Novel mathematical neural models for visual attention

Kang Li

31 October 2016

PhD Thesis

Submitted to the PhD School of the Faculty of Science,
University of Copenhagen

Department of Mathematical Sciences
Department of Psychology
University of Copenhagen
Denmark

Kang Li

Department of Mathematical Sciences
Department of Psychology
University of Copenhagen
Universitetsparken 5
2100 Copenhagen Ø

kang@math.ku.dk / kang.li.gnak@gmail.com

The PhD thesis was submitted to the PhD School of the Faculty of Science, University of Copenhagen on 31st of October, 2016.

Principal supervisor:
Susanne Ditlevsen
Department of Mathematical Sciences, University of Copenhagen

Co-supervisor:
Søren Kyllingsbæk
Department of Psychology, University of Copenhagen

Assessment committee:

Bo Markussen (chair)
Department of Mathematical Sciences, University of Copenhagen, Denmark

Patricia Reynaud-Bouret
Laboratoire J. A. Dieudonné, Université de Nice Sophia-Antipolis, France

Hans Colonius
Department of Psychology, Oldenburg University, Germany

ISBN: 978-87-7078-920-2

Abstract

Visual attention has been extensively studied in psychology, but some fundamental questions remain controversial. We focus on two questions in this study. First, we investigate how a neuron in visual cortex responds to multiple stimuli inside the receptive field, described by either a response-averaging or a probability-mixing model. Second, we discuss how stimuli are processed during visual search, explained by either a serial or a parallel mechanism.

Here we present novel mathematical methods to answer the psychology questions from a neural perspective, combining the formulation of neural explanations for the visual attention theories and spiking neuron models for single spike trains. Statistical inference and model selection are performed and various numerical methods are explored. The designed methods also give a framework for neural coding under visual attention theories. We conduct both analysis on real data and theoretical study with simulations.

Our findings are shown in separate projects. First, the probability-mixing model is favored over the response-averaging model, shown by analysis on experimental data from monkeys. Second, both parallel and serial processing exist, with a tendency of being parallel in the beginning and a tendency of being serial later on, shown by another set of experimental data from monkeys. Third, we show that the probability-mixing and response-averaging model can be separated and parameters can be successfully estimated for either model in a more realistic biophysical system, supported by simulation study. Finally, we present the decoding of multiple temporal stimuli under these visual attention theories, also in a realistic biophysical situation with simulations.

Dansk resume

Visuel opmærksomhed er et intenst forskningsfelt indenfor kognitiv psykologi, men visse fundamentale spørgsmål er stadig kontroversielle. Vi fokuserer på to spørgsmål i denne afhandling. Først undersøger vi hvordan en neuron i det visuelle cortex reagerer når flere stimuli præsenteres i neuronens receptive felt, enten ved brug af en model hvor responsen antages at være et gennemsnit over responsen ved de enkelt stimuli, eller ved brug at en sandsynlighedsmodel, hvor responsen er som ved et af de enkelte stimuli, og hvilket stimuli bestemmes udfra en vis sandsynlighedsfordeling. Dernæst diskuterer vi hvordan visuelle stimuli bearbejdes, enten ved den såkaldte serielle eller den såkaldte parallelle mekanisme.

Vi præsenterer nye matematiske metoder til at svare på disse psykologiske spørgsmål ud fra et neuralt perspektiv, hvor vi kombinerer neurale forklaringsmodeller for de forskellige teorier indenfor visuel opmærksomhed med statistiske modeller for målinger af neural elektrisk aktivitet. Vi laver statistisk inferens og modelvalg og forskellige numeriske metoder undersøges. Metoderne sætter også en ramme for neural afkodning under de forskellige teorier fra visuel opmærksomhed. Vi analyserer både eksperimentelt data og laver teoretiske studier med simuleret data.

Vores resultater præsenteres i forskellige manuskripter. Først finder vi at den sandsynlighedsteoretiske model passer bedre til data end modellen, der tager gennemsnit over responser, hvilket vises ved analyser af neurofysiologisk data fra aber. Dernæst finder vi at både seriel og parallel processering synes at finde sted, med en tendens til at være parallel lige efter stimulus præsenteres, og senere lader processeringen til at blive mere seriel, vist ved analyse af et andet neurofysiologisk datasæt målt på aber. Efterfølgende viser vi at de to modeller kan adskilles og parametre kan estimeres for begge modeller i et mere realistisk biologisk system, understøttet af et simulationsstudie. Endelig præsenterer vi afkodning af multiple tidsligt varierende stimuli under disse teorier fra visuel opmærksomhed, også i en realistisk biofysisk situation ved simulationer.

Contents

Chapter 1

Introduction

Visual attention is a cognitive process during which important visual information from a complicated environment is efficiently selected. Visual attention is usually studied in psychology via behavioral tasks, by proposing hypothesis and analyzing the behavioral results. Though intuitive and straightforward, the behavioral analysis does not explain how attention works on a more detailed biological level. This is where this research project is motivated. We want to explain the macro behavior of visual attention from the micro perspective of single neurons, the processing units of our nervous system.

The *receptive field* (RF) of a neuron in the visual system is defined as the spatial area in which stimulation changes the firing pattern of the neuron. In higher visual processing areas, for example visual area V5/MT and IT, the RF of a neuron can be very large (Smith et al., 2001; Gattass et al., 2005), integrating multiple stimuli from the visual world (Orhan and Ma, 2015). The interesting question is how the relevant information is obtained from these multiple stimuli from a single neuron perspective. To answer this, we revisit two classical and fundamental questions in the field of visual attention: how the multiple competing stimuli are attended and how they are processed. Both questions have been long debated with opposing theories in psychology. We reexplain these theories from a neural perspective.

The psychological Theory of Visual Attention (TVA) proposed by Bundesen (1990) provides a unified computational mechanism to explain visual cognition and attentional selection. In TVA, attentional selection is performed by the two mechanisms of filtering and pigeonholing, which gives the attentional weights of objects and the processing speed of categorization though mathematical equations. TVA explains a wide range of empirical findings from behavioral studies (Bundesen, 1990). The Neural Theory of Visual Attention (NTVA) (Bundesen et al., 2005) is an interpretation of TVA at a single neuron level. The attentional weights and the processing speed from TVA correspond to the probabilities of neurons attending to the objects and neuronal firing rate. NTVA explains a wide range of neurophysiological findings under visual attention (Bundesen et al., 2005). Here we base our mathematical neural models on NTVA and apply the models to both empirical and simulated single-neuron data.

1.1 Neural methods for visual attention

Attending multiple stimuli

Two opposing models have been proposed to explain neuronal attention to multiple stimuli in a RF. In the *response-averaging* model (Reynolds et al., 1999), the response of a neuron to multiple stimuli is a weighted average of responses to each single stimulus. By contrast, in the *probability-mixing* model based on NTVA (Bundesen et al., 2005), a neuron responds to each single stimulus with probabilities. Suppose there are K stimuli, denoted as $S = \{S_k\}_{k=1,2,...,K}$. The response of a neuron to any stimulus S_k is $I(S_k)$. Following the response-averaging model, $I(S) = \sum_{k=1}^{K} I(S_k)\beta_k$, where $\{\beta_k\}$ are weights satisfying $\sum_{k=1}^{K} \beta_k = 1$. Following the probability-mixing model, $I(S) = I(S_k)$ with probability α_k, where we have $\sum_{k=1}^{K} \alpha_k = 1$. The response-averaging model treats all stimuli as a single integrated object, while the probability-mixing model preserves each individual stimulus.

Processing multiple stimuli

How multiple stimuli are processed has been explained by two opposing mechanisms, the serial and parallel processing (see Bundesen and Habekost (2008); Nobre and Kastner (2013) for reviews). Though extensive studies have been conducted using behavioral methods, this question remains highly controversial. We introduce a novel neural perspective to distinguish between serial and parallel processing, under the hypothesis of probability mixing where a neuron only attends to a single stimulus at a time. In serial processing, all neurons with the same RF attend to the same stimulus at any given time, and they may switch together to another stimulus after finishing processing the current stimulus. In parallel processing, each neuron may attend to any stimulus independently, and all stimuli are processed in parallel.

Stimulus and observation

The stimulus and observation are the input and output of our analysis. Think of the brain as a black box containing the visual attention hypotheses. We input the stimulus into the black box, which then outputs neural observation. We construct mathematical models for the stimulus and observation, and study the brain black box. Various types of stimuli can be used. In experiments, we can use isolated static images, moving bars, random dot patterns containing many small moving dots, and so on. In simulations, we can use deterministic stimulus described by functions, e.g. a constant or sinusoidal function, or stochastic stimuli following a stochastic process, e.g. a Brownian motion. Regarding neural observation, we consider the neural spike train, which is a discrete sequence of times indicating neuronal spiking events. The spiking observation is extremely noisy and big variance is observed both within a trial and across trials. Traditional methods depend on spiking rate averaged across trials, avoiding the variance and extracting important information. However, by averaging across trials we may lose useful information. It could be the variance that actually matters. Our methods are implemented by modeling each single spike train, in the above neural

frameworks of visual attention.

1.2 Research contribution

Our aim is to explore, develop and verify neural models for visual attention at a single neuron level. In particular, we want to compare the probability-mixing and response-averaging model and distinguish between the parallel and serial processing with neural explanations. We also want to explore the neural code relating neural spikes to complicated external stimuli by applying visual attention theories.

The main contribution of this research is developing novel probabilistic models and statistical methods for the visual attention hypotheses using observations of each single neuron at each single trial. We combine the visual attention theories from psychology and the spike train models from computational neuroscience. The results provide conclusion and further insights regarding both psychological meanings and mathematical properties. We present our contribution in four papers, whether published, submitted or in preparation. Paper I compares the response-averaging and probability-mixing model using experimental data. Paper II studies serial and parallel processing and how to distinguish between the two, also using experimental data. Both papers use relatively simple and approximate models for neural observations. Papers III and IV, on the other hand, employ more realistic but challenging models and are conducted in a more theoretical way with simulation studies. Paper III provides systematic methods of parameter estimation and model selection for the response-averaging and probability-mixing models. Paper IV works on stimulus reconstruction discussing both serial and parallel processing hypotheses.

The thesis is structured as follows. Chapter 2 introduces the mathematical background including various basic topics on probabilistic models and statistical methods used in the thesis. Chapter 3 presents the application of mathematical models to neuroscience and psychology. Models for spike trains and methods for visual attention are shown first, and then the topic of neural coding under visual attention hypotheses is discussed. Chapter 4 gives the overview of the published papers and on-going projects, as well as future prospects. Finally, all papers and manuscripts are attached.

Chapter 2

Mathematical Background

This chapter gives a brief introduction to various mathematical topics.

2.1 Point processes

For the topic of point process, we refer the reader to a comprehensive introduction by Daley and Vere-Jones (2003). A point process describes the discrete occurrences of phenomena in either time or space, in one or more dimensions. The realization of a point process contains subsequent isolated points. We consider one dimensional point processes describing events on a time line. For this one dimensional process, we have the following four equivalent descriptions:

(a) counting measures;

(b) nondecreasing integer-valued step functions;

(c) sequences of points;

(d) sequences of intervals.

We now provide notation for the (a), (c) and (d) descriptions. A point process described by a sequence of time points is denoted by

$$\{t_i\}_{i=0,1,2,\ldots}, \tag{2.1}$$

with $t_0 < t_1 < t_2 < \cdots$. Alternatively, a sequence of time intervals is denoted by

$$\{\tau_i\}_{i=1,2,\ldots}, \tag{2.2}$$

where $\tau_i = t_i - t_{i-1}$ is the ith interval. Let $N(A)$ denote the number of occurrences inside a time interval A:

$$N(A) = \#\{i; t_i \in A\}. \tag{2.3}$$

Further, we simplify $N([0,t))$ on the interval $[0,t)$ as $N(t)$ for a positive time $t > 0$.

The conditional intensity function (CIF) describes the probability of an event occurring in a short interval around some time t, conditional on the past event times $H_t = \{t_i; t_i < t\}$. The CIF can be defined by

$$\lambda(t|H_t) = \lim_{\Delta t \to 0} \frac{\Pr(N(t+\Delta t) - N(t) = 1|H_t)}{\Delta t}. \tag{2.4}$$

The interval Δt is so small that there can at most be one event within Δt. It follows that the probability of observing an event in the time interval $[t, t+\Delta t)$ is given by $\lambda(t|H_t)\Delta t$.

Consider a finite point process realization $\{t_i\}_{i=0,1,\ldots,N}$ inside an observation interval $[0,T]$ satisfying $0 \leq t_0$ and $t_N \leq T$. It can be shown (Daley and Vere-Jones, 2003) that the likelihood of such realization, with relevant model parameters θ, is given by

$$L(\{t_i\}_{i=0,1,\ldots,N}; \theta) = \left[\prod_{i=1}^{N} \lambda(t_i|H_t;\theta)\right] \exp\left\{-\int_0^T \lambda(s|H_s;\theta)ds\right\}. \tag{2.5}$$

2.1.1 Examples of point processes

Example 2.1.1 (Renewal process)**.** Consider the interval description $\{\tau_i\}$ for a point process. If all the inter-event intervals are independent and identically distributed (i.i.d.), then this point process is a renewal process. In such processes, every time an event occurs, the probability of a subsequent event resets (the process starts over).

The likelihood of renewal processes are extremely simple since all observed data are i.i.d. Here we show three examples of the renewal process, with different features and applications.

Example 2.1.2 (Poisson process)**.** If the inter-event interval τ follows an exponential distribution with a constant rate parameter λ, we then have a Poisson process with rate λ, which is a widely used point process in many scientific areas. By letting the CIF of a point process be a constant λ, we obtain a Poisson process with rate λ. A central property of the Poisson process is being memoryless with a constant CIF. We also obtain a realization of Poisson process when we sample points uniformly from a fixed interval.

Example 2.1.3 (Gamma process)**.** If the inter-event interval τ follows a Gamma distribution with a shape parameter α and a rate parameter β, then the renewal process is a Gamma process. With two parameters, the Gamma process gives more flexibility than the Poisson process and the CIF is a function of time.

Example 2.1.4 (First-passage times of Wiener process)**.** If we let the distribution of τ be an inverse Gaussian distribution, the renewal process becomes a sequence of first-passage times of a Wiener diffusion process (Chhikara, 1988). The two parameters of the inverse Gaussian distribution are related to the drift and diffusion parameters of the Wiener process and the constant passage threshold.

For more complicated situations, it might not be appropriate to assume the renewal property. We then require more general point processes.

Example 2.1.5 (Cox process)**.** If the CIF of the Poisson process is not a constant but also a stochastic process itself, then we have a Cox process (Cox, 1955). In applications the stochastic CIF often models a latent state in a system, and the point process events are the observed data. Cox processes include a large collection of such latent-observed models depending on the choice of the stochastic CIF.

Example 2.1.6 (Hawkes process)**.** The Hawkes process (Hawkes, 1971) is often referred to as the self-exciting process, because its CIF contains direct effects of the history of event times. An example of such CIF after discretization is given by

$$\lambda(t|H_t) = \lambda_0 \exp\left\{ a_0 + \sum_{j=1}^{h} a_j (N(t-(j-1)\Delta t) - N(t-j\Delta t)) \right\}, \tag{2.6}$$

where λ_0 is a base intensity parameter, a_0 is an offset parameter, and $\{a_j\}_{j=1,2,\ldots,h}$ are weight parameters for the history covering events up to $h\Delta t$ ago subject to an appropriate discretization time Δt. Depending on the weights, we obtain different effects of the past events. For example, the effect at a certain delay time could be excitatory if the weight is greater than 0, inhibitory if less than 0, or nonexistent if equal to 0. By adjusting the weight parameters, we may obtain a large variety of self-exciting processes. Papers I and II employ Hawkes processes of similar types.

2.2 Diffusion processes

A diffusion process is a continuous time stochastic process satisfying the strong Markov property with almost surely continuous sample paths. The classic literature by Karlin and Taylor (1981) introduces diffusion processes highlighting boundary problems and various differential equations. Diffusion processes are usually formulated as the solution of stochastic differential equations in modern literature (see e.g. Oksendal (2013) for an introduction).

2.2.1 Stochastic differential equation

Many phenomena arising in various disciplines contain noise and deterministic differential equations do not suffice. Stochastic differential equations (SDEs) extend the models to a stochastic context. A typical SDE driven by Gaussian white noise is given by

$$dX_t = \mu(x,t)dt + \sigma(x,t)dB_t, \tag{2.7}$$

where $\mu(x,t)$ and $\sigma(x,t)$ are continuous deterministic functions, and $B(t)$ denotes the standard Brownian motion which brings stochasticity. The term $\frac{dB_t}{dt} = W_t$ gives Gaussian white noise and we have

$$B(t_2) - B(t_1) \sim N(0, |t_2 - t_1|), \tag{2.8}$$

i.e. a normal distribution with mean 0 and variance $|t_2 - t_1|$. The solution of the SDE is a diffusion process $\{X(t); t \geq 0\}$. The function $\mu(x,t)$ gives the infinitesimal drift coefficient and $\sigma^2(x,t)$ gives the infinitesimal variance (diffusion coefficient).

Example 2.2.1 (Wiener process)**.** A very simple and widely used diffusion process is the Wiener process, which is the solution to an SDE for which the drift and diffusion coefficients are constant:

$$dX_t = \mu dt + \sigma dB_t. \tag{2.9}$$

If $\mu = 0$ and $\sigma = 1$ we obtain the standard Wiener process, equal to the standard Brownian motion.

Example 2.2.2 (Ornstein-Uhlenbeck process)**.** Another popular diffusion process is the Ornstein-Uhlenbeck (OU) process, the solution to the following SDE

$$dX_t = \theta(\mu - X_t)dt + \sigma dB_t, \tag{2.10}$$

where $\theta > 0$. The feature of the OU process is a tendency of moving towards a mean value μ, which finds application in many biological and financial areas.

Example 2.2.3 (Feller process)**.** The Feller process (Feller et al., 1951) is given by the solution of the SDE

$$dX_t = \theta(\mu - X_t)dt + \sigma\sqrt{X_t}dB, \tag{2.11}$$

with $\mu, \theta > 0$. The notable feature is that the sample path is non-negative, because when $X(t)$ approaches 0 the variance becomes very small and $X(t)$ will then evolve towards μ. The Feller process was originally introduced by Feller et al. (1951) to model population growth. It is also called the Cox-Ingersoll-Ross process (Cox et al., 1985) in financial literature.

2.2.2 Transition functions and related PDEs

Consider the diffusion process $\{X(t); t \geq 0\}$ given by the solution of the SDE

$$dX_t = \mu(x,t)dt + \sigma(x,t)dB_t. \tag{2.12}$$

Denote the transition distribution function for the transition from $X(s) = x$ to $X(t) = y$ as

$$P(x,y,s,t) := \Pr(X(t) \leq y | X(s) = x), \tag{2.13}$$

for $t \geq s$, and the transition density function as

$$p(x,y,s,t) := \frac{dP(x,y,s,t)}{dy}. \tag{2.14}$$

The transition density function $p(x,y,s,t)$ satisfies the Kolmogorov backward equation, given by the PDE

$$\frac{\partial p}{\partial s} = \frac{1}{2}\sigma^2(x,s)\frac{\partial^2 p}{\partial x^2} + \mu(x,s)\frac{\partial p}{\partial x}. \tag{2.15}$$

The transition distribution function $P(x, y, s, t)$ also satisfies the same PDE

$$\frac{\partial P}{\partial s} = \frac{1}{2}\sigma^2(x, s)\frac{\partial^2 P}{\partial x^2} + \mu(x, s)\frac{\partial P}{\partial x}. \tag{2.16}$$

In addition to the backward equation, the transition density $p(x, y, s, t)$ also satisfies the Kolmogorov forward equation

$$\frac{\partial p}{\partial t} = \frac{1}{2}\frac{\partial^2}{\partial y^2}\left[\sigma^2(y, t)p\right] - \frac{\partial}{\partial y}\left[\mu(y, t)p\right], \tag{2.17}$$

also referred to as the Fokker-Planck equation. Unlike the backward equation, the transition distribution function $P(x, y, s, t)$ for a general diffusion process does not satisfy the same forward equation. However, if we set a lower reflecting boundary $x^- < x$ for the diffusion process, then we can derive a forward PDE for $P(x, y, s, t)$ given by (Li et al., 2016; Iolov et al., 2014; Hurn et al., 2005)

$$\frac{\partial P}{\partial t} = \frac{1}{2}\sigma^2(y, t)\frac{\partial^2 P}{\partial y^2} - \mu(y, t)\frac{\partial P}{\partial y}. \tag{2.18}$$

This equation for $P(x, y, s, t)$ is used in both paper III and IV.

The pertinent variables for the backward equations are s and x, the initial state of the transition, while for the forward equations are t and y, the final state of the transition. So the names "backward" and "forward" follow. The forward and backward equations are used in different scenarios depending on whether we know the starting state or the final state of the transition.

The solution of these PDEs gives the transition probability function, which is useful for e.g. parameter estimation. However, analytical solution exists only for simple diffusion processes such as the Brownian motion or the OU process. For a general diffusion process where the drift and diffusion coefficients are given by complicated nonlinear functions, we need to solve the PDEs numerically, which requires boundary conditions. The forward and backward equations we have here belong to parabolic equations, whose numerical solution requires a time condition and two space boundary conditions. The finite-difference method (Press, 2007) of the parabolic equation amounts to solving a tridiagonal system, for which there exist efficient algorithms and parallelism on GPUs applies (for example see a performance benchmark by Andreetta et al. (2015)).

The boundary conditions of $p(\cdot)$ or $P(\cdot)$ are related to the diffusion processes. We can set a upper and lower boundary for the diffusion path, in general either absorbing or reflecting, which gives corresponding boundary expressions for $p(\cdot)$ or $P(\cdot)$. On the time dimension, we set a initial condition for the forward equations or a final condition for the backward equations. See paper III for examples of numerically solving the two forward equations by setting appropriate boundary conditions.

2.2.3 First passage time

The first passage time (FPT) of a diffusion process (Redner, 2001; Ricciardi and Sato, 1990) is the first time when the diffusion path passes a threshold different from the initial value. Consider the diffusion process $\{X(t); t \geq 0\}$. If $X(s) = x$ at time s and we set a upper threshold $E > x$, the FPT is defined by

$$T_E := \inf\{t > s; X(t) \geq E, X(s) = x\}. \tag{2.19}$$

Denote the distribution function of the FPT by

$$F(t, s) = \Pr(T_E \leq t), \tag{2.20}$$

and the density function of the FPT by

$$f(t, s) = \frac{dF(t, s)}{dt}. \tag{2.21}$$

Obtaining the FPT probabilities is important for parameter estimation using first passage observation data. Here we introduce two methods to calculate the FPT probability functions using PDEs and integral equations (IEs), respectively. Papers III and IV employ these methods.

PDE method

The key idea is to link the FPT probability to the transition probability of diffusion processes. The probability that the diffusion path starting from time s has not yet reached the threshold E at time t is the FPT survival probability $1 - F(t, s)$. Meanwhile, it is also the probability of transitions starting from $X(s) = x$ to any value below the boundary $X(t) < E$, $P_E(x, E, s, t)$, subject to an absorbing boundary at E. We use the subscript E to denote the absorbing boundary at E. The absorbing boundary ensures that the $P_E(x, E, s, t)$ here only contains diffusions not yet reached the threshold during interval (s, t). Thus, we have $1 - F(t, s) = P_E(x, E, s, t)$, and the expanded expression gives

$$f(s, t) = \frac{\partial F(t, s)}{\partial t} = -\frac{\partial P_E(x, E, s, t)}{\partial t} = -\frac{\partial}{\partial t}\int_{-\infty}^{E} p_E(x, y, s, t)dy. \tag{2.22}$$

This links the transition distribution and density function to the FPT distribution and density function. The transition functions can be obtained by numerically solving the forward or backward equations with an absorbing boundary at E (with other appropriate boundary conditions). Then the FPT density function is calculated via numerical integration and differentiation.

IE method

In the IE approach, we rely on the Volterra equations of the first and second kind. The FPT density function is inside the integral term. Here we use the original threshold-free transition

density function, $p(x, y, s, t)$, which describes the transition from $X(s) = x$ to $X(t) = y$ subject to no boundary.

The first-kind Volterra IE (Fortet equation) combines the FPT density $f(t, s)$ with $p(x, y, s, t)$ using the law of total probability:

$$p(x, E, s, t) = \int_s^t f(t', s)p(E, E, t', t)dt'. \tag{2.23}$$

The second-kind Volterra IE is given by

$$f(t, s) = -2\psi(x, E, s, t) + 2\int_s^t f(t', s)\psi(E, E, t', t)dt', \tag{2.24}$$

where

$$\psi(x, y, s, t) = \frac{\partial}{\partial t}\int_{-\infty}^{y} p(x, y', s, t)dy'. \tag{2.25}$$

The advantage of the second-kind IE is that it provides better numerical stability and accuracy by overcoming a singularity issue when $t \to s$ (Li et al., 2016; Paninski et al., 2008).

The solution of the IEs directly gives the FPT density $f(t, s)$. For the numerical solution, we need to set an initial condition where $f(s, s) = 0$. Then $f(t, s)$ at any time $t > s$ can be evaluated through forward evolution of the IE.

Note that in order to solve the IEs efficiently, we need analytical expression for the threshold-free transition density $p(x, y, s, t)$. Furthermore, for the second-kind IE we need analytical expression for the time-derivative of the space-integration of $p(x, y, s, t)$, i.e. $\psi(x, y, s, t)$. Fortunately, for the Weiner and the OU processes, the explicit analytical expression for $p(x, y, s, t)$ exists and is a Gaussian. See for example the SDEs in papers III and IV.

2.3 Parameter estimation

A central part of statistical inference is parameter estimation, where we obtain the optimal guess of the parameters for a statistical model based on observed data. The estimation theory is well-established in statistics.

2.3.1 Frequentist and Bayesian

The frequentist and Bayesian are two perspectives for the parameter inference. From a freqentist point of view, the parameters of some model are fixed and the observed data are random variables, and we need to give an optimal guess of the parameters based on the data. A typical way is to find the maximum likelihood estimator which gives the distribution that best describes the data (maximizing the likelihood). The uncertainty of the estimator is explained by the confidence interval. The Bayesian point of view, on the other hand, treats the parameters also as random variables like observed data. The uncertainty of the estimation

is explained via the parameter distribution conditional on the data. Bayesian inference aims at finding this parameter distribution, typically through Monte Carlo sampling. Instead of the full distribution, we can also find the optimal value that maximizes the parameter distribution, which is called maximum a posteriori (MAP).

2.3.2 Maximum likelihood estimation

Denote the observed data as $z = \{z_1, z_2, \ldots, z_n\}$ and the parameters of a model as θ. The likelihood function of the data is defined by the joint probability:

$$L(z; \theta) = P(z|\theta). \tag{2.26}$$

The likelihood function is seen as a function of parameters θ. For discrete models, we use the probability mass function and for continuous models, we use the probability density function. In practice, $P(\cdot)$ can contain both at the same time depending on the models. The log-likelihood is usually used:

$$\ell(z; \theta) = \log L(z; \theta). \tag{2.27}$$

In maximum likelihood estimation (MLE), we aim at finding the optimal θ values that maximize the (log-) likelihood function:

$$\hat{\theta} = \arg\max_{\theta} \ell(z; \theta), \tag{2.28}$$

which is called the ML estimator. The MLE is particularly appealing for parameter estimation since, under certain conditions of the model, the ML estimator possesses the following properties as the data size n go to infinity:

1. Consistency: the ML estimator converges to the true parameter θ^* in probability, $\hat{\theta} \xrightarrow{P} \theta^*$;

2. Asymptotic normality: the distribution of $\hat{\theta}$ converges to a normal distribution, $\hat{\theta} \xrightarrow{d} N(\theta^*, \sigma_\theta^2)$, where the asymptotic variance σ_θ^2 can be calculated from the Fisher information;

3. Efficiency: the ML estimator, among all well-behaved estimators, has the smallest variance.

The search of the ML estimator $\hat{\theta}$ falls into the discipline of numerical optimization (Nocedal and Wright, 2006), for which the mathematical theories are extensively studied and there exist various types of algorithms. Mostly used are the iterative methods that converge to the solution within a finite number of steps. Some algorithms require calculating the derivatives, i.e. the Jacobian and the Hessian matrix, while others are derivative-free. A common problem of numerical optimization is the local minimum problem, where the solution falls to a local minimum or saddle point and the true global minimum is missed. Global optimization algorithms (Horst et al., 2000) are some of the efforts taken to fight against the local minimum

problem. Performing multiple local optimizations with different initial values are commonly used by people, which is also a central idea for some global optimization algorithms. To obtain better results, we can first perform the global optimization, and then use the global optimum as the initial point and perform the local algorithm to achieve a higher accuracy.

2.3.3 Bayesian inference

In Bayesian inference, we aim at finding the distribution of parameters conditional on data, $P(\theta|z)$, called the posterior distribution. Following Bayes' theorem, we have

$$P(\theta|z) = \frac{P(z|\theta)P(\theta)}{P(z)} \propto P(z|\theta)P(\theta). \tag{2.29}$$

The distribution of parameters $P(\theta)$ is called the prior distribution, representing our prior knowledge or proposal of the parameters, and $P(z|\theta)$ is the likelihood of data conditional on the choice of parameters, which is identical to the likelihood function in MLE.

A common method to find the full posterior distribution is the Markov chain Monte Carlo (MCMC) approach, where parameters are sampled via a Markov chain and are then plugged into the Metropolis-Hastings rejection algorithm (Hastings, 1970). Another method, somewhat corresponding to the MLE, is to find the optimal parameter that maximize the posterior distribution, called the MAP estimator:

$$\hat{\theta}_{MAP} = \arg\max_{\theta} \{\log P(\theta|z)\}. \tag{2.30}$$

Searching for the MAP estimator can be done by numerical optimization methods.

2.3.4 Expectation Maximization and log-sum-exp

In many cases, there are unobserved or latent data during parameter inference. Sometimes the model itself includes latent variables that are not observable, and sometimes the data quality is low and some observations are simply missing. The ordinary MLE calculates the marginal likelihood of only the observed data, which is often intractable due to e.g. integrals or numerical issues. The expectation maximization (EM) algorithm (Dempster et al., 1977) seeks to maximize the marginal likelihood iteratively with the help of the complete likelihood, the joint probability of observed and latent data.

Denote z' the latent data, and the complete likelihood is given by $P(z, z'|\theta)$. The EM algorithm consists of iterations. Given a starting position θ_0 at iteration 0, the following two steps are performed in subsequent iterations $i = 1, 2, \ldots$:

1. (E-step) Obtain the expectation of the log-likelihood of complete data, with respect to the conditional distribution of z' given z and previous parameter estimates θ_{i-1}:

$$Q(\theta|\theta_{i-1}) = \mathbb{E}_{z'|z,\theta_{i-1}} \left[\log P(z, z'|\theta)\right]. \tag{2.31}$$

2. (M-step) Find the optimal parameters maximizing $Q(\theta|\theta_{i-1})$:

$$\theta_i = \arg\max_{\theta} Q(\theta|\theta_{i-1}), \tag{2.32}$$

which is the estimates for the current iteration i.

It can be shown that following the EM iterations the marginal likelihood is non-decreasing.

A common situation, where the EM algorithm applies, is fitting mixture models. The marginal likelihood of a mixture model always contains product of sums, which becomes log of sums during MLE with the log-likelihood. This usually causes numerical over- or under-flow issues. With the EM algorithm, the indexes of mixture are treated as the latent variable and the log of sums disappears in the expectation $Q(\theta|\theta_{i-1})$.

An alternative way to overcome the numerical issue in mixture models is the log-sum-exp trick (Press, 2007). Consider a situation where we want to calculate the log of sums, $\log \sum_i x_i$, where x_i are calculated from some distribution according to a model. If the data size is big, directly calculating x_i can give very big and intractable results, while calculating $\log x_i$ provides elegant and tractable results. We can calculate the log of sums by providing only $\log x_i$:

$$\log \sum_i x_i = a + \log \sum_i \exp\{\log x_i - a\}, \tag{2.33}$$

where $a = \max_i(\log x_i)$. The numerical issues are then avoided. Though both EM and log-sum-exp overcome the numerical issue, we find EM is statistically more efficient (with smaller variance) through data augmentation than direct MLE with the marginal likelihood; see Paper III for examples of comparison.

2.4 Model selection and checking

Apart from parameter estimation for a particular model, in statistical inference we are often faced with a large collection of models and we need to perform model selection to find the best model. This extends the idea of optimization from the parameter space to a broader model space. The most widely used model selection methods are the Akaike information criterion (AIC) and the Bayesian information criterion (BIC); see Burnham and Anderson (2002) for a detailed discussion regarding AIC, BIC and other information theoretic approaches. For the best-fitting model after model selection, we also want to assess its goodness of fit (GOF) by performing model checking (model assessment, validation).

2.4.1 AIC and BIC

Akaike information criterion

Denote our model of interest by M with parameters θ, and the number of parameters equals K. Denote by G the truth for data generation (with no parameters). The observed data is

denoted by $z = \{z_1, z_2, \ldots, z_n\}$ with data size n.

The AIC is based on an approximately unbiased estimation of the Kullback-Leibler (KL) divergence between M and G, $I(G, M)$, using the likelihood evaluated at the ML estimator (Akaike, 1974). Asymptotically as $n \to \infty$, if the model M is sufficiently "good" in the sense that the KL divergence between M and G is small enough, we have

$$\log L(z; \hat{\theta}) - K = C - \hat{\mathbb{E}}_{\hat{\theta}}[I(G, M_{\hat{\theta}})]. \tag{2.34}$$

$\hat{\theta}$ is the ML estimator for θ, and $M_{\hat{\theta}}$ is the model M under the ML estimator. $L(z; \hat{\theta})$ evaluates the likelihood for the ML estimator. The right hand side contains a constant C and the estimator of the relative expected KL divergence between M and G. The AIC is given by

$$AIC = -2 \log L(z; \hat{\theta}) + 2K, \tag{2.35}$$

which is a linear transformation of $\hat{\mathbb{E}}_{\hat{\theta}}[I(G, M_{\hat{\theta}})]$ with scale 2. Thus, a smaller AIC means smaller KL divergence between the considered model M and the truth G.

Bayesian information criterion

The BIC (Schwarz et al., 1978) is derived through the posterior distribution of the model M given the data z. Under uniform prior for all models, we have

$$P(M|z) \propto P(z|M)p(M) \propto p(z|M) = \int P(z|\theta, M)P(\theta|M)d\theta. \tag{2.36}$$

The BIC is given by

$$BIC = -2 \log P(z|M). \tag{2.37}$$

For data size $n \to \infty$, the BIC can be approximated using the ML estimator $\hat{\theta}$ for model M as:

$$BIC \approx -2 \log L(z; \hat{\theta}) + K \log n + C, \tag{2.38}$$

where the likelihood $L(\cdot)$ is calculated using the model M. The constant C is omitted in practical calculations. The BIC turns out equivalent to the minimum description length (MDL) proposed by Rissanen (1998).

Model probability

AIC stands for the relative KL information and BIC is related to the posterior distribution of the model M. A result for both AIC and BIC is the expression for the posterior probability of the ith model M_i:

$$P(M_i|z) \propto \exp\left\{-\frac{1}{2}\Delta_i\right\}, \tag{2.39}$$

where Δ_i is the difference between the AIC (BIC) of M_i and the minimal AIC (BIC). Then we have the weight (probability) of each model among all considered models:

$$w_i = \frac{\exp\left\{-\frac{1}{2}\Delta_i\right\}}{\sum_j \exp\left\{-\frac{1}{2}\Delta_j\right\}}. \tag{2.40}$$

An important consequence is that for both AIC and BIC among different models, only the difference Δ_i matters. For two models, if $\Delta = 2$, then the stronger model is 2.718 times as likely as the weak model. For a difference $\Delta = 7$, the stronger model becomes 33.12 times as likely. A rule of thumb is that if the difference between two models is greater than 10, then the result is strongly significant meaning no support for the weaker model (Burnham and Anderson, 2002).

2.4.2 Model checking

Model selection gives the stronger model and shows how much more likely the stronger model is than the other weaker ones. However, model selection does not show how well the model fits the data. Here we introduce some generic methods for assessing GOF.

Uniformity test

The uniformity test comes from a well-known transformation. With the ML estimator $\hat{\theta}$, we calculate

$$u_i = \Pr(z \leq z_i|\hat{\theta}), \tag{2.41}$$

for each data point $z_i, i = 1, 2, \ldots$. If the model under its ML estimator is correct in the sense that it describes the data sufficiently well, then the residuals $\{u_i\}$ will approximately follow the standard uniform distribution $U(0, 1)$. Thus, $\{u_i\}$ are called uniform residuals. Evaluating the uniformity of those residuals grants us a GOF assessment. To check for the equality of the reference standard uniform distribution to the empirical residual distribution, we may employ Quantile-Quantile (QQ) plots or probability-probability (PP) plots for intuitive and straightforward comparison. In addition, the Kolmogorov-Smirnov (KS) test serves as a statistical hypothesis test for comparing two distributions. The calculation of $\{u_i\}$ requires calculating the cumulative distribution function (CDF), whether analytically or numerically.

RMSD

In many cases, it may be difficult or expensive to calculate the CDF using a model, but easy to obtain the prediction of the data or the prediction of some statistic of the data. Usually in such data there are an input part and an output part, $z = \{z^{in}, z^{out}\}$. Suppose that we can easily obtain

$$\hat{z}^{out} = \mathbb{E}[z^{out}|z^{in}, \hat{\theta}], \tag{2.42}$$

the prediction of z^{out} given z^{in}, or

$$\hat{S}(z^{out}) = \mathbb{E}[S(z^{out})|z^{in}, \hat{\theta}], \tag{2.43}$$

the prediction subject to some statistic $S(\cdot)$. To assess the goodness of prediction, we can use the root mean square deviation (RMSD; or root mean square error, RMSE) between the

prediction and the observation:

$$RMSD_{\hat{\theta}} = \sqrt{\mathbb{E}[(z^{out} - \hat{z}^{out})^2]} \tag{2.44}$$

$$RMSD^S_{\hat{\theta}} = \sqrt{\mathbb{E}[(S(z^{out}) - \hat{S}(z^{out}))^2]}. \tag{2.45}$$

Cross validation

If we calculate the ML estimator from some data and apply model checking to the same data, we can run into the problem of overfitting, obtaining better GOF assessment than we should have. A common method to avoid overfitting is to perform cross validation. A k-fold cross validation means separating data randomly into k parts and we run model fitting for k times. Every time we systematically leave one part out, and fit the model using the other $k-1$ parts. Then we check the GOF of the fitted model using the left out part. The GOF results are then merged together from all k runs. The cross validation procedure applies to both uniformity test and RMSD evaluation.

2.5 State-space models

State-space models (SSMs), also known as hidden Markov models (HHMs), are widely used hidden-observed models in many areas. A SSM consists of a hidden Markov chain $\{X_t; t \geq 0\}$ and observed data $\{Y_t; t \geq 0\}$ depending on $\{X_t\}$. We consider SSMs in discrete time, i.e. $t = 0, 1, \ldots$. In some cases, we set a maximal time T representing the observation interval: $t = 0, 1, \ldots, T$. We denote time-adjacent variables by $X_{a:b} = \{X_t; a \leq t \leq b\}$ and $Y_{a:b} = \{Y_t; a \leq t \leq b\}$. Denote by θ the parameters contained in the SSM.

Since $\{X_t\}$ is a Markov chain, we have

$$P(X_{t+1}|X_{0:t}) = P(X_{t+1}|X_t). \tag{2.46}$$

For the observation $\{Y_t\}$ we consider two types of models. The first is the standard HHM where $\{Y_t\}$ are independent conditional on $\{X_t\}$:

$$P(Y_{t+1}|Y_{0:t}, X_{0:t+1}) = P(Y_{t+1}|X_{t+1}). \qquad \text{(Model I)} \tag{2.47}$$

The second is a dependent HMM where $\{Y_t\}$ are not independent even conditional on $\{X_t\}$:

$$P(Y_{t+1}|Y_{0:t}, X_{0:t+1}) = P(Y_{t+1}|X_{t+1}, Y_{0:t}), \qquad \text{(Model II)} \tag{2.48}$$

referred to as the Markov-switching model (Hamilton and Raj, 2013; Krishnamurthy and Ryden, 1998). Figure 2.1 illustrates the structure of the two models. Model I has a more simple structure and is more widely used in applications. In many other applications (see e.g. Paper IV; Hamilton (1989); Kim et al. (1999)), we will have to preserve the dependency between $\{Y_t\}$ and use Model II. If $\{X_t\}$ are discrete variables we have a discrete SSM, and if $\{X_t\}$ follow a continuous distribution we have a continuous SSM. The discrete SSM is much easier to handle than the continuous one due to finite and traversable state space.

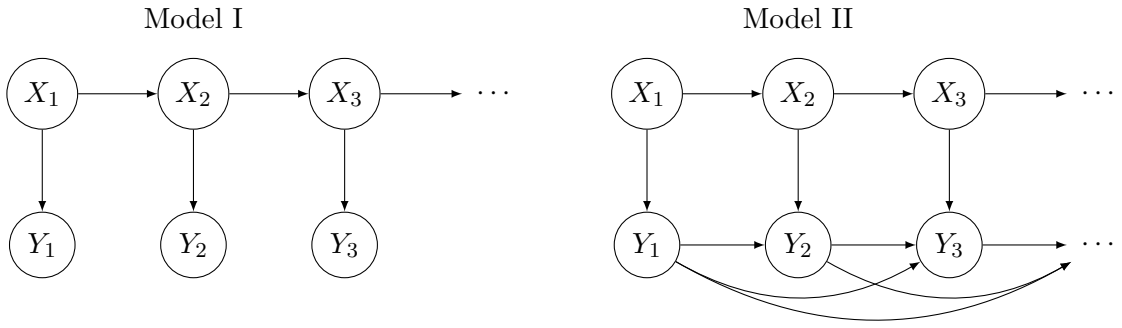


Figure 2.1: Diagrams of Model I and Model II.

In the following sections, we introduce general sequential Monte Carlo (SMC) methods and algorithms for the continuous SSM. The discrete version is discussed later. All the SMC methods here apply to both Model types I and II. See Kantas et al. (2015) for a recent review of SMC methods for SSMs and Cappé et al. (2009) for comprehensive theories in statistical inference of SSMs.

We start by introducing the following notation:

$$X_{0:t}|Y_{0:t} = y_{0:t} \sim p(x_{0:t}|y_{0:t}); \quad \text{(Filtering)} \tag{2.49}$$
$$X_t|Y_{0:t} = y_{0:t} \sim p(x_t|y_{0:t}); \quad \text{(Filtering at } t\text{)} \tag{2.50}$$
$$X_t|Y_{0:T} = y_{0:T} \sim p(x_t|y_{0:T}); \quad \text{(Smoothing at } t\text{)} \tag{2.51}$$
$$Y_{0:T} \sim p(y_{0:T}). \quad \text{(Likelihood)} \tag{2.52}$$

2.5.1 Filtering

Here we assume the parameters θ are known. Our goal is to obtain the posterior distribution of the hidden states $p(x_{0:t}|y_{0:t})$, which is a high-dimensional distribution for a long time series. We decompose the filtering probability as follows

$$p(x_{0:t}|y_{0:t}) = \frac{p(x_{0:t-1}|y_{0:t-1})}{p(y_t|y_{0:t-1})} p(x_t|x_{t-1}) p(y_t|x_t, y_{0:t-1}). \tag{2.53}$$

For Model I the term $p(y_t|x_t, y_{0:t-1}) = p(y_t|x_t)$. In the following we assume Model I for simplicity. The decomposition grants us a sequential way to sample the filtering distribution. Suppose a proposal distribution where x_t can be easily sampled, denoted by $q(x_t|x_{t-1}, y_t)$ for $t > 0$ and $q_0(x_0|y_0)$ for $t = 0$. The filtering distribution can be sampled using a sequential importance sampling (SIS) method, where we sample x_t from $q(\cdot)$ and the importance weights are given by

$$w_0(x_0) = \frac{p_0(x_0)p(y_0|x_0)}{q_0(x_0|y_0)} \tag{2.54}$$
$$w_t(x_t) = \frac{p(x_t|x_{t-1})p(y_t|x_t)}{q(x_t|x_{t-1}, y_t)}. \tag{2.55}$$

Sampling at each time t with size N (N "particles"), the full filtering distribution can be explored. One problem in SIS using a finite number of sampling size is degeneracy, meaning that as time goes on the variance of the N importance weights approaches 0 and very few particles make a difference. To counter this, we resample particles at each iteration, removing unimportant samples, which gives the sequential importance sampling resampling (SISR) method (Doucet et al., 2000) shown in Algorithm 2.5.1. Replacing the proposal distribution $q(x_t|x_{t-1}, y_t)$ by the transition density of the hidden Markov chain, $p(x_t|x_{t-1})$, the weight $w_t(x_t)$ reduces to $p(y_t|X_t)$, and the corresponding SISR is referred to as the bootstrap filter (Gordon et al., 1993). In addition to resampling, another method attempting to further avoid particle degeneracy is using auxiliary variables and calculating one-step ahead weights. The yielded algorithm is called the auxiliary particle filter (APF) (Pitt and Shephard, 1999).

Algorithm 2.5.1 SISR

Initialization: At $t = 0$ for all $i \in \{1, 2, \ldots, N\}$:

1: Sample $X_0^i \sim q_0(x_0|y_0)$

2: Calculate $W_0^i = w_0(X_0^i)$ and $\widetilde{W}_0^i = \frac{W_0^i}{\sum_i^N W_0^i}$

Iteration: At $t = 1, 2, \ldots$ for all $i \in \{1, 2, \ldots, N\}$:

3: Resample using $\left\{\widetilde{W}_{t-1}^i\right\}$, giving $\left\{\widetilde{X}_{0:t-1}^i\right\}$

4: Sample $X_t^i \sim q(x_t|\widetilde{X}_{t-1}^i, y_t)$ and $X_{0:t}^i = \{\widetilde{X}_{0:t-1}^i, X_t^i\}$

5: Calculate $W_t^i = w_t(X_t^i)$ and $\widetilde{W}_t^i = \frac{W_t^i}{\sum_i^N W_t^i}$

Using the Monte Carlo samples, we then have the following approximations for the filtering and likelihood calculation (Kantas et al., 2015; Cappé et al., 2009). The filtering distribution is approximated using particles as

$$\hat{p}(\mathrm{d}x_{0:t}|y_{0:t}) = \sum_{i=1}^{N} \widetilde{W}_t^i \delta_{X_{0:t}^i}(\mathrm{d}x_{0:t}), \tag{2.56}$$

where δ is the Dirac measure. The posterior mean is often used as the prediction of the hidden states:

$$\hat{\mathbb{E}}[g(X_{0:t})|y_{0:t}] = \sum_{i=1}^{N} \widetilde{W}_t^i g(X_{0:t}^i) \tag{2.57}$$

for some relevant function $g(\cdot)$. In addition to the full filtering distribution, we can also approximate the filtering distribution at t by marginalization:

$$\hat{p}(\mathrm{d}x_t|y_{0:t}) = \sum_{i=1}^{N} \widetilde{W}_t^i \delta_{X_t^i}(\mathrm{d}x_t). \tag{2.58}$$

For the approximation of the filtering distribution at t, univariate kernel density smoothing over $\{X_t^i\}$ can be employed with weights $\{\widetilde{W}_t^i\}$. Another result from SISR is the approximation of the likelihood at t:

$$\hat{p}(y_t|y_{0:t-1}) = \frac{1}{N} \sum_{i=1}^{N} W_t^i, \tag{2.59}$$

using the un-normalized weights $\{W_t^i\}$. The full likelihood is given by

$$\hat{p}(y_{0:T}) = \hat{p}(y_0) \prod_{t=1}^{T} \hat{p}(y_t|y_{0:t-1}). \tag{2.60}$$

2.5.2 Smoothing

Again we assume the parameters θ are known. The goal of smoothing is to obtain the distribution of X_t conditional on all observations $\{y_{0:T}\}$, $p(x_t|y_{0:T})$. A straightforward way is a simple marginalization after filtering up to T:

$$\hat{p}(\mathrm{d}x_t|y_{0:T}) = \sum_{i=1}^{N} \widetilde{W}_T^i \delta_{X_t^i}(\mathrm{d}x_t). \tag{2.61}$$

If the observation is not up to T but $t+l$ with some lag l, we still have the approximation

$$\hat{p}(\mathrm{d}x_t|y_{0:T}) \approx \sum_{i=1}^{N} \widetilde{W}_{t+l}^i \delta_{X_t^i}(\mathrm{d}x_t), \tag{2.62}$$

which is called the fixed-lag smoothing (Doucet et al., 2000). The theory behind is the so-called forgetting property of SSMs (Cappé et al., 2009).

The smoothing through marginalization endures problems due to particle degeneracy and resampling. The variety of X_t becomes extremely limited after a large l. The method below aims at directly calculating the smoothing probability without using marginalization. The algorithm is based on the following decomposition:

$$p(x_t|y_{0:T}) = p(x_t|y_{0:t}) \int p(x_{t+1}|x_t) \frac{p(x_{t+1}|y_{0:T})}{\int p(x_{t+1}|x_t) p(x_t|y_{0:t}) \mathrm{d}x_t} \mathrm{d}x_{t+1}. \tag{2.63}$$

Observe that $p(x_t|y_{0:t})$ is the filtering density at t and $p(x_{t+1}|y_{0:T})$ is the smoothing density at $t+1$. This provides a method for calculating the smoothing distribution: We first perform an ordinary forward filtering to obtain $p(x_t|y_{0:t})$ for all $t = 0, 1, \ldots, T$, and then a backward smoothing to obtain $p(x_{t+1}|y_{0:T})$ for $t = T-1, T-2, \ldots, 0$. This method is referred to as the forward-filtering backward-smoothing (FFBS) algorithm. In particle implementation, we approximate the integrals with Monte Carlo samples, and the smoothing weights are given by

$$\widetilde{V}_t^i = \widetilde{W}_t^i \sum_{j=1}^{N} \frac{p(x_{t+1}^j|x_t^i) \widetilde{V}_{t+1}^j}{\sum_{l=1}^{N} p(x_{t+1}^j|x_t^l) \widetilde{W}_t^l}. \tag{2.64}$$

The approximation of the smoothing distribution using FFBS is:

$$\hat{p}(\mathrm{d}x_t|y_{0:T}) = \sum_{i=1}^{N} \widetilde{V}_t^i \delta_{X_t^i}(\mathrm{d}x_t). \tag{2.65}$$

Alternatives to the FFBS are the generalized two-filter (Briers et al., 2010) and the forward smoothing algorithm (Del Moral et al., 2010).

2.5.3 Parameter estimation for SSMs

In the above we have assumed the parameters θ are known. Here we briefly introduce methods to estimate θ for a general SSM.

One method is to evaluate the pseudo-marginal, the approximation of the marginal likelihood $L(y_{0:T};\theta) = p(y_{0:T}|\theta)$ using the SMC method (2.60) shown before. The ML estimator, given by

$$\hat{\theta} = \arg\max_{\theta} \log L(y_{0:T};\theta), \tag{2.66}$$

can be obtained by iterative numerical optimization algorithms. In situations where the gradient $\nabla_\theta \log L(y_{0:T};\theta)$ can be explicitly evaluated at θ_k for iteration k using e.g. the Fisher information, we can perform the steepest gradient ascent algorithm (Cappé et al., 2009). When the gradient is not explicitly available, we can evaluate it numerically using the finite-difference method. Furthermore, the gradient-free algorithms are also available to obtain the ML estimator. Note that for MLE with SMC, the original SISR is not suitable because it yields discontinuous likelihood function evaluations due to resampling. Techniques (Hürzeler and Künsch, 2001; Malik and Pitt, 2011) should be taken to overcome the discontinuous problem. In addition to MLE, the pseudo-marginal likelihood can be used inside the Baysian inference framework, for calculating the Metropolis-Hasting rejection probabilities (Lin et al., 2000).

Another method is to maximize the pseudo-marginal using the EM algorithm based on Monte Carlo sampling (MCEM) (Wei and Tanner, 1990). For a general SSM, at iteration k in EM the expectation $Q(\theta|\theta_{k-1})$ with respect to the distribution $p(x_{0:T}|\theta_{k-1}, y_{0:T})$ can not be obtained analytically. The MCEM approach calculates $Q(\theta|\theta_{k-1})$ by integrating over the SMC approximation of the filtering distribution $p(x_{0:T}|\theta_{k-1}, y_{0:T})$ with parameters θ_{k-1}. The estimates θ_k at step k can then be obtained through optimization.

2.5.4 SMC with parameter learning

In many applications, we are interested in the filtering or smoothing distribution, but we have unknown parameters θ or the parameters θ are assumed changing over time. In the SMC procedure, we will also estimate parameters by on-line parameter learning. To this end, we can simply augment the state space and define the new states as $X'_t = \{X_t, \theta_t\}$, i.e., the parameters are also part of the states. Let the proposal of θ_t be

$$\theta_t \sim N(\theta_{t-1}, \sigma^2), \tag{2.67}$$

i.e., propagation with Gaussian white noise. Then ordinary particle filters e.g. SISR and APF can be employed over the new states X'_t. The filtering distribution is the joint distribution $p(x_{0:t}, \theta_{0:t}|y_{0:t})$. Both the state distribution $p(x_{0:t}|y_{0:t})$ and the parameter distribution $p(\theta_{0:t}|y_{0:t})$ can be obtained by marginalization. Besides filtering, the corresponding smoothing distributions for x_t and θ_t can also be obtained from $p(x_t, \theta_t|y_{0:T})$.

However, the artificial propagation of θ using Gaussian noise with some arbitrary variance σ^2 introduces information loss over time (Liu and West, 2001). To overcome this, Liu and West

(2001) proposed an extension of the APF where θ are propagated following a kernel density smoothing approach. It then follows that the parameters θ_t are sampled with

$$\theta_t \sim N(\phi\theta_{t-1} + (1-\phi)\bar{\theta}_{t-1}, h^2 v_{t-1}), \tag{2.68}$$

where $\bar{\theta}_{t-1}$ and v_{t-1} are the mean and the variance of the posterior $p(\theta_{t-1}|y_{0:t-1})$, evaluated by particle approximation. The constants $\phi = (3\delta - 1)/2\delta$ and $h^2 = 1 - \phi^2$ are evaluated using a discount factor $\delta \in (0, 1]$, typically around 0.95 to 0.99. The discount factor δ defines the kernel location shrinkage (Liu and West, 2001) and scaling size by changing the mean and the variance of the Gaussian. This kernel smoothing approach was later extended by Rios and Lopes (2013) to overcome possible issues of parameter degeneracy (Carvalho et al., 2010).

