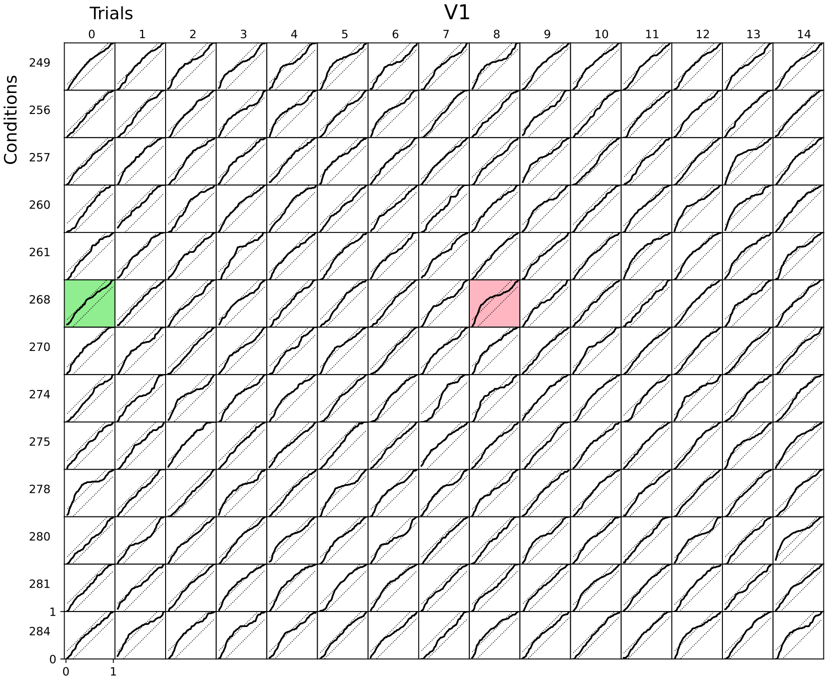
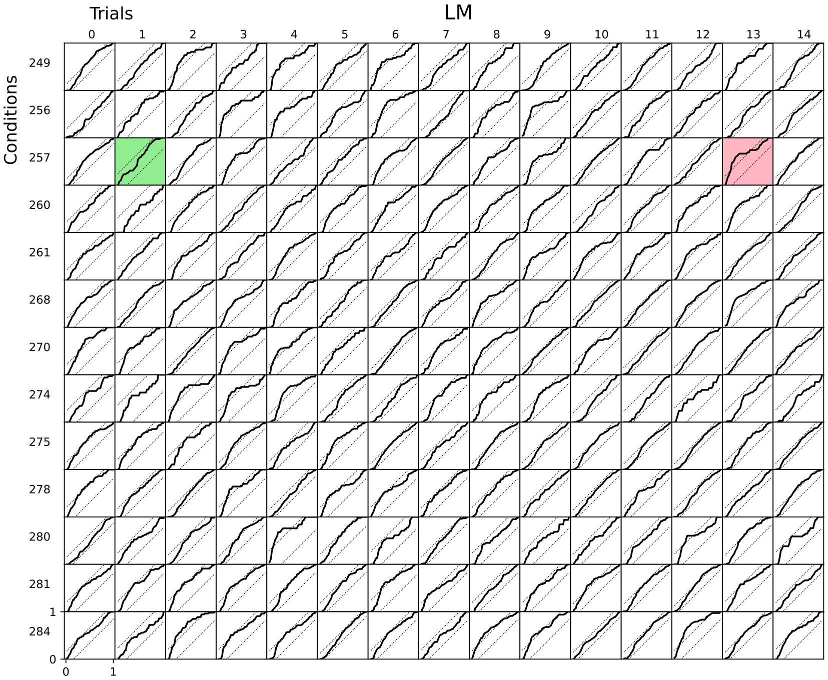


Figure C.10: **The location of recorded neurons.** The figure shows the memberships and the locations of the neurons in every area (by column) and every condition (by row). This is an expansion of Fig 4.7. The membership is determined by the posterior median. The dot shape or color represent its membership. More precisely, the location of the neuron is the location of the channel that records the neuron signal. One channel can record the signal from more than one neuron.


Trials
V1
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
Conditions
249
256
257
260
261
268
270
274
275
278
280
281
284
1
0
0
1


Trials
LM
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
Conditions
249
256
257
260
261
268
270
274
275
278
280
281
284
1
0
0
1

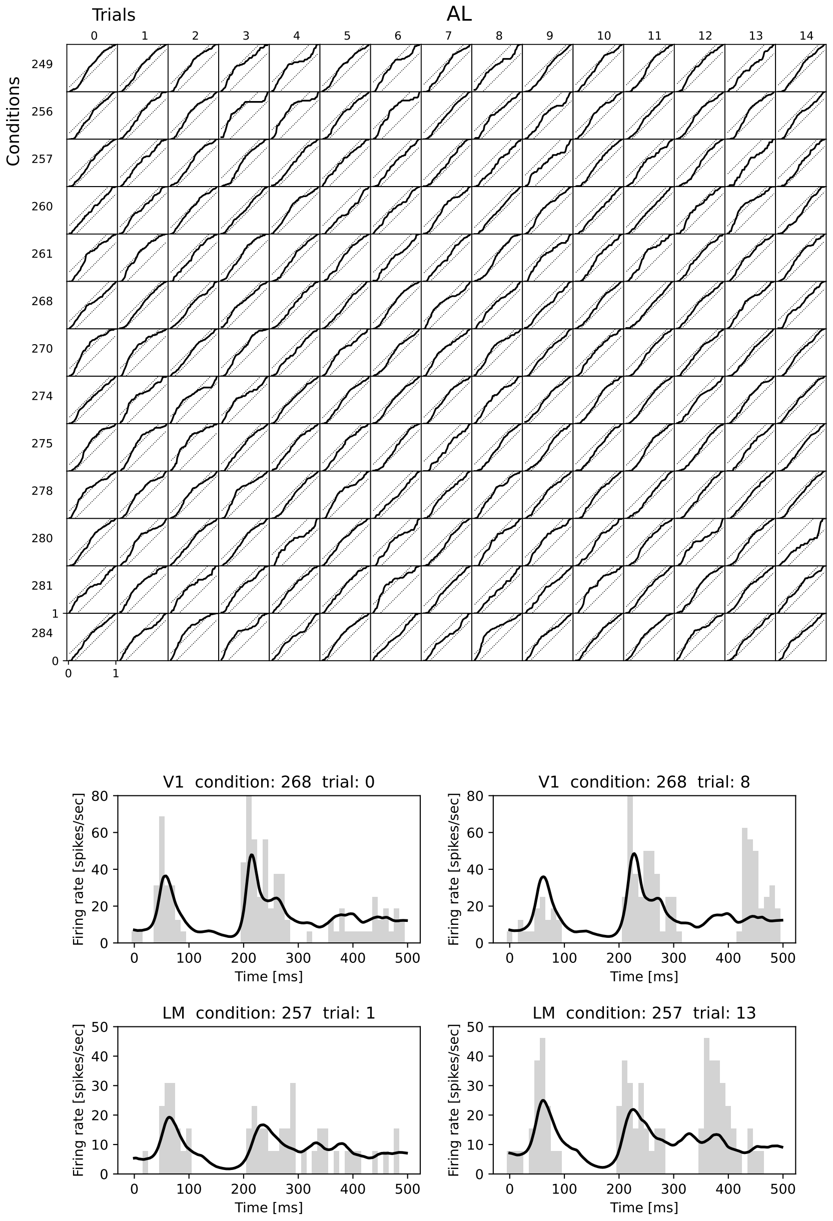


Figure C.12: **Goodness-of-fit test.** KS test for “pop” group of all three regions (shown by checker boards), all conditions (shown by rows), and all trials (shown by columns). The dotted lines are 99% CI of KS test. At the bottom of the figure, we show some examples of good fits (highlighted by green in the grid) and bad fits (highlighted by red in the grid). Some trials may have unexpected activities that are different from the templates ($f_{a,c}^{\text{pop}}$), but our method is still able to find Peak-1 and Peak-2 positions. Since we model the activity by matching the template, it may loss some details of each trial and fail some goodness-of-fit tests, but it can accurately capture the main features of interest. For example, the plot of V1 condition 268 trial 8 has a bump at the end of the trial, which is not common in other trials of the same condition. The example of V1 268 trial 0 is a representative trial of the condition, which matches the template very well.

Condition id	249	256	257	260	261	268	270	274	275	278	280	281	284
Temporal frequency [Hz]	8	15	8	4	8	4	8	4	8	8	8	15	15
Orientation [deg]	90	270	315	315	135	45	45	90	0	225	270	315	45
Spatial frequency [cycles/deg]	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
Contrast [%]	80	80	80	80	80	80	80	80	80	80	80	80	80

Table C.1: **Stimulus conditions** Parameters for 13 trials in session `798911424` used for our analysis. The contrast and the spatial frequency are the same for all conditions.

Model	Y	X	R^2
A0	AL-P2	V1-P2	0.71 (0.60,0.82)
A1	AL-P2	V1-P1,LM-P1,AL-P1	0.34 (0.14,0.53)
A2	AL-P2	V1-P1,LM-P1,AL-P1,V1-P2	0.85 (0.75,0.89)
A3	AL-P2	V1-P1,LM-P1,AL-P1,LM-P2	0.92 (0.85,0.95)
A4	AL-P2	V1-P1,LM-P1,AL-P1,V1-P2, LM-P2	0.92 (0.85,0.95)

Difference	R^2
A2 - A0	0.12 (0.06,0.19)
A2 - A1	0.50 (0.35,0.64)
A3 - A1	0.58 (0.41,0.72)
A3 - A2	0.07 (0.04,0.14)
A4 - A2	0.08 (0.05,0.14)
A4 - A3	0.01 (0.00,0.01)

Model	Y	X	R^2
B0	LM-P2	V1-P2	0.79 (0.72,0.84)
B1	LM-P2	V1-P1,LM-P1,AL-P1	0.32 (0.09,0.54)
B2	LM-P2	V1-P1,LM-P1,AL-P1,V1-P2	0.87 (0.79,0.91)
B3	LM-P2	V1-P1,LM-P1,AL-P1,LM-P2	0.92 (0.84,0.95)
B4	LM-P2	V1-P1,LM-P1,AL-P1,V1-P2, LM-P2	0.94 (0.89,0.96)

Difference	R^2
B2 - B0	0.07 (0.03,0.14)
B2 - B1	0.57 (0.34,0.75)
B3 - B1	0.61 (0.40,0.78)
B3 - B2	0.05 (0.00,0.10)
B4 - B2	0.07 (0.04,0.12)
B4 - B3	0.02 (0.00,0.05)

Model	Y	X	R^2
C0	V1-G	LM-G	0.62 (0.54,0.69)
C1	V1-G	P1s,P2s	0.34 (0.23,0.42)
C2	V1-G	LM-G,P1s	0.72 (0.65,0.80)
C3	V1-G	LM-G,P2s	0.65 (0.56,0.72)
C4	V1-G	LM-G,,P1s,P2s	0.74 (0.67,0.82)

Difference	R^2
C0 - C1	0.28 (0.18,0.36)
C2 - C0	0.10 (0.04,0.14)
C3 - C0	0.02 (0.00,0.07)
C4 - C1	0.39 (0.31,0.49)
C4 - C0	0.11 (0.07,0.16)

Model	Y	X	R^2
D0	AL-G	LM-G	0.73 (0.68,0.78)
D1	AL-G	P1s, P2s	0.72 (0.64,0.77)
D2	AL-G	LM-G,P1s	0.81 (0.77,0.85)
D3	AL-G	LM-G,P2s	0.80 (0.76,0.83)
D4	AL-G	LM-G,P1s,P2s	0.85 (0.81,0.88)

Difference	R^2
D1 - D0	0.01 (-0.06,0.11)
D2 - D0	0.07 (0.04,0.13)
D3 - D0	0.07 (0.02,0.12)
D4 - D1	0.13 (0.08,0.19)
D4 - D0	0.12 (0.07,0.17)

Table C.2: Regression analysis. See details in the main text.

Regression analysis The regression analysis studies the contribution of some features to predicting other features. The table contains different sets of linear regression models. Tables on the left show the R^2 posteriors median and 95% CI in the parentheses [50]. Y is the dependent variable and X indicates the explanatory variables. The tables on the right show the posterior of R^2 differences between models. As LM is a putative intermediate region between V1 and AL, we compare how much early features P1 timing, late feature LM-P2 timing and V1-P2 timing explain the variance of AL-P2 timing by running regression analysis. We design 5 scenarios: A0, the independent variable only has V1-P2 timing; A1, the independent variables only have P1 timing; A2, besides P1 timing, it considers V1-P2 timing; A3, besides P1 timing, it adds LM-P2 timing; A4, it contains the timing of P1 and both V1-P2 and LM-P2 timing. See the tables on the first row labeled by A. By comparing A0 and A2, P1 timing adds very limited improvement. By comparing the R^2 of A2, A3 with A1, both V1-P2 and LM-P2 timing explain much more than P1 timing. But LM-P2 timing contributes more than V1-P2 timing does, shown by the comparison between A2 and A3. The improvement from A2 to A4 implies that given V1-P2 timing, LM-P2 timing can still add extra prediction power, but not the other way around (nearly zero improvement from A3 to A4). We apply similar analysis to AL. 5 models with the select variables. See the tables on the second row labeled by B. The conclusions are similar except that given AL-P2 timing, V1-P2 timing can still make some contribution to predicting LM-P2 timing (shown by the contrast between B4 and B3). This further affirms our hypothesis that LM is an intermediate region between V1 and AL. We also conclude that P2 timing features are more relevant in predicting AL-P2 timing or LM-P2 timing rather than P1 timing features, this may be due to large time lag between P1 and P2. The last two sets of models labeled by C and D are related to G. We design 5 scenarios: C0, the independent variable only has LM-G; C1, the independent variables include all P1 and P2 timing; C2, besides LM-G, it includes all P1 timing; C3, besides LM-G, it includes all P2 timing; C4, includes LM-G and all P1 and P2 timing. Models labeled by D have similar design. LM-G can predict both V1-G and AL-G very well. One difference between C models and D models is that P1s and P2s together can predict AL-G much better than V1-G, see models C1 and D1 and related model differences. This means AL-G is correlated with activities in a broader regions or types than V1-G. This is probably because AL is a higher-order region.

A1 G	91	96	94	91	95	93	93	95
	A1 P1	95	93	93	93	92	93	90
		A1 P2	91	94	93	93	95	92
			A2 G	87	94	88	94	88
				A2 P1	93	89	93	92
					A2 P2	92	95	91
						A3 G	91	95
							A3 P1	95
								A3 P2

Table C.3: **CI coverage** CI coverage ratio percentage of each pair of features correlation. A1, A2, A3 stand for 3 virtual areas. G, P1, P2 stand for Gain, Peak-1, Peak-2.

A1 G	93	96	94	92	96	92	92	95
	A1 P1	95	92	92	93	92	93	94
		A1 P2	89	93	90	89	92	89
			A2 G	88	91	85	92	87
				A2 P1	92	92	92	92
					A2 P2	91	92	88
						A3 G	92	91
							A3 P1	92
								A3 P2

Table C.4: **CI coverage** The simulation scenario with mild neuron-to-neuron variance. The table is similar to C.3.

A1 G	93	96	95	92	95	92	94	95
	A1 P1	95	92	92	92	92	93	91
		A1 P2	89	92	92	92	96	90
			A2 G	89	93	84	95	87
				A2 P1	91	94	92	91
					A2 P2	95	93	90
						A3 G	92	92
							A3 P1	92
								A3 P2

Table C.5: **CI coverage** The simulation scenario with large neuron-to-neuron variance. The table is similar to C.3.

Appendix D

Appendix for Chapter 5

D.1 Optimization algorithm for the continuous-time point process regression model

D.1.1 Updating rules

Eq (5.2) is optimized using Newton's method. ϕ_w, ϕ_h are bases defined as,

$$\phi_w(t) := \int W(t-s) N_i(\mathrm{d}s), \qquad \phi_h(t) := \int h_{i\to j}(t-s) N_i(\mathrm{d}s) \tag{D.1}$$

Eq (5.4) can be rewritten as,

$$\tilde{\lambda}_j(t) = \beta_j \cdot 1 + \beta_w \phi_w(t) + \beta_h \phi_h(t) = \Psi(t)\boldsymbol{\beta} \tag{D.2}$$

$\Psi(t)$ represents the bases, $\boldsymbol{\beta}$ is a vector of all coefficients. If the coupling filter is fitted using non-parametric method, such as spline fitting $h_{i\to j}(s) = \beta_{h,1}B_1(s) + ... + \beta_{h,k}B_k(s)$. $B_1, ..., B_k$ are bases.

$$\phi_{h,1}(t) := \int B_1(t-s) N_i(\mathrm{d}s), ..., \ \phi_{h,k}(t) := \int B_k(t-s) N_i(\mathrm{d}s)$$

$$\tilde{\lambda}_j(t) = \beta_j \cdot 1 + \beta_w \phi_w(t) + \beta_{h,1}\phi_{h,1}(t) + ... + \beta_{h,k}\phi_{h,k}(t) = \Psi(t)\boldsymbol{\beta}$$

The first-order and second-order derivatives of the target equations are,

$$\frac{\partial L}{\partial \boldsymbol{\beta}} = -\int_0^T \frac{\Psi(s)}{\tilde{\lambda}_j(s)} \mathrm{d}N_j(s) + \int_0^T \Psi(s)\mathrm{d}s$$
$$\frac{\partial^2 L}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^T} = \int_0^T \frac{\Psi(s)\Psi(s)^T}{\tilde{\lambda}_j(s)^2} \mathrm{d}N_j(s).$$

The advantage the model Eq (D.2) is that the integral term $\int \Psi(s)\mathrm{d}s$ can be calculated in closed form if the bases Ψ are designed carefully. We only have such a convenience when the intensity function is modeled in linear scale but not others. For example, consider the model in logarithmic scale $\log\lambda(t) = \Psi(t)\beta$, then the derivative of the negative log-likelihood function becomes,

$$\frac{\partial L}{\partial \boldsymbol{\beta}} = -\int_0^T \Psi(s)\mathrm{d}N_j(s) + \int_0^T \Psi(s)e^{\Psi(s)\boldsymbol{\beta}}\mathrm{d}s.$$

Usually it is not tractable to calculate the second term so it is approximated by binning the data. Our model thus does not need to specify the time resolution. Another benefit of using continuous-time model is that the number of data points is small, which is proportional to the number of spikes instead of the number of time bins. For example, if the bin width is 1 ms, then for one 1-second long trial, it needs to store 1000 data points. If the trial has 20 spikes, the continuous-time model only needs to keep 20 data points. The memory space is 50 times smaller.

If the regression bases have form Eq (D.1) with kernel, then

$$\int_0^T \Psi(t)\mathrm{d}t = \int_0^T\int_0^T K(t-s)N_i(\mathrm{d}s) = N_i(T)\int_{\mathrm{R}} K(s)\mathrm{d}s.$$

If K is a Normal window function or a square window function, the above integral is simple. The boundary effect can be removed in the integral by only considering a few time point close to 0 or T. Next, we show how to calculate such integral if K is B-spline, which is widely used in non-parametric curve fitting. For example, the coupling filter in Supplementary D.4.5 is fitted using B-splines non-parametrically.

The B-splines are defined using Cox-de Boor recursion equations. t_i are knots (with repeated padding). p is the degree of the spline polynomial. When $p = 3$, these are the cubic splines.

$$\begin{aligned} B_{i,0}(x) =& \mathbb{I}_{[t_i,t_{i+1})}(x) \\ B_{i,p}(x) =& \frac{x-t_i}{t_{i+p}-t_i}B_{i,p-1}(x) + \frac{t_{i+p+1}-x}{t_{i+p+1}-t_{i+1}}B_{i+1,p-1}. \end{aligned}$$

Knot padding is important to create proper splines. If $p = 3$ and the distinct knot locations are $(0, 1, 2)$, the input knots should be $(0, 0, 0, 0, 1, 2, 2, 2, 2)$. The knots need extra p repeated knots of the two ends. If there are K distinct knots, then there are $K + 2p$ input knots. The total number of basis is $K + p - 1$.

Lemma D.1.1. *For the B-spline curve defined above, the integral of the curve has closed-form as follows,*

$$\int_{-\infty}^{\infty} B_{i,p}(s)\mathrm{d}s = \frac{t_{i+p+1}-t_i}{p+1}. \tag{D.3}$$

Proof. The support of each basis spans over $p + 1$ knot-intervals (including the padded knots on the ends),

$$\mathrm{supp}(B_{i,p}) =[t_i, t_{i+p+1})$$

$$\frac{\mathrm{d}}{\mathrm{d}x}B_{i,p}(x) = \frac{p}{t_{i+p}-t_i}B_{i,p-1}(x) - \frac{p}{t_{i+p+1}-t_{i+1}}B_{i+1,p-1}(x).$$

The support of the derivative is almost the same as the basis except for a few 0 derivative points.

$$\text{supp}(\frac{\mathrm{d}}{\mathrm{d}x}B_{i,p}) \subseteq [t_i, t_{i+p+1})$$

We reform the derivative properties to get the integral [15].

$$\frac{\mathrm{d}}{\mathrm{d}x}\sum_{i=0}^{\infty} c_i B_{i,p+1}(x) = \sum_{i=0}^{\infty}(p+1)\frac{c_i - c_{i-1}}{t_{i+p+1} - t_i}B_{i,p}(x)$$

c_i are some arbitrary coefficients. Next we set $c_0, ..., c_{i-1} = 0$, $c_i, c_{i+1}, ... = 1$.

$$\frac{\mathrm{d}}{\mathrm{d}x}\sum_{j=i}^{\infty} c_j B_{j,p+1}(x) = \frac{\mathrm{d}}{\mathrm{d}x}\sum_{j=i}^{i+p} B_{j,p+1}(x) = \frac{p+1}{t_{i+p+1} - t_i}B_{i,p}(x)$$

The first equation simplifies the sum due to the supports of bases. Then take the integral on both side,

$$\int_{-\infty}^{x} B_{i,p}(s)\mathrm{d}s = \int_{t_i}^{x} B_{i,p}(s)\mathrm{d}s = \frac{t_{i+p+1} - t_i}{p+1}\sum_{j=i}^{\infty} B_{j,p+1}(s) = \frac{t_{i+p+1} - t_i}{p+1}\sum_{j=i}^{i+p} B_{j,p+1}(x)$$

The area under the curve of a basis is,

$$\int_{-\infty}^{\infty} B_{i,p}(s)\mathrm{d}s = \int_{t_i}^{t_{i+p+1}} B_{i,p}(s)\mathrm{d}s = \frac{t_{i+p+1} - t_i}{p+1}\sum_{j=i}^{i+p} B_{j,p+1}(t_{i+p+1})$$

Consider the summation term,

$$\begin{aligned}
\sum_{j=i}^{i+p} B_{j,p+1}(t_{i+p+1}) =& B_{i,p+1}(t_{i+p+1}) + B_{i+1,p+1}(t_{i+p+1}) + ... + B_{i+p,p+1}(t_{i+p+1}) \\
=& \Big(\frac{t_{i+p+1} - t_i}{t_{i+p+1} - t_i}B_{i,p}(t_{i+p+1}) + \frac{t_{i+p+2} - t_{i+p+1}}{t_{i+p+2} - t_{i+1}}B_{i+1,p}(t_{i+p+1})\Big) \\
&+ \Big(\frac{t_{i+p+1} - t_{i+1}}{t_{i+p+2} - t_{i+1}}B_{i+1,p}(t_{i+p+1}) + \frac{t_{i+p+3} - t_{i+p+1}}{t_{i+p+3} - t_{i+2}}B_{i+2,p}(t_{i+p+1})\Big) + ... \\
&+ \Big(\frac{t_{i+p+1} - t_{i+p}}{t_{i+2p+1} - t_{i+p}}B_{i+p,p}(t_{i+p+1}) + \frac{t_{i+2p+2} - t_{i+p+1}}{t_{i+2p+2} - t_{i+p+1}}B_{i+p+1,p}(t_{i+p+1})\Big) \\
=& B_{i,p}(t_{i+p+1}) + B_{i+1,p}(t_{i+p+1}) + ... + B_{i+p+1,p}(t_{i+p+1}) \\
=& B_{i,p-1}(t_{i+p+1}) + B_{i+1,p}(t_{i+p+1}) + ... + B_{i+p+2,p-1}(t_{i+p+1}) \\
=& B_{i,0}(t_{i+p+1}) + B_{i+1,0}(t_{i+p+1}) + ... + B_{i+2p+1,0}(t_{i+p+1}) = 1
\end{aligned}$$

So the conclusion holds. □

D.1.2 Plug-in estimator for the smoothing kernel width

As discussed in Fig 5.4, the selection of the smoothing kernel width σ_w is determined by maximizing the likelihood and it is insensitive to the coupling filter amplitude or timescale. This motivates the design of the plug-in estimator of σ_w, which means the optimal kernel width σ_w can be determined without knowing details of the coupling filter, or even without any modeling of the coupling filter. The plug-in estimator is a shortcut of the model selection, especially in the situations where the coupling filters might change from trial to trial, or from neuron to neuron. As will be shown in Lemma D.2.2 and D.2.4, the negative log-likelihood of a trial on $[0, T]$ as function of σ_w can be approximated as the following ignoring the constant term,

$$L(\sigma_w) \approx \frac{1}{2\bar{\lambda}_j} \begin{pmatrix} \langle \varphi_w, \mathbf{s}_j \rangle \\ \langle \varphi_h, \mathbf{s}_j \rangle \end{pmatrix}^T \begin{pmatrix} \langle \varphi_w, \varphi_w \rangle & \langle \varphi_w, \varphi_h \rangle \\ \langle \varphi_h, \varphi_w \rangle & \langle \varphi_h, \varphi_h \rangle \end{pmatrix}^{-1} \begin{pmatrix} \langle \varphi_w, \mathbf{s}_j \rangle \\ \langle \varphi_h, \mathbf{s}_j \rangle \end{pmatrix}$$

$$\varphi_w = W * (\mathbf{s}_i - \bar{\lambda}_i), \quad \varphi_h = h * (\mathbf{s}_i - \bar{\lambda}_i)$$

$\langle \cdot, \cdot \rangle$ denotes the inner product on interval $[0, T]$. As demonstrated in Fig 5.4, the peak position of the log-likelihood is insensitive to the amplitude or the timescale coupling filter h, so we arbitrarily select $h(t) = \mathbb{I}_{[0,\sigma_h]}(t)$, $\sigma_h = 30$ ms. Having a very small filter length, say $\sigma = 1$ ms, or dropping the terms related to h could make the surrogate approximated likelihood curves in the above equation a little different from the likelihood obtained through the optimization in Eq 5.2. So we still keep the terms related to h. $\mathbf{s}_i, \mathbf{s}_j$ are spike trains, $\langle \varphi_w, \mathbf{s}_j \rangle := \int_0^T \varphi_w(t) N_j(\mathrm{d}t)$. The inner product is calculated by discretizing the time series into small time bins, for example 1 ms. So $\mathbf{s}_i, \mathbf{s}_j$ are arrays of binary values with 1 indicating the spikes. This does not go against the advantage of the continuous-time model. For a large dataset, the plug-in estimator can be calculated using a subset of samples. We grid search σ_w with 5 ms step size to find the largest $L(\sigma_w)$. The estimator is not very sensitive to σ_w. For example, as shown in Fig 5.3, Supplementary D.4.4 and D.4.15, selecting kernel scale 20 ms larger or smaller will not affect the estimated value a lot.

D.2 Theoretical properties of the estimator

In this section, we provide theoretical derivations of the estimator properties, including bias, standard error, risk, and asymptotic Normality. All of these are built on the second-order stationary condition, which is described as follows [34],

Lemma D.2.1. *Let ξ be a second-order stationary random measure on $\mathcal{X}$. It satisfies two properties.*

1. *The first-moment measure is $M_{\xi,1}(A) := \mathbb{E}\xi(A)$, where A is a set in the Borel σ-field of $\mathcal{X}$, satisfies,*
$$M_{\xi,1}(\mathrm{d}x) = \bar{\lambda}\mathrm{d}x \tag{D.4}$$
where $\bar{\lambda}$ is a constant, which is called the mean density.