2.5.5 Discrete and hybrid SSM

Before, we have been focusing on continuous SSMs. We have a discrete SSM if the states $\{X_t\}$ follow discrete distributions with some probability mass function $P(X_{0:t} = x_{0:t}) = f(x_{0:t})$. Discrete SSMs are much easier to deal with since we can explicitly calculate the filtering and smoothing distributions by traversing all possible discrete states, assuming, of course, the number of states is not too big. The likelihood can be obtained likewise, which enables us to perform accurate parameter estimation via MLE or EM.

In other cases, we may need to use hybrid SSMs whose state space contains both continuous and discrete distributions. We will again have to perform particle approximation for the continuous distributions. Regarding the discrete ones, we have two options: treat a discrete distribution just like a continuous one and let a particle contain one discrete value, or marginalize out the discrete state by a summation over all possible values. The appropriate option can be used depending on specific situations.

The SMC methods for discrete, continuous and hybrid SSMs apply to both models of type I and II, with only a trivial difference in evaluating $p(y_t|x_t, y_{0:t})$ during forward filtering. Paper II employs a discrete SSM with three layers, and Paper IV uses a hybrid SSM augmented with parameters. Both are of Model II type. The SSM in Paper II has very weak dependence within $\{Y_t\}$ conditional on $\{X_{0:t}\}$ while the SSM in Paper IV has relatively stronger and longer dependence.

Chapter 3

Neural Models for Visual Attention

Neurons are the basic processing units in our nerve system. The function of a neuron is to generate, receive and propagate signals. Neurons store signals in the form of action potentials, also called spikes, which can travel along neuronal synapses from one neuron to the next. A temporal sequence of spikes is called a spike train. The generation of spike trains is related to external stimuli. Here we only consider visual stimuli, and the processing of the stimuli are described by theories in visual attention. Spike train models and visual attention theories are the two basic components for our neural encoding models.

Neural coding (Rieke, 1999; Brown et al., 2004) refers to the connection between external stimuli and neural responses (e.g. spike trains). In neural encoding, we construct models mapping from stimuli to neural responses, through parameter estimation and model selection. In neural decoding, by contrast, we infer the stimuli based on neural responses using the constructed encoding model.

In this chapter we first introduce probabilistic models and statistical methods for spike trains and visual attention. Then we discuss in general the construction of encoding models and possible methods for stimulus decoding. The neural coding here is performed for visual stimuli under visual attention theories.

3.1 Spike train models

Neurons generate spikes through membrane voltage, which is controlled by various ion channels across the membrane. The voltage is extremely dynamic, affected by presynaptic spikes as well as the neuron itself. A spike is formed when the voltage rapidly rises very high and falls back to normal in a short time interval. From a mathematical point of view, a spike train is a time series of spiking events with a hidden underlying dynamical system for the membrane voltage. The influential Hodgkin-Huxley (HH) model by Hodgkin and Huxley (1952) describes the voltage with ion currents through four differential equations, which is formulated following biophysical characteristics of the membrane. Though developed more

than 60 years ago, the HH model is still frequently used nowadays for studying neuronal dynamics. However, for situations where we only have spike train observations, the HH model is difficult to deal with due to a hidden complex ionic system. Thus, we need to establish approximate models bypassing detailed ion channels. The books by Gerstner and Kistler (2002) and Gerstner et al. (2014) contain comprehensive introduction of various realistic and approximate spiking neuronal models. Here we briefly describe two models used throughout the thesis: the point process model and the leaky integrate-and-fire (LIF) model.

3.1.1 Point process model

A spike train, without considering its underlying biophysical characteristic, is simply a point process of spiking events. Point process models (Kass et al., 2014) for spike trains describe the discrete events, and the underlying firing mechanism is approximated into the conditional intensity function (CIF) which gives the probability of spiking within a short interval Δt. See Section 2.1 for a mathematical description of point processes and the CIF.

A widely applied model of the CIF is a Hawkes model incorporating autoregressive spiking history (Truccolo et al., 2005; Pillow et al., 2008). With the same notation as Section 2.1, this neural CIF after time discretization is given by

$$\lambda(t|H_t) = \exp\left\{\lambda_0(t) + a_0 + \sum_{j=1}^{h} a_j(N(t-(j-1)\Delta t) - N(t-j\Delta t))\right\}. \tag{3.1}$$

The base intensity term $\lambda_0(\cdot)$ describes the base firing rate, which is a function of time and/or external stimulus given by the visual attention models. The discretization interval Δt is often set to 1 ms, roughly the duration of a neuronal spike. The autoregressive order h can be set according to the type of neurons in the experiment, and AIC or BIC can be used to select h. For population of neurons, we may need to take into account the interaction between neurons, and the interaction enters the CIF in the autoregressive form of other neurons' spiking history. An important feature of such CIF models is that, when fitting spike train data, the model can be formulated as a generalized linear model (GLM) (Truccolo et al., 2005) and the model fitting procedure will be simple and efficient.

3.1.2 LIF model

The LIF model attempts to achieve a more detailed biophysical realism but still remains tractable. There is a large collection of different types of LIF models: deterministic or stochastic, linear or nonlinear, one or more dimensional, etc (see Burkitt (2006); Sacerdote and Giraudo (2013) for reviews). Here we present a stochastic LIF model incorporating spiking history effects.

In the LIF model, the membrane voltage is described by a diffusion process (see Section 2.2), which is the solution to the following OU-type SDE

$$\mathrm{d}X_t = (-\gamma(X_t - \mu) + I_s(t) + I_h(t))\mathrm{d}t + \sigma \mathrm{d}B_t. \tag{3.2}$$

The diffusion $\{X(t)\}$ models the stochastic evolution of the (dimensionless) voltage. The term $I_s(t)$ is the response to stimuli, and in this model the response equals the stimulus numerically, $I_s(t) = S(t)$, and is given by the visual attention models. The history effect is given by $I_h(t) = \sum_{\tau \in H_t} k_h(t - \tau)$, where H_t represents the history spike times up to time t and $k_h(\Delta) = \eta_1 e^{-\eta_2 \Delta} - \eta_3 e^{-\eta_4 \Delta}$ is a spike response kernel given by the difference of two decaying exponentials. So the history effect $I_h(t)$ is a convolution of discrete history spikes using the kernel $k_h(\Delta)$. The rest terms γ, μ and σ are parameters.

In this model, the voltage starts from a reset value, $X(0) = x_0$, and evolves stochastically driven by effects from stimuli, spiking history and Gaussian white noise. Once $X(t)$ passes a threshold value, x_{th}, a spike is generated at time t and the voltage is immediately reset to x_0. Thus, the spike train is a sequence of first passage times of the underlying diffusion process with respect to a certain threshold. By changing parameters of the response kernel, we can create different types of patterns in the spike train (Li et al., 2016), and we are then able to model various types of real spiking phenomena.

3.2 Visual attention

We aim at explaining visual attention for multiple stimuli from a neural perspective. Here we present the mathematical approaches for the neural explanations of the visual attention theories. First we introduce the development of the psychological Theory of Visual Attention (TVA) and the Neural Theory of Visual Attention (NTVA). Then we discuss the statistical methods for probability mixing versus response averaging and parallel versus serial processing inspired by the neuronal interpretation of NTVA.

3.2.1 TVA and NTVA

The book by Bundesen and Habekost (2008) presents a comprehensive introduction to the development and application of TVA and NTVA. Here we briefly summarize the key points.

TVA

TVA (Bundesen, 1990) is an effort to develop a unified computational mechanism of visual attention. Elegant mathematical equations are employed to account for visual cognition and attentional selection. TVA consists of a combined mechanism of filtering and pigeonholing, described respectively by a weight equation and a rate equation. Filtering effectively selects relevant objects having a particular feature by increasing their weights through the weight equation. Pigeonholing classifies the selected objects into a particular category by increasing the processing speed of that category through the rate equation. Combining filtering and pigeonholing, TVA successfully explains a wide range of empirical findings, including focused attention for selecting targets over distractors and divided attention for processing multiple simultaneous targets; see Bundesen (1990); Bundesen and Habekost (2008) and literature

therein.

TVA is a psychological model focusing on behavioral analysis without touching the interpretation in a neuronal level. In this thesis work, however, we need to investigate how attentional selection is conducted from a neuronal perspective.

Single neurons in visual attention

Empirical studies in neurophysiology have revealed the effects of visual attention on single neurons, summarized in the following four types (Bundesen and Habekost, 2008). First, when presented by multiple competing stimuli, neurons show strong variability in firing rate, affected by the attended object (Moran and Desimone, 1985; Chelazzi et al., 1998, etc). Second, when presented by a single stimulus, neuronal firing rate scales depending on the attention to the object (Treue and Trujillo, 1999, etc). Third, neurons show baseline shifts in firing rate when nothing is presented yet but a stimulus is expected to appear (Chelazzi et al., 1998, etc). Last, neurons sharing the same attended stimulus show increased synchronization in their activities (Fries et al., 2001, etc). Many cognitive models have been proposed to interpret these empirical findings in a neuron level, among which are the gain control models (e.g. Hillyard et al. (1998); Reynolds (2005)) and bias competition models (e.g. Desimone and Duncan (1995)), but none has managed to account for all the four aspects (Bundesen and Habekost, 2008). NTVA, on the other hand, is a successful attempt to cover all the empirical findings.

NTVA

NTVA (Bundesen et al., 2005) is an interpretation of TVA in a single neuron level. A central assumption in NTVA is that a neuron can only represent a single object at any given time. Both filtering and pigeonholing in TVA find their corresponding neuronal explanation. In filtering, the attentional weight of a stimulus from TVA gives the probability that a neuron represents the stimulus. For multiple stimuli, a neuron can attend to any stimulus, but has a tendency (higher probability) of attending to the target rather than the distractors. In pigeonholing, the scaling of processing speed for a particular categorization from TVA corresponds to the scaling of the firing rate of neurons responsible for making that categorization. Thus, in NTVA filtering increases the number of neurons representing a particular object and pigeonholing increases the firing rate of neurons performing a particular categorization.

NTVA explains all the four types of empirical findings mentioned previously and fits a broad range of experimental data; see Bundesen et al. (2005) and literature therein. First, for multiple stimuli filtering predicts that neurons can attend to the target as well as distractors with probabilities given by the weights. We expect strong variability of firing rate from multiple simultaneous neurons attending to different stimuli and from the same neuron during attention reallocation. Second, for single stimulus pigeonholing predicts a scaling of the firing rate of a neuron when the attention is directed to a feature signaled by the neuron. Third, the baseline activity shift can be explained by the Visual Short Term Memory (VSTM) mapping

in NTVA. Finally, NTVA predicts the increase of firing synchronization when neurons are receiving input from the common lower-level cells.

As a summary, TVA and NTVA provide a general unified computational mechanism to think and interpret visual attention, which explains the empirical neurophysiological findings extremely well. In this thesis work, we develop our statistical methods based on NTVA, and apply the methods to single-neuron data.

3.2.2 Probability mixing and response averaging

Probability mixing and response averaging are two opposing hypotheses for attention of multiple stimuli. The probability-mixing model is closely related to NTVA by applying the attentional weight and the bias parameter, which can be explained by filtering and pigeonholing; see Paper I for the discussion of their relation. The response-averaging model arises from the research done by Reynolds et al. (1999) which was later formulated as the normalization model in Reynolds and Heeger (2009). Reynolds et al. (1999) showed that the empirical firing rates of neurons presented by a stimulus pair, averaged across trials, was equal to a weighted average of the firing rates responding to each single stimulus. The selection between probability mixing and response averaging is an attempt not only to answer the fundamental question regarding neuronal response to multiple stimuli but also to verify the hypothesis of NTVA that a neuron only represents a single object at a time.

As in the Introduction section, we denote K isolated stimuli by $S = \{S_k; k = 1, 2, \ldots, K\}$, and the response of a neuron to S or any S_k by $I(S)$ and $I(S_k)$. The stimulus S can be temporal such that S is a function of time. The meaning of "response" varies in different context. It may refer to the firing rate in most empirical analysis and simple neuron models. While in the more biophysical models, the response refers to the effect of presynaptical spikes caused by the stimulus, and thus it enters the model as a contribution to the membrane voltage change (Li et al., 2016). In the probability-mixing model, we have

$$I(S) = I(S_k) \text{ with probability } \alpha_k, \tag{3.3}$$

and in the response-averaging model, we have

$$I(S) = \frac{1}{K} \sum_{k=1}^{K} I(S_k)\beta_k. \tag{3.4}$$

In both models, the response to single stimulus is given by

$$I(S_k) = v(S_k)a(S_k) + I_k. \tag{3.5}$$

Here, $v(S_k)$ gives the base response to the stimulus S_k as a tuning function of the strength of S_k, which is often modeled by a Gaussian or a Von Mises (circular Gaussian) function in empirical analysis (Shokhirev et al., 2006). The term $a(S_k)$ gives the attentional bias from particular experimental settings. For example, if there is a cued stimulus and S_k is the same as the cue, then the neuronal attention will be directed to S_k and the bias $a(S_k)$ can become bigger than 1. Finally, I_k represents a constant offset response irrelevant to the strength of

S_k and attentional bias. The formulation here corresponds to the rate equation in NTVA (Bundesen and Habekost, 2008; Bundesen et al., 2005). To separate the probability-mixing and respone-averaging model, we have to conduct single trial analysis rather than averaging across trials. Indeed, the average firing rate across trials will be identical for the two models with the same parameter values, and the actual difference resides in the variance across trials.

Dip test

Following the probability-mixing model, the repeated trial spike trains for the same neuron under the same condition may attend different stimuli, and the firing rates from all repetitions form a multimodal distribution. While following the response-averaging model, all repetitions share the same response and the firing rates will follow a unimodal distribution. The Dip test (Hartigan and Hartigan, 1985) is a statistical test for unimodality of an empirical distribution, where the null hypothesis is that the distribution is unimodal. Dip tests can be used for a straightforward empirical test on the firing rates of repetitions. A significant p-value means multimodality and the probability-mixing is favored. In practical application, the firing rate data may be too scarce for a reliable test, and we can instead apply the unimodal test on the inter-spike intervals (ISIs).

Attentional switching

If a neuron is presented to multiple stimuli for a long duration, in the probability-mixing model the attentional target may switch, even if the probability parameters stay the same. For example, Fiebelkorn et al. (2013) found that the sustained attention can fluctuate as frequently as 4 to 8 Hz. This random switching may be modeled by a Markov chain with appropriate discretized time step. The state space is the index of stimuli. With constant or temporal transition probability matrix of the Markov chain, we can describe various types of switching mechanisms, e.g. memoryless or autoregressive switching. Taking a sufficiently small time step, we can also well approximate continuous-time switching.

3.2.3 Parallel and serial processing

Parallel and serial processing explain, from different hypotheses, how multiple stimulus objects are processed after stimulus presentation. Countless behavioral and decision-making tasks have been analyzed for the comparison between parallel and serial processing (Bundesen and Habekost, 2008; Nobre and Kastner, 2013), but the neural explanation behind the two processing mechanisms is rarely touched. We first very briefly summarize the previous behavioral studies on parallel versus serial processing, and then present a short introduction of the neural methods used in the thesis work.

The behavioral studies mainly focused on mean response times, response time distributions and their relation to the display size. In the experiments by Sternberg (1966, 1969a,b); Schneider and Shiffrin (1977), the observer needed to identify the target from distractors and

respond as quickly as possible with a positive (target present) or a negative (target absent) answer, and the response times were recorded. Treisman et al. (1977) conducted similar research but applied more detailed discrimination between the target and the distractors, introducing single-feature search with one physical feature to distinguish the target and conjunction search combining two physical features. The mean response times were found to follow linear functions of the display size, which can be easily explained by a simple serial processing mechanism. However, as Townsend and Ashby (1983) analyzed, the serial model can be mimicked by a parallel model with limited processing capacity, in which not all objects can be processes at the same time and response times can be described by linear functions of the display size. Later Bricolo et al. (2002) employed highly inefficient tasks, and fitted response times to theoretical distributions using Gaussians and Exponentials. They found evidence for serial processing in different experimental schemes and the simple mimicry using parallel processing with limited capacity cannot explain all their experimental data. A new multi-feature whole-report paradigm was introduced by Bundesen et al. (2003) and further applied in Kyllingsbæk and Bundesen (2007), where the observer needed to process two separate features from each of two objects. Results showed evidence of processing only one feature from each of the two objects before interruption. This partial processing from both objects strongly favored parallel processing.

As discussed above, with the classical interpretation for parallel and serial processing, behavioral studies have produced results supporting both processing mechanisms. Here we apply more direct neural explanation, combining visual attention theories and neural spike train analysis. We describe parallel and serial processing with a unified neural framework, modeling single neurons with attentional switching. The difference of the two processing mechanisms from a neural perspective resides in whether neurons tend to attend to the same stimulus at the same time or they tend to split the attention independently. Our neural framework is based on probability mixing and NTVA, where a neuron only attends to a single stimulus at any given time with probabilities. The core idea to distinguish between parallel and serial processing is presented below.

Consider a case where there are in total N simultaneous neurons responding to 2 stimuli. The N neurons are homogeneous so that they are not distinguishable. Denote the attentional target of any neuron by X, and $X \in \{1, 0\}$ is a binary variable representing stimulus 1 or 2. We favor serial processing if we observe any of the following two cases.

1. Strong correlation: the correlation between the attentions of any two neurons X_i and X_j, $Cor(X_i, X_j)$, approaches 1;
2. Extreme probability: the probability of any neuron attending to stimulus 1 is extreme, i.e. $P(X = 1)$ approaches 0 or 1.

If neither happens, i.e. $Cor(X_i, X_j)$ is sufficiently different from 1 *and* $P(X = 1)$ is sufficiently different from both 0 and 1, we favor parallel processing. Therefore, parallel and serial processing can be distinguished by calculating $Cor(X_i, X_j)$ and $P(X = 1)$.

Another method is to use a single statistic. Denote the probability mass function (PMF) of the number of neurons attending to stimulus 1 by $P(\#\{i; X_i = 1\} = n) = f(n)$. If $f(n)$ near

0 or N is larger than elsewhere then we prefer serial processing, and if $f(n)$ near $N/2$ is larger then we prefer parallel processing. To describe such behavior of $f(n)$, we use a deviation statistic D_N given by

$$D_N = \frac{\sum_{n=0}^{N} |n - N/2| f(n)}{N/2}. \tag{3.6}$$

The statistic $D_N \in [0, 1]$ gives the expected deviation between the number of neurons attending to stimulus 1 and half of the total neuron number. It can also be explained as the deviation between the observed processing mechanism and the perfect parallel processing with uniform weights among stimuli. The bigger D_N is, the more we favor serial processing; and vice versa. To remove the dependence of D_N on the neuron number N, we can evaluate the asymptotic version

$$D^* = \lim_{N \to \infty} D_N, \tag{3.7}$$

which can be calculated explicitly for certain models.

To calculate the criteria of correlations, probabilities, and the deviation statistics, we model single-neuron data using visual attention theories with spiking neuron models, and infer parameters describing neuronal attention. The criteria can be calculated using the inferred parameters. Both processing mechanisms are described by the same model, but distinguished by parameters. See Paper II for the details of the models.

The methods introduced above applies only to cases of 2 stimuli. If there are more stimuli, apart from the neural model itself which becomes harder to fit, we also need to consider more cases and more dimensions for the correlation, the PMF, etc, to distinguish between parallel and serial processing. The methods quickly suffers from the curse of dimensionality as the number of stimuli goes bigger. We stick to the situation according to the studied experiment and do not pursue a generalization for higher dimensions in the current study.

3.3 Neural coding with visual attention

All the studies with neural modeling in this thesis, in a broad sense, belong to neural coding. We either conduct encoding by parameter estimation and model selection, or perform decoding by inferring stimulus and attention. The results of both encoding and decoding give us explanations on relevant questions in neuroscience or psychology. In the following we discuss general approaches for encoding and decoding using neural spike data under the visual attention hypotheses.

3.3.1 Encoding

Suppose we have stimuli denoted by X, spike train data Y, and model parameters θ. Denote by Z the latent attentional variables and possible hidden variables in the spiking neuron model if any. In the encoding model, X is first plugged into the visual attention model giving processed stimulus information, which is then taken by the spiking neuron model generating spikes. For example the visual attention hypotheses enter the CIF model (3.1) through the

base firing rate term $\lambda_0(\cdot)$ and the LIF model (3.2) through the stimulus response term $I_s(\cdot)$. Here X and Y are known from experiments, Z is latent and unknown, and θ is unknown and to be estimated.

For parameter estimation, we may obtain the ML estimator

$$\hat{\theta} = \arg\max_{\theta} \log P(X, Y|\theta), \tag{3.8}$$

where $P(X, Y|\theta)$ is the marginal likelihood

$$P(X, Y|\theta) = \int P(X, Y|Z, \theta) P(Z|\theta) \mathrm{d}Z. \tag{3.9}$$

In models where the attentional variables are discrete, the integral can be evaluated using sums. When Z includes continuous variables from e.g. complex neuron spiking models, the pseudo-marginal method can be employed by sampling Z; see parameter estimation of SSMs in 2.5.3. Besides straightforward MLE, we may also apply the EM algorithm by maximizing

$$Q(\theta|\theta') = \mathbb{E}_{Z|X,Y,\theta'}[\log P(X, Y, Z|\theta)]. \tag{3.10}$$

Again, if the conditional expectation is difficult to compute, we may employ sampling following the MCEM method. For example, Ditlevsen et al. (2014) provided inference for the two dimensional stochastic Morris-Lecar neuronal model under partial observations by maximizing the pseudo-marginal likelihood with EM using SMC particle filtering. The conditional likelihood of data given latent variables, $P(X, Y|Z, \theta)$, can be obtained by the likelihood of spike trains under the corresponding model, e.g. the likelihood formula for point processes (Truccolo et al., 2005; Kass et al., 2014) and the first-passage time solution for LIF models (Paninski et al., 2004, 2008, 2007; Iolov et al., 2014; Li et al., 2016). Apart from the above likelihood methods, parameter estimation of certain LIF encoding models can also be conducted by minimizing an error function based on the Fortet's equation (Ditlevsen and Lansky, 2007; Lansky and Ditlevsen, 2008; Ditlevsen and Ditlevsen, 2008), and using moment estimators derived by formulating martingales (Ditlevsen and Lansky, 2005, 2006).

Regarding model selection, ordinary methods can be applied, for example AIC and BIC. For model checking, we can apply uniform residual tests and the predictive error of spike trains Y. The later can be the RMSD of firing rates, the RMSD of ISIs, or some spike train metric (distance between spike trains; see e.g. Victor and Purpura (1997); van Rossum (2001)). The uniformity test of point process models can be performed on the time rescaling transformations (Brown et al., 2002; Haslinger et al., 2010) that give standard Poisson processes if the model is correct.

After neural encoding, apart from a ready-to-use neural model, we draw meaningful explanations or even central conclusions regarding both visual attention and neural spiking mechanism (see Papers I and II) from the parameter estimates.

3.3.2 Decoding

Consider the encoding model with the same notation as the above encoding section. In decoding, Y and θ are known, Z is unknown, and X can be known or unknown in different

situations. In addition, we may also have unknown parameters η that could include the hyperparameters of stimuli, non-constant parameters (e.g. trial-dependent or time-dependent), etc. In a typical situation, X is unknown and we need to obtain the posterior distribution of $\{X, Z, \eta\}$:

$$P(X, Z, \eta|Y, \theta). \tag{3.11}$$

Sometimes we are interested in only the stimuli X, or only the attention information Z, and sometimes we want to know both. This can be done by marginalization:

$$P(X, Z|Y, \theta) = \int P(X, Z, \eta|Y, \theta)\, \mathrm{d}\eta, \tag{3.12}$$

$$P(X|Y, \theta) = \iint P(X, Z, \eta|Y, \theta)\, \mathrm{d}\eta\, \mathrm{d}Z, \tag{3.13}$$

$$P(Z|Y, \theta) = \iint P(X, Z, \eta|Y, \theta)\, \mathrm{d}\eta\, \mathrm{d}X. \tag{3.14}$$

Another situation is that we also have knowledge about the input stimuli X and need to decode the neuronal attention, i.e., the distribution

$$P(Z|X, Y, \theta) = \int P(Z, \eta'|X, Y, \theta)\, \mathrm{d}\eta', \tag{3.15}$$

where η' does not contain hyperparameters from X since we have the full information about X.

If the stimuli are constant and the dimension of $\{X, Z, \eta\}$ is small, we can apply MAP estimation on e.g. distribution (3.11):

$$\{\hat{X}, \hat{Z}, \hat{\eta}\}_{MAP} = \arg\max_{X,Z,\eta} \log P(X, Z, \eta|Y, \theta), \tag{3.16}$$

or even on the marginals if the integrals can be easily evaluated. MCMC can also be used to get the full distribution.

However, in most real applications e.g. brain-machine interfaces (BMI) (Lebedev and Nicolelis, 2006; Waldert et al., 2009), X is temporal and needs to be decoded in real time. This leads to the SSM formulation of decoding. SSMs have become a central idea in decoding tasks since the pioneering work by Brown et al. (1998) applying point processes in decoding with spike trains; also see Paninski et al. (2010) for a review of SSMs in neuroscience and the literature therein.

Suppose we want to decode X and Z, and the states for the SSM are $\{X_t, Z_t, \eta_t; t \geq 0\}$. It amounts to obtaining the filtering or smoothing distribution given by

$$P(X_t, Z_t, \eta_t|Y_{0:t}, \theta) \tag{3.17}$$

and

$$P(X_t, Z_t, \eta_t|Y_{0:T}, \theta), \tag{3.18}$$

respectively. The particle methods introduced in Section 2.5, in particular methods for hybrid SSMs with parameter learning, can be employed to approximate the filtering or smoothing

distributions. The decoding of stimuli or attention is then obtained. In addition to the SMC methods, for simple spiking models e.g. point processes, the optimal decoding estimation can be obtained at each time step numerically through MAP estimator $\{\hat{X}_t, \hat{Z}_t, \hat{\eta}_t\}_{MAP}$, or even analytically via Gaussian approximation on the posterior distribution; for example see Brown et al. (1998); Eden et al. (2004).

Filtering distribution gives online decoding where estimates are calculated immediately after receiving the observation at t. The full smoothing gives offline decoding after receiving the full observation up to T. Finally, there is also "semi-online" decoding, where estimates of a past time $t - l$ are calculated based on the observation up to t through the partial smoothing distribution, $P(X_{t-l}, Z_{t-l}, \eta_{t-l}|Y_{0:t}, \theta)$. Semi-online decoding are used when we allow for some lag l before reporting estimates and desire better accuracy.

Decoding of visual attention provides posterior inference of what could have happened regarding neuronal attention given concrete observations, for example $P(Z_s|Y_{0:t}, \theta), s \leq t$. By contrast, the encoding model show neurons' general property and prior knowledge of what could happen in the future providing stimuli, for example $P(Z_t|X_t, \theta)$.

Chapter 4

Overview and Prospects

We first provide an overview for each project individually, and then propose possible future development.

4.1 Overview of studies

4.1.1 Paper I

Paper I presents the statistical model selection between the response-averaging and the probability-mixing model from experimental spike train data from the middle temporal (MT) visual area of rhesus monkeys. A mixture of two isolated random dot patterns are used as the stimuli and neural spike trains are recorded as the observation. For the spike train of each single trial, we employ the point process encoding model incorporating the processed stimulus information by the visual attention models. After parameter estimation, the selection between the two visual attention model is performed using AIC, BIC, uniformity tests, and predictability by RMSD of firing rates. Furthermore, we also conduct empirical selection using the dip unimodality test on ISIs. In addition to main model selection analysis, the possibility of attentional switching and neuronal population correlation are explored as well.

All model selection methods show the probability-mixing model is favored. The unimodality empirical analysis also suggests increased multimodality under stimulus mixtures. The possible attentional switching seems not to affect the model selection. The population correlation is rare, but could be due to the limited sample size.

4.1.2 Paper II

In Paper II we distinguish between serial and parallel processing in visual search using spike trains from simultaneously recorded neurons in monkey prefrontal cortex. The stimuli are two static images located on both sides of the vision. Again we employ the point process

encoding model for spike trains. Regarding the two visual search hypotheses, we apply a HMM with Binomial distributions, a unified model to describe both serial and parallel processing depending on the parameter setting. The model is based on a neural explanation assuming probability mixing. To distinguish between the two hypotheses, we use the neuronal correlation of attention, the probability of attention, and the deviation statistic D_n. From the parameter estimates in the encoding stage, we assess how the neuron performs visual search in general, and from the attention estimation in the decoding stage, we infer how the neuron could have allocated the attention given observations. In addition to the Binomial-HMM model, we also tried a correlated Binomial model for neuronal attention, following the same strategy to distinguish between serial and parallel processing.

Results of this study show evidence of both serial and parallel processing at all time steps. However, we find, the simultaneously recorded neurons tend to split attention between both stimuli in the early stage after stimulus onset, suggesting stronger parallel processing. Afterwards they tend to focus together on the same stimulus, indicating serial processing. Further, in the early stage neurons show a tendency of attending to the contralateral stimulus. The above finding is based on simultaneous neurons in one location, which already show parallel processing. From the perspective of the whole brain with two hemispheres, both showing tendency for the contra side, we may conclude strong evidence of parallel processing in the early stage of visual search.

4.1.3 Paper III

Paper III performs model selection between response averaging and probability mixing using a LIF encoding model incorporating long-term spike history effects on spike trains. The study is done in a more theoretical way with simulations. We conduct the parameter estimation using various numerical methods for diffusions, and perform the model selection with AIC, BIC and uniformity tests. The distinguishability of response averaging and probability mixing is systematically explored, using different response kernels, stimulus types, weight parameters, stimulus dissimilarities and data sizes.

The simulation study shows that parameter estimation of both response averaging and probability mixing can be successfully done for all settings. Intuitively, the distinguishability relies on sufficient data size, stimulus dissimilarity and weight parameters. However, the required sufficiency of settings, we find, is surprising low for a high accuracy of correct model selection. In addition to the main goal of model selection between the two visual attention hypotheses, as a side contribution we also establish a LIF model with tunable response kernels for various types of spiking patterns, and compare the efficiency and accuracy of parameter estimation of four numerical methods using PDEs and IEs.

4.1.4 Paper IV

In Paper IV we conduct decoding of multiple stimuli given spike trains under probability mixing using the LIF model. Two decoding scenarios are considered. The first is based on an

interval of fixed attention with unknown number of stimuli, for which we propose a decoding algorithm by clustering spike trains and decoding each categories, bypassing the deficiency of mixture models. The second is a more realistic situation with temporal stimulus mixture and switching attentions. We formulate SSMs for stimuli and attention, and employ various SMC particle methods. For population decoding, we consider both serial and parallel processing. The former assumes fully correlation among neurons and decodes only one stimulus component at a given time, while the latter assumes fully independence among neurons and can decode multiple components given sufficient data.

For the first scenario, our simulation study indicates the proposed algorithm performs better than the basic method using mixture models. For the second scenario, we systematically compare various particle methods in situations with different stimulus numbers and spike train numbers. Though all methods achieve good decoding results with small RMSD values, their performance difference varies depending on the situations.

4.1.5 Contribution summary

This thesis study brings a novel methodology development combining mathematical spiking neuron models for single spike trains with formulated neural explanations of visual attention theories. We perform both application to experimental data and theoretical study with simulations, and obtain interpretation and insights in both neuroscience/psychology and mathematics. The study also provides a framework of neural coding under visual attention, extending the already extensively studied topics of neural encoding and decoding by taking into account how complicated stimuli are processed by the brain through attention. We hope this idea of understanding visual attention by mathematical modeling of single neurons provides a worthy and promising new perspective, which could draw more attention from researchers and receive further inspiring ideas.

4.2 Prospects

4.2.1 Real data analysis

Based on the two studies on experimental data presented in Papers I and II, we could have the following extensions.

1. Currently we have only been using the point process model with the CIF defined in an autoregressive manner, which provides a GLM framework. We could approximate further for even more efficient model fitting, for example using the renewal process models. In addition, the history autoregressive order in the CIF can be decided by model selection with AIC/BIC, as in Truccolo et al. (2005).

2. We may consider the full spike train, instead of the short interval of around 500ms right now. The full spike train can be as long as 3000ms in Paper I and 2000ms in

Paper II. That will give us more robust estimation of the spiking activity, in particular, the spontaneous activity. However, we will need to consider the stimulus change and attentional switching during the whole recording period.

3. In Paper I the model checking procedure could also include a decoding analysis of the direction of stimuli, besides the RMSD of firing rates currently used.

4. The response-averaging and probability-mixing model can be further extended such that we consider the background as a third stimulus and consider the whole stimulus pair also as a single stimulus object. This gives a mixture of more than two stimuli. Likewise, the parallel and serial processing may also take into account the fixation point and other noisy effects.

5. As said in Paper II, we only consider the separation of serial and parallel processing based on two stimuli. If for example we consider other objects in the vision, such as the central dot, there will be more than two stimuli. A theoretical study may be conducted to generalize our original methods and consider any number of stimuli. Techniques need to be developed to avoid the curse of dimensionality.

4.2.2 Simulation study

In the simulation studies in Papers III and IV, we aim at more complex models and scenarios and evaluate the applicability in theory.

1. To improve the efficiency of numerical computation, the solution of PDEs can be rearranged using matrix multiplications and implemented in parallel on GPU devices.

2. We have found different performances among the four numerical methods using PDEs and IEs for parameter estimation. We may conduct further theoretical study on the efficiency and accuracy of the numerics, which could lead to for example optimal designs of grid sizes in different scenarios for the four methods. This will be a contribution to the general first-passage time problem.

3. We have not considered encoding in an attention-switching case with long spike trains, though the decoding is indeed performed with attention switching. With high dimensional latent attentional variables, the encoding will include parameter estimation of general SSMs, for which we can apply the methods introduced in the previous chapters. Besides attention switching, other extensions may be colored non-Gaussian noise, two or more dimensional LIF models, etc.

4. For particle filtering with parameter learning, besides the kernel smoothing method by Liu and West (2001), we may also employ the idea of the SMC^2 algorithm (Chopin et al., 2013), where we perform a particle filter in the parameter space and each particle itself also runs a particle filter in the latent state space.

4.2.3 Additional topics

A general serial/parallel processing model with attentional switching

The Binomial-HMM model used in Paper II for serial and parallel processing, and the neuronal switching model used in Paper IV for decoding serial and parallel populations, rely on different assumptions, but the two can be unified in a more general HMM model, shown in Figure 4.1.

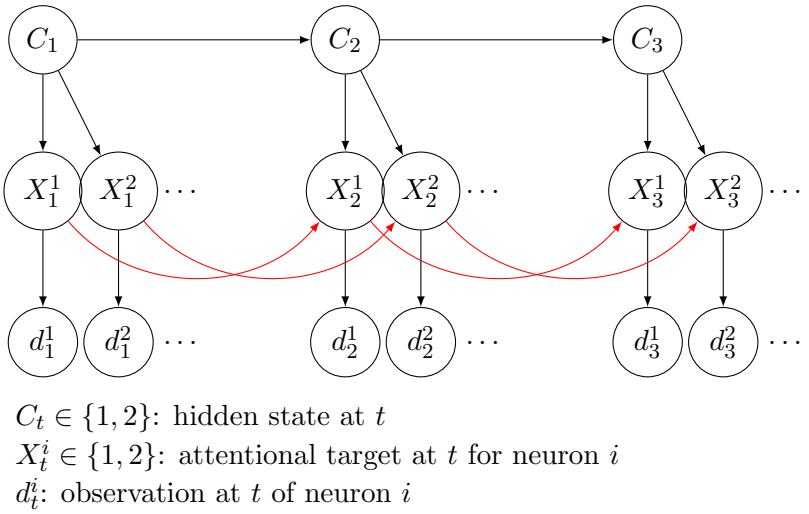


Figure 4.1: Diagram of a generalized HMM model for attention switching.

The difference between this generalized version and the Binomial-HMM is that there is direct dependence between X_t^i and X_{t-1}^i, i.e. the attentional target of the same neuron now follows a Markov chain along time steps given $\{C_{1:t}\}$. While in the Binomial-HMM $\{X_{1:t}\}$ are conditionally independent given hidden states $\{C_{1:t}\}$. We simplify the model as such because the Binomial-HMM is easier to fit with simpler structure and less parameters, especially considering limited data size and big noise in experimental spike trains. Though being simpler, the Binonial-HMM still models neuronal correlations and can describe both serial and parallel processing. In the attention switching model for parallel processing in Paper IV, there are dependencies among $\{X_{1:t}\}$, but we discard the hidden states C_t. So the attention switching of each neuron follows a Markov mechanism and all neurons are independent. Note that with on-line parameter learning of the transition probabilities we can still achieve the temporal effects brought by the states C_t.

Single neuron approach for decision making

Decision making is another widely studied topic in psychology. The most famous computational method for decision making is the drift-diffusion model (DDM) (Smith, 2000), where the decision procedure is described by a Wiener process with non-zero drift and diffusion coefficients. The Wiener process accumulates evidence, and a decision is made once it passes the threshold. A common paradigm in research is the two-alternative forced choice task (2AFC), where the

brain needs to choose one from two alternatives. For the DDM with 2AFC, the evidence can be either positive or negative representing tendency to either choice, and there are two threshold values for making the two choices. This is a first-passage time problem with two boundaries. Solving the Fokker-Planck equation for the diffusion with two absorbing boundaries gives us the first-passage time probability of reaching either boundary. Further, we may solve another differential equation and obtain the probability of the diffusion reaching one boundary before the other (Karlin and Taylor, 1981).

However, a convincing detailed neural explanation of decision making is lacking; current research in neural theory mainly focus on the lateral intraparietal cortex (LIP) which controls the eye movement. The firing rate of LIP neurons averaged across trials was found to follow the DDM (Gold and Shadlen, 2007). The average firing rate evolves stochastically as a Wiener process and the decision is made when the firing rate reaches a threshold. On the other hand, Latimer et al. (2015) proposed that the firing rate of a LIP neuron could jump to a threshold value instantaneously, instead of slowly accumulating. The average of jumping steps across trials also gives the stochastic accumulating behavior. The difference of the two models resides in single spike trains, somewhat similar to our study on response averaging and probability mixing. Their proposal was verified and supported by statistical inference based on single spike trains using spiking neuron models, and model selection using DIC. Though the new approach is difficult to be immediately digested and the validity is questioned by researchers using spike train averaging (Shadlen et al., 2016; Latimer et al., 2016), the emerging novel methods for decision making modeling single spike trains have brought illuminating ideas.

Incorporating probability mixing and NTVA, a new study for decision making may be based on single spike train modeling of visual cortex neurons. We assume each neuron attend to only one stimulus at any given time, and the resources allocated to a stimulus may be represented by the number of neurons attending to it, possibly also weighted by the firing rate of each neuron. Combined with the neural explanation of serial and parallel processing, we may infer/decode the resource allocation of each stimulus for a given time interval. The temporal resource allocation can be related to the evidence term in either DDM or the stepping model for the decision making procedure. This is a preliminary idea, though, and the validity still needs to be checked in details.

Bibliography

Akaike, H. (1974). A new look at the statistical model identification. *IEEE transactions on automatic control*, 19(6):716–723.

Andreetta, C., Begot, V., Berthold, J., Elsman, M., Henriksen, T., Nordfang, M.-B., and Oancea, C. (2015). A financial benchmark for gpgpu compilation. Technical report.

Bricolo, E., Gianesini, T., Fanini, A., Bundesen, C., and Chelazzi, L. (2002). Serial attention mechanisms in visual search: A direct behavioral demonstration. *Journal of cognitive neuroscience*, 14(7):980–993.

Briers, M., Doucet, A., and Maskell, S. (2010). Smoothing algorithms for state–space models. *Annals of the Institute of Statistical Mathematics*, 62(1):61–89.

Brown, E., Barbieri, R., Ventura, V., Kass, R., and Frank, L. (2002). The time-rescaling theorem and its application to neural spike train data analysis. *Neural Computation*, 14(2):325–346.

Brown, E. N., Frank, L. M., Tang, D., Quirk, M. C., and Wilson, M. A. (1998). A statistical paradigm for neural spike train decoding applied to position prediction from ensemble firing patterns of rat hippocampal place cells. *The Journal of Neuroscience*, 18(18):7411–7425.

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

Bundesen, C. (1990). A theory of visual attention. *Psychological review*, 97(4):523.

Bundesen, C. and Habekost, T. (2008). *Principles of visual attention: Linking mind and brain*. Oxford University Press.

Bundesen, C., Habekost, T., and Kyllingsbæk, S. (2005). A neural theory of visual attention: bridging cognition and neurophysiology. *Psychological Review*, 112(2):291.

Bundesen, C., Kyllingsbæk, S., and Larsen, A. (2003). Independent encoding of colors and shapes from two stimuli. *Psychonomic Bulletin & Review*, 10(2):474–479.

Burkitt, A. N. (2006). A review of the integrate-and-fire neuron model: I. homogeneous synaptic input. *Biological cybernetics*, 95(1):1–19.

Burnham, K. P. and Anderson, D. R. (2002). *Model selection and multimodel inference: A practical information-theoretic approach*. Springer.

Cappé, O., Moulines, E., and Rydén, T. (2009). Inference in hidden markov models. In *Proceedings of EUSFLAT Conference*, pages 14–16.

Carvalho, C., Johannes, M. S., Lopes, H. F., and Polson, N. (2010). Particle learning and smoothing. *Statistical Science*, 25(1):88–106.

Chelazzi, L., Duncan, J., Miller, E. K., and Desimone, R. (1998). Responses of neurons in inferior temporal cortex during memory-guided visual search. *Journal of neurophysiology*, 80(6):2918–2940.

Chhikara, R. (1988). *The Inverse Gaussian Distribution: Theory: Methodology, and Applications*, volume 95. CRC Press.

Chopin, N., Jacob, P. E., and Papaspiliopoulos, O. (2013). Smc2: an efficient algorithm for sequential analysis of state space models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 75(3):397–426.

Cox, D. R. (1955). Some statistical methods connected with series of events. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 129–164.

Cox, J. C., Ingersoll Jr, J. E., and Ross, S. A. (1985). A theory of the term structure of interest rates. *Econometrica: Journal of the Econometric Society*, pages 385–407.

Daley, D. and Vere-Jones, D. (2003). An introduction to the theory of point processes, volume i: Elementary theory and methods of probability and its applications.

Del Moral, P., Doucet, A., and Singh, S. (2010). Forward smoothing using sequential monte carlo. *arXiv preprint arXiv:1012.5390*.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38.

Desimone, R. and Duncan, J. (1995). Neural mechanisms of selective visual attention. *Annual review of neuroscience*, 18(1):193–222.

Ditlevsen, S. and Ditlevsen, O. (2008). Parameter estimation from observations of first-passage times of the Ornstein–Uhlenbeck process and the Feller process. *Probabilistic Engineering Mechanics*, 23(2):170–179.

Ditlevsen, S. and Lansky, P. (2005). Estimation of the input parameters in the Ornstein-Uhlenbeck neuronal model. *Physical Review E*, 71:011907.

Ditlevsen, S. and Lansky, P. (2006). Estimation of the input parameters in the Feller neuronal model. *Physical Review E*, 73:061910.

Ditlevsen, S. and Lansky, P. (2007). Parameters of stochastic diffusion processes estimated from observations of first-hitting times: Application to the leaky integrate-and-fire neuronal model. *Physical Review E*, 76(4):041906.

Ditlevsen, S., Samson, A., et al. (2014). Estimation in the partially observed stochastic morris–lecar neuronal model with particle filter and stochastic approximation methods. *The annals of applied statistics*, 8(2):674–702.

Doucet, A., Godsill, S., and Andrieu, C. (2000). On sequential monte carlo sampling methods for bayesian filtering. *Statistics and computing*, 10(3):197–208.

Eden, U. T., Frank, L. M., Barbieri, R., Solo, V., and Brown, E. N. (2004). Dynamic analysis of neural encoding by point process adaptive filtering. *Neural computation*, 16(5):971–998.

Feller, W. et al. (1951). *Diffusion processes in genetics*. University of California Press Berkeley, Calif.

Fiebelkorn, I. C., Saalmann, Y. B., and Kastner, S. (2013). Rhythmic sampling within and between objects despite sustained attention at a cued location. *Current Biology*, 23(24):2553–2558.

Fries, P., Reynolds, J. H., Rorie, A. E., and Desimone, R. (2001). Modulation of oscillatory neuronal synchronization by selective visual attention. *Science*, 291(5508):1560–1563.

Gattass, R., Nascimento-Silva, S., Soares, J. G. M., Lima, B., Jansen, A. K., Diogo, A. C. M., Farias, M. F., Marcondes, M., Botelho, E. P., Mariani, O. S., Azzi, J., and Fiorani, M. (2005). Cortical visual areas in monkeys: Location, topography, connections, columns, plasticity and cortical dynamics. *Philosophical Transactions of the Royal Society of London: B*, 360:709–731.

Gerstner, W. and Kistler, W. M. (2002). *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge university press.

Gerstner, W., Kistler, W. M., Naud, R., and Paninski, L. (2014). *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge University Press.

Gold, J. I. and Shadlen, M. N. (2007). The neural basis of decision making. *Annu. Rev. Neurosci.*, 30:535–574.

Gordon, N. J., Salmond, D. J., and Smith, A. F. (1993). Novel approach to nonlinear/non-gaussian bayesian state estimation. In *IEE Proceedings F-Radar and Signal Processing*, volume 140, pages 107–113. IET.

Hamilton, J. D. (1989). A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica: Journal of the Econometric Society*, pages 357–384.

Hamilton, J. D. and Raj, B. (2013). *Advances in Markov-Switching Models: Applications in Business Cycle Research and Finance*. Springer Science & Business Media.

Hartigan, J. A. and Hartigan, P. (1985). The dip test of unimodality. *The Annals of Statistics*, pages 70–84.

Haslinger, R., Pipa, G., and Brown, E. (2010). Discrete time rescaling theorem: determining goodness of fit for discrete time statistical models of neural spiking. *Neural Computation*, 22(10):2477–2506.

Hastings, W. K. (1970). Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109.

Hawkes, A. G. (1971). Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1):83–90.

Hillyard, S. A., Vogel, E. K., and Luck, S. J. (1998). Sensory gain control (amplification) as a mechanism of selective attention: electrophysiological and neuroimaging evidence. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 353(1373):1257–1270.

Hodgkin, A. L. and Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, 117(4):500.

Horst, R., Pardalos, P. M., and Van Thoai, N. (2000). *Introduction to global optimization*. Springer Science & Business Media.

Hurn, A., Jeisman, J., and Lindsay, K. (2005). ML estimation of the parameters of SDEs by numerical solution of the fokker-planck equation. In *MODSIM 2005: International Congress on Modelling and Simulation: Advances and Applications for Management and Decision Making*, pages 849–855.

Hürzeler, M. and Künsch, H. R. (2001). Approximating and maximising the likelihood for a general state-space model. In *Sequential Monte Carlo methods in practice*, pages 159–175. Springer.

Iolov, A., Ditlevsen, S., and Longtin, A. (2014). Fokker–Planck and Fortet equation-based parameter estimation for a Leaky Integrate-and-Fire model with sinusoidal and stochastic forcing. *The Journal of Mathematical Neuroscience*, 4(1):4.

Kantas, N., Doucet, A., Singh, S. S., Maciejowski, J., Chopin, N., et al. (2015). On particle methods for parameter estimation in state-space models. *Statistical science*, 30(3):328–351.

Karlin, S. and Taylor, H. E. (1981). *A second course in stochastic processes*. Elsevier.

Kass, R. E., Eden, U. T., and Brown, E. N. (2014). *Analysis of neural data*. Springer.

Kim, C.-J., Nelson, C. R., et al. (1999). *State-space models with regime switching: classical and Gibbs-sampling approaches with applications*, volume 2. MIT press Cambridge, MA.

Krishnamurthy, V. and Ryden, T. (1998). Consistent estimation of linear and non-linear autoregressive models with markov regime. *Journal of time series analysis*, 19(3):291–307.

Kyllingsbæk, S. and Bundesen, C. (2007). Parallel processing in a multifeature whole-report paradigm. *Journal of Experimental Psychology: Human Perception and Performance*, 33(1):64.

Lansky, P. and Ditlevsen, S. (2008). A review of the methods for signal estimation in stochastic diffusion leaky integrate-and-fire neuronal models. *Biological Cybernetics*, 99:253–262.

Latimer, K. W., Yates, J. L., Meister, M. L., Huk, A. C., and Pillow, J. W. (2015). Single-trial spike trains in parietal cortex reveal discrete steps during decision-making. *Science*, 349(6244):184–187.

Latimer, K. W., Yates, J. L., Meister, M. L. R., Huk, A. C., and Pillow, J. W. (2016). Response to comment on "single-trial spike trains in parietal cortex reveal discrete steps during decision-making". *Science*, 351(6280):1406–1406.

Lebedev, M. A. and Nicolelis, M. A. (2006). Brain–machine interfaces: past, present and future. *TRENDS in Neurosciences*, 29(9):536–546.

Li, K., Bundesen, C., and Ditlevsen, S. (2016). Responses of leaky integrate-and-fire neurons to a plurality of stimuli in their receptive fields. *The Journal of Mathematical Neuroscience*, 6(1):1–33.

Lin, L., Liu, K., and Sloan, J. (2000). A noisy monte carlo algorithm. *Physical Review D*, 61(7):074505.

Liu, J. and West, M. (2001). Combined parameter and state estimation in simulation-based filtering. In *Sequential Monte Carlo methods in practice*, pages 197–223. Springer.

Malik, S. and Pitt, M. K. (2011). Particle filters for continuous likelihood evaluation and maximisation. *Journal of Econometrics*, 165(2):190–209.

Moran, J. and Desimone, R. (1985). Selective attention gates visual processing in the extrastriate cortex. *Science*, 229(4715):782–784.

Nobre, K. and Kastner, S. (2013). *The Oxford handbook of attention*. Oxford University Press.

Nocedal, J. and Wright, S. (2006). *Numerical optimization*. Springer Science & Business Media.

Oksendal, B. (2013). *Stochastic differential equations: an introduction with applications*. Springer Science & Business Media.

Orhan, A. E. and Ma, W. J. (2015). Neural population coding of multiple stimuli. *Journal of Neuroscience*, 35:3825–3841.

Paninski, L., Ahmadian, Y., Ferreira, D. G., Koyama, S., Rad, K. R., Vidne, M., Vogelstein, J., and Wu, W. (2010). A new look at state-space models for neural data. *Journal of computational neuroscience*, 29(1-2):107–126.

Paninski, L., Haith, A., and Szirtes, G. (2008). Integral equation methods for computing likelihoods and their derivatives in the stochastic integrate-and-fire model. *Journal of computational neuroscience*, 24(1):69–79.

Paninski, L., Pillow, J., and Lewi, J. (2007). Statistical models for neural encoding, decoding, and optimal stimulus design. *Progress in brain research*, 165:493–507.

Paninski, L., Pillow, J. W., and Simoncelli, E. P. (2004). Maximum likelihood estimation of a stochastic integrate-and-fire neural encoding model. *Neural computation*, 16(12):2533–2561.

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

Pitt, M. K. and Shephard, N. (1999). Filtering via simulation: Auxiliary particle filters. *Journal of the American statistical association*, 94(446):590–599.

Press, W. H. (2007). *Numerical recipes: The art of scientific computing*. Cambridge University Press, 3 edition.

Redner, S. (2001). *A guide to first-passage processes*. Cambridge University Press, Cambridge.

Reynolds, J. H. (2005). Visual cortical circuits and spatial attention. *Neurobiology of attention*, pages 42–49.

Reynolds, J. H., Chelazzi, L., and Desimone, R. (1999). Competitive mechanisms subserve attention in macaque areas v2 and v4. *The Journal of Neuroscience*, 19(5):1736–1753.

Reynolds, J. H. and Heeger, D. J. (2009). The normalization model of attention. *Neuron*, 61(2):168–185.

Ricciardi, L. M. and Sato, S. (1990). Diffusion processes and first-passage-time problems. *Lectures in applied mathematics and informatics*, pages 206–285.

Rieke, F. (1999). *Spikes: exploring the neural code*. MIT press.

Rios, M. P. and Lopes, H. F. (2013). The extended liu and west filter: Parameter learning in markov switching stochastic volatility models. In *State-Space Models*, pages 23–61. Springer.

Rissanen, J. (1998). *Stochastic complexity in statistical inquiry*, volume 15. World scientific.

Sacerdote, L. and Giraudo, M. T. (2013). Stochastic integrate and fire models: a review on mathematical methods and their applications. In *Stochastic biomathematical models*, pages 99–148. Springer.

Schneider, W. and Shiffrin, R. M. (1977). Controlled and automatic human information processing: I. detection, search, and attention. *Psychological review*, 84(1):1.

Schwarz, G. et al. (1978). Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464.

Shadlen, M. N., Kiani, R., Newsome, W. T., Gold, J. I., Wolpert, D. M., Zylberberg, A., Ditterich, J., de Lafuente, V., Yang, T., and Roitman, J. (2016). Comment on "single-trial spike trains in parietal cortex reveal discrete steps during decision-making". *Science*, 351(6280):1406–1406.

Shokhirev, K., Kumar, T., and Glaser, D. (2006). The influence of cortical feature maps on the encoding of the orientation of a short line. *Journal of Computational Neuroscience*, 20(3):285–297.

Smith, A. T., Singh, K. D., Williams, A. L., and Greenlee, M. W. (2001). Estimating receptive field size from fMRI data in human striate and extrastriate visual cortex. *Cerebral Cortex*, 11:1182–1190.

Smith, P. L. (2000). Stochastic dynamic models of response time and accuracy: A foundational primer. *Journal of mathematical psychology*, 44(3):408–463.

Sternberg, S. (1966). High-speed scanning in human memory. *Science*, 153(3736):652–654.

Sternberg, S. (1969a). The discovery of processing stages: Extensions of donders' method. *Acta psychologica*, 30:276–315.

Sternberg, S. (1969b). Memory-scanning: Mental processes revealed by reaction-time experiments. *American scientist*, 57(4):421–457.

Townsend, J. T. and Ashby, F. G. (1983). *Stochastic modeling of elementary psychological processes*. CUP Archive.

Treisman, A., Sykes, M., and Gelade, G. (1977). Selective attention and stimulus integration. *Attention and performance VI*, pages 333–361.

Treue, S. and Trujillo, J. C. M. (1999). Feature-based attention influences motion processing gain in macaque visual cortex. *Nature*, 399(6736):575–579.

Truccolo, W., Eden, U. T., Fellows, M. R., Donoghue, J. P., and Brown, E. N. (2005). 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.

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

Victor, J. D. and Purpura, K. P. (1997). Metric-space analysis of spike trains: theory, algorithms and application. *Network: computation in neural systems*, 8(2):127–164.

Waldert, S., Pistohl, T., Braun, C., Ball, T., Aertsen, A., and Mehring, C. (2009). A review on directional information in neural signals for brain-machine interfaces. *Journal of Physiology-Paris*, 103(3):244–254.

Wei, G. C. and Tanner, M. A. (1990). A monte carlo implementation of the em algorithm and the poor man's data augmentation algorithms. *Journal of the American statistical Association*, 85(411):699–704.

Papers and Manuscripts

I Neurons in Primate Visual Cortex Alternate Between Responses to Multiple Stimuli in Their Receptive Field

Published in Frontiers in Computational Neuroscience, 10:141 (2016)
DOI: 10.3389/fncom.2016.00141

Kang Li
Department of Mathematical Sciences, Department of Psychology
University of Copenhagen

Vladislav Kozyrev
German Primate Center, Cognitive Neuroscience Laboratory
Bernstein Center for Computational Neuroscience
Institute for Neuroinformatics, Ruhr University Bochum

Søren Kyllingsbæk
Department of Psychology
University of Copenhagen

Stefan Treue
German Primate Center, Cognitive Neuroscience Laboratory
Bernstein Center for Computational Neuroscience
Faculty for Biology and Psychology, Goettingen University

Susanne Ditlevsen
Department of Mathematical Sciences
University of Copenhagen

Claus Bundesen
Department of Psychology
University of Copenhagen

frontiers in Computational Neuroscience

ORIGINAL RESEARCH
published: 27 December 2016
doi: 10.3389/fncom.2016.00141

Neurons in Primate Visual Cortex Alternate between Responses to Multiple Stimuli in Their Receptive Field

Kang Li[1,2†], *Vladislav Kozyrev*[3,4,5,6†], *Søren Kyllingsbæk*[2], *Stefan Treue*[3,4,7‡], *Susanne Ditlevsen*[1*‡] *and Claus Bundesen*[2‡]

[1] Department of Mathematical Sciences, University of Copenhagen, Copenhagen, Denmark, [2] Department of Psychology, University of Copenhagen, Copenhagen, Denmark, [3] Cognitive Neuroscience Laboratory, German Primate Center, Goettingen, Germany, [4] Bernstein Center for Computational Neuroscience, Goettingen, Germany, [5] Chair Theory of Cognitive Systems, Institute for Neuroinformatics, Ruhr University Bochum, Bochum, Germany, [6] Visual Cognition Lab, Department of Medicine/Physiology, University of Fribourg, Fribourg, Switzerland, [7] Faculty for Biology and Psychology, Goettingen University, Geottingen, Germany

A fundamental question concerning representation of the visual world in our brain is how a cortical cell responds when presented with more than a single stimulus. We find supportive evidence that most cells presented with a pair of stimuli respond predominantly to one stimulus at a time, rather than a weighted average response. Traditionally, the firing rate is assumed to be a weighted average of the firing rates to the individual stimuli (response-averaging model) (Bundesen et al., 2005). Here, we also evaluate a probability-mixing model (Bundesen et al., 2005), where neurons temporally multiplex the responses to the individual stimuli. This provides a mechanism by which the representational identity of multiple stimuli in complex visual scenes can be maintained despite the large receptive fields in higher extrastriate visual cortex in primates. We compare the two models through analysis of data from single cells in the middle temporal visual area (MT) of rhesus monkeys when presented with two separate stimuli inside their receptive field with attention directed to one of the two stimuli or outside the receptive field. The spike trains were modeled by stochastic point processes, including memory effects of past spikes and attentional effects, and statistical model selection between the two models was performed by information theoretic measures as well as the predictive accuracy of the models. As an auxiliary measure, we also tested for uni- or multimodality in interspike interval distributions, and performed a correlation analysis of simultaneously recorded pairs of neurons, to evaluate population behavior.

Keywords: probability-mixing, response-averaging, primate visual cortex, multiple stimuli, point process, model selection

Edited by:
Mayank R. Mehta, University of California, Los Angeles, USA

Reviewed by:
Ko Sakai, University of Tsukuba, Japan
Yuwei Cui, Numenta, Inc, USA

***Correspondence:**
Susanne Ditlevsen
susanne@math.ku.dk

[†] *Co-first author.*
[‡] *Co-senior author.*

Received: *17 September 2016*
Accepted: *12 December 2016*
Published: *27 December 2016*

Citation:
Li K, Kozyrev V, Kyllingsbæk S, Treue S, Ditlevsen S and Bundesen C (2016) Neurons in Primate Visual Cortex Alternate between Responses to Multiple Stimuli in Their Receptive Field. Front. Comput. Neurosci. 10:141. doi: 10.3389/fncom.2016.00141

1. INTRODUCTION

The receptive field (RF) of a neuron in the visual system is the region within the visual field in which stimulation can affect the neuron's response. To understand visual information processing, it is fundamental to understand how the benefits of large RFs (integrating spatial information to allow encoding more complex and spatially extensive visual stimuli) are achieved without the loss of

spatial precision caused by combining the responses to multiple stimuli in the RF into one response of the neuron.

In primary visual cortex, RFs are small, allowing for a direct high-resolution representation of stimulus position in retinotopic coordinates. Moving up the hierarchy of extrastriate visual areas, both in the temporal and dorsal pathways, RF sizes grow substantially (Smith et al., 2001; Gattass et al., 2005). This is generally seen as an adaptation to the functional specialization of these areas for more complex aspects of the visual environment, creating a need for integrating information over larger spatial areas, such as when encoding faces (Kanwisher and Yovel, 2006) in the ventral pathway or optic flow patterns (Gilmore et al., 2007) in the dorsal pathway. However, the benefit of spatial integration comes with the cost of a loss of information about the individual features when multiple stimuli fall in the RF, which happens frequently in mid- or high-level visual cortical areas (Orhan and Ma, 2015).

Most single-cell studies on processing in extrastriate visual cortex have focused on single stimuli, and most studies of responses to multiple stimuli have viewed the recorded activities as an integration of the responses that would have been evoked by each of the stimuli presented alone. This approach has led to the observation that the average firing rate to multiple stimuli is not the sum but rather a weighted average of the responses evoked by the individual stimuli when these are presented alone (Recanzone et al., 1997; Britten and Heuer, 1999; Reynolds et al., 1999; Zoccolan et al., 2005; Busse et al., 2009; Lee and Maunsell, 2009; MacEvoy et al., 2009; Reynolds and Heeger, 2009; Nandy et al., 2013). Here we show that looking only at the responses to multiple stimuli averaged across many trials has obscured the possibility that neurons multiplex the responses to the individual stimuli in time, shifting between response states dominated by individual stimuli (Bundesen et al., 2005; Bundesen and Habekost, 2008).

Reynolds et al. (1999) showed that a typical cell in visual area V2 or V4 responds to a pair of objects in its classical RF by adopting a rate of firing which, averaged across trials, equals a weighted average of the firing rates when objects are presented alone. We analyzed two opposing models, the two models being prototypes for how multiple stimuli are being processed on the single trial level, and both leading to the observed average behavior over trials. In the response-averaging model (e.g., Reynolds et al., 1999), the firing rate of a cell to a pair of stimulus objects in its classical RF is a weighted average of the firing rates to the individual objects. By contrast, in the probability-mixing model (Bundesen et al., 2005), the cell responds to the pair of objects as if only one of the objects were present in any given trial. Here we compare the abilities of the two models to account for spike trains recorded from single cells in area MT in response to (a) unidirectional moving random dot patterns (RDPs) presented singly in the RF and (b) nonoverlapping bidirectional pairs of such patterns in the RF. For unidirectional patterns, the two models coincide. Results from bidirectional pairs support the probability-mixing model over the response-averaging model.

2. MATERIALS AND METHODS

2.1. Experimental Procedures

The comparison between the response-averaging model and the probability-mixing model was performed by analysis of spike trains recorded from single cells in area MT. The data and computer code are available at Li et al. (2016). In this study, two rhesus monkeys were trained to perform visual tasks (see **Figure 1A**). Before each trial of the main experiment, a fixation spot (small red square) appeared in the middle of a computer screen. The monkey was trained to maintain its gaze on the fixation spot throughout each trial. It initiated a trial by pressing a lever. Immediately afterwards a cue was presented, which specified a target stimulus. The target, which could be either a RDP (*attend-in* condition) or the fixation spot (*attend-fix* condition), was later presented during the trial shown alone or together with distracting RDPs. The monkey was rewarded with a drop of juice for detecting a transient change in the target and responding by releasing the lever within 150–650 ms after the change.

In the *attend-fix* condition, the color of the fixation spot changed from red to gray when the monkey pressed the lever. The monkey was supposed to keep attention on the fixation spot. After 600 ms, two distractor RDPs were presented inside the RF of the recorded MT neuron and two were presented outside the RF (see **Figure 1A**). Each distractor pattern could change its motion (by increase in speed with 67% or clock- or counterclockwise change in direction by 45°) for a period of 130 ms beginning at a randomly chosen moment between 800 and 2400 ms after the onset of the RDPs. The monkey was required to detect a luminance change in the fixation spot which occurred within the same time window. For all cells, spike trains were recorded when two nonoverlapping patterns were simultaneously present in their RFs. For the majority of cells, spike trains were also recorded when only one pattern was present in the RF (at one of two locations, aperture 1 or 2, used for the bidirectional stimulation; see the *unidirectional conditions fix1* and *fix2* in **Figure 1A**).

In the *attend-in* condition, the fixation spot remained red during the whole trial. The cue was a moving RDP in aperture 1 presented for 600 ms. It had the same location and moved in the same direction as the target RDP. After the cue, a blank screen was shown for 800 ms (delay) followed by a display of the target RDP accompanied by three distractor RDPs. The first change in motion within the trial took place between 400 and 1200 ms after the onset of the patterns and could occur in either the target or one of the distractors. The transient change in speed or direction of motion was the same as the change used in the *attend-fix* condition. The target change took place inside aperture 1 in the RF of the recorded neuron.

In the bidirectional conditions, direction of motion in aperture 2 was always 120° clockwise relative to that in aperture 1 (see **Figure 1A**). To determine a direction tuning curve for a neuron in a given condition (see **Figure 2**), both motion components were varied in steps of 30°. In all cells, full tuning curves were determined for the *attend-fix* and *attend-in*

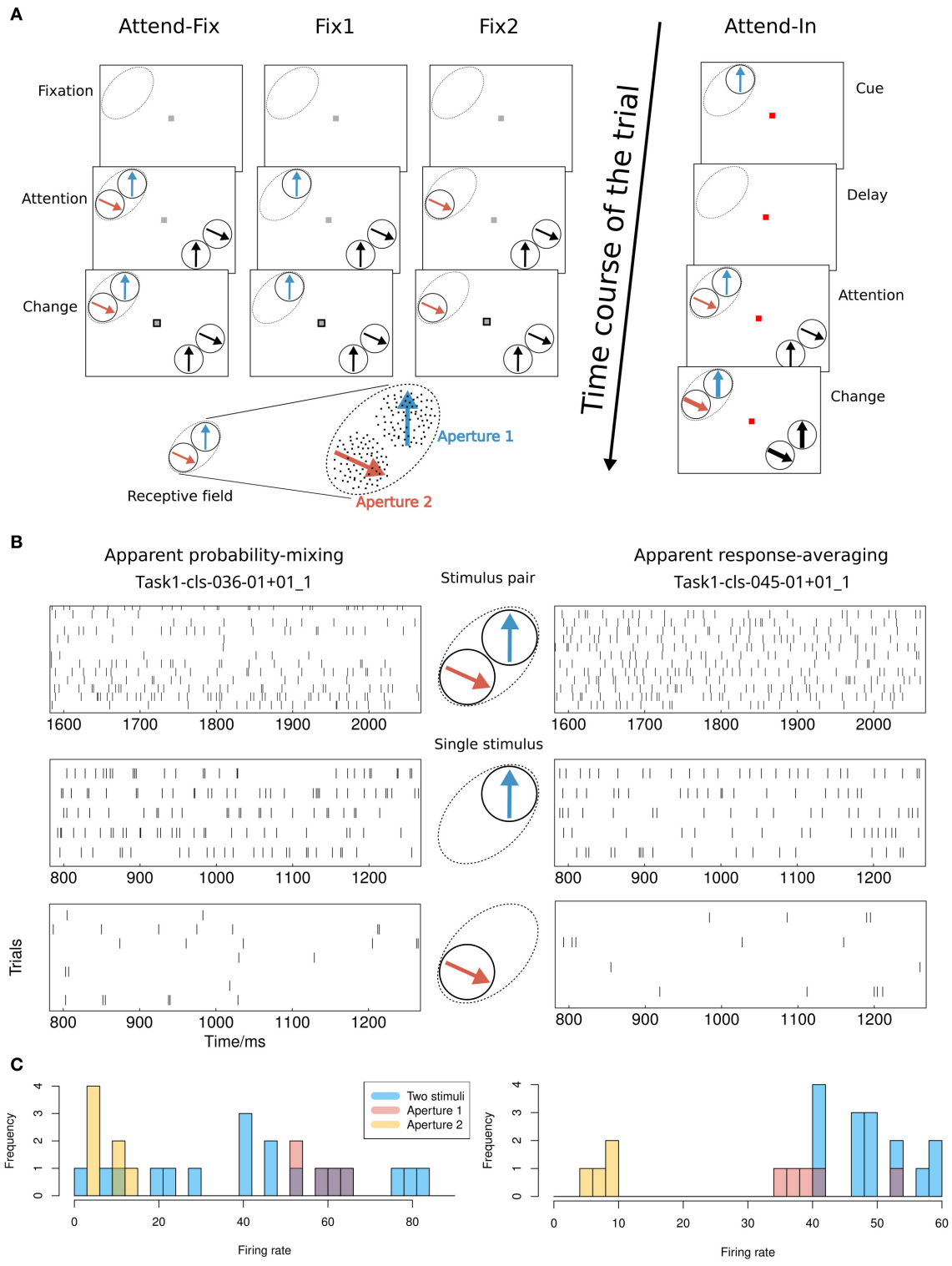


FIGURE 1 | Experimental setup and possible results. (A) Visual stimuli and behavioral tasks. The lower left shows an example of the stimulus layout in the RF. The classical RF of an MT neuron is indicated by dashed ovals (not visible on the screen). Bidirectional motion patterns composed of two adjacent separated RDPs that moved within two stationary virtual apertures were used both inside and outside the RF. Apertures 1 and 2 were placed within the RF. In the bidirectional-motion condition *attend-in*, the monkey was required to detect a transient change in either speed or direction of motion of the cued target RDP. In the bidirectional-motion condition *attend-fix* and unidirectional motion conditions *fix1* and *fix2*, the monkey was required to detect a transient change in the luminance of the fixation spot. **(B)** Possible results. Illustration of the difference between the probability-mixing model and the response-averaging model by spike trains generated by stimulus pairs and single stimuli, respectively. The spike trains are taken from two neurons indicated in the scatter plots in **Figure 4A** as a square (apparent probability-mixing) and a triangle (apparent response-averaging). **(C)** Histograms of the empirical firing rates of the data in **(B)**.

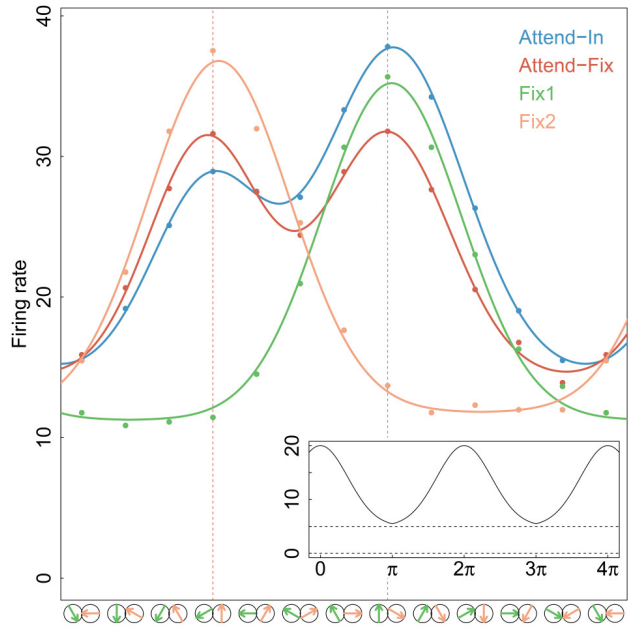


FIGURE 2 | Tuning curves. Gaussian tuning curves (full-drawn lines) fitted to the average firing rates (dots) at each direction of movement and for each experimental condition of the 84 neurons that were tested in both bi- and uni-directional trials. In the analysis, each neuron has its own tuning curve, here an average curve is shown for illustration. The directions of movements in apertures 1 and/or 2 are shown by green and yellow arrows, respectively, along the *x*-axis. The two vertical dotted lines indicate the stimulus directions that were closest to the preferred direction, in aperture 1 (right) and aperture 2 (left). An example of the periodic Gaussian function, Equation (7), is shown in the insert on bottom-right, with parameters $A = 15, D = 0, \sigma = 1.2, r_0 = 5$.

conditions. Recording of responses to the unidirectional components of the bidirectional stimuli, when each of the components was presented alone, provided two additional tuning curves.

2.1.1. Monkey Training and Surgery

Two male rhesus monkeys (*Macaca mulatta*) were extensively trained to perform visual attentional tasks. The animals were implanted with a custom-made titanium implant to prevent head movements during training and recording, and a recording chamber (Crist Instruments, Hagerstown, MD, USA) on top of a craniotomy over the left (monkey C) or the right (monkey H) parietal lobe. The chamber positions were based on anatomical MRI scans.

All animal procedures of this study have been approved by the responsible regional government office [Niedersächsisches Landesamt für Verbraucherschutz und Lebensmittelsicherheit (LAVES)] under the permit numbers 33.42502/08-07.02 and 33.14.42502-04-064/07. The animals were group-housed with other macaque monkeys in facilities of the German Primate Center in Goettingen, Germany in accordance with all applicable German and European regulations. The facility provides the animals with an enriched environment (including a multitude of toys and wooden structures; Calapai et al., 2016), natural as well as artificial light, exceeding the size requirements of the European regulations, including access to outdoor space. Surgeries were performed aseptically under isoflurane anesthesia using standard techniques (see Martinez-Trujillo and Treue, 2004), including appropriate peri-surgical analgesia and monitoring to minimize potential suffering. The German Primate Center has several staff veterinarians that regularly monitor and examine the animals and consult on any procedures. During the study the animals had unrestricted access to food and fluid, except on the days where data were collected or the animal was trained on the behavioral paradigm. On these days the animals were allowed unlimited access to fluid through their performance in the behavioral paradigm. Here the animals received fluid rewards for every correctly performed trial. Throughout the study the animals' psychological and medical welfare was monitored by the veterinarians, the animal facility staff and the lab's scientists, all specialized on working with non-human primates. The two animals were healthy at the conclusion of our study and were used in follow-up studies.

2.1.2. Experimental Procedure

Single unit action potentials were recorded extracellularly with single tungsten electrodes (FHC, Inc., Bowdoinham, ME, USA) after penetration of the dura with a sharp guide tube. The electrode was advanced using a hydraulic micropositioner (David Kopf Instruments, Tujunga, CA, USA). Impedances ranged from 0.5 to 2.8 MΩ. Neuronal activity was amplified and filtered (bandpass 150–5000 Hz). Action potentials in the majority of recorded units were sorted online using the Plexon data acquisition system (Plexon Inc., Dallas, TX, USA). In the first recording sessions action potentials were isolated using a window discriminator (BAK Electronics Inc., Mount Airy, MD, USA). Area MT was identified by its anatomical position, the high proportion of direction-selective cells, and the typical size-eccentricity relationship of RFs. Eye positions were monitored using a video-based eye tracking system (ET-49, Thomas Recording, Giessen, Germany). Eye positions were sampled at 230 Hz, digitized and stored at 200 Hz. Fixation was controlled during the recordings to stay within a window of 1.2° radius around the fixation spot.

2.1.3. Visual Stimuli

The experiments were conducted using an Apple Macintosh computer running custom software and a Sony Trinitron (22″) monitor with 75 Hz refresh rate. The monkey viewed the display binocularly in a dimly lit room from a distance of 57 cm. The spatial resolution of the display was 40 pixels per degree of visual angle. The shape of the RF, as well as its preferred direction and speed were estimated in a separate mapping and tuning session performed before the main task. The bidirectional stimuli were two RDPs presented within stationary adjacent virtual apertures matching the excitatory part of the RF (see **Figure 1A**). Another pair of RDPs was presented far outside the RF in the opposite visual hemifield symmetrically to the first pair with respect to the fixation point. Each RDP had a density of 10 dots per square degree. The width of each dot was 6 min of arc. All dots were white (luminance 85 cd/m^2) and were displayed on a gray background (luminance 15 cd/m^2). The basic speed of

the dots in the RDP was matched to the preferred speed of the neuron, usually between 4 and 16°/s. The 12 directions of the patterns used to recover the tuning curve were chosen such that one of them was well-aligned with the preferred direction of the neuron.

See also Kozyrev et al. (under revision) for more details on monkey training and surgery, experimental procedures, and visual stimuli.

2.2. Data Used for Analysis

The recorded spike trains covered about the first 3000 ms of each trial. **Figure 3** shows all spike trains from an example neuron. The periods of fixation, cue, delay, and intervals extracted for analysis are indicated with different colors. The onset of the target is indicated by the red dashed lines. Clear *delay* and *burst* effects are seen: When the RDP appears on the screen, the neuron has after a short delay a period of bursting behavior. We excluded the first 200 ms because of a large variability in the strength and length of the initial transient period around 50–200 ms. The latter was probably dependent on adaptation to the cue and other factors which are not considered by the relatively simple models we tested here. Thus, only the time interval from 200 to 700 ms after the onset of the RDPs were analyzed. Excluding the transient response epoch in the analysis is widely done, and this time window is also used by Katzner et al. (2009) and Martınez-Trujillo and Treue (2002) as the period where the MT neurons show robust attentional modulation. In case the speed or direction of motion of an RDP changed before 700 ms, the analysis interval terminated when the change occurred. We chose this interval for analysis in order to bypass the delay and burst periods and analyze an approximately constant firing rate.

In total 166 neurons have been recorded. However, we required at least two spike trains for each condition to include a neuron into further analysis, which resulted in 109 analyzed neurons. Summary statistics on number of trials and neurons can be found in **Table 1**. In an *attend-out* condition, the target always moved in either the preferred or the null direction of the recorded neuron, and the stimulus in the RF always moved in the preferred direction. Accordingly, the results from the attend-out condition could not be analyzed on a par with results from the other conditions. These data were therefore discarded. Regarding behavioral performance, we only included trials where the monkey detected the transient change and responded correctly (see Experimental procedures).

2.3. Notation

Index d indicates the 12 directions: $d \in \{0, \frac{\pi}{6}, \frac{2\pi}{6}, \ldots, \frac{11\pi}{6}\}$, and $l \in \{1, 2\}$ indicates (the location of) the stimulus, which is either aperture 1 or aperture 2. The index $c \in \mathcal{C}$ indicates the experimental condition; $\mathcal{C} = \{$*attend-fix, attend-in, fix1, fix2*$\}$. In condition *fix1*, the unidirectional RDP appears in aperture 1, and in *fix2*, the RDP appears in aperture 2. Consider the time interval $(0, T]$ extracted for analysis, where for simplicity we set the start point 200 ms after onset of the stimulus to be at time 0, and thus $T \leq 500$ ms. The interval contains a sequence of N spikes: $0 < t_1 < t_2 < \cdots < t_N < T$, where t_i is the time of occurrence of the ith spike. We write $\tau = (0, t_1, t_2, \ldots, t_N, T)$, and $N(t)$ denotes the number of spikes that occurred in the time interval $(0, t]$ for $0 < t \leq T$.

2.4. Data Analysis

The spike trains were modeled as stochastic point processes (Truccolo et al., 2005; Kass et al., 2014, chap. 19). The conditional intensity function (Daley and Vere-Jones, 1988) of a general point process model is defined by

$$\lambda(t|H_t) = \lim_{\Delta t \to 0} \frac{\Pr(N(t+\Delta t) - N(t) = 1|H_t)}{\Delta t}, \qquad (1)$$

where H_t denotes the spike history up to time t. Then $\lambda(t|H_t)\Delta t$ approximates the probability of observing a spike in $(t, t+\Delta t]$ for Δt small.

The likelihood of observing spike train τ is (Daley and Vere-Jones, 1988; Kass et al., 2014)

$$L(\tau;\theta) = \left[\prod_{i=1}^{N} \lambda(t_i|H_{t_i};\theta)\right] \exp\left\{-\int_0^T \lambda(s|H_s;\theta)ds\right\} \qquad (2)$$

where θ is a vector of model specific parameters, which should be estimated from data. The parameter vector θ for the two models will be specified in Section 2.5. In practice the measurements of the spike times are discrete, indicating whether or not they occur in time intervals of length $\Delta t = 1$ ms, where Δt is so small that it contains at most one spike and the conditional intensity function can be assumed constant within each interval. We approximate the integral in Equation (2) by a discrete sum and obtain

$$L(\tau;\theta) \approx \left[\prod_{i=1}^{N} \lambda(t_i|H_{t_i};\theta)\right] \exp\left\{-\sum_{n=1}^{\frac{T}{\Delta t}} \lambda(n\Delta t|H_{n\Delta t};\theta)\Delta t\right\}. \qquad (3)$$

Truccolo et al.' modeled the spike train as a discrete sequence of conditional Bernoulli events, and obtained the same result as Equation (3) through probability mass functions (Truccolo et al., 2005).

Spike trains from different trials are assumed independent, and the likelihood of the entire data set will therefore be the product of individual likelihoods of the form (equation 2). Parameters are assumed constant for all trials from a neuron, but can differ from neuron to neuron. The estimation can therefore be done individually for each neuron. For each neuron, the likelihood of the recorded spike trains was computed by use of the conditional intensity function assuming, in turn, the response-averaging and the probability-mixing models.

The *conditional intensity function* is modeled with three components: (1) a base firing rate, r_l, computed using Gaussian tuning curves (see below), which describes the effect of stimulus l and its direction of movement; (2) a scaling function depending on time, $a(t)$; and (3) the effects of the spike history, $h(H_t)$. It is assumed to be of the following form:

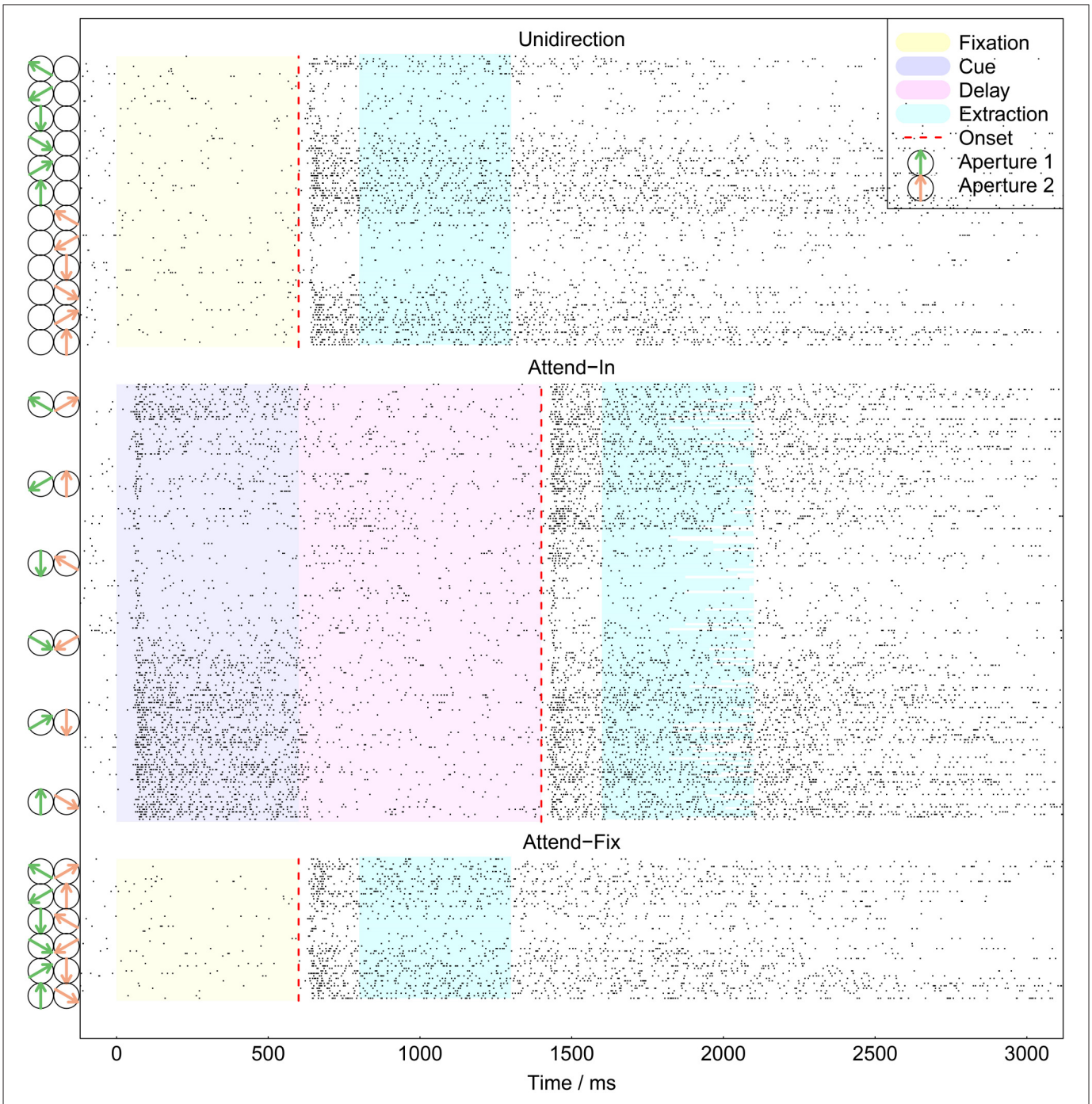


FIGURE 3 | All recorded spike trains from example neuron. Spikes are shown as points. The extraction intervals indicate the data used for analysis. They varied in length between trials in the *attend-in* condition, in which the imperative change in the target RDP could happen <700 ms after the onset of the stimulus. At time 0 the monkey pressed a lever to start a trial. The dashed red lines indicate the onset of the target. To the left are shown the presented RDPs, for readability only every second stimulus is shown.

$$\lambda(t|H_t) = r_l \exp\left\{a(t) + h(H_t)\right\}. \quad (4)$$

The trend in the firing rate is modeled linearly (Cox and Lewis, 1966), $a(t) = \gamma_0 t$, where γ_0 is a parameter. Since the firing rate decreases over time, γ_0 is expected to be negative.

For the history component we use linear addition of the spikes in the past m time units:

$$h(H_t) = \sum_{i=1}^{m} \gamma_i \Delta N_{t-i\Delta t}, \quad (5)$$

TABLE 1 | Summary statistics of sample sizes.

Condition	Number of combinations	Quantiles of number of trials				
		Min	10%	50%	90%	Max
	Neurons × stimuli					
fix1	84 × 12	2	3	4	6	7
fix2	84 × 12	2	3	4	6	7
attend-fix	109 × 12	2	3	4	7	16
attend-in	109 × 12	3	8	12	18	31
	Neurons					
fix1	84	25	34	48	69	84
fix2	84	25	34	48	70	84
attend-fix	109	25	36	55	85	186
attend-in	109	61	90	138	207	272

The neurons measured in conditions fix1 *and* fix2 *are a subset of the neurons measured during the other two conditions.*

where $\Delta N_t \in \{0, 1\}$ denotes whether or not there is a spike in the interval $[t, t + \Delta t)$. Parameter γ_i is a spike response weight and quantifies the effect of having a spike i steps back in time. If it is negative, the effect is inhibitory, if it is positive it is excitatory. In the data analysis, $m = 10$ has been used. We have repeated the analysis with other memory lengths, but for larger m, the estimates of γ_i were close to zero, and the estimates of other parameters were stable, not changing the conclusions from the analysis, see **Figure 9F**.

The final model for the conditional intensity function used in the analysis is thus:

$$\lambda(t|H_t) = r_l \exp\left[\gamma_0 t + \sum_{i=1}^{10} \gamma_i \Delta N_{t-i\Delta t}\right]. \quad (6)$$

A *Gaussian tuning curve* is used to model the firing rate r as function of direction of motion d, with mean in the preferred direction, D, of the neuron. The preferred direction was estimated in a separate mapping and tuning session performed before the main task. For simplicity, we therefore set $D = 0$, and measure the direction of the stimulus RDP in deviation from the preferred direction. Since the stimulus is a direction (an angle), the rate function should be periodic with period 2π, and we apply the method given by Shokhirev et al. (2006), see also Treue S. and Trujillo J. (1999). For a neuron responding to stimulus l moving in direction d, the firing rate is given by

$$r_l = f(d|A_l, \sigma_l, r_0) = A_l \exp\left[-\frac{\| d - D \|_{2\pi}^2}{2\sigma_l^2}\right] + r_0, \quad (7)$$

where A_l denotes the amplitude (directional gain), σ_l denotes the standard deviation (selectivity of the preferred direction), and r_0 is the spontaneous firing rate in absence of a stimulus. The first two depend on the stimulus. The function $\| d - D \|_{2\pi} = \mathrm{mod}(d - D + \pi, 2\pi) - \pi$ ensures that the firing rate is periodic and symmetric around D. **Figure 2** shows the mean firing rates fitted by Gaussian tuning curves. The unidirectional cases are modeled by single Gaussian curves, and the bidirectional cases are modeled by a mixture of two Gaussian curves. Along the x-axis, an upward arrow indicates the preferred direction. The insert illustrates the periodic function.

2.4.1. Stimulus Weights in Bidirectional Conditions

In the *attend-fix* condition two RDPs are shown in the RF, one in aperture 1 and one in aperture 2. The neuron may favor one location over the other, which is modeled by assigning a weight to each location. These weights will be modified in the *attend-in* condition, where the weight to the attended location is expected to increase. Let $w_{c,l}$ denote the weight of stimulus l under a bidirectional experimental condition c, such that $w_c = w_{c,1} + w_{c,2}$ denotes the sum of the weights. Let $p_c = w_{c,1}/w_c$ and $1 - p_c = w_{c,2}/w_c$ denote the normalized weights.

2.4.2. Attentional Scaling Parameters

In the *attend-in* condition, a prior cue shows a replica of the stimulus to be attended (stimulus 1) including its location and direction of movement. The cue causes a multiplicative increase in the rate of firing in response to the cued stimulus (the stimulus in aperture 1). Sometimes the cue also changes the rate of firing in response to the uncued stimulus (stimulus 2). We use a scaling parameter a_l multiplying the amplitude A_l to model such attentional effects for stimulus l. The resulting firing rate is $r_l = f(d|a_l A_l, \sigma_l, r_0)$. Without loss of generality, the scaling parameter a may be assumed to have a value of 1 in conditions *attend-fix*, *fix1*, and *fix2*, in which directions of movement are irrelevant to the task to be performed by the monkeys.

2.5. Models

Let the rates of firing of the recorded cell be r_1 and r_2, respectively, when objects 1 and 2 are presented alone in the classical RF of the cell.

The *probability-mixing* model assumes a neuron responds to one and only one of the stimuli within its RF at a time, and the probability of responding to a particular stimulus depends on the weight of that stimulus. Hence, the probability that a neuron under a bidirectional experimental condition c reacts to stimulus l is given by p_c. Thus,

$$r = \begin{cases} r_1, & \text{with probability } p_c \\ r_2, & \text{with probability } 1 - p_c \end{cases}, \quad (8)$$

where r_l is given by Equation (7), except that A_l is substituted by $a_l A_l$ in the *attend-in* condition. The likelihood of all data from one neuron is then

$$\mathcal{L}(\theta) = \prod_{c \in \mathcal{C}} \prod_{k=1}^{12} \prod_{j=1}^{m_{c,k}} \left(p_c L(\tau_{c,k,j}; \theta_1) + (1 - p_c) L(\tau_{c,k,j}; \theta_2)\right), \quad (9)$$

where $p_c = 1$ in the *fix1* condition, $p_c = 0$ in the *fix2* condition, the individual likelihoods $L(\cdot; \cdot)$ are given by Equation (2), and θ_l contains the stimulus specific parameters (A_l, σ_l, a_l) besides the common parameters $(r_0, p_{attend-in}, p_{attend-fix}, \gamma_0, \gamma_1, \ldots, \gamma_{10})$.

Thus, θ contains all 20 parameters. Here $\tau_{c,k,j}$ denotes the spike train under the cth condition, kth direction and jth trial, where $m_{c,k}$ is the number of trials under the specific experimental condition.

A numeric overflow issue arises when computing the log likelihood, since it may contain the logarithm of the sum of two small numbers, $\log(\delta_1 + \delta_2)$. This happens especially when the spike train is long, and the current parameters in the optimization algorithm are far from the optimal ones. We apply the *log-sum-exp* formula (Press, 2007): $\log(\delta_1 + \delta_2) = \log^* \delta + \log(e^{\log \delta_1 - \log^* \delta} + e^{\log \delta_2 - \log^* \delta})$, where $\log^* \delta = \max(\log \delta_1, \log \delta_2)$.

The *response-averaging model* assumes the firing rate to be a weighted average rate over all stimuli,

$$r = p_c r_1 + (1 - p_c) r_2. \tag{10}$$

The likelihood is

$$\mathcal{L}(\theta) = \prod_{c \in \mathcal{C}} \prod_{k=1}^{12} \prod_{j=1}^{m_{c,k}} L(\tau_{c,k,j}; \theta). \tag{11}$$

The number of parameters in the response-averaging model is one less than the probability-mixing model, because in the *attend-in* case not all three parameters $(p_{attend-in}, a_1, a_2)$ can be identified. We define $b_1 = p_{attend-in}\, a_1$ and $b_2 = (1 - p_{attend-in})\, a_2$. In **Table 2** the parameters entering in θ for the two models are summarized.

In the unidirectional conditions, the response-averaging model and the probability-mixing model make the same predictions, and the firing rate is given by Equation (7). In the bidirectional conditions, the predictions of the two models differ as follows. In the *response-averaging* model, the firing rate to a stimulus pair in the *attend-fix* condition is a weighted average of the responses (firing rates) obtained to the individual stimuli in the unidirectional conditions (given by equation 7). However, the firing rate to a stimulus pair in the *attend-in* condition is a weighted average of *scaled* versions of the responses to the individual stimuli in the unidirectional conditions, where the scaling factor (*gain factor*) for a stimulus varies with the location of the stimulus (aperture 1, which showed the stimulus to be attended, vs. aperture 2, which showed a stimulus to be ignored). In the *probability-mixing* model, the firing rate to a stimulus pair in the *attend-fix* condition is a probability mixture of the responses (firing rates) to the individual stimuli in the unidirectional conditions. The firing rate to a stimulus pair in the *attend-in* condition is a probability mixture of scaled versions of the responses to the individual stimuli in the unidirectional conditions, where the scaling factor (gain factor) for a stimulus again varies with the location of the stimulus (aperture 1 vs. aperture 2).

TABLE 2 | Parameters entering the parameter vector θ of the two models.

Model	Parameter	Explanation
Common	γ_0	Decay constant
	$(\gamma_1, \gamma_2, \ldots, \gamma_{10})$	Spike response weights
	(A_1, D_1, σ_1)	Parameters for the tuning curve of stimulus 1
	(A_2, D_2, σ_2)	Parameters for the tuning curve of stimulus 2
	r_0	Spontaneous firing rate
	$p_{attend-fix}$	Probability/weight of stimulus 1 in *attend-fix*
Probability-mixing	$p_{attend-in}$	Probability of stimulus 1 in *attend-in*
	a_1	Attentional scaling of stimulus 1
	a_2	Attentional scaling of stimulus 2
Response-averaging	$b_1 = p_{attend-in} \cdot a_1$	Identifiable parameter for stimulus 1
	$b_2 = (1 - p_{attend-in}) \cdot a_2$	Identifiable parameter for stimulus 2

2.5.1. Diagnostic Neurons

Whereas, some neurons are highly diagnostic in distinguishing between the response-averaging and the probability-mixing model when a certain pair of stimuli is presented in apertures 1 and 2 (see **Figure 1A**), responses of other neurons cannot be used for distinguishing between the two models. One example of a neuron that fails to distinguish between the models is a neuron that almost always responds as if only the stimulus in aperture 1 is present. Such a neuron behaves (to an arbitrarily good approximation) in accordance with a response-averaging model in which the response to the stimulus in aperture 1 is weighted much stronger than the response to the stimulus in aperture 2. At the same time, the neuron behaves in accordance with a probability-mixing model in which the probability of responding to the stimulus in aperture 1 is nearly 1. This, however, does not mean aperture 2 is not inside the RF, since the neuron does respond when a single stimulus is present in either aperture 1 or 2 alone. The above example occurs if one stimulus has a much stronger attentional weight than the other.

Another example of a neuron that cannot be used for distinguishing between the two models is a neuron in which the rate of firing is nearly the same for the stimulus in aperture 1 as for the stimulus in aperture 2. Regardless of the distribution of weights across the two stimuli, the neuron behaves in accordance with both a response-averaging model (averaging equals single firing rates) and a probability-mixing model (mixing equals single firing rates). In our experimental setup this is never the case, since the bidirectional stimuli always differ with 120 , and for all neurons there are trials where this difference force firing rates to be different, as seen from the Gaussian tuning curves in **Figure 2**.

Examples of neurons that are highly diagnostic in distinguishing between the response-averaging and the probability-mixing model are neurons with close to equal weighting of stimuli in apertures 1 and 2 but very different

responses to the two stimuli. **Figure 1B** exemplifies the expected behavior of such neurons according to the probability-mixing model and according to the response-averaging model, respectively. **Figure 1C** shows histograms of empirical firing rates of the corresponding spike trains in **Figure 1B**. As can be seen, according to the probability-mixing model, the neuron responds either to stimulus one or to stimulus two, which generates a wide variation in firing rates (bimodal distribution). In contrast, by the response-averaging model, the responses to stimulus pairs all have similar rates (unimodal distribution). We defined a *diagnostic neuron* based on the estimated probabilities (in the probability-mixing model) or the weights (in the response-averaging model). These two example neurons are indicated by a square and a triangle, respectively, in **Figure 4**. We call a neuron diagnostic if either the two $p_{attend-fix}$ estimates from the two models both are between 0.2 and 0.8, or if $p_{attend-in}$ in the probability-mixing model fulfills the same criterion. This provides 90 diagnostic neurons, out of the 109 analyzed neurons. All analyses were performed on the entire data set, but where relevant, we indicate partial results only including the diagnostic neurons, and we highlight the type of neuron in the figures.

Note that whether a neuron is diagnostic or not does not reflect how well the models fit the data of that neuron. It only indicates that diagnostic neurons behave differently under the two models, whereas non-diagnostic neurons behave similarly under the two models, and contain little information for model selection.

2.5.2. Relation of Probability-Mixing Model to NTVA

The probability-mixing model is closely related to the Neural Theory of Visual Attention (NTVA) (Bundesen et al., 2005; Bundesen and Habekost, 2008). Attentional weights and the ways they are computed and used are the same in the probability-mixing model as in NTVA. In particular, in both the probability-mixing model and NTVA, the probability that an MT neuron represents an object x in its classical receptive field equals

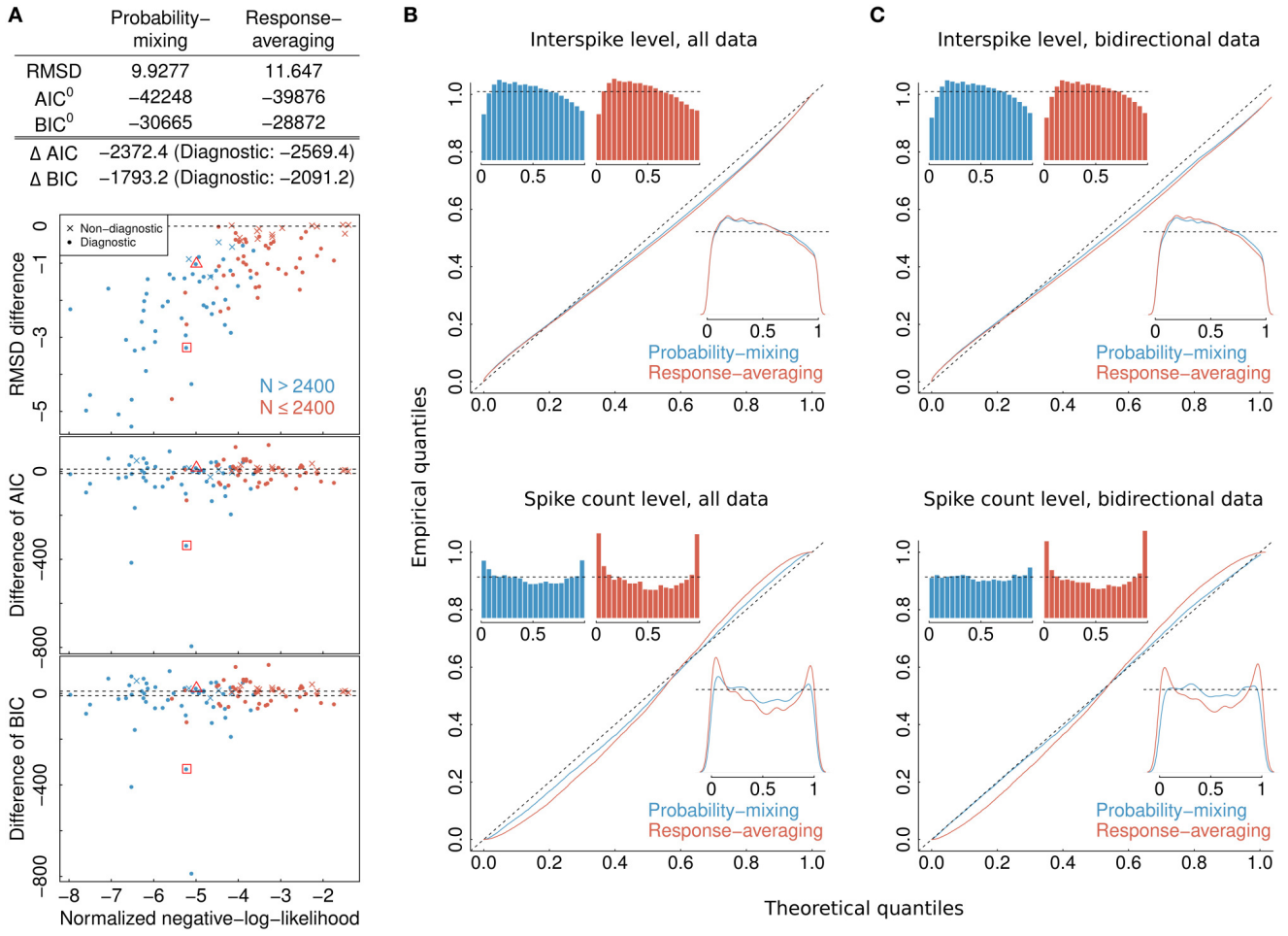


FIGURE 4 | Model selection and model checking. (A) Differences in BIC, AIC, and RMSD values between the probability-mixing model and the response-averaging model (the former minus the latter). In all three cases, a smaller value means a better fit, so negative differences favor the probability-mixing model, whereas positive differences favor the response-averaging model. The squared and the triangled points are the example neurons from **Figure 1B**. The table on top provides the total saturated AIC, saturated BIC and RMSD values for each model. **(B)** QQ-plots of uniform residuals on interspike level (top) and on spike count level (bottom) for both models based on all observed data. The inserts are histograms (top) and density plots (bottom) of the uniform residuals. **(C)** The same as in **(B)** but calculating the uniform residuals on only bidirectional data.

the attentional weight of object x divided by the sum of the attentional weights of all objects in the receptive field of the neuron. Also, the nature of attentional weights is the same in the two models. Thus, in both models, the attentional weights may depend on many different features of the objects, including features computed in areas other than MT.

Consider a trial in which an MT neuron that prefers motion in direction D represents a stimulus l, moving in direction d. Given that the neuron represents stimulus l, it responds as though stimulus l were the only object in its receptive field. By the rate equation of NTVA, the activation of the neuron, $v(l, D)$, equals the product of the strength of the sensory evidence that stimulus l moves in direction D, $\eta(l, D)$, and the bias in favor of seeing movement in direction D, β_D. In the current article, among others (Bundesen and Habekost, 2008), a base rate, r_0, is effectively added to the product of $\eta(l, D)$ and β_D. Thus, according to NTVA,

$$v(l, D) = \eta(l, D)\beta_D + r_0, \tag{12}$$

where $\eta(l, D)$ may be given by

$$\eta(l, D) = A_l \exp\left[-\frac{\| d - D \|^2_{2\pi}}{2\sigma^2}\right] \tag{13}$$

as suggested by Equation (7). By NTVA, $\eta(l, D)$ is independent of attention, but the bias parameter β_D depends on the attentional condition. In conditions *attend-fix*, *fix1*, and *fix2*, directions of motion are task-irrelevant, whence β_d (a measure of the importance of seeing motion in direction d) is a small number, say, β_0, for all directions d. In condition *attend-in*, however, the stimulus in aperture 1 moves in the cued direction, whence β for its actual motion direction (= the cued direction) has a large value (say, β_1). Thus, the categorization that the stimulus in aperture 1 moves in the cued direction is supported by both sensory evidence and perceptual bias. By contrast, in the same condition, the stimulus in aperture 2 moves in a direction that diverges from the cued direction by 120°, whence β for its actual motion direction has a smaller value (say, β_2).

By Equations (12) and (13), the predicted firing rates remain constant if all β values are multiplied by a positive constant k while all amplitude parameters A_l are divided by the same constant k. Accordingly, without loss of generality, β_0 can be set to a value of 1 if (i) β_1 and β_2 are changed in direct proportion to β_0 and (ii) amplitude parameters A_1 and A_2 are changed in inverse proportion to β_0. After these rescalings, the resulting values of β_1 and β_2 can be identified with scaling parameters a_1 and a_2, respectively. That is, $a_1 = \beta_1/\beta_0$ and $a_2 = \beta_2/\beta_0$.

Finally, we can extend NTVA to account for effects of presentation time t and spike history H_t by letting $v(l, D, t|H_t)$ be the conditional intensity function for a spike train and assuming that

$$v(l, D, t|H_t) = v(l, D) \exp\left[\gamma_0 t + \sum_{i=1}^{10} \gamma_i \Delta N_{t-i\Delta t}\right], \tag{14}$$

where $v(l, D)$ is given by Equation (12).

In the suggested interpretation, the cue shown in the *attend-in* condition cues a particular direction of motion to be attended by pigeonholing (i.e., by setting β high for this direction) (Bundesen et al., 2005; Bundesen and Habekost, 2008). In addition to being used for pigeonholing, the cue can also be used for filtering (Bundesen et al., 2005; Bundesen and Habekost, 2008), in particular, filtering by location (by giving high attentional weight to stimuli that are located in aperture 1) and/or filtering by direction of motion (giving high attentional weight to stimuli that are moving in a particular direction).

2.6. Model Selection by Relative Goodness of Fit and Cross-Validation

The main aim of our article is to compare the abilities of the probability-mixing and the response-averaging models to explain the data. To select the best-fitting model, we use the Bayesian Information Criterion (BIC) and the Akaike information criterion (AIC), which compare likelihood values correcting for the number of parameters (Burnham and Anderson, 2002). Since only the difference of AIC (BIC) can be used for model comparison (Burnham and Anderson, 2002; Claeskens and Hjort, 2008), we subtract out the null deviance from the AIC (BIC) values for both models while preserving the difference. The null deviance is defined by $-2\log(L^0)$, where L^0 is the likelihood value of the null model assuming that all spike trains from one neuron have the same firing rate. Given the two models, the weight in favor of the model with the lowest AIC (BIC) value is given by $1/(1 + \exp(-\Delta/2))$ (Burnham and Anderson, 2002; Claeskens and Hjort, 2008), where Δ is the difference between the two AIC (BIC) values, and the weight in favor of the model with the highest value is given by $\exp(-\Delta/2)/(1 + \exp(-\Delta/2))$. Heuristically, the weight can be interpreted as the probability of the model to be the best among the considered models, in the sense of Kullback-Leibler information loss (Burnham and Anderson, 2002; Claeskens and Hjort, 2008).

This approach of statistical model selection to determine the most plausible model, each offering opposing biological explanations, using advanced statistical point process models to analyze single spike trains instead of trial-averaged responses, was also employed recently in Latimer et al. (2015). Here they determine whether firing rates during decision-making in the macaque lateral intraparietal area are gradually accumulating evidence toward a decision threshold, or whether decisions are taken as instantaneous jumps in the firing rates.

Model selection was done on individual neurons. However, assuming that the neurons we tested accomplished the same kind of processing but were statistically independent, the overall likelihood in favor of the probability-mixing and the response-averaging model, respectively, equals multiplication of the likelihoods of all of the individual neuron, or equivalently, summation of log-likelihood values, corresponding to summation of AIC (and approximately summation of BIC) values. We therefore also obtained overall AIC (BIC) values for the two models from the overall likelihoods, the numbers of parameters summed across all neurons, and the sample sizes of the data.

In addition to AIC and BIC criteria, we use the root mean squared deviation (RMSD) between observed and predicted firing rates and uniformity tests for general goodness of fit.

Empirical and theoretical firing rates can be compared to judge the goodness of fit. A quantitative measure is the RMSD between empirical and predicted rates for all spike trains of a neuron:

$$\text{RMSD} = \sqrt{\frac{1}{K}\sum_{i=1}^{K}(r_i - \hat{r}_i)^2}, \tag{15}$$

where K is the total number of spike trains. The empirical rate, r, is given by $r = N/T$, where N is the number of spikes, and T is the total time of the spike train. The theoretical rate, $\hat{r}$, was estimated by

$$\hat{r} = \frac{1}{T}\int_0^T \lambda(t|H_t, \hat{\theta})dt. \tag{16}$$

In the probability-mixing model, stimulus decoding is first applied. Stimulus decoding in a mixture model is finding which stimulus, l^*, the neuron is most probably responding to given a spike train and the estimated parameters. This is a classification problem, and solved by the stimulus that maximizes the posterior probability of l given the spike train τ and estimated parameters $\hat{\theta}$: $l^* = \text{argmax}_l P(l|\tau, \hat{\theta})$. Thus, in Equation (16) the classified stimulus is used.