2. *The second-moment measure is* $M_{\xi,2}(A \times B) := \mathbb{E}\xi(A)\xi(B)$. A, B *are sets in the Borel* σ*-field of* $\mathcal{X}$. *The second-moment can be expressed as the product of a Lebesgue component* $\mathrm{d}x$ *and a reduced measure, say* $\breve{M}_{\xi,2}$. $\breve{m}_{\xi,2}$ *is the density of the reduced measure* $\breve{M}_{\xi,2}(\mathrm{d}u) = \breve{m}_{\xi,2}(u)\mathrm{d}u$. *The following equation holds,*

$$\int_{\mathcal{X}}\int_{\mathcal{X}} f(s,t)M_{\xi,2}(\mathrm{d}s \times \mathrm{d}t) = \int_{\mathcal{X}}\int_{\mathcal{X}} f(x, x+u)\mathrm{d}x \cdot \breve{m}_{\xi,2}(u)\mathrm{d}u \tag{D.5}$$

The reduced second-moment measure $\breve{M}_{\xi,2}$ is symmetric, positive, positive-definite and translation-bounded. Details can be found in [34, proposition 8.1.I, 8.1.II]. The mean corrected process is $\tilde{\xi}(A) := \xi(A) - \bar{\lambda}\ell(A)$. Similarly, the reduced covariance measure and its density can be defined as,

$$\breve{C}_{\xi,2}(\mathrm{d}u) := \breve{M}_{\tilde{\xi},2}(\mathrm{d}u) = \breve{M}_{\xi,2}(\mathrm{d}u) - \bar{\lambda}^2\mathrm{d}u \tag{D.6}$$

$$\breve{c}_{\xi,2}(u) = \breve{m}_{\xi,2}(u) - \bar{\lambda}^2 \tag{D.7}$$

Similar definitions can be used for two different second-order stationary processes ξ, ζ.

$$M_{\xi\zeta,2}(A \times B) := \mathbb{E}\xi(A)\zeta(B) \tag{D.8}$$

$$\int_{\mathcal{X}}\int_{\mathcal{X}} f(s,t)M_{\xi\zeta,2}(\mathrm{d}s \times \mathrm{d}t) = \int_{\mathcal{X}}\int_{\mathcal{X}} f(x, x+u)\mathrm{d}x \cdot \breve{m}_{\xi\zeta,2}(u)\mathrm{d}u \tag{D.9}$$

$$\breve{C}_{\xi\zeta,2}(\mathrm{d}u) = \breve{M}_{\xi\zeta,2}(\mathrm{d}u) - \bar{\lambda}_\xi\bar{\lambda}_\zeta\mathrm{d}u \tag{D.10}$$

$$\breve{c}_{\xi\zeta,2}(u) = \breve{m}_{\xi\zeta,2}(u) - \bar{\lambda}_\xi\bar{\lambda}_\zeta \tag{D.11}$$

Lemma D.2.2. *Assuming* $f_{i,j}$ *is second-order stationary, the bias of the estimator* $\hat{\beta}_h$ *in model* (5.2) *is approximated as,*

$$\mathrm{bias}(\hat{\beta}_h) \approx \frac{\mathrm{Denominator}}{\mathrm{Denominator}} \tag{D.12}$$

$$\begin{aligned} \mathrm{Numerator} = & \left(\int_{\mathbb{R}} [W * W](s)\breve{c}_{N,2}(\mathrm{d}s)\mathrm{d}s\right) \cdot \left(\int_{\mathbb{R}} h(s)\breve{c}_{N\Lambda,2}(s)\mathrm{s}\right) \\ & - \left(\int_{\mathbb{R}} [h * W](s)\breve{c}_{N,2}(\mathrm{d}s)\mathrm{d}s\right) \cdot \left(\int_{\mathbb{R}} W(s)\breve{c}_{N\Lambda,2}(s)\mathrm{s}\right) \\ \mathrm{Denominator} = & \left(\int_{\mathbb{R}} [W * W](s)\breve{c}_{N,2}(\mathrm{d}s)\mathrm{d}s\right) \cdot \left(\int_{\mathbb{R}} [h * h^-](s)\breve{c}_{N,2}(\mathrm{d}s)\mathrm{d}s\right) \\ & - \left(\int_{\mathbb{R}} W(s)\breve{c}_{N\Lambda,2}(s)\mathrm{s}\right)^2 \end{aligned} \tag{D.13}$$

$\breve{c}_{N,2}$ *is the reduced second-order moment measure intensity of spike count measure* $N_i(\cdot)$; $\breve{c}_{N\Lambda,2}$ *is the reduced second-order moment measure intensity of between spike count measure* $N_i(\cdot)$ *and intensity measure* $\Lambda_i(\cdot)$ *as described in Lemma D.2.1. If the shared activity* $f_{i,j}$ *is the cluster process in Eq* (5.6) *with parameters* σ_I, ρ, *and the coupling filter has form in Eq* (5.7) *with* *parameters σ_h then we have the closed-form as follows,*

$$
\begin{aligned}
\text{Numerator}(\sigma_w) =& \left(\frac{\rho}{2\sqrt{\pi}\sqrt{\sigma_w^2 + \sigma_I^2}} + \frac{\bar{\lambda}_i}{2\sqrt{\pi}\sigma_w} \right) \cdot \left(\frac{\rho}{2} \text{erf}\left(\frac{\sigma_h}{2\sigma_I} \right) \right) \\
& - \left(\frac{\rho}{2} \text{erf}\left(\frac{\sigma_h}{2\sqrt{\sigma_w^2/2 + \sigma_I^2}} \right) + \frac{\bar{\lambda}_i}{2} \text{erf}\left(\frac{\sigma_h}{\sqrt{2}\sigma_w} \right) \right) \cdot \left(\frac{\rho}{2\sqrt{\pi}\sqrt{\sigma_w^2/2 + \sigma_I^2}} \right) \\
\text{Denominator}(\sigma_w) =& \left(\frac{\rho}{2\sqrt{\pi}\sqrt{\sigma_w^2 + \sigma_I^2}} + \frac{\bar{\lambda}_i}{2\sqrt{\pi}\sigma_w} \right) \cdot \\
& \left(\rho \left[\sigma_h \text{erf}\left(\frac{\sigma_h}{2\sigma_I} \right) - \frac{2\sigma_I}{\sqrt{\pi}} \left(1 - e^{-\frac{\sigma_h^2}{4\sigma_I^2}} \right) \right] + \bar{\lambda}_i \sigma_h \right) \\
& - \left(\frac{\rho}{2} \text{erf}\left(\frac{\sigma_h}{2\sqrt{\sigma_w^2/2 + \sigma_I^2}} \right) + \frac{\bar{\lambda}_i}{2} \text{erf}\left(\frac{\sigma_h}{\sqrt{2}\sigma_w} \right) \right)^2
\end{aligned}
\tag{D.14}
$$

$\bar{\lambda}_i = \mathbb{E}[N_i(\mathrm{d}t)/\mathrm{d}t] = \alpha_i + \rho$. $\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} \mathrm{d}t$.

Proof. The bases of the regression model (5.4) include a constant, the nuisance variable $\overline{\mathbf{s}_i} = W * \mathbf{s}_i$, and the coupling filter $h_{i \to j} * \mathbf{s}_i$. $\tilde{\lambda}_j$ in Eq (5.4) can be rewritten as,

$$
\tilde{\lambda}_j(s) = \beta_j + \beta_w \varphi_w(s) + \beta_h \varphi_h(s)
$$

φ_w, φ_h are mean-subtracted bases defined as,

$$
\begin{aligned}
\varphi_w(s) &:= \int W(s-t)(N_i(\mathrm{d}t) - \bar{\lambda}_i \mathrm{d}t) \\
\varphi_h(s) &:= \int h_{i \to j}(s-t)(N_i(\mathrm{d}t) - \bar{\lambda}_i \mathrm{d}t)
\end{aligned}
$$

The above mean normalization will not change the value of the coefficients β_w and β_h, but the baseline coefficient β_j is different from model (5.4). Define the following shorthands,

$$
S_{ww} := \langle \varphi_w, \varphi_w \rangle, \quad S_{hh} := \langle \varphi_h, \varphi_h \rangle, \quad S_{hw} = S_{wh} := \langle \varphi_w, \varphi_h \rangle S_{w\lambda} := \langle \varphi_w, \lambda_i \rangle, \quad S_{h\lambda} := \langle \varphi_h, \lambda_i \rangle,
\tag{D.15}
$$

$\langle \cdot, \cdot \rangle$ denotes the inner product between two functions on interval $[0, T]$.

$$
\begin{aligned}
& \frac{\partial L}{\partial \beta_j} = -\int_0^T \frac{1}{\tilde{\lambda}_j(s)} N_j(\mathrm{d}s) + \int_0^T 1 \mathrm{d}s \\
=& T - \int_0^T \frac{1}{\beta_j + \beta_w \varphi_w(s) + \beta_h \varphi_h(s)} N_j(\mathrm{d}s) = T - \int_0^T \frac{1}{\beta_j} \cdot \frac{1}{1 + \frac{\beta_w}{\beta_j}\varphi_w(s) + \frac{\beta_h}{\beta_h}\varphi_h(s)} N_j(\mathrm{d}s) \\
=& T - \int_0^T \frac{1}{\beta_j} \cdot \left(1 - \frac{\beta_w}{\beta_j}\varphi_w(s) - \frac{\beta_h}{\beta_j}\varphi_h(s) \right) N_j(\mathrm{d}s) + o\left(\int_0^T \frac{1}{\beta_j} \cdot \left(\frac{\beta_w}{\beta_j}\varphi_w(s) + \frac{\beta_h}{\beta_j}\varphi_h(s) \right) N_j(\mathrm{d}s) \right) \\
=& T - \frac{N_j(T)}{\beta_j} + O\left(\int_0^T \frac{1}{\beta_j} \cdot \left(\frac{\beta_w}{\beta_j}\varphi_w(s) + \frac{\beta_h}{\beta_j}\varphi_h(s) \right) N_j(\mathrm{d}s) \right) \approx T - \frac{N_j(T)}{\beta_j}
\end{aligned}
$$

The quantity in omitted term

$$
\begin{aligned}
&\int_0^T \varphi_h(s) N_j(\mathrm{d}s) \approx \mathbb{E}\left[\int_0^T \varphi_h(s) N_j(\mathrm{d}s)\middle| N_i\right] = \int_0^T \varphi_h(s)\lambda_j(s)\mathrm{d}s \\
=&\int_0^T \varphi_h(s)\Big(\text{constant} + f_{i,j}(s) + \alpha_h\varphi_h(s)\Big)\mathrm{d}s \\
=&\int_0^T \varphi_h(s)\Big(\alpha_i + f_{i,j}(s) + \alpha_h\varphi_h(s)\Big)\mathrm{d}s = \int_0^T \varphi_h(s)\Big(\lambda_i(s) + \alpha_h\varphi_h(s)\Big)\mathrm{d}s
\end{aligned}
$$

The conditional expectation is over count process $N_j(\cdot)$, and the equation holds because of the Campbell lemma [93, Lemma 1.1]. So the omitted term or the approximation error is

$$\frac{\beta_w}{\beta_j^2}(S_{w\lambda} + \alpha_h S_{wh}) + \frac{\beta_w}{\beta_j^2}(S_{h\lambda} + \alpha_h S_{hh})$$

Terms $S_{w\lambda}, S_{wh}, S_{h\lambda}, S_{hh}$ will be derived in Lemma D.2.6. The trial is in time interval $[0, T]$. So the baseline is approximately,

$$\hat{\beta}_j \approx \frac{N_j(T)}{T} \approx \bar{\lambda}_j$$

The baseline parameter is approximately fixed at the mean firing rate $\bar{\lambda}_j$, so we only need to consider the negative log-likelihood function $L(\beta_w, \beta_h)$ of two parameters $\boldsymbol{\beta} = (\beta_w, \beta_h)^T$. It can be approximated using the quadratic form,

$$L \approx \frac{1}{2}\boldsymbol{\beta}^T H \boldsymbol{\beta} + \boldsymbol{b}^T\boldsymbol{\beta} + \text{constant} \tag{D.16}$$

$$\frac{\partial L}{\partial \boldsymbol{\beta}} \approx H\boldsymbol{\beta} + \boldsymbol{b}, \quad H = \frac{\partial^2 L}{\partial \boldsymbol{\beta}\partial \boldsymbol{\beta}^T} \tag{D.17}$$

$$
\begin{aligned}
H =& \frac{\partial^2 L}{\partial \beta \partial \beta^T} = \int_0^T \frac{\Psi(s)\Psi(s)^T}{\tilde{\lambda}_j(s)^2} N_j(\mathrm{d}s) \approx \mathbb{E}_{N_j}\left[\int_0^T \frac{\Psi(s)\Psi(s)^T}{\tilde{\lambda}_j(s)^2} N_j(\mathrm{d}s)\middle| N_i\right] \\
=& \int_0^T \frac{\Psi(s)\Psi(s)^T}{\tilde{\lambda}_j(s)^2}\lambda_j(s)\mathrm{d}s \approx \frac{1}{\bar{\lambda}_j}\int_0^T \Psi(s)\Psi(s)^T \mathrm{d}s
\end{aligned}
$$

$\Psi(s) = (\varphi_w, \varphi_h)^T$ is the vector of two bases. The parameter **b** in Eq (D.16) can be solved using two special solutions $\hat{\boldsymbol{\beta}}^B$ and $\hat{\boldsymbol{\beta}}^C$.

$$
\begin{aligned}
\hat{\beta}_h^B = 0, &\quad \frac{\partial L(\hat{\beta}_w^B, 0)}{\partial \beta_w} = 0 \\
\hat{\beta}_w^C = 0, &\quad \frac{\partial L(0, \hat{\beta}_h^C)}{\partial \beta_h} = 0
\end{aligned}
$$

As for $\hat{\beta}_w^B$, let us compare it with model $\tilde{\lambda}_j = \beta_j + \beta_w \varphi_w$

$$\begin{aligned}
0 = \frac{\partial L_T(\hat{\beta})}{\partial \beta_w} &= -\int_0^T \frac{\varphi_w(s)}{\tilde{\lambda}_j(s)} \mathrm{d}N_j(s) + \int_0^T \varphi_w(s) \mathrm{d}s \\
&= -\int_0^T \varphi_w(s) \frac{1}{\bar{\lambda}_j + \hat{\beta}_w^B \varphi_w(s)} \mathrm{d}N_j(s) = -\frac{1}{\bar{\lambda}_j} \int_0^T \varphi_w(s) \frac{1}{1 + \frac{\hat{\beta}_w^B}{\bar{\lambda}_j} \varphi_w(s)} \mathrm{d}N_j(s) \\
&= -\frac{1}{\bar{\lambda}_j} \int_0^T \varphi_w(s) \left(1 - \frac{\hat{\beta}_w^B}{\bar{\lambda}_j} \varphi_w(s)\right) \mathrm{d}N_j(s) + o\left(\frac{\hat{\beta}_w^B}{\bar{\lambda}_j^2} \int_0^T \varphi_w(s)\varphi_w(s) \mathrm{d}N_j(s)\right) \\
&\approx \mathbb{E}\left[-\frac{1}{\bar{\lambda}_j} \int_0^T \varphi_w(s) \left(1 - \frac{\hat{\beta}_w^B}{\bar{\lambda}_j} \varphi_w(s)\right) \mathrm{d}N_j(s) \middle| N_i\right] \\
&= -\frac{1}{\bar{\lambda}_j} \int_0^T \varphi_w(s) \left(1 - \frac{\hat{\beta}_w^B}{\bar{\lambda}_j} \varphi_w(s)\right) \lambda_j(s) \mathrm{d}s
\end{aligned}$$

Then we can derive the $\hat{\beta}_w^B$,

$$\hat{\beta}_w^B \approx \bar{\lambda}_j \frac{\langle \varphi_w, \lambda_j \rangle}{\langle \varphi_w^2, \lambda_j \rangle} \approx \bar{\lambda}_j \frac{\langle \varphi_w, \lambda_j \rangle}{\langle \varphi_w^2, \bar{\lambda}_j \rangle} = \frac{\langle \varphi_w, \lambda_j \rangle}{\langle \varphi_w, \varphi_w \rangle}$$

Similarly, we have

$$\hat{\beta}_h \approx \bar{\lambda}_j \frac{\langle \varphi_h, \lambda_j \rangle}{\langle \varphi_h^2, \lambda_j \rangle} \approx \frac{\langle \varphi_h, \lambda_j \rangle}{\langle \varphi_h, \varphi_h \rangle}$$

The MLE then is,

$$\hat{\beta} \approx -H^{-1} \hat{\boldsymbol{b}} \tag{D.18}$$

$$H \approx \frac{1}{\bar{\lambda}_j} \begin{pmatrix} S_{ww} & S_{wh} \\ S_{hw} & S_{hh} \end{pmatrix}, \quad \boldsymbol{b} \approx -\frac{1}{\bar{\lambda}_j} \begin{pmatrix} \langle \varphi_w, \lambda_j \rangle \\ \langle \varphi_h, \lambda_j \rangle \end{pmatrix}$$

So we have the estimator $\hat{\beta}_h$,

$$\begin{aligned}
\hat{\beta}_h \approx& \frac{S_{ww} \cdot \langle \varphi_h, \lambda_j \rangle - S_{hw} \cdot \langle \varphi_w, \lambda_j \rangle}{S_{ww} S_{hh} - S_{wh}^2} \\
=& \frac{S_{ww} \cdot \langle \varphi_h, \alpha_j' + f_{i,j} + \alpha_h \varphi_h \rangle - S_{hw} \cdot \langle \varphi_w, \alpha_j' + f_{i,j} + \alpha_h \varphi_h \rangle}{S_{ww} S_{hh} - S_{wh}^2} \\
=& \frac{S_{ww} \cdot \langle \varphi_h, f_{i,j} \rangle - S_{hw} \cdot \langle \varphi_w, f_{i,j} \rangle}{S_{ww} S_{hh} - S_{wh}^2} + \alpha_h \cdot \frac{S_{ww} \cdot \langle \varphi_h, \varphi_h \rangle - S_{hw} \cdot \langle \varphi_w, \varphi_h \rangle}{S_{ww} S_{hh} - S_{wh}^2} \\
=& \frac{S_{ww} \cdot \langle \varphi_h, \alpha_i + f_{i,j} \rangle - S_{hw} \cdot \langle \varphi_w, \alpha_i + f_{i,j} \rangle}{S_{ww} S_{hh} - S_{wh}^2} + \alpha_h \\
\approx& \alpha_h + \frac{S_{ww} \langle \varphi_h, \lambda_i \rangle - S_{hw} \langle \varphi_w, \lambda_i \rangle}{S_{ww} S_{hh} - S_{hw}^2}
\end{aligned}$$

So the bias of the estimator is approximately,

$$\mathrm{bias}(\hat{\beta}_h) \approx \frac{S_{ww} S_{h\lambda} - S_{hw} S_{w\lambda}}{S_{ww} S_{hh} - S_{hw}^2} \tag{D.19}$$

Lemma D.2.6 shows the derivation of the inner products $S_{ww}, S_{hh}, S_{hw}, S_{w\lambda}, S_{h\lambda}$. □

Corollary D.2.2.1. *When the smoothing kernel becomes infinitely narrow, the bias in Eq* (D.14) *satisfies*

$$\lim_{\sigma_w \to 0} \frac{\text{Numerator}}{\text{Denominator}} \to \frac{\frac{\rho}{2}\text{erf}\left(\frac{\sigma_h}{2\sigma_I}\right)}{\rho\left[\sigma_h \text{erf}\left(\frac{\sigma_h}{2\sigma_I}\right) - \frac{2\sigma_I}{\sqrt{\pi}}\left(1 - e^{-\frac{\sigma_h^2}{4\sigma_I^2}}\right)\right] + \bar{\lambda}_i \sigma_h} \tag{D.20}$$

Corollary D.2.2.2. *When the smoothing kernel becomes infinitely wide, the bias in Eq* (D.14) *satisfies*

$$\lim_{\sigma_w \to \infty} \frac{\text{Numerator}}{\text{Denominator}} \to \frac{\frac{\rho}{2}\text{erf}\left(\frac{\sigma_h}{2\sigma_I}\right)}{\rho\left[\sigma_h \text{erf}\left(\frac{\sigma_h}{2\sigma_I}\right) - \frac{2\sigma_I}{\sqrt{\pi}}\left(1 - e^{-\frac{\sigma_h^2}{4\sigma_I^2}}\right)\right] + \bar{\lambda}_i \sigma_h} \tag{D.21}$$

Corollary D.2.2.3. *If the regression model Eq* (5.4) *does not include the nuisance variable, which becomes a typical Hawkes process, then the bias of the estimator is*

$$\text{bias}(\hat{\beta}_h) \approx \frac{S_{h\lambda}}{S_{hh}} \approx \frac{\frac{\rho}{2}\text{erf}\left(\frac{\sigma_h}{2\sigma_I}\right)}{\rho\left[\sigma_h \text{erf}\left(\frac{\sigma_h}{2\sigma_I}\right) - \frac{2\sigma_I}{\sqrt{\pi}}\left(1 - e^{-\frac{\sigma_h^2}{4\sigma_I^2}}\right)\right] + \bar{\lambda}_i \sigma_h} \tag{D.22}$$

The approximation is similar to Lemma D.2.2. Note that the three corollaries have the same results.

Lemma D.2.3. *The variance of the estimator Eq* (5.2) *for one trial on time interval* $[0, T]$ *is approximated as,*

$$\text{Var}(\hat{\beta}_h) \approx \frac{\text{Numerator}}{\text{Denominator}} \tag{D.23}$$

$$\text{Numerator} = \frac{\bar{\lambda}_j}{T}\left(\int_{\mathbb{R}} [W * W](s) \breve{c}_{N,2}(\mathrm{d}s)\mathrm{d}s\right) \tag{D.24}$$

$$\text{Denominator} = \left(\int_{\mathbb{R}} [W * W](s) \breve{c}_{N,2}(\mathrm{d}s)\mathrm{d}s\right) \cdot \left(\int_{\mathbb{R}} [h * h^-](s) \breve{c}_{N,2}(\mathrm{d}s)\mathrm{d}s\right) \tag{D.25}$$

$$- \left(\int_{\mathbb{R}} W(s) \breve{c}_{N\Lambda,2}(s)\mathrm{s}\right)^2 \tag{D.26}$$

If $f_{i,j}$ follows the cluster process in Eq (5.6)*, then*

$$
\begin{aligned}
\mathrm{Numerator}(\sigma_w) =& \frac{\bar{\lambda}_j}{T}\left(\frac{\rho}{2\sqrt{\pi}\sqrt{\sigma_w^2+\sigma_I^2}}+\frac{\bar{\lambda}_i}{2\sqrt{\pi}\sigma_w}\right) \\
\mathrm{Denominator}(\sigma_w) =& \left(\frac{\rho}{2\sqrt{\pi}\sqrt{\sigma_w^2+\sigma_I^2}}+\frac{\bar{\lambda}_i}{2\sqrt{\pi}\sigma_w}\right)\cdot \\
& \left(\rho\left[\sigma_h \mathrm{erf}\left(\frac{\sigma_h}{2\sigma_I}\right)-\frac{2\sigma_I}{\sqrt{\pi}}\left(1-e^{-\frac{\sigma_h^2}{4\sigma_I^2}}\right)\right]+\bar{\lambda}_i\sigma_h\right) \\
& -\left(\frac{\rho}{2}\mathrm{erf}\left(\frac{\sigma_h}{2\sqrt{\sigma_w^2/2+\sigma_I^2}}\right)+\frac{\bar{\lambda}_i}{2}\mathrm{erf}\left(\frac{\sigma_h}{\sqrt{2}\sigma_w}\right)\right)^2
\end{aligned}
\tag{D.27}
$$

The proof is similar to Lemma D.2.2 using the Fisher information.

Corollary D.2.3.1. *If $\sigma_w \to 0$ of the variance in Eq* (D.27) *will converge*

$$
\lim_{\sigma_w\to 0}\frac{\mathrm{Numerator}}{\mathrm{Denominator}} \to \frac{\bar{\lambda}_i}{T}\left(\rho\left[\sigma_h \mathrm{erf}\left(\frac{\sigma_h}{2\sigma_I}\right)-\frac{2\sigma_I}{\sqrt{\pi}}\left(1-e^{-\frac{\sigma_h^2}{4\sigma_I^2}}\right)\right]+\bar{\lambda}_i\sigma_h\right)^{-1} \tag{D.28}
$$

Corollary D.2.3.2. *If $\sigma_w \to \infty$ of the variance in Eq* (D.27) *will converge*

$$
\lim_{\sigma_w\to \infty}\frac{\mathrm{Numerator}}{\mathrm{Denominator}} \to \frac{\bar{\lambda}_i}{T}\left(\rho\left[\sigma_h \mathrm{erf}\left(\frac{\sigma_h}{2\sigma_I}\right)-\frac{2\sigma_I}{\sqrt{\pi}}\left(1-e^{-\frac{\sigma_h^2}{4\sigma_I^2}}\right)\right]+\bar{\lambda}_i\sigma_h\right)^{-1} \tag{D.29}
$$

Corollary D.2.3.3. *If the regression model Eq* (5.4) *does not include the nuisance variable, which becomes a typical Hawkes process, then the variance of the estimator is*

$$
\mathrm{Var}(\hat{\beta}_h) \approx \frac{\bar{\lambda}}{S_{hh}} \approx \frac{\bar{\lambda}_i}{T}\left(\rho\left[\sigma_h \mathrm{erf}\left(\frac{\sigma_h}{2\sigma_I}\right)-\frac{2\sigma_I}{\sqrt{\pi}}\left(1-e^{-\frac{\sigma_h^2}{4\sigma_I^2}}\right)\right]+\bar{\lambda}_i\sigma_h\right)^{-1}
$$

The proof is similar to Lemma D.2.2. The three corollaries above have the same results.

Lemma D.2.4. *The negative log-likelihood of Eq* (5.2) *can be approximated by*

$$
L \approx \frac{1}{2\bar{\lambda}_j}\begin{pmatrix} S_{w\lambda}+\alpha_h S_{hw} \\ S_{h\lambda}+\alpha_h S_{hh}\end{pmatrix}^T \begin{pmatrix} S_{ww} & S_{wh} \\ S_{hw} & S_{hh}\end{pmatrix}^{-1} \begin{pmatrix} S_{w\lambda}+\alpha_h S_{hw} \\ S_{h\lambda}+\alpha_h S_{hh}\end{pmatrix} + \mathrm{constant} \tag{D.30}
$$

The values $S_{ww}, S_{hh}, S_{hw}, S_{w\lambda}, S_{h\lambda}$ is shown in Lemma D.2.6.

The proof is similar to Lemma D.2.2.

Lemma D.2.5. *Let the estimator be the model in Eq* (5.2)*. Assume $|\hat{\beta}|$, the variance of $\hat{\beta}_h$, $\frac{\partial L}{\partial \beta}$, and the third-order derivative of the intensity function and log-intensity function are bounded; $\frac{\partial L}{\partial \beta}$ is a continuous function of β; and $\mathbb{E}|\hat{\beta}_h-\alpha_h|^2 = o\left(\frac{1}{\sqrt{R}}\right)$, where R is the number of trials. Then $\hat{\beta}$ is asymptotically Normal*

$$
\sqrt{R}(\hat{\beta}-\alpha) \sim N\left(\boldsymbol{0}, I(\beta)^{-1}\right) \tag{D.31}
$$

$I(\beta)$ is the Fisher information.

Proof. We expand the negative log-likelihood function of MLE $\hat{\beta} = (\hat{\beta}_h^c, \hat{\beta}_h)$ at the true value $\alpha = (\hat{\beta}_h^c, \alpha_h)$, while other entries $\hat{\beta}_h^c$ are the same as MLE. R is the number of trials. α_h is the true value of coupling filter coefficient. We have the following,

$$\begin{aligned}0 = &\frac{1}{\sqrt{R}}\frac{\partial L}{\partial \beta}\bigg|_{\beta=\alpha} + \frac{1}{R}\,\frac{\partial^2 L}{\partial\beta\partial\beta^T}\bigg|_{\beta=\alpha} \cdot \sqrt{R}(\hat{\beta}-\alpha)\\ &+ \sqrt{R}(\hat{\beta}-\alpha)^T\left[\frac{1}{R}\int_0^T H(t)\mathrm{d}t - \frac{1}{R}\int_0^T G(t)N_j(\mathrm{d}t)\right](\hat{\beta}-\alpha)\end{aligned}$$

The last term is the reminder as in Taylor's theorem in the Lagrange form. H, G are the third-order derivative of the intensity function and of the log-intensity, which are assumed to be bounded by M, see [106, theorem 5, condition C4]. H, G are evaluated at some value between $\hat{\beta}$ and α. With the assumption that $\mathbb{E}|\hat{\beta}_h - \alpha_h|^2 = o(1/\sqrt{R})$. $I(\alpha)$ is the Fisher information.

$$\frac{1}{R}\cdot\frac{\partial^2 L(\beta)}{\partial\beta\partial\beta^T}\bigg|_{\beta=\alpha} \xrightarrow{p} \mathrm{E}\left[\frac{\partial^2 L}{\partial\beta\partial\beta^T}\right] = I(\alpha)$$

$$\left\|\frac{1}{\sqrt{R}}\frac{\partial L}{\partial\beta} - I(\alpha)\sqrt{R}(\hat{\beta}-\alpha)\right\| \le M\sqrt{R}\|(\hat{\beta}-\alpha)\|^2$$

as $R \to \infty$. As we have iid trials,

$$\sqrt{R}(\hat{\beta}-\alpha) \sim N\left(\mathbb{E}\left[\frac{1}{\sqrt{R}}\frac{\partial L}{\partial\beta}\right], \frac{1}{R}I(\alpha)^{-1}\mathrm{Cov}\left[\frac{\partial L}{\partial\beta}\right]I(\alpha)^{-1}\right)$$

As assumed $\mathbb{E}|\hat{\beta}_h - \alpha_h|^2 = o(1/\sqrt{R})$ and $\hat{\beta}_h$ is bounded, so $\hat{\beta}_h - \alpha_h \xrightarrow{p} 0$, $\frac{\partial L}{\partial\beta}$ is a continuous function of β,

$$\frac{\partial L}{\partial\beta}\bigg|_{\beta=\alpha} = -\int_0^T \frac{\Psi(s)}{\tilde{\lambda}_j(s)}\mathrm{d}N_j(s) + \int_0^T \Psi(s)\mathrm{d}s \xrightarrow{p} \frac{\partial L}{\partial\beta}\bigg|_{\beta=\hat{\beta}} = 0$$

so $\mathbb{E}\left[\frac{1}{\sqrt{R}}\frac{\partial L}{\partial\beta}\right] \to 0$. For the covariance we have,

$$\begin{aligned}\mathbb{E}\left[\frac{\partial L}{\partial\beta}\frac{\partial L}{\partial\beta^T}\right] =&\mathbb{E}\left[\int_0^T\int_0^T \Psi(u)\Psi^T(v)\left(\mathrm{d}u\mathrm{d}v - \mathrm{d}u\frac{N_j(\mathrm{d}v)}{\tilde{\lambda}_j(v)} - \mathrm{d}v\frac{N_j(\mathrm{d}u)}{\tilde{\lambda}_j(u)} + \frac{N_j(\mathrm{d}u)N_j(\mathrm{d}v)}{\tilde{\lambda}_j(u)\tilde{\lambda}_j(v)}\right)\right]\\ =&\mathbb{E}\left[\iint_{u\neq v} + \iint_{u=v}\right], \quad \text{if } u=v,\ N_j(\mathrm{d}u)N_j(\mathrm{d}v) = N_j(\mathrm{d}u)\\ =&\left(-\int_0^T \Psi(u)\frac{\lambda_j(u)}{\tilde{\lambda}_j(u)}\mathrm{d}u + \Psi(u)\mathrm{d}u\right)\left(-\int_0^T \Psi(v)\frac{\lambda_j(v)}{\tilde{\lambda}_j(v)}\mathrm{d}v + \Psi(v)\mathrm{d}v\right)^T\\ &+\int_0^T \frac{\Psi(u)\Psi(u)^T}{\tilde{\lambda}_j(u)^2}\lambda_j(u)\mathrm{d}u \to \mathbb{E}\left[\frac{\partial^2 L(\beta)}{\partial\beta\partial\beta^T}\right]\end{aligned}$$

Finally we have,

$$\operatorname{Cov}\left[\frac{\partial L}{\partial \beta}\right] \to \mathbb{E}\left[\frac{\partial^2 L(\beta)}{\partial \beta \partial \beta^T}\right]$$

Note L represents the negative log-likelihood here. Taking this into the Normal distribution of the estimator leads to the conclusion.

□

Remark The proof procedure and conditions are different from some previous works, such as [106], because the model we propose is not necessarily in the same parametric family of the true model. In some situations as shown in Lemma D.2.2, if the smoothing kernel widht is not selected properly, the bias can be nonzero, so the MLE is not always consistent, which is an important difference from the theory in [106]. This is why we add conditions such as $\mathbb{E}|\hat{\beta}_h - \alpha_h|^2 = o\left(\frac{1}{\sqrt{R}}\right)$. With these conditions, the MLE can still get the good properties like typical ones. These conditions are feasible. As shown in Fig 5.3, the bias is able to be zero. We assume the samples are collected by iid trials, so the variance convergence in the rate of $O(1/R)$. We verify the asymptotic normality using simulations in Supplementary D.4.7. Numerical simulations shows that even the bias is non-negligible, the asymptotic Normality still holds.

Another difference is the way the data is aggregated. From practical point of view, especially in neuroscience data collection, data are collected session by session not through one continuous recording. This affects how the central limited theorem is applied. Our theory uses the basic version which assume data samples are iid. But some works use different versions of central limit theorem assuming time segments have weak temporal dependency and it decays quickly [29, 106].

Lemma D.2.6. *If the model follows Eq* (5.6) *and Eq* (5.7)*, then the inner products defined in Eq* (D.15) *can be derived in closed-form. Details are in the proof.*

Proof. Apply Lemma D.2.1, D.2.7, and D.2.8,

$$
\begin{aligned}
\frac{1}{T}S_{ww} \approx& \int_{\mathbb{R}} [W * W](s)\breve{c}_{N,2}(\mathrm{d}s)\mathrm{d}s \\
=& \int_{\mathbb{R}} \frac{1}{2\sigma_w\sqrt{\pi}} \exp\left\{-\frac{s^2}{4\sigma_w^2}\right\} \left(\frac{\rho}{2\sigma_I\sqrt{\pi}} \exp\left\{-\frac{s^2}{4\sigma_I^2}\right\} + \bar{\lambda}_i\delta(s)\right) \mathrm{d}s \\
=& \frac{\rho}{2\sqrt{\pi}\sqrt{\sigma_w^2+\sigma_I^2}} + \frac{\bar{\lambda}_i}{2\sqrt{\pi}\sigma_w} \\
\frac{1}{T}S_{hh} \approx& \int_{\mathbb{R}} [h * h^-](s)\breve{c}_{N,2}(\mathrm{d}s)\mathrm{d}s \\
=& \int_{\mathbb{R}} \left[\mathrm{rect}\left(\frac{u}{\sigma_h}-\frac{1}{2}\right) * \mathrm{rect}\left(-\frac{u}{\sigma_h}-\frac{1}{2}\right)\right](s) \left(\frac{\rho}{2\sigma_I\sqrt{\pi}} \exp\left\{-\frac{s^2}{4\sigma_I^2}\right\} + \bar{\lambda}_i\delta(s)\right) \mathrm{d}s \\
=& \rho\left[\sigma_h \mathrm{erf}\left(\frac{\sigma_h}{2\sigma_I}\right) - \frac{2\sigma_I}{\sqrt{\pi}}\left(1-\exp\left\{\frac{\sigma_h^2}{4\sigma_I^2}\right\}\right)\right] + \bar{\lambda}_i\sigma_h \\
\frac{1}{T}S_{hw} \approx& \int_{\mathbb{R}} [h * W](s)\breve{c}_{N,2}(\mathrm{d}s)\mathrm{d}s \\
=& \int_{\mathbb{R}} \left[\mathrm{rect}\left(\frac{u}{\sigma_h}-\frac{1}{2}\right) * \phi_{\sigma_W}(u)\right](s) \left(\frac{\rho}{2\sigma_I\sqrt{\pi}} \exp\left\{-\frac{s^2}{4\sigma_I^2}\right\} + \bar{\lambda}_i\delta(s)\right) \mathrm{d}s \\
=& \frac{\rho}{2}\mathrm{erf}\left(\frac{\sigma_h}{\sqrt{2\sigma_w^2+4\sigma_I^2}}\right) + \frac{\bar{\lambda}_i}{2}\mathrm{erf}\left(\frac{\sigma_h}{\sqrt{2}\sigma_w}\right) \\
\frac{1}{T}S_{w\lambda} \approx& \int_{\mathbb{R}} W(s)\breve{c}_{N,\lambda,2}(s)\mathrm{s} \\
=& \int_{\mathbb{R}} \frac{1}{\sigma_w\sqrt{2\pi}} \exp\left\{-\frac{s^2}{2\sigma_w^2}\right\} \left(\frac{\rho}{2\sigma_I\sqrt{\pi}} \exp\left\{-\frac{s^2}{4\sigma_I^2}\right\}\right) \mathrm{d}s \\
=& \frac{\rho}{\sqrt{2\sigma_w^2+4\sigma_I^2}\cdot\sqrt{\pi}} \\
\frac{1}{T}S_{h\lambda} \approx& \int_{\mathbb{R}} h(s)\breve{c}_{N\Lambda,2}(s)\mathrm{s} \\
=& \int_{\mathbb{R}} \mathbb{I}_{[0,\sigma_h]}(s) \left(\frac{\rho}{2\sigma_I\sqrt{\pi}} \exp\left\{-\frac{s^2}{4\sigma_I^2}\right\}\right) \mathrm{d}s \\
=& \frac{\rho}{2}\mathrm{erf}\left(\frac{\sigma_h}{2\sigma_I}\right)
\end{aligned}
$$

□

Lemma D.2.7. *Assume the point process $N_i(\cdot)$ is second-order stationary (Lemma D.2.1) with reduced second-order moment measure intensity $\breve{c}_{N,2}$, intensity function λ_i, and mean intensity*

$\bar{\lambda}_i$. *Define two mean subtracted processes,*

$$\varphi_1(s) := \int W_1(s-t)(N_i(\mathrm{d}t) - \bar{\lambda}_i \mathrm{d}t)$$
$$\varphi_2(s) := \int W_2(s-t)(N_i(\mathrm{d}t) - \bar{\lambda}_i \mathrm{d}t)$$

Then the inner product on interval $[0, T]$ *between the processes is*

$$\langle \varphi_1, \varphi_2 \rangle \approx T \int_{\mathbb{R}} [W_1 * W_2^-](r) \breve{c}_{N,2}(r) \mathrm{d}r \tag{D.32}$$

$W_2^-(x) := W_2(-x)$.

$$\langle \varphi_w, \lambda_i \rangle \approx \int_{\mathbb{R}} W(r) \breve{c}_{N\Lambda,2}(r) \mathrm{d}s \mathrm{d}t \tag{D.33}$$

$\breve{c}_{N\Lambda,2}$ *is the reduced second-order moment measure intensity between two random measure* $N_i(\cdot)$ *and* $\Lambda_i(\cdot)$. $\Lambda_i(A) := \int_A \lambda_i(t) \mathrm{d}t$ *is the intensity measure.*

Proof.

$$\begin{aligned}
\langle \varphi_1, \varphi_2 \rangle &= \int_0^T \int_0^T \int_0^T \underbrace{W_1(t-u)}_{s:=t-u} \underbrace{W_2(t-v)}_{W_2^-(x):=W_2(-x)} \left(N_i(\mathrm{d}u) - \bar{\lambda}_i \mathrm{d}u\right) \left(N_i(\mathrm{d}v) - \bar{\lambda}_i \mathrm{d}v\right) \mathrm{d}t \\
&= \int_0^T \int_0^T \int_{-u}^{T-u} W_1(s) \underbrace{W_2^-((v-u)-s)}_{u:=u, r:v-u} \mathrm{d}s \left(N_i(\mathrm{d}u) - \bar{\lambda}_i \mathrm{d}u\right) \left(N_i(\mathrm{d}v) - \bar{\lambda}_i \mathrm{d}v\right) \\
&= \int_0^T \int_{-u}^{T-u} \int_{-u}^{T-u} W_1(s) W_2^-(r-s) \mathrm{d}s \cdot \breve{c}_{N,2}(r) \mathrm{d}r \cdot \mathrm{d}u \\
&\approx \int_0^T \int_{-u}^{T-u} [W_1 * W_2^-](r) \breve{c}_{N,2}(r) \mathrm{d}r \cdot \mathrm{d}u \approx T \int_{\mathbb{R}} [W_1 * W_2^-](r) \breve{c}_{N,2}(r) \mathrm{d}r
\end{aligned}$$

The approximation error comes from the boundary effect. If the kernels W_1, W_2 decays fast, the error can be ignored.

$$\begin{aligned}
\langle \varphi_w, \lambda_i \rangle &= \int_0^T \int_0^T W(t-u) \left(N_i(\mathrm{d}u)) - \bar{\lambda}_i \mathrm{d}u\right) \lambda_i(t) \mathrm{d}t \\
&= \int_0^T \int_0^T W(t-u) \left(N_i(\mathrm{d}u)) - \bar{\lambda}_i \mathrm{d}u\right) \left(\lambda_i(t) \mathrm{d}t - \bar{\lambda}_i \mathrm{d}t\right) \\
&\approx T \int_{\mathbb{R}} W(r) \breve{c}_{N\Lambda,2}(r) \mathrm{d}s \mathrm{d}t
\end{aligned}$$

□

Remark Many works that study the second-order stationary point process is in the frequency-domain [10, 19, 20, 21, 64, 97, 104] and [34, ch. 8]. All of our analysis is in time-domain. If we apply the Parseval's theorem to Eq (D.32), it equivalently shifts almost all results into frequency-domain.

$$\int_{\mathbb{R}} [W_1 * W_2^-](r)\breve{c}_{N,2}(r)\mathrm{d}r = \int_{\mathbb{R}} \widehat{W_1}(f) \cdot \widehat{W_2^-}(f) \cdot \Gamma_{N,2}(\mathrm{d}f)$$

where $\widehat{W_1}, \widehat{W_2^-}$ are the spectrum of kernels, and $\Gamma_{N,2}$ is called *Bartlett spectrum* for point process or *Bochner spectrum* for wide-sense process (see [34, ch. 8] and [19]). This can shift the time-domain analysis into the frequency-domain. This work does not include any frequency properties of the estimator, but it is promising to interpret some steps using the Bartlett spectrum measure in the future work.

Lemma D.2.8. *Consider the cluster process in Eq* (5.6)*. Let $\phi(\cdot)_{\sigma_I}$ be a window function with scale σ_I, t_i^c be the points of the center process which is generated by homogeneous Poisson process with intensity ρ. α_i is the baseline. The intensity function has form,*

$$\lambda_i(t) = \alpha_i + \sum_i \phi_{\sigma_I}\left(t - t_i^c\right) \tag{D.34}$$

$\Lambda_i(\cdot) := \int_A \lambda_i(t)\mathrm{d}t$ *is the intensity measure with respect to the intensity λ_i. $N_i(\cdot)$ is the corresponding count measure. Assume ϕ_{σ_I} is a Normal window with mean zero and standard deviation σ_I. The reduced covariance measure intensity of $\Lambda_i(t)$ is,*

$$\breve{c}_{\Lambda,2}(u) = \rho \cdot [\phi_{\sigma_I} * \phi_{\sigma_I}](u) = \frac{\rho}{\sqrt{4\pi\sigma_I^2}} \exp\left\{-\frac{u^2}{4\sigma_I^2}\right\} \tag{D.35}$$

Similarly, the reduced covariance measure intensity the point process $N_i(t)$ is,

$$\breve{c}_{N,2}(u) = \rho \cdot [\phi_{\sigma_I} * \phi_{\sigma_I}](u) + \bar{\lambda}_i\delta(u) = \frac{\rho}{\sqrt{4\pi\sigma_I^2}} \exp\left\{-\frac{u^2}{4\sigma_I^2}\right\} + \bar{\lambda}_i\delta(u) \tag{D.36}$$

$$\breve{c}_{N\Lambda,2}(u) = \rho \cdot [\phi_{\sigma_I} * \phi_{\sigma_I}](u) = \frac{\rho}{\sqrt{4\pi\sigma_I^2}} \exp\left\{-\frac{u^2}{4\sigma_I^2}\right\} \tag{D.37}$$

Proof. The first-moment property of the intensity is,

$$\bar{\lambda}_i = \mathbb{E}[\lambda(t)] = \mathbb{E}\left[\int_0^\infty \phi_{\sigma_I}\left(t - s\right) N(\mathrm{d}s)\right] = \int \phi_{\sigma_I}\left(t - s\right)(\alpha_i + \rho)\mathrm{d}s = \alpha_i + \rho$$

The *reduced covariance* for the second-moment stationary process is defined as,

$$\breve{c}_{\Lambda,2}(u) = \mathbb{E}[\lambda_i(x)\lambda_i(x+u)] - \mathbb{E}[\lambda_i(x)]\mathbb{E}[\lambda_i(x+u)] = \mathbb{E}[\lambda_i(x)\lambda_i(x+u)] - \bar{\lambda}_i^2$$

The second-moment measure of homogeneous Poisson process is [64],

$$\breve{M}_{N,2}^c(\mathrm{d}v) = \bar{\lambda}_i\delta(v)\mathrm{d}v + \bar{\lambda}_i^2\mathrm{d}v$$

The second equation holds due to the Campbell lemma [93, Lemma 1.1]. $N^c(\cdot)$ is the count measure of the center process.

$$\begin{aligned}
&\breve{m}_{\Lambda,2}(u) = \mathbb{E}\left[\frac{\Lambda_i(\mathrm{d}x)\Lambda_i(x+\mathrm{d}u)}{\mathrm{d}x\mathrm{d}u}\right] = \mathbb{E}[\lambda_i(x)\lambda_i(x+u)] \\
=&\mathbb{E}[(\alpha_i + f_{i,j}(x))(\alpha_i + f_{i,j}(x+u))] = \mathbb{E}[f_{i,j}(x)f_{i,j}(x+u)] + 2\rho\alpha_i + \alpha_i^2 \\
=&\mathbb{E}\left[\left(\int_{\mathbb{R}} \phi_{\sigma_I}(x-s)\,\mathrm{d}N^c(s)\right)\left(\int_{\mathbb{R}} \phi_{\sigma_I}(x+u-r)\,\mathrm{d}N^c(r)\right)\right] + 2\rho\alpha_i + \alpha_i^2 \\
=&\mathbb{E}\left[\int_{\mathbb{R}}\int_{\mathbb{R}} \phi_{\sigma_I}(x-s)\,\phi_{\sigma_I}(x+u-r)\,\mathrm{d}N^c(s)\mathrm{d}N^c(r)\right] + 2\rho\alpha_i + \alpha_i^2 \\
=&\int_{\mathbb{R}} \mathrm{d}s \int_{\mathbb{R}} \phi_{\sigma_I}(x-s)\,\phi_{\sigma_I}(x+u-(s+v))\,\breve{M}^c_{N,2}(\mathrm{d}v) + 2\rho\alpha_i + \alpha_i^2 \\
=&\bar{\lambda}_i^2 + \rho\int_{\mathbb{R}} \phi_{\sigma_I}(s)\,\phi_{\sigma_I}(u-s)\,\mathrm{d}s = \bar{\lambda}_i^2 + \rho[\phi_{\sigma_I} * \phi_{\sigma_I}](u)
\end{aligned}$$

The reduced covariance measure intensity of the count measure can be derived as follows.

$$\begin{aligned}
&M_{N,2}(\mathrm{d}t \times (t+\mathrm{d}u)) = \mathrm{d}t \cdot \breve{M}_{N,2}(\mathrm{d}u) \\
=&\mathbb{E}\left[N_i(\mathrm{d}t)N_i(t+\mathrm{d}u)\right] = \mathbb{E}_\Lambda\left[\mathbb{E}_N\left[N(\mathrm{d}t)N(t+\mathrm{d}u)|\Lambda_i\right]\right] \\
=&\bar{\lambda}_i\delta(u)\mathrm{d}u + \mathbb{E}_\lambda\left[\Lambda_i(\mathrm{d}t)\Lambda_i(t+\mathrm{d}u)\right] = \bar{\lambda}_i\delta(u)\mathrm{d}u\mathrm{d}t + \breve{m}_{\Lambda,2}(u)\mathrm{d}u\mathrm{d}t
\end{aligned}$$

So we have,

$$\begin{aligned}
\breve{m}_{N,2}(u) &= \bar{\lambda}_i\delta(u) + \breve{m}_{\Lambda,2}(u) \\
\breve{c}_{N,2}(u) &= \bar{\lambda}_i\delta(u) + \breve{c}_{\Lambda,2}(u)
\end{aligned}$$

Similarly, the reduced second-order covariance intensity is,

$$\begin{aligned}
&M_{N\Lambda,2}(\mathrm{d}t \times (t+\mathrm{d}u)) = \mathrm{d}t \cdot \breve{M}_{N\Lambda,2}(\mathrm{d}u) \\
=&\mathbb{E}\left[N(\mathrm{d}t)\Lambda(t+\mathrm{d}u)\right] = \mathbb{E}_\Lambda\left[\mathbb{E}_N\left[N(\mathrm{d}t)\Lambda(t+\mathrm{d}u)|\lambda\right]\right] \\
=&\mathbb{E}_\Lambda\left[\Lambda(\mathrm{d}t)\Lambda(t+\mathrm{d}u)\right] = \breve{m}_{\Lambda,2}(u)\mathrm{d}u\mathrm{d}t
\end{aligned}$$

So we have,

$$\begin{aligned}
\breve{m}_{N\Lambda,2}(u) &= \breve{m}_{\Lambda,2}(u) \\
\breve{c}_{N\Lambda,2}(u) &= \breve{c}_{\Lambda,2}(u)
\end{aligned}$$

□

D.3 Application to neuroscience dataset

D.3.1 Materials

We applied our method to the Allen Brain Observatory Visual Coding Neuropixels [121]. It uses multiple high-density extracellular electrophysiology probes to simultaneously record spiking activity from many areas in the mouse brain, especially the visual cortex. The animals were passively presented with visual stimuli while the head was fixed. The details of the experimental setup can be found in [121]. Our work uses drifting gratings as it includes many repeated trials, the trials are long and stimuli strongly elicit neural responses. The drifting gratings have 40 conditions which are combinations of 8 different orientations (0°, 45°, 90°, 135°,180°, 225°, 270°, 315°, clockwise from 0° = right-to-left) and 5 different temporal frequencies (1, 2, 4, 8, 15 Hz). The spatial frequency is 0.04 cycles/deg and the contrast is 80% for all trials. One condition has 15 repeated trials. A trial lasts for 3 sec with 2 sec stimulus and 1 sec blank screen. The sequence of the conditions is randomly arranged. The baseline condition has 30 trials with a grey screen. The number of neurons in visual cortical areas recorded by one probe ranges, roughly, from 40 to 100. Usually, 6 probes are recorded at the same time. The dataset assigns unique identities for all properties, such as conditions, trials, neurons, etc. In this paper, we refer to those identities directly. We analyzed mouse session 798911424 using randomly selected 19 drifting gratings conditions, totally $19 \times 15 = 285$ trials. The time window of each trial is $[400, 2000]$ ms dropping the initial part of the trial after the activity gets more stable. The conditions are: 275, 246, 268, 270, 284, 274, 249, 263, 265, 261, 286, 258, 278, 267, 280, 256, 260, 257, 281. We selected top 30% most active neurons (thresholded by the mean firing rate) including 28 V1 neuron, 24 LM neurons, 27 AL neurons.

D.3.2 Details of Fig 5.1

The real data is composed of spike trains of two neurons, $i =$951102476 and $j =$951109307. We considered the coupling effects $i \to j$ (i leads j). The trials include 3819, 3828, 3859, 3882, 3886, 3906, 3912, 3917, 3922, 3924, 3925, 3930, 3932, 3942, 3946, 3948, 3951, 3953, 3959, 3966, 3980, 3995, 31020, 31033, 31035, 31040, 31048, 31054, 31055, 31114, 31129, 31152, 31161, 31170, 31173, 31174, 31177, 31179, 31182, 31186, 31190, 31194, 49209, 49211, 49220, 49262, 49290, 49301, 49304, 49305, 49327, 49353, 49378, 49390, 49418, 49420. We selected these trials as they were identified as “inhibitory” type coupling effect using our algorithm. In Fig 5.1A, the jitter window width is 120 ms. Smaller or larger jitter window width yields similar results. The time bin for the spike train is 2 ms. The pointwise CI or the simultaneous CI were calculated using 1000 surrogate jitter spike trains. More details of the jitter-based CCG can be found in [3]. In Fig 5.1V, the coupling filter was estimated using the B-splines with 6 evenly spaced knots (9 spline bases). The smoothing kernel width is $\sigma_w = 60$ ms.

For the simulation data, the spike trains are generated using the model in Eq 5.6 and Eq 5.7. The activity $f_{i,j}$ in the true model is set as a cluster process in Eq (5.6) with $\sigma_I = 40$ ms. The firing rate of the center process $\rho = 12$ spikes/sec. The baselines are $\alpha_j = \alpha_i = 1$ spikes/sec. The square window filter width is $\sigma_h = 50$ ms and $\alpha_h = -5$ spikes/sec in Eq (5.7). The simulation has 56 trials and the length of the trial is 2 sec, which is the same sample size as the real data in Fig 5.1A,B,C. Different trials are assigned with a different randomly generated $f_{i,j}$. The coupling filter is estimated using B-splines with 5 evenly distributed knots (8 spline bases).

D.3.3 Algorithm for filter type identification

The model is estimated using hard EM algorithm, that alternatively updates the coupling filter templates as shown in Fig 5.5 and coupling filter type of each coupling filter on each trial.

Update coupling filter templates. $Z_{r,i\to j}$ is the coupling filter type of neurons $i \to j$ on trial r, which is a categorical variable. h_I is the coupling filter templates, $I = \{0, 1, 2, 3, 4\}$ with respect to "no effect", "excitatory", "inhibitory", and two oscillatory types. We put all trials of the same type from all neurons together to estimate the coupling filter template.

$$h_I = \arg\min_h \left\{ \min_{\beta_j,\beta_w} \left\{ \sum_{Z_{r,i\to j}=I} \left(-\sum_{s\in N_j} \log \tilde{\lambda}_{r,i\to j}(s) + \int_0^T \tilde{\lambda}_{r,i\to j}(s)\mathrm{d}s \right) \right\} \right\},\ I = 1,2,3,4 \tag{D.38}$$

The template for "no effect" (I=0) is fixed as 0. $\tilde{\lambda}_{r,i\to j}$ is constructed similar to Eq 5.4, which is a linear function of β_j, β_w, h. For type 1 and 2, we estimate the coupling filters using a square window with width 50 ms similar to Eq 5.7; for type 3 and 4, we use B-spline with 8 evenly distributed knots. The fitted templates are in Fig 5.5.

Update coupling filter types. H represents all templates, N_i, N_j represents all spike train data.

$$\begin{aligned}
Z_{r,i\to j} =& \arg\max_Z \Big\{ p(Z|H, N_i, N_j) \Big\} \\
=& \arg\max_Z \left\{ \int p(Z|\beta_j, \beta_w, H, N_i, N_j) \cdot p(\beta_j, \beta_w)\mathrm{d}\beta_j \mathrm{d}\beta_w \right\} \\
\approx& \arg\max_Z \left\{ \max_{\beta_j,\beta_w} \Big\{ p(Z|\beta_j, \beta_w, H, N_i, N_j) \Big\} \right\} \\
=& \arg\max_Z \left\{ \max_{\beta_j,\beta_w} \Big\{ p(N_j|\beta_j, \beta_w, h_Z, N_i) \cdot p(Z) \Big\} \right\} \\
=& \arg\min_Z \left\{ \min_{\beta_j,\beta_w} \left\{ \left(-\sum_{s\in N_j} \log \tilde{\lambda}_{r,i\to j}(s) + \int_0^T \tilde{\lambda}_{r,i\to j}(s)\mathrm{d}s \right) - \log p(Z) \right\} \right\}
\end{aligned} \tag{D.39}$$

The third row marginalizes the nuisance parameters using the approximation through BIC [39]. $\tilde{\lambda}_{r,i\to j}$ is constructed similar to Eq 5.4, which is a linear function of β_j, β_w, h. $p(Z),\ Z = 0, 1, 2, 3, 4$ is the weight of the coupling filter type. The optimization over the negative log-likelihood can be calculated using Newton's method as shown in Appendix D.1. The model converges quickly. The updating terminates if the coefficients of the coupling filter templates and the group sizes change very small. In the first updating rule, conditioning on the spike trains and the rest variables, the coupling filters for each type can be calculated independently. In the second updating rule, the coupling filter type of each pair of neurons and each trial can be calculated independently. So the algorithm can be highly implemented in parallel to improve the performance.

A standard mixture model updates the group weights in every iteration. But in our situation, the clusters of types 0,1,2 are very close since the coupling effect is not very strong, they might become unbalanced or get merged because the distributions of these groups are not well isolated with a clear boundary. To overcome this issue, we fix the weights $p(Z)$ as a constant, which is reduced to a clustering method similar to k-means. As the weights do not change from trial to trial, it will not introduce the trial-to-trial variation. To verify whether the conclusion is sensitive to this modification, we present more results in Supplementary D.4.14 with different uneven weights. The number of significant outcomes is slightly different, but the patterns are similar.

For the initialization, we started with exploring the dataset in the following ways,

1. The whole dataset is too large. We selected the most active 30% of neurons determined by the spike count of all trials. Then randomly selected some pairs of neurons among V1→LM and LM→AL.

2. Run CCG on these pairs of spike trains. Strong coupling effects would first be picked out, just like the examples in Fig D.24type 3 and type 4. We grouped them into two clusters based on the CCG curves (run k-means on CCG curves). Then applied the point process regression method to get the coupling filter templates of type 3 and type 4. The CCG results also helped with determining the coupling filter length, where the end of the filter had small values.

3. The rest pairs had weak coupling effects. CCG was not powerful enough to distinguish them, and most parts of the curves stayed within the confident bands. But we found some curves were above zero, some were below, and some fluctuated evenly around zero, as shown in the examples in Fig 5.1 and Fig D.24. This motivated us to aggregate the data (along with the time lag instead of focusing at one lag) to detect the weak signal. Stepping back from estimating the detailed shapes of the curves, we simply chose square window coupling filters, one being positive, one being negative and one being zero. Their initial values are set to 3 spikes/sec, -3 spikes/sec and 0 spikes/sec. The length of the filter was roughly determined by checking CCG and running non-parametric fitting. As discussed in Supplementary D.4.6, we preferred to choose shorter filters than longer ones.

4. The smoothing kernel width of W was determined using plug-in estimator as shown in Appendix D.1.2 by sub-sampling 5000 trials. As described in Fig 5.4, the selection of the smoothing kernel width is invariant of coupling filter properties, it does not matter what the coupling filter type the trials have. We tried several times and got a similar value. (We later on came back to verify if the fitted templates changed too much if the kernel width was set to a smaller or a larger value). The kernel width used for the analysis in the main text is 60 ms.

5. Once we got the templates of the 5 groups, we could update the coupling filter types of all neurons on each trial. Then alternatively run the updating rules above.

D.4 Supplementary information

In this section, we will present more simulation study from section D.4.1 to section D.4.11 as summarized at the end of section 5.2. The topics include: Timescale-varying background; Non-shared fluctuating background; Fast-changing background; Bayesian model; Non-parametric fitting for the coupling filter; Selection of coupling filter length; Asymptotic Normality of the estimator; Hypothesis testing example; Background activity with Laplacian window function; Multivariate regression and partial relation; Self-coupling effect; Rate coupling and delayed shared input. We will also provide supplementary materials to the real data application, where we will show: More examples of coupling filters and jitter-based CCG; Clustering with unbalanced weights; Results with different smoothing kernel widths; Goodness-of-fit; Results using different dataset.

D.4.1 Timescale-varying background activity

The shared activity $f_{i,j}$ in Eq (5.6) is composed of a sequence of Gaussian windows with fixed scale σ_I. The locations of the windows are determined by a homogeneous Poisson process with intensity ρ. This simple random process is second-order stationary and many properties can be calculated in closed-form formula as shown in Lemma D.2.8. σ_I controls how fast the activity changes. If σ_I is smaller, the activity will change faster. This makes the theoretical derivations much simpler.

Here we design a similar process but the scale of the window σ_I is no longer a fixed value, where $\sigma_{I,i}$ changes in a continuous range. Every time point of the center process t_i^c is assigned with a different scale $\sigma_{I,i}$ randomly. The settings of this simulation scenario are the same as Fig 5.3 except that $\sigma_{I,i}$ is set by a uniform random variable between 80 ms and 140 ms. The process $f_{i,j}$ changes faster at smaller $\sigma_{I,i}$, and changes slower at larger $\sigma_{I,i}$.

$$f_{i,j} = \sum_i \phi_{\sigma_{I,i}} \left(t - t_i^c\right) \tag{D.40}$$

The true coupling filter is a square window $h_{i \to j}(t) = \alpha_h \cdot \mathbb{I}_{[0,\sigma_h]}(t)$, where the timescale is $\sigma_h = 30$ ms, the amplitude is $\alpha_h = 2$ spikes/sec.