2.7. Model Control by Uniform Residuals

2.7.1. Uniformity Test

A common method is to apply the time rescaling theorem (Brown et al., 2002; Haslinger et al., 2010). For a spike train τ, the transformations

$$Z_i = \int_{t_i}^{t_{i+1}} \lambda(s|H_s)ds \tag{17}$$

for $i = 1, 2, \ldots, N-1$ are exponentially distributed with rate parameter 1, and thus,

$$Z = \int_0^T \lambda(s|H_s)ds \tag{18}$$

is the total time of a Poisson process with rate parameter 1 having N events. The above is true if and only if $\lambda(s|H_s)$ represents the true conditional intensity function. This provides uniformity tests both on interspike interval level: $F_{exp}(Z_i|1) \sim U(0,1)$, where $F_{exp}(Z_i|1)$ is the exponential distribution function with rate 1, and on spike count level: $F_{pois}(N|Z) \sim U(0,1)$, where $F_{pois}(N|Z)$ is the Poisson distribution function with parameter Z. In the latter case, the discrete distribution is approximated by the uniform distribution by taking the average value of $F_{pois}(N|Z)$ and $F_{pois}(N-1|Z)$.

Intuitively, if and only if the model correctly describes the observed neuronal behavior, providing the correct spiking probability at each discretized time step Δt, the transformation Equation (18) is distributed as a standard Poisson process. We verify the similarity between the transformation and the standard Poisson process, by checking the uniform residuals calculated on the two levels described above against a uniform distribution, by Quantile-Quantile (QQ) plots and histograms. If QQ-plots fall close to the indentity line, it indicates that the model describes the true neuronal behavior well, as well as if histograms are standard uniform, i.e., it has approximately equal number of residuals within each bin in the interval (0, 1).

2.8. Unimodality Tests

The response-averaging model predicts a unimodal distribution of firing rates, whereas the probability-mixing model predicts a multimodal distribution when the neuron is exposed to bidirectional stimuli and firing rates to unidirectional stimuli are different. The unimodality test is a statistical test for unimodality of an empirical distribution, i.e., whether the distribution shows a single mode or multiple modes. The dip test (Hartigan and Hartigan, 1985) is one method to perform the unimodality test. A significant p-value of a dip test rejects the hypothesis that there is a single mode and indicates multiple modes in the empirical distribution. Thus, we can perform the dip test as an empirical measure for the probability-mixing or the response-averaging model. We tried to employ dip tests to test for unimodality of a distribution on the firing rates, but the data are too sparse to provide useful information. One particular obstacle is that when estimating empirical firing rates (by spike counts) on discretized intervals, if these intervals are too narrow, only a few spikes or none will be present in most intervals. Then the empirical firing rates only take a few distinct values, repeated many times, and the test always turns out positive since the rates seem to follow a discrete distribution. If intervals are not narrow, there will only be a few data points, not enough for a test. Instead, as an auxiliary measure, we tested unimodality of the distribution of interspike intervals (ISIs). There is no reason to expect the ISI distribution to be unimodal, even if the distribution of firing rates is, since memory effects may create complex behavior in the distribution of ISIs. However, if a particular neuron does not show a multimodal ISI distribution while being exposed to a unidirectional stimulus, but the distribution changes to multimodal when bidirectional stimuli are presented, there is some indication that this multimodality could be caused by the bidirectional stimuli, supporting the probability mixing model.

3. RESULTS

Our basic observations were sequences of action potentials (spike trains) emitted by individual MT neurons in the different conditions of the experiment in response to visual movement in different directions. Models were fitted to the spike train data by maximum likelihood estimation using numerical optimization algorithms. A global optimization with the dividing rectangles algorithm (Jones et al., 1993) was first performed, and the resulting estimates were then used as initial values for a local optimization with the Nelder-Mead simplex algorithm (Nelder and Mead, 1965), providing the final estimates. All parameters were estimated simultaneously.

3.1. Results from Model Selection by Relative Goodness of Fit and Cross-Validation

To select one of the two models, we calculated the RMSD, AIC, and BIC values. The lower plots in **Figure 4A** shows, for each individual neuron, the difference between the AIC (BIC) value given the best-fitting probability-mixing model and the AIC (BIC) value given the best-fitting response-averaging model, with the color indicating neurons with many observed spikes (more than 2400 spikes, cyan) or few observed spikes (<2400 spikes, magenta; the spike counts include all spikes from the given neuron inside the observation windows in the four experimental conditions). Diagnostic neurons are indicated with dots, non-diagnostic neurons are indicated with crosses. Values below 0 favor the probability-mixing model, values above 0 favor the response-averaging model. The difference in AIC (BIC) values is plotted against the sum of the negative log-likelihood values from the two models normalized by number of spikes, such that data points to the left are more trustworthy (approximately coinciding with those with larger sample sizes). Two dotted lines are drawn at ± 10, representing the difference value of 10. This is the value suggested in Burnham and Anderson (2002) as the critical value for the less plausible model to have essentially no support in the data compared with the better model. A few neurons (depicted near the bottom of the plot) seemed highly diagnostic in distinguishing between the response-averaging and the probability-mixing model. Many other neurons failed to distinguish between the two models (neurons with values near zero). This could be due to limited sample sizes, since the cyan neurons are more trustworthy with larger sample sizes, and indeed tend to fall below 0. Furthermore, as expected, the non-diagnostic neurons typically have values around 0.

The values resulting from analyzing all neurons together are shown as AIC^0 (BIC^0) in the table at the top of **Figure 4A**. These values can be interpreted as the explanatory evidence in the models compared to the null model (Harrell, 2001), see Section 2.5 for definition of the null model. Furthermore, the differences between the two AIC (BIC) values, ΔAIC (ΔBIC), are indicated in the same table, both for all neurons, and for diagnostic neurons only. The overall AIC and BIC values, aggregating all the information from individual neurons, are much smaller for the probability-mixing model than for the response-averaging model, so both the AIC and the BIC strongly favor the probability-mixing model. Indeed, both (absolute) differences are greater than $\Delta = 1000$. Thus, given the two models, according to both the AIC and the BIC criteria, the weight in favor of the probability-mixing model is $1/(1 + \exp(-\Delta/2)) \approx 1$, and the weight in favor of the response-averaging model is $\exp(-\Delta/2)/(1 + \exp(-\Delta/2)) \approx 0$, see Section 2.6.

The upper plot in **Figure 4A** shows the difference between the RMSD between observed and predicted firing rates for the best-fitting probability-mixing model and the RMSD for the best-fitting response-averaging model. The RMSD values were calculated using 10-fold cross-validation on spike trains of each neuron. For most of the neurons, the RMSD for the best-fitting probability-mixing model was smaller than the RMSD for the best-fitting response-averaging model, and this is particularly obvious for more trustworthy neurons, and for diagnostic neurons. The RMSD for all data for both models are shown in the top table. As the AIC and the BIC, the RMSD criterion also favors the probability-mixing model. The RMSD results are more consistently in favor of the probability-mixing model for all diagnostic neurons compared with the AIC and BIC results. Note the different perspectives of these model selection methods: RMSD measures the predictive accuracy while AIC (BIC) measures the information loss of the proposed model from the truth. We conclude that the probability-mixing model predicts behavior of independent trials better or at least as well as the response-averaging model on all neurons.

The overall conclusion is that the analysis supports the probability-mixing over the response-averaging model.

3.2. Results from Model Control by Uniform Residuals

The computations of AIC and BIC values show that the probability-mixing model fits the data better than does the response-averaging model, but neither information criterion tells us the absolute (as distinct from the relative) goodness of fit. For either model, goodness of fit to the spike trains of the neurons was evaluated by uniformity tests, both on interspike level and on spike count level (see Section 2). We merged all results based on Equation (17) from all spike trains of all neurons, to obtain uniform residuals on the interspike interval level, and all results based on Equation (18) to obtain uniform residuals on the spike count level. The uniform residuals were checked graphically in **Figure 4B** by histograms and QQ plots against the standard Uniform distribution. The histograms and plots of events at the interspike interval level show nearly the same goodness of fit for the probability-mixing model and the response-averaging model, but the histograms and plots of events at the spike count level show better fits for the probability-mixing model compared with the response-averaging model, as can be seen from the cyan QQ-plot being closer to the identity line, and cyan histograms being more uniform than the magenta ones in the lower plots. However, neither model is perfect. If a model is correct we expect the uniform residuals to lie on the identity line, which is not strictly the case for either one of the two models. We conjecture that this was partly caused by boundary effects inducing bias because the observation intervals were never longer than 500 ms. To check this, we conducted a simulation study, first simulating from both models using the estimated parameters, with observation intervals of both 500 and 2000 ms, and then estimating with both models (**Figure 5**). The interval of 2000 ms was chosen to be large enough for boundary effects to be negligible. The results suggested that the misfits could be explained, in part, by finite sample effects. Another feature not accounted for in the model is overdispersion, i.e., that the data show a larger variance than predicted by the model. This

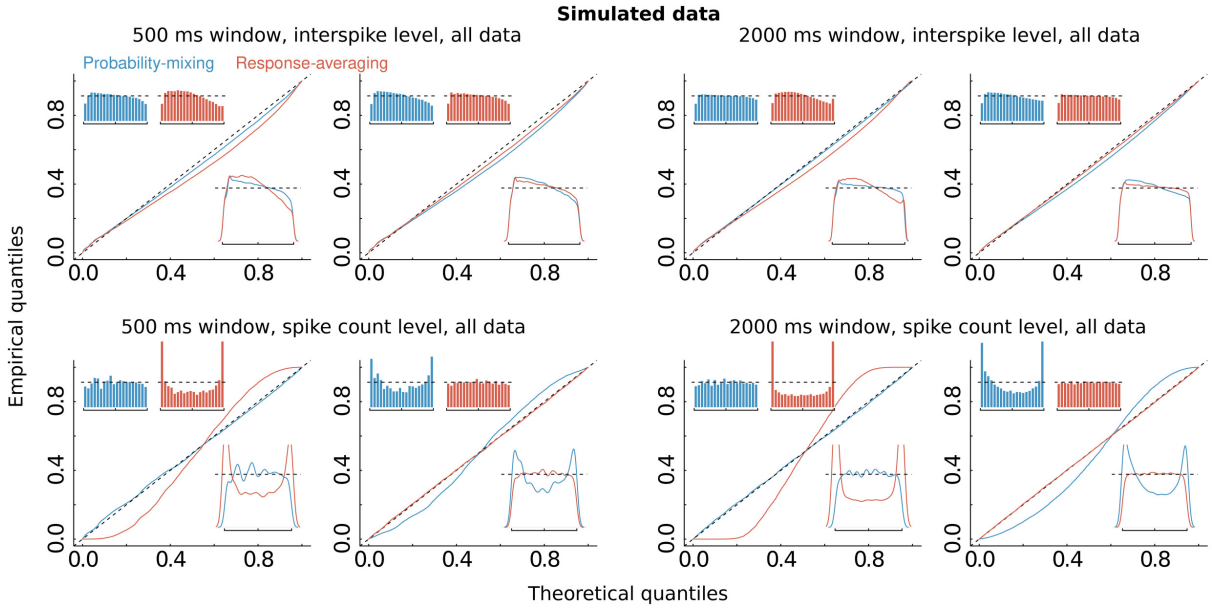


FIGURE 5 | Simulations. The same as **Figure 4B**, but on simulated data, using the estimated parameters, and simulating from either of the two models, with observation windows of 500 ms (left panel with four figures) and of 2000 ms (right panel with the other four figures), respectively. Upper and lower panels show results at interspike and spike count levels, respectively. Panels in columns 1 and 3 show results for the data simulated from the probability-mixing model, panels in columns 2 and 4 show results for the data simulated from the response-averaging model.

occurs for example if parameter values fluctuate from trial to trial, whereas the model assumes these constant. We therefore also plotted the uniform residuals using only the bidirectional data (**Figure 4C**), and the fit clearly improved, suggesting overdispersion.

In the analysis it is implicitly assumed that under the probability-mixing model, the represented object does not change during the course of a trial of 500 ms. This is done to obtain more statistical validity, but might be questionable from a biological point of view. For example, Fiebelkorn et al. (2013) found that sustained attention naturally fluctuates with a periodicity of 4–8 Hz, with reweighting between different objects occuring at 4 Hz. To check the validity of using the full length of the 500 ms interval, we also tried splitting the data, reanalyzing separately on the first (0–250ms) and on the second (250–500 ms) halves. The analysis was conducted the same way as for the full 500 ms interval. The results on RMSD, AIC, and BIC are shown in **Figure 6** for the first half (**Figure 6A**) and the second half (**Figure 6B**). There are only small and not relevant differences between the two halves for each criterion. At both halves, the RMSD favors the probability-mixing model, particularly for neurons with a large number of observations. The AIC and BIC also show similar distribution patterns between the two halves. A paired Wilcoxon signed-rank test was done for the differences ΔAIC at the first half against the second half with the null hypothesis being that ΔAIC does not change between halves. The obtained p-value is 0.858, implying no evidence of changes in ΔAIC. The test on ΔBIC gives $p = 0.830$,

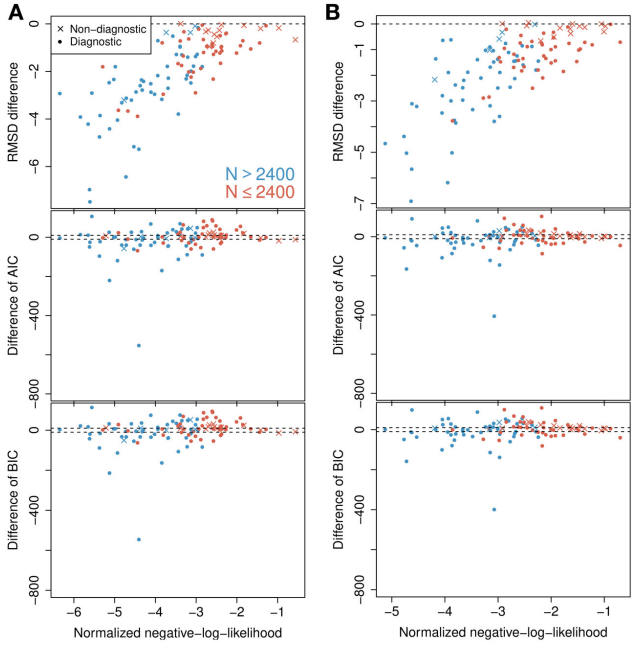


FIGURE 6 | Model selection on first and second halves of the observed time intervals. Differences in BIC, AIC, and RMSD values between the probability-mixing model and the response-averaging model (the former minus the latter). **(A)** Analysis on first 250 ms of the observed time intervals (first half of the data). **(B)** Analysis on next 250 ms of the observed time intervals (second half of the data). Compare to **Figure 4A** for analysis of the full interval.

leading to the same conclusion. To summarize, the conclusions are essentially the same as for the full interval, and model fitting on the shorter intervals provide no extra information. Thus, we analyze the full 500 ms interval exploiting the entire data.

3.3. Results for Unimodality Tests

In **Figure 7**, dip tests of unimodality of the ISI distribution are illustrated for each neuron in each of the 12 direction-of-motion stimulus pairs. Each lattice point in the mesh figure represents one test, and blue lattice points (upper panels) show results that are statistically significant ($p < 0.05$) against unimodality (i.e., indicating at least two modes). In the upper left panel, data from the unidirectional stimulus conditions *fix1* and *fix2* are combined for the 84 neurons tested in these conditions. They were combined after normalizing by multiplying the ISIs by the average firing rate of the corresponding neuron and condition, so that the average firing rate of any neuron in any condition was 1. This was done in order not to observe an artificial bimodal distribution, caused by different response properties in aperture 1 and 2. Similar results are obtained by splitting in the two aperture conditions without normalization (results not shown). The bidirectional stimulus conditions were not normalized. Of the 1008 tests on unidirectional stimulus data, 6.05% were positive. This is close to the expected 5% from the coverage properties of the test, so it appears that under unidirectional stimulus, the ISI distributions are not multimodal. In the two upper panels to the right, data from the bidirectional stimulus *attend-fix* and *attend-in* are shown for the same 84 neurons (below the black line) and for the remaining 25 neurons (above the black line), which were not tested during unidirectional stimulus. Of the 1008 tests on bidirectional stimulus data from those neurons that were also tested in the unidirectional stimulus conditions, 6.8% (*attend-fix*) and 14.3% (*attend-in*) were positive. Including also the 25 neurons only tested in the bidirectional stimulus conditions, these numbers were 10.6 and 19.6% out of 1308 tests, respectively. Note that fewer significant lattice points appear in the *attend-fix* condition than in the *attend-in* condition, which is probably due to smaller sample sizes; see **Table 2**. The yellow lattice points are those corresponding to condition 5, where the stimulus in aperture 1 is 120° from the preferred direction, and the stimulus in aperture 2 is −120° from the preferred direction. This is the only condition where on average the firing rates for the two stimuli are equal, see the green and orange tuning curves on **Figure 2**, and thus, no bimodality is expected for most of the neurons in this condition. Indeed, in this case only 7.3 and 9.2% were significant. Condition 11, where the stimulus directions are ±60°, could also be expected to have equal firing rates for the two stimuli, and thus no multimodality, but since the firing rates are higher here, small differences in tuning curves for the two apertures result in large differences in firing rates, and thus, multimodality can still occur. In all three upper panels, most *p*-values are non-significant. However, compared with unidirectional stimulus conditions, more significant *p*-values appear in bidirectional stimulus conditions, mainly in condition *attend-in*, suggesting that stimulus plurality caused multimodality. This is illustrated in the lower panels, where changes from either significant to non-significant (red, 2.6% for *attend-fix*, 1.4% for *attend-in*) or from non-significant to significant (green, 3.4% for *attend-fix*, 9.6% for *attend-in*) *p*-values are indicated.

3.4. Population Behavior of Probability-Mixing

In the probability-mixing model, a neuron attends to only one of a plurality of stimuli. A natural question is then whether in any given trial, individual neurons within a critical population behave consistently or independently. We therefore investigated correlations of nearby neurons. In the data, at most two neurons were recorded simultaneously, and there are 25 such neuron pairs. The two neurons do not necessarily have the same preferred direction of motion, but they differ at most 60° in their preferred direction. If neurons act consistently, we expect higher correlations for those pairs with the same preferred direction. We calculated the correlation of the firing rates of each neuron pair at different RDP-motion stimulus pairs using Spearman's correlation coefficient and Spearman's correlation test. Conditions *attend-in* and *attend-fix* are combined to make the sample size larger. The idea is that if two neurons have highly correlated attended stimuli, the correlation coefficient of rates will be large; otherwise, the coefficient will be near 0: Let two vectors $X = (X_1, X_2, \ldots, X_n)$ and $Y = (Y_1, Y_2, \ldots, Y_n)$ denote the firing rates of two neurons from n trials at a given stimulus pair. The corresponding X_i and Y_i are the firing rates of two neurons in the same trial i. Since there are two stimuli, X_i and Y_i could represent either stimulus. If the firing rates of the two neurons are positively correlated, then X_i and Y_i likely represent the same stimulus. If the firing rates are negatively correlated, then X_i and Y_i likely represent opposite stimuli. In both situations, non-zero correlation between X and Y is expected, assuming two stimuli generate sufficiently different firing rates. On the other hand, if the attended stimuli are not correlated, nor will the firing rates X and Y be correlated.

The top left panel in **Figure 8** shows the heat map of Spearman's correlation coefficients, and the top right panel shows the stronger positive correlations in red and stronger negative correlations in blue. The bottom left shows *p*-values from Spearman's correlation test for correlation being 0. The bottom right shows significant *p*-values in blue. The ratio of significant *p*-values over all 12 × 25 cells is 11.7%.

Most correlations are weak. However, we find a few stronger correlations in some neuron pairs, with a slight trend toward higher positive correlations for those with the same preferred direction, and negative correlations for those with differing preferred directions. The conclusions have to be interpreted with caution, though, since data on simultaneously recorded neurons are scarce.

3.5. Parameter Estimates from Maximum Likelihood

Parameter estimates, see **Table 2** for a summary of model parameters, are illustrated in **Figure 9** comparing the probability-mixing model with the response-averaging model and comparing

aperture 1 with aperture 2. The upper panels, **Figures 9A,B**, provide the sum of A (directional gain) and r_0 (firing rate without stimulus) from the Gaussian tuning curve, i.e., the maximal firing rate. The estimates from the probability-mixing model tend to be smaller than the estimates from the response-averaging model. **Figures 9C,D** cover only the probability-mixing model, because the weights and attentional scaling parameters are not identifiable in the response-averaging model. In **Figure 9C** the probabilities of responding to aperture 1 in the *attend-fix* condition ($p_{attend-fix}$) are plotted against the probabilities of responding to aperture 1 in the *attend-in* condition ($p_{attend-in}$). As expected, the probability of responding to aperture 1 is increased when attention is directed toward it, i.e., $p_{attend-in}$ tends to be larger than $p_{attend-fix}$ and also larger than 0.5. In **Figure 9D** attentional effects for aperture 2 (a_2) are plotted against effects for aperture 1 (a_1) in the *attend-in* condition. The effect of the cue is clearly detected: a_1 tends to be larger than a_2, and also larger than 1, i.e., attention increases the firing rate. In **Figure 9E** the identifiable parameters b_1 and b_2 in the response-averaging model are plotted against the corresponding values calculated from estimates in the probability-mixing model. Again, aperture 1 (b_1) yields larger values than aperture 2 (b_2), which is expected because of the cue. In **Figure 9F** the 10 spike response weight parameter estimates from the conditional intensity function are plotted. We use median values and quantiles. The first value γ_1 is much more negative than the others, implying that a spike suppresses a spike in the next instance, corresponding to the refractory period. The spike response weight values decay to zero, illustrating the length of the memory of the spike history.

4. DISCUSSION

Responses of sensory neurons to multiple presentations of identical stimuli can be highly variable ("cortical variability"; Goris et al., 2014; Cui et al., 2016). In this article we focus on one possible source of such cortical variability, namely, variation in which stimulus a sensory neuron responds to at a given time in a certain trial. Specifically, we aimed to determine if neurons in extrastriate visual cortex encode the presence of more than one distinct stimulus in their receptive field by alternating between response states, each predominantly representing one of the stimuli in the receptive field (Bundesen et al., 2005). We found evidence in support of such a multiplexing behavior by analyzing spike trains of individual trials (rather than average responses across trials) from neurons in visual cortical area MT of rhesus

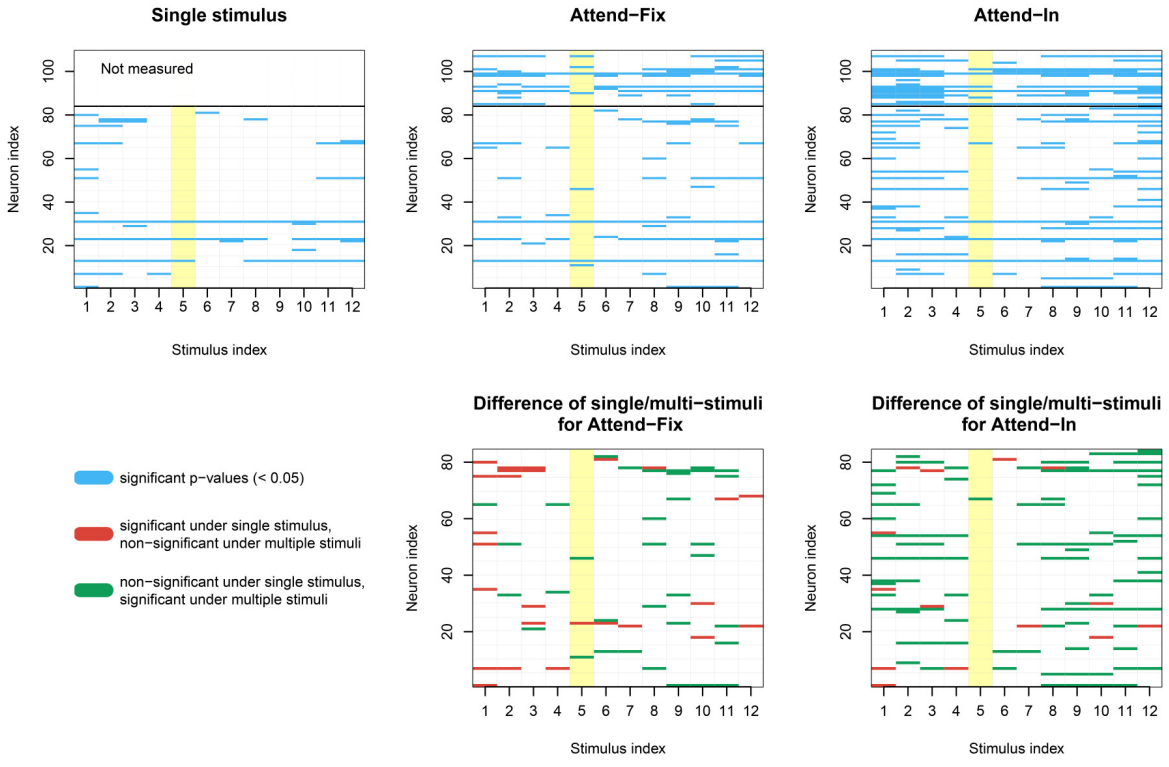


FIGURE 7 | Results of dip tests. (Upper panels) Dip tests of each neuron at each condition for single stimulus trials (left panel) and stimulus mixture trials (middle and right panels) are illustrated. Each lattice point represents one test, and blue lattices are statistically significant ($p < 0.05$) against unimodality for the corresponding neuron and condition. The yellow lattice points are those corresponding to condition 5, where the stimulus in aperture 1 is 120° from the preferred direction, and the stimulus in aperture 2 is −120° from the preferred direction. This is the only condition where on average the firing rates for the two stimuli are equal, see the green and orange tuning curves on **Figure 2**, and thus, no bimodality is expected. **(Lower panels)** Changes from either significant to non-significant (red) or from non-significant to significant (green) p-values.

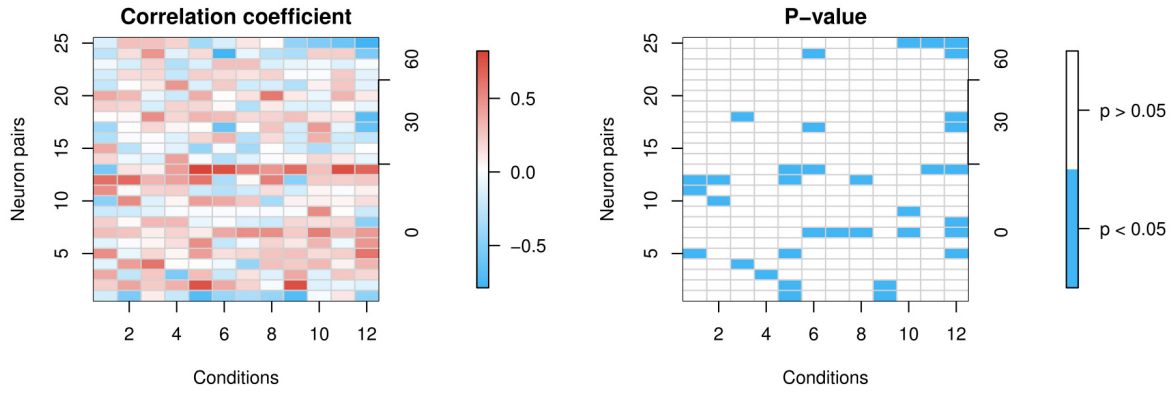


FIGURE 8 | Correlation of firing rates between neuron pairs. The *x*-axes represent the 12 conditions and the left *y*-axes represent the 25 neuron pairs that are simultaneously recorded. Each lattice point in the mesh corresponds to one neuron pair at one condition. The 25 neuron pairs are ordered by the difference in degrees between the pair's preferred directions (0, 30, or 60°) shown in the right *y*-axes. If they differ in their preferred direction, then when one of the neurons is presented with its preferred direction, the other is not, and vice versa. So we expect less correlation in that case, whereas if they share the same preferred direction, there is more reason to believe they might be correlated. **(Left panel)** shows correlation coefficients. **(Right panel)** shows in blue significant *p*-values at a 5% level for the two-sided test of zero correlation.

monkeys. Our approach is based on recent advances in statistics (chap. 19, Kass et al., 2014) that allowed us to distinguish responses from trial to trial. Employing statistical model selection using AIC, BIC, and RMSD, and model control using time rescaling and uniformity tests we find support for probability-mixing, i.e., serial switches between response states, distinct from the response-averaging suggested by pooling responses across multiple trials. Unimodality tests provide further support for multiplexing behavior by showing that stimulus plurality increases the probability of statistically significant multimodality of the interspike interval distribution.

For decades responses of sensory neurons in primate visual cortex have been investigated with single stimuli and their parametric variation. This has resulted in a very detailed understanding of the input-output-relationship of neurons in well-studied areas like primary visual cortex V1, area V4 along the temporal processing pathway and, most relevant for the current study, the middle-temporal area MT in the dorsal pathway.

More recently, particularly in MT, studies have focused on neuronal responses when multiple moving stimuli are present (spatially separated or in spatially coincident motion as transparent random dot patterns or sine wave gratings) in a given receptive field. Such studies have investigated "sensory" conditions, i.e., when none of the stimuli were behaviorally relevant (Snowden et al., 1991; Recanzone et al., 1997; Britten and Heuer, 1999; Treue et al., 2000; Majaj et al., 2007), as well as "attentional" conditions, i.e., task designs where one of the stimuli were behaviorally relevant (Seidemann and Newsome, 1999; Treue and Trujillo, 1999; Patzwahl and Treue, 2009; Niebergall et al., 2011a,b; Ni et al., 2012). All of these studies implicitly or explicitly assume that neurons always respond to multiple stimuli in their receptive field with a single response state that represents an integration (averaging with or without scaling or gain control) of the individual stimulus responses.

Here we successfully challenge this assumption by providing evidence for the ability of neurons to maintain distinct representations of the stimuli inside a given receptive field.

This ability to encode multiple stimuli by separate response states of individual neurons endows the visual system with a powerful feature, not present if the neurons combine the multiple stimulus responses into a common response. Indeed, once the responses have been averaged over all stimuli, reconstructing single stimuli from average responses at later stages of processing seems difficult if not impossible (Orhan and Ma, 2015). This is a core issue in understanding cortical representations of complex scenes, since they often have multiple stimuli placed in the same receptive field, particularly in the large receptive fields common in higher extrastriate cortical areas. If such neurons would integrate all stimuli inside their receptive field such "stimulus mixing" would severely compromise the brain's ability to maintain spatially detailed representations in natural vision (Orhan and Ma, 2015). The multiplexing we observe instead allows the information about which stimulus caused a particular neuronal response to be preserved and maintained across a series of processing stages from primary visual cortex through areas in extrastriate cortex.

Beyond this benefit, the temporal multiplexing of information provides a unique opportunity to selectively modulate the individual representations of the various stimuli contributing to a neuron's response. Such a reweighing has been suggested by models of attention since the perceptual effect of visual attention can often be described as an increase in the perceptual strength of attended stimuli at the expense of the perceptual strength of unattended stimuli.

One of these attention models, the Neural Theory of Visual Attention (NTVA; Bundesen et al., 2005), a neural interpretation of the mathematical Theory of Visual Attention (TVA; Bundesen, 1990), explicitly proposes that a neuron, when presented with a plurality of stimuli in its RF, responds to only one of them

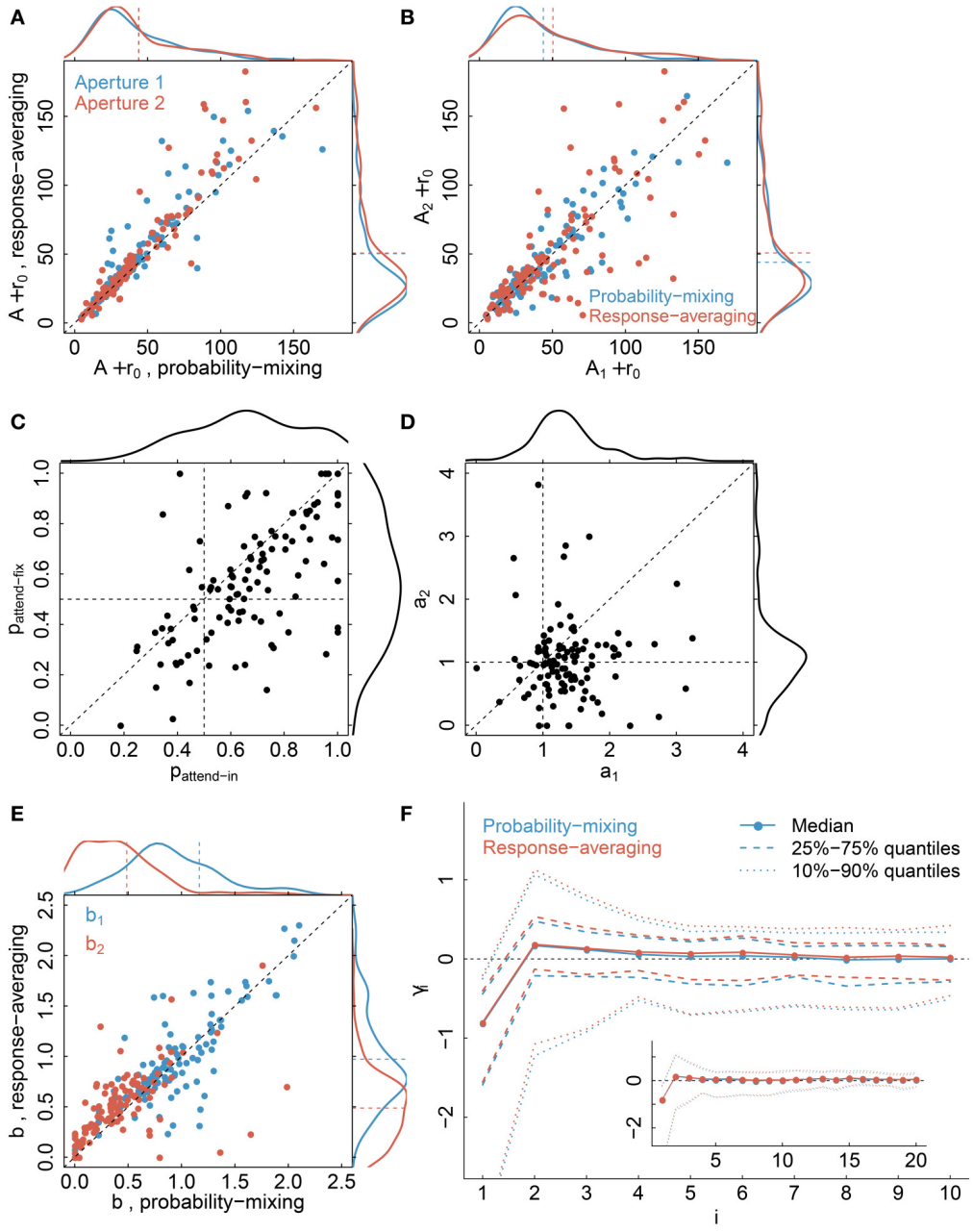


FIGURE 9 | Parameter estimates. The plots compare the probability-mixing model with the response-averaging model and aperture 1 with aperture 2. **(A)** The sum of A and r_0 from the response-averaging model is plotted against the probability-mixing model, using cyan for aperture 1 and magenta for aperture 2. The data densities are plotted on the top and on the right side, with dashed lines indicating the means. **(B)** We use the same estimates as in **(A)**, but plot aperture 2 against aperture 1, and use cyan for the probability-mixing model and magenta for the response-averaging model. **(C,D)** are only for the probability-mixing model, since these parameters are not all identifiable in the response-averaging model. **(C)** The probabilities of responding to aperture 1 in the *attend-fix* condition ($p_{attend-fix}$) are plotted against *attend-in* condition ($p_{attend-in}$). **(D)** Attentional effects for aperture 2 (a_2) are plotted against aperture 1 (a_1). **(E)** The identified parameters b_1 and b_2 in the response-averaging model are plotted against the corresponding values calculated from estimates in the probability-mixing model. **(F)** The medians of 10 spike response weight parameters from the conditional intensity function are plotted, together with the central 50 and 80% of the empirical distributions. We also fitted models with a memory of 20 ms, and the resulting estimates are plotted in the insert figure.

at a time. This hypothesis has not been tested before but was suggested by Bundesen et al. (2005) for computational and biological reasons (survival value), and it fits in with the way in which attentional modulations of sensory processing (in particular, so-called "filtering") are explained in NTVA. In TVA stimulus representations race (compete) to become encoded into visual short-term memory (VSTM) before it is filled up. This race is influenced (biased) by attentional weights

and perceptual biases, so that certain objects and features have higher probabilities of being perceived (encoded into VSTM). Thus the TVA presaged what later became known as the biased competition model of attention (Desimone and Duncan, 1995; Reynolds et al., 2000). Our data suggest that biased competition accounts of attentional responses need to be extended to allow for an alternation between response states rather than a single response state representing the outcome of the biased competition between the different stimulus representations.

The TVA is also compatible with the feature similarity gain model (Treue and Trujillo, 1999; Martinez-Trujillo and Treue, 2004). This model proposes that attention modulates brain activity by multiplicatively scaling neuronal responses with gain factors. The magnitude of a given gain factor represents the similarity between the stimulus preferences of the neuron and the currently attended features. In this model a selective enhancement or suppression of individual stimuli (based either on the stimulus' spatial location or its features Xue et al., 2016) is achieved on the population level because attention to a given feature increases the responses of all neurons preferring the same or similar features. In the TVA, the gain factor in question is the multiplicative perceptual bias toward feature i (β_i), which is applied to neurons that are coding feature i. Incorporating the observed multiplexing into the feature similarity gain model would further elaborate the approach of the model to selective enhancement of attended features and locations.

Our observation that neuronal responses alternate between response states is reminiscent of the hypothesis that stimulus sampling under continuous attentional allocation follows a periodic process (Busch and VanRullen, 2010). While this potential link is intriguing, our data did not allow us to test the duration of individual response states to see whether they match the 7 Hz oscillations observed in the Busch and VanRullen study. On the other hand, the analysis of our small set of recordings from neuronal pairs suggests that neurons that share sensory preferences (with respect to motion direction in our case) tend to encode the same of two stimuli at a given time while neurons with different preferences tend to anti-correlate in their response states. This supports the hypothesis that the whole population of neurons responding to a given stimulus configuration tends to alternate their individual response states in a coordinated fashion.

The serial multiplexing we observe also allows us to account for other observations when multiple stimuli are combined within the same receptive field. This is most apparent for the case in which two RDPs moving in different directions are spatially superimposed, creating the percept of two surfaces sliding across each other. As documented in Treue et al. (2000), combining two directions with an angular separation of 30–60° creates a stimulus in which the two component motions are easily distinguishable perceptually, but causes a neural population response (averaged across trials) that is single-lobed, suggestive of a single direction in the receptive field. While the perception of two directions under such conditions can be explained by assuming a particular decoding mechanism, our observed multiplexing of the individual stimulus representations provides other types of explanation for the apparent discrepancy between neural responses and perception. Additionally, the distinct encoding of the two motion surfaces through separate response states might also allow the visual system to separately manipulate the individual stimulus representations as apparent in the perceptual (Marshak and Sekuler, 1979) and physiological (Helmer et al., 2016) repulsion of the perceived angular separation in such transparent motion patterns.

In summary, this study suggests and documents a neuronal coding scheme that temporally multiplexes information from multiple stimuli within the receptive fields of neurons in extrastriate visual cortex. This allows nervous systems to enjoy the benefits of large receptive fields (spatial integration of information to achieve more complex selectivities) without suffering from the disadvantage that large receptive fields pool the responses to multiple stimuli and thus lose critical information about their individual contribution to the cell's overall response. Such a system could also reconcile the observation of perceptual separability of multiple stimuli (such as surfaces in transparent motion) with the apparent pooling of information within the spatial extent of receptive fields in extrastriate visual cortex.

AUTHOR CONTRIBUTIONS

KL, SK, SD, and CB conceptualized the research. VK and ST designed and performed experiments. KL and SD designed the statistical methodology. KL performed the analysis and prepared figures. All authors interpreted the results. KL, ST, SD, and CB wrote the paper. All authors approved the final version of the paper.

FUNDING

The work was part of the Dynamical Systems Interdisciplinary Network, University of Copenhagen. The work of VK and ST was supported by the Volkswagen Foundation (grant I/79868), the Bernstein Center of Computational Neuroscience Göttingen (grants 01GQ0433 and 01GQ1005C) of the BMBF and the German Research Foundation (DFG) Research Unit 1847 "The Physiology of Distributed Computing Underlying Higher Brain Functions in Non-Human Primates".

ACKNOWLEDGMENTS

We thank Robert Kass, John Duncan and Jeffrey Schall for valuable comments and suggestions during the preparation of the manuscript.

REFERENCES

Britten, K. H., and Heuer, H. W. (1999). Spatial summation in the receptive fields of MT neurons. *J. Neurosc.* 19, 5074–5084.

Brown, E. N., Barbieri, R., Ventura, V., Kass, R. E., and Frank, L. M. (2002). The time-rescaling theorem and its application to neural spike train data analysis. *Neural Comput.* 14, 325–346. doi: 10.1162/08997660252741149

Bundesen, C. (1990). A theory of visual attention. *Psychol. Rev.* 97:523.

Bundesen, C., and Habekost, T. (2008). *Principles of Visual Attention: Linking Mind and Brain*. Oxford: Oxford University Press.

Bundesen, C., Habekost, T., and Kyllingsbæk, S. (2005). A neural theory of visual attention: bridging cognition and neurophysiology. *Psychol. Rev.* 112:291. doi: 10.1037/0033-295X.112.2.291

Burnham, K. P., and Anderson, D. R. (2002). *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach*. New York, NY: Springer.

Busch, N. A., and VanRullen, R. (2010). Spontaneous eeg oscillations reveal periodic sampling of visual attention. *Proc. Natl. Acad. Sci. U.S.A.* 107, 16048–16053. doi: 10.1073/pnas.1004801107

Busse, L., Wade, A. R., and Carandini, M. (2009). Representation of concurrent stimuli by population activity in visual cortex. *Neuron* 64, 931–942. doi: 10.1016/j.neuron.2009.11.004

Calapai, A., Berger, M., Niessing, M., Heisig, K., Brockhausen, R., Treue, S., et al. (2016). A cage-based training, cognitive testing and enrichment system optimized for rhesus macaques in neuroscience research. *Behav. Res. Methods*. doi: 10.3758/s13428-016-0707-3. [Epub ahead of print]. Available online at: http://link.springer.com/article/10.3758/s13428-016-0707-3

Claeskens, G., and Hjort, N. L. (2008). *Model Selection and Model Averaging*, *Vol. 330*. Cambridge: Cambridge University Press.

Cox, D. R., and Lewis, P. A. (1966). *The Statistical Analysis of Series of Events*. London: Chapman and Hall.

Cui, Y., Liu, L. D., McFarland, J. M., Pack, C. C., and Butts, D. A. (2016). Inferring cortical variability from local field potentials. *J. Neurosci.* 36, 4121–4135. doi: 10.1523/JNEUROSCI.2502-15.2016

Daley, D. J., and Vere-Jones, D. (1988). *An Introduction to the Theory of Point Processes*, *Vol. 2*. New York, NY: Springer.

Desimone, R., and Duncan, J. (1995). Neural mechanisms of selective visual attention. *Annu. Rev. Neurosci.* 18, 193–222.

Fiebelkorn, I. C., Saalmann, Y. B., and Kastner, S. (2013). Rhythmic sampling within and between objects despite sustained attention at a cued location. *Curr. Biol.* 23, 2553 – 2558. doi: 10.1016/j.cub.2013.10.063

Gattass, R., Nascimento-Silva, S., Soares, J. G., Lima, B., Jansen, A. K., Diogo, A. C., et al. (2005). Cortical visual areas in monkeys: location, topography, connections, columns, plasticity and cortical dynamics. *Philos. Transact. R. Soc. Lond. B* 360, 709–731. doi: 10.1098/rstb.2005.1629

Gilmore, R. O., Hou, C., Pettet, M. W., and Norcia, A. M. (2007). Development of cortical responses to optic flow. *Vis. Neurosci.* 24, 845–856. doi: 10.1017/S0952523807070769

Goris, R. L., Movshon, J. A., and Simoncelli, E. P. (2014). Partitioning neuronal variability. *Nat. Neurosci.* 17, 858–865. doi: 10.1038/nn.3711

Harrell, F. E. (2001). *Regression Modeling Strategies*. New York, NY: Springer Science & Business Media.

Hartigan, J. A., and Hartigan, P. M. (1985). The dip test of unimodality. *Ann. Stat.* 13, 70–84.

Haslinger, R., Pipa, G., and Brown, E. (2010). Discrete time rescaling theorem: determining goodness of fit for discrete time statistical models of neural spiking. *Neural Comput.* 22, 2477–2506. doi: 10.1162/NECO_a_00015

Helmer, M., Kozyrev, V., Stephan, V., Treue, S., Geisel, T., and Battaglia, D. (2016). Model-free estimation of tuning curves and their attentional modulation, based on sparse and noisy data. *PLoS ONE* 11:e146500. doi: 10.1371/journal.pone.0146500

Jones, D. R., Perttunen, C. D., and Stuckman, B. E. (1993). Lipschitzian optimization without the lipschitz constant. *J. Optim. Theory Appl.* 79, 157–181. doi: 10.1007/BF00941892

Kanwisher, N., and Yovel, G. (2006). The fusiform face area: a cortical region specialized for the perception of faces. *Philos. Trans. R. Soc. Lond. B* 361, 2109–2128. doi: 10.1098/rstb.2006.1934

Kass, R. E., Eden, U. T., and Brown, E. N. (2014). *Analysis of Neural Data*. New York, NY: Springer.

Katzner, S., Busse, L., and Treue, S. (2009). Attention to the color of a moving stimulus modulates motion-signal processing in macaque area mt: evidence for a unified attentional system. *Front. Syst. Neurosci.* 3:12. doi: 10.3389/neuro.06.012.2009

Latimer, K. W., Yates, J. L., Meister, M. L., Huk, A. C., and Pillow, J. W. (2015). Single-trial spike trains in parietal cortex reveal discrete steps during decision-making. *Science* 349, 184–187. doi: 10.1126/science.aaa4056

Lee, J., and Maunsell, J. H. (2009). A normalization model of attentional regulation of single unit responses. *PLoS ONE* 4:e4651. doi: 10.1371/journal.pone.0004651

Li, K., Vozyrev, V., Kyllingsbæk, S., Treue, S., Ditlevsen, S., and Bundesen, C. (2016). Data from: Neurons in primate visual cortex alternate between responses to multiple stimuli in their receptive field. *Dryad Digit. Repos.* doi: 10.5061/dryad.88pv1

MacEvoy, S. P., Tucker, T. R., and Fitzpatrick, D. (2009). A precise form of divisive suppression supports population coding in the primary visual cortex. *Nat. Neurosci.* 12, 637–645. doi: 10.1038/nn.2310

Majaj, N. J., Carandini, M., and Movshon, J. A. (2007). Motion integration by neurons in macaque mt is local, not global. *J. Neurosci.* 27, 366–370. doi: 10.1523/JNEUROSCI.3183-06.2007

Marshak, W., and Sekuler, R. (1979). Mutual repulsion between moving visual targets. *Science* 205, 1399–1401.

Martinez-Trujillo, J., and Treue, S. (2004). Feature-based attention increases the selectivity of population responses in primate visual cortex. *Curr. Biol.* 14, 744–751. doi: 10.1016/j.cub.2004.04.028

Martınez-Trujillo, J. C., and Treue, S. (2002). Attentional modulation strength in cortical area mt depends on stimulus contrast. *Neuron* 35, 365–370. doi: 10.1016/S0896-6273(02)00778-X

Nandy, A. S., Sharpee, T. O., Reynolds, J. H., and Mitchell, J. F. (2013). The fine structure of shape tuning in area V4. *Neuron* 78, 1102–1115. doi: 10.1016/j.neuron.2013.04.016

Nelder, J. A., and Mead, R. (1965). A simplex method for function minimization. *Comput. J.* 7, 308–313.

Ni, A. M., Ray, S., and Maunsell, J. H. (2012). Tuned normalization explains the size of attention modulations. *Neuron* 73, 803–813. doi: 10.1016/j.neuron.2012.01.006

Niebergall, R., Khayat, P. S., Treue, S., and Martinez-Trujillo, J. C. (2011a). Expansion of mt neurons excitatory receptive fields during covert attentive tracking. *J. Neurosci.* 31, 15499–15510. doi: 10.1523/JNEUROSCI.2822-11.2011

Niebergall, R., Khayat, P. S., Treue, S., and Martinez-Trujillo, J. C. (2011b). Multifocal attention filters targets from distracters within and beyond primate mt neurons' receptive field boundaries. *Neuron* 72, 1067–1079. doi: 10.1016/j.neuron.2011.10.013

Orhan, A. E., and Ma, W. J. (2015). Neural population coding of multiple stimuli. *J. Neurosci.* 35, 3825–3841. doi: 10.1523/JNEUROSCI.4097-14.2015

Patzwahl, D. R., and Treue, S. (2009). Combining spatial and feature-based attention within the receptive field of mt neurons. *Vis. Res.* 49, 1188–1193. doi: 10.1016/j.visres.2009.04.003

Press, W. H. (2007). *Numerical Recipes: The Art of Scientific Computing, 3rd Edn*. Cambridge: Cambridge University Press.

Recanzone, G. H., Wurtz, R. H., and Schwarz, U. (1997). Responses of MT and MST neurons to one and two moving objects in the receptive field. *J. Neurophysiol.* 78, 2904–2915.

Reynolds, J. H., Chelazzi, L., and Desimone, R. (1999). Competitive mechanisms subserve attention in macaque areas V2 and V4. *J. Neurosci.* 19, 1736–1753.

Reynolds, J. H., and Heeger, D. J. (2009). The normalization model of attention. *Neuron* 61, 168–185. doi: 10.1016/j.neuron.2009.01.002

Reynolds, J. H., Pasternak, T., and Desimone, R. (2000). Attention increases sensitivity of v4 neurons. *Neuron* 26, 703–714. doi: 10.1016/S0896-6273(00)81206-4

Seidemann, E., and Newsome, W. T. (1999). Effect of spatial attention on the responses of area mt neurons. *J. Neurophysiol.* 81, 1783–1794.

Shokhirev, K., Kumar, T., and Glaser, D. (2006). The influence of cortical feature maps on the encoding of the orientation of a short line. *J. Comput. Neurosci.* 20, 285–297. doi: 10.1007/s10827-006-6485-7

Smith, A. T., Singh, K. D., Williams, A. L., and Greenlee, M. W. (2001). Estimating receptive field size from fMRI data in human striate and extrastriate visual cortex. *Cereb. Cortex* 11, 1182–1190. doi: 10.1093/cercor/11.12.1182

Snowden, R. J., Treue, S., Erickson, R. G., and Andersen, R. A. (1991). The response of area mt and v1 neurons to transparent motion. *J. Neurosci.* 11, 2768–2785.

Treue, S., Hol, K., and Rauber, H. J. (2000). Seeing multiple directions of motion-physiology and psychophysics. *Nat. Neurosci.* 3, 270–276. doi: 10.1038/72985

Treue, S., and Trujillo, J. (1999). Feature-based attention influences motion processing gain in macaque visual cortex. *Nature* 399, 575–579.

Treue, S., and Martínez Trujillo J. C. (1999). Feature-based attention influences motion processing gain in macaque visual cortex. *Nature* 399, 575–579.

Truccolo, W., Eden, U. T., Fellows, M. R., Donoghue, J. P., Brown, E. N., et al. (2005). A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects. *J. Neurophysiol.* 93, 1074–1089. doi: 10.1152/jn.00697.2004

Xue, C., Kaping, D., Baloni, R. S., Krishna, B. S., and Treue, S. (2016). Spatial attention reduces burstiness in macaque visual cortical area MST. *Cereb. Cortex*. doi: 10.1093/cercor/bhw326. [Epub ahead of print]. Available online at: http://cercor.oxfordjournals.org/content/early/2016/11/22/cercor.bhw326.abstract

Zoccolan, D., Cox, D. D., and DiCarlo, J. J. (2005). Multiple object response normalization in monkey inferotemporal cortex. *J. Neurosci.* 25, 8150–8164. doi: 10.1523/JNEUROSCI.2058-05.2005

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

II Distinguishing Between Parallel and Serial Processing in Visual Attention by Analyzing Single Spike Trains

To be submitted shortly

Kang Li
Department of Mathematical Sciences, Department of Psychology
University of Copenhagen

Søren Kyllingsbæk
Department of Psychology
University of Copenhagen

Susanne Ditlevsen
Department of Mathematical Sciences
University of Copenhagen

Claus Bundesen
Department of Psychology
University of Copenhagen

Distinguishing Between Parallel and Serial Processing in Visual Attention by Analyzing Spike Trains

Kang Li, Søren Kyllingsbæk, Claus Bundesen, Susanne Ditlevsen
Department of Mathematical Sciences, Department of Psychology
University of Copenhagen

Abstract

Serial and parallel processing in visual search have been long debated in psychology but the processing mechanism remains an open issue. Serial processing allows only one object at a time to be processed, whereas parallel processing assumes that various objects are processed simultaneously. Here we present novel neural models for the two types of processing mechanisms based on analysis of simultaneously recorded spike trains using electrophysiological data from prefrontal cortex of rhesus monkeys while processing task-relevant visual displays. We combine mathematical models describing neuronal attention and point process models for spike trains. The same model can explain both serial and parallel processing by adopting different parameter regimes. We present statistical methods to distinguish between serial and parallel processing based on both maximum likelihood estimates and decoding analysis of the attention when two stimuli are presented simultaneously. Results show that both processing mechanisms are in play for the simultaneously recorded neurons, but neurons tend to follow parallel processing in the beginning after the onset of the stimulus pair, whereas they tend to serial processing later on.

keywords: parallel processing, serial processing, spike train, visual attention, probability mixing, probabilistic modeling, statistical inference, point processes, prefrontal cortex, decoding

1 Introduction

A fundamental question in theories of visual search is whether the process is serial or parallel for given types of stimulus material (for comprehensive reviews, see Bundesen and Habekost (2008); Nobre and Kastner (2013)). In serial search, only one stimulus is attended at a time, whereas in parallel search, several stimuli are attended at the same time. The question of serial versus parallel search has been extensively investigated by behavioral methods in cognitive psychology, but it is still highly controversial. In this article, we briefly review extant empirical methods and their results and then present and exemplify a new method for distinguishing between serial and parallel visual search. The method is based on analysis of spike trains measured in prefrontal cortex of rhesus monkeys while being exposed to a pair of stimuli, which the animal should detect and later respond to with a saccade towards a target object.