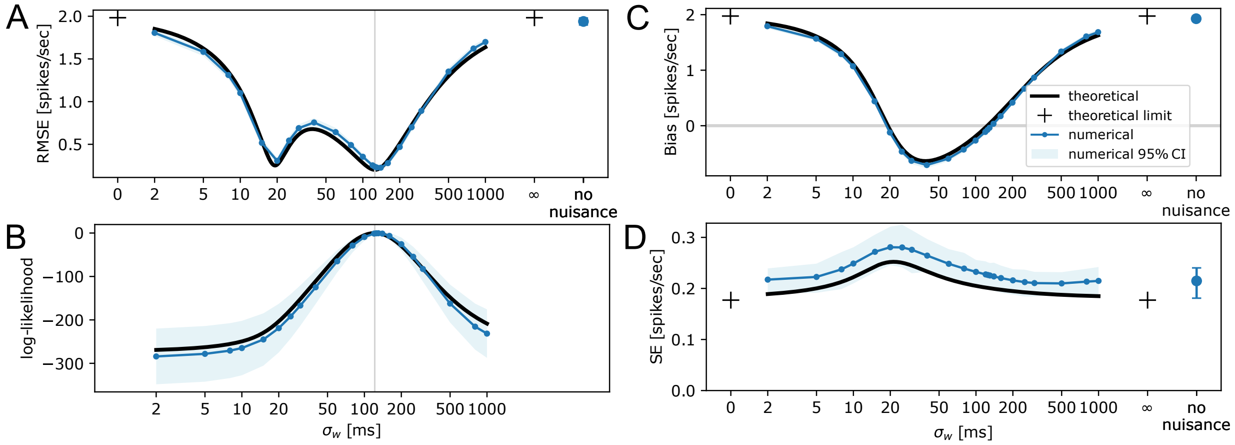


Figure D.1: **Simulation results of the coupling filter estimator with varying background activity timescale.** The figure is presented in the same ways as Fig 5.3. The simulation details are in the text. The shared activity $f_{i,j}$ in Eq (5.6) is replaced with Eq (D.40) with varying timescale. The results are similar to Fig 5.3. The dark curves show the equivalent theoretical approximation using the model in Eq 5.2 and Fig 5.2 with fixed timescale $\sigma_I = 100$ ms with manually tuned σ_I to match the numerical results.

As shown in Fig D.1, the selected kernel width σ_w will balance the varying timescale, and it can still the select estimator with small risk and low bias, indicated by the vertical lines in Fig D.1 A and B. Similar to Fig 5.3D, the SE does not change too much as the smoothing kernel width σ_w changes. The model can balance the bias, which can be explained by its properties in Fig 5.3C. If the timescale of the background σ_I is fixed and consider the bias of the estimator near the right root. If σ_w is larger than the right root, the bias will be positive, if σ_w is a little smaller than the right root, the bias will become negative. In this scenario, the timescale of the background σ_I varies. The optimal σ_w is relatively large for the activity with small σ_I, so the bias is positive for fast-changing part. The optimal σ_w is relatively small for the sessions with large σ_I, so the bias is negative for slow-changing part of the activity. With proper selection of σ_w, the estimator will balance the overall bias between fast- and slow-changing activities, and it can still achieve zero bias. Together with the SE, the risk properties remain similar in Fig D.1 A.

To verify the reasoning, we compare the numerical results with the equivalent theoretical approximation shown in the dark curves in Fig D.1. The theoretical method is for the model in Eq 5.2 and Fig 5.2 with fixed timescale for the background by manually tune the timescale as $\sigma_I = 100$ ms to match the numerical curves. The behavior of the estimator for the varying-timescale background activity is almost equivalent to the case with fixed-timescale background activity. The SE of the numerical results is slightly larger though.

Finally, we also want to point out that the model could fail if the shared activity is a mixture of very distinct timescale, for example when it is a compound of 100 ms and 5 ms. The shared fast-changing input can be the spike trains from a subpopulation. We will further discuss this problem in Supplementary D.4.10.

D.4.2 Non-shared fluctuating background

The baselines of the model in Fig 5.2 are constant α_i and α_j, and the only fluctuating background activity comes from the shared component $f_{i,j}$. In this section, we consider non-constant baselines $f_i(t)$ and $f_j(t)$ for each neuron exclusively. The diagram of the new model is in Fig D.2. The only differences from Fig 5.2 are the extra components $f_i(t), f_j(t)$. The settings of the simulation are also the same as the case in the main text Fig 5.3 except for $f_i(t)$, $f_j(t)$, and $f_{i,j}(t)$.

$$\begin{aligned} f_i(t) &= \sum_n \phi_{\sigma_I}\left(t - t_n^{i,c}\right) \\ f_j(t) &= \sum_m \phi_{\sigma_I}\left(t - t_m^{j,c}\right) \end{aligned} \tag{D.41}$$

$f_i(t)$ and $f_j(t)$ are created using independent cluster processes. The intensity of the center processes is 10 spikes/sec. $t_n^{i,c}, t_m^{j,c}$ are the time points of the center processes for $f_i(t), f_j(t)$ respectively. The intensity ρ of the center process for $f_{i,j}$ is adjusted from 30 spikes/sec to 20 spikes/sec so the overall mean firing rates of the neurons are similar to Fig 5.3. The window function ϕ_{σ_I} of $f_i(t), f_j(t)$ and $f_{i,j}(t)$ are Gaussian with the same timescale σ_I. The true coupling filter is a square window $h_{i\to j}(t) = \alpha_h \cdot \mathbb{I}_{[0,\sigma_h]}(t)$, where the timescale is $\sigma_h = 30$ ms, the amplitude is $\alpha_h = 2$ spikes/sec.

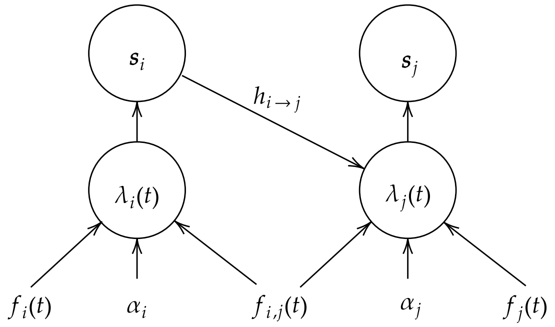


Figure D.2: **Diagram for the coupling neurons model with exclusive fluctuating baselines.** The diagram is the same as Fig 5.2 except for the baseline $f_i(t)$, $f_j(t)$ for neuron i and j separately. The random process $f_i(t), f_j(t)$ are also created using cluster process in Eq D.41. Details of the model are in the text.

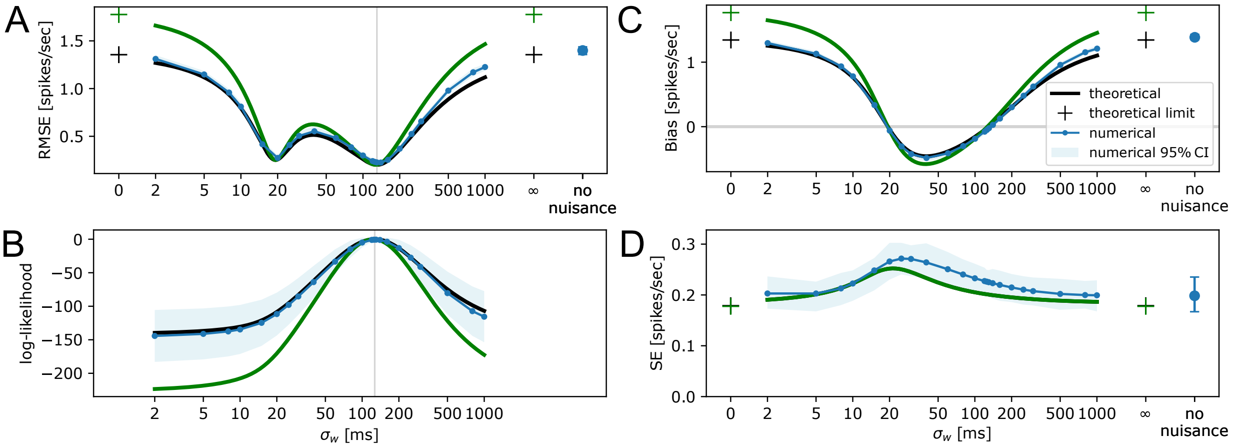


Figure D.3: **Simulation results of the scenario with exclusive fluctuating baselines.** The figure shows the simulation results of the model described in Fig D.2, including RMSE, bias, SE and log-likelihood in a similar way as Fig 5.3. The details of the simulation settings are in the text. The green curves are the theoretical approximation of the basic model in the main text (Fig 5.2 without exclusive components f_i or f_j), where the intensity of the center process for $f_{i,j}$ is $\rho = 20$ spikes/sec, same as the current simulation settings. The dark curves are similar theoretical approximations of the basic model in Fig 5.2, but the intensity of the background center process is $\rho = 10$ spikes/sec, which is lower than the intensity of this simulation scenario. The dark curves match the numerical curves better than the green ones.

The results are shown in Fig D.3. Similar to the scenario in Fig 5.3, the SE does not change a lot as the timescale of the smoothing kernel σ_w varies. The bias can still achieves zero. The optimal model with small risk can be selected by maximizing the likelihood indicated by the vertical lines in Fig D.16A and B.

Next, we compare the numerical results with the theoretical approximation of the basic model in the main text without the baselines f_i, f_j. If Fig D.3, the green curves correspond to the basic model with the same center process intensity $\rho = 20$ spikes/sec of $f_{i,j}$; In all plots in Fig D.3, the numerical results match the dark curves better. These comparisons imply that adding the non-shared fluctuating activity can actually help the estimator. It is equivalent to reducing the strength of the shared activity. The shared background activity contaminates the coupling effect estimation because it is correlated with the spike trains. Without considering the background, such correlation will be counted as the contribution of the coupling filter. Adding non-shared fluctuating activity can reduce the correlation between the background and the coupling effect so it improves the estimation.

D.4.3 Fast-changing background

In the main text, the shared activity changes slower than the coupling effects. We will show in this section, this is not a necessary assumption or constraint. Even the background changes faster than the coupling effect, the model still holds the ideal properties such as optimal model selection and low bias.

We consider two scenarios by setting the timescale of the $f_{i,j}$ in Eq 5.6 to $\sigma_I = 20$ ms (Fig D.4 A,B,C,D), and $\sigma_I = 5$ ms (Fig D.4 E,F,G,H). The rest settings of the simulation scenarios are the same as the main text in Fig 5.3, where the true coupling filter is a square window $h_{i\to j}(t) = \alpha_h \cdot \mathbb{I}_{[0,\sigma_h]}(t)$, the timescale is $\sigma_h = 30$ ms, the amplitude is $\alpha_h = 2$ spikes/sec.

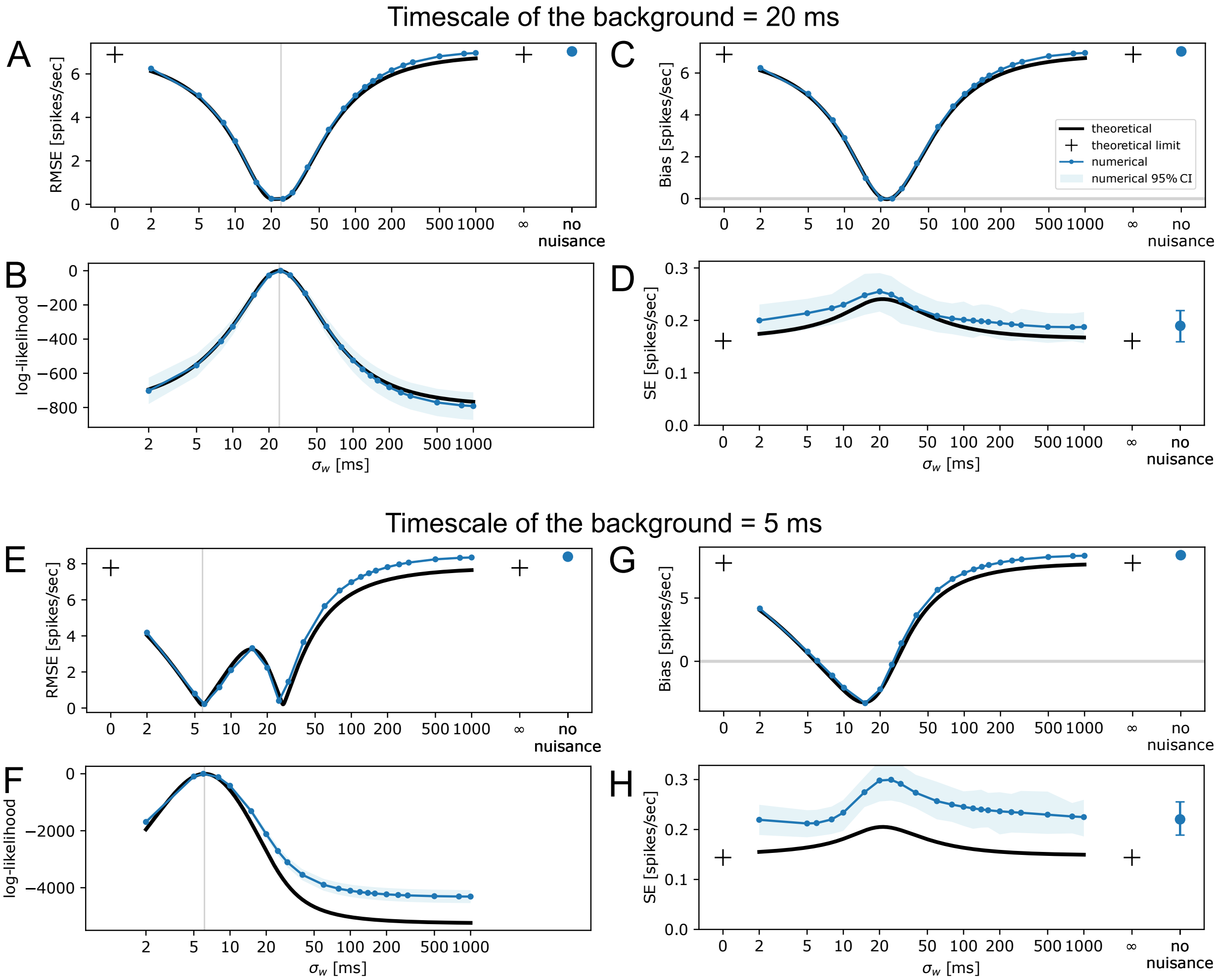


Figure D.4: **Fast-changing background with small σ_I.** We present the properties of the coupling filter estimator in the same way as Fig 5.3. The simulation settings are the same as Fig 5.3 except that in **A, B, C, D** $\sigma_I = 20$ ms, in **E, F, G, H** $\sigma_I = 5$ ms.

As already shown in the main Fig 5.4A, if σ_I decreases, the labeled "min-2" of the risk will shift toward left. Fig D.4A shows an extreme case that if σ_I keeps decreasing, two local minimums will merge to one minimum. In Fig D.4C, the bias property has a similar change; two roots in the main Fig 5.3C will merge to one root. The property of SE does not change too much, see Fig D.4D and 5.3D. If $\sigma_I < 20$ ms, the "min-2" point labeled in Fig 5.3 will move to the left, and it can still be selected by the likelihood function, see Fig D.4E and F. When σ_I has a very small value, such as 5 ms in Fig D.4, the theoretical approximation of the SE begins to have a large deviation from the numerical value. We think this is related to the large error of the quantity S_{hw} in Lemma D.2.6. Next, similar to Fig 5.4, we will explore how the timescale of the shared activity σ_I, the timescale of coupling effect σ_h, and the amplitude of coupling filter α_h are related to the above properties when σ_I is very small.

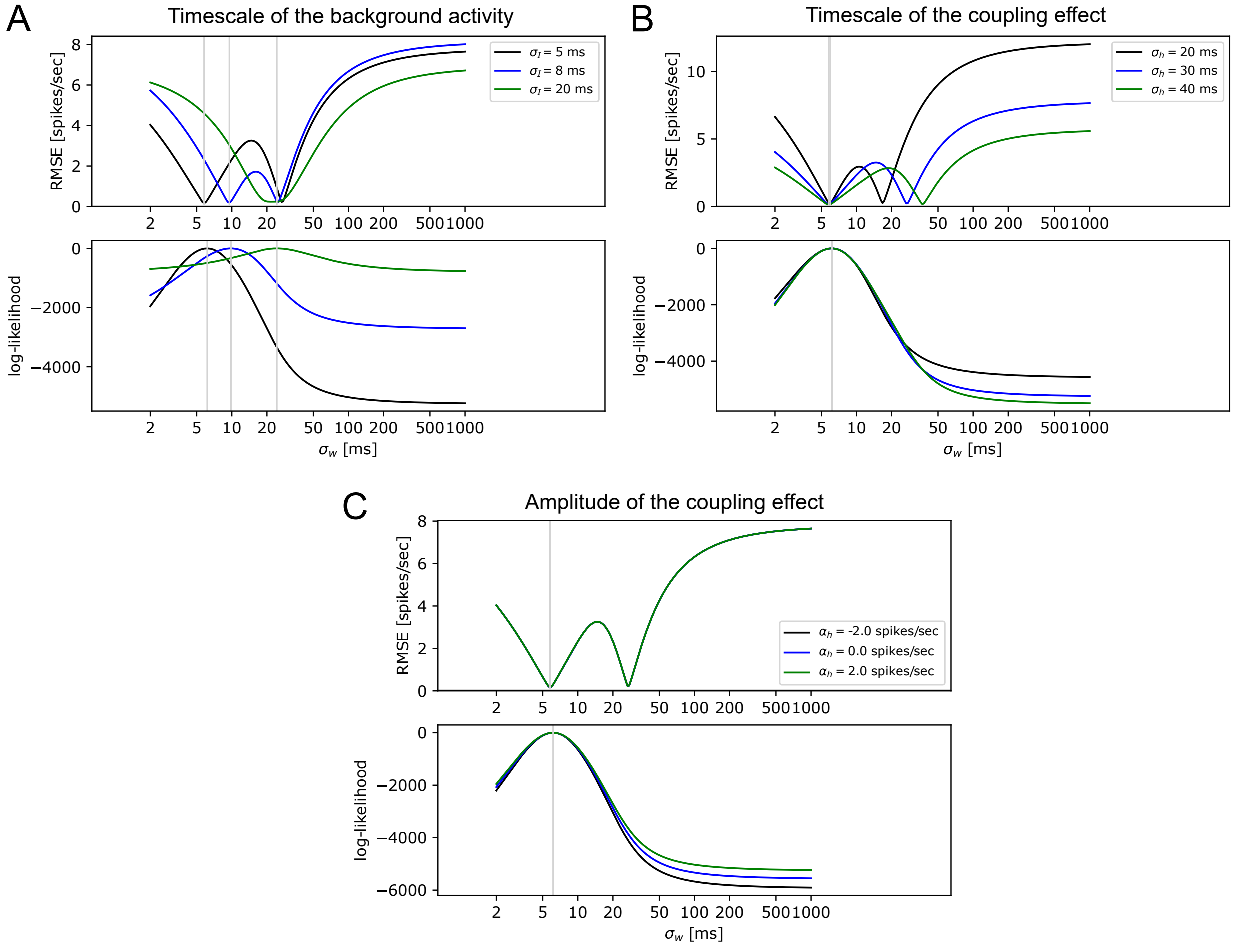


Figure D.5: **Properties of the estimator with fast-changing background.** This figure is analog to Fig 5.4 but the time scale σ_I of the shared activity $f_{i,j}$ is very small. The settings are the same as Fig 5.3, 5.4, and D.5 except for different σ_I in A, different σ_h in B and different α_h in C. We only show the theoretical RMSE and log-likelihood curves. The numerical results are very close. Some numerical results have been shown in Fig D.4. The log-likelihood functions may have different offsets, we align them by the peak to zero (maximum value across σ_w). **A** If σ_I is around 20 ms, two local minimum values of the risk curve may merge to one, which agrees with the numerical result in Fig D.4A. If $\sigma_I < 20$ ms, the selected optimal kernel width σ_w will be at the left local minimum of the risk. Notice that when σ is around 20 ms, the shape of the log-likelihood near the maximum value is more blunted than others. In these cases, the timescale of the coupling filter $\sigma_h = 50$ ms and the amplitude $\alpha_h = 2$ spikes/sec are fixed. **B** If σ_I ms is very small, it will be the right local minimum of the risk that is associated with the coupling filter timescale, which is opposite to Fig 5.4B. If the timescale of the coupling filter σ_h decreases, the right local minimum of the risk will move to the left. σ_h does not change the left local minimum or the maximum point of the likelihood. In these cases, $\sigma_I = 5$ ms and $\alpha_h = 2$ spikes/sec are fixed. **C** Similar to Fig 5.4C, the amplitude of the coupling filter α_h does not change the risk curve or the position of the maximum likelihood. In these cases, $\sigma_I = 5$ ms and $\sigma_h = 50$ spikes/sec are fixed.

Fig D.5 shows the estimator's properties when parameters σ_I, σ_h, and α_h are tuned in a similar way as Fig 5.4 in the main text, but the time scale σ_I of the shared activity $f_{i,j}$ here is a much smaller value. As already shown in Fig D.4, when σ_I is very small, the optimal smoothing kernel width σ_w will move to the left (indicated by the vertical line), and it still matches the maximum likelihood. Two local minimum values may merge to one when σ_I is around 20 ms. If $\sigma_I < 20$ ms, the selected optimal kernel width σ_w will be at the left local optimal. Notice that when σ_I is around 20 ms, the shape of the log-likelihood near the maximum value is blunter than others. So it may become more difficult to accurately choose the optimal model. In Fig 5.4B, the left local minimum of the risk labeled by "min-1" is associated with the coupling filter timescale σ_h. But if σ_I is very small (say $\sigma_I = 5$ ms in Fig D.5B), it will be the right local minimum of the risk that is associated with the coupling filter timescale. If the timescale of the coupling filter σ_h decreases, the right local minimum of the risk will move to the left. σ_h does not change the left local minimum and the maximum point of the likelihood. The amplitude of the coupling filter α_h does not change the risk curve or the position of the maximum likelihood similar to Fig 5.4. Notice that when σ_I is very small, the risk of the non-optimal model is much larger than the scenario in Fig 5.3 with large σ_I. The worst risk can go up to 12 spikes/sec (dark curve in Fig D.5B), while in Fig 5.3 the worst case is only about 2 spikes/sec.

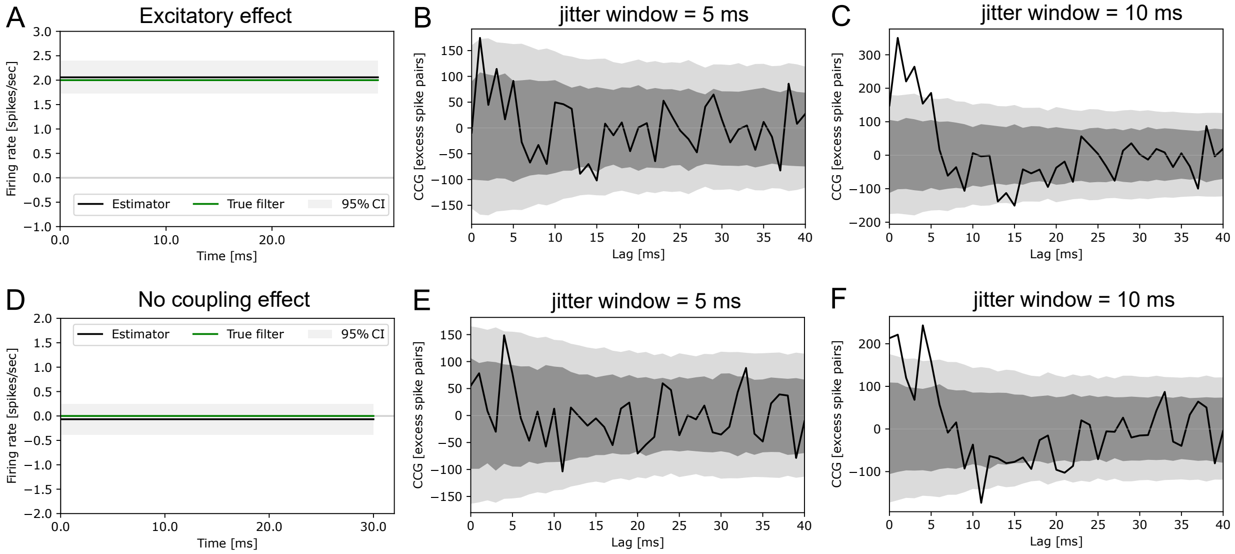


Figure D.6: **A comparison between the estimations of the coupling effect with fast-changing background.** The figure compares the performance of the estimators of the coupling effect. The simulation settings are the same as Fig 5.3 except that the timescale of the background $\sigma_I = 5$ ms is very small. The timescale of the coupling filter as in Eq (5.7) is $\sigma_h = 30$ ms. In A, B, and C, the amplitude of the true coupling filter is $\alpha_h = 2$ spikes/sec. In D, E, and F, the amplitude of the true coupling filter is $\alpha_h = 0$ spikes/sec. **A, D** The estimator of the point process regression. It can accurately estimate the true filter, which is supported by the analysis in Fig D.4 and D.5. **B, E** Jitter-based CCG. The time bin for the spike train is 1 ms. The jitter window width is set as 5 ms, close to the timescale of the background activity. The dark grey band is pointwise 95% CI, and the light grey band is simultaneous 95% CI. The result is acquired from 1000 surrogate jitter samples. The CCG method detects a small excitatory effect before lag = 5 ms no matter whether the neurons have true coupling effect. **C, F** Similar to B except that the jitter window width is 10 ms. In both B and C, the jitter-based CCG method can only detect a small effect before 5 ms lag or 7 ms lag. A large part of the coupling effect between 0 to 30 ms is buried under the CI band. However, such an effect is due to the fast-changing background, but not the neuron-to-neuron coupling effect.

A significant advantage of our model over the jitter-based model is that the proposed model does not assume the background activity changes slower than the coupling effect, and the model can automatically find the optimal timescale. The jitter-based method can not avoid such an assumption due to its nature of conditional inference. The null hypothesis states that the coupling effects do not change faster than the jitter window width. Thus the samples under the null distribution are obtained by randomly jittering the spikes within the jitter window. If the background changes as fast as the coupling effect, such bootstrapping method can not maintain the temporal structure of the background activity, so it can not split the background artifacts and the coupling effects. In other words, if the jitter window is set a little larger than the coupling effects, it can not tell whether the detected effect belongs to the background or the spike-to-spike interaction. Some other bootstrapping methods have the same issue for exactly the same reason [33]. Fig D.6 compares the point process regression method and jitter-based CCG method. The simulation scenario is the same as the basic model in Fig 5.3 except that the timescale of the background activity is very small $\sigma_I = 5$ ms. The numerical properties of the estimator have been shown in Fig D.4E-H. The true coupling filter is a square window $h_{i\to j}(t) = \alpha_h \cdot \mathbb{I}_{[0,\sigma_h]}(t)$. The timescale of the coupling effect is $\sigma_h = 30$ ms. The true amplitude of the coupling filter is $\alpha_h = 2$ spikes/sec in Fig D.6A,B,C, and $\alpha_h = 0$ spikes/sec in Fig D.6E,E,F. In both cases, the regression method can accurately estimate the true estimator, which agrees with the numerical and theoretical results in Fig D.4. In Fig D.6 B and C, the CCG method with jitter window width = 5 or 10 ms can detect some excitatory effect in a lag range smaller than 5 ms or 10 ms. Nevertheless, it misses the excitatory effect between lag=10 to 30 ms. It is unreasonable to use a larger jitter window, as it will not match the background timescale. The CCG results are similar to another example in Fig D.6E, F, where there is no coupling effect. The detected significant data points are totally due to the fast-changing background. So for the results in Fig D.6B and C, we can not conclude that the jitter method has removed the background artifacts and the significant effect is caused by the coupling effect.

D.4.4 Simple Bayesian model

The regression model Eq (5.2) is probabilistic, so it can be easily adopted for Bayesian inference. In this section, we present some simple Bayesian models where the scale σ_w of the smoothing kernel W in Eq (5.5) can be treated as a random variable. We want to investigate how incorporating the uncertainty of the smoothing kernel width affects the estimation of the coupling filter, and how the variance of the background timescale affects the uncertainty of the smoothing kernel width. The posterior of the coupling filter coefficients obtained using sampling-based method can also verify the Normality property in the regression method when the sample size dominates the prior.

We consider two Bayesian models below and the basic point process regression model in Eq 5.2. The likelihood of the model is same as Eq (5.2)-(5.5). The coupling filter is estimated using a square window, same as Eq (5.7). We choose non-informative flat priors for all the variables. As the sample size is large, the posterior does not heavily rely on the prior. Model 2 is similar to the regression model in Eq 5.2, where the kernel width σ_w is selected using the same way as the regression model and held as fixed. Model 2 and the regression model are expected to have similar results. In Model 1, σ_w is a random variable. We performed the estimation on two datasets: The first dataset is the same as the example in Fig 5.3 (details are in the main text); The second one is the same as the scenario in Supplementary D.4.1, where the timescale of the background activity σ_I randomly changes in a continuous range between 80 ms and 140 ms. The true coupling filter is a square window $h_{i\to j}(t) = \alpha_h \cdot \mathbb{I}_{[0,\sigma_h]}(t)$ where the amplitude of the coupling filter is $\alpha_h = 2$ spikes/sec. The timescale of the square window is $\sigma_h = 30$ ms. We used the Hastings-Metropolis method for the model inference, which was a Monte Carlo Markov Chain (MCMC) sampler. The posterior was acquired by drawing 1000 samples. The model was initialized using the basic point process regression method Eq (5.2). The basic regression model approximates the estimator's distribution using Normal distribution; the mean is the MLE $\hat{\beta}_h$, and the standard error is from the Fishier information. See the discussion of the asymptotic Normality in Lemma D.2.5.

Model 1:

$$\begin{aligned}\beta_j, \beta_w, \beta_h, \sigma_w &\propto 1\\ p(\beta_j, \beta_w, \beta_h, \sigma_w | \mathbf{s}_j, \mathbf{s}_i) &\propto p(\mathbf{s}_j | \mathbf{s}_i, \beta_j, \beta_w, \beta_h, \sigma_w)\end{aligned}$$

where $p(\mathbf{s}_j | \mathbf{s}_i, \beta_j, \beta_w, \beta_h, \sigma_w)$ is the likelihood function of the point process similar to Eq (5.3). σ_w is a variable of the model.

Model 2: σ_w is fixed and the parameter selection follows the regression method.

$$\begin{aligned}\beta_j, \beta_w, \beta_h &\propto 1\\ p(\beta_j, \beta_w, \beta_h | \mathbf{s}_j, \mathbf{s}_i) &\propto p(\mathbf{s}_j | \mathbf{s}_i, \beta_j, \beta_w, \beta_h, \sigma_w)\end{aligned}$$

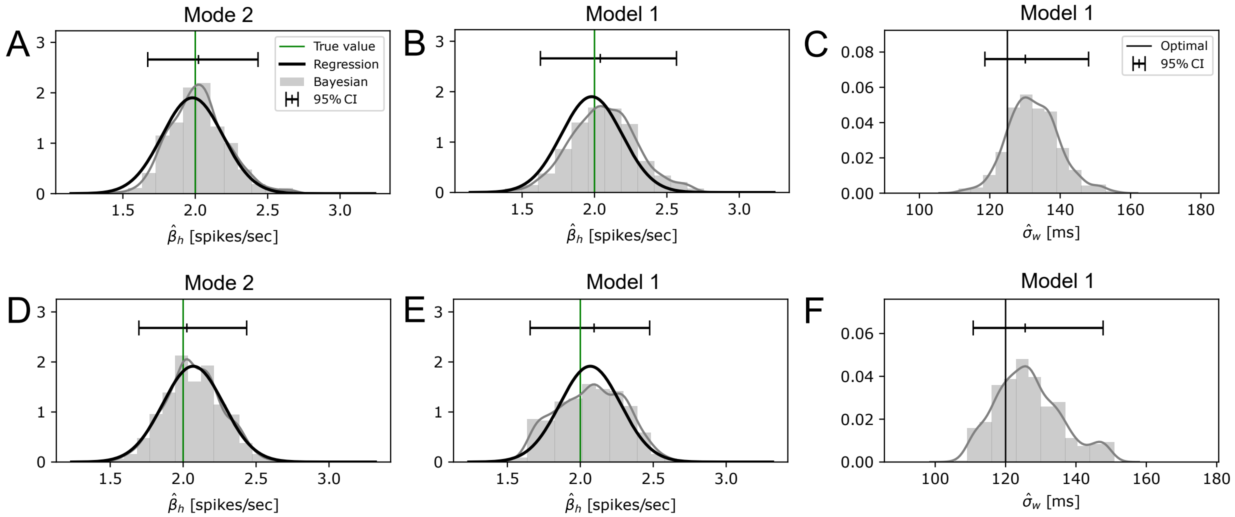


Figure D.7: **Applications of Bayesian model 1 and model 2 to two datasets.** The figure shows the posterior of the estimated coupling filter amplitude $\hat{\beta}_h$ of Bayesian models 1 and 2 (grey histograms in A,B,D,E) and the posterior of the smoothing kernel scale $\hat{\sigma}_w$ of model 1 (grey histograms in C, F). The solid dark curves are the Normal distributions of $\hat{\beta}_h$ obtained using the point process regression model Eq (5.2). **A, B, C** Applications of Bayesian models 1, 2, and the basic regression model to the dataset in Fig (5.3). The timescale of the shared activity $f_{i,j}$ is fixed at $\sigma_I = 100$ ms. Details of the dataset description is in the main text. In C, the mode of the kernel scale is 130 ms, the 95% CI is [119, 148] ms. The optimal kernel scale σ_w selected by the regression model is 125 ms. **D, E, F** Applications of Bayesian models 1, 2, and the basic regression model to the dataset in section D.4.1, where the timescale of the shared activity σ_I varies from 80 ms to 140 ms. In F, the mode of the kernel scale is 126 ms, the 95% CI is [111, 148] ms. The optimal kernel scale σ_w selected by the regression model is 120 ms.

Fig D.7 presents the estimated coupling filter coefficient of Bayesian model 1, 2, and the basic regression model using two simulation datasets. One dataset has fixed shared activity timescale $\sigma_I = 100$ ms, shown in plot A,B,C; the other dataset has a time-varying timescale in a continuous range between 80 ms and 140 ms, shown in plots D,E,F. In A and D, the posterior distributions of

$\hat{\beta}_h$ (grey histogram) and the estimated distribution of the regression model (solid curves) are vary close. This can be a side proof of the result in Supplementary D.4.7, that $\hat{\beta}_h$ has asymptotic Normal distribution. In both datasets (first row and second row), by comparing the results between model 1 and model 2, incorporating the uncertainty of the smoothing kernel scale σ_w does not change the posterior of $\hat{\beta}_h$ too much. As shown in Fig 5.4, the selected smoothing kernel scale σ_w is related to the timescale of the shared activity σ_I. If σ_I increases, the corresponding selected σ_w will increase by around the same amount. By comparing Fig D.7C and F, the CI width does not change a lot (from 29 ms in C to 37 ms in F) when the timescale of the background switched from a fixed value $\sigma_I = 100$ ms to a randomly varying value in $[80, 140]$ ms. So the uncertainty of the σ_w does not directly reflect the variance of the shared activity timescale.

D.4.5 Non-parametric fitting for the coupling filter

For simplicity, the models presented in the main text and many sections in the supplementary use a square window for the coupling filter. In this section, we consider non-parametric fitting for the coupling filter through B-spline bases. The linear form of the intensity function in Eq 5.4 can be easily extended for this purpose. The coupling filter now is estimated as a linear combination of B-spline bases as follows,

$$h_{i\to j}(s) = \beta_{h,1}B_1(s) + ... + \beta_{h,k}B_k(s)$$

where $B_1, ..., B_k$ are spline bases. Define the covariates in the regression,

$$\phi_{h,1}(t) := \int B_1(t-s)N_i(\mathrm{d}s), ..., \ \phi_{h,k}(t) := \int B_k(t-s)N_i(\mathrm{d}s)$$

The intensity function in Eq 5.4 becomes,

$$\tilde{\lambda}_j(t) = \beta_j + \beta_w \overline{\mathbf{s}_i}(t) + \beta_{h,1}\phi_{h,1}(t) + ... + \beta_{h,k}\phi_{h,k}(t)$$

$\overline{\mathbf{s}_i}(t)$ is same as the coarsened spike train in Eq 5.5. The coefficients of the coupling filter $\beta_{h,1}, ..., \beta_{h,k}$ can still be estimated using the model in Eq 5.2. The optimization algorithm is in Appendix D.1. We applied the non-parametric fitting to the dataset in Fig 5.3. The true coupling filter is a square window $h_{i\to j}(t) = \alpha_h \cdot \mathbb{I}_{[0,\sigma_h]}(t)$ where the amplitude of the coupling filter is $\alpha_h = 2$ spikes/sec, the timescale of the square window is $\sigma_h = 30$ ms. The coupling filter is estimated in lag window $[0, 50]$ ms using B-splines with 9 equal-distance knots. We evaluate the risk using root-mean-integral-square error (RMISE). The RMISE between the true coupling filter $h(t)$ and the estimator $\hat{h}(t)$ is defined as follows. $L_h = 50$ ms is the length of the coupling filter.

$$\mathrm{RMISE}(h, \hat{h}) := \sqrt{\frac{1}{L_h}\int_0^{L_h} \left(\hat{h}(t) - h(t)\right)^2 \mathrm{d}t}$$

We evaluate the bias and the standard error of the filter at lag 5, 15, 25 ms. The result is shown in Fig D.8 below.

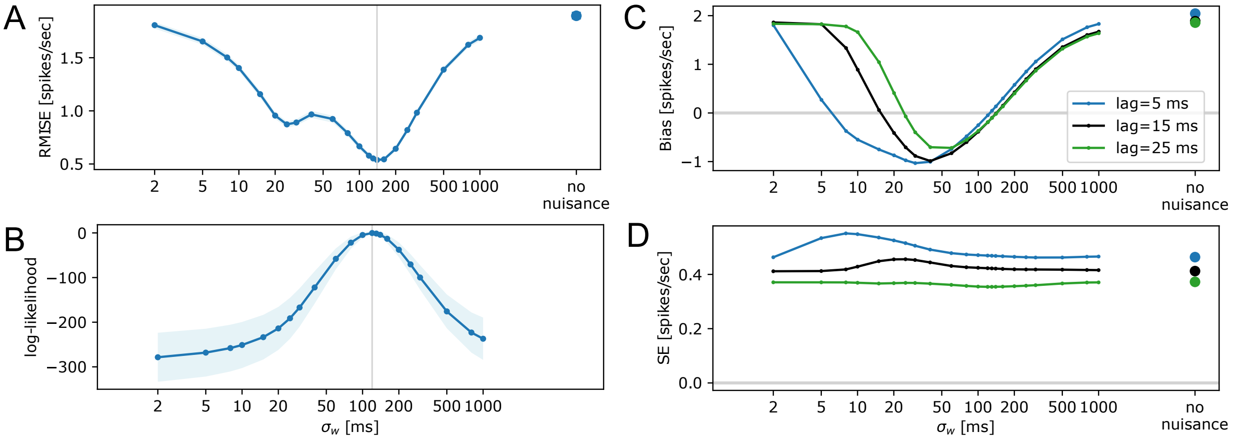


Figure D.8: **Non-parametric fitting for the coupling filter.** The dataset and the non-parametric estimator are described in the test. The results are presented in the same way as Fig 5.3. **A** RMISE of the estimated coupling filter as a function of smoothing kernel width σ_w of W in Eq (5.5). The vertical line indicates the minimum risk. **B** The maximum log-likelihood as function of σ_w. Since the likelihood functions may have different offsets, we align them by the peak (the maximum value across σ_w) to zero, then calculate the mean and pointwise standard deviation. The vertical line indicates the peak of the mean log-likelihood. **C** The bias of the estimator is evaluated at lag = 5, 15, 25 ms. **D** The standard error of the estimator is evaluated at lag = 5, 15, 25 ms.

The risk curve and the log-likelihood curve are similar to the result in Fig 5.3. The optimal model with minimum risk can be selected by maximizing the likelihood, which is the same as the basic regression scenario in Fig 5.3. The difference is that, in the non-parametric fitting, the left local minimum risk has a higher value than the right local minimum. While in the basic fitting case, two local minimum values are close (labeled by "min-1" and "min-2" in Fig 5.3). This can be explained by decomposing the risk into bias and SE shown in Fig D.8C and D. If the smoothing kernel width σ_w is around 130 ms, the bias values at different lags of the coupling filter are nearly the same. But if σ_w is around 20 ms, the bias values at different lags have large divergence: the beginning part of the estimator at lag=5 ms has negative bias, the middle part at lag=15 ms has around zero bias, and the end part of the estimator at lag=25 ms has positive bias. The SE of the estimator at different lags does not change a lot as σ_w varies. So overall, the RMISE has a much larger value near lag=20 ms than at lag=130 ms. These properties are further demonstrated in Fig D.9.

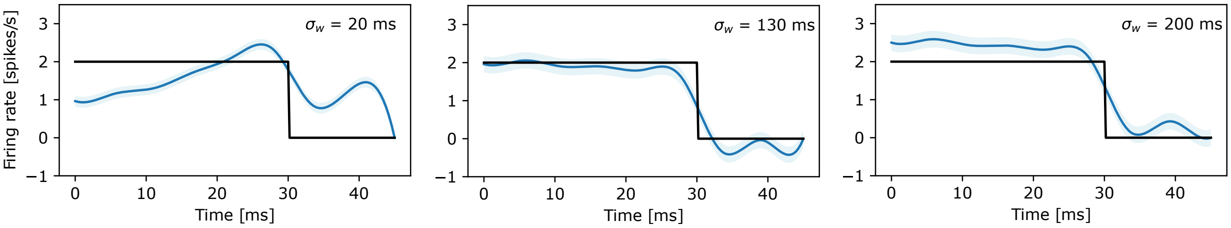


Figure D.9: **Non-parametric fitting for the coupling filter.** The figure compares the true coupling filter (dark) and the estimator (blue). The light blue band is pointwise 95% CI. The coupling filters were fitted in the same way as described in Fig D.8. This figure picks out some fitted estimators with different smoothing kernel widths $\sigma_w = 20, 130, 200$ ms. If the smoothing kernel width is too small ($\sigma_w = 20$ ms), the bias values of the estimator at different lags have large differences. This matches the bias curves shown in Fig D.8C. At the beginning part of the estimated coupling filter around lag=5 ms, the bias is negative, and at the end part around lag=25 ms, the bias is positive. If the smoothing kernel width is selected optimally ($\sigma_w = 130$ ms), the fitted coupling filter matches the true filter very well. If the smoothing kernel width is too wide ($\sigma_w = 200$ ms), the whole estimated coupling filter has uniform positive bias at different lags. This agrees with Fig D.8C that multiple bias curves with different lags beyond $\sigma_w = 130$ ms are very close.

D.4.6 Selection of coupling filter length.

The simulation scenario in the main text in Fig 5.3 and many scenarios in the supplementary sections simplify the coupling filter estimation using a square window and assume the timescale of the coupling effect σ_h in Eq (5.7) is known. In this section, we show the consequences of unmatched coupling filter timescale. Because in practice, the timescale of the coupling effect is usually unknown.

We used the same dataset in Fig 5.3, where the true coupling filter is a square window. The amplitude $\alpha_h = 2$ spikes/sec and the window width is $\sigma_h = 30$ ms. We applied two versions of the regression model to the dataset. Both versions used a square window as the coupling filter estimator, but one with shorter timescale $\sigma_{h,1} = 20$ ms, the other with longer timescale $\sigma_{h,2} = 40$ ms. We present the results in Fig D.10 in the same way as Fig 5.3.

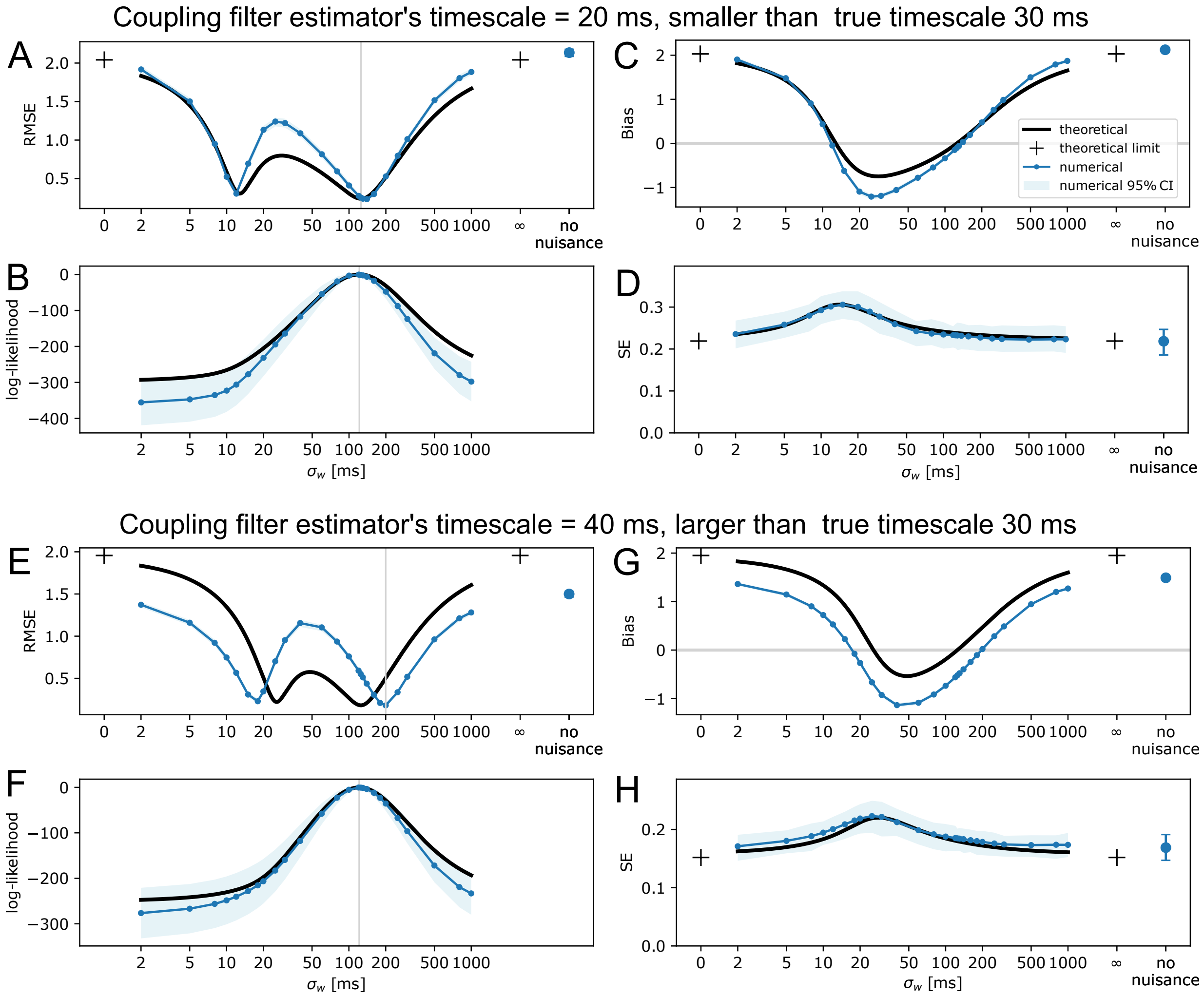


Figure D.10: **Consequences of unmatched coupling filter timescale.** The dataset is the same as Fig 5.3, where the true coupling filter is a square window in Eq 5.7. The amplitude is $\alpha_h = 2$ spikes/sec and the window width is $\sigma_h = 30$ ms. We tested the regression model with unmatched coupling filter width. The model in **A, B, C, D** estimates the coupling filter using a shorter timescale $\sigma_{h,1} = 20$ ms. The model in **E, F, G, H** estimates the coupling filter using a longer timescale $\sigma_{h,1} = 40$ ms. As a reference, the dark curves show the theoretical approximation using the basic regression model by setting the coupling filter timescale as 20 ms in A-D, and 40 ms in E-H. The rest settings are the same as the simulation.

If the coupling filter is estimated using a shorter timescale ($\sigma_{h,1} = 20$ ms) as shown in Fig D.10A,B,C,D, the selected model still has the minimum risk indicated by the vertical line. The dark curves in Fig D.10A,B,C,D, show the theoretical approximation of the properties using the basic regression model in Eq (5.2)-(5.5) by setting the coupling filter timescale as 20 ms instead, which can be seen as the expected properties of the model. The absolute values of the bias are larger than expected if σ_w is between 10 ms and 120 ms or larger than 200 ms. But the roots of the bias still match the expected position. The SE is not affected by the unmatched timescale. So the optimal selection of σ_w does not change. As a contrast, if the coupling filter timescale of the estimator ($\sigma_{h,2} = 40$ ms) is longer than the truth (30 ms), the consequence is more severe. As shown in Fig D.10 G, the actual bias is uniformly lower than the expected bias. The SE is not affected. So the consequence is that the selected smoothing kernel width σ_w (Fig D.10 F vertical line) does not match the actual risk minimum (Fig D.10 E vertical line).

By combining the results of the two cases, we recommend users select shorter coupling filter timescale if they are not confident about the coupling filter timescale, or using non-parametric fitting as in Supplementary D.4.5.

D.4.7 Asymptotic Normality of the estimator

In this section, we perform simulations to verify the asymptotic Normality property of the estimator, see Lemma D.2.5. The dataset is the same as Fig 5.3. The true coupling filter is a square window $h_{i\to j}(t) = \alpha_h \cdot \mathbb{I}_{[0,\sigma_h]}(t)$, where the amplitude is $\alpha_h = 2$ spikes/sec, and the timescale is $\sigma_h = 30$ ms. The estimator for the coupling effect $\hat{\beta}_h$ is Eq 5.2 and 5.7. We compare the empirical distribution of the estimators with the theoretical distribution. The theoretical distribution is Normal, where the mean is the true value α_h plus the theoretical bias (Lemma D.2.2), the standard deviation is the theoretical standard error (Lemma D.2.3).

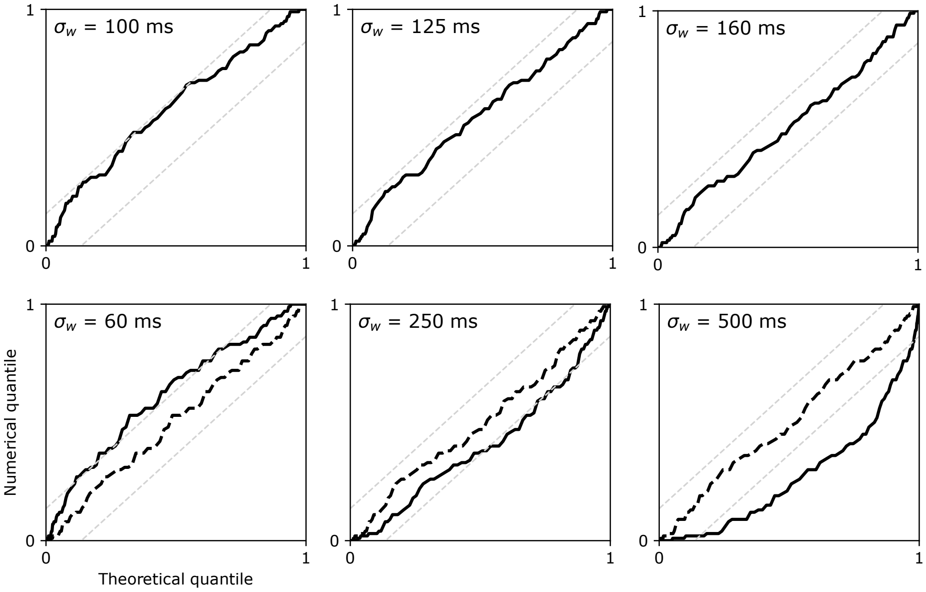


Figure D.11: **Normality of the estimator's distribution.** The dataset is the same as Fig 5.3 including 100 repetitions. The figure shows the Q-Q plots of the empirical distribution of the estimator against the theoretical Normal distribution (see details in the text), shown in the dark curves. The straight dashed grey lines are 95% CI. In the first row, the empirical distribution matches the theoretical distribution very well at the optimal model ($\sigma_w = 125$ ms) and at models close to the optimal ($\sigma_w = 100, 160$ ms). We also evaluate the model at many other different smoothing kernel widths. If σ_w is too small or too large (second row, $\sigma_w = 60, 250, 500$ ms), the empirical distribution has a large deviation from the theoretical distribution. This is caused by the error of the theoretical approximation of the bias, see Fig 5.3C. If the mean of the theoretical Normal distribution is replaced by the mean of the numerical estimators (mean of all estimators), but the standard error of the theoretical distribution remains the same, the distributions can match very well (dashed dark curves in the second row).

As shown by Fig D.11 first row, the estimator has Normal distribution at the optimal selection of $\sigma_w = 125$ ms and near-optimal selections $\sigma_w = 100, 160$ ms. In Fig D.11 second row, if σ_w is too small or too large, the empirical distributions will have large deviations. This is caused by the error of the theoretical approximation of the mean but not by the theoretical approximation of the standard error. The dashed curves in Fig D.11 second row show the good match after replacing the theoretical mean with the empirical mean (mean of all estimators).

D.4.8 Hypothesis testing example

The regression model can be adopted for hypothesis testing problems. The simulation scenario in this section is similar to the case in Fig 5.3. The background activity is a cluster point process in Eq (5.6), and the coupling filter is a square window $h_{i\to j}(t) = \alpha_h \cdot \mathbb{I}_{[0,\sigma_h]}(t)$ in Eq (5.7). Each simulation dataset only has 10 trials. We reduce the sample size to make the tasks more difficult, and the performances of different estimators will be more distinguishable. The length of the trial is 5 seconds, the time scale of the background activity is $\sigma_I = 100$ ms. The intensity of the center process is $\rho = 30$ spikes/sec. The baselines of two neurons are $\alpha_i = \alpha_j = 10$ spikes/sec. The coupling filter is estimated using a square window with known timescale $\sigma_h = 30$ ms. The amplitude of the coupling filter is $\alpha_h = 0$ spikes/sec in the null cases without coupling effects. We include three true positive scenarios with coupling filter amplitudes $\alpha_h = 2, -2, 1$ spikes/sec respectively. The dataset generating and the model fitting procedure was repeated for 100 times.

Consider the null hypothesis

$$H_0 : \; \hat{\beta}_h = 0$$

$\hat{\beta}_h$ is the estimator for α_h. The inference method is a direct application of the properties of the estimator. The smooth kernel is $\sigma_w = 125$ ms, which is chosen by maximizing the likelihood. $\hat{\beta}_h$ has asymptotic normal distribution (see details in Supp D.4.7 and Appendix D.2), so the p-value can be easily calculated accordingly. The alternative method is jitter-based CCG, where the time bin width is 2 ms, the jitter window width is 100 ms. The CCG with shorter or longer jitter window width, for example 60 ms or 140 ms, gives similar results, so the figures are not shown. The p-value of the method is obtained by considering the multiple testing across all time lags between 0 and 30 ms, which is the same as the true coupling filter length. The calculation detail is in [3] supplementary document.

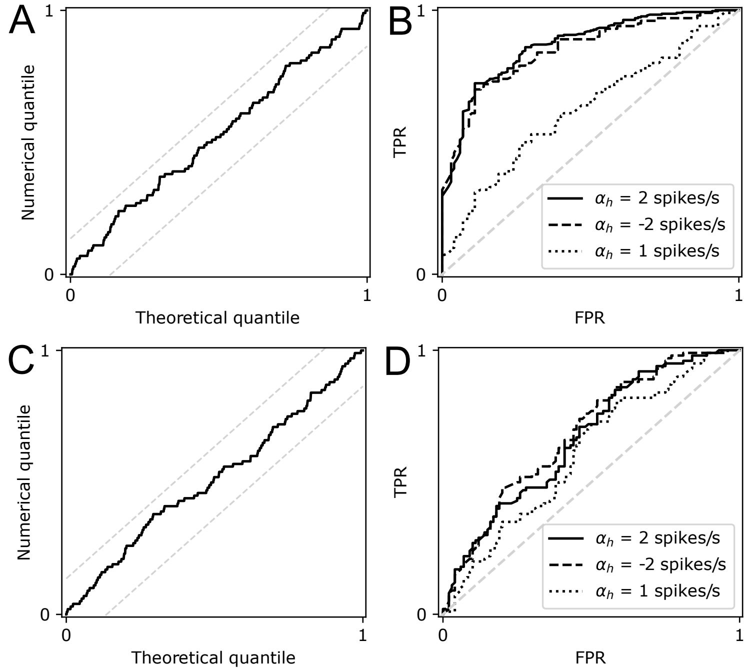


Figure D.12: **Hypothesis testing examples.** We compare the point process regression model (**A, B**) with the jitter-based cross-correlation (CCG) method (**C, D**) using simulations. The simulation details are in the text. The left column is the Q-Q plot which compares the p-values distribution under the null (numerical quantile along the y-axis) with the uniform distribution (theoretical quantile along the x-axis). The dashed line is the 95% CI. Both methods yield valid p-value distributions. The right column shows the results of ROC analysis with the false positive rate (FPR) along the x-axis, and the true positive rate (TPR) along the y-axis. The score of an outcome is the p-value of the hypothesis test. When $\alpha_h = 2, -2$ spikes/sec, the area under the curve of our method is larger than that of the jitter-based CCG method, so our method is more powerful. However, when the coupling effect is weak $\alpha_h = 1$ spikes/sec, neither of the methods has satisfactory performance.

Fig D.12A verifies that the p-value distribution of the basic point process regression method under the null is uniform. This provides another verification of the estimator's Normality distribution, which has already been shown in Supplementary D.4.7. The p-value of the CCG method under the null is also valid as shown in Fig D.12C. Fig D.12B and D compare the ROC analysis. The score of a test outcome is the p-value. The point process regression method has better performance when the amplitude of the coupling effect is $\alpha_h = 2, -2$ spikes/sec. Both methods have poor performance if the coupling effect is very weak $\alpha_h = 1$ spikes/sec. Usually, the jitter-based method focuses on pointwise statistic at a specific time lag, which can ignore the connection between adjacent time lags. We think the power of the CCG method can be improved by considering the time lag dependency and designing the multiple hypothesis test more carefully, but it is not the main interest of this paper.

D.4.9 Background activity with Laplacian window function

The fluctuating shared activity of the simulation scenario shown in Fig 5.3 is modeled as a cluster process in Eq (5.6) composed of superimposed Gaussian window functions. The smoothing kernel in Eq (5.5) is also Gaussian. In this section, we verify that if the Gaussian window function of the background activity is replaced by another window with a different shape, the estimator will not be affected. Here we replace the Gaussian function with a Laplacian window function [3]. The shared activity $f_{i,j}$ in Fig 5.2 becomes,

$$f_{i,j}(t) = \sum_i \phi_{\sigma_I}\left(t - t_i^c\right)$$

$$\phi_{\sigma_I}(x) = \frac{1}{\sqrt{2}\sigma_I} \exp\left\{-\frac{|x|}{\sigma_I/\sqrt{2}}\right\} \tag{D.42}$$

where t_i^c are the time points of the center process, σ_I controls the timescale of the window. The simulation settings of this scenario are the same as the Fig 5.3 except for the new window function. The timescale is $\sigma_I = 100$ ms.