1.1 Behavioral methods for distinguishing between serial and parallel visual search

Analyses of effects of display set size on mean response times

In typical experiments on visual search, the task of the observer is to indicate as quickly as possible if a certain type of target is present in a display. Positive (target present) and negative (target absent) mean response times are analyzed as functions of the display set size (the number of items in the display). The

method of analysis was laid out by Sternberg (1966, 1969a,b) and further developed by Schneider and Shiffrin (1977). The foundation is as follows.

In a simple serial model, items are scanned one at a time. When an item is scanned, it is classified as a target or a distractor. The order in which items are scanned is independent of their status as targets versus distractors. A negative response is made when all items have been scanned and classified as distractors. Thus, the number of items processed before a negative response is made equals the display set size, N. Furthermore, the rate of increase in mean negative response time as a function of display set size N equals the mean time taken to process one item, Δt.

In a self-terminating simple serial search process, a positive response is made as soon as a target is found. Because the order in which items are scanned is independent of their status as targets or distractors, the number of items processed before a positive response is made varies at random between 1 and N with a mean of $(1 + N)/2$. Thus, the rate of increase in mean positive response time as a function of display set size N equals one half of the mean time taken to process one item, $\Delta t/2$.

Treisman et al. (1977) introduced an influential distinction between feature and conjunction search. In feature search, the target possesses a simple physical feature (e.g., the color red) that distinguishes the target clearly from all of the distractors. In this case search is fast and little affected by display set size. In conjunction search, the target differs from the distractors by possessing a particular conjunction of physical features (e.g., both a particular color and a particular shape), but the target is not unique in any of the component features of the conjunction (i.e., in color or in shape). For example, the target can be a red B with black Bs and red Xs as distractors.

In typical experiments on conjunction search, positive and negative mean response times have been approximately linear functions of display set size with substantial slopes and positive-to-negative slope ratios of about 1:2. This pattern of results accords with predictions from self-terminating simple serial models, and Treisman and her colleagues have proposed that conjunction search is done by scanning items one at a time. Experiments on feature search with low target-distractor discriminability have yielded similar results (Treisman and Gormican, 1988).

Analyses of effects of display set size on error rates and response time distributions

In a parallel model of attention, several stimuli can be attended at the same time. The first detailed parallel model of visual processing of multi-element displays was the independent channels model proposed by Eriksen and his colleagues (e.g., Eriksen and Lappin (1965); Eriksen and Spencer (1969)). It was based on the assumption that display items presented to separated foveal areas are processed in parallel and independently up to and including the stage of pattern recognition. The independent channels model has been used to account for effects of display set size on error rates.

The linear relations between mean response time and display set size predicted by simple serial models are difficult to explain by parallel models with independent channels. However, the linear relations can be explained by parallel models with limited processing capacity. The following example of mimicry between serial and parallel models was published independently by Atkinson, Holmgren, and Juola and Townsend in 1969.

Consider a display of items that are processed in parallel. Let the processing speed for an item (i.e., the hazard function for the processing time of the item) equal the amount of processing capacity allocated to the item (cf. Bundesen (1990)). Suppose (a) the total processing capacity spread across items in the display remains constant, and (b) whenever an item completes processing, the capacity that was allocated to the item is instantaneously redistributed among any items that remain to be completed. If so, then the mean time taken to complete the parallel processing of the display increases as a linear function of the display set size, mimicking the behavior of a simple serial processor (see Townsend and Ashby (1983), for further analyses of serial-parallel model mimicry).

Bricolo et al. (2002) found evidence of self-terminating serial processing by analysis of response time distributions in a speeded task with highly inefficient search. Search arrays were drawn in light gray against a dark background and consisted of mutually highly similar crosses. A target cross was present in 50% of the trials, and mean response times were approximately linearly increasing functions of display set size with slopes that were twice as steep for target-absent trials (nearly 200 ms per item) as for

target-present trials (100 ms per item). Response time distributions for target-absent responses of individual observers were closely fitted by convolutions of a Gaussian (base response time) distribution with exponential distributions (one for each item in the display). Response time distributions for target-present responses were fitted as probability mixtures of convolutions of the Gaussian distribution with $1, \ldots, N$ exponential distributions, respectively, for displays with N items (some responses being based on the first item that was scanned, others on the second item, and so on). Similar results were obtained in a paradigm in which the position of the target within arrays of constant size was varied instead of varying the number of items. Observers were cued to start search from a particular end of the array, which yielded a position effect that was comparable in size to the display set size effect found in the first experiment. Both the serial model for the first experiment and the serial model for the second experiment could be mimicked by parallel models. However, whereas the parallel model for the set size experiment assumes that processing capacity is reallocated whenever an item finishes processing by being classified as a distractor, the parallel model for the position experiment assumes no reallocation of processing capacity during a trial. It seemed not possible to account for both experiments by a single, reasonably simple parallel model.

Demonstrations of mental states of partial information about each of a multitude of stimuli

Bundesen et al. (2003) introduced a multi-feature whole-report paradigm for investigating serial versus parallel processing: Suppose two features must be processed from each of two stimuli (i.e., a total of four features). Let processing be interrupted before all of the four features have completed processing. If, and only if, processing is parallel, there will be cases in which just one feature from each of the two stimuli completes processing before the interruption. This event, in which the observer has only partially encoded each of the two stimuli, should never happen when processing is serial. Thus, states with partial information from more than one stimulus are strong indicators of parallel processing. In the experiment of Bundesen et al. (2003) (see Kyllingsbæk and Bundesen (2007) for replications and extensions), observers were presented with brief exposures of pairs of colored letters and asked to report both the color and the identity of each letter. The results showed strong evidence of states of partial information from each of the two stimuli (e.g., information of just the identity of one of the letters and just the color of the other one), and the results were fitted strikingly well by a simple parallel-processing model assuming mutually independent processing of the four features.

1.2 Method based on analysis of spike trains

As exemplified above, previous methods for distinguishing between serial and parallel visual search have been based on behavioral data, and the evidence obtained by these methods has been somewhat indirect. In this article, we present a new method for distinguishing between serial and parallel visual search, a method based on analysis of electrophysiological data. The method relies on the probability-mixing model for single neuron processing (Li et al., 2016), derived from the Neural Theory of Visual Attention (Bundesen and Habekost, 2008), which states that when presented with a plurality of stimuli a neuron only responds to one stimulus at any given time. By probabilistic modeling and statistical inference using multiple simultaneously recorded spike trains, we infer and decode what each of the recorded neurons are responding to, providing a mean to distinguish between parallel processing and serial processing on a neuronal level. The new method appears more direct than previous methods.

Consider an experiment in which we record the action potentials or spikes from each of a number of visual cortical neurons of the same type (e.g., a set of functionally similar MT neurons with overlapping receptive fields; see, e.g., Li et al. (2016)). Suppose two stimuli (Stimulus 1 and 2) are both within the classical receptive fields of all of the recorded neurons, but otherwise the receptive fields are empty. In this situation, we may test whether processing is parallel in the sense that on any given trial, some of the recorded neurons represent Stimulus 1 throughout the trial, while others, working concurrently ("in parallel") with the first ones, represent Stimulus 2 throughout the trial. We may assume that a neuron represents Stimulus 1 rather than Stimulus 2 if the likelihood of the observed spike trains becomes higher by assuming that the neuron represents Stimulus 1. We may also test whether processing is strictly serial (i.e., one stimulus at a time) by testing, for example, whether there is a time interval Δ_1 in which all of the MT neurons represent Stimulus 1 and a time interval Δ_2, nonoverlapping with Δ_1, in which all of the MT neurons represent Stimulus 2. Again, we may assume that a neuron represents Stimulus 1

rather than Stimulus 2 if the likelihood of the observed spike trains becomes higher by assuming that the neuron represents Stimulus 1.

Strictly parallel processing of two or more stimuli such that processing begins and ends at precisely the same times may hardly be expected in a biological system. The same is true of strictly serial processing; not every one of the neurons from which we record may represent Stimulus 1 in time interval Δ_1, and not every one may represent Stimulus 2 in time interval Δ_2. In analyses of biological systems, strictly parallel processing and strictly serial processing must be regarded as idealizations. However, we will show how to measure the goodness of approximation of search processes in the brain to simple serial and parallel search.

2 Materials and Methods

Here we present two models that relate the theories of visual attention to neuronal behavior, providing a tool to distinguish or quantify between parallel and serial processing through spike train analysis. Under the assumption of serial processing, the neurons are correlated, acting together as a population. This dependence can arise through two different pathways: 1) There exists an underlying variable driving the neurons towards attending to the same stimulus, creating a dependence, even if the neurons are conditionally independent given the state of this underlying variable. 2) The neurons are directly positively correlated, driving them to synchronize their attention.

The first pathway can be described by a hidden Markov model, where the hidden Markov chain describes different states influencing the neuronal attention. If time is discretized and there are two stimuli, this leads to a mixture of Binomials at each discretized time step, where the Binomial distributions give the probabilities of attending each stimulus, and these probabilities depend on the hidden state of the Markov chain. The second pathway can be represented by a correlated Binomial model, a mixture of an ordinary Binomial and a modified Bernoulli, which is used independently at each discretized time step. For both models, the attended stimulus for each neuron is unobserved, and the inference is based on spike trains. We estimate parameters by maximum likelihood estimation (MLE) by marginalizing out the unobserved attention variables. The estimated parameters in either model describe neuronal properties and are used to obtain a *prior* measurement of the degree of parallel or serial processing. For both models, we also perform a decoding analysis, where we apply the fitted model to the data and obtain the posterior probabilities of the latent attention variables. This is an estimate of which stimuli the neurons were most probably attending to given their observed spike trains. The decoding of attentional behavior gives a *posterior* measurement of the degree of parallel or serial processing. The diagram in Figure 2.1 summarizes the flow of the analysis including parameter estimation, decoding and interpretation. We start by introducing the statistical methods to distinguish between parallel and serial processing, then we present the two models, and finally, we present the experimental data used in the analyses.

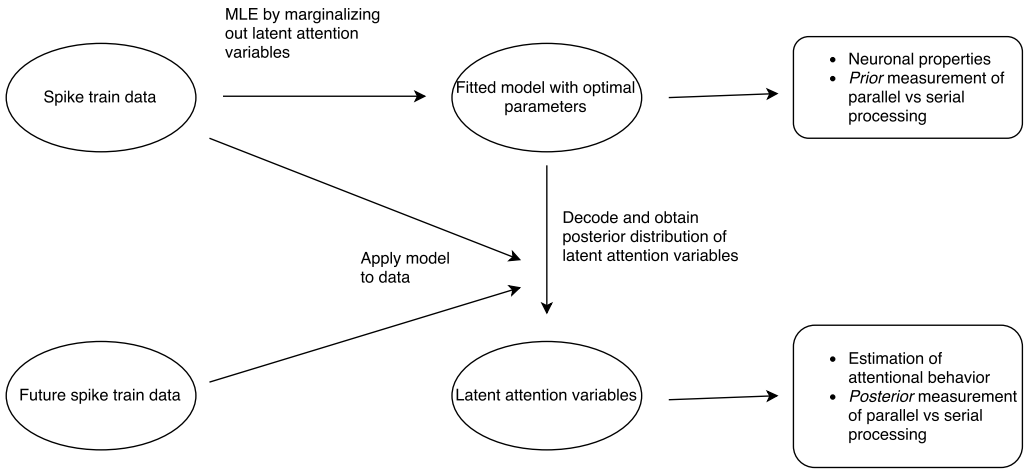


Figure 2.1: Flow diagram of the analysis.

2.1 Distinguishing between parallel and serial processing

In this Section we define different *prior* measures of the degree of serial and parallel processing based on the estimated parameters of the models when a population of n neurons are presented with two non-overlapping stimuli in their receptive fields. These measures will vary with time, i.e., depend on the time since stimulus onset, but for ease of notation, we suppress time from the notation here. Later we will introduce the time dependency. We assume a homogeneous situation where all neurons follow the same distribution and are exchangeable, except for individual firing rates as responses to single stimuli. First, we consider the marginal distribution of the attended stimulus for each neuron. Let p denote the marginal probability of attending to one of the stimuli, say stimulus 1, such that the probability of attending stimulus 2 is $1 - p$. If the neurons are independent, then the probability that all neurons attend the same stimulus is $p^n + (1 - p)^n$, and if the neurons are positively correlated, this is a lower bound of the probability that all neurons attend the same stimulus. Thus, p provides a measure of the tendency of serial or parallel processing. A narrow distribution (extreme probability, p either close to 0 or 1) favors serial processing, since in this case most neurons will attend the same stimulus. On the contrary, a wide distribution (non-extreme probability, p close to 0.5) favors parallel processing, since in this case neuronal attention will tend to split between the two stimuli. Second, we consider correlations between neurons. Since the neurons are exchangeable, the correlation coefficient, denoted by ρ, between any two neurons (pairwise correlation) is identical. Stronger positive correlation implies more tendency to serial processing, no matter what the probability of each stimulus is. Thus, if either the correlation is strong (ρ close to 1) or p is close to 0 or 1, serial processing is favored, while if both the correlation is weak and the probability is not extreme, parallel processing is favored. We summarize the different cases in Table 2.1.

Table 2.1: Effects of neural attentional probability and correlation to serial and parallel processing. Extreme probability implies a probability close to 0 or 1, and strong correlation implies a correlation close to 1.

	Extreme probability	Non-extreme probability
Strong correlation	Serial	Serial
Weak correlation	Serial	Parallel

We now propose a single statistic as an alternative measure to distinguish between serial and parallel processing. Again, we suppose to have a stimulus mixture of two components and a population of n neurons attending to the mixture. The number of neurons, X, attending to the first stimulus follows a distribution with probability mass function (PMF) $f(x)$ for $x \in \{0, 1, \ldots, n\}$, such that $P(X = x) = f(x)$, which depends on the specific model. A distribution centered around $n/2$ indicates apparent parallel processing, and a distribution centered at 0 and/or n indicates apparent serial processing. Note that this population distribution incorporates both the marginal probability of attention of the single neurons and the correlation between neuron pairs. We define a statistic D_n as a measure of the degree of serial or parallel processing, given by

$$D_n = \frac{\sum_{x=0}^{n} |x - n/2| f(x)}{n/2}. \tag{2.1}$$

The statistic D_n can be explained as a normalized expected deviation between the number of neurons attending to one stimulus and the half of the total number of neurons, or the deviation between the expected processing mechanism and the perfect parallel processing with uniform weights on stimuli. If we split the neuron population according to which stimulus they attend giving two proportions (summing to 1), then D_n is the average difference between the two proportions, and it can take values between 0 and 1. The smaller D_n is, the more parallel processing is favored. The D_n statistic depends on the total number of neurons n. However, if we consider specific models for the PMF, for example the Binomial models introduced below, the dependence of n can be removed by using the asymptotic version

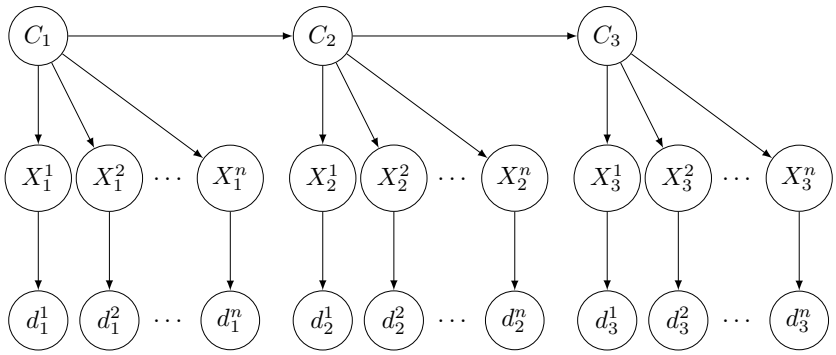
$$D^* = \lim_{n \to \infty} D_n.$$

To summarize, to measure the degree of serial and parallel processing by the estimated model (*prior* measurement), we can use the attentional probability, the correlation of neuronal attention, and the deviation statistic D_n or D^*.

2.2 Hidden Markov Model and a Mixture of Binomial Distributions

In this Section we present a model where some underlying variable drives the attention of the neurons. To combine the visual attention hypotheses with neuronal dynamics, we adopt a hidden Markov model (HMM). We discretize the duration of the trial into T smaller intervals and the HMM is defined over the T time steps. This model is based on the basic probability-mixing model for the attention of single neurons employed in Li et al. (2016), where a neuron responds to a stimulus mixture with certain probabilities, such that the single neuron at any given time represents only one of the stimuli in the mixture. We let these probabilities, which can be interpreted as attentional weights, depend on the underlying hidden Markov chain, which introduces correlation between neurons, even if they are conditionally independent given the hidden state, and the probabilities evolve over time following the dynamics of the hidden Markov chain. Note that this implies that within each of the T intervals, model parameters governing the stochastic neuronal activity (the spike train generation) are constant. For simplicity we use two hidden states, but more could be used. A transition between hidden states introduces a weight reassignment of the attention to the stimuli, and thus, new laws for the generation of spike trains.

Let $C_t \in \{c_1, c_2\}$ denote the hidden state at time t, let $X_t^i \in \{0, 1\}$ denote the attended stimulus of neuron i at time t for $i = 1, \ldots, n$, and let d_t^i denote the observed spike train of neuron i in the t'th interval. Figure 2.2 shows a diagram of the HMM when $T = 3$. Conditional on C_t, $\{X_t^i\}_{i=1,\ldots,n}$ are independent. We set $X_t^i = 1$ when neuron i attends stimulus 1 at time t, and $X_t^i = 0$ when attending stimulus 2.



$C_t \in \{c_1, c_2\}$: hidden state at t
$X_t^i \in \{0, 1\}$: attended stimulus at t for neuron i.
d_t^i: observation of spike train in interval t of neuron i

Figure 2.2: Diagram of the HMM for neuronal attentions from a group of n neurons to a mixture of two stimuli, using $T = 3$ discretized time steps.

Let the initial distribution of the Markov chain be given by $\boldsymbol{\lambda}$ and the transition probability matrix (TPM) by $\boldsymbol{\Gamma}$:

$$\begin{aligned} \boldsymbol{\lambda} &= \begin{bmatrix} \lambda & 1-\lambda \end{bmatrix}, \\ \boldsymbol{\Gamma} &= \begin{bmatrix} \gamma_{11} & 1-\gamma_{11} \\ \gamma_{21} & 1-\gamma_{21} \end{bmatrix}, \end{aligned} \tag{2.2}$$

where $0 \leq \lambda, \gamma_{11}, \gamma_{21} \leq 1$. The TPM $\boldsymbol{\Gamma}$ depends on the stimulus pair, but the initial distribution $\boldsymbol{\lambda}$ is only related to the location of the attended stimulus and is thus the same for all stimulus pairs. We denote by $\boldsymbol{\Gamma}_m$ the TPM of condition m.

Conditional on C_t, neurons are independent. Denote the probability of attending to stimulus 1 given state c by $\alpha_{c1} = P(X_t^i = 1 | C_t = c)$, yielding the matrix:

$$\boldsymbol{A} = \begin{bmatrix} \alpha_{c_1 1} & 1-\alpha_{c_1 1} \\ \alpha_{c_2 1} & 1-\alpha_{c_2 1} \end{bmatrix}. \tag{2.3}$$

Attention probabilities and correlations Calculating the probability distribution of X_t^i is straightforward following the HMM. Let $P(C_t = c_1) = \pi_t$ and $P(C_t = c_2) = 1 - \pi_t$ denote the distribution of the hidden state, and $P(X_t^i = 1) = p_t$ and $P(X_t^i = 0) = 1 - p_t$ denote the distribution of the attended stimulus at time t. Then

$$\begin{bmatrix}\pi_t & 1-\pi_t\end{bmatrix} = \boldsymbol{\lambda}\boldsymbol{\Gamma}^{t-1}; \tag{2.4}$$

$$\begin{bmatrix}p_t & 1-p_t\end{bmatrix} = \boldsymbol{\lambda}\boldsymbol{\Gamma}^{t-1}\boldsymbol{A}. \tag{2.5}$$

Straightforward calculations yield the correlation ρ_t between two neurons X_t^i and X_t^j:

$$p_t = \pi_t\alpha_{c_1 1} + (1-\pi_t)\alpha_{c_2 1}; \tag{2.6}$$

$$\mathrm{Var}(X_t^i) = \pi_t\alpha_{c_1 1} + (1-\pi_t)\alpha_{c_2 1} - (\pi_t\alpha_{c_1 1} + (1-\pi_t)\alpha_{c_2 1})^2; \tag{2.7}$$

$$\begin{aligned}\mathrm{Cov}(X_t^i X_t^j) &= \pi_t\alpha_{c_1 1}\alpha_{c_1 1} + (1-\pi_t)\alpha_{c_2 1}\alpha_{c_2 1} - (\pi_t\alpha_{c_1 1} + (1-\pi_t)\alpha_{c_2 1})^2 \\ &= \pi_t(1-\pi_t)(\alpha_{c_1 1} - \alpha_{c_2 1})^2; \end{aligned} \tag{2.8}$$

$$\rho_t = \frac{\mathrm{Cov}(X_t^i X_t^j)}{\sqrt{\mathrm{Var}(X_t^i)}\sqrt{\mathrm{Var}(X_t^j)}}. \tag{2.9}$$

The values p_t and ρ_t can be used to measure the degree of serial and parallel processing as explained in Table 2.1.

A mixture of two Binomials Investigating the HMM structure, at each time point t we can view the neuronal attention behavior for the n neurons as a mixture of two Binomial distributions, $Bin2(\pi_t, \alpha_{c_1 1}, \alpha_{c_2 1}, n)$, by marginalizing out the hidden state C_t. The weight of the first Binomial component is π_t, and the probability parameter of the cth Binomial is α_{c1} for $c = c_1, c_2$. The number of Binomial trials equals the number of simultaneously recorded neurons n. The PMF for a mixture of two Binomials is

$$f_{Bin2}(x|\pi_t, \alpha_{c_1 1}, \alpha_{c_2 1}, n) = \pi_t f_{Bin}(x|\alpha_{c_1 1}, n) + (1-\pi_t) f_{Bin}(x|\alpha_{c_2 1}, n), \tag{2.10}$$

where $f_{Bin}(\cdot)$ is the PMF of the Binomial distribution.

Figure 2.3 illustrates the mixture of two Binomials using $n = 10$ neurons for different parameter sets. The probability p and the correlation ρ are also calculated for each case. The figure illustrates how the probability and the correlation affect serial and parallel processing. Only when p is not close to 0 or 1 and the correlation is weak, shown in the bottom-right panel, the 10 neurons tend to split between the two stimuli, indicating parallel processing. Otherwise, a large majority of neurons attend to the same stimulus, suggesting serial processing.

The D_n statistic is calculated using Equation (2.1). For the mixture of two Binomials in (2.10), the asymptotic version is given by

$$D^* = \lim_{n\to\infty} D_n = 2\left(\pi_t|\alpha_{c_1 1} - 0.5| + (1-\pi_t)|\alpha_{c_2 1} - 0.5|\right). \tag{2.11}$$

The corresponding D_n and D^* values are also shown in Figure 2.3. In 3 of the 4 cases, the non-asymptotic statistic is equal to the asymptotic version, even if n is as small as 10. In the last case (bottom-right panel) we see that $D^* < D_n$, so if more neurons are involved, we expect even more clear parallel processing for the given parameters.

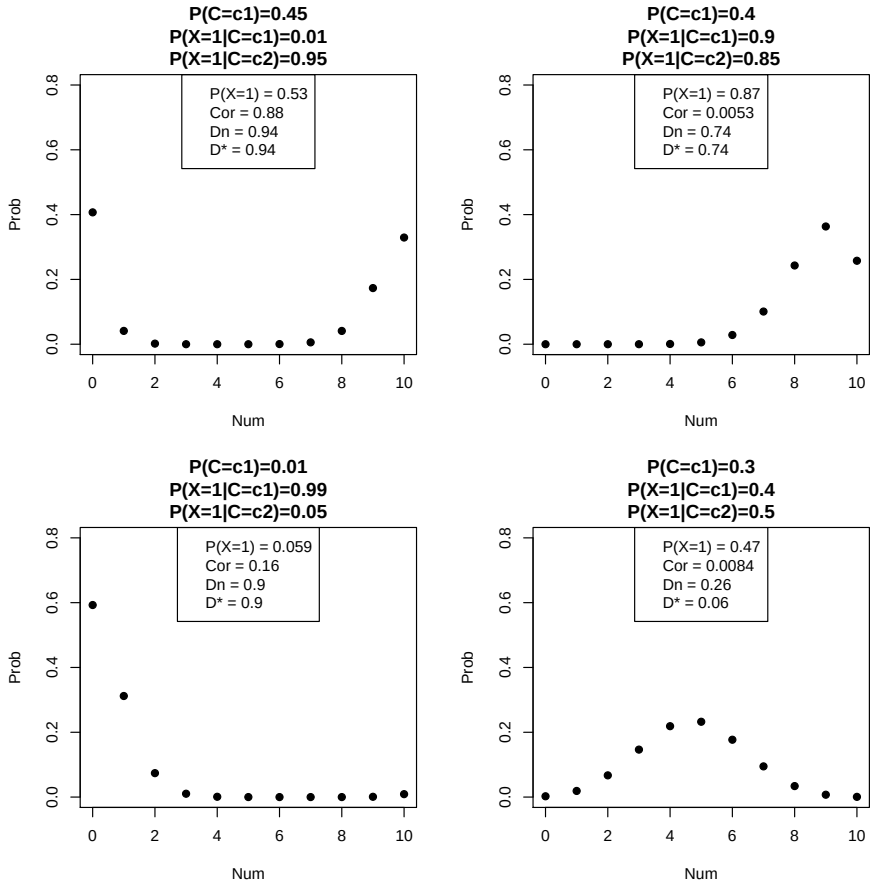


Figure 2.3: The PMF of a mixture of two Binomials for different parameter settings.

2.3 Correlated Binomial model

In this Section we present the correlated Binomial model where the neurons are assumed directly correlated. It was studied in Luceño (1995); Diniz et al. (2010), and is denoted by $CBin(n, p, \rho)$, where n is the number of correlated Bernoulli trials (simultaneously recorded neurons in our model setting), $0 \leq p \leq 1$ is the success probability, and ρ is the correlation coefficient. In this model the number of successes x follows a mixture of two distributions. One is an ordinary Binomial distribution with parameters n and p. The other is a fully correlated distribution where $x \in \{0, n\}$, which can be viewed as a modified Bernoulli distribution with support $\{0, n\}$ with parameter p. The weight of the Bernoulli component is the correlation coefficient ρ. The probability mass function is given by

$$f_{CBin}(x|n, p, \rho) = (1-\rho) f_{Bin}(x|n, p) + \rho p^{\frac{x}{n}} (1-p)^{\frac{n-x}{n}} I_{\{0,n\}}(x), \tag{2.12}$$

where $I_{\{0,n\}}(x)$ is an indicator function which equals 1 for $x \in \{0, n\}$ and 0 otherwise.

We discretize the observation interval as before, and at each discretized time step, we apply a correlated Binomial distribution. We assume the distribution at the first step identical for all stimulus pairs, since this is the initiation of the processing mechanism before the specific stimuli are perceived, but at all later steps, the distribution depends on the stimulus pair. Thus, at $t = 1$ the simultaneously recorded neurons follow $CBin(n, p_1, \rho_1)$, and at $t > 1$ they follow $CBin(n, p_{t,m}, \rho_{t,m})$ for stimulus pair m. We do not assume a dependence structure over time, as in the HMM, and the behavior at each time step is independent of the behavior at other time steps. Instead, the correlation between simultaneously recorded neurons are modeled directly by the parameter ρ in the correlated Binomial distribution. Compared with the HMM, where the correlation is described through the attentional reassignment with a Markov chain, the correlated Binomial model is more direct.

Here in the correlated Binomial model, we denote by C_t the hidden index, indicating either the Binomial ($C_t = c_1$) or the Bernoulli ($C_t = c_2$) component in the mixture.

Measuring the degree of serial and parallel processing For the correlated Binomial model, the probability of attention is directly obtained from the parameter $p_{t,m}$, and the correlation is obtained from $\rho_{t,m}$. The asymptotic version of the deviation statistic D^* is given by

$$D^* = \frac{(1-\rho)|p-0.5| + 0.5\rho}{0.5}. \tag{2.13}$$

Figure 2.4 shows for different parameter values the PMF of the correlated Binomial distribution. The D_n and D^* are also indicated.

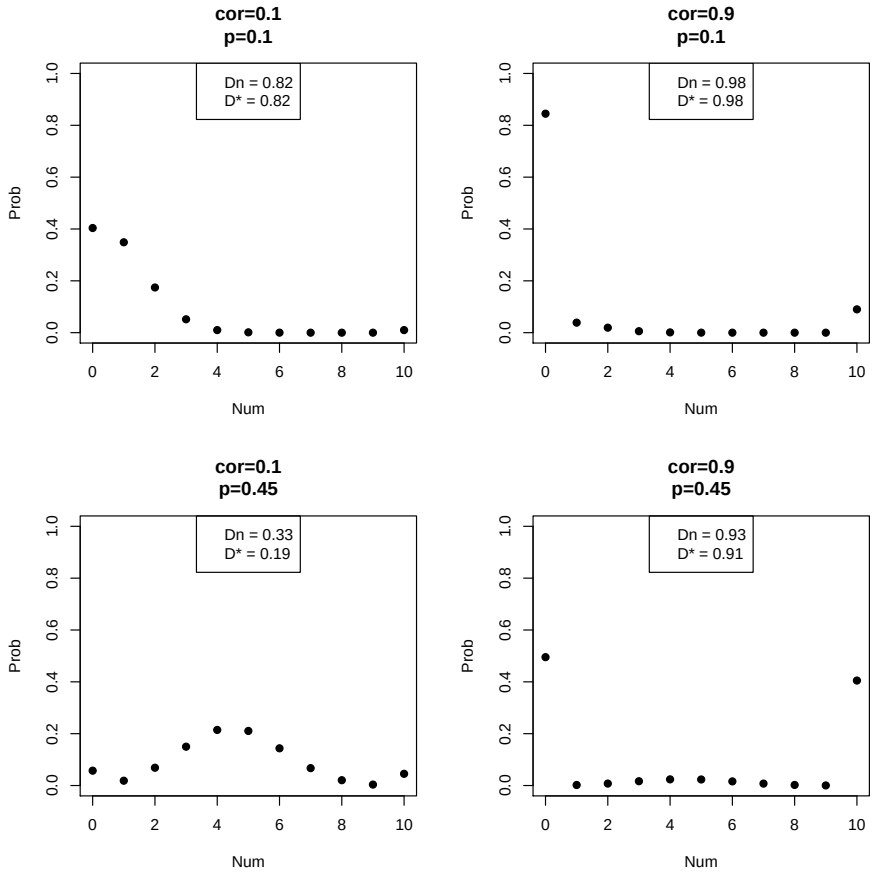


Figure 2.4: The PMF of the correlated Binomial distribution for different parameter settings.

2.4 Decoding

Decoding means to infer the attended stimulus from the observations and the estimated parameters. For readability, we suppress time and neuron indicator from the notation, denoting the hidden state by C, the attended stimulus by X and the data by d. The posterior of X given d is

$$P(X|d) = \sum_c P(X|C=c,d)P(C=c|d). \tag{2.14}$$

The strategy is to first estimate $P(C|d)$ and then $P(X|C,d)$ conditional on C. We are particularly interested in the PMF and the deviation statistic of the attended stimuli, which we can calculate using

$P(X|C,d)$ for different states C. In the following, the decoding is explained for the two models in more detail.

Decoding in the Binomial-HMM

First we decode the hidden states C_t in the Binomial-HMM model. It is performed at each discretized time step following the forward-backward algorithm. Let $d_{s:t}^{\mathcal{N}_k}$ denote the spike trains in intervals s to t, for $1 \leq s \leq t \leq T$ in trial k, where $\mathcal{N}_k$ denotes the simultaneous neurons recorded in the trial k. The probability of C_t conditional on the observed spike trains at all time intervals $1:T$ can be expressed as

$$P(C_t|d_{1:T}^{\mathcal{N}_k}) \propto P(d_{t+1:T}^{\mathcal{N}_k}|C_t)P(C_t|d_{1:t}^{\mathcal{N}_k}), \tag{2.15}$$

where

$$P(C_t|d_{1:t}^{\mathcal{N}_k}) \propto P(d_t^{\mathcal{N}_k}|C_t) \sum_{C_{t-1}} P(C_t|C_{t-1})P(C_{t-1}|d_{1:t-1}^{\mathcal{N}_k}) \tag{2.16}$$

is the forward probability, calculated recursively by a forward sweep over $1:T$, and

$$P(d_{t+1:T}^{\mathcal{N}_k}|C_t) = \sum_{C_{t+1}} P(d_{t+2:T}^{\mathcal{N}_k}|C_{t+1})P(d_{t+1}^{\mathcal{N}_k}|C_{t+1})P(C_{t+1}|C_t) \tag{2.17}$$

is the backward probability, calculated recursively by a backward sweep over $T:1$. When calculating the forward and backward probabilities, the likelihood conditional on the hidden state, $P(d_t^{\mathcal{N}_k}|C_t)$, is obtained by conditioning on $\{X_t^i\}_{i\in\{\mathcal{N}_k\}}$:

$$P(d_t^{\mathcal{N}_k}|C_t) = \prod_{i\in\mathcal{N}_k} \sum_{X_t^i\in\{0,1\}} P(d_t^i|X_t^i)P(X_t^i|C_t). \tag{2.18}$$

After decoding the hidden state $P(C_t|d_{1:T}^{\mathcal{N}_k})$, the next is to decode $\{X_t^i\}_{i\in\{\mathcal{N}_k\}}$ conditional on C_t:

$$P(X_t^i|d_t^{\mathcal{N}_k}, C_t) = P(X_t^i|d_t^i, C_t) \propto P(d_t^i|X_t^i, C_t)P(X_t^i|C_t). \tag{2.19}$$

For each data set in trial k, $d_{1:T}^{\mathcal{N}_k}$, we have thus obtained the posterior probability of the hidden states $P(C_t|d_{1:T}^{\mathcal{N}_k})$ and the attended stimulus of each spike train $P(X_t^i|d_t^i, C_t)$, at all time steps $t = 1,\ldots,T$. This yields the marginal posterior $P(X_t^i|d_{1:T}^{\mathcal{N}_k}) = \sum_{C_t\in\{c_1,c_2\}} P(X_t^i|d_t^i, C_t)P(C_t|d_{1:T}^{\mathcal{N}_k})$.

At each time step t, conditional on C_t, spike trains are independent and the posterior probabilities $P(X_t^i|d_t^i, C_t)$ are different from spike train to spike train. Thus, for the attended stimuli of all neurons we have a Poisson Binomial distribution, a generalization of the ordinary Binomial distribution where each Bernoulli trial has a distinct success probability (Hodges and Le Cam, 1960). The PMF of the Poisson Binomial distribution is calculated numerically using methods from Hong (2013). Marginalizing out $C_t \in \{c_1, c_2\}$, at each time step t we then have a mixture of two Poisson Binomial distributions. The PMF of this mixture distribution can be regarded as probabilities of the number of neurons that have attended stimulus one, conditional on their observed spike trains. Furthermore, the deviation statistic D_n can also be obtained from the PMF.

Decoding in the Correlated Binomial

In the correlated Binomial model, the attended stimuli of all simultaneous spike trains at one time step follow a correlated Binomial distribution, a mixture of an ordinary Binomial and a fully correlated Bernoulli. Data between different time steps and different trials are independent. Thus, decoding can simply be done independently for each discretized time step in each trial. Now, let C_t be an index indicating either the Binomial or the Bernoulli component in the mixture. As previous, we first decode C_t by calculating $P(C_t|d_t^{\mathcal{N}_k})$, then find the PMF by calculating $P(X_t^i|d_t^i, C_t)$.

Following the correlated Binomial model,

$$P(C_t|d_t^{\mathcal{N}_k}) \propto P(d_t^{\mathcal{N}_k}|C_t)P(C_t), \tag{2.20}$$

where we calculate the two cases $C_t = c_1$ and $C_t = c_2$ following the two components as in eq. (2.28) to be shown later. Then for each case of C_t we decode the attended stimulus X_t^i. When $C_t = c_1$,

i.e., the Binomial case, X_t^i is obtained for each spike train independently with $P(X_t^i|d_t^i, C_t = c_1) \propto P(d_t^i|X_t^i, C_t = c_1)P(X_t^i|C_t = c_1)$, resulting in a Poisson Binomial distribution. When $C_t = c_2$, i.e., the fully correlated Bernoulli case, the attended stimuli of all neurons are the same, which is obtained by $P(X_t|d_i^{\mathcal{N}_k}, C_t = c_2) \propto P(d_i^{\mathcal{N}_k}|X_t, C_t = c_2)P(X_t|C_t = c_2)$, and the result is still a modified Bernoulli. Finally, the PMF is a mixture of a Poisson Binomial and a modified Bernoulli.

2.5 Experimental Data

To distinguish between parallel and serial processing, we use the neural spike train data recorded from neurons in prefrontal cortex of two rhesus monkeys presented with two visual stimuli in experiments conducted by Kadohisa et al. (2013). They studied dynamic attentional construction, and found that in the early stage after stimulus onset when processing competing stimuli, the global attention is distributed among all objects with each neuron having a tendency towards its contralateral hemifield. In the late stage, the global attention is reallocated and neurons are redirected to the target stimulus. The data contain multiple simultaneously recorded neurons responding to two competing stimuli. The data are organized in daily sessions, and each session consists of a different set of recorded neurons. We only analyze the sessions where at least five neurons are recorded to have enough data to distinguish between parallel and serial processing, yielding a total of 48 sessions. Figure 2.5 shows a typical trial of the experiment. Each trial began with a central cue indicating the target object of the specific trial. Two different cues were paired with two alternative targets. After a brief delay, a choice display was presented for 500 ms containing two objects to the right and left of the fixation point. The objects could be either the cued target (T), an inconsistent non-target (NI) because it was used as a target on other trials, a consistent nontarget (NC) never serving as a target, or nothing but a gray dot (NONE). In the following we call a combination of two stimuli a *condition*. Table 2.2 shows the 12 possible conditions. The stimuli locations were denoted by whether they were contra- or ipsilateral with respect to the recorded neuron. For illustration purposes, in the figures left represents contralateral and right ipsilateral. After a brief delay, the monkey was rewarded with a drop of liquid for a saccade to the T location, or if no T had been presented, for maintaing fixation (no-go response) for later reward. Figure 2.6 A shows an example of the structure of the data in one session. In this example, five neurons are simultaneously recorded. One condition is repeated in multiple trials, and each trial might record some or all of the five neurons. To get an overall idea of the sample sizes, histograms in Figure 2.6 B and C show the average number of trials per condition over the 48 sessions, and the average number of simultaneously recorded neurons per trial over sessions, respectively.

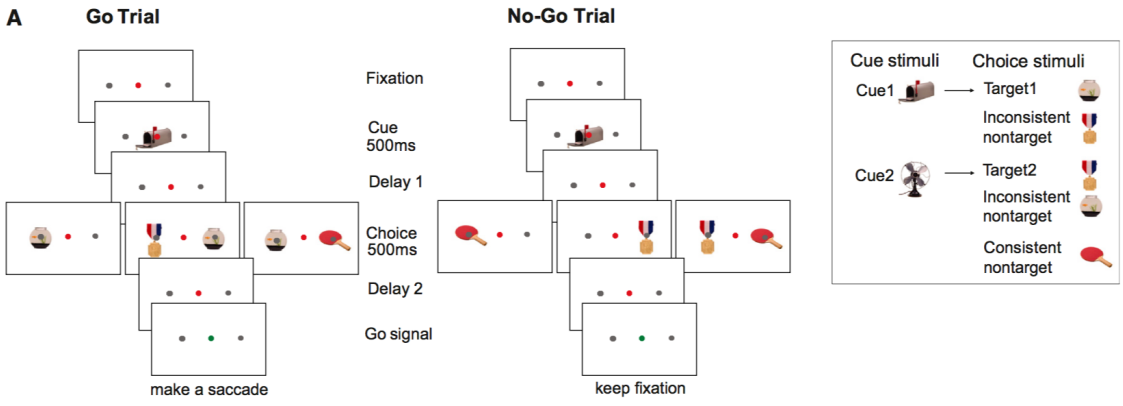


Figure 2.5: The trial setup in the experiment conducted by Kadohisa et al. (2013). Following fixation on a central red dot, each trial began with a central cue indicating the target object. The cue was paired with two alternative targets. After a brief delay, a choice display was presented containing two objects to the right and left of the fixation point. The objects could be either the cued target (T), an inconsistent non-target (NI) because it was used as a target on other trials, a consistent nontarget (NC) never serving as a target, or nothing but a gray dot (NONE). After a brief delay, the monkey was rewarded with a drop of liquid for a saccade to the T location (in the cases shown to the left), or if no T had been presented (cases shown in the middle), for maintaing fixation (no-go response) for later reward. To the right are shown two examples of stimuli combinations.

Table 2.2: The 12 conditions used in the trials (combinations of stimuli). Conditions can be merged into three, indicated by table cells: target in the contralateral side, target in the ipsilateral side, and all combinations with no target. Contra- and ipsilateral sides are with respect to the recorded neuron.

condition	1	2	3	4	5	6	7	8	9	10	11	12
con	T	T	T	NI	NC	NONE	NI	NI	NC	NC	NONE	NONE
ipsi	NI	NC	NONE	T	T	T	NC	NONE	NONE	NI	NI	NC

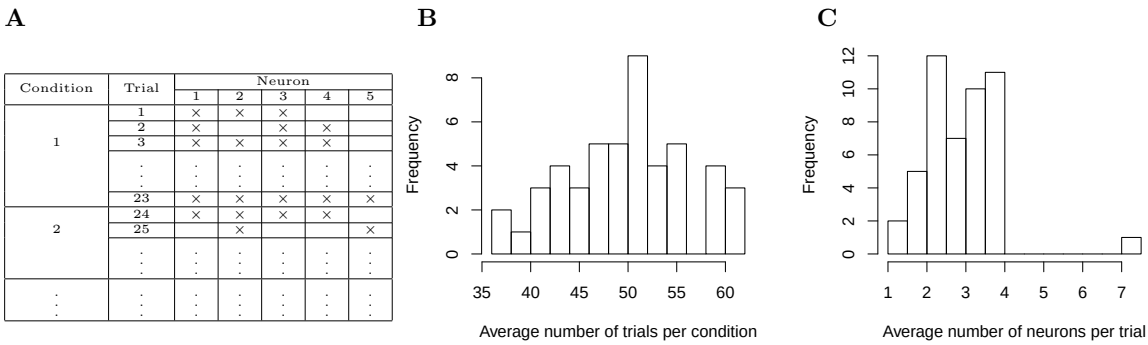


Figure 2.6: Sample sizes. A: Example of number of trials and recorded neurons in a daily session. The cross symbol (×) indicates that the neuron is recorded in the given trial. In this session, five neurons are recorded. Condition 1 was used in 23 trials, and Trial 1 and 2 each uses three neurons, but not the same ones. B: Average number of trials per condition in 48 sessions. C: Average number of neurons per trial in 48 sessions. In all sessions, at least 5 neurons are recorded. Histograms are based on 48 numbers (one for each session).

We will analyze the choice phase where the two stimuli are shown. In Figures 2.7 and 2.8 are shown the recorded spike trains of two example cells during this phase and 100 ms around it. The red line is a kernel smoothing of firing rates over time, plotted on top of the spike trains. The 12 subplots show the 12 conditions with the titles indicating stimulus on the contra- (left) and ipsilateral (right) sides with respect to the recorded neuron. The two figures show two complementary neurons. The neuron "MN110411task_3_0" in Figure 2.7 favors the target with a higher firing rate for T, and its attention starts from the contralateral stimulus and is later redirected to the target stimulus, following the overall tendency of most neurons reported by Kadohisa et al. (2013). On the other hand, the neuron "mj081029a_8_0" in Figure 2.8 shows a tendency to the ipsilateral stimulus in the early stage, and later the attention is again redirected to the target stimulus. Further, for this neuron there are more variability between trials under the same condition.

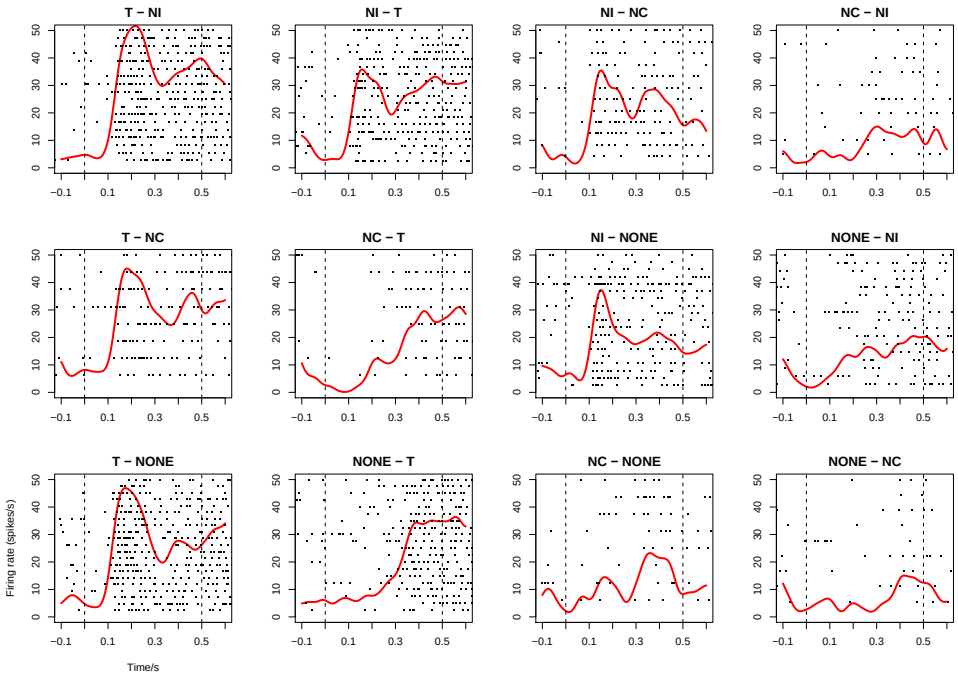


Figure 2.7: Spike trains recorded from an example cell (neuron MN110411task_3_0). Raster plots recorded under 12 conditions, indicated in the title of the subplot, together with a kernel smoothing estimate of the firing rate shown in red. The left stimulus in the title indicates the stimulus of the contralateral side, and the right indicates the stimulus on the ipsilateral side with respect to the recorded neuron. The dashed lines indicate the interval of the choice phase where two stimuli are shown.

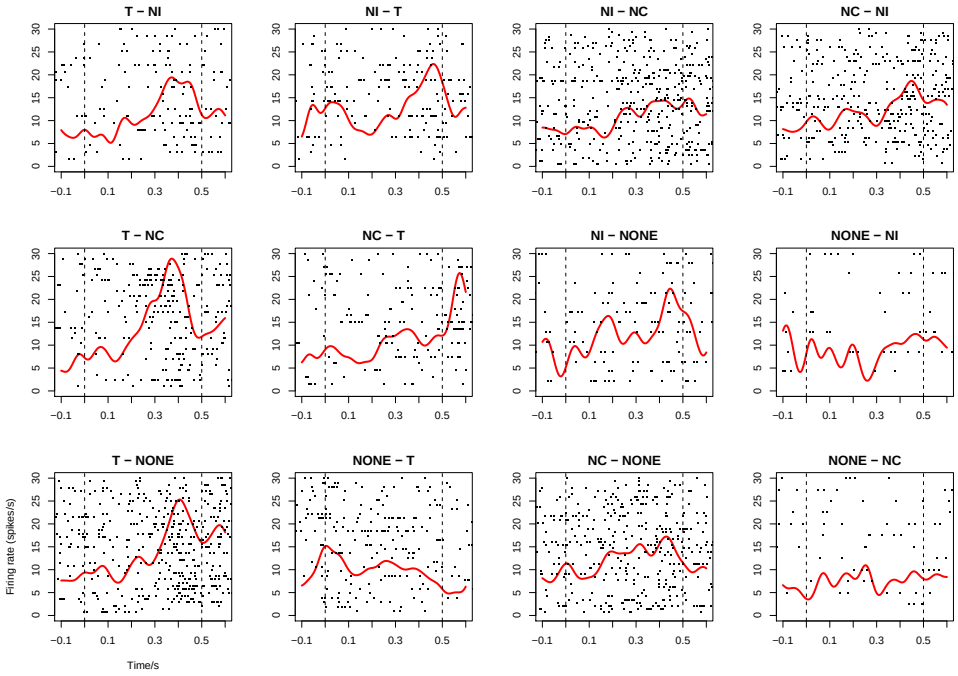


Figure 2.8: Spike trains recorded from an example cell (mj081029a_8_0). Raster plots recorded under 12 conditions, indicated in the title of the subplot, together with a kernel smoothing estimate of the firing rate shown in red. The left stimulus in the title indicates the stimulus of the contralateral side, and the right indicates the stimulus on the ipsilateral side with respect to the recorded neuron. The dashed lines indicate the interval of the choice phase where two stimuli are shown.

The above figures present repeated trials of a single neuron, but not simultaneously recorded spike trains in single trials. In Figure 2.9, we show simultaneously recorded neurons in two conditions of the session "MN110411". Different trials are shown in two colors alternately, and all simultaneously recorded spike trains within one trial are shown in the same color. The comparison of serial and parallel processing catches the difference among simultaneously recorded neurons within one trial in terms of their attended stimulus, which is hard or impossible to analyze by traditional methods by averaging across neurons and trials. We thus develop a new methodology modeling each single spike train and the correlation between spike trains. The serial and parallel processing can be distinguished using the estimated parameters.

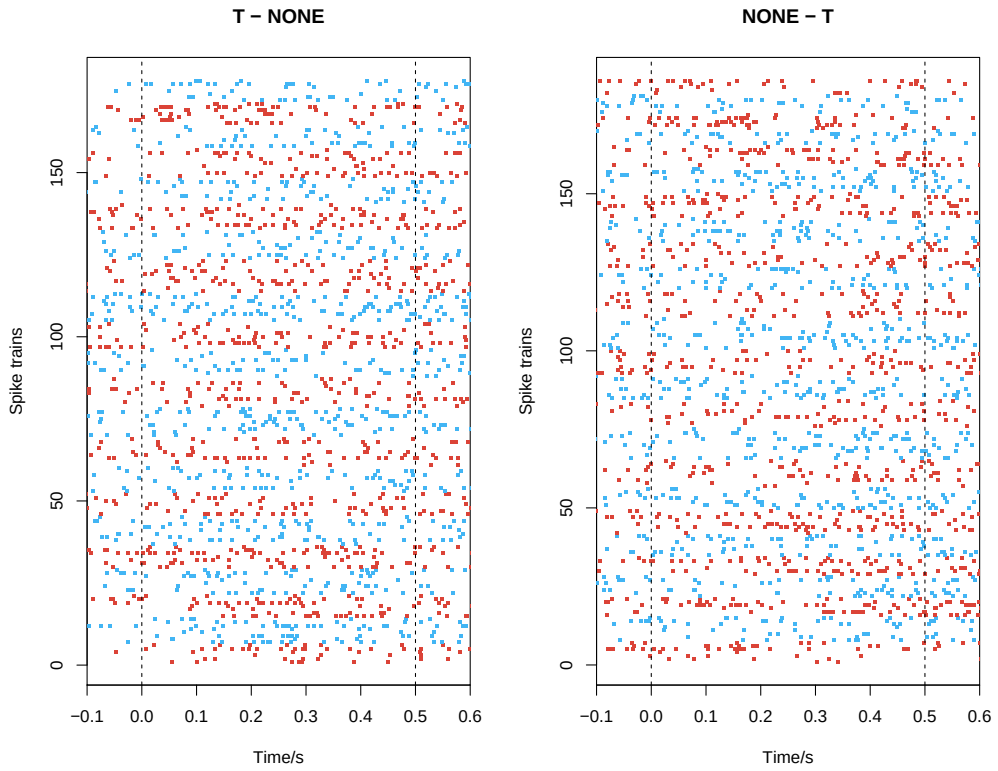


Figure 2.9: Spike trains of simultaneously recorded neurons in session "MN110411" for two conditions. Each point in the figure denotes a spike at the time indicated by the x-axis. Different trials are presented alternately using the red and blue colors, and the simultaneously recorded spike trains within one trial are shown in the same color. The left and right panels show two different conditions.

To account for neuronal response times, we discard the first 100 ms after stimulus onset, using the interval from 100 to 500 ms in the choice phase when estimating the parameters of the two models.

2.6 Likelihood functions

The spike trains are modelled by point processes using conditional intensity functions (CIF) (Daley and Vere-Jones, 2003; Kass et al., 2014), see also Li et al. (2016). Suppose a spike train d in the interval $[T_s, T_e]$ contains the spike times $d = \{t_1, t_2, \dots\}$ with $T_s \le t_1 < t_2 < \cdots \le T_e$, and that it attends to the same stimulus during the entire interval. The probability of d given the attended stimulus X_t is given by (Kass et al., 2014; Truccolo et al., 2005)

$$P(d|X_t) = \left[\prod_{\tau \in d} h(\tau|H_\tau; X_t)\right] \exp\left\{-\int_{T_s}^{T_e} h(s|H_s; X_t)ds\right\}, \tag{2.21}$$

where H_s is the spike history up to time s, and $h(s|H_s; X_t)$ is the conditional intensity function, which we model using

$$h(s|H_s; X_t) = r \exp\left\{\beta_0 s + \sum_{j=1}^{10} \beta_j \Delta N_{s-ju}\right\}. \tag{2.22}$$

The base firing rate r is neuron specific and a function of the attended stimulus and the location (contra- or ipsilateral). For each neuron, there are 7 rate parameters, representing T, NI and NC at either side, and a parameter for NONE. The exponential term models the influence of past spikes on the neuronal activity. For simplicity, we assume that only past spikes of the neuron itself have an effect. All neurons are assumed to share the same set of β parameters $\beta_j, j = 0, 1, 2, \ldots, 10$. The constant $u = 1$ms is the discretization unit and $\Delta N_{s-ju} \in \{0, 1\}$ denotes whether there is a spike at j time units before the current time s. Finally, $\beta_0 s$ models an exponential decay of the firing rate over time from time 0 when stimuli appear.

Let $\mathcal{M}$ denote the considered conditions (stimulus pairs) and let $|\mathcal{M}|$ denote the number of conditions. For simplicity, we do not always distinguish between all 12 conditions shown in Table 2.2, but sometimes merge them into classes depending on our emphasis, such that there will be fewer parameters to estimate. In particular, we will consider the three classes of conditions indicated in the table, defined by whether there is a target in the stimulus pair, and if there is, whether it is contra- or ipsilateral. Under condition m, let the set $\mathcal{K}_m$ contain all the conducted trials. In trial k, let the set $\mathcal{N}_k$ contain all the simultaneously recorded neurons and let $d_t^{\mathcal{N}_k}$ denote the spike trains from these neurons in the t'th interval. Each $\mathcal{N}_k$ is a subset of the set of all neurons $\mathcal{N}$ used in the session, $\mathcal{N}_k \subseteq \mathcal{N}$, because not all neurons are used in all trials.

Model fitting in the Binomial-HMM The likelihood function of all spike trains in one session is given by

$$L = \prod_{m \in \mathcal{M}} \prod_{k \in \mathcal{K}_m} \left\{ \boldsymbol{\lambda} \boldsymbol{P}(d_1^{\mathcal{N}_k}|C_1) \prod_{t=2}^{T} \left[\boldsymbol{\Gamma}_m \boldsymbol{P}(d_t^{\mathcal{N}_k}|C_t)\right] \right\}. \tag{2.23}$$

We denote the conditional probability of the $\mathcal{N}_k$ spike trains at time t given C_t by a diagonal matrix:

$$\boldsymbol{P}(d_t^{\mathcal{N}_k}|C_t) = \begin{bmatrix} P(d_t^{\mathcal{N}_k}|C_t = c_1) & 0 \\ 0 & P(d_t^{\mathcal{N}_k}|C_t = c_2) \end{bmatrix}. \tag{2.24}$$

By conditioning on $X_t^{\mathcal{N}_k}$, we obtain

$$\begin{aligned} P(d_t^{\mathcal{N}_k}|C_t = c) &= \prod_{i \in \mathcal{N}_k} \left[P(d_t^i|C_t = c)\right] \\ &= \prod_{i \in \mathcal{N}_k} \left[P(X_t^i = 1|C_t = c)P(d_t^i|X_t^i = 1) + P(X_t^i = 0|C_t = c)P(d_t^i|X_t^i = 0)\right] \\ &= \prod_{i \in \mathcal{N}_k} \left[\alpha_{c1} P(d_t^i|X_t^i = 1) + (1 - \alpha_{c1})P(d_t^i|X_t^i = 0)\right], \end{aligned} \tag{2.25}$$

or in matrix notation:

$$P(d_t^{\mathcal{N}_k}|C_t = c) = \prod_{i \in \mathcal{N}_k} \left\{ \boldsymbol{I}_c \boldsymbol{A} \begin{bmatrix} P(d_t^i|X_t^i = 1) \\ P(d_t^i|X_t^i = 0) \end{bmatrix} \right\}, \tag{2.26}$$

$$\boldsymbol{I}_c = \left\{ \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, c = c_1 \right., \tag{2.27}$$

where $P(d_t^i|X_t^i)$ is given in (2.21). We obtain maximum likelihood estimates of the parameters by maximizing the likelihood function. The parameters to be inferred are summarized in Table 2.3.

Table 2.3: Parameters to be estimated for each session in the Binomial-HMM model.

Name	Explanation	Dimension
$\boldsymbol{\lambda} = \begin{bmatrix} \lambda & 1-\lambda \end{bmatrix}$	Initial distribution, the same for all conditions $\mathcal{M}$	1
$\boldsymbol{\Gamma}_m = \begin{bmatrix} \gamma_{11}^m & 1-\gamma_{11}^m \\ \gamma_{21}^m & 1-\gamma_{21}^m \end{bmatrix}$	Transition probability matrix for each condition $m \in \mathcal{M}$	$2\lvert\mathcal{M}\rvert$
$\boldsymbol{A} = \begin{bmatrix} \alpha_{11} & 1-\alpha_{11} \\ \alpha_{21} & 1-\alpha_{21} \end{bmatrix}$	Conditional probability of neuronal attention	2
r	Base firing rates, different for each neuron in $\mathcal{N}$	$7\lvert\mathcal{N}\rvert$
β	Weights in the CIF model, the same for all neurons $\mathcal{N}$	11

Model fitting in the correlated Binomial model Under the correlated Binomial model, the simultaneously recorded neurons follow a mixture of a Binomial and a modified Bernoulli. The likelihood of the spike trains in condition m at time t in trial k, $d_t^{\mathcal{N}_k}$, is given by

$$P_m(d_t^{\mathcal{N}_k}) = (1-\rho_{t,m}) \underbrace{\prod_{i\in\mathcal{N}_k} \left[P(d_t^i|X_t^i=1)p_{t,m} + P(d_t^i|X_t^i=0)(1-p_{t,m})\right]}_{\text{Binomial}} + \underbrace{\rho_{t,m} \left\{ p_{t,m} \prod_{i\in\mathcal{N}_k} P(d_t^i|X_t^i=1) + (1-p_{t,m}) \prod_{i\in\mathcal{N}_k} P(d_t^i|X_t^i=0) \right\}}_{\text{modified Bernoulli}}, \tag{2.28}$$

where $P(d_t^i|X_t^i)$ is given in (2.21). Then the likelihood of the data of an entire session is given by:

$$L = \prod_{m\in\mathcal{M}} \prod_{k\in\mathcal{K}_m} \prod_{t=1}^{T} P_m(d_t^{\mathcal{N}_k}). \tag{2.29}$$

The parameters of this model are summarized in Table 2.4.

Table 2.4: Parameters that need to be estimated for the independent correlated Binomial model.

Name	Explanation	Dimension
$\rho_{t,m}$	Correlation coefficients at each condition $m \in \mathcal{M}$ and time $t = 1, \ldots, T$	$\lvert\mathcal{M}\rvert \cdot (T-1) + 1$
$p_{t,m}$	Probability parameter at each condition $m \in \mathcal{M}$ and time $t = 1, \ldots, T$	$\lvert\mathcal{M}\rvert \cdot (T-1) + 1$
r	Base firing rates, one for each neuron in $\mathcal{N}$	$7\lvert\mathcal{N}\rvert$
β	Weights in the CIF model, the same for all neurons $\mathcal{N}$	11

We summarize the differences of the Binomial-HMM and the correlated Binomial model in Table 2.5. In both models, it is assumed that in the early stage, i.e., the first discretized interval from 100 ms to $100 + 400/T$ ms, neuronal attention is only affected by the position of stimuli (ipsi- or contralateral) and not by stimulus types (T, NI, NC or NONE). This assumption is supported by the empirical findings by firing rate averaging showing attentional reallocation over time (Kadohisa et al., 2013). It is also assumed that under the same stimulus types, the attentional parameters are identical, implying that in all the trials of one condition, neurons follow the same attentional probabilities. There may be differences from trial to trial, but the trials follow the same distribution.

Table 2.5: Differences of the Binomial-HMM and the correlated Binomial model.

	Binomial-HMM	Correlated Binomial
Motivation	Extends the probability-mixing model with dynamic weight re-assignment.	Treats neuronal attention as correlated Binomial variables.
Neuronal correlation	Described through the Markov chain.	Modeled directly by parameters.
Parameter dimension	$14 + 2\|\mathcal{M}\| + 7\|\mathcal{N}\|$	$13 + 2\|\mathcal{M}\|(T-1) + 7\|\mathcal{N}\|$
Meaning of C	Hidden state of the Markov chain, each state giving different stimulus weights.	State of neurons being either completely independent or fully positively correlated.

3 Results

We present here the parameter estimates and decoding for both models fitted to the spike train data. The models are fitted to each of the 48 sessions independently. For a discretization with T steps, we assign equal length, $400/T$ ms, to all time intervals. We use two different discretizations of $T = 3$ and 5, and two different classes of conditions with either all 12 or only 3 classes determined by whether there is a target in the stimulus pair, and in that case, whether it is contra- or ipsilateral (see Table 2.2).

3.1 Parameter estimation in Binomial-HMM

Figure 3.1 illustrates parameter estimates for the Binomial-HMM. In a and b we show results with discretization $T = 3$, and in c we also present results for $T = 5$.

Figure 3.1a shows the probability of attending to the stimulus at the contralateral side, $p_t = P(X_t = 1)$, for different conditions for each time step, as kernel density plots from all 48 estimates. Three line types (solid, dashed and dotted) indicate the three time steps, and four colors represent four types of conditions. At $t = 1$ all conditions follow the same distribution, so there is a single black curve. For the subsequent time steps, the condition types are: stimulus pairs with T on the ipsilateral side; stimulus pairs with T on the contralateral side; stimulus pairs with NONE on the ipsilateral side; and stimulus pairs with NONE on the contralateral side. The figure illustrates that neuronal attention slightly prefers the contralateral stimulus in the beginning right after stimulus onset (the black density curve is centered slightly towards larger values than 0.5), and later on tends to follow T and avoid NONE. Note that here we conduct model inference using all 12 conditions, and combine similar conditions together for presentation.

In Figure 3.1b, the estimates of the correlation ρ_t are plotted against the estimates $|p_t - 0.5|$ (difference of the probability of the contralateral stimulus from 0.5, or probability "extremeness") for each of the three time steps $t = 1, 2, 3$, on top of a two-dimensional kernel density estimate (bandwidth: 0.15) of the points as heatmaps. There are 48 estimates in the left panel at $t = 1$, and 48×12 estimates in the middle and right panels from 12 conditions in 48 sessions. At $t = 1$ before applying the TPM all conditions follow the same distribution. A straight line is plotted on the anti-diagonal line for easier reading. The lower left region of the heatmap represents a tendency of parallel processing, and all other regions represent a tendency of serial processing. In all panels, but most accentuated in the left panel, a big portion of the estimates fall in the lower left region, and at later times, the estimates tend to move to the other regions. This implies that, in an early stage stimuli tend to be processed in parallel. Later on more and more neurons share the same attended stimulus in the form of serial processing. There is evidence supporting both processing mechanisms for all time steps throughout the whole spike train. Moreover, we see that over time, the correlation tends to get smaller while the probability becomes more extreme.

In Figure 3.1c we investigate the asymptotic deviation statistic D^*. The average D^* is calculated over the 48 session estimates for each condition. The left panel shows the D^* values obtained from parameter estimates using all 12 conditions with $T = 3$. The middle and right panels show results for $T = 5$, the midle panel using the 3 classes of merged conditions, and the right panel using all 12 conditions. In all cases, D^* grows larger over time, implying stronger serial processing. Further, different settings of discretization and condition merging give different results. The differences caused by using $T = 5$ instead of $T = 3$ may be due to smaller sample sizes (shorter spike trains with only few spikes).

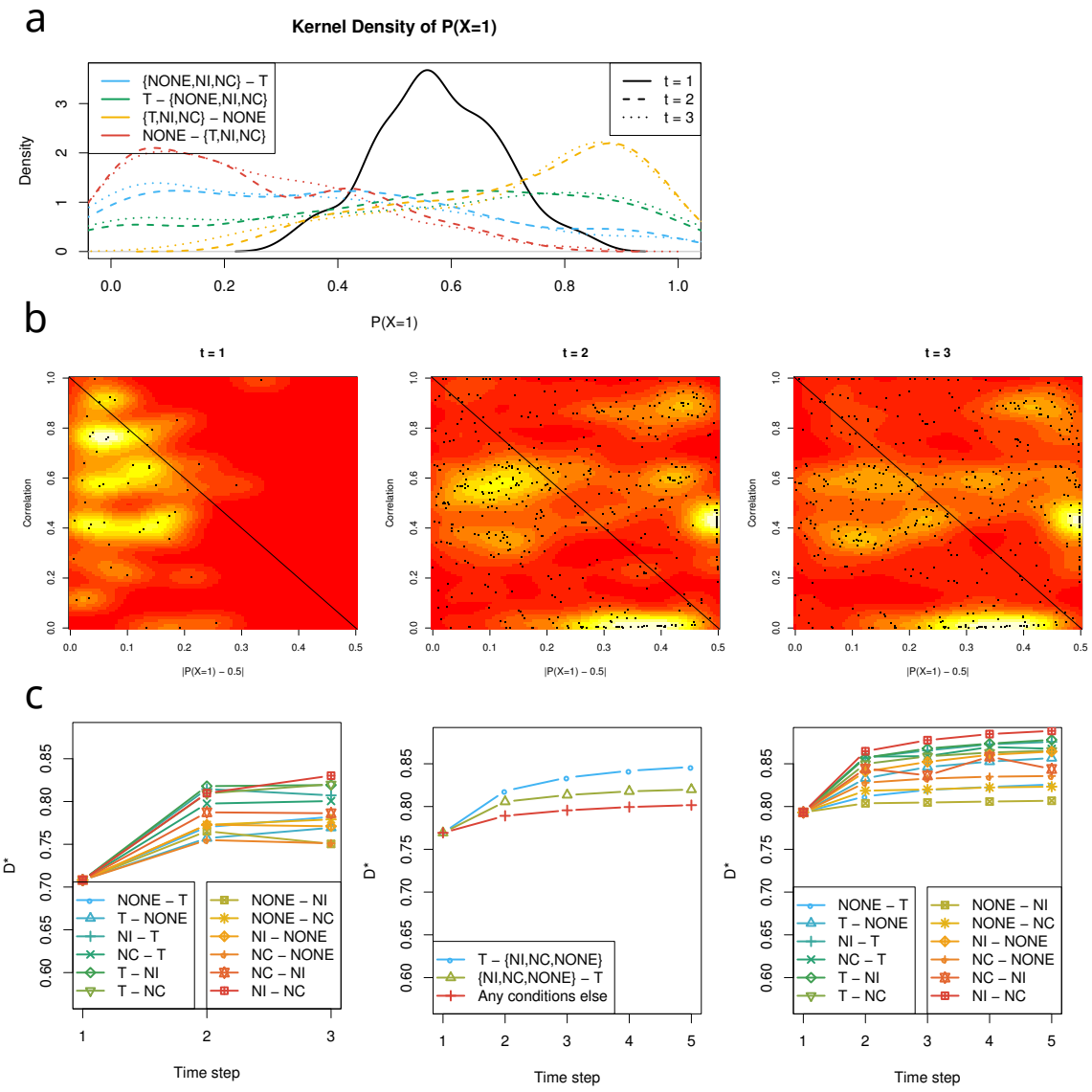


Figure 3.1: Results for the Binomial-HMM. Figures a and b are obtained using $T = 3$ and 12 conditions. Figure c uses also $T = 5$ and the 3 merged conditions. a) Kernel density estimation of the estimates of $P(X = 1)$, i.e. the probability of a neuron attending to the contralateral stimulus. b) Correlation estimates vs probability extremeness estimates at the different time steps, on top of a two-dimensional kernel density estimate as heatmaps. c) Estimates of D^* for $T = 3$ with all 12 conditions (left), $T = 5$ with the 3 merged conditions (middle), and $T = 5$ with all 12 conditions (right).

3.2 Parameter estimation in correlated Binomial

The estimates of the correlated Binomial model is shown in Figure 3.2. Figure explanations are as in Figure 3.1. We obtain similar results as for the Binomial-HMM. In Figure 3.2b, we see apparent parallel

processing at $t = 1$, while later on the correlation for most estimates goes to either 1 or 0, and the probability becomes more extreme.

The correlated Binomial model is essentially a mixture model of an independent and a fully correlated component. From figure b we find that at $t > 1$, in most of the 48×12 estimates the weight parameter of the mixture (i.e. the correlation coefficient) is very close to either 1 or 0, meaning one component is dominating over the other. This is because of the small number of simultaneously recorded neurons in most trials (see Figure 2.6), which is insufficient for obtaining good estimates in a mixture model. This is a weakness of the correlated Binomial model since it only contains two extreme components representing either full independence or full correlation. Model fitting of the correlated Binomial model on limited sample sizes can bias the correlation parameter. To check this presumption, we looked at the estimates from session "MN110411", the right-most neuron in the right panel of Figure 2.6 with the largest number of simultaneously recorded neurons, and found that the estimates of the correlation lie almost uniformly across 0 to 1, indicating that the estimates of either 0 or 1 of the correlation in other sessions can be an artefact of small sample sizes.

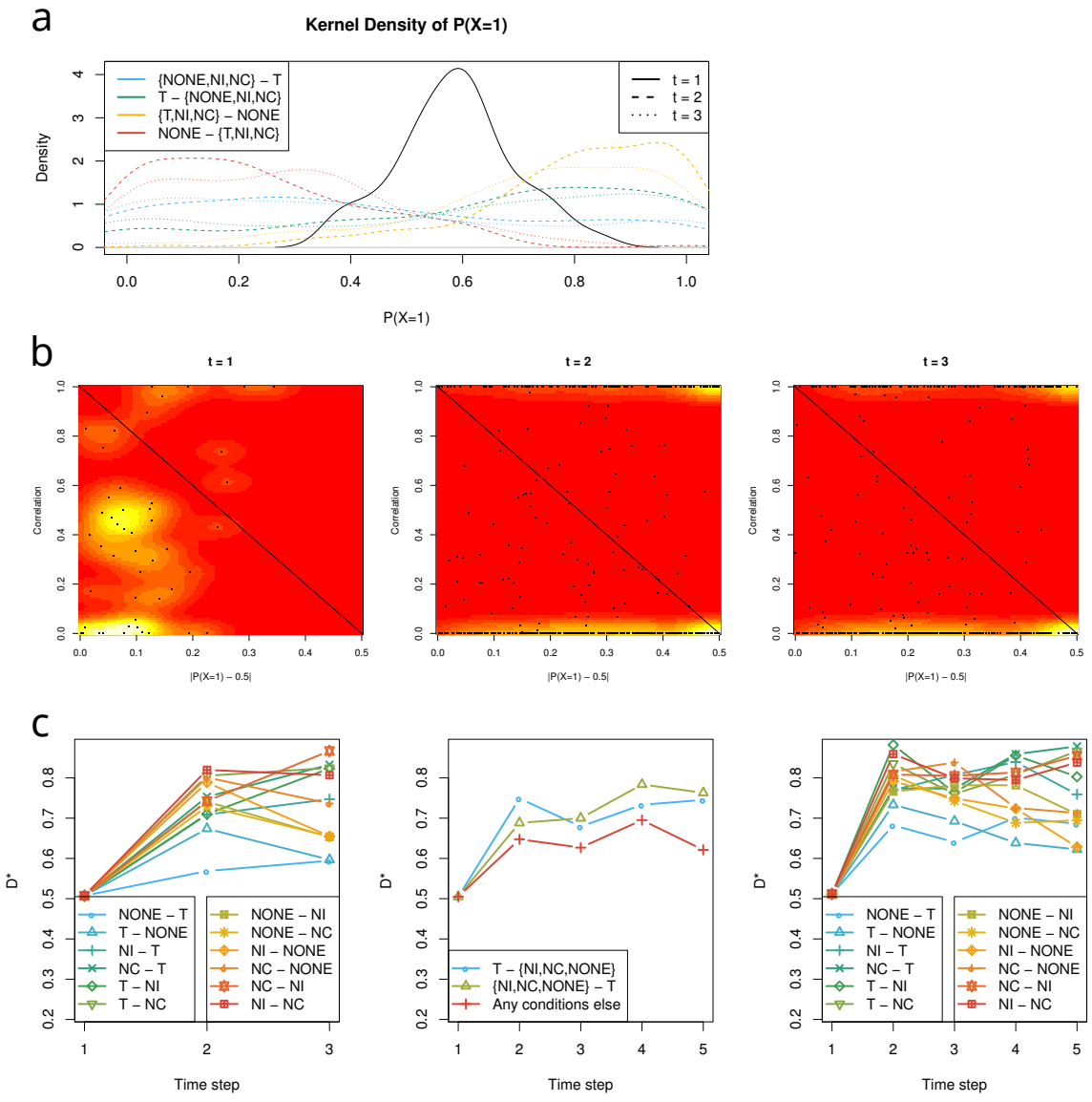


Figure 3.2: Results for the correlated Binomial model. See caption of Figure 3.1 for explanation.

3.3 Decoding

Here we decode the attended stimulus of each neuron conditional on the observed spike trains. The parameters used in the decoding algorithms are the estimated parameters obtained by MLE. In the Binomial-HMM model we show results using both $T = 3$ and $T = 5$, and in the correlated Binomial model using only $T = 3$.

Figure 3.3 shows the decoding of the attended stimulus for an example trial containing 10 simultaneously recorded spike trains in session "MN110411", condition NONE-T. The three rows show decoding of the same data set, with the upper panel using Binomial-HMM with $T = 3$, the middle panel using Binomial-HMM with $T = 5$ and the lower panel using the correlated Binomial model with $T = 3$. The left panel shows the data (10 simultaneously recorded spike trains in a trial) together with dashed lines indicating the discretization. In the middle panel, the table named "$P(X = 1|d)$" gives the posterior probability of each spike train attending the contra stimulus at each time step, with the dashed lines indicating the time steps corresponding to the left panels. Estimates in red color indicate higher probability of attending the contralateral stimulus and blue color indicates higher probability of attending the ipsilateral stimulus. Note that the target is located in the ipsilateral side. The table named "$P(C = c_1|d)$" gives the posterior probability of the hidden state being c_1. For the Binomial-HMM, the hidden state indicates the index of the Binomial component, and for the correlated Binomial model, the first hidden state is the independent Binomial component and the second is the fully correlated Bernoulli component. Since the hidden state means something different in the two models, the $P(C = c_1|d)$ values are different. The right panel shows the PMF of the number of neurons attending to the contralateral stimulus conditional on the spike train data for each time step, with the D_n values shown in the legend. These values are calculated by eq. (2.1) using the estimated $f(x)$ from the right panels.

In Figure 3.4 we combine all D_n values from all trials in all sessions, and plot the kernel density estimate for each time step (upper panel) as well as the evolution of the average of D_n over time (lower panel). In the upper panel, different line types indicates different time steps. If a trial has few simultaneously recorded spike trains, the D_n values will be biased. Thus, we only consider data with at least a certain minimum of simultaneously recorded spike trains. The minimum number is denoted by n in the top left legend. We have tried two options, $n = 2$ and $n = 5$, respectively. Also note that the minimum number n in a trial here is different from the number of simultaneously recorded neurons in a session, because in most trials not all simultaneously recorded neurons are used. We selected data such that the number of simultaneously recorded neurons in a session is at least 5, but in most trials the simultaneously recorded spike trains can be fewer. The D_n values estimated from trials with $n = 5$ (shown in red) are smaller than the values estimated from trial with $n = 2$ (shown in blue), which is expected because using a smaller number of spike trains when calculating D_n creates more bias towards overestimating D_n. On the other hand, using $n = 5$ yields less data than $n = 2$ and is less trustworthy. In the lower panel are shown the corresponding plots of average D_n over time for each decoding model for $n = 2$. Note that the similar plots in Figures 3.1c and 3.2c are *prior* measures based on estimated parameters, and the plots here in the lower panel are *posterior* measures based on the decoded attended stimulus. In all models and both choices of n, there is evidence of both parallel and serial processing at all time steps. Over time, D_n tends to be larger, meaning a larger degree of serial processing. Finally, as found previously, differences of D_n between certain conditions are often larger than differences between time steps.

4 Discussion

In this study we combine the point process neuron models describing spike trains with the neural interpretations of serial and parallel processing hypotheses in visual search. We propose a Binomial-HMM and a correlated Binomial model to describe neuronal attention in neurophysiological measurements from prefrontal cortex in rhesus monkeys. Results show that parallel processing is favored in some sessions while serial processing is favored in other sessions, and there is evidence for both parallel and serial processing at all discretized time steps. Considering the overall result, the D^* values suggest a tendency towards parallel processing in the early stage after stimulus onset, and serial processing in the late stage. This means that, right after stimulus onset, neurons tend to split to attend different stimuli, and later neurons become more synchronized sharing the same attended stimulus. Furthermore, at the early stage

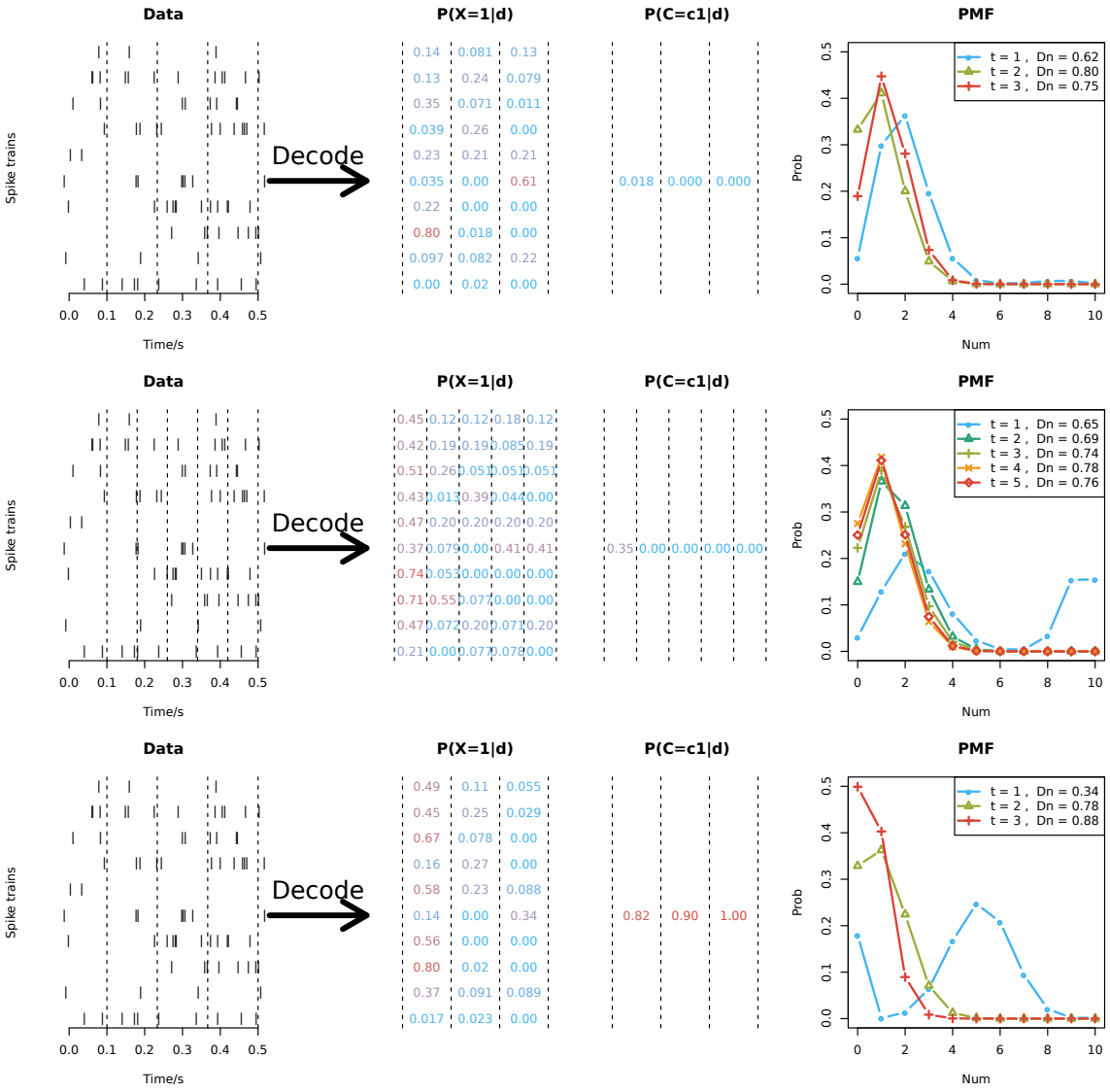


Figure 3.3: Decoding of an example trial using different models. Here all models use all 12 *conditions. The top panel shows the results of the Binomial-HMM model using* $T = 3$*. The middle panel shows the Binomial-HMM model with* $T = 5$*. The bottom panel shows the correlated model with* $T = 3$*. In the left are shown the simultaneously recorded spike trains from a trial in session "MN110411", and in the middle and right are shown the relevant results calculated from parameter estimates; see the text for details.*

neurons prefer the contralateral stimulus, while in the late stage neurons favor the T and avoid NONE, which agrees with the study conducted by averaging across spike trains (Kadohisa et al., 2013).

Decoding analysis provides posterior probabilities of neuronal attentions at each time step for each trial, yielding an estimate of the PMF and therefore also D_n. This can be used to analyze attentional behavior for any given simultaneously recorded spike trains in future trials. The conclusions regarding parallel and serial processing from the overall distribution of D_n on all trials and sessions from the decoding analysis are the same as in the prior analysis using only parameter estimates. Note that although both the prior and posterior analysis provide similar results, the conclusions regarding neuronal attentional properties should be drawn from the prior analysis based on the MLE. The MLE gives the optimal estimation of the neuronal properties based on all the available data. The decoding analysis, on the other hand, estimates what the neuron's attention could have been during a specific trial based on the data from this trial, and the uncertainty of the decoding is represented by posterior distributions.

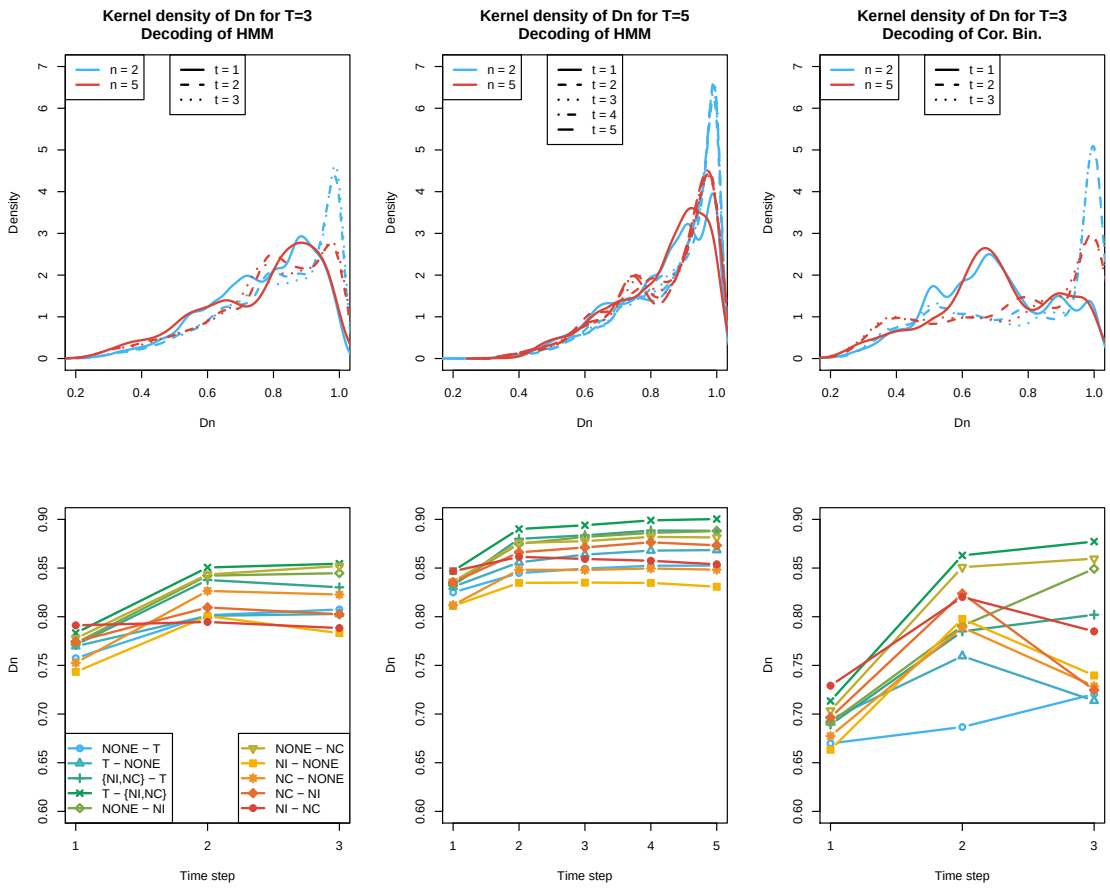


Figure 3.4: Decoding results of D_n over all trials in all sessions. The left and middle panels show results for the Binomial-HMM with $T = 3$ and $T = 5$, respectively. The right panel shows results for the correlated Binomial model with $T = 3$. The upper panel provides D_n averaged over all conditions of the whole data, and the lower panel separates between the 12 different conditions. See the text for details.

The article by Kadohisa et al. (2013) reported parallel processing in the early stage considering the whole brain including both hemispheres. The same conclusion is drawn from our analysis, where we find that the neurons prefer the contralateral stimulus in the early stage, and integrating both hemispheres gives simultaneous parallel processing. Furthermore, there exists not only such parallel processing considering the whole brain, but also parallel processing based on neurons in a single recording site, as supported by our finding. Though the simultaneously recorded neurons in one location show a tendency towards the contralateral stimulus in the early stage, there is strong evidence showing they split their attention between stimuli located on both sides in a parallel way.

The models here are fitted to the specific data set from Kadohisa et al. (2013) and the model structure contains the experimental conditions specific for this data set. However, with trivial adjustments, the models also apply to generic neurophysiologal data that consist of simultaneously recorded spike trains. Currently the models and methods only support two stimuli, and a future extension could be the generalization to an arbitrary number of stimuli.

The two models, the Binomial-HMM and the correlated Binomial model, give different results regarding the measurements of the degree of serial and parallel processing, both in parameter estimates and decoding analysis. This is partly because the two models are based on different assumptions. The biological reality of attention, which we try to describe with models, is complicated, and the two models approximate the reality and explain neural attention from different perspectives. Further, the experimental data are noisy with limited sample size and the models contain a large number of parameters, which leads to large variance of estimators. For one trial or session, the difference between the two models

could be large, but the overall results of the two models over a large number of sessions produce similar conclusions. It makes more sense to have comparisons under the same model. For example, we compare different conditions or different time steps only under the same model.

Another issue is the variability in results from different sessions for the same model. We assume the whole prefrontal area follow a probabilistic model and we want to estimate the model parameters. However, in each session we only have a small subset with 5 to 20 simultaneously recorded neurons from a recording site, and the number is even smaller for single trial (Figure 2.6), with each neuron having its distinct firing rate and attentional pattern (Figures 2.7 and 2.8). Thus, there is a large variance of the estimates from session to session, and we obtain the overall result by averaging and applying kernel density estimation. To obtain more stable and accurate results we will need to use a larger simultaneously recorded population of neurons.

References

Bricolo, E., Gianesini, T., Fanini, A., Bundesen, C., and Chelazzi, L. (2002). Serial attention mechanisms in visual search: A direct behavioral demonstration. *Journal of cognitive neuroscience*, 14(7):980–993.

Bundesen, C. (1990). A theory of visual attention. *Psychological review*, 97(4):523.

Bundesen, C. and Habekost, T. (2008). Principles of visual attention: Linking mind and brain.

Bundesen, C., Kyllingsbæk, S., and Larsen, A. (2003). Independent encoding of colors and shapes from two stimuli. *Psychonomic Bulletin & Review*, 10(2):474–479.

Daley, D. and Vere-Jones, D. (2003). An introduction to the theory of point processes, volume i: Elementary theory and methods of probability and its applications.

Diniz, C. A., Tutia, M. H., Leite, J. G., et al. (2010). Bayesian analysis of a correlated binomial model. *Brazilian Journal of Probability and Statistics*, 24(1):68–77.

Eriksen, C. W. and Lappin, J. S. (1965). Internal perceptual system noise and redundancy in simultaneous inputs in form identification. *Psychonomic Science*, 2(1-12):351–352.

Eriksen, C. W. and Spencer, T. (1969). Rate of information processing in visual perception: Some results and methodological considerations. *Journal of Experimental Psychology*, 79(2p2):1.

Hodges, J. L. and Le Cam, L. (1960). The poisson approximation to the poisson binomial distribution. *The Annals of Mathematical Statistics*, 31(3):737–740.

Hong, Y. (2013). On computing the distribution function for the poisson binomial distribution. *Computational Statistics & Data Analysis*, 59:41–51.

Kadohisa, M., Petrov, P., Stokes, M., Sigala, N., Buckley, M., Gaffan, D., Kusunoki, M., and Duncan, J. (2013). Dynamic construction of a coherent attentional state in a prefrontal cell population. *Neuron*, 80(1):235–246.

Kass, R. E., Eden, U. T., and Brown, E. N. (2014). *Analysis of neural data*. Springer.

Kyllingsbæk, S. and Bundesen, C. (2007). Parallel processing in a multifeature whole-report paradigm. *Journal of Experimental Psychology: Human Perception and Performance*, 33(1):64.

Li, K., Vozyrev, V., Kyllingsbæk, S., Treue, S., Ditlevsen, S., and Bundesen, C. (2016). Neurons in primate visual cortex alternate between responses to competing stimuli in their receptive field.

Luceño, A. (1995). A family of partially correlated poisson models for overdispersion. *Computational statistics & data analysis*, 20(5):511–520.

Nobre, K. and Kastner, S. (2013). *The Oxford handbook of attention*. Oxford University Press.

Schneider, W. and Shiffrin, R. M. (1977). Controlled and automatic human information processing: I. detection, search, and attention. *Psychological review*, 84(1):1.

Sternberg, S. (1966). High-speed scanning in human memory. *Science*, 153(3736):652–654.

Sternberg, S. (1969a). The discovery of processing stages: Extensions of donders' method. *Acta psychologica*, 30:276–315.

Sternberg, S. (1969b). Memory-scanning: Mental processes revealed by reaction-time experiments. *American scientist*, 57(4):421–457.

Townsend, J. T. and Ashby, F. G. (1983). *Stochastic modeling of elementary psychological processes*. CUP Archive.

Treisman, A. and Gormican, S. (1988). Feature analysis in early vision: evidence from search asymmetries. *Psychological review*, 95(1):15.

Treisman, A., Sykes, M., and Gelade, G. (1977). Selective attention and stimulus integration. *Attention and performance VI*, pages 333–361.

Truccolo, W., Eden, U. T., Fellows, M. R., Donoghue, J. P., and Brown, E. N. (2005). 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.

III Responses of Leaky Integrate-and-Fire Neurons to a Plurality of Stimuli in Their Receptive Fields

Published in The Journal of Mathematical Neuroscience, 6(1):1 (2016)
DOI: 10.1186/s13408-016-0040-2

Kang Li
Department of Mathematical Sciences, Department of Psychology
University of Copenhagen

Claus Bundesen
Department of Psychology
University of Copenhagen

Susanne Ditlevsen
Department of Mathematical Sciences
University of Copenhagen

The work based on Paper III and IV won a best poster award in the International Conference of Mathematical NeuroScience 2016.

Journal of Mathematical Neuroscience (2016) 6:8
DOI 10.1186/s13408-016-0040-2

The Journal of Mathematical Neuroscience
a SpringerOpen Journal

RESEARCH Open Access

Responses of Leaky Integrate-and-Fire Neurons to a Plurality of Stimuli in Their Receptive Fields

Kang Li[1,2] · Claus Bundesen[2] · Susanne Ditlevsen[1]

Received: 21 November 2015 / Accepted: 30 April 2016 / Published online: 23 May 2016

Abstract A fundamental question concerning the way the visual world is represented in our brain is how a cortical cell responds when its classical receptive field contains a plurality of stimuli. Two opposing models have been proposed. In the response-averaging model, the neuron responds with a weighted average of all individual stimuli. By contrast, in the probability-mixing model, the cell responds to a plurality of stimuli as if only one of the stimuli were present. Here we apply the probability-mixing and the response-averaging model to leaky integrate-and-fire neurons, to describe neuronal behavior based on observed spike trains. We first estimate the parameters of either model using numerical methods, and then test which model is most likely to have generated the observed data. Results show that the parameters can be successfully estimated and the two models are distinguishable using model selection.

Keywords Probability-mixing · Response-averaging · Parameter estimation · Model selection · Visual attention

Abbreviations

LIF	Leaky integrate-and-fire
PDE	Partial differential equation
IE	Integral equation
EM	Expectation-maximization
ISI	Interspike interval
MLE	Maximum likelihood estimation/estimator
PDF	Probability density function

S. Ditlevsen
susanne@math.ku.dk

1 Department of Mathematical Sciences, University of Copenhagen, Universitetsparken 5, Copenhagen, 2100, Denmark

2 Department of Psychology, University of Copenhagen, Øster Farimagsgade 2A, Copenhagen, 1353, Denmark

CDF Cumulative distribution function
QQ Quantile–quantile
KS Kolmogorov–Smirnov
DIC Deviance information criterion
AIC Akaike information criterion
BIC Bayesian information criterion

1 Introduction

The receptive field of a neuron in the visual system can be defined as the spatial area in which stimulation changes the firing pattern of the neuron. In primary visual cortex, receptive fields are small, with typical values of, for example, 0.5–2 deg of visual angle near the fovea. Moving up the hierarchy of extrastriate visual areas along either the dorsal [1] or the temporal [2] pathway, receptive field sizes grow substantially [3, 4], reaching, for example, a value of about 30 deg in the inferotemporal cortex. A plausible explanation is that since these areas process more complex aspects of the visual environment, information has to be integrated over larger spatial areas, such as when encoding faces [5] in the ventral pathway or optic flow patterns [6] in the dorsal one. Typically, receptive fields that are so big will contain a plurality of distinct stimulus objects rather than just a single stimulus object [7]. The way a cortical cell responds when its classical receptive field contains a plurality of stimuli is a basic question concerning the way the visual world is represented in our brain.

1.1 Probability-Mixing and Response-Averaging

In a pioneering study, Reynolds et al. [8] found that a typical cell in visual area V2 or V4 in monkeys responded to a pair of objects in its classical receptive field by adopting a rate of firing which, averaged across trials, equaled a weighted average of the responses to the individual objects when these were presented one at a time, with greater weight on an object the more attention was directed to the object. Reynolds et al. accounted for their data by proposing that on each individual trial, the firing rate of a cell to a plurality of stimulus objects equaled a weighted average of the firing rates to the individual objects when these were presented alone. Bundesen et al. [9, 10] proposed an alternative explanation of the data of Reynolds et al. by pointing out that the effects observed in firing rates that were averaged across trials could be explained by assuming that on each individual trial, when a plurality of objects were presented, the cell responded as if just one of the objects was presented alone, so that across trials, the response of the cell was a probability mixture of the responses to the individual objects when these were presented alone.

In the *response-averaging* model proposed by Reynolds et al. [8] (see also [11–18]), the neuron responds with a weighted average of the responses to single stimuli. By contrast, in the *probability-mixing* model proposed by Bundesen et al. [9], the neuron responds at any given time to only one of the single stimuli with certain probabilities. Suppose the stimulus $S(t)$ presented to the neuron consists of K separated single stimuli, denoted by $S_1(t), \ldots, S_K(t)$. In the response-averaging model, the

neuron responds with a weighted average of responses to single stimuli, $\sum_k \beta_k I_k(t)$, with β_k being the weights, and $\sum_k \beta_k = 1$. Here $I_k(t)$ denotes the effects that S_k has on the spiking neuron model, which we set to be the stimulus current. In the probability-mixing model, the response of the neuron equals one of the responses the neuron would have had if only a single stimulus was presented according to a probability mixture with probabilities $\alpha_1, \ldots, \alpha_K$, and $\sum_k \alpha_k = 1$.

In our previous study [19], we compared the abilities of the probability-mixing model and the response-averaging model to account for spike trains (i.e., times of action potentials obtained from extracellular recordings) recorded from single cells in the middle temporal visual area (MT) of rhesus monkeys. Point processes were employed to model the spike trains. Results supported the probability-mixing model.

In this article, we combine the probability-mixing and the response-averaging model with the leaky integrate-and-fire (LIF) model, to describe neuronal behavior based on observed spike trains. This is cast in a general setting, where the stimulus $S(t)$ is represented as an input current to the neuron. The spike train data are simulated using the LIF model, responding either to a single stimulus or to a stimulus pair. In the case of stimulus pair, both response averaging and probability mixing are used. The first goal of the paper is to estimate parameters of either of the two models from spike train data. The second goal is to test which of the two models are most likely to have generated the observed data.

1.2 The Leaky Integrate-and-Fire Model

The LIF models have been extensively applied to model the membrane potential evolution in single neurons in computational neuroscience (for reviews, see [20, 21]). The model has some biophysical realism while still maintaining mathematical simplicity. The simplest LIF model is an Ornstein–Uhlenbeck (OU) process with constant conductance, leak potential, and diffusion coefficient. More biophysical realism can be obtained by allowing for post-spike currents generated by past spikes [22]. Here we use post-spike currents generated via three types of kernels [23, 24]: bursting, decaying, and delaying kernel, all modeled by the difference between two decaying exponentials, but any kernel could be used.

1.3 Temporal Stimulus

Constant stimuli are simple to handle and are widely used in both experiments and modeling work. However, real world stimuli are generally time varying. If they for example contain oscillatory components, the generated spike trains might also contain oscillations in the firing rates. Here we use three types of stimuli: oscillatory stimuli described by sinusoidal functions, pulsing stimuli modeled by piecewise constant functions, and stochastic stimuli described by OU processes.

1.4 Method Summary

We combine the models describing neuronal response to a plurality of stimuli, namely the probability-mixing model and the response-averaging model, with the LIF framework, for different types of stimuli and response kernels. Parameter estimation is

done by maximum likelihood using first-passage time probabilities of diffusion processes [25]. We solve the first-passage time problem by numerically solving either a partial differential equation (PDE), the Fokker–Planck equation, or an integral equation (IE), the Volterra integral equation. Numerical solutions of these equations have been extensively explored and applied in the computations of neuronal spike trains [26–28]. Inspired by these previous studies, we apply four numerical methods, including two Fokker–Planck related PDEs and two kinds of Volterra IEs, and compare the performance of the four methods. We also describe and compare two alternative methods for maximizing the likelihood function of the probability-mixing model, which are direct maximization of the marginal likelihood and the expectation–maximization (EM) algorithm. Finally, we show that the probability-mixing model and the response-averaging model can be distinguished in the LIF framework, by comparing parameter estimates and through uniform residual tests.

2 Leaky Integrate-and-Fire Model with Stimuli Mixtures

The evolution of the membrane potential is described by the solution to the following stochastic differential equation:

$$\begin{aligned} dX(t) &= b\big(X(t),t\big)\,dt + \sigma\,dW(t) \\ &= \big(-\gamma\big(X(t)-\mu\big) + I(t) + H(t)\big)\,dt + \sigma\,dW(t), \\ X(0) &= x_0; \qquad X\big(t_j^+\big) = x_0, \\ t_j &= \inf\big\{t > t_{j-1} : X(t) = x_{\text{th}}\big\} \quad \text{for } j \geq 1, t_0 = 0, \end{aligned} \tag{1}$$

where t_j^+ denotes the right limit taken at t_j. The drift term $b(\cdot)$ contains three currents: the leak current $-\gamma(X(t)-\mu)$, where γ is the decay rate and μ is the reversal potential, the stimulus-driven current $I(t)$, and the post-spike current $H(t)$. The potential $X(t)$ evolves until it reaches the threshold, x_{th}, where it resets to x_0. Since the membrane potential $X(t)$ is not observed, but only the spike times $d = (t_1, t_2, \ldots)$, we can use any values for threshold and reset suitable for the numerical calculation. The noise is described by the standard Wiener process, $W(t)$, and the diffusion parameter, σ. The interspike intervals (ISIs) are defined by $t_{j+1} - t_j$.

The stimulus current $I(t)$ is shaped from the external stimulus current through a stimulus kernel $k_s(t)$ as $I(t) = \int_{-\infty}^{t} k_s(t-s)S(s)\,ds$, where $S(s)$ denotes the external current at time s. Similarly, the post-spike current arises from past spikes through a response kernel $k_h(t)$ by $H(t) = \int_{-\infty}^{t} k_h(t-s)\mathbb{I}(s)\,ds$. Here $\mathbb{I}(s) = \sum_{\tau\in d}\delta(s-\tau)$ describes the spike train, where $\delta(\cdot)$ denotes the Dirac delta function.

In this work, the stimulus kernel is assumed without memory, such that $k_s(t) = \delta(t)$. Then the stimulus current $I(t)$ is completely determined by the stimulus at time t, e.g., $I(t) = S(t)$. The response kernel is assumed to be the difference of two exponentials decaying over time,

$$k_h(t) = \eta_1 e^{-\eta_2 t} - \eta_3 e^{-\eta_4 t} \tag{2}$$

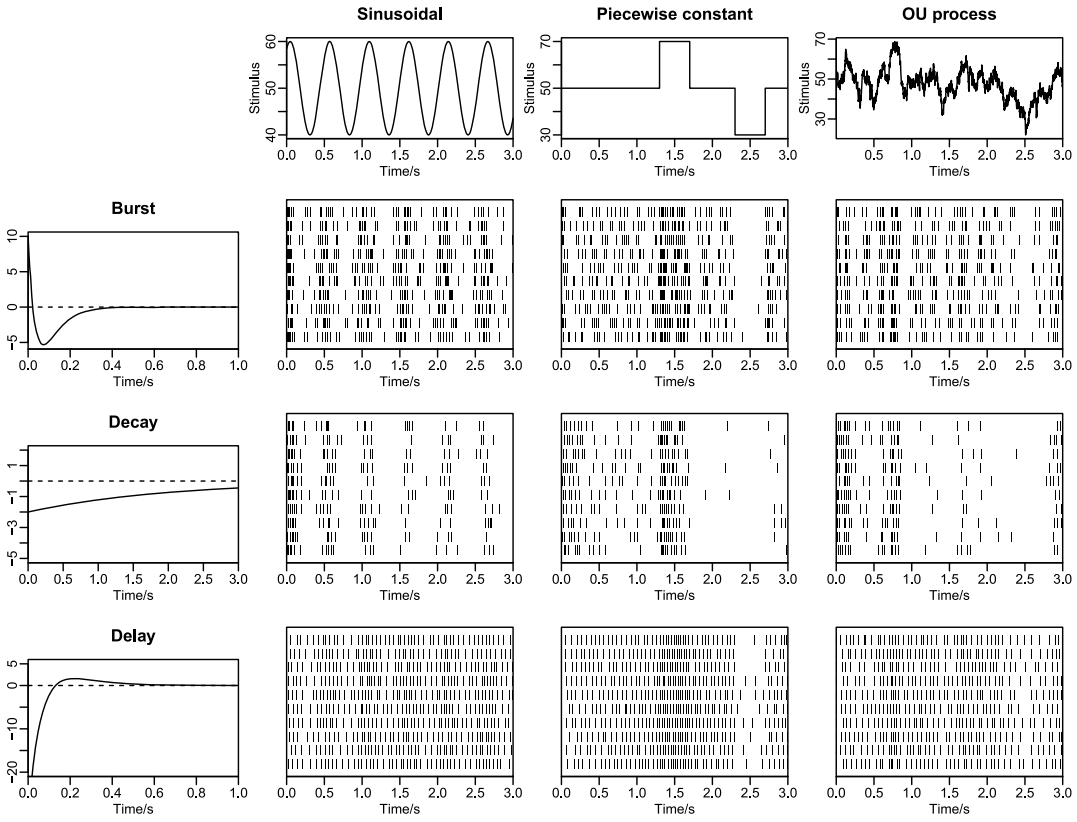


Fig. 1 Realization of spike trains for different combinations of response kernels and stimuli. *Top panels* show the three stimulus types; sinusoidal, piecewise constant and Ornstein–Uhlenbeck process. *Left panels* show the burst, decay and delay response kernels. The *nine middle panels* illustrate spike train patterns for the different combinations of response kernels and stimuli. The patterns produced by each response kernel are apparent; the bursts of spikes for the burst kernel, the firing rate adaptation of the decay kernel, and the refractory period by the delay kernel (no short ISIs). Likewise, the patterns produced by each stimulus are apparent; periodicity by the sinusoidal, abrupt changing intensities by the piecewise constant, and slowly fluctuating changes in intensity by the random stimulus

with four positive parameters, $\eta = (\eta_1, \eta_2, \eta_3, \eta_4)$. By adjusting the parameters, different kernels are obtained. Note that in practice the four parameters are not identifiable, because different parameter sets can result in very similar kernels. Therefore, when we later verify parameter estimates we will not check each individual estimate, but only plot the estimated shape of the kernel function, which is the quantity of interest.

Three types of kernels are used, shown in the left panels of Fig. 1. The *bursting* kernel is characterized by being positive in the beginning, then turning negative, and finally converging toward 0, which happens when $\eta_1 > \eta_3$ and $\eta_2 > \eta_4$. It follows that the most recent spikes have excitatory effects for the current spike probability, but the accumulation of past spikes has inhibitory effects, resulting in rhythmic spiking with bursts. The *decaying* kernel only has one negative exponential by setting $\eta_1 = 0$. The parameters η_3 and η_4 are small such that the inhibitory effects are small but long-lasting, making the firing rate decay slowly over time. The *delaying* kernel has parameters $\eta_1 < \eta_3$ and $\eta_2 < \eta_4$. It is negative in the beginning, then turns positive, and finally converges to 0. The most recent spikes have inhibitory effects,

neutralized later on by the accumulation of excitatory effects, resulting in delaying the immediate formation of a new spike after a spike, preventing short ISIs, which models the refractory period. In the center panels example spike trains for the different kernels and different stimuli are illustrated.

2.1 Current from Stimulus Mixture

Suppose that inside the receptive field of the neuron there are at least two separated non-overlapping stimuli, which we will call a stimulus mixture. According to the probability-mixing model [9], the neuron responds to only one stimulus at any given time with certain probabilities. Thus, for a total of K stimuli, the stimulus-driven current, $I(t)$, follows a probability mixture:

$$I(t) = S_k(t), \quad \text{with probability } \alpha_k \tag{3}$$

for $k = 1, \ldots, K$ and $\sum_{k=1}^{K} \alpha_k = 1$. Recall that the stimulus kernel $k_s(t) = \delta(t)$ and thus, the current caused by the kth stimulus $I_k(t) = S_k(t)$. According to the response-averaging model [11], the current is a weighted average of all stimuli currents:

$$I(t) = \sum_{k=1}^{K} \beta_k S_k(t). \tag{4}$$

The leak current and the spike response current do not depend on the stimuli.