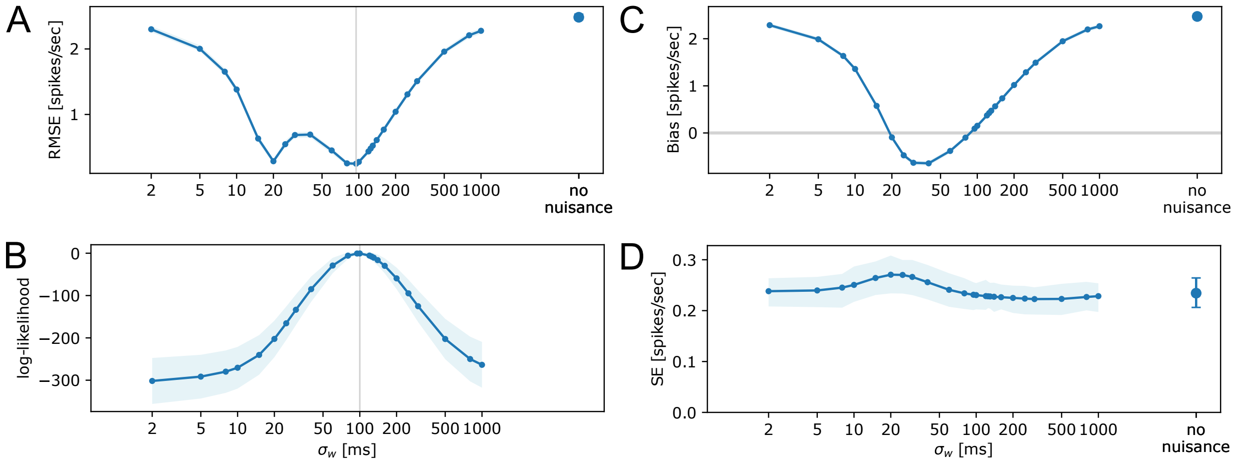


Figure D.13: **Results using Laplacian window in the background activity.** Similar to Fig 5.3, we present the results in the same way except that the window function of the background activity is replaced by Eq D.42 with timescale $\sigma_I = 100$ ms. The properties of the estimator and the conclusion do not change.

The numerical results are shown in Fig D.13. Similar to Fig 5.3, the properties of the estimator and the conclusion do not change[1]

D.4.10 Multivariate regression and partial relation

Multivariate regression is a natural extension of the basic regression model introduced in Eq 5.2-5.5 and diagram Fig 5.2. We briefly mentioned in Supplementary D.4.3 that if the shared input

[1] Author note: It should be very easy to derive the theoretical results for the Laplacian window. I only need to recalculate the quantities in Lemma D.2.8. But I did not have enough time to do so before the thesis defense. I will do this after.

between two neurons is a mixture of activities with very distinct timescales, the model can fail in eliminating all artifacts. This section continues the discussion.

The simulation scenarios are shown in the diagrams in Fig D.14. Consider two neurons X, Y that are driven by some shared input, but there is no coupling effect between them. The first diagram in Fig D.14A is the model in the main text Fig 5.2 and Fig 5.3 with relatively slow-changing background $f_{i,j}$. In the diagram Fig D.14B, we only include fast-changing driving activity. Z represent a subpopulation of 4 neurons. The spikes of neurons in Z influence X, Y through coupling filters $h_{Z \to X} = h_{Z \to Y} = \alpha_Z \cdot \mathbb{I}_{\sigma_Z}$, $\alpha_Z = 15$ spikes/sec, $\sigma_Z = 30$ ms. The activity of neurons in Z and X, Y have constant baselines of 20 spikes/sec. The diagram in Fig D.14C includes both slow-changing activity $f_{i,j}$ as in A and fast-changing activity driven by spikes from Z as in B. The coupling filters are the same as plot B. The neurons in Z and X, Y are all driven by fluctuating background activity $f_{i,j}$. $f_{i,j}$ is a linear Cox process in Eq 5.6, where the intensity of the center process is $\rho = 20$ spikes/sec, and the window function is Gaussian with scale $\sigma_I = 100$ ms. The constant baseline for all neurons is 10 spikes/sec. The simulation dataset has 200 5-second trials. This section mainly studies scenarios B and C. The filters were fitted using non-parametric method in Appendix D.1.

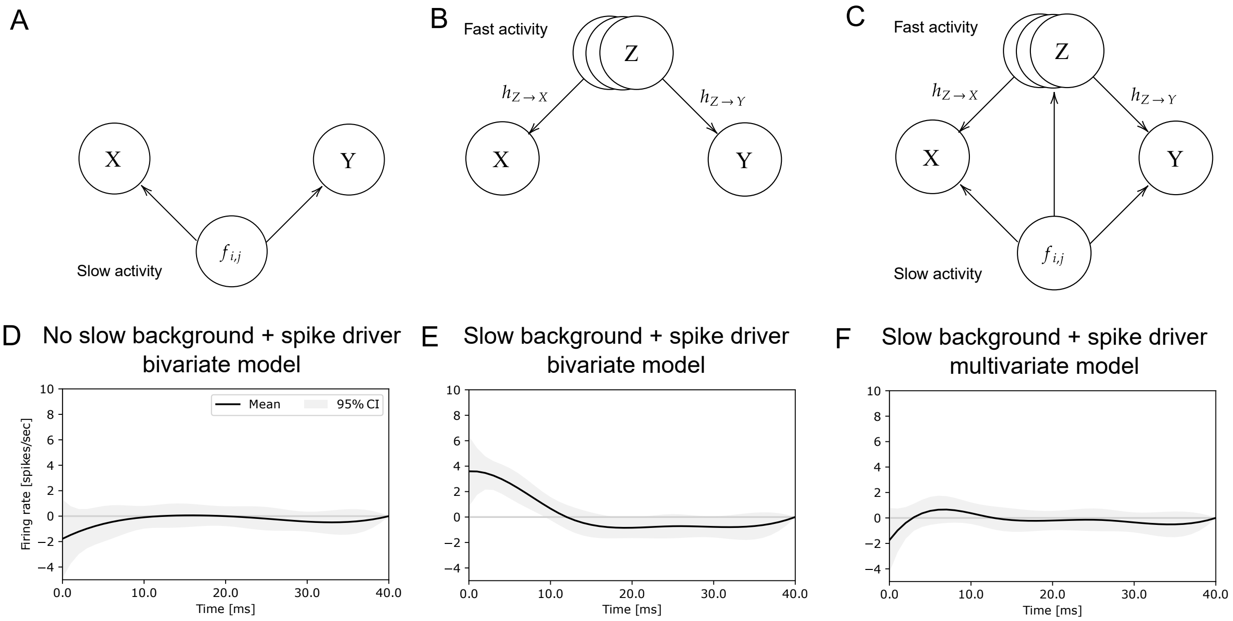


Figure D.14: **Shared driving factors on multiple timescales.** **A** The diagram as already shown in Fig 5.2 without coupling effects. Two neurons X, Y are driven by slow-changing activity $f_{i,j}$. **B** Neurons X, Y are driven by fast-changing activity triggered by Z. Spikes from a subpopulation Z drives the activities of X and Y through coupling filters $h_{Z\to X}$ and $h_{Z\to Y}$, but there is no direct coupling filter between X and Y. Simulation details are in the text. **C** A combination of the cases in A and B, where neurons X, Y are driven by both fast-changing activity from Z and slow-changing activity $f_{i,j}$. **D** Estimated coupling filter $X \to Y$ using the basic bivariate regression model. The spike trains are generated using diagram B. The model can handle artifacts caused only by fast-changing activity. **E** Estimated coupling filter $X \to Y$ using the basic bivariate regression model. The spike trains are generated using diagram C with background on two distinct timescales. The model can not fully remove the artifacts. **F** The estimated coupling filter $X \to Y$ using a multivariate regression in Eq D.43. Conditioning on both estimated $f_{i,j}$ and Z, the artifacts can be removed.

Fig D.14D,E,F show the results. In Fig D.14D, the data was generated by the model in Fig D.14B with only fast-changing component. The basic bivariate regression model is able to eliminate the artifacts caused by fast-changing background activity. This has already been shown in Supplementary D.4.3. Fig D.14E shows the estimated filter using the dataset generated by the model in In Fig D.14C. The filter falsely shows excitatory effect before lag=10 ms. The bivariate model in Eq 5.2 fails in eliminating the artifacts when the shared input is on two distinct timescales. If we can observe the spike trains of Z, then the multivariate regression can help with eliminating the artifacts related to the fast-changing activity. Consider the intensity function for the multivariate point process regression,

$$\begin{aligned}\tilde{\lambda}_Y(t) =& \beta_Y + \beta_w\, \overline{\mathbf{s}_X}(t) + \int_0^t h_{X\to Y}(t-\tau)N_X(\mathrm{d}\tau)\\ &+ \int_0^t h_{Z_1\to Y}(t-\tau)N_{Z_1}(\mathrm{d}\tau) + ... + \int_0^t h_{Z_4\to Y}(t-\tau)N_{Z_4}(\mathrm{d}\tau)\end{aligned} \tag{D.43}$$

$h_{X\to Y}$ is the coupling filter between neuron $X \to Y$. $h_{Z_1\to Y}, ..., h_{Z_4\to Y}$ are coupling filters of neurons $Z_1 \to Y$,..., $Z_4 \to Y$, which are estimated using square window with known timescale as in Eq 5.7. Since all neurons are driven by the same slow-changing activity $f_{i,j}$, the nuisance variable only has one component $\overline{\mathbf{s}_X}(t)$. If there is evidence showing different pairs of neurons are driven by different sources, the model can include more nuisance variables, such as $\overline{\mathbf{s}_{Z_1}}(t)$,...,$\overline{\mathbf{s}_{Z_4}}(t)$. We skip further discussion on this topic.

Fig D.14F shows the results using multivariate regression. The false-positive coupling effect shown in Fig D.14E does not appear in F. This example is analog to conditional correlation. In diagram Fig D.14A, $X \perp Y|f_{i,j}$. If $f_{i,j}$ is approximated properly, the artifacts can be removed. Similarly, in diagram Fig D.14B, $X \perp Y|Z$. In diagram Fig D.14C, X and Y are not necessarily independent only conditioning on either $f_{i,j}$ or X, $X \not\perp Y|f_{i,j}$, or $X \not\perp Y|Z$. However, $X \perp Y|f_{i,j}, Z$, where bot factors of $f_{i,j}$ and Z need to be included so that X, Y can become conditionally independent. The artifacts related to the fast-changing activity is noticeable if the influence is very strong. In simulation scenario Fig D.14C, the coupling filters $h_{Z\to Y}$ of four neurons all have large amplitude $\alpha_Z = 15$ spikes/sec. If the impact of the spikes trains from Z is small, the artifacts can be ignored.

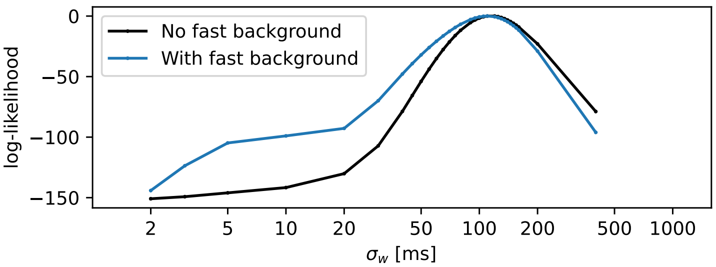


Figure D.15: **A comparison between log-likelihood curves with or without fast-changing background.** The blue curve is the log-likelihood curve of a dataset generated by the model in Fig D.14C. The blue curve is the log-likelihood curve of a dataset generated by the model in Fig D.14A. The left part of the blue curve with small σ_w is elevated due to the presence of the fast-changing background.

The above multivariate regression provides a solution in a special situation where all the spike trains from Z are observed. However, if the input of neurons from all their connected neurons can not be recorded at the same time, the method still can not eliminate all artifacts. So a more general solution is needed for the mixture of distinct background activity timescales. We propose the following model,

$$\tilde{\lambda}_j(t) := \beta_j + \beta_w^{\text{slow}}\ \overline{\mathbf{s}_i}^{\text{slow}}(t) + \beta_w^{\text{fast}}\ \overline{\mathbf{s}_i}^{\text{fast}}(t) + \int_0^t h_{i\to j}(t-\tau)N_i(\mathrm{d}\tau)$$
$$\overline{\mathbf{s}_i}^{\text{slow}}(t) = [W^{\text{slow}} * \mathbf{s}_i](t), \quad \overline{\mathbf{s}_i}^{\text{fast}}(t) = [W^{\text{fast}} * \mathbf{s}_i](t)$$

Instead of using one smoothing kernel in Eq 5.4, the new model includes two smoothing kernels to cover both fast-changing activity and slow-changing activity. The timescale of the background activity varies in a relatively small range, then there is no need for such a model, see Supplementary D.4.1. Such a design is also motivated by the shape of the likelihood curve in Fig D.15. The likelihood with mixture timescales is not obvious multimodal, but the left part of the curve with small kernel width is higher than the likelihood curve without the fast-changing component. This delivers a message of a mixture of timescales. The bias is significant only when the input is very strong. We will leave the research on this type of issue in the future. But we first need to find out if such strong fast-changing background exists in real data.

D.4.11 Self-coupling effect

The basic regression model in the diagram Fig 5.2 only considers the spike-to-spike coupling effect from one neuron to another. In this section, besides the cross-neuron coupling effect, we take the self-coupling effect into account. The self-coupling effect means that the spikes generated from a neuron will feedback to itself and influence its own intensity function in the future. Fig D.16 shows 3 ways of self-coupling: A, the self-coupling effect only occurs on the source neuron i; B, the self-coupling effect only occurs on the target neuron j ; C, the self-coupling effect appears on both neurons. We have not yet developed the theory with self-coupling components, so we only present numerical results through simulations. The data is still fitted using the model in Eq (5.2)-(5.5) without the self-coupling component. The goal in this session is not to model the self-coupling effect explicitly, but to study how the basic regression model is affected by the extra self-coupling as artifacts.

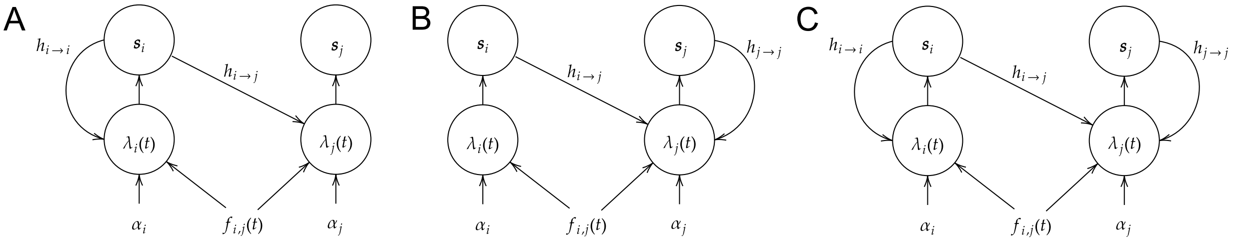


Figure D.16: **Self-coupling effect diagram.** The diagrams are similar to Fig 5.2 except for the extra self-coupling components. **A** The self-coupling effect only appears on the source neuron i. **B** The self-coupling effect only appears on the target neuron j. **C** The self-coupling effect appears on both neurons i, j.

The simulation settings for diagram A are the following. The activity $f_{i,j}$ is set as a cluster process in Eq (5.6) with $\sigma_I = 100$ ms. The intensity of the center process is $\rho = 20$ spikes/sec, the constant baseline intensity for all neurons is $\alpha_j = \alpha_i = 20$ spikes/sec. The square window filter width is $\sigma_h = 30$ ms and amplitude $\alpha_h = 2$ spikes/sec. The self-coupling filter $h_{i\to i}$ is also a square window function with amplitude -15 spikes/sec and width 10 ms, mimicking the refractory period. One simulation case has 200 trials and the length of a trial is 5 sec. Different trials were assigned with randomly generated $f_{i,j}$ independently. The spike trains were fitted using the basic regression model in diagram Fig 5.2 and Eq (5.7). The numerical properties were estimated using 100 replicated simulation cases. The results are shown in Fig D.17. The properties are similar to the case in Fig 5.3. The model can still select the proper smoothing kernel width.

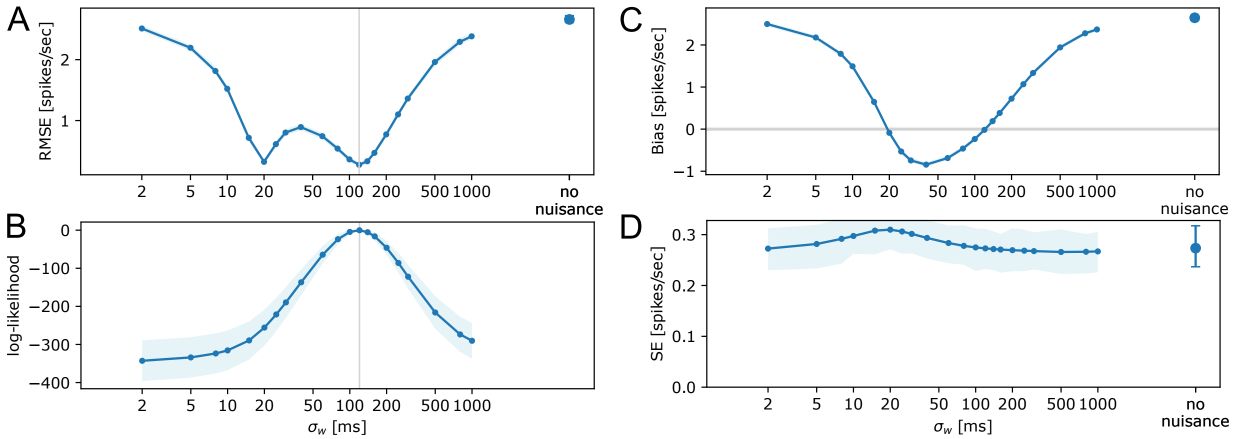


Figure D.17: **Source neuron self-coupling effect.** These results correspond to the model diagram in Fig D.16A. The results are presented in the same way as Fig 5.3. Details of the data and model fitting are in the text.

The simulation settings for diagram B are similar to diagram A. The self-coupling filter $h_{j\to j}$ is a square window function with amplitude -20 spikes/sec and width 20 ms, mimicking the refractory period of neuron j. The results are shown in Fig D.18. In this situation, the self-coupling effect misleads the model to select a smaller smoothing kernel width σ_w. The risk at the selected model by maximizing the likelihood is not too large though. As a comparison, the risk, bias, and SE of the estimator at different σ_w become much smaller than the source neuron self-coupling case in Fig D.17A.

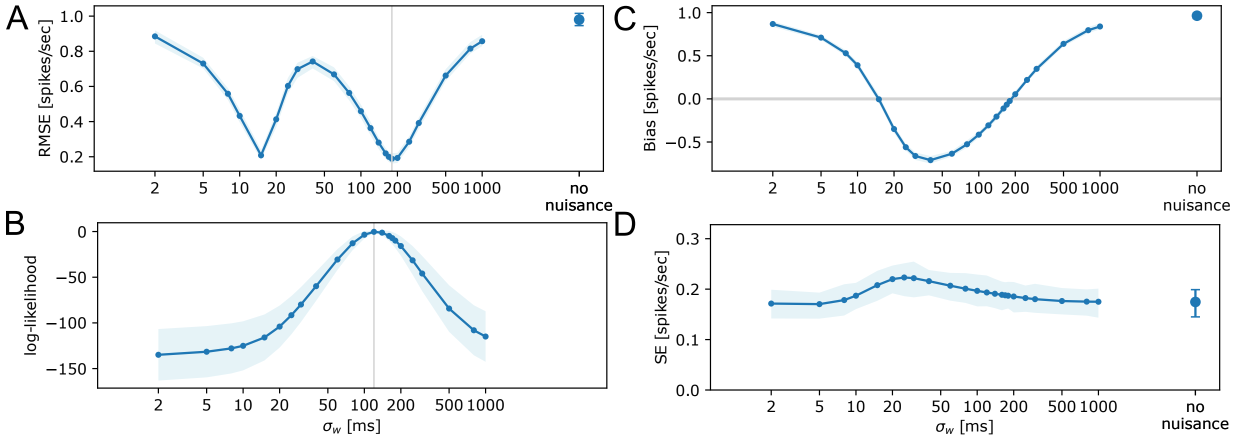


Figure D.18: **Source neuron self-coupling effect.** These results correspond to the model diagram in Fig D.16B. The results are presented in the same way as Fig 5.3. Details of the data and model fitting are in the text.

The simulation settings for diagram C are similar to diagrams A and B. The model includes both self-coupling filters $h_{i\to i}$ on the source neuron in diagram A and $h_{j\to j}$ on the target neuron in diagram B. The results are in Fig D.19. The case inherits the issue from diagram B that the self-coupling effect on the target neuron can mislead the model to select a smaller kernel width. The bottom line is that the introduced extra risk is small.

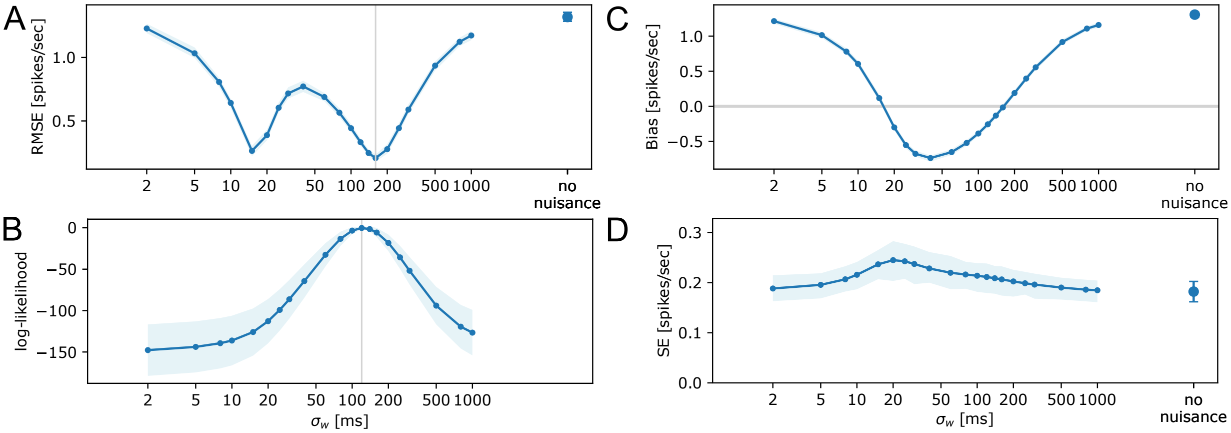


Figure D.19: **Self-coupling effect on both source and target neurons.** These results correspond to the model diagram in Fig D.16C. The results are presented in the same way as Fig 5.3. Details of the data and model fitting are in the text.

Another issue about self-coupling is that it can cause lack of fit. Next we perform goodness-of-fit test using KS test based on time rescaling theorem [24, 62], also see Supplementary D.4.16. Fig D.20 below use some examples to compare three scenarios of self-coupling with respect to the diagrams in Fig D.16.

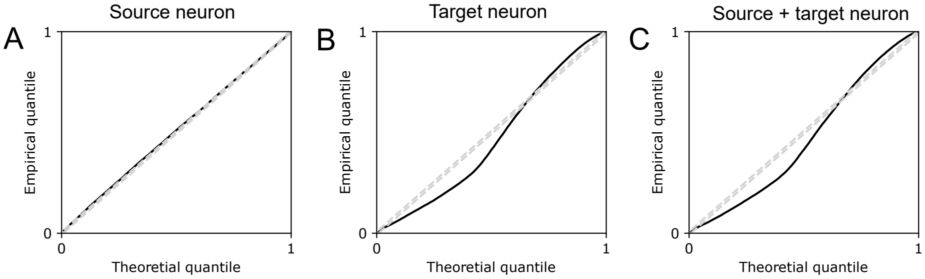


Figure D.20: **Influence of self-coupling effect on goodness-of-fit test.** The figure shows the goodness-of-fit test based on KS test as described in Supplementary D.4.16. 3 scenarios correspond to the model diagrams in Fig D.16 and numerical simulations in Fig D.17, D.18, and D.19. **A** Only the source neuron has self-coupling effect. The goodness-of-fit is not affected. **B** Only the target neuron has self-coupling effect. The bivariate regression model in the main text Eq (5.2)-(5.5) shows lack of fit. **C** Both source and target neurons have self-coupling effect. This case still has the problem in B that the basic regression model has lack of fit on some level.

The basic regression model shows a lack of fit only when the self-coupling effect involves the target neuron. If the goal of the model is to estimate the cross-neuron coupling effect not the self-coupling effect, the introduced risk or bias of the estimator can be alleviated by selecting slightly larger smoothing kernel width than the one maximizing the likelihood. If the small error is acceptable, the kernel width correction is not necessary.

D.4.12 Rate coupling and delayed shared activity

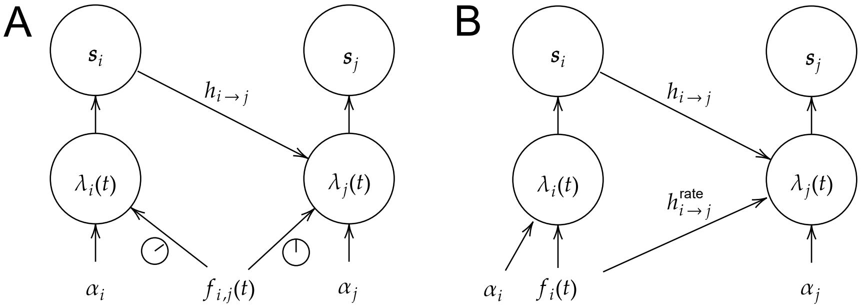


Figure D.21: **Rate coupling and delayed shared input.** **A** Similar to the scenario in Fig 5.3 and diagram Fig 5.2, two neurons share the same input but with different delays. Having different delays results in that the lagged neuron's activity can be predicted by the leading neuron somehow at a certain time as the leading neuron receives the same information earlier. **B** The special case in A is under a more general model, rate coupling model, where two neurons interact with not only spike-to-spike, but through rate coupling filter $h^{\text{rate}}_{i\to j}$.

The coupling filter describes the influence from neuron i's spikes to neuron j's intensity function. Is that possible if two neurons do not have any spike-to-spike coupling effect, but the estimated coupling effect is totally due to the coupling between the underlying intensity functions on a fine timescale? In this section, we study a scenario where two neurons, the source neuron i and target neuron j in diagram Fig D.21A, have the same shared input $f_{i,j}$, but the activity arrives at two neurons with subtle different delays.

$$\begin{aligned}\lambda_j(t|\mathcal{H}_t) &= \alpha_j + f_{i,j}(t-\tau_{\text{lag}}) + \int_0^t h_{i\to j}(t-r)N_i(\mathrm{d}r)\\ \lambda_i(t|\mathcal{H}_t) &= \alpha_i + f_{i,j}(t)\end{aligned}$$

where τ_{lag} is the lag between the source neuron and the target neuron. If $\tau > 0$, the share activity arrives at the source neuron earlier, If $\tau < 0$, the share activity arrives at the source neuron later. Delaying a function is equivalent to convolving the function with a delayed Dirac delta function

$$f_{i,j}(t-\tau_{\text{lag}}) = [f_{i,j}(s) * \delta(s-\tau_{\text{lag}})](t)$$

So the model in diagram Fig D.21A can be seen as a *rate coupling* as shown in panel B. The model equations are rewritten as,

$$\begin{aligned}\lambda_j(t|\mathcal{H}_t) &= \alpha_j + \int_0^t h^{\text{rate}}_{i\to j}(t-s)f_i(s)\mathrm{d}s + \int_0^t h_{i\to j}(t-r)N_i(\mathrm{d}r)\\ \lambda_i(t|\mathcal{H}_t) &= \alpha_i + f_i(t), \quad f_i(t) = f_{i,j}(t)\end{aligned}$$

where $h^{\text{rate}}_{i\to j}(t) = \delta(t-\tau_{\text{lag}})$. $f_i = f_{i,j}$ is the exclusive input for neuron i. Neuron i's firing rate affects neuron j through the rate coupling filter $h^{\text{rate}}_{i\to j}$. Besides delaying, the shared activity can be transformed using more complicated functions. For example, an arbitrary rate coupling filter function can be approximated using a sequence of delta functions,

$$h^{\text{rate}}_{i\to j}(t) \approx h^{\text{rate}}_{i\to j}(0)\delta(t) + h^{\text{rate}}_{i\to j}(\Delta)\delta(t-\Delta) + ... + h^{\text{rate}}_{i\to j}(k\Delta)\delta(t-k\Delta) + ...$$

Next, we only present the results of the simple case with delayed shared input. The diagram of the model is in Fig D.21. The simulations include two scenarios, where the settings are the same as Fig 5.3 except that: scenario 1, the target neuron leads the source neuron by 20 ms. So the shared activity of the source neurons can be roughly predicted by the target neuron's activity as it arrives at the target neuron earlier; scenario 2, the source neuron leads the target neuron by 20 ms. So the activity of the target neuron can be roughly predicted by the source neuron's activity for the same reason. The simulation results are shown in Fig D.22.

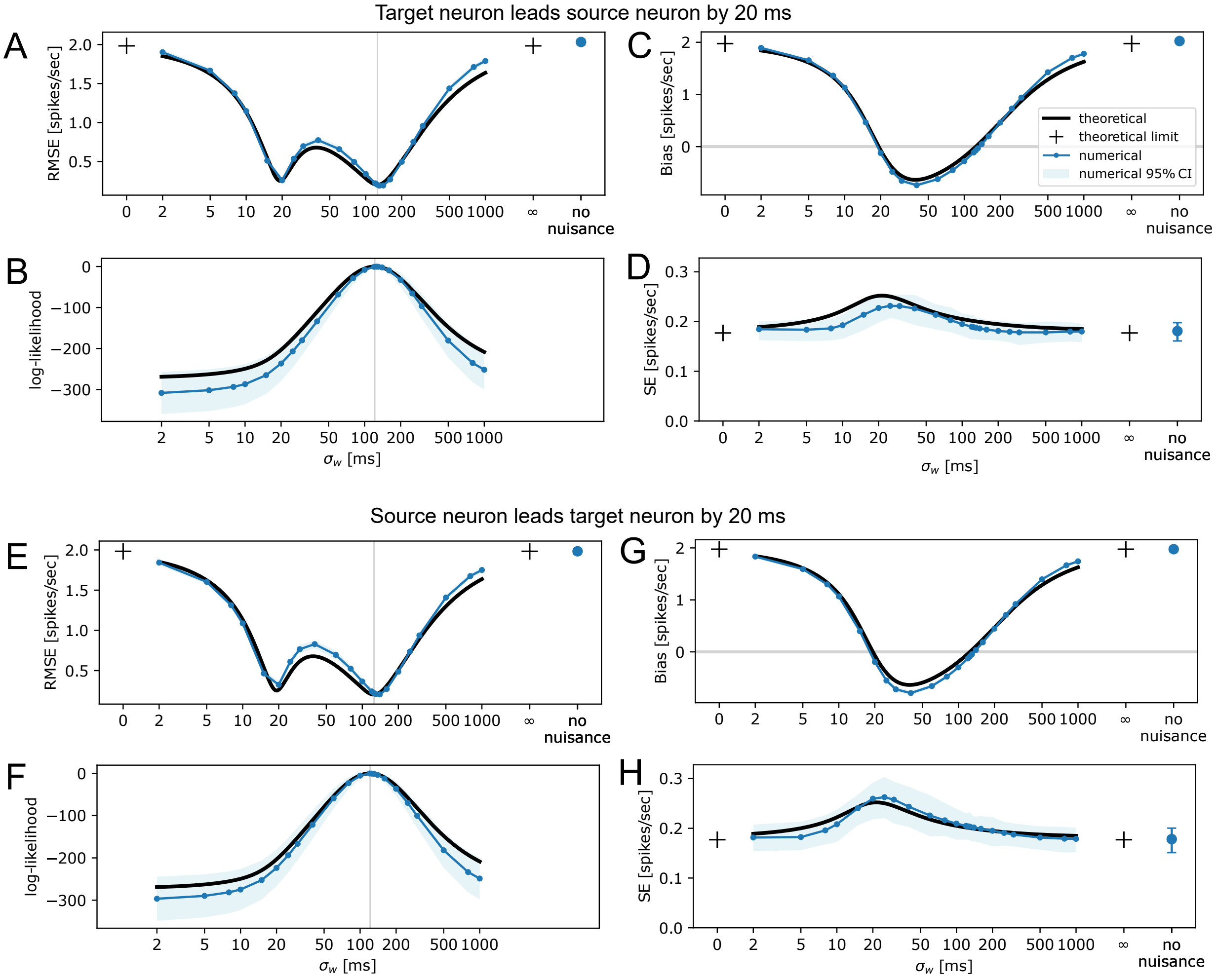


Figure D.22: **Delayed shared input.** The figure presents the properties of the estimator in the same way as Fig 5.3. The settings of the simulation are in the text. The dark curves show the theoretical results exactly the same as the dark curves in Fig 5.3 without considering the delays of the shared activity. **A,B,C,D** The shared activity $f_{i,j}$ arrives at the target neuron 20 ms earlier. **E,F,G,H** The shared activity $f_{i,j}$ arrives at the source neuron 20 ms earlier. Delaying the shared input does not change the results significantly.

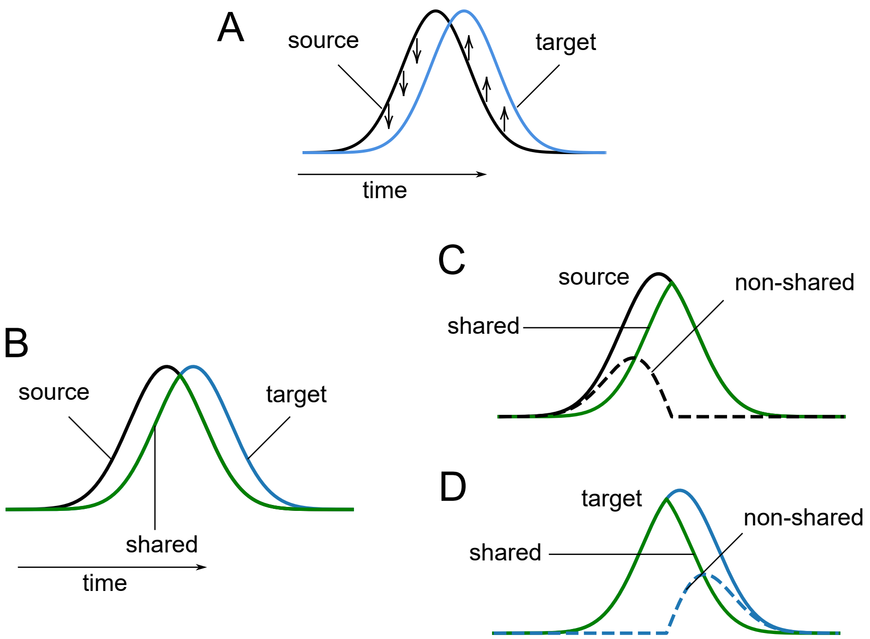


Figure D.23: **Intuitive explanations of delayed shared input. A** The source neuron leads the target neuron. The inhibitory (down arrows) and excitatory (up arrows) effects will be balanced out. **B,C,D** Explanation using decomposition. The inputs between two neurons are not fully overlapped due to the delay. It can be decomposed into the shared component colored in green, and non-shared parts in dashed curves.

Fig D.22 show the numerical results. The dark curves are the theoretical results from Fig 5.3 without considering the delays between neurons. By comparing the numerical results and the reference theoretical results, delaying the shared input does not modify the properties of the estimator too much.

The negligible effects can be explained in two ways as shown in Fig D.23. In Fig D.23A, the shared activity arrives at the source neuron (dark curve) earlier than the target neuron (blue curve). Before the activity reaches the peak, the source activity is higher than the target neuron, which demonstrates an inhibitory effect (down arrows in panel A). After the peak, the source neuron activity is lower than the target neuron, so it shows an excitatory effect (up arrows in panel A). Overall, the excitatory and inhibitory effects will be balanced. If the source neuron leads the target neuron, the explanation will be the same.

Another explanation is using decomposition as shown in Fig D.23B,C,D. Similar to panel A, the shared activity arrives at the source neuron (dark curve) earlier than the target neuron (blue curve). The activities can be decomposed into two the shared components and the non-shared parts for each neuron as shown in panels C and D. The shared part is colored in green in panels B,C,D. The decomposition of the source neuron is in panel C, its non-shared part is the dashed dark curve. The decomposition of the target neuron is in panel D, its non-shared part is the dashed blue curve. This situation can be reduced to Supplementary D.4.2 with non-shared input. The green curve corresponds to $f_{i,j}$ in diagram Fig D.2, the source neuron's non-shared component (dark dashed curve in Fig D.21) corresponds to f_i, and the target neuron's non-shared component corresponds to f_j. As the lag in the simulation is only 20 ms, the non-shared components are very small. The non-shared components will not undermine the estimator.

D.4.13 More examples of coupling filters and jitter-based CCG

This section shows an example of Fig 5.6 by checking the shapes of the fitted coupling filter and CCG. We consider the coupling filter of neurons V1→LM (neuron id `951102686`→`951108867`). The type of the coupling filter changes from trial to trial; it can be inhibitory on some trial, and switches to excitatory on some other trials; it can be weak such as no effect, or has strong interactions like the oscillatory types. As the spike trains are noisy and the sample size of each trial is small, we pool all the trials of the same type together and re-fit the coupling filters using square windows and non-parametric method. We also compare the results with the jitter-based CCG as another verification. The results are shown in Fig D.24.

Similar to Fig 5.1, the jitter-based CCG method hardly detects weak signals such as type 1 or 2, and it can not easily distinguish the types between 0, 1, and 2. The jitter-based CCG can detect strong coupling effect such as the oscillatory types. In types 3 and 4, the shapes of the CCG curves are very close to the coupling filters. The point process regression allows us to aggregate all the information in a lag window to estimate the coupling filter with one parameter using a square window, or a few parameters using B-spline. The results will be more effective and significant than the method using pointwise statistic. In Fig D.24 type 0,1, and 2, the square window coupling filters show significant excitatory and inhibitory coupling effect, and the coupling filters are obviously distinguishable. We also estimate those effects using non-parametric fitting. For example, the shapes of the type 2 coupling filters are not strict square, but most part of the curve between lag=0 ms and lag=50 ms is below the x-axis. We admit the modeling strategy using square windows loses many details of the coupling effect, but it is effective in terms of justifying basic properties of the coupling effect and implementing for a massive dataset. Given the limited dataset and large variance from neuron to neuron, and from trial to trial, the method has to scarify non-relevant details.

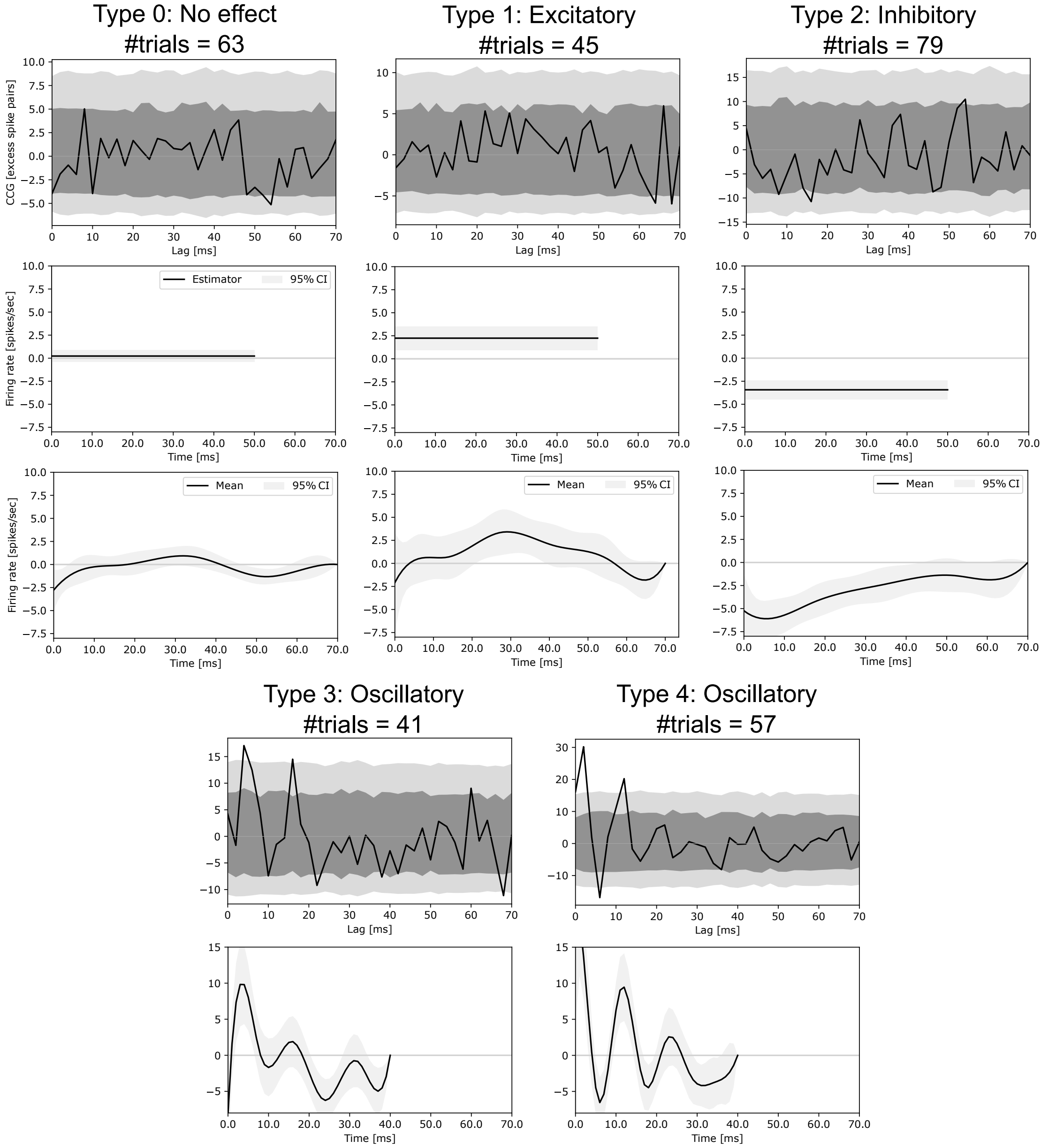


Figure D.24: **Re-fitting of coupling effect by pooling the trials of the same type.** The figure shows the jitter-based CCG and the fitted coupling filters for one pair of neurons among V1→LM (neuron id 951102686→951108867). After identifying the coupling filter types on each trial, we pool the trials of the same type together, then calculate the jitter-based CCG and estimate the coupling filters. The number of trials of each type is in the title of each plot. The details of the jitter-based CCG is the same as Fig 5.1. The second row of types 0, 1, and 2 shows the fitted coupling filters using square windows. The third row shows the fitted coupling filters using the non-parametric method.

Fig D.25 shows the fitted coupling filter using the same trials as Fig D.24 type 2 without the nuisance variable $\overline{\mathbf{s}_i}$ in Eq (5.4). As the model does not correctly remove the artifacts, the fitted coupling filter changes from inhibitory to excitatory. Since the bias is positive as shown in Fig 5.3, the model erroneously detects the inhibitory effect as excitatory, so it totally reverses the conclusion and will fail the coupling filter clustering. A similar example has already been shown in Fig 5.1. So it is important to remove the artifacts carefully.

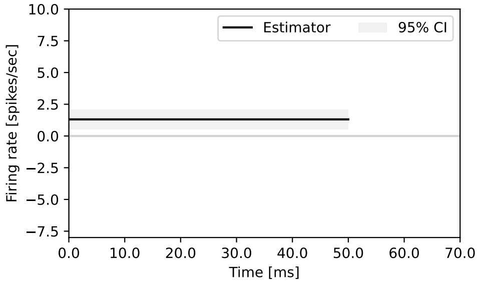


Figure D.25: **Re-fitting type 2 inhibitory coupling filter without considering fluctuating background.** This example replicated the fitting of Fig D.24 type 2 trials, but the coupling filter is fitted using a constant baseline without the nuisance variable $\overline{\mathbf{s}_i}$ in Eq (5.4). The fitted filter becomes excitatory.

D.4.14 Clustering with unbalanced weights

Our algorithm alternatively updates the coupling filter templates and identifies the coupling filter types as shown in Appendix D.3.3 Eq (D.38) and (D.39). In step Eq (D.39), unlike a typical mixture model, we do not update the portion of the coupling filter groups. This is because the coupling filters on each trial between “no effect”, “excitatory” and “inhibitory” are close. The distributions of parameters of these types do not have a clear boundary. Without fixing the group portion, those coupling filters can get merged during the clustering. In other words, the coupling filter type identification on each trial is sensitive to the group portion. The clustering algorithm is reduced to the k-means-type method. Since the group portion is fixed with equal weights for all the trials, it does not introduce trial-to-trial variance.

In this section, we verify the results using different fixed group weights. The results are shown in Fig D.26. In Fig D.26A,B, the weights are 0.20, 0.25, 0.15, 0.2, 0.2. In Fig D.26C,D, the weights are 0.25, 0.2, 0.15, 0.2, 0.2. The filter type order is the same as Fig 5.5.

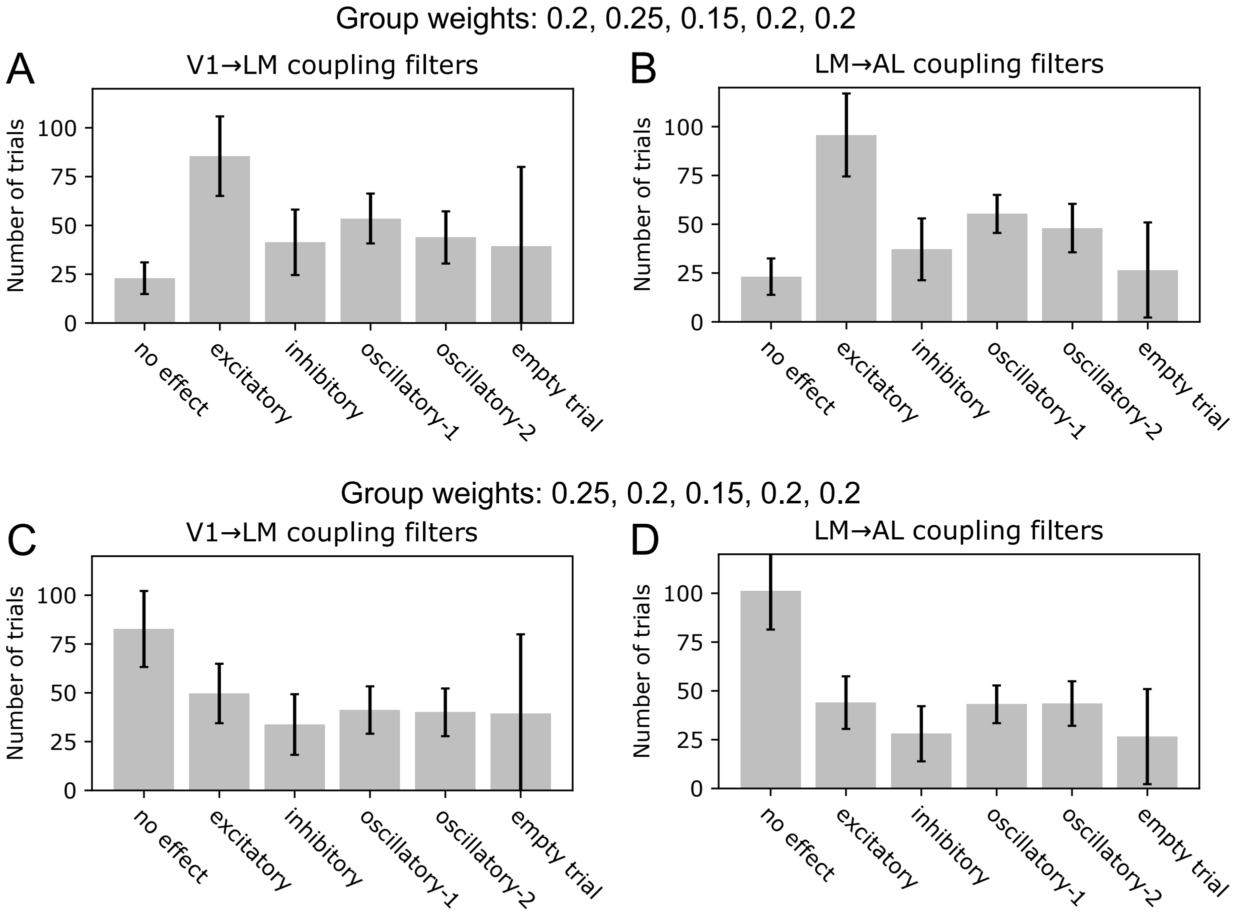


Figure D.26: **Coupling filter type frequency with different group weights.** In **A** and **B**, the weights are 0.20, 0.25, 0.15, 0.2, 0.2. In **C** and **D**, the weights are 0.25, 0.2, 0.15, 0.2, 0.2. The filter type order is the same as Fig 5.5.

D.4.15 Results with different smoothing kernel widths

As shown in Fig 5.3 and Supplementary D.4.4, the estimator is not very sensitive to the selected smoothing kernel width σ_w in Eq 5.5. If σ_w is 10 ms larger of smaller, the results do not change a lot.

The smoothing kernel width $\sigma_w = 60$ ms was selected by the plug-in estimator using a subset of samples (Appendix D.1.2). In this section, we verify the results by repeating the analysis using different kernel widths $\sigma_w = 50, 80$ ms. The results are shown in Fig D.27 in the same way as Fig D.27. The kernel width is 50 ms in Fig D.27 A, B. The kernel width is 70 ms in Fig D.27 C, D.

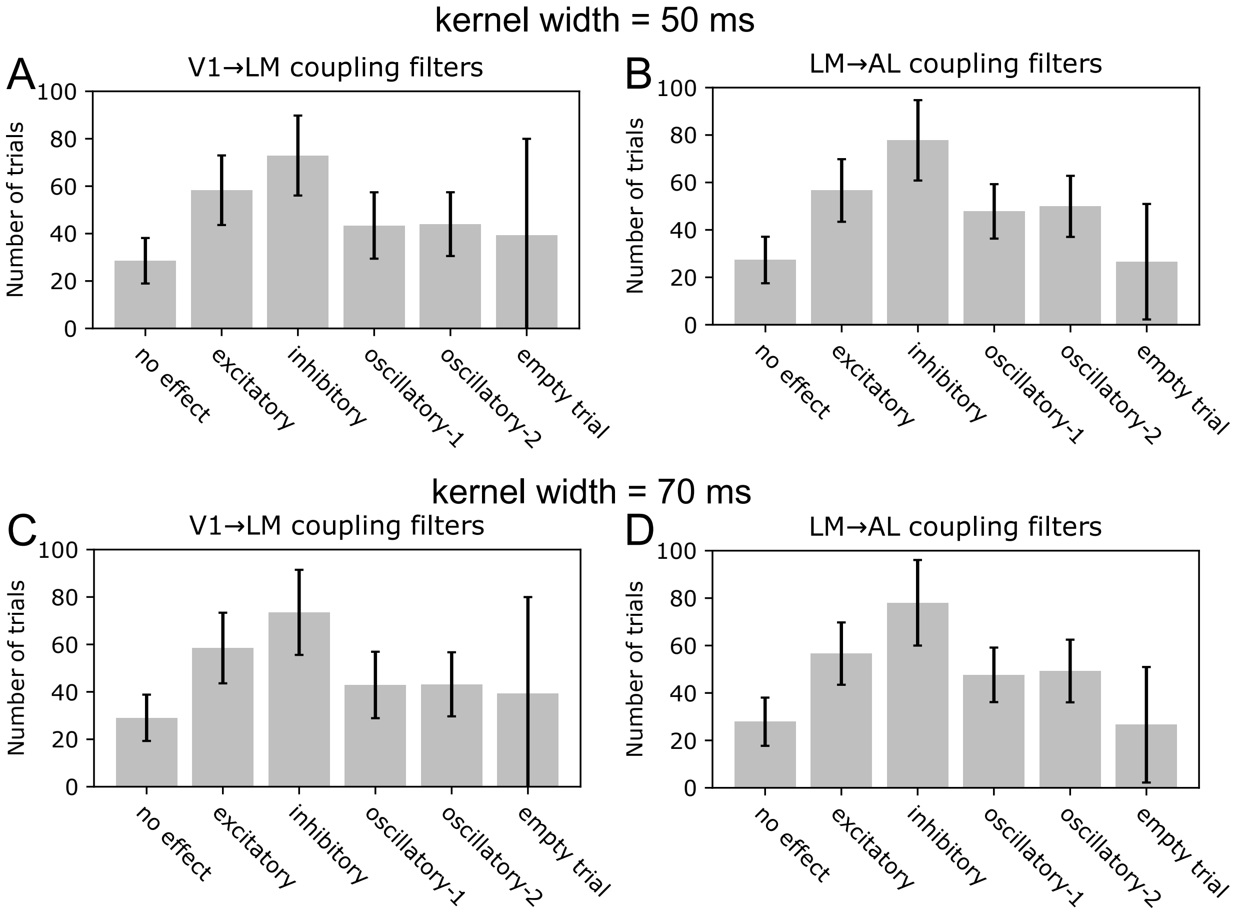


Figure D.27: **Coupling filter type frequency with different smoothing kernel widths.** Similar to Fig 5.6, we replicated the analysis with different kernel widths σ_w. In **A** and **B** $\sigma_w = 50$ ms. In **C** and **D** $\sigma_w = 70$ ms. The results do not change a lot with different kernel widths.

D.4.16 Goodness-of-fit

Goodness-of-fit test was assessed with the Kolmogorov-Smirnov (KS) test based on the time-rescaling theorem [24, 62]. The theorem states that the transformed inter-spike intervals follow the unit exponential distribution. The KS test is used to compare the empirical distribution and the target distribution. A good fit should have a straight curve along the diagonal in the Q-Q plot. Fig D.28 shows part of the results. We performed the test on all trials for a pair of neurons $i \to j$. The fitted intensity function for each trial came from the last step of the iteration in Eq (D.39). The integral transformation of the intervals was approximated by discretizing the intensity into 1 ms bins.

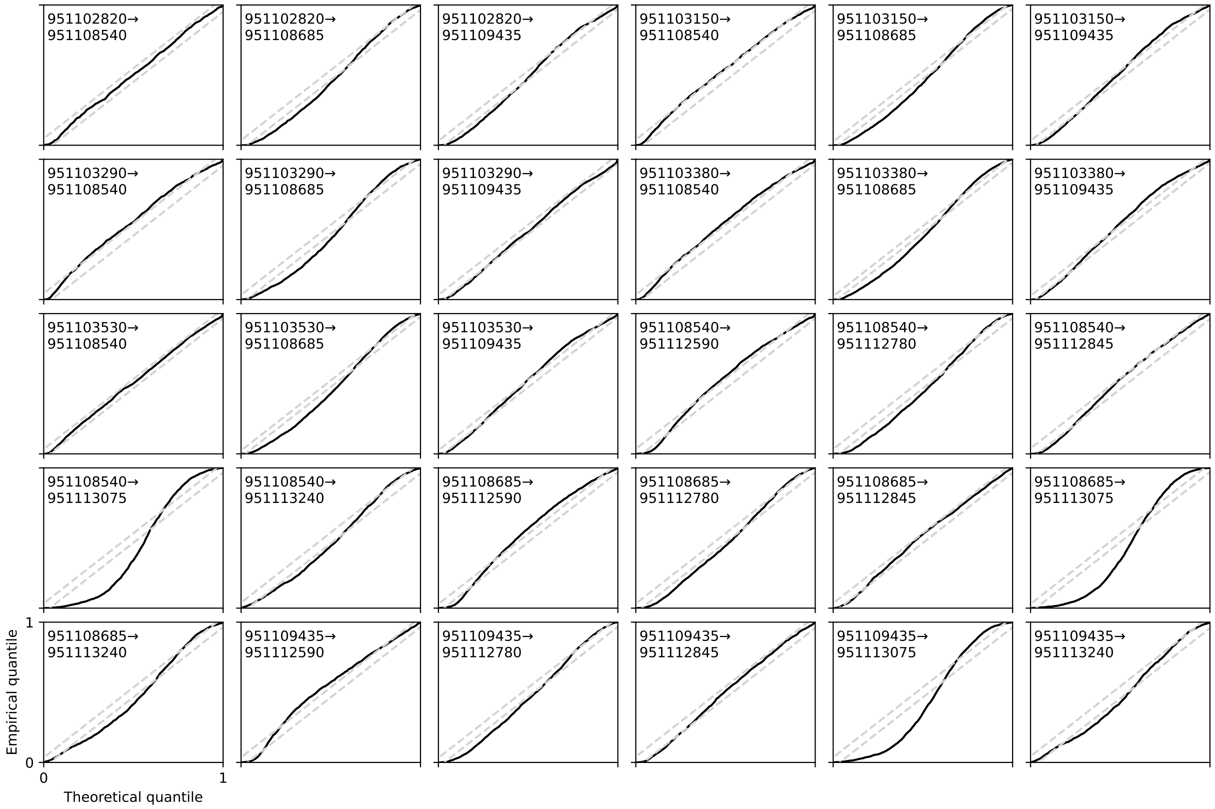


Figure D.28: **Goodness-of-fit test.** The KS tests for part of fitted filters. Each plot shows the results of a coupling filter of a pair of neurons. The neurons' identities are labeled in the corner. The test includes all trials for a pair of neurons. The grey dashed lines are 99% CI. A good fit should have a straight curve along the diagonal.

In Fig D.28, most curves are along the diagonal or have small deviations except for a few cases. For example the coupling filters with neuron `951113075` as the target neuron in the last two rows. The lack of fit is related to the self-coupling effect of the target neuron. This is why coupling filters with `951113075` as the target neuron all have a bad fit. We replicate this situation using simulations in Supplementary D.4.11. Missing modeling the self-coupling effect may result in selecting smaller smoothing kernel width, but the introduced risk or bias is small. We repeated the analysis using a larger smoothing kernel width in Supplementary D.4.15. The conclusion does not change.

D.4.17 Results using different dataset

In this section, we repeat the analysis using the conditions totally different from the analysis in the main text. The same animal `798911424` is the same. The condition numbers include `247`, `248`, `250`, `251`, `252`, `253`, `254`, `255`, `259`, `262`, `264`, `266`, `269`, `271`, `272`, `276`, `277`, `279`, `282`, `283`, `285`. These are the rest conditions of the drifting gratings. The results are shown in Fig D.29 and D.30.

The shapes of the coupling filter templates are almost identical to the Fig 5.5. The frequency of each type remains similar to Fig 5.6.

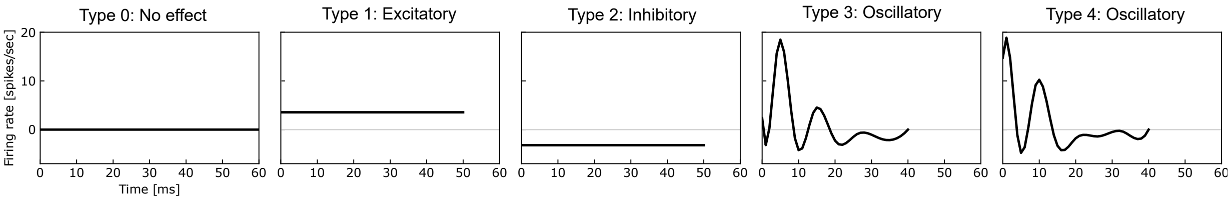


Figure D.29: **Fitted coupling filter templates.** The layout of the figure is the same as Fig 5.5 but with new dataset.