In the top panels of Fig. 1 three types of stimuli are illustrated. A *sinusoidal* stimulus is defined by

$$S(t) = s_1 \sin(s_2 t + s_3) + s_4 \tag{5}$$

with four parameters $s_{\sin} = (s_1, s_2, s_3, s_4)$ describing the stimulus. Note that it also covers a constant stimulus for $s_1 = 0$. A *piecewise constant* stimulus is defined by

$$S(t) = \begin{cases} s_1, & t_1 \le t < t_2, \\ s_2, & t_2 \le t < t_3, \\ \ldots, & \\ s_n, & t_n \le t < t_{n+1}, \end{cases} \tag{6}$$

with parameters $s_{pw} = (s_1, s_2, \ldots, s_n, t_1, t_2, \ldots, t_{n+1})$. A *stochastic* stimulus is given by an OU process described by the SDE:

$$dS(t) = \big(s_1 - S(t)\big)\, dt + s_2\, dW(t) \tag{7}$$

with two parameters $s_{\mathrm{OU}} = (s_1, s_2)$. We assume throughout that the stimuli currents are known. Spike patterns from combinations of different types of stimuli and response kernels are shown in Fig. 1. Clear bursting, decaying and delaying effects can be seen.

Two example spiking patterns together with their voltage traces generated from either a sinusoidal or a constant stimulus together with a bursting post-spike kernel are shown in Fig. 2. There are bursts of spikes occasionally even under constant

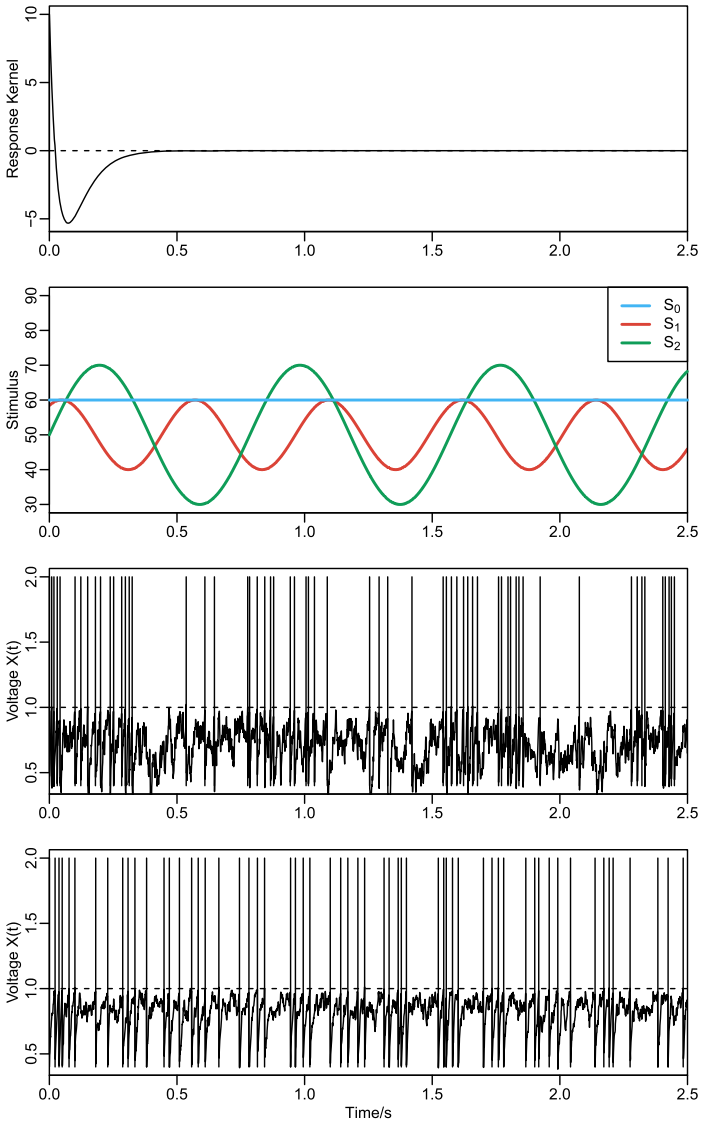


Fig. 2 Illustration of voltage traces resulting from a bursting response kernel and sinusoidal stimuli. **(a)** Bursting response kernel in Eq. (2) with parameters $\eta = (50, 25, 40, 15)$. **(b)** Examples of sinusoidal stimuli in Eq. (5). *Blue*: constant with $s_0 = (0, \cdot, \cdot, 60)$. *Red*: $s_1 = (10, 12, 1, 50)$. *Green*: $s_2 = (20, 8, 0, 50)$. **(c)** An example realization of membrane potential evolution, Eq. (1), responding to the sinusoidal signal s_1, and **(d)** responding to the constant signal s_0

stimulus caused by the bursting response kernel. A sinusoidal stimulus causes long bursts, and in addition, the bursting kernel causes a clear separation of small burst periods also within the long bursting period.

3 Maximum Likelihood Estimation Using First-Passage Time Probabilities

Our objective here is to estimate the parameters μ and σ from (1), the response kernel function k_h in (2) represented by the parameter vector η, and either the probability vector of the stimuli in the mixture, $\alpha = (\alpha_1, \ldots, \alpha_K)$, under the probability-mixing model, or the vector of weights in the average, $\beta = (\beta_1, \ldots, \beta_K)$, in the response-averaging model. The estimation of the decay rate γ is difficult when there is no access to the membrane potential, but only spike times are observable, as discussed in [29, 30]. We therefore assume γ is known. The vector of all parameters in the model is

thus θ, where $\theta = (\mu, \sigma, \eta, \alpha)$ in the probability-mixing model, and $\theta = (\mu, \sigma, \eta, \beta)$ in the response-averaging model. The stimulus is assumed known and the stimulus parameter vector s is therefore not estimated.

A similar LIF model with different stimulus and response kernels on single piecewise constant stimuli was used in Paninski et al. [24]. They showed that parameters can be estimated using MLE by solving the Fokker–Planck equation, covering also discussion of non-white noise and interneuronal interactions. The model was later applied to experimental data collected from retina of macaque monkeys [31]. Here we estimate parameters in the LIF model for various temporal stimuli and different response kernels, using four different numerical methods to calculate the likelihood function, within the framework of either the probability-mixing or the response-averaging model.

Suppose we observe N spike trains, $D = (d_1, \ldots, d_N)$, all responding to the same stimulus mixture, where the ith spike train consists of N_i spike times, $d_i = (t_1^i, \ldots, t_{N_i}^i)$. The jth ISI of the ith spike train is then given by $t_{j+1}^i - t_j^i$. Assume that each measured spike train, i.e., each trial, is sufficiently short, such that, under the probability-mixing model, the neuron is only responding to one stimulus within the stimulus mixture, not switching the response within the trial.

3.1 First-Passage Times and Probability Distributions

Modeling the spike train data as threshold crossings of the underlying diffusion process representing the unobserved membrane potential belongs to the so-called first-passage time problem [32, 33]. For models with no effects from past spikes, such that ISIs are assumed i.i.d., one approach is to build loss functions using the Fortet equation [29, 30]; see also [34]. A more general method, which allows for the post-spike effects in model (1), is to use maximum likelihood estimation (MLE) from numerical solutions of PDEs or IEs for the conditional distribution of the spike times or equivalently, the ISIs.

We use the following notation for the probability density functions (PDFs) and cumulative distribution functions (CDFs) of interest:

$$
\begin{aligned}
f\big(x, t|\mathcal{H}_t, \theta, S(t)\big) &\quad \text{(time-evolving PDF of the membrane potential)},\\
F\big(x, t|\mathcal{H}_t, \theta, S(t)\big) &\quad \text{(time-evolving CDF of the membrane potential)},\\
g\big(t|\mathcal{H}_t, \theta, S(t)\big) &\quad \text{(PDF of the spike time)},\\
G\big(t|\mathcal{H}_t, \theta, S(t)\big) &\quad \text{(CDF of the spike time)}.
\end{aligned}
$$

All the above distributions depend on the spike history up to time t, denoted by $\mathcal{H}_t$, the parameter vector θ and the stimulus $S(t)$. In the following, we sometimes suppress these dependencies in the notation for readability. We write $g_k(t;\theta) = g(t|\mathcal{H}_t, \theta, S_k(t))$ for the probability density of the spike time when the neuron is only presented with the single stimulus k.

The probability that the neuron has not yet fired at time t, $1 - G(t)$, is equal to the probability that the membrane potential has not yet reached x_{th}, $F(x_{\text{th}}, t)$. Thus, the

probability density of a spike time is given by [24, 27, 35]

$$g(t) = -\frac{\partial}{\partial t} F(x_{\text{th}}, t) = -\frac{\partial}{\partial t} \int_{-\infty}^{x_{\text{th}}} f\left(x', t\right) dx'. \tag{8}$$

The solution of the Fokker–Planck equation provides $f(x,t)$ and $F(x,t)$, and therefore also $g(t)$. The solution of the Volterra integral equation directly provides $g(t)$ [36]. Calculating $g(t)$ enables us to do MLE, as explained in Sects. 3.5 and 3.6 below.

3.2 Fokker–Planck Equation

The PDF of X_t in Eq. (1) with a resetting threshold, $f(x,t)$, solves the Fokker–Planck equation, defined by the following PDE [21, 27, 33]:

$$\partial_t f(x,t) = -\partial_x \left(b(x,t) f(x,t)\right) + \frac{\sigma^2}{2} \partial_{xx}^2 f(x,t), \tag{9}$$

with absorbing boundary condition $f(x_{\text{th}}, t) = 0$ and initial condition $f(x,0) = \delta(x - x_0)$. To solve the equation numerically we also impose a reflecting boundary condition at a small value $x = x^-$, where the flux equals 0: $J(x^-, t) = -b(x^-, t) f(x^-, t) + \sigma^2 \partial_x f(x^-, t)/2 = 0$. We call this method the Fokker–Planck PDF method.

Another approach is to formulate the PDE for the CDF, i.e., $F(x,t)$ [27, 35] (see Appendix A.2):

$$\partial_t F(x,t) = -b(x,t) \partial_x F(x,t) + \frac{\sigma^2}{2} \partial_{xx}^2 F(x,t), \tag{10}$$

with equivalent boundary conditions: $\partial_x F(x_{\text{th}}, t) = 0$, $F(x^-, t) = 0$, and initial condition: $F(x,0) = H(x - x_0)$, where $H(\cdot)$ is the Heaviside step function. This is then called the Fokker–Planck CDF method.

Both PDEs are solved numerically using the Crank–Nicholson finite difference method, together with the Thomas algorithm efficiently solving tridiagonal systems [37]. Whichever method we use, we can always obtain the PDF (CDF) from the CDF (PDF) by numerical differentiation (integration).

3.3 Volterra Integral Equation

The first-kind Volterra IE (Fortet equation) combines the first-passage time PDF $g(t)$ with the threshold-free membrane potential PDF $f^*(x, t|v, s)$ using the law of total probability [29, 30]:

$$f^*(x_{\text{th}}, t|x_0, 0) = \int_0^t f^*(x_{\text{th}}, t|x_{\text{th}}, s) g(s)\, ds. \tag{11}$$

For the OU model (1), the threshold-free PDF $f^*(x, t|v, s)$ is Gaussian [33, 38]:

$$f^*(x, t|v, s) = \frac{1}{\sqrt{2\pi V(t|s)}} \exp\left\{-\frac{(x - M(t|v,s))^2}{2V(t|s)}\right\}, \tag{12}$$

with mean

$$M(t|v,s) = ve^{-\gamma(t-s)} + \int_s^t I_{\text{total}}(u)e^{-\gamma(t-u)}\,du \tag{13}$$

and variance

$$V(t|s) = \frac{\sigma^2}{2\gamma}\big(1 - e^{-2\gamma(t-s)}\big). \tag{14}$$

The total current is denoted by $I_{\text{total}}(t) = \gamma\mu + I(t) + H(t)$.

The initial condition for the IE is $g(0) = 0$. Using this, we can solve the equation recursively and obtain $g(t)$.

The second-kind Volterra IE is defined by [39]

$$g(t) = -2\psi(x_{\text{th}}, t|x_0, 0) + 2\int_0^t \psi(x_{\text{th}}, t|x_{\text{th}}, s)g(s)\,ds, \tag{15}$$

where

$$\begin{aligned}\psi(x,t|v,s) &= \partial_t \int_{-\infty}^{x} f^*\big(x', t|v, s\big)\,dx' \\ &= f^*(x,t|v,s)\left[\gamma x - I_{\text{total}}(t) - \frac{\sigma^2}{2V(t|s)}\big(x - M(t|v,s)\big)\right].\end{aligned} \tag{16}$$

A modification of $\psi(x,t|v,s)$ is proposed to avoid a singularity when $t \to s$ [36, 39] (see Appendix A.3):

$$\phi(x,t|v,s) = \frac{1}{2}f^*(x,t|v,s)\left[\gamma x - I_{\text{total}}(t) - \frac{\sigma^2}{V(t|s)}\big(x - M(t|v,s)\big)\right]. \tag{17}$$

The second Volterra IE can also be solved numerically. It requires more computation time than the first-kind, but has higher accuracy.

3.4 Computational Time Complexity

For both the Fokker–Planck PDE and the Volterra IE methods, the time complexity is directly related to the grid size for the numerical solution. Specifically, suppose that the grid size of the time discretization is n and the size of the space discretization is m. Then the Fokker–Planck method has complexity on the order of $O(mn)$ and the Volterra method is on the order of $O(n^2)$ (native implementation requires $O(n^3)$, but techniques are applied to reduce the complexity to $O(n^2)$; see [36]). Furthermore, the computation is largely affected by the response kernel used. A discretization is applied to approximate the nonlinear kernel by a piecewise constant function with sufficiently small segmentation length. The values of the constant segments are calculated and stored in a data vector when the parameters are updated. Then inside an optimization loop, the kernel function is evaluated by referring to this data vector.

3.5 Marginal Likelihood of the Probability-Mixing Model

Under the probability-mixing model, the marginal likelihood function of the ith spike train $d_i = (t_1^i, \ldots, t_{N_i}^i)$ for a mixture of K stimuli is given by

$$L(\theta; d_i) = \sum_{k=1}^{K} \alpha_k \prod_{j=1}^{N_i} g_k\big(t_j^i; \theta\big), \tag{18}$$

and thus the marginal log-likelihood of all N spike trains $D = (d_1, \ldots, d_N)$ is

$$\ell(\theta; D) = \sum_{i=1}^{N} \log\left(\sum_{k=1}^{K} \alpha_k \prod_{j=1}^{N_i} g_k\big(t_j^i; \theta\big) \right). \tag{19}$$

Marginal refers to the observed data D; see Sect. 3.5.1 below for a definition of the full data. MLEs are then obtained by maximizing (19). The log-likelihood function consists of logarithms of sums, and the calculations are prone to encounter numerical over- or underflow issues. To overcome this, we apply the log-sum-exp formula [37].

3.5.1 Optimizing the Likelihood Using the Expectation-Maximization Algorithm

As an alternative to optimizing directly the log-likelihood function (19), the EM algorithm [40] is well suited to solve optimization problems for mixture models and is simple to implement. The EM algorithm treats the unknown stimulus mixture component which the neuron responds to as unobserved data, or latent variables. We write $Y = (y_1, \ldots, y_N)$ where $y_i \in \{1, 2, \ldots, K\}$, for the latent variables indicating which single stimulus each spike train is responding to. The full data then include both the observed spike trains D and the unobserved stimuli response Y.

The EM algorithm is an iterative procedure. In each iteration, the expectation of the full data log-likelihood conditional on the parameters from the previous iteration, is maximized to obtain the optimal parameters for the current iteration. The algorithm runs until convergence, i.e., the difference of parameter estimates is sufficiently small between two adjacent iterations. We use the notation θ for the current parameter to estimate, and θ_{-1} for the parameter estimated in the previous iteration, and likewise for the components of the probability vector α, i.e., α_k and $(\alpha_k)_{-1}$.

In each iteration, the conditional expectation of the full data log-likelihood is (see Appendix A.1 for the derivation),

$$\begin{aligned} Q(\theta|\theta_{-1}) &= \mathbb{E}\big[\log L_c(\theta; D, Y)|\theta_{-1}, D\big] \\ &= \sum_{i=1}^{N} \left[\sum_{k=1}^{K} P(y_i = k|\theta_{-1}, d_i) \left(\log \alpha_{y_i} + \sum_{j=1}^{N_i} \log g\big(t_j^i|y_i, \theta\big) \right) \right], \end{aligned} \tag{20}$$

where the conditional probability is obtained using the Bayes formula:

$$P(y_i = k|\theta_{-1}, d_i) = \frac{(\alpha_k)_{-1} \prod_{j=1}^{N_i} g(t_j^i|y_i = k, \theta_{-1})}{\sum_{l=1}^{K} (\alpha_l)_{-1} \prod_{j=1}^{N_i} g(t_j^i|y_i = l, \theta_{-1})}. \tag{21}$$

The EM algorithm requires the calculation of the likelihood of the spike train for all components in the mixture. Thus, the EM algorithm has (approximately) the same time complexity regarding the number of evaluations of density functions as the calculation of the marginal likelihood.

3.6 Likelihood of the Response-Averaging Model

In the response-averaging model, the neuron responds to a weighted average of stimuli, and the model does not follow a probability mixture. The likelihood is given by

$$L(\theta; D) = \prod_{i=1}^{N} \prod_{j=1}^{N_i} g\left(t_j^i; \theta\right), \tag{22}$$

where $g(t)$ is now the probability density of spiking at time t when the neuron is responding to a weighted average of all K stimuli, $\sum_{k=1}^{K} \beta_k S_k(t)$.

3.7 Model Checking: Uniformity Test

The goodness-of-fit can be verified by uniformity tests using the CDF $G(t)$ for all spike times in D. If the model perfectly describes the data, then the residuals

$$z_j^i = G\left(t_j^i\right) \tag{23}$$

follow a standard uniform distribution, $z_j^i \sim \mathrm{U}(0, 1)$. We then merge all the residuals for a specific model, and test the residuals against the uniform distribution. Quantile–quantile (QQ) plots and the Kolmogorov–Smirnov (KS) test can be employed to check for uniformity.

4 Simulation Study

To illustrate the approach, we first detail the simulation study of the bursting kernel and the sinusoidal stimulus. Then results using the other types of kernels and stimuli are briefly illustrated and summarized.

Traces from model (1) using the bursting response kernel shown in Fig. 2(a), and one of the two sinusoidal stimuli shown in Fig. 2(b) or a mixture thereof was simulated according to the Euler–Maruyama scheme with a time step size of 0.1 ms. The process was run until reaching the threshold x_{th} where the time was recorded. The process was then reset to x_0 and started again, while the stimulus continued without any interruption, and the previously recorded spike times entered in the calculation of the post-spike currents. This was continued until the spike train was 4 s long, containing around 60 to 70 spikes. Table 1 shows the values of the parameters used for simulation and numerical computation.

Parameter estimation was split in two, in agreement with how a typical experiment would be conducted. First we simulated spike trains responding to single stimuli. Note that in this case the probability-mixing and the response-averaging models are

Table 1 Parameter values used in the simulation study

Category	Parameter	Value	Explanation
Sinusoidal stimulus	s_1	(10, 12, 1, 50)	First stimulus
	s_2	(20, 8, 0, 50)	Second stimulus
Unknowns to estimate	η	(50, 25, 40, 15)	Bursting response kernel
	α	(0.4, 0.6)	Probability mixing
	β	(0.4, 0.6)	Response averaging
	μ	0.5	Reversal potential
	σ	1	Diffusion parameter
Numerical computation	Δt	0.002	Time discretization
	Δx	0.02	Space discretization
	x^-	0	Lower reflecting boundary
Neuronal characteristics	x_0	0.4	Reset potential
	x_{th}	1	Spike threshold
	γ	100	Decay rate

the same, and $\alpha = \beta = 1$ are one-dimensional. The data set contains 10 spike trains, with five attending the first single stimulus and the other five attending the second single stimulus. Using this data set, we estimated parameters of the response kernel, η, and parameters of the diffusion model, μ and σ.

Second, we simulated spike trains using a mixture of the two sinusoidal stimuli. Two data sets were simulated, one data set consisting of 10 spike trains following the probability-mixing model, and another data set consisting of 10 spike trains following the response-averaging model. To check if the two models could be distinguished, we fitted the data using the probability-mixing model and the response-averaging model on both data sets, resulting in four combinations. During this stage, we fixed the response kernel parameters η to values estimated in the first step, and estimated again μ, σ, as well as α or β, depending on the model. There are therefore two sets of estimates of μ and σ for each trial. The purpose is threefold; first of all, these parameters might slightly drift in a real neuron when changing the stimulus (even if we do not change them in the simulation); second, it is of interest to understand the statistical accuracy and uncertainty of these parameter estimates when inferred in the two experimental settings; and third, comparing estimates from both single stimulus and stimulus mixtures can serve as model control, as explained below. When fitting the probability-mixing model on the data generated from this same model, we used both the marginal MLE and the EM algorithm. The above simulation and estimation procedure was repeated 100 times, generating 100 sets of estimates.

The simulation study serves different purposes. First, the four numerical methods to obtain the PDFs of the spike times, namely the first Volterra, second Volterra, Fokker–Planck PDF, and Fokker–Planck CDF, should be evaluated and compared. This is done on single stimulus spike train data. Second, the quality of the parameter estimates should be assessed, as well as how important it is to use the correct model for the estimation. This is conducted using spike trains simulated from stimulus mix-

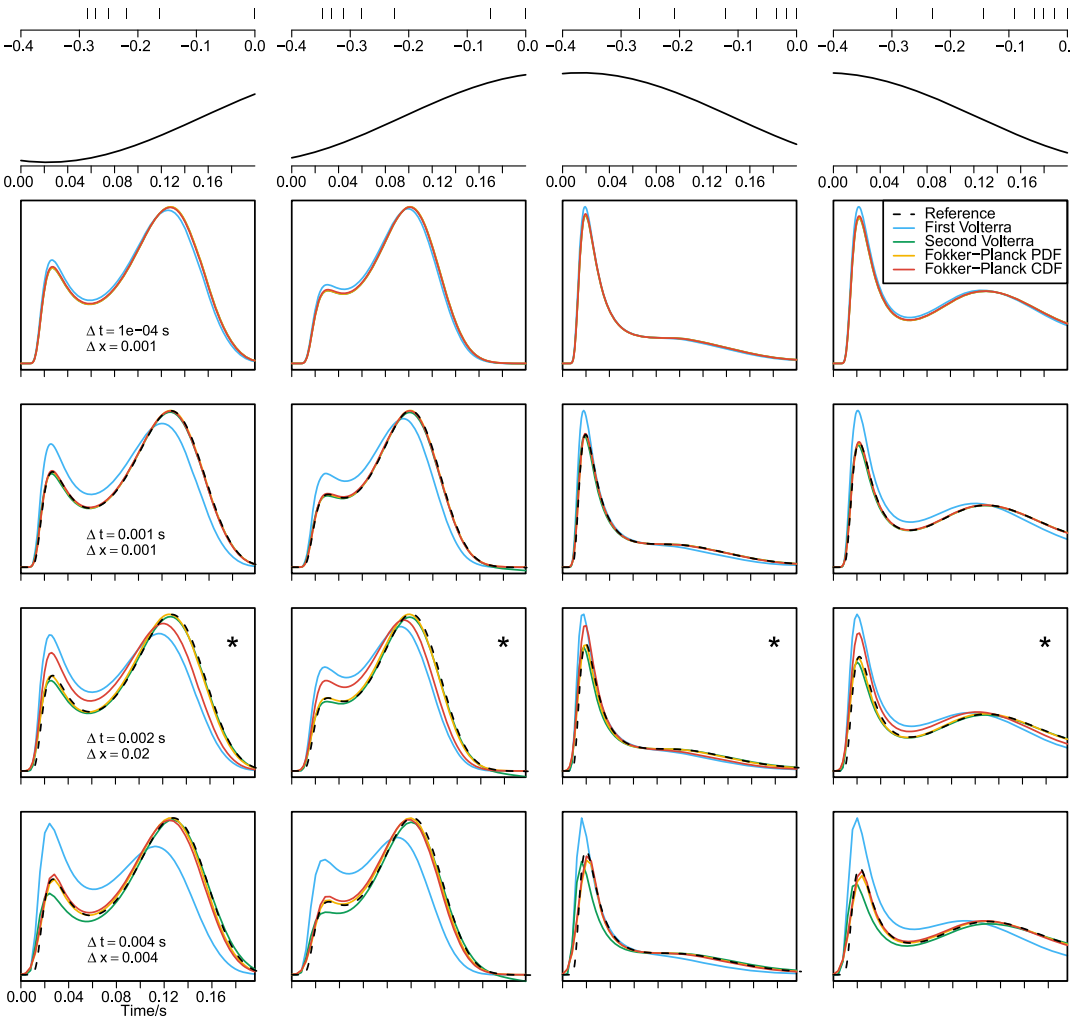


Fig. 3 Four example ISI probability density functions, $g(t)$, calculated with four methods using different grid sizes. The *column panels* show the four different ISIs, with the spike history indicated in the *top* (with different times axes) of each column, and the sinusoidal stimulus for the corresponding time periods. The *panels in the four lower rows* show solutions of the different PDEs and IEs using increasing grid sizes in each row. In the *three lower rows*, the density function from the panels above using the second Volterra method with high accuracy is plotted as the *reference* line. As expected, the solutions become less accurate as the grid size increases. The *second row from the bottom*, *indicated with a star in the upper right corner*, shows the grid size used for estimation in the main analysis, which leads to decent approximations for all four methods

tures. Also the performance of the marginal MLE and the EM algorithm in the case of the probability-mixing models should be compared. Third, it should be evaluated if it is possible to detect which of the two models generated the data. Results from these three analyses are presented in the following.

4.1 Numerical Solutions of the Partial Differential and Integral Equations

Figure 3 shows the PDFs of four example ISIs, i.e., for four different histories of past spikes, calculated by the four numerical methods, first Volterra, second Volterra, Fokker–Planck PDF and Fokker–Planck CDF, under single stimulus trials. Time has

been set to 0 at the last spike time. The examples are taken from a spike train attending to the single stimulus s_1. Each column shows one example ISI, with the spike history indicated above the column (with different time axes) and the corresponding sinusoidal stimulus (same time axes as the PDFs), for four different grid sizes. The four boxed panels in each column show the solutions of the PDEs and the IEs for the ISI on top. A reference dashed black line obtained with high accuracy has been added in all panels for comparison. The grid size is given by Δt for the time discretization, and Δx for the space discretization, and varies from row to row. As expected, for large grid sizes (small number of bins), the performance of the four methods differ (see the three lower rows of boxed panels), but the four results converge for decreasing grid sizes (see the upper row of boxed panels). We find that the first Volterra method is more sensitive to the grid size, while the Fokker–Planck PDF method is the most robust. In the parameter estimation below, we use $\Delta t = 0.002$ s and $\Delta x = 0.02$ shown in the row indicated with a star.

Figure 4(a) and (b) show the time-evolving PDF and CDF of X_t from the numerical solutions of the Fokker–Planck equation, for the ISI of the first column of Fig. 3. Time has been set to 0 at the last spike time. At 0, the PDF equals the (discretized) Dirac delta function, and the CDF equals the Heaviside step function, since at spike times, the voltage always resets to a fixed value, x_0. As time increases, the PDF shows how the probability flows out at the threshold; and the CDF at the voltage threshold illustrates the survival probability.

Figure 4(c) shows in the upper panels three examples of spike times PDFs, $g(t)$, and the lower panels show a corresponding example trace for each, plotted on top of their time-evolving PDFs of $X(t)$, $f(x,t)$, as heat-images. The three ISIs are taken from the left, middle left, and middle right panels of Fig. 3.

4.2 Results from Single Stimulus Trials

Parameter estimates of μ and σ from the 100 repetitions are shown in Fig. 5 as boxplots. In the lower panels, the time elapsed and the number of loops for optimization are also plotted. The means and standard deviations of parameter estimates are given in Table 2. The first Volterra method is less stable and less accurate, which is expected due to the lower accuracy in solving the spike time PDFs shown in Fig. 3. The second Volterra performs best for the estimation of σ, and the Fokker–Planck PDF performs best for μ, while the Fokker–Planck CDF does not perform as well as any of the two. On the other hand, the first Volterra and the Fokker–Planck CDF are less computational expensive. The Fokker–Planck CDF method is used in later analysis of stimulus mixtures, considering both accuracy and efficiency, though the Fokker–Planck PDF with a finer grid is used when performing KS-tests for model selection below. We also find that different methods result in different systematic estimation bias. When estimating μ some methods tend to overestimate and others tend to underestimate, whereas when estimating σ all methods have a tendency to overestimate.

In Fig. 6, the 100 estimated response kernels from the four methods are plotted together as colored lines. The parameters of the kernel are in practice not identifiable, so we evaluate by plotting the shape of the kernel function. All methods achieved

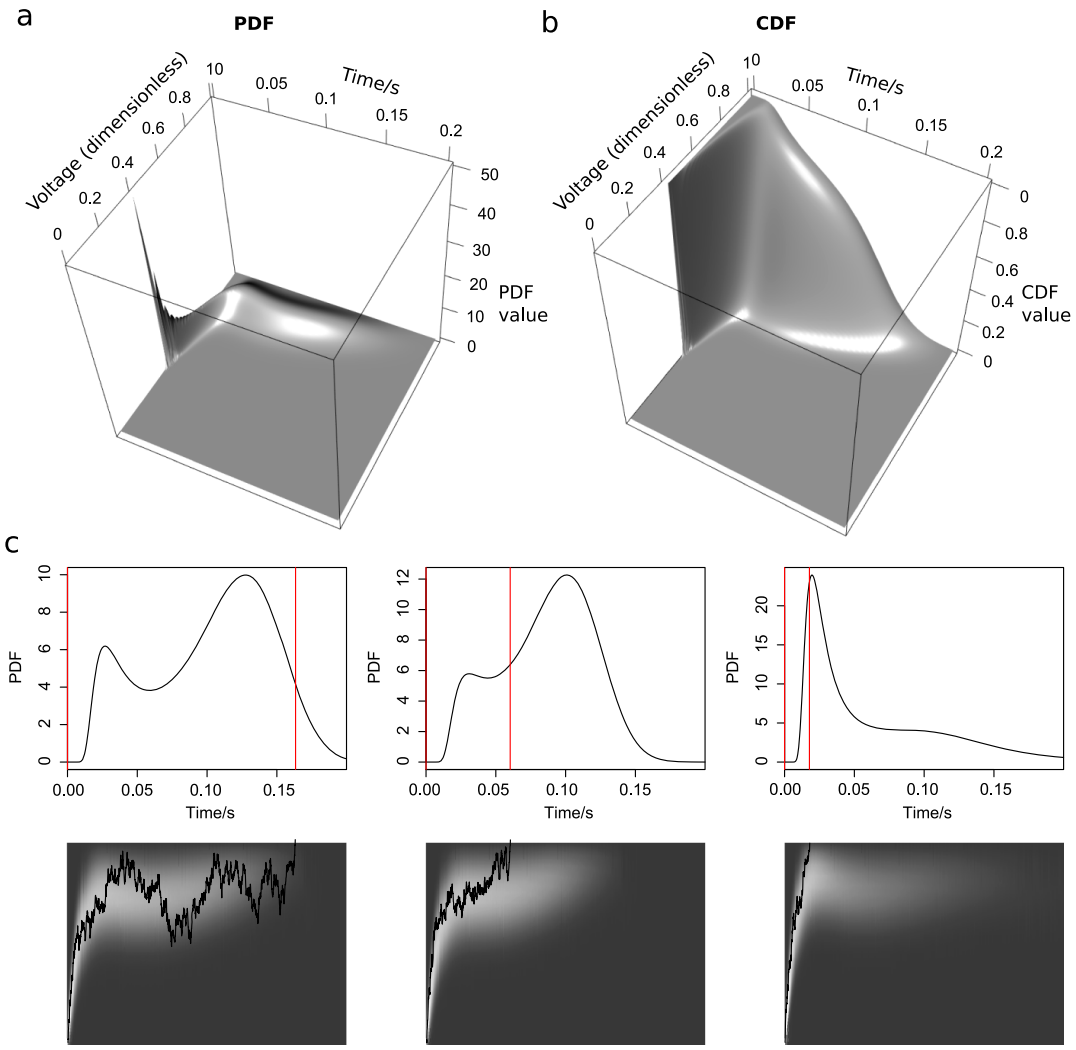


Fig. 4 Solutions of the PDEs and the IEs and example traces. The time-evolving **(a)** PDF, $f(x,t)$, and **(b)** CDF, $F(x,t)$, from the solutions of the Fokker–Planck equation for the ISI in the left column of Fig. 3. **(c)** Three example ISIs taken from the left, middle left and middle right columns of Fig. 3. The *upper panels* show the PDFs with *red lines* indicating the spike times. The *lower panels* show the time-evolving voltage PDFs as a heat image together with the realization of the voltage path. The *brighter region* in the heat image corresponds to larger PDF values. The time when the voltage trace hits the threshold in the heat image corresponds to the spike time shown in the *upper panel* as a *red line*. Note that in the *upper panel*, the time intervals with larger ISI PDF values are where the probability (*bright region*) flows faster out of the threshold in the *lower panel*

good results, capturing the overall shape. The two PDE methods obtained slightly better results, whereas the IE methods are systematically biased.

In Fig. 7(a) are QQ-plots of the uniform residuals calculated using the transformation from Eq. (23) for the four methods. The uniform residuals are pooled together from all 100 repetitions. Again, all four methods are competitive but biased, with a different bias for PDE methods and for IE methods. This bias, arising from the numerical approximations, has to be taken into account when later testing which model generated the data, forcing us to use a finer and computationally more expensive grid size.

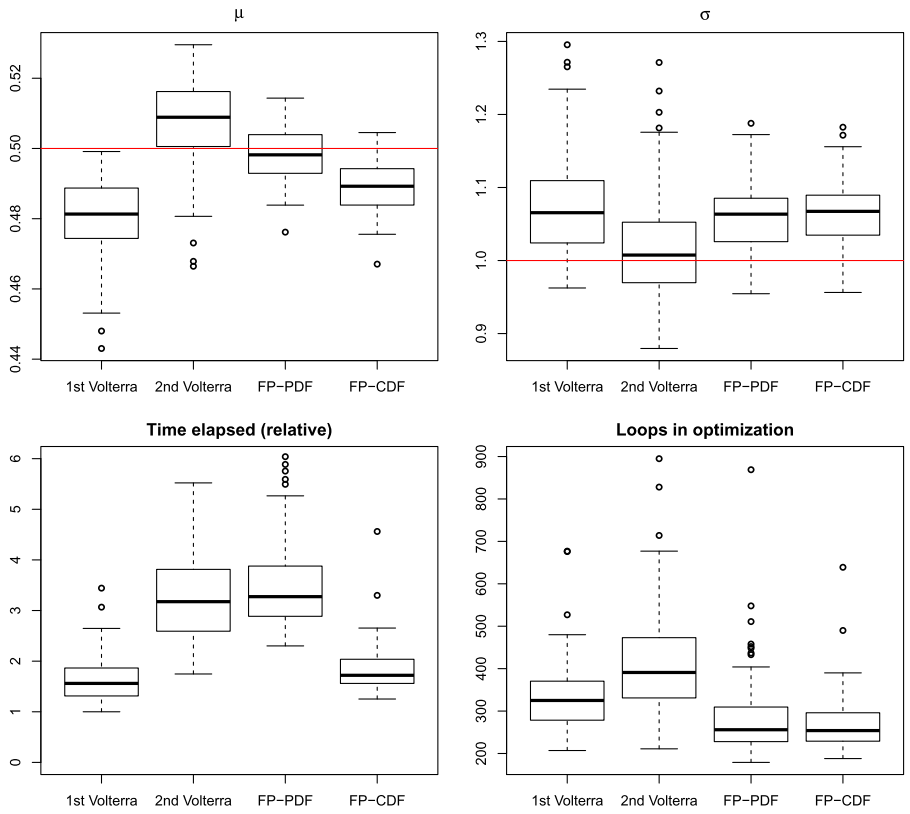


Fig. 5 Parameter estimates and computational time. *Upper panels*: Box-plots of parameter estimates for μ (*left*) and σ (*right*) from 100 repetitions of single stimulus data. The *red lines* are the true values used in the simulations. *Lower panels*: The time elapsed and number of loops for the optimization

Table 2 Average $\pm$ standard deviation of 100 parameter estimates from single stimulus data

	μ	σ
True value	0.5	1
First Volterra	0.4800 ± 0.01095	1.076 ± 0.06913
Second Volterra	0.5066 ± 0.01287	1.020 ± 0.07281
Fokker–Planck PDF	0.4981 ± 0.00730	1.060 ± 0.04567
Fokker–Planck CDF	0.4889 ± 0.00698	1.065 ± 0.04442

4.3 Distinguishing Between Response-Averaging and Probability-Mixing

The following results show that the two models can be distinguished for parameter values such that the two models are sufficiently different, which will be defined below in Sect. 4.6. Each model is fitted using the Fokker–Planck CDF method, both on data simulated according to the correct model as well as the wrong model. Figure 8 shows the estimation of μ, σ, and α or β, depending on the model, and Table 3 reports the means and standard deviations of estimates. Accurate estimation is achieved only if we apply the correct model to the corresponding data, the wrong model fitted to data generated by the other model clearly shows bad results. This implies that it is important to use the correct model for reliable inference, but we

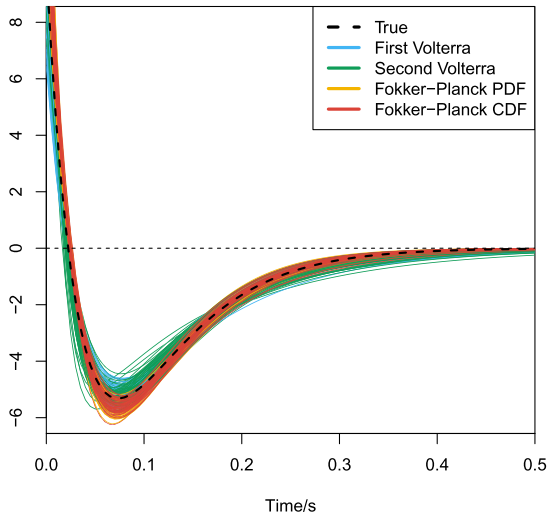


Fig. 6 Estimates of the response kernel from 100 simulated data sets fitted to single stimulus data with the four numerical methods, each method has its own color. The *dashed black curve* is the true kernel used in the simulations

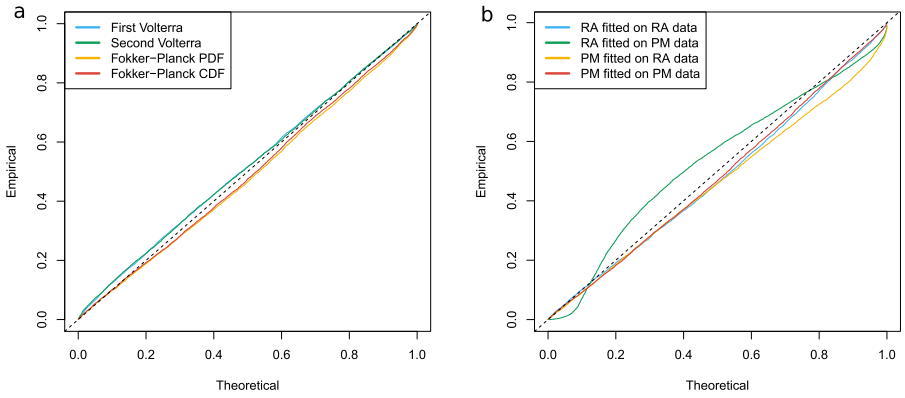


Fig. 7 Model control. **(a)** QQ plots of the uniform residuals calculated using the transformation in Eq. (23) for the four methods fitted on single stimulus data and a grid size of $\Delta t = 0.002$ s and $\Delta x = 0.02$. The uniform residuals are pooled together from all 100 repetitions of the simulations. The bias is different for PDE methods and for IE methods, seen from how the points deviate from the identity line. **(b)** QQ plots of the uniform residuals of the probability-mixing (PM) model and the response-averaging (RA) model fitted on data simulated from both models responding to a stimulus mixture. For example, RA fitted on PM data means fitting the response-averaging model to data simulated from the probability-mixing model. From the QQ-plots a wrong model can be rejected

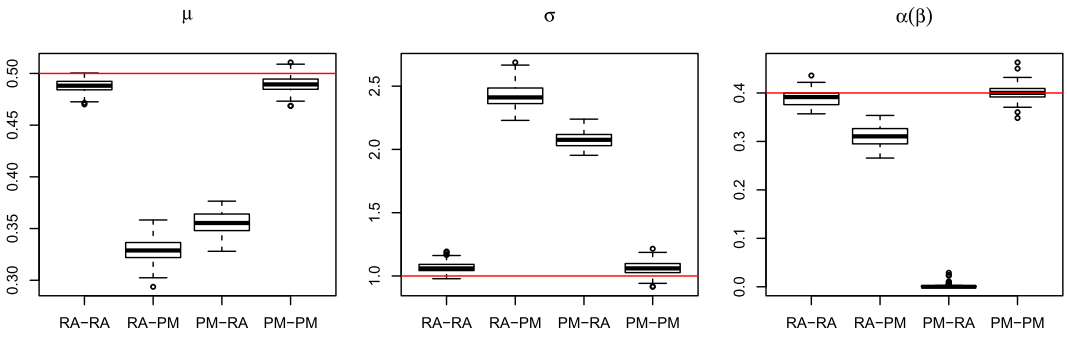


Fig. 8 Parameter estimates of the probability-mixing (PM) model and the response-averaging (RA) model fitted to data simulated from both models responding to a stimulus mixture. For example, PM-RA means parameter estimates of the probability-mixing model fitted to data simulated from the response-averaging model

Table 3 Average ± standard deviation of 100 parameter estimates using the response-averaging (RA) model and the probability-mixing (PM) model on data sets simulated according to the two models

	μ	σ	α_1 (PM)/β_1 (RA)
True value	0.5	1	0.4
RA on RA data	0.4876 ± 0.00658	1.067 ± 0.04441	0.3888 ± 0.01564
PM on RA data	0.3553 ± 0.01087	2.077 ± 0.06482	0.0017 ± 0.00467
RA on PM data	0.3288 ± 0.01191	2.429 ± 0.09216	0.3098 ± 0.02161
PM on PM data (Marginal)	0.4891 ± 0.00844	1.062 ± 0.05609	0.4013 ± 0.01636
PM on PM data (EM)	0.4889 ± 0.00813	1.063 ± 0.05410	0.3988 ± 0.01012

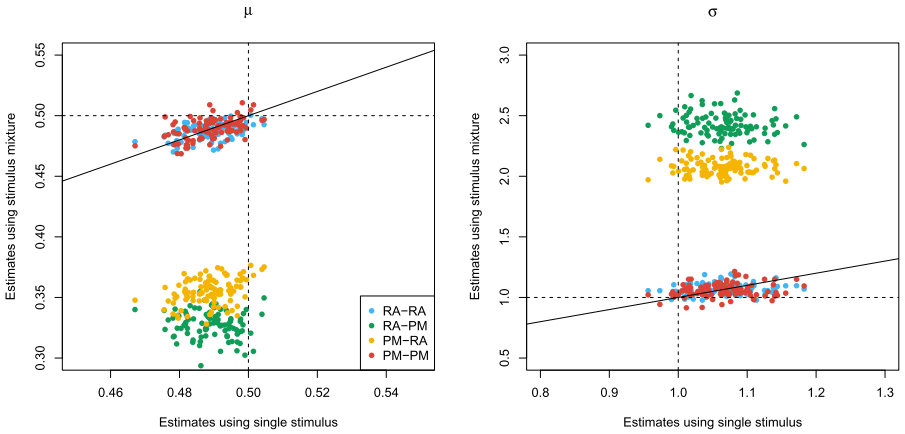


Fig. 9 Estimates of μ and σ estimated from stimulus mixture data under either the probability-mixing or the response-averaging model plotted against the estimates from single stimulus data, for 100 repetitions. The *straight lines* are identity lines, the *dashed lines* are the true values used in the simulations. *Different colors* differentiate which model is fitted on which data for the stimulus mixture. The estimates from a stimulus mixture differ significantly from the estimates from a single stimulus when the model is wrong

can also use this to distinguish the two models. If estimates of μ and σ change considerably from estimation on single stimulus data to estimation on stimulus mixture data, then one should suspect that the used model is wrong. This is illustrated in Fig. 9, where scatterplots of estimates from stimulus mixture data assuming a specific model is plotted against estimates from single stimulus data. The straight lines are identity lines. When the correct model is used, estimates are clustered around the identity line, but clearly separated away from the identity line if the model used for fitting is wrong. To formalize the model selection procedure, QQ plots of uniform residuals using Eq. (23) from all 100 repetitions are shown in Fig. 7(b), where points away from the identity line indicate the model is wrong. The lines for the wrong model selections are clearly worse than the correct models, but even the correct models show a significant deviation from the identity lines, which would turn out as also the correct model being rejected in a KS-test. This is most probably due to the numerical approximations, as also seen in Fig. 7(a). To check this, we conducted the same estimation procedure with the Fokker–Planck PDF method using a finer grid of $\Delta t = 0.0005$ s and $\Delta x = 0.01$, and repeated for 20 times. Results are reported in Table 4, where it is clear that with a finer grid, the KS-test works

Table 4 Rejection ($p < 0.05$) rate based on the Kolmogorov–Smirnov test for uniformity done on each repetition

Method	Low accuracy*	High accuracy**
RA on RA data	32/100	1/20
RA on PM data	100/100	20/20
PM on RA data	100/100	20/20
PM on PM data	32/100	0/20

*Fokker–Planck CDF method with $\Delta t = 0.002$ s and $\Delta x = 0.02$

**Fokker–Planck PDF method with $\Delta t = 0.0005$ s and $\Delta x = 0.01$

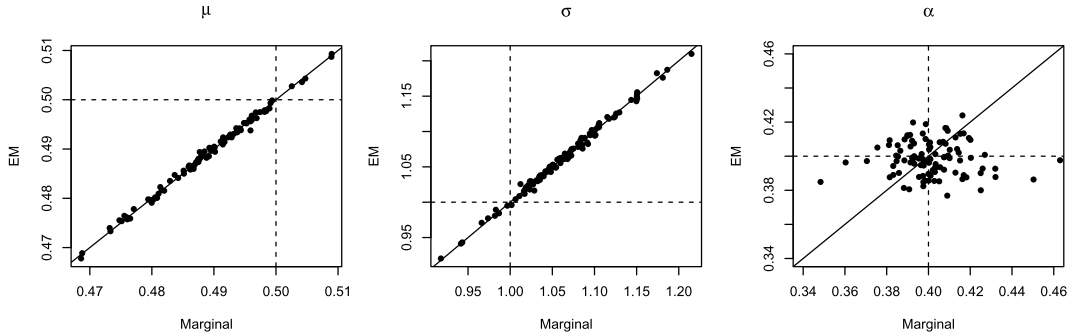


Fig. 10 Scatter plots of the estimates using the EM algorithm against MLE with the marginal probability for 100 repetitions. The *dashed lines* are the true values used in the simulations. The two methods give almost the same results for μ and σ, whereas some zero-mean random fluctuations are seen for α. In this case, the EM algorithm appears to be the most precise

as desired with high power to detect deviations from the correct model. We suggest that for parameter estimation a very fine grid is not needed, whereas for model control, the numerical approximation of the spike time PDF has to be precise. To conclude, the two models are distinguishable for the parameter settings explored here.

4.4 Probability-Mixing with EM

In the previous section, the marginal MLE was used when fitting the probability-mixing model. Here we compare the performance of the marginal MLE and the EM algorithm on the probability-mixing model fitted to the corresponding data. Figure 10 shows scatterplots of estimates obtained by the two methods, and the last two rows in Table 3 show the means and standard deviations. The two methods provide similar results, and have the same accuracy for all three parameters. However, the variance of the EM algorithm is slightly smaller, particularly for α. The computational burden in one loop of the numerical optimization for the two methods is approximately the same.

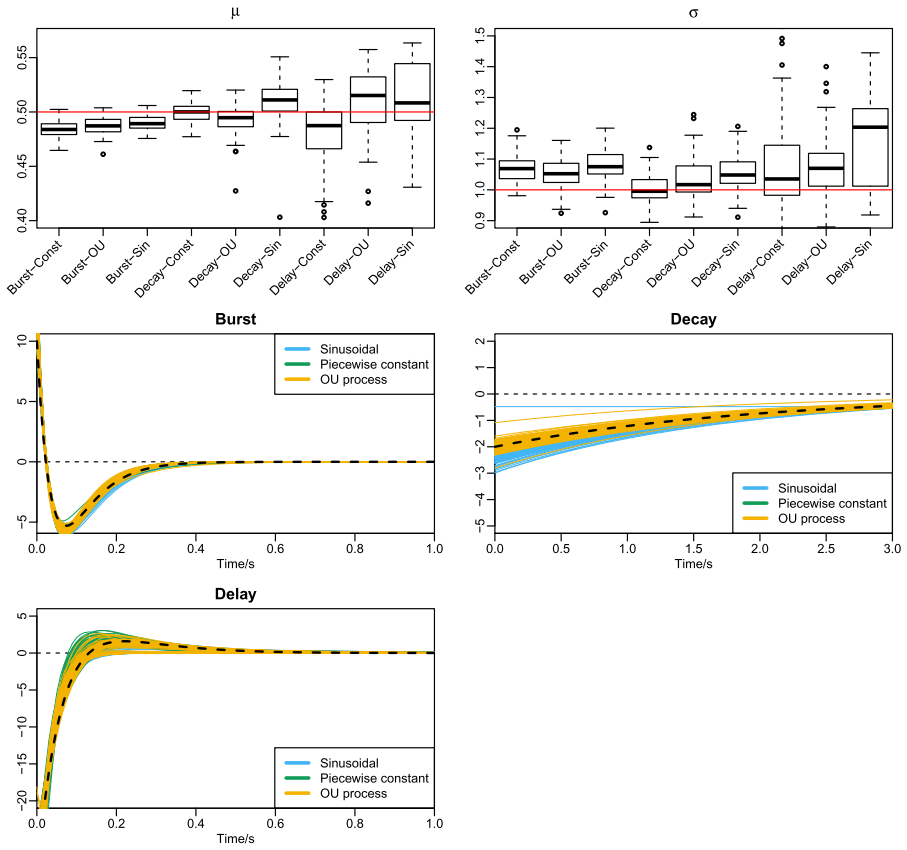


Fig. 11 Parameter estimates of single stimuli for different combinations of response kernels and stimuli. *Top panels* show the estimates of μ (*left*) and σ (*right*) as box plots. The *x-axis* shows the nine combinations, for example Burst-Const means the burst kernel with a piecewise constant stimulus, Delay-OU means the delay kernel with a stochastic stimulus generated by the OU process, and so on. The delay kernel induces the largest variance in parameter estimates. *Middle* and *bottom panels* show the estimates of the three types of response kernels. *Different colors* distinguish between the three stimulus types

4.5 Generalizations

In this section we only apply the Fokker–Planck CDF method and analyze the model for different types of response kernels and stimuli.

Single stimulus. We analyze nine combinations of response kernels and stimuli. For each combination we simulate 10 spike trains following one single stimulus. Figure 1 shows the combinations and the realizations of spike trains. On these spike trains parameters and response kernels are estimated. The simulations are then repeated 100 times. For the stochastic stimulus, we use a single realization so that the stimulus is identical in all repetitions and the statistical performance of the estimators can be assessed. The estimates of parameters and response kernels are shown in Fig. 11. The estimates using the delay kernel have larger variance, possibly due to our specific choice of kernel parameters that makes the spiking rate less sensitive to stimulus strength (see bottom panels of Fig. 1). The estimates of parameters and kernels for all combinations are acceptable. The parameters used for the response kernels and stimuli are shown in Table 5.

Table 5 Parameter values for all response kernels and stimuli used in the single stimulus study for the generalized analysis

	Category	Parameter value
Stimulus, s	Sinusoidal	(10, 12, 1, 50)
	Piecewise constant	(50, 70, 50, 30, 50, 60, 0, 1.3, 1.7, 2.3, 2.7, 3.8, 5)
	OU process	(50, 20)
Response kernel, η	Bursting	(50, 25, 40, 15)
	Decay	(0, 0, 2, 0.5)
	Delay	(20, 8, 50, 15)

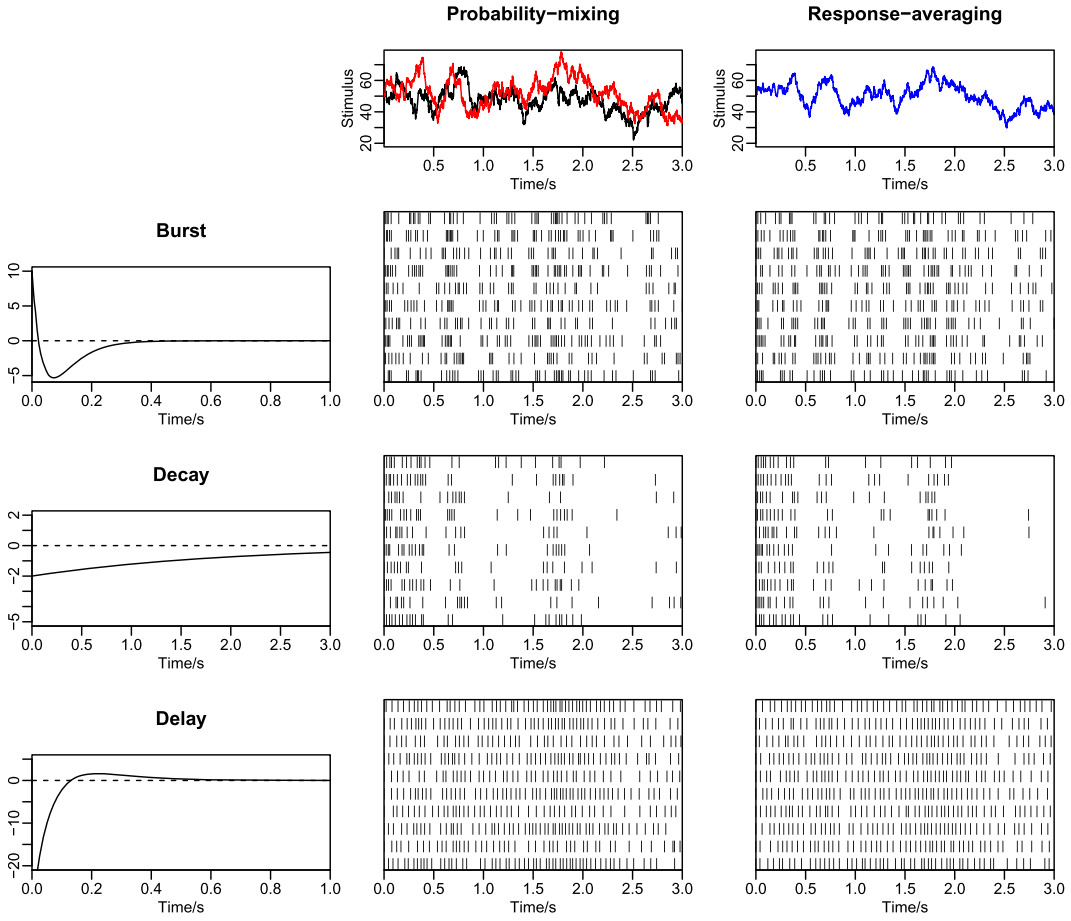


Fig. 12 Realization of spike trains for a stimulus mixture consisting of two OU processes for three types of response kernels, assuming either probability-mixing (*left*) or response-averaging (*right*). In the *top panels*, the *left* shows the two stimuli, and the *right* shows the weighted average of the two. For the 10 spike trains simulated from the probability-mixing model, four respond to the same stimulus and six respond to the other

Stimulus mixtures. We use two OU processes as stimuli, and apply all three types of response kernels. The top panels of Fig. 12 show the two stochastic stimuli, and their weighted average. The latter is what neurons respond to according to the response-averaging model. For each combination, we simulate 10 spike trains, using

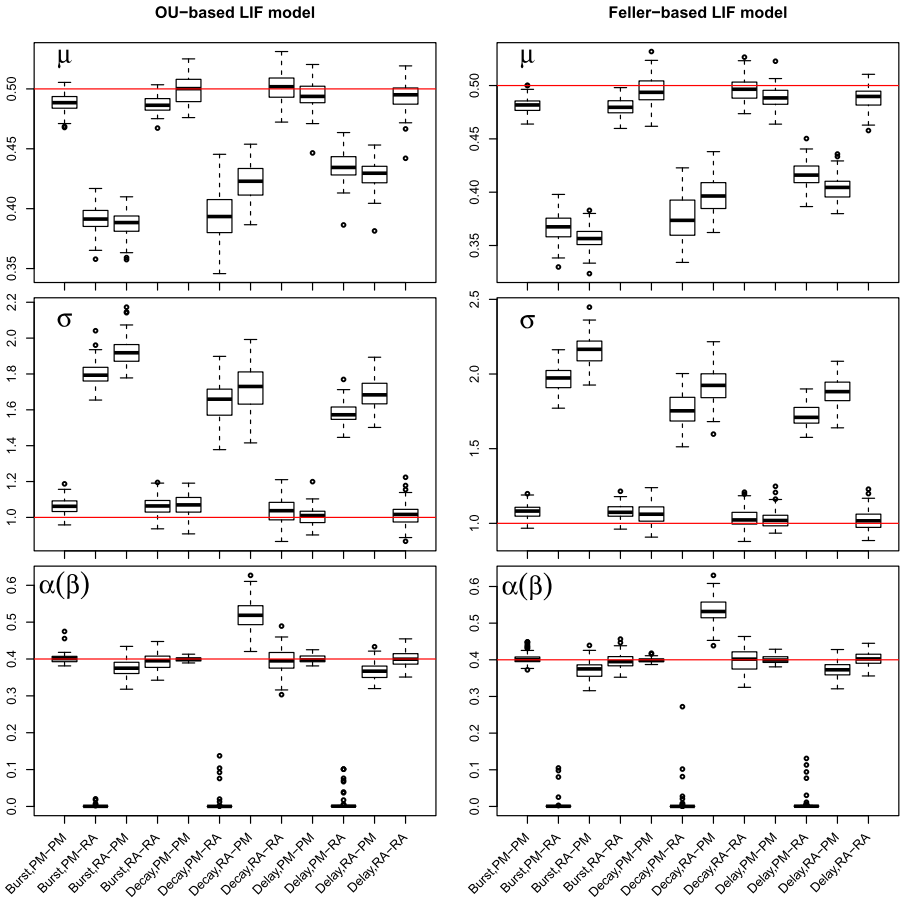


Fig. 13 Parameter estimates for a stimulus mixture consisting of two OU processes for three types of response kernels, assuming either probability mixing or response averaging. In the *left panel* is shown the estimates of the OU-based LIF model, and in the *right panel* is shown the Feller-based LIF model. In both *left and right panels*, the *x-axis* shows 12 cases combining response kernels, probability mixing and response averaging. For example, Decay, PM-RA means fitting the probability-mixing model to data simulated from the response-averaging model, using the decay kernel

identical stimuli in each repetition. Results are shown in the left panels of Fig. 13, where both the probability-mixing (PM) model and the response-averaging (RA) model are fitted to data generated from both models. When fitting the probability-mixing model, only the EM algorithm is applied. We employ the same strategy as in the main analysis: we first estimate parameters on data generated from single stochastic stimuli, and then fix the response kernel and estimate the other parameters on data generated from stochastic stimulus mixture. The results for all three kernels on a stochastic stimulus mixture are the same as the main analysis above using the bursting kernel and sinusoidal stimuli: we obtain accurate estimates of all parameters only if we apply the correct model to the corresponding data.

State dependent noise. Finally, the diffusion term in the LIF model (1) was modified to include the square root of $X(t)$ as in the Feller model [41–43], yielding

$$X(t) = \big(-\gamma\big(X(t) - \mu\big) + I(t) + H(t)\big)\,dt + \sigma\sqrt{X(t)}\,dW(t). \tag{24}$$

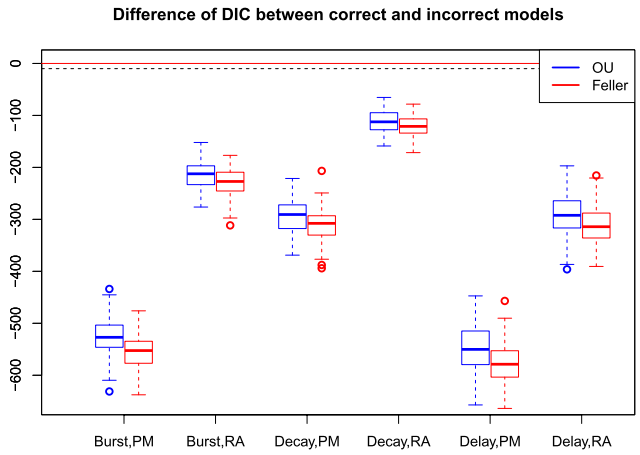


Fig. 14 Difference of DIC between correct and incorrect models. We calculate the difference of DIC between fitting the correct model to the corresponding data and fitting the incorrect model to the same data, and plot the difference as box-plots for 100 repetitions. The *x-axis* shows different combinations of kernel and data. For example, Burst, PM means the difference of DIC between using correct model (PM) and incorrect model (RA) on PM data, under the burst kernel. Likewise, Delay, RA means the difference of DIC between using correct model (RA) and incorrect model (PM) on RA data, under the delay kernel. *Blue* stands for the OU-based LIF model and *red* stands for the Feller-based model. A difference of -10 is shown as a *dashed line*. A difference greater than ± 10 is regarded as strong evidence of supporting one model over the other [44]

Table 6 Rejection ($p < 0.05$) rate based on the Kolmogorov–Smirnov test for uniformity, using different response kernels with the mixture of stochastic stimuli

		RA-RA	RA-PM	PM-RA	PM-PM
OU	Burst	22/100	99/100	100/100	19/100
	Decay	1/100	100/100	83/100	1/100
	Delay	30/100	77/100	97/100	34/100
Feller	Burst	23/100	100/100	95/100	22/100
	Decay	0/100	100/100	81/100	1/100
	Delay	30/100	84/100	100/100	37/100

Results of both the OU-based and the Feller-based LIF models are shown

The same analysis as in the previous section was repeated using two OU processes as stimuli and three types of response kernels. Results are shown in the right panels of Fig. 13, which are almost the same as the results using the original LIF model shown in the left panels.

Model selection. In stimulus mixture analysis, model selection is conducted for both the OU-based and the Feller-based LIF models. In Fig. 14 we compare the deviance information criterion (DIC) between the correct and the incorrect model. The DIC difference equals -2 times the difference of the log-likelihoods, because the two models have the same number of parameters. The correct model is strongly supported in every case. Table 6 shows rejection ($p < 0.05$) ratios using KS-tests for all combinations in the stimulus mixture analysis. We also tried other pairs of stochastic

stimulus mixtures (results not shown) and found that the more similar the two stimuli are, the more the rejection ratios tend to decrease, whether using the correct or the incorrect model, and if two stimuli are more different, all rejection ratios tend to increase, including rejections of the true model. Finally, as expected the KS-test rejection ratio is sensitive to data size: using smaller number of spike trains reduces the rejection ratio. In particular, the rejection of fitting the PM model to RA data (PM-RA) with the decay kernel, and fitting the RA model to PM data (RA-PM) with the delay kernel, is extremely sensitive to similarity of stimuli and data size. This makes the KS-tests less robust. Thus, we recommend using the KS-tests together with other model selection methods for more reliable conclusions.

4.6 Model Selection Accuracy

The results above show that parameters can be inferred and the correct model can be determined for the specific parameter choices used in the simulations. Here we explore the model selection accuracy for varying parameter values including the weight, stimulus dissimilarity, stimulus strength and number of spike trains. In the following analysis, we use the bursting response kernel, a mixture of two stochastic stimuli and the Fokker–Planck CDF method. To introduce a stimulus dissimilarity, a sinusoidal perturbation is added to one of two identical OU processes, $\tilde{S}(t) = S(t) + a\sin(10t)$, where t is measured in seconds and a is the perturbation size. To change the stimulus strength, the OU processes are linearly scaled using $\tilde{S}(t) = bS(t)$ where b denotes the scaling size.

We focus on model selection accuracy without reporting parameter estimates. Model selection is denoted successful if the DIC for the true model is more than 2 smaller than the wrong model. This is the value suggested in [44] to indicate substantial empirical support for the selected model compared to the other model. Figure 15 explores model selection results as a function of parameter values, and provides an overall picture how these parameters affect model selection. The conveyed message verifies our intuition: model selection is more reliable if the stimuli are more different, the weights are more even, the stimulus difference is stronger or the sample size is larger (a larger number of spike trains). The first three make the responses of the two models more different, and the last provides more statistical power. Furthermore, the thresholds of these parameter values in terms of successful model selection are surprisingly low. A weight value of 0.2 and a perturbation size around 6 (i.e., around 10 % of the stimulus strength) are sufficient to ensure a decent selection. For a more even weight of 0.4, only a perturbation size of 3 (around 5 %) is necessary to provide good model selection for both RA and PM data. Indeed, 5 % perturbation in a stimulus is undetectable by a simple graphical inspection of the spike trains (bottom panels in the figure), but the finer statistical analysis can detect the difference between the models. Even with small weight and stimulus dissimilarity, model selection can be improved by using stronger stimuli or enlarging the sample size with more spike trains. Note that these analyses are easily generalized for a given problem at hand by first estimating the response kernel of a given neuron under a given stimulus, and then simulating data with this response kernel and stimulus, varying parameters of the two models. That will indicate for which parameter values the model selection can be trusted.

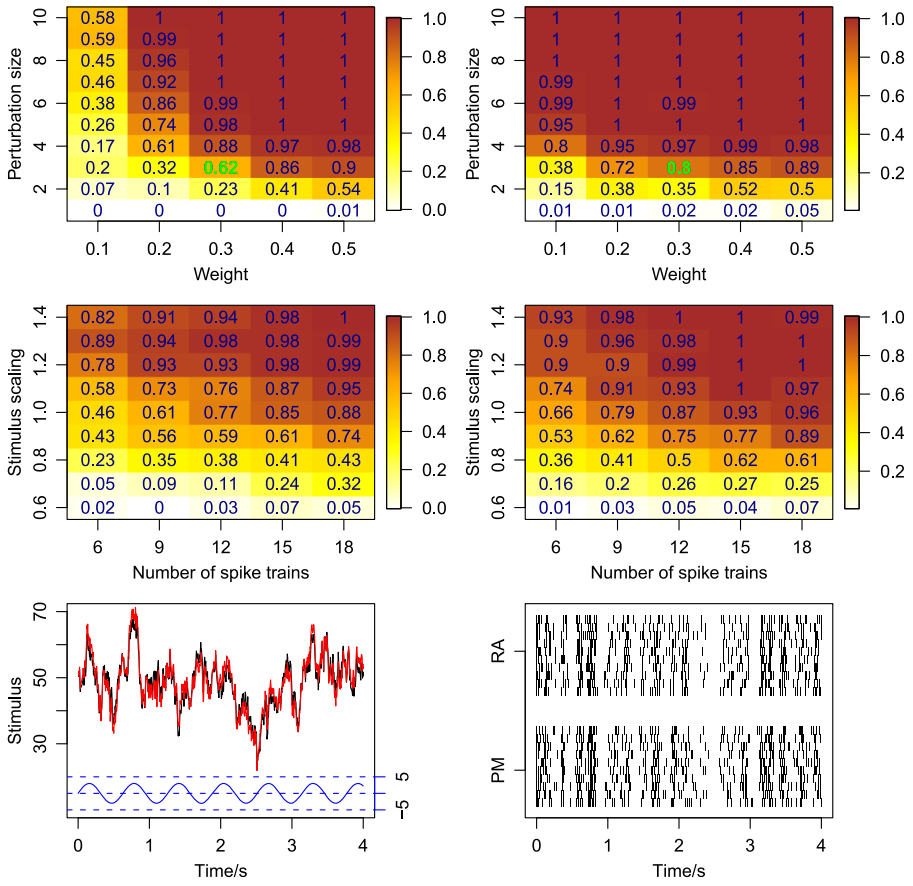


Fig. 15 Model selection accuracy. Successful selection is defined as a DIC difference greater than 2, and the proportion of correctly identified models is calculated over 100 repetitions. Note that a not correctly identified model in most cases means that the DIC difference was smaller than 2, not that the wrong model was selected. *Top left*: proportion of correctly identified models with weights from 0.1 to 0.5 and perturbation size from 1 to 10 for RA data, using 10 spike trains. *Top right*: the same for PM data. *Middle left*: proportion of correctly identified models for number of spike trains of 6 to 18 and stimulus scaling from 0.6 to 1.4 for RA data, using a weight of 0.3 and a perturbation size of 3, shown in *green* in the *top panels*. *Middle right*: the same for PM data. *Bottom left*: the two stimuli curves (*black* and *red*) with perturbation size 3 (sinusoidal curve shown in *blue*) used for the cases shown in *green* in *top panels*. *Bottom right*: example spike trains following either RA or PM, using the two stimuli shown in the *left*, with weight 0.3

5 Discussion

5.1 Estimation of the Decay Rate

We have shown that parameter inference can be successfully conducted for the probability-mixing and the response-averaging model on corresponding data incorporating different response kernels for LIF neurons. The decay rate γ has been assumed known. We also attempted to estimate all parameters including γ (results not shown), but the optimization often finds local minima and leads to low accuracy. The estimation of γ seems to suffer from identifiability problems, due to only observing spike times and not the underlying membrane potential. Nevertheless, to estimate γ we may fix it at different values and run the optimization procedure for the rest of

the parameters, and then compare the model fit for the different γ values. This is not pursued here.

5.2 Bias of the Numerical Methods

We found that the parameter estimates and the QQ plots from the four methods suffer from over- and underestimation issues. The MLE is based on the first-passage time probabilities, which we obtain using four numerical methods, Fokker–Planck PDF, Fokker–Planck CDF, first Volterra and second Volterra. Because of the intrinsic differences between these methods, discretization leads to different biases of the calculated spike time PDFs. As seen from Fig. 3, when increasing the grid size, the first Volterra and the Fokker–Planck CDF methods tend to increase the PDF value in the beginning of the ISI, while the second Volterra tends to slightly decrease it. The low accuracy of the first Volterra method arises from a singularity of $f^*(x,t|v,s)$ when $v = x$ and $t \to s$. However, by removing the singularity the second Volterra is more accurate for numerical computations.

5.3 Efficiency of Numerical Methods

We choose the Fokker–Planck CDF method for estimation of mixtures, because it achieves a well-behaved balance between accuracy and computational burden. Table 2 also shows that this method has the smallest variance on parameter estimates.

Although the first Volterra method is the computationally fastest, it has poor convergence, as seen from the number of loops in the bottom right panel in Fig. 5. Overall, the PDE methods tend to converge faster than the IE methods.

The performance is affected by the grid size. The estimates in Fig. 5 uses $\Delta t = 0.002$ s and $\Delta x = 0.02$. This discretization setting generally achieves acceptable computation times and statistical accuracy, but as shown in Sect. 4.3, a finer grid is needed for model selection. One may tweak the grid sizes in order to obtain separate settings for each of the four methods to obtain comparable efficiency and accuracy. However, considering that in practical data the errors come from many sources like measurement errors and approximate modeling, the optimal discretization on simulated data is of less importance and interest. Thus, we suggest the current setting as providing a generally good balance, and we will not investigate this further.

5.4 EM for Better Estimation of Mixture Probabilities

Figure 10 shows that the estimation of the mixture probability parameter α is slightly less stable for the marginal MLE than for the EM algorithm. The EM algorithm implicitly enlarges the data size by using latent variables for the mixture probability, referred to as *data augmentation* [45]. The complete-data log-likelihood function used in the M step does not contain logarithms of sums, making the estimation more stable. By iteratively updating the expectation in the E step and obtaining stable estimation in the M step, the EM algorithm improves the stability when inferring the probability-mixing model, and in general, mixture models.

Although the EM algorithm performs better, it is only slightly better for α and the improvement is negligible or non-existent for μ and σ. This is because we only use

two components in the mixture, which does not generate notable differences between the marginal MLE and the EM algorithm. A larger advantage of the EM algorithm can be expected under more complex stimulus mixtures. Furthermore, the response kernel is fixed, and the two methods use the same initial values for μ and σ (obtained from the single stimulus trials) in the optimization procedure, which also contributes to the similarity of results between the two methods.

5.5 Extension of Noise

In this paper a one-dimensional stochastic differential equation model driven by a Wiener process for the membrane potential has been considered, which arises as an approximation to Stein's model [46], leading to the OU model, or to the extended model including reversal potentials, proposed by Tuckwell [41], leading to the Feller model [42]. The model does not take into account specific dynamics of synaptic input or ion channels, which affects the dynamics, see, e.g., [47–49], where the autocorrelations of the synaptic input is shown to be an important factor. This is partially accounted for in our model through the memory kernels. Incorporating autocorrelated synaptic input or ion channel dynamics would lead to a multi-dimensional model. In principle, the first-passage time probabilities could then be obtained by solving multi-dimensional Fokker–Planck equations [24]. However, the statistical problem is further complicated by the incomplete observations, since typically only the membrane potential is measured, as studied in [50]. In even more realistic models non-Gaussian noise can be included, for example combining the diffusion process with discrete stochastic synaptic stimulus arrivals, leading to a jump-diffusion process, whose Fokker–Planck equation is generalized as an integro-differential equation [51]. Solving multi-dimensional or generalized Fokker–Planck equations are significantly more expensive and exact MLE becomes less appealing. This is not pursued here.

5.6 The Response-Averaging Model

The response-averaging model used here is slightly different from the response-averaging model by Reynolds et al. [8]. In our model the average is calculated over the currents for each stimulus, while in their model the average is calculated over the firing rates for each stimulus. The reason is as follows. In a spiking neuron model like the LIF model, the generation of each single spike rather than the firing rate is modeled. Whether in the probability-mixing model, the response-averaging model or any other model, the spiking is affected by stimuli only through currents. Our model is formulated based on this idea, using a unified spike-generating mechanism for both the probability-mixing and the response-averaging model. The resulting firing rate averaged over a time window from a weighted average of single stimuli, will also be a weighted average of firing rates from single stimuli but with different weights. Our response-averaging model therefore provides the same consequence in terms of firing rates as the model by Reynolds et al.

5.7 Model Selection of Probability-Mixing and Response-Averaging

We finish by addressing the possible model selection methods for probability mixing and response averaging on real data. We have shown that the probability-mixing

and the response-averaging models can be clearly distinguished if fitted on simulated data. However, real data will likely not follow exactly one of the two models, but one of the models might give a better description of the data than the other. We might need to design more sophisticated methods for model checking and model selection. Apart from conducting uniformity tests based on the uniform residuals from the transformation (23), such as the KS-test as we have done, we can compare the Akaike information criterion (AIC) and Bayesian information criterion (BIC) between the two models. We have used a unified DIC method due to equal number of parameters, but AIC and BIC should be used if two models have differing numbers of parameters. Furthermore, the model can also be checked by evaluating the performance of prediction (of spikes) and decoding (of stimuli), using methods such as root mean squared deviation (RMSD) between empirical and predicted values. See [19] for the use of these approaches to distinguish between the two models on experimental data from the middle temporal visual area of rhesus monkeys.

Competing Interests

The authors declare that they have no competing interests.

Authors' Contributions

KL, SD: Conceived and designed the research. KL: Performed all analyses, simulations and figures. All authors interpreted the results. All authors wrote the paper.

Acknowledgements The work is part of the Dynamical Systems Interdisciplinary Network, University of Copenhagen.

Appendix

A.1 The EM Algorithm for Stimulus Mixtures

The complete likelihood for the full data (D, Y) is

$$\begin{aligned} L_c(\theta; D, Y) &= \prod_{i=1}^{N} \prod_{j=1}^{N_i} g\left(t_j^i, y_i|\theta\right) \\ &= \prod_{i=1}^{N} P(y_i|\theta) \prod_{j=1}^{N_i} g\left(t_j^i|y_i, \theta\right) \\ &= \prod_{i=1}^{N} \alpha_{y_i} \prod_{j=1}^{N_i} g\left(t_j^i|y_i, \theta\right). \end{aligned} \tag{25}$$

A.1.1 Expectation Step

The expectation of the full data log-likelihood conditional on the previous parameters θ_{-1} and the observed data D is

$$\begin{aligned} Q(\theta|\theta_{-1}) &= \mathbb{E}\big[\log L_c(\theta; D, Y)|\theta_{-1}, D\big] \\ &= \mathbb{E}\left[\sum_{i=1}^{N}\left(\log \alpha_{y_i} + \sum_{j=1}^{N_i}\log g\big(t_j^i|y_i,\theta\big)\right)\Big|\theta_{-1}, D\right] \\ &= \sum_{i=1}^{N}\left[\mathbb{E}\left(\log \alpha_{y_i} + \sum_{j=1}^{N_i}\log g\big(t_j^i|y_i,\theta\big)\Big|\theta_{-1}, D\right)\right] \\ &= \sum_{i=1}^{N}\left[\sum_{k=1}^{K} P(y_i = k|\theta_{-1}, d_i)\left(\log \alpha_{y_i} + \sum_{j=1}^{N_i}\log g\big(t_j^i|y_i,\theta\big)\right)\right]. \end{aligned} \tag{26}$$

The conditional probability of the latent variable is obtained from Bayes formula:

$$\begin{aligned} P(y_i = k|\theta_{-1}, d_i) &= \frac{P(y_i = k|\theta_{-1})\prod_{j=1}^{N_i} g(t_j^i|y_i = k, \theta_{-1})}{\sum_{l=1}^{K} P(y_i = l|\theta_{-1})\prod_{j=1}^{N_i} g(t_j^i|y_i = l, \theta_{-1})} \\ &= \frac{(\alpha_k)_{-1}\prod_{j=1}^{N_i} g(t_j^i|y_i = k, \theta_{-1})}{\sum_{l=1}^{K}(\alpha_l)_{-1}\prod_{j=1}^{N_i} g(t_j^i|y_i = l, \theta_{-1})}. \end{aligned} \tag{27}$$

A.1.2 Maximization Step

In the Maximization step, the new parameter θ is obtained by optimizing the conditional expectation $Q(\theta|\theta_{-1})$. A new iteration is then initiated using θ as the previous parameter. The loops run until θ and θ_{-1} are sufficiently close.

A.2 The Fokker–Planck CDF Method

Plugging $f(x,t) = \partial_x F(x,t)$ into the Fokker–Planck PDE

$$\partial_t f(x,t) = -\partial_x\big(b(x,t)f(x,t)\big) + \frac{\sigma^2}{2}\partial_{xx}^2 f(x,t) \tag{28}$$

gives

$$\partial_t \partial_x F(x,t) = -\partial_x\left[b(x,t)\partial_x F(x,t) - \frac{\sigma^2}{2}\partial_x\partial_x F(x,t)\right]. \tag{29}$$

Integrating both sides w.r.t. x yields

$$\partial_t F(x,t) = -b(x,t)\partial_x F(x,t) + \frac{\sigma^2}{2}\partial_{xx}^2 F(x,t) + C(t). \tag{30}$$

Recall the lower reflecting boundary at $x = x^-$, where $F(x^-, t) = 0$ and thus $\partial_t F(x,t)|_{x=x^-} = 0$. We also see that the flux equals 0, so

$$\begin{aligned} J\big(x^-, t\big) &= -b\big(x^-, t\big) f\big(x^-, t\big) + \frac{\sigma^2}{2} \partial_x f(x,t)|_{x=x^-} \\ &= -b(x,t)\partial_x F(x,t)|_{x=x^-} + \frac{\sigma^2}{2} \partial_{xx}^2 F(x,t)|_{x=x^-} \\ &= 0. \end{aligned} \tag{31}$$

Thus, $C(t) = 0$, and we obtain the PDE for $F(x,t)$:

$$\partial_t F(x,t) = -b(x,t)\partial_x F(x,t) + \frac{\sigma^2}{2} \partial_{xx}^2 F(x,t). \tag{32}$$

A.3 Removing the Singularity in the Second-Kind Volterra Equation

The singularity arises because $f^*(x,t|v,s)$ diverges when $v = x$ and $t \to s$. This can be resolved by the method proposed by [39]. Note that the substitution of $\psi(x,t|v,s)$ in Eq. (15) with any function of the form

$$\phi(x,t|v,s) = \psi(x,t|v,s) + \lambda(t) f^*(x,t|v,s) \tag{33}$$

will also satisfy the second Volterra equation, since

$$\begin{aligned} p(t) &= -2\psi(x,t|v,s) - 2\lambda(t) f^*(x,t|v,s) + 2\int_0^t \psi(x_{\text{th}},t|x_{\text{th}},s) p(s)\,ds \\ &\quad + 2\lambda(t) \int_0^t f^*(x_{\text{th}},t|x_{\text{th}},s) p(s)\,ds \\ &= -2\psi(x,t|v,s) + 2\int_0^t \psi(x_{\text{th}},t|x_{\text{th}},s) p(s)\,ds, \end{aligned} \tag{34}$$

where we have applied the first Volterra equation, Eq. (11).

We then set $\phi(x,t|v,s)$ to 0 as $t \to s$ by letting

$$\begin{aligned} \lambda(t) &= -\lim_{t\to s} \frac{\psi(x,t|v,s)}{f^*(x,t|v,s)} \\ &= -\lim_{t\to s} \left[\gamma x - I_{\text{total}}(t) - \frac{\sigma^2}{2V(t|s)} \big(x - M(t|x,s)\big) \right] \\ &= -\gamma x + I_{\text{total}}(t) + \lim_{t\to s} \left[\gamma \frac{x - xe^{-\gamma(t-s)} - \int_s^t I_{\text{total}}(u) e^{-\gamma(t-u)} du}{1 - e^{-2\gamma(t-s)}} \right] \\ &= -\gamma x + I_{\text{total}}(t) + \gamma \lim_{t\to s} \left[\frac{gxe^{-\gamma(t-s)} - I_{\text{total}}(t) e^{-\gamma(t-t)}}{2\gamma e^{-2\gamma(t-s)}} \right] \\ &= \frac{I_{\text{total}}(t) - \gamma x}{2}. \end{aligned} \tag{35}$$

Then we have

$$\phi(x,t|v,s)=\frac{1}{2}f^{*}(x,t|v,s)\left[\gamma x-I_{\text{total}}(t)-\frac{\sigma^{2}}{V(t|s)}\big(x-M(t|v,s)\big)\right], \qquad (36)$$

and the singularity will be removed when $v=x$ and $t\to s$.

References

1. Gilmore RO, Hou C, Pettet MW, Norcia AM. Development of cortical responses to optic flow. Vis Neurosci. 2007;24:845–56.
2. Kanwisher N, Yovel G. The fusiform face area: a cortical region specialized for the perception of faces. Philos Trans R Soc Lond B. 2006;361:2109–28.
3. Smith AT, Singh KD, Williams AL, Greenlee MW. Estimating receptive field size from fMRI data in human striate and extrastriate visual cortex. Cereb Cortex. 2001;11:1182–90.
4. Gattass R, Nascimento-Silva S, Soares JGM, Lima B, Jansen AK, Diogo ACM, Farias MF, Marcondes M, Botelho EP, Mariani OS, Azzi J, Fiorani M. Cortical visual areas in monkeys: location, topography, connections, columns, plasticity and cortical dynamics. Philos Trans R Soc Lond B. 2005;360:709–31.
5. Kanwisher N, Yovel G. The fusiform face area: a cortical region specialized for the perception of faces. Philos Trans R Soc Lond B, Biol Sci. 2006;361(1476):2109–28.
6. Gilmore RO, Hou C, Pettet MW, Norcia AM. Development of cortical responses to optic flow. Vis Neurosci. 2007;24(6):845–56.
7. Freeman J, Simoncelli EP. Metamers of the ventral stream. Nat Neurosci. 2011;14(9):1195–201.
8. Reynolds JH, Chelazzi L, Desimone R. Competitive mechanisms subserve attention in macaque areas V2 and V4. J Neurosci. 1999;19:1736–53.
9. Bundesen C, Habekost T, Kyllingsbæk S. A neural theory of visual attention: bridging cognition and neurophysiology. Psychol Rev. 2005;112(2):291–328.
10. Bundesen C, Habekost T. Principles of visual attention: linking mind and brain. Oxford: Oxford University Press; 2008.
11. Reynolds JH, Heeger DJ. The normalization model of attention. Neuron. 2009;61(2):168–85.
12. Zoccolan D, Cox DD, DiCarlo JJ. Multiple object response normalization in monkey inferotemporal cortex. J Neurosci. 2005;25(36):8150–64.
13. Recanzone GH, Wurtz RH, Schwarz U. Responses of MT and MST neurons to one and two moving objects in the receptive field. J Neurophysiol. 1997;78(6):2904–15.
14. Britten KH, Heuer HW. Spatial summation in the receptive fields of MT neurons. J Neurosci. 1999;19(12):5074–84.
15. Nandy AS, Sharpee TO, Reynolds JH, Mitchell JF. The fine structure of shape tuning in area V4. Neuron. 2013;78(6):1102–15.
16. Busse L, Wade AR, Carandini M. Representation of concurrent stimuli by population activity in visual cortex. Neuron. 2009;64(6):931–42.
17. MacEvoy SP, Tucker TR, Fitzpatrick D. A precise form of divisive suppression supports population coding in the primary visual cortex. Nat Neurosci. 2009;12(5):637–45.
18. Lee J, Maunsell JH. A normalization model of attentional regulation of single unit responses. PLoS ONE. 2009;4:e4651.
19. Li K, Kozyrev V, Kyllingsbæk S, Treue S, Ditlevsen S, Bundesen C. Neurons in primate visual cortex alternate between responses to multiple stimuli in their receptive field. Submitted. 2016.
20. Burkitt AN. A review of the integrate-and-fire neuron model: I. Homogeneous synaptic input. Biol Cybern. 2006;95(1):1–19.
21. Sacerdote L, Giraudo MT. Stochastic integrate and fire models: a review on mathematical methods and their applications. In: Bachar B, Batzel JJ, Ditlevsen S, editors. Stochastic biomathematical models with applications to neuronal modeling. New York: Springer; 2013. p. 99–148. (Lecture notes in mathematics, vol. 2058).
22. Gerstner W, Kistler WM. Spiking neuron models: single neurons, populations, plasticity. Cambridge: Cambridge University Press; 2002.

23. Gerstner W, Van Hemmen JL, Cowan JD. What matters in neuronal locking? Neural Comput. 1996;8(8):1653–76.
24. Paninski L, Pillow JW, Simoncelli EP. Maximum likelihood estimation of a stochastic integrate-and-fire neural encoding model. Neural Comput. 2004;16(12):2533–61.
25. Sirovich L, Knight B. Spiking neurons and the first passage problem. Neural Comput. 2011;23(7):1675–703.
26. Russell A, Orchard G, Dong Y, Mihalas S, Niebur E, Tapson J, Etienne-Cummings R. Optimization methods for spiking neurons and networks. IEEE Trans Neural Netw. 2010;21(12):1950–62.
27. Iolov A, Ditlevsen S, Longtin A. Fokker–Planck and Fortet equation-based parameter estimation for a leaky integrate-and-fire model with sinusoidal and stochastic forcing. J Math Neurosci. 2014;4(1):4.
28. Dong Y, Mihalas S, Russell A, Etienne-Cummings R, Niebur E. Parameter estimation of history-dependent leaky integrate-and-fire neurons using maximum-likelihood methods. Neural Comput. 2011;23(11):2833–67.
29. Ditlevsen S, Lansky P. Parameters of stochastic diffusion processes estimated from observations of first-hitting times: application to the leaky integrate-and-fire neuronal model. Phys Rev E. 2007;76(4):041906.
30. Ditlevsen S, Ditlevsen O. Parameter estimation from observations of first-passage times of the Ornstein–Uhlenbeck process and the Feller process. Probab Eng Mech. 2008;23(2):170–9.
31. Pillow JW, Paninski L, Uzzell VJ, Simoncelli EP, Chichilnisky EJ. Prediction and decoding of retinal ganglion cell responses with a probabilistic spiking model. J Neurosci. 2005;25(47):11003–13.
32. Redner S. A guide to first-passage processes. Cambridge: Cambridge University Press; 2001.
33. Karlin S, Taylor HM. A second course in stochastic processes. vol. 2. Houston: Gulf Pub; 1981.
34. Lansky P, Ditlevsen S. A review of the methods for signal estimation in stochastic diffusion leaky integrate-and-fire neuronal models. Biol Cybern. 2008;99:253–62.
35. Hurn AS, Jeisman J, Lindsay K. ML estimation of the parameters of SDEs by numerical solution of the Fokker–Planck equation. In: MODSIM 2005: international congress on modelling and simulation: advances and applications for management and decision making. 2005. p. 849–55.
36. Paninski L, Haith A, Szirtes G. Integral equation methods for computing likelihoods and their derivatives in the stochastic integrate-and-fire model. J Comput Neurosci. 2008;24(1):69–79.
37. Press WH. Numerical recipes: the art of scientific computing. 3rd ed. Cambridge: Cambridge University Press; 2007.
38. Ditlevsen S, Lansky P. Estimation of the input parameters in the Ornstein–Uhlenbeck neuronal model. Phys Rev E. 2005;71:011907.
39. Buonocore A, Nobile AG, Ricciardi LM. A new integral equation for the evaluation of first-passage-time probability densities. Adv Appl Probab. 1987;19:784–800.
40. Dempster AP, Laird NM, Rubin DB. Maximum likelihood from incomplete data via the EM algorithm. J R Stat Soc, Ser B, Methodol. 1977;39:1–38.
41. Tuckwell HC. Synaptic transmission in a model for neuronal activity. J Theor Biol. 1979;77:65–81.
42. Lansky P, Lanska V. Diffusion approximations of the neuronal model with synaptic reversal potentials. Biol Cybern. 1987;56:19–26.
43. Ditlevsen S, Lansky P. Estimation of the input parameters in the Feller neuronal model. Phys Rev E. 2006;73:061910.
44. Burnham KP, Anderson DR. Model selection and multimodel inference: a practical information-theoretic approach. New York: Springer; 2003.
45. Hastie T, Tibshirani R, Friedman J, Hastie T, Friedman J, Tibshirani R. The elements of statistical learning. vol. 2. New York: Springer; 2009.
46. Stein RB. A theoretical analysis of neuronal variability. Biophys J. 1965;5:173–95.
47. Brunel N, Sergi S. Firing frequency of leaky integrate-and-fire neurons with synaptic current dynamics. J Theor Biol. 1998;195(1):87–95.
48. Moreno R, de la Rocha J, Renart A, Parga N. Response of spiking neurons to correlated inputs. Phys Rev Lett. 2002;89:288101.
49. Moreno-Bote R, Parga N. Role of synaptic filtering on the firing response of simple model neurons. Phys Rev Lett. 2004;92:028102.
50. Ditlevsen S, Samson A. Estimation in the partially observed stochastic Morris–Lecar neuronal model with particle filter and stochastic approximation methods. Ann Appl Stat. 2014;8(2):674–702.
51. Hanson FB. Applied stochastic processes and control for jump-diffusions: modeling, analysis, and computation. vol. 13. Philadelphia: SIAM; 2007.

IV Neural Decoding with Probability Mixing for Leaky Integrate-and-Fire Neurons

To be submitted shortly

Kang Li
Department of Mathematical Sciences, Department of Psychology
University of Copenhagen

Susanne Ditlevsen
Department of Mathematical Sciences
University of Copenhagen

The work based on Paper III and IV won a best poster award in the International Conference of Mathematical NeuroScience 2016.

Neural Decoding with Probability Mixing for Leaky Integrate-and-Fire Neurons

Kang Li, Susanne Ditlevsen
Department of Mathematical Sciences
Department of Psychology
University of Copenhagen

Abstract

Neural coding relates neural observations to external stimuli using computational methods. For encoding we estimate parameters and construct the optimal neural models, and for decoding we infer the stimuli back from observed data. Here we perform neural decoding for a mixture of multiple stimuli using the leaky integrate-and-fire model describing neural spike trains, under the visual attention hypothesis of probability mixing in which the neuron only attends to a single stimulus at any given time. We propose a new algorithm to decode deterministic stimuli and develop various sequential Monte Carlo particle methods to decode stochastic stimuli. The likelihood of spike trains is obtained through the first-passage time probabilities obtained by solving the Fokker-Planck equations. We show by simulation studies that both the deterministic and stochastic stimuli can be successfully decoded, and different particle methods give different performances depending on the scenarios.

keywords: neural decoding, visual attention, probability mixing, spike train, state space model, particle filter

1 Introduction

Neural coding is the science of characterizing the relationship between a stimulus presented to a neuron or an ensemble of neurons, and the neuronal responses. Neural encoding refers to the map from stimulus to response, i.e., how the neurons respond to a specific stimulus. For example, if we can construct an encoding model, it can be used to predict responses to other stimuli. This was the subject of our previous paper (Li et al., 2016a). Neural decoding refers to the reverse map, from response to stimulus, and the challenge is to reconstruct a stimulus, or certain aspects of that stimulus, from the evoked spike train. Neural coding is extensively studied in computational neuroscience.

Our aim here is to decode complicated multiple stimuli from neural spike trains. We combine biophysical spiking neural models with visual attention theories, bridging computational neuroscience and psychology. Following the visual attention model, complicated multiple stimuli are viewed as probability mixtures. For deterministic stimuli, the standard decoding method is unstable and inefficient, and we propose a new cluster decoding algorithm overcoming the deficiencies. For stochastic stimuli, we explore various sequential Monte Carlo methods, for both single neurons and an ensemble of simultaneously recorded neurons. The two visual search mechanisms in psychology, the parallel and the serial processing, are applied after decoding neuron ensembles.

1.1 Neural Decoding

Given neural observations, the decoding process reconstructs the unknown stimulus information encoded by the neural system. Neural decoding plays an important role in understanding the mechanisms of

neurons and the brain. Well-performing algorithms of decoding constitute necessary components of brain-machine interfaces (Lebedev and Nicolelis, 2006; Waldert et al., 2009). Different methods have been explored to study neural decoding. Some methods focus on regression-related approaches building linear models between spike trains and the corresponding stimulus by optimal linear estimation (OLE) (Georgopoulos et al., 1986; Rieke, 1999). Machine learning methods are also employed to stimulus decoding, such as artificial neural networks (Warland et al., 1997), kernel regression (Eichhorn et al., 2003), and a recently developed approach using kernel-based neural metrics (Brockmeier et al., 2014). These methods employ general statistical techniques and omit the specific spike-generating mechanism of neural response. On the other hand, stimulus decoding may directly employ spiking neural models that describe the spike generating mechanisms from stimuli (Koyama et al., 2010; Paninski et al., 2007; Pillow et al., 2011; Truccolo et al., 2005). Various encoding models can be used. Approximate methods using point processes treat the spikes in a spike train as sequential random events, which can be equivalently formulated as generalized linear models (GLM) for model fitting (Truccolo et al., 2005; Kass et al., 2014). Meanwhile, there are also biophysically-motivated methods like integrate-and-fire models, which study the stochastic evolution of the membrane potential. In decoding tasks, these encoding models are used in the posterior distribution to obtain the optimal stimuli. The decoding of constant stimulus can be obtained from the posterior distribution using maximum a posteriori (MAP) or Monte Carlo methods. The decoding of temporal stimulus can be discretized as a sequence of constant decoding tasks, which can be solved by Kalman filtering (Wu et al., 2006) or particle sequential Monte Carlo methods (Paninski et al., 2010; Kelly and Lee, 2003; Brockwell et al., 2004; Shoham et al., 2005).

1.2 Modeling Visual Attention

Stimulus Mixture and Probability Mixing

We define a stimulus mixture to be multiple non-overlapping stimuli inside the receptive field of a neuron. We assume that the neuronal response to a stimulus mixture follows the probability-mixing model (Bundesen et al., 2005; Li et al., 2016a), where the neuron responds at any given time to only one of the single stimuli in the mixture with certain probabilities. This model enables us to accurately perform decoding, i.e., to recover the single stimulus that caused the response.

Neural Explanation of Parallel and Serial Processing

The two opposing visual search mechanisms of parallel and serial processing have been long debated in psychology and empirical behavioral experiments have shown evidence supporting both mechanisms (Bundesen and Habekost, 2008; Nobre and Kastner, 2013; Townsend, 1990; Fific et al., 2008). According to serial processing, multiple objects are processed sequentially by the brain, and according to parallel processing, multiple objects are processed concurrently in parallel. We explain parallel and serial processing from a neural perspective, based on the Neural Theory of Visual Attention (NTVA) (Bundesen et al., 2005) stating that a neuron can only represent a single object at any time. It follows that in serial processing, all neurons in the high level visual cortex must respond to the same single object at any given time. While in parallel processing, neurons can split the attention, responding to different objects at the same time. Here we do not aim to select one mechanism over the other. Rather, we will assume either mechanism, and perform decoding in both cases.

Deterministic and Stochastic Stimulus

We assume two types of stimuli. A stimulus is *deterministic* if it contains negligible noise and can be expressed using a deterministic function, for instance a constant or a sinusoidal stimulus. A stimulus is *stochastic* if it contains strong and inevitable noise apart from a deterministic trend, for example a stimulus described by a stochastic diffusion process. Decoding deterministic stimuli amounts to estimating the parameters defining the deterministic function, and decoding stochastic stimuli requires obtaining parameter estimates as well as a high-dimensional stimulus distribution at all time steps.

Two Attentional Regimes: Fixed Attention and Markov Switching

Consider the case where a neuron is responding to a mixture of multiple stimuli following the probability-mixing model. One possible situation is that the neuronal response is fixed, responding to the same

stimulus component in the mixture during the whole trial. Another possible situation is that the neuron switches between stimuli, only responding to a certain stimulus for some time whereafter it switches to another stimulus, and the switching is random following a Markov chain with certain transition probabilities. In both situations, the neuron can only respond to one single stimulus in the mixture at a time.

1.3 Leaky Integrate-and-Fire Model

The leaky integrate-and-fire (LIF) models are simple diffusion models for the dynamics of the membrane potential in single neurons (Burkitt, 2006; Sacerdote and Giraudo, 2013), the most common being an Ornstein-Uhlenbeck (OU) process with constant conductance, leak potential, and diffusion coefficient. The model can be extended by incorporating post-spike currents with a spike-response kernel function (Kistler et al., 1997). Here we first focus on a bursting response kernel (Gerstner et al., 1996) (rhythmic spiking), then we try two other kernels causing a decay of the spiking rate (adaptation) and a delay of spike formation (refractory period). We have previously used these kernels for studying parameter estimation in LIF models responding to a plurality of stimuli in the same visual attention framework (Li et al., 2016a).

1.4 Method Summary

We apply decoding for stimulus mixtures in a LIF encoding framework with the probability-mixing visual attention model. We consider two cases: 1) decoding simple mixtures of deterministic stimuli with fixed neuronal attention, and 2) decoding complex mixtures of stochastic stimuli with Markov attentional switching. In the deterministic case, we first apply the original Bayesian decoding using maximum a posteriori (MAP), then propose a decoding algorithm specific for mixtures by applying k-means clustering to spike trains. In the stochastic case, we formulate a state-space model and employ different particle filtering and smoothing techniques, approximating the posterior distribution of the stimulus at each discretized point in time. We also investigate two hypotheses of the theory of visual search, namely the serial processing and the parallel processing, for decoding of neuron populations.

2 Encoding Model

The encoding model is the same as used in Li et al. (2016a). We will briefly repeat it here for convenience.

2.1 The Leaky Integrate-and-Fire Model

The evolution of the membrane potential is described by the solution to the following stochastic differential equation:

$$\begin{aligned} dX(t) &= b(X(t),t)dt + \sigma dW(t) \\ &= \left(-a(X(t)-\mu) + I(t) + H(t)\right)dt + \sigma dW(t), \\ X(0) &= x_0 \quad ; \quad X(t_j^+) = x_0 \\ t_j &= \inf\{t > t_{j-1} \,:\, X(t) = x_{th}\} \qquad \text{for } j \geq 1,\ t_0 = 0, \end{aligned} \tag{2.1}$$

where t_j^+ denotes the right limit taken at t_j. The drift term $b(\cdot)$ contains three currents: the leak current $-a(X(t)-\mu)$, where $a > 0$ is the decay rate and μ is the reversal potential, the stimulus driven current $I(t)$, and the post-spike current $H(t)$. The potential $X(t)$ evolves until it reaches the threshold, x_{th}, where it resets to x_0. The membrane potential $X(t)$ is not measured, only the spike times $d = (t_1, t_2, \dots)$ are observed. Thus, the scaling of X is arbitrary, and we can use any values for threshold and reset. We set $x_0 = 0$ and $x_{th} = 1$ such that X is measured in units of the distance between reset and spike threshold. The noise is modelled by the standard Wiener process, $W(t)$, with diffusion parameter, $\sigma > 0$.

The stimulus current $I(t)$ is shaped from the external stimulus $S(t)$ through a stimulus kernel $k_s(t)$; $I(t) = \int_{-\infty}^{t} k_s(t-s)S(s)ds$. The post-spike current arises from past spikes convoluted with a response kernel $k_h(t)$; $H(t) = \int_{-\infty}^{t} k_h(t-s)\mathbb{I}(s)ds$. Here $\mathbb{I}(s) = \sum_{\tau \in d} \delta(s-\tau)$ represents the spike train, where $\delta(\cdot)$ denotes the Dirac delta function.

We assume a stimulus kernel without delay, such that $k_s(t) = \delta(t)$, implying that $I(t) = S(t)$. The response kernel is assumed to be the difference of two exponentials decaying over time,

$$k_h(t) = \eta_1 e^{-\eta_2 t} - \eta_3 e^{-\eta_4 t} \tag{2.2}$$

with four positive parameters, $\eta = (\eta_1, \eta_2, \eta_3, \eta_4)$. By adjusting the parameters, different kernels are obtained. Three types of kernels are used here, described in Table 2.1 and shown in the left panels of Fig. 5.1. In the center panels example spike trains generated from the different kernels and different stimuli are illustrated.

Table 2.1: Characteristics of response kernels used in the encoding model.

Kernel	Description	Parameter	Interpretation
Bursting	first positive, then negative, then vanishing	$\eta_1 > \eta_3$, $\eta_2 > \eta_4$	recent spikes have excitatory effects, accumulation of spikes has inhibitory effects, resulting in rhythmic spiking with bursts
Decaying	first negative, then vanishing	$\eta_1 = 0$, η_3, η_4 small	inhibitory effects are small but long-lasting, making the firing rate decay slowly over time
Delaying	first negative, then positive, then vanishing	$\eta_1 < \eta_3$, $\eta_2 < \eta_4$	recent spikes have inhibitory effects, accumulation of spikes has excitatory effects, preventing short interspike intervals (refractory period)

2.2 Likelihood of a Spike Train

Suppose there are a total of K stimuli inside the receptive field of the neuron, $S = (S^1, \ldots, S^K)$. According to the probability-mixing encoding model, the stimulus-driven current, $I(t)$, follows a probability mixture:

$$I(t) = S^k(t), \text{ with probality } \alpha_k, \tag{2.3}$$

for $k = 1, \ldots, K$, where $\sum_{k=1}^{K} \alpha_k = 1$. Then the probability of a spike train d generated under the exposure of the K stimuli is also a mixture distribution,

$$p(d|S, \alpha) = \sum_{k=1}^{K} \alpha_k p(d|S^k), \tag{2.4}$$

where $p(d|S^k)$ is the probability of generating spike train d from the single stimulus S^k. It equals the product of the probability densities of all spike times within d, where the dependence between spike times is accounted for by conditioning on the history of past spike times, $\mathcal{H}_{t_{i-1}}$,

$$p(d|S^k) = \prod_i g(t_i|S^k, \mathcal{H}_{t_{i-1}}), \tag{2.5}$$

where $g(t_i|S^k, \mathcal{H}_{t_{i-1}})$ is the conditional probability density of spiking at time t given the kth stimulus and the spike history up to the previous spike time t_{i-1}. The probability density $g(\cdot)$ can be obtained from the density of the first-passage time of model (2.1), which we calculate by numerically solving the Fokker-Planck equation; see Appendix A. Assume we repeat the experiment M times with the same stimulus mixture, and thus record M spike trains. Denote by $D = (d^{(1)}, \ldots, d^{(M)})$ the data set of all the measured spike trains. If spike trains are assumed independent, then the likelihood is

$$p(D|S, \alpha) = \prod_{m=1}^{M} p(d^{(m)}|S, \alpha) = \prod_{m=1}^{M} \sum_{k=1}^{K} \alpha_k \prod_{i=1}^{N_m} g(t_i^m|S^k, \mathcal{H}_{t_{i-1}}^m), \tag{2.6}$$

where t_i^m is the ith spike time in the mth spike train, which has N_m spikes, and $\mathcal{H}_{t_{i-1}}^m$ is the spike history of the mth spike train up to the previous spike time.

3 Decoding of Deterministic Stimuli with Fixed Attention

First, we assume that the neuron attends the same stimulus $S^k(t)$ during a single trial, i.e., for the whole period of one observed spike train. Later we will allow for the neuron to change attention during a single trial. Here, we use a constant or a sinusoidal deterministic stimulus. The sinusoidal stimulus is defined by the four parameters function $S^k(t) = s_1^k \sin(s_2^k t + s_3^k) + s_4^k$, which also covers a constant stimulus by setting $s_1^k = 0$.

In decoding tasks, all the parameters related to the encoding model are assumed known, for example inferred from previous experiments. In our case, these parameters are the LIF parameters, namely the decay rate a, the diffusion parameter σ, and parameters of the response kernel η. All parameters related to the stimulus are unknown, namely parameters for the sinusoidal stimulus mixture $s^k = (s_1^k, \ldots, s_4^k), k = 1, \ldots, K$, and the weights of each stimulus component (attentional parameters) $\alpha = (\alpha_1, \ldots, \alpha_K)$. The decoding task is then to infer K and the $(5K)$-dimensional parameter vector $\theta = (s^1, \ldots, s^K, \alpha)$, which completely defines the attended stimulus, since it is deterministic. We will later relax the assumptions, allowing for switching between attended stimuli not only from trial to trial but also during a single trial, as well as stochastic stimulus mixtures.

3.1 Direct Bayesian Decoding with MAP

A standard way of decoding the optimal stimuli is by maximum a posteriori (MAP) estimation (Dayan and Abbott, 2001). The input to the decoding algorithm is spike trains responding to an unknown stimulus, assuming the encoding model known, as well as the distribution of possible stimuli, $p(S)$. Decoding using MAP estimation can in principle easily be applied if the number of mixtures and the probabilities of the single stimuli in the mixture distribution are known. However, when no such detailed information about the stimuli is available, the MAP estimation becomes computationally prohibitive, and is not robust, as we will now explain.

Assume we observe M spike trains D, obtained from multiple non-simultaneous experimental repetitions, under some unknown stimulus mixture. The number of stimuli K in the mixture and their weight values α are assumed unknown. Then the decoding algorithm should not only estimate the stimulus S, but also K giving the dimension of the problem, as well as α as nuisance parameters. For fixed K, the MAP decoding maximizes the joint posterior probability of S and α, given the data D and the inferred encoding model:

$$p(S, \alpha|D) \propto p(D|S, \alpha)p(S, \alpha) \tag{3.1}$$

where $p(D|S, \alpha)$ is given in (2.6). The distribution $p(S, \alpha)$ can be chosen to fit the specific problem. Here we use a uniform distribution. Because K is unknown, the decoding algorithm should be run for multiple $K = 1, 2, \ldots$, and the optimal K is chosen to be the one that minimizes the Bayesian Information Criterion (BIC) value, which chooses the model with the highest likelihood value penalized by the number of parameters.

One problem of the Bayesian MAP decoding for a mixture when K is unknown is the many parameters, which renders the optimization unstable, especially for high values of K and complicated temporal stimuli. Another problem is that the time complexity increases drastically as the number of components becomes large. The computing time T_{MAP} is almost completely determined by the calculation of the conditional probability densities of spike times in (2.5). Define the time needed for the calculation of the probability of one spike train under a single stimulus component to be a time unit. For M spike trains and K stimuli in the mixture, the time for one calculation of the likelihood function in the numerical optimization is then $M \cdot K$ time units. Suppose the optimization procedure for M spike trains and K stimuli needs to calculate the likelihood function $m_{M,K}$ times to converge. Then $m_{M,K}$ increases as M and K increase, because both imply more terms in the likelihood[1]. Thus, for the whole decoding

[1]The number of calculations depends on many other things, for example on the initial values. Here we ignore those refinements, and consider only the case where larger