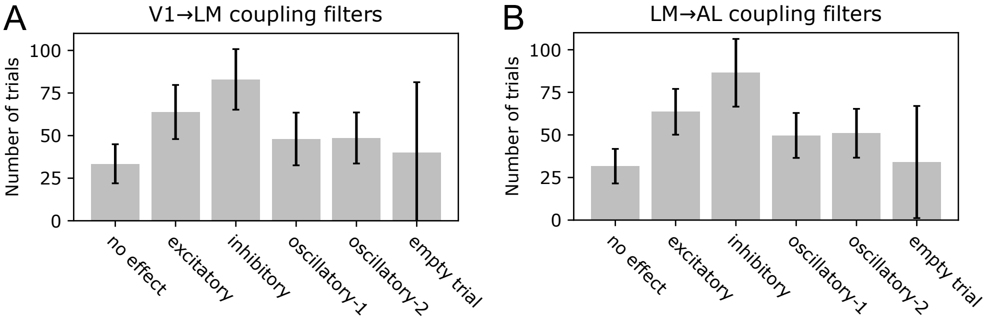


Figure D.30: **Frequency of coupling filter types of all coupling filters.** Similar to Fig 5.6, the figure counts the average frequency of coupling filters from V1→LM in **A** and coupling filters from LM→AL in **B**. The error bar is the standard deviation.

Bibliography

[1] Mara Almog and Alon Korngreen. A quantitative description of dendritic conductances and its application to dendritic excitation in layer 5 pyramidal neurons. *Journal of Neuroscience*, 34(1):182–196, 2014.

[2] Ignacio Alvarez, Jarad Niemi, and Matt Simpson. Bayesian inference for a covariance matrix. *arXiv preprint arXiv:1408.4050*, 2014.

[3] Asohan Amarasingham, Matthew T Harrison, Nicholas G Hatsopoulos, and Stuart Geman. Conditional modeling and the jitter method of spike resampling. *Journal of Neurophysiology*, 107(2):517–531, 2012.

[4] Yalda Amidi, Behzad Nazari, Saeed Sadri, Uri T Eden, and Ali Yousefi. Parameter estimation in synaptic coupling model using a point process modeling framework. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 2362–2365. IEEE, 2018.

[5] Kamilla Angelo and Troy W Margrie. Population diversity and function of hyperpolarization-activated current in olfactory bulb mitral cells. *Scientific reports*, 1(1): 1–11, 2011.

[6] Kamilla Angelo, Ede A Rancz, Diogo Pimentel, Christian Hundahl, Jens Hannibal, Alexander Fleischmann, Bruno Pichler, and Troy W Margrie. A biophysical signature of network affiliation and sensory processing in mitral cells. *Nature*, 488(7411):375–378, 2012.

[7] Bruno B Averbeck, Peter E Latham, and Alexandre Pouget. Neural correlations, population coding and computation. *Nature reviews neuroscience*, 7(5):358–366, 2006.

[8] Riccardo Barbieri, Loren M Frank, David P Nguyen, Michael C Quirk, Victor Solo, Matthew A Wilson, and Emery N Brown. Dynamic analyses of information encoding in neural ensembles. *Neural computation*, 16(2):277–307, 2004.

[9] John Barnard, Robert McCulloch, and Xiao-Li Meng. Modeling covariance matrices in terms of standard deviations and correlations, with application to shrinkage. *Statistica Sinica*, pages 1281–1311, 2000.

[10] Maurice S Bartlett. Statistical estimation of density functions. *Sankhyā: The Indian Journal of Statistics, Series A*, pages 245–254, 1963.

[11] Maurice S Bartlett. The spectral analysis of two-dimensional point processes. *Biometrika*, 51(3/4):299–311, 1964.

[12] Sam Behseta, Tamara Berdyyeva, Carl R Olson, and Robert E Kass. Bayesian correction for attenuation of correlation in multi-trial spike count data. *Journal of neurophysiology*, 101(4):2186–2193, 2009.

[13] Yoram Ben-Shaul, Hagai Bergman, Ya'acov Ritov, and Moshe 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.

[14] U. S. Bhalla and J. M. Bower. Exploring parameter space in detailed single neuron models: simulations of the mitral and granule cells of the olfactory bulb. *Journal of Neurophysiology*, 69(6):1948–1965, 1993. doi: 10.1152/jn.1993.69.6.1948. PMID: 7688798.

[15] M Bhatti and P Bracken. The calculation of integrals involving b-splines by means of recursion relations. *Applied mathematics and computation*, 172(1):91–100, 2006.

[16] Adrian G Bondy, Ralf M Haefner, and Bruce G Cumming. Feedback determines the structure of correlated variability in primary visual cortex. *Nature neuroscience*, 21(4): 598–606, 2018.

[17] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[18] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

[19] Pierre Brémaud, Laurent Massoulié, and Andrea Ridolfi. Power spectra of random spike fields and related processes. *Advances in applied probability*, 37(4):1116–1146, 2005.

[20] David R Brillinger. The spectral analysis of stationary interval functions. In *Vol. 1 Theory of Statistics*, pages 483–514. University of California Press, 1972.

[21] David R Brillinger. Cross-spectral analysis of processes with stationary increments including the stationary $G/G/\infty$ queue. *The Annals of Probability*, pages 815–827, 1974.

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

[23] Emery N Brown, Loren M Frank, Dengda Tang, Michael C Quirk, and Matthew A Wilson. A statistical paradigm for neural spike train decoding applied to position prediction from ensemble firing patterns of rat hippocampal place cells. *Journal of Neuroscience*, 18(18): 7411–7425, 1998.

[24] Emery N Brown, Riccardo Barbieri, Valérie Ventura, Robert E Kass, and Loren M Frank. The time-rescaling theorem and its application to neural spike train data analysis. *Neural computation*, 14(2):325–346, 2002.

[25] Emery N Brown, Riccardo Barbieri, Uri T Eden, and Loren M Frank. Likelihood methods for neural spike train data analysis. *Computational neuroscience: A comprehensive approach*, pages 253–286, 2003.

[26] Shawn D Burton, G Bard Ermentrout, and Nathaniel N Urban. Intrinsic heterogeneity in oscillatory dynamics limits correlation-induced neural synchronization. *Journal of neurophysiology*, 108(8):2115–2133, 2012.

[27] Nicholas T Carnevale and Michael L Hines. *The NEURON book*. Cambridge University Press, 2006.

[28] Shizhe Chen, Ali Shojaie, Eric Shea-Brown, and Daniela Witten. The multivariate hawkes process in high dimensions: Beyond mutual excitation. *arXiv preprint arXiv:1707.04928*, 2017.

[29] Shizhe Chen, Daniela Witten, and Ali Shojaie. Nearly assumptionless screening for the mutually-exciting multivariate hawkes process. *Electronic journal of statistics*, 11(1): 1207, 2017.

[30] Yu Chen, Qi Xin, Valérie Ventura, and Robert E Kass. Stability of point process spiking neuron models. *Journal of computational neuroscience*, 46(1):19–32, 2019.

[31] Zhe Chen, Sujith Vijayan, Riccardo Barbieri, Matthew A Wilson, and Emery N Brown. Discrete-and continuous-time probabilistic models and algorithms for inferring neuronal up and down states. *Neural computation*, 21(7):1797–1862, 2009.

[32] Marlene R Cohen and Adam Kohn. Measuring and interpreting neuronal correlations. *Nature neuroscience*, 14(7):811, 2011.

[33] Ann Cowling, Peter Hall, and Michael J Phillips. Bootstrap confidence regions for the intensity of a poisson point process. *Journal of the American Statistical Association*, 91 (436):1516–1524, 1996.

[34] Daryl J Daley and David Vere-Jones. *An introduction to the theory of point processes: volume I: elementary theory and methods*. Springer, 2003.

[35] Michael J Daniels and Robert E Kass. A note on first-stage approximation in two-stage hierarchical models. *Sankhyā: The Indian Journal of Statistics, Series B*, pages 19–30, 1998.

[36] A Demmler and C Reinsch. Oscillation matrices with spline smoothing. *Numerische Mathematik*, 24(5):375–382, 1975.

[37] Xinyi Deng, Emad N Eskandar, and Uri T Eden. A point process approach to identifying and tracking transitions in neural spiking dynamics in the subthalamic nucleus of parkinson's patients. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 23(4):046102, 2013.

[38] Peter Diggle. A kernel method for smoothing point process data. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 34(2):138–147, 1985.

[39] Ilaria DiMatteo, Christopher R Genovese, and Robert E Kass. Bayesian curve-fitting with free-knot splines. *Biometrika*, 88(4):1055–1071, 2001.

[40] Charles R Doss, James M Flegal, Galin L Jones, Ronald C Neath, et al. Markov chain monte carlo estimation of quantiles. *Electronic Journal of Statistics*, 8(2):2448–2478, 2014.

[41] Lea Duncker and Maneesh Sahani. Temporal alignment and latent gaussian process factor inference in population spike trains. *bioRxiv*, page 331751, 2018.

[42] Uri T Eden and Emery N Brown. Continuous-time filters for state estimation from point

process models of neural data. *Statistica Sinica*, 18(4):1293, 2008.

[43] Uri T Eden, Loren M Frank, Riccardo Barbieri, Victor Solo, and Emery N Brown. Dynamic analysis of neural encoding by point process adaptive filtering. *Neural computation*, 16(5):971–998, 2004.

[44] Michael Eichler, Rainer Dahlhaus, and Johannes Dueck. Graphical modeling for multivariate hawkes processes with nonparametric link functions. *Journal of Time Series Analysis*, 38(2):225–242, 2017.

[45] Ronald A Fisher. Frequency distribution of the values of the correlation coefficient in samples from an indefinitely large population. *Biometrika*, 10(4):507–521, 1915.

[46] Efi Foufoula-Georgiou and Dennis P Lettenmaier. Continuous-time versus discrete-time point process models for rainfall occurrence series. *Water Resources Research*, 22(4):531–542, 1986.

[47] Loren M Frank, Uri T Eden, Victor Solo, Matthew A Wilson, and Emery N Brown. Contrasting patterns of receptive field plasticity in the hippocampus and the entorhinal cortex: an adaptive filtering approach. *Journal of Neuroscience*, 22(9):3817–3830, 2002.

[48] Roberto F Galán, G Bard Ermentrout, and Nathaniel N Urban. Optimal time scale for spike-time reliability: theory, simulations, and experiments. *Journal of neurophysiology*, 99(1):277–283, 2008.

[49] Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian Data Analysis*. CRC Press, 2013.

[50] Andrew Gelman, Ben Goodrich, Jonah Gabry, and Aki Vehtari. R-squared for bayesian regression models. *The American Statistician*, 2019.

[51] Felipe Gerhard, Gordon Pipa, Bruss Lima, Sergio Neuenschwander, and Wulfram Gerstner. Extraction of network topology from multi-electrode recordings: is there a small-world effect? *Frontiers in computational neuroscience*, 5:4, 2011.

[52] Felipe Gerhard, Tilman Kispersky, Gabrielle J Gutierrez, Eve Marder, Mark Kramer, and Uri Eden. Successful reconstruction of a physiological circuit with known connectivity from spiking activity alone. *PLoS Comput Biol*, 9(7):e1003138, 2013.

[53] Felipe Gerhard, Moritz Deger, and Wilson Truccolo. On the stability and dynamics of stochastic spiking neuron models: Nonlinear hawkes process and point process glms. *PLoS computational biology*, 13(2):e1005390, 2017.

[54] Julijana Gjorgjieva, Guillaume Drion, and Eve Marder. Computational implications of biophysical diversity and multiple timescales in neurons and synapses for circuit performance. *Current opinion in neurobiology*, 37:44–52, 2016.

[55] Lindsey L Glickfeld and Shawn R Olsen. Higher-order areas of the mouse visual cortex. *Annual review of vision science*, 3:251–273, 2017.

[56] Nathan W Gouwens, Jim Berg, David Feng, Staci A Sorensen, Hongkui Zeng, Michael J Hawrylycz, Christof Koch, and Anton Arkhipov. Systematic generation of biophysically detailed models for diverse cortical neuron types. *Nature communications*, 9(1):1–13, 2018.

[57] Nathan W Gouwens, Staci A Sorensen, Jim Berg, Changkyu Lee, Tim Jarsky, Jonathan Ting, Susan M Sunkin, David Feng, Costas A Anastassiou, Eliza Barkan, et al. Classification of electrophysiological and morphological neuron types in the mouse visual cortex. *Nature neuroscience*, 22(7):1182–1195, 2019.

[58] Nathan W Gouwens, Staci A Sorensen, Fahimeh Baftizadeh, Agata Budzillo, Brian R Lee, Tim Jarsky, Lauren Alfiler, Anton Arkhipov, Katherine Baker, Eliza Barkan, et al. Toward an integrated classification of neuronal cell types: morphoelectric and transcriptomic characterization of individual gabaergic cortical neurons. *BioRxiv*, 2020.

[59] Yong Gu, Sheng Liu, Christopher R Fetsch, Yun Yang, Sam Fok, Adhira Sunkara, Gregory C DeAngelis, and Dora E Angelaki. Perceptual learning reduces interneuronal correlations in macaque visual cortex. *Neuron*, 71(4):750–761, 2011.

[60] Julie A Harris, Stefan Mihalas, Karla E Hirokawa, Jennifer D Whitesell, Hannah Choi, Amy Bernard, Phillip Bohn, Shiella Caldejon, Linzy Casal, Andrew Cho, et al. Hierarchical organization of cortical and thalamic connectivity. *Nature*, 575(7781):195–202, 2019.

[61] Matthew T Harrison, Asohan Amarasingham, and Robert E Kass. Statistical identification of synchronous spiking. *Spike timing: Mechanisms and function*, page 77, 2013.

[62] Robert Haslinger, Gordon Pipa, and Emery Brown. Discrete time rescaling theorem: determining goodness of fit for discrete time statistical models of neural spiking. *Neural computation*, 22(10):2477–2506, 2010.

[63] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.

[64] Alan G Hawkes. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1):83–90, 1971.

[65] Etay Hay, Sean Hill, Felix Schürmann, Henry Markram, and Idan Segev. Models of neocortical layer 5b pyramidal cells capturing a wide range of dendritic and perisomatic active properties. *PLoS Comput Biol*, 7(7):e1002107, 2011.

[66] Michael L Hines, Thomas Morse, Michele Migliore, Nicholas T Carnevale, and Gordon M Shepherd. Modeldb: a database to support computational neuroscience. *Journal of computational neuroscience*, 17(1):7–11, 2004.

[67] Alan L Hodgkin and Andrew F Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, 117 (4):500–544, 1952.

[68] Alan Huang, Matthew P Wand, et al. Simple marginally noninformative prior distributions for covariance matrices. *Bayesian Analysis*, 8(2):439–452, 2013.

[69] Allen Institute. *Visual Coding - Neuropixels*, 2020 (accessed 2020). URL `https://portal.brain-map.org/explore/circuits/visual-coding-neuropixels`.

[70] Hiroyuki Ito and Satoshi Tsuji. Model dependence in quantification of spike interdependence by joint peri-stimulus time histogram. *Neural computation*, 12(1):195–217, 2000.

[71] Eugene M Izhikevich. Simple model of spiking neurons. *IEEE Transactions on neural networks*, 14(6):1569–1572, 2003.

[72] Eugene M Izhikevich. Which model to use for cortical spiking neurons? *IEEE transactions on neural networks*, 15(5):1063–1070, 2004.

[73] Xiaoxuan Jia, Joshua H Siegle, Séverine Durand, Greggory Heller, Tamina Ramirez, and Shawn R Olsen. Multi-area functional modules mediate feedforward and recurrent processing in visual cortical hierarchy. *bioRxiv*, 2020.

[74] Xiaolong Jiang, Shan Shen, Cathryn R Cadwell, Philipp Berens, Fabian Sinz, Alexander S Ecker, Saumil Patel, and Andreas S Tolias. Principles of connectivity among morphologically defined cell types in adult neocortex. *Science*, 350(6264), 2015.

[75] Don H Johnson. Point process models of single-neuron discharges. *Journal of computational neuroscience*, 3(4):275–299, 1996.

[76] James J Jun, Nicholas A Steinmetz, Joshua H Siegle, Daniel J Denman, Marius Bauza, Brian Barbarits, Albert K Lee, Costas A Anastassiou, Alexandru Andrei, Çağatay Aydın, et al. Fully integrated silicon probes for high-density recording of neural activity. *Nature*, 551(7679):232–236, 2017.

[77] Robert E Kass and Valérie Ventura. A spike-train probability model. *Neural computation*, 13(8):1713–1720, 2001.

[78] Robert E Kass, Ryan C Kelly, and Wei-Liem Loh. Assessment of synchrony in multiple neural spike trains using loglinear point process models. *The annals of applied statistics*, 5(2B):1262, 2011.

[79] Robert E Kass, Uri T Eden, and Emery N Brown. *Analysis of neural data*, volume 491. Springer, 2014.

[80] Stephen Keeley, David Zoltowski, Yiyi Yu, Spencer Smith, and Jonathan Pillow. Efficient non-conjugate gaussian process factor models for spike count data using polynomial approximations. In *International Conference on Machine Learning*, pages 5177–5186. PMLR, 2020.

[81] Stephen L Keeley, David M Zoltowski, Mikio C Aoi, and Jonathan W Pillow. Modeling statistical dependencies in multi-region spike train data. *Current Opinion in Neurobiology*, 2020.

[82] Ryan Kelly. *BARS*, 2020 (accessed 2020). URL `http://www.cnbc.cmu.edu/~rkelly/code.html`.

[83] Ryan C Kelly and Robert E Kass. A framework for evaluating pairwise and multiway synchrony among stimulus-driven neurons. *Neural computation*, 24(8):2007–2032, 2012.

[84] Naomi Keren, Noam Peled, and Alon Korngreen. Constraining compartmental models using multiple voltage recordings and genetic algorithms. *Journal of neurophysiology*, 2005.

[85] Naomi Keren, Dan Bar-Yehuda, and Alon Korngreen. Experimentally guided modelling of dendritic excitability in rat neocortical pyramidal neurones. *The Journal of physiology*, 587(7):1413–1437, 2009.

[86] Sanggyun Kim, David Putrino, Soumya Ghosh, and Emery N Brown. A granger causality measure for point process models of ensemble neural spiking activity. *PLoS computational biology*, 7(3):e1001110, 2011.

[87] Seung-Jean Kim, Kwangmoo Koh, Stephen Boyd, and Dimitry Gorinevsky. \ell_1 trend filtering. *SIAM review*, 51(2):339–360, 2009.

[88] Natalie Klein, Josue Orellana, Scott L Brincat, Earl K Miller, and Robert E Kass. Torus graphs for multivariate phase coupling analysis. *The Annals of Applied Statistics*, 14(2): 635–660, 2020.

[89] Ryota Kobayashi, Shuhei Kurita, Anno Kurth, Katsunori Kitano, Kenji Mizuseki, Markus Diesmann, Barry J Richmond, and Shigeru Shinomoto. Reconstructing neuronal circuitry from parallel spike trains. *Nature communications*, 10(1):1–13, 2019.

[90] Shinsuke Koyama, Uri T Eden, Emery N Brown, and Robert E Kass. Bayesian decoding of neural spike trains. *Annals of the Institute of Statistical Mathematics*, 62(1):37, 2010.

[91] Mark A Kramer. An introduction to field analysis techniques: The power spectrum and coherence. *The Science of Large Data Sets: Spikes, Fields, and Voxels. Short Course by the Society for Neuroscience. https://www. sfn. org/~/media/SfN/Documents/Short% 20Courses/2013% 20Short% 20Course% 20II/Short% 20Course*, 202, 2013.

[92] Mark A Kramer and Uri T Eden. *Case studies in neural data analysis: a guide for the practicing neuroscientist*. MIT Press, 2016.

[93] Yu A Kutoyants. *Statistical inference for spatial Poisson processes*, volume 134. Springer Science & Business Media, 1998.

[94] Joonyeol Lee, Mati Joshua, Javier F Medina, and Stephen G Lisberger. Signal, noise, and variation in neural and sensory-motor latency. *Neuron*, 90(1):165–176, 2016.

[95] Jungah Lee, HyungGoo R Kim, and Choongkil Lee. Trial-to-trial variability of spike response of v1 and saccadic response time. *Journal of neurophysiology*, 104(5):2556–2572, 2010.

[96] Michael W Levine. The distribution of the intervals between neural impulses in the maintained discharges of retinal ganglion cells. *Biological cybernetics*, 65(6):459–467, 1991.

[97] PAW Lewis. Remarks on the theory, computation and application of the spectral analysis of series of events. *Journal of Sound and Vibration*, 12(3):353–375, 1970.

[98] Zachary F Mainen and Terrence J Sejnowski. Reliability of spike timing in neocortical neurons. *Science*, 268(5216):1503–1506, 1995.

[99] Eve Marder. Variability, compensation, and modulation in neurons and circuits. *Proceedings of the National Academy of Sciences*, 108(Supplement 3):15542–15548, 2011.

[100] Eve Marder and Adam L Taylor. Multiple models to capture the variability in biological neurons and networks. *Nature neuroscience*, 14(2):133–138, 2011.

[101] Henry Markram, Eilif Muller, Srikanth Ramaswamy, Michael W Reimann, Marwan Abdellah, Carlos Aguado Sanchez, Anastasia Ailamaki, Lidia Alonso-Nanclares, Nicolas Antille, Selim Arsever, et al. Reconstruction and simulation of neocortical microcircuitry.

Cell, 163(2):456–492, 2015.

[102] James H Marshel, Marina E Garrett, Ian Nauhaus, and Edward M Callaway. Functional specialization of seven mouse visual cortical areas. *Neuron*, 72(6):1040–1054, 2011.

[103] MathWorks. Cross power spectral density, 2020. URL `https://www.mathworks.com/help/signal/ref/cpsd.html`.

[104] Moira A Mugglestone and Eric Renshaw. A practical guide to the spectral analysis of spatial point processes. *Computational Statistics & Data Analysis*, 21(1):43–65, 1996.

[105] Cristopher M Niell and Michael P Stryker. Highly selective receptive fields in mouse visual cortex. *Journal of Neuroscience*, 28(30):7520–7536, 2008.

[106] Yoshiko Ogata. The asymptotic behaviour of maximum likelihood estimators for stationary point processes. *Annals of the Institute of Statistical Mathematics*, 30(2):243–261, 1978.

[107] Yosihiko Ogata. Statistical models for earthquake occurrences and residual analysis for point processes. *Journal of the American Statistical association*, 83(401):9–27, 1988.

[108] Murat Okatan, Matthew A Wilson, and Emery N Brown. Analyzing functional connectivity using a network likelihood model of ensemble neural spiking activity. *Neural computation*, 17(9):1927–1961, 2005.

[109] Krishnan Padmanabhan and Nathaniel N Urban. Disrupting information coding via block of 4-ap-sensitive potassium channels. *Journal of neurophysiology*, 112(5):1054–1066, 2014.

[110] Volker Pernice, Benjamin Staude, Stefano Cardanobile, and Stefan Rotter. How structure determines correlations in neuronal networks. *PLoS Comput Biol*, 7(5):e1002059, 2011.

[111] Jonathan W Pillow, Jonathon Shlens, Liam Paninski, Alexander Sher, Alan M Litke, EJ Chichilnisky, and Eero P Simoncelli. Spatio-temporal correlations and visual signalling in a complete neuronal population. *Nature*, 454(7207):995–999, 2008.

[112] Christophe Pouzat and Antoine Chaffiol. Automatic spike train analysis and report generation. an implementation with r, r2html and star. *Journal of neuroscience methods*, 181 (1):119–144, 2009.

[113] Aaditya Ramdas and Ryan J Tibshirani. Fast and flexible admm algorithms for trend filtering. *Journal of Computational and Graphical Statistics*, 25(3):839–858, 2016.

[114] Nancy Reid. The roles of conditioning in inference. *Statistical Science*, 10(2):138–157, 1995.

[115] Mohammad R Rezaei, Anna K Gillespie, Jennifer A Guidera, Behzad Nazari, Saeid Sadri, Loren M Frank, Uri T Eden, and Ali Yousefi. A comparison study of point-process filter and deep learning performance in estimating rat position using an ensemble of place cells. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 4732–4735. IEEE, 2018.

[116] Fabio Rigat, Mathisca de Gunst, Jaap van Pelt, et al. Bayesian modelling and analysis of spatio-temporal neuronal networks. *Bayesian Analysis*, 1(4):733–764, 2006.

[117] SAND8. Eighth international workshop statistical analysis of neuronal data (sand8). `http://sand.stat.cmu.edu/SAND8/index.html`, 2017. Accessed: 2021-11-28.

[118] Federico Scala, Dmitry Kobak, Shen Shan, Yves Bernaerts, Sophie Laturnus, Cathryn Rene Cadwell, Leonard Hartmanis, Emmanouil Froudarakis, Jesus Ramon Castro, Zheng Huan Tan, et al. Layer 4 of mouse neocortex differs in cell types and circuit organization between sensory areas. *Nature communications*, 10(1):1–12, 2019.

[119] Federico Scala, Dmitry Kobak, Matteo Bernabucci, Yves Bernaerts, Cathryn R Cadwell, Jesus R Castro, Leonard Hartmanis, Xiaolong Jiang, Sophie R Laturnus, Elanine Miranda, et al. Phenotypic variation within and across transcriptomic cell types in mouse motor cortex. *bioRxiv*, 2020.

[120] James G Scott, Ryan C Kelly, Matthew A Smith, Pengcheng Zhou, and Robert E Kass. False discovery rate regression: an application to neural synchrony detection in primary visual cortex. *Journal of the American Statistical Association*, 110(510):459–471, 2015.

[121] Joshua H Siegle, Xiaoxuan Jia, Séverine Durand, Sam Gale, Corbett Bennett, Nile Graddis, Greggory Heller, Tamina K Ramirez, Hannah Choi, Jennifer A Luviano, et al. Survey of spiking in the mouse visual system reveals functional hierarchy. *Nature*, 592(7852): 86–92, 2021.

[122] Bernard W Silverman. Spline smoothing: the equivalent variable kernel method. *The annals of Statistics*, pages 898–916, 1984.

[123] Anne C Smith and Emery N Brown. Estimating a state-space model from point process observations. *Neural computation*, 15(5):965–991, 2003.

[124] Anne C Smith, Joao D Scalon, Sylvia Wirth, Marianna Yanike, Wendy A Suzuki, and Emery N Brown. State-space algorithms for estimating spike rate functions. *Computational Intelligence and Neuroscience*, 2010, 2010.

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

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

[127] Nicholas A Steinmetz, Christof Koch, Kenneth D Harris, and Matteo Carandini. Challenges and opportunities for large-scale electrophysiology with neuropixels probes. *Current opinion in neurobiology*, 50:92–100, 2018.

[128] Nicholas A Steinmetz, Cagatay Aydin, Anna Lebedeva, Michael Okun, Marius Pachitariu, Marius Bauza, Maxime Beau, Jai Bhagat, Claudia Böhm, Martijn Broux, et al. Neuropixels 2.0: A miniaturized high-density probe for stable, long-term brain recordings. *Science*, 372(6539), 2021.

[129] Ian Stevenson. Tracking Advances in Neural Recording. `https://stevenson.lab.uconn.edu/scaling/`, 2021. [Online; accessed June 2021].

[130] Ian H Stevenson, James M Rebesco, Nicholas G Hatsopoulos, Zach Haga, Lee E Miller, and Konrad P Kording. Bayesian inference of functional connectivity and network struc-

ture from spikes. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 17(3):203–213, 2008.

[131] Adam L Taylor, Jean-Marc Goaillard, and Eve Marder. How multiple conductances determine electrophysiological properties in a multicompartment model. *Journal of Neuroscience*, 29(17):5573–5586, 2009.

[132] Surya Tokdar, Peiyi Xi, Ryan C Kelly, and Robert E Kass. Detection of bursts in extracellular spike trains using hidden semi-markov point process models. *Journal of computational neuroscience*, 29(1-2):203–212, 2010.

[133] Shreejoy J Tripathy, Krishnan Padmanabhan, Richard C Gerkin, and Nathaniel N Urban. Intermediate intrinsic diversity enhances neural population coding. *Proceedings of the National Academy of Sciences*, 110(20):8248–8253, 2013.

[134] Wilson Truccolo, Uri T Eden, Matthew R Fellows, John P Donoghue, and Emery 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.

[135] Valérie Ventura, Can Cai, and Robert E Kass. Trial-to-trial variability and its effect on time-varying dependency between two neurons. *Journal of neurophysiology*, 94(4):2928–2939, 2005.

[136] Giuseppe Vinci, Valérie Ventura, Matthew A Smith, and Robert E Kass. Separating spike count correlation from firing rate correlation. *Neural computation*, 28(5):849–881, 2016.

[137] Alison I Weber and Jonathan W Pillow. Capturing the dynamical repertoire of single neurons with generalized linear models. *Neural computation*, 29(12):3260–3289, 2017.

[138] Anqi Wu, Nicholas A Roy, Stephen Keeley, and Jonathan W Pillow. Gaussian process based nonlinear latent structure discovery in multivariate spike train data. *Advances in neural information processing systems*, 30:3496, 2017.

[139] Wei Wu and Anuj Srivastava. An information-geometric framework for statistical inferences in the neural spike train space. *Journal of Computational Neuroscience*, 31(3): 725–748, 2011.

[140] Byron M Yu, John P Cunningham, Gopal Santhanam, Stephen I Ryu, Krishna V Shenoy, and Maneesh Sahani. Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. *Journal of neurophysiology*, 102(1):614–635, 2009.

[141] Pengcheng Zhou, Shawn D Burton, Adam C Snyder, Matthew A Smith, Nathaniel N Urban, and Robert E Kass. Establishing a statistical link between network oscillations and neural synchrony. *PLoS computational biology*, 11(10):e1004549, 2015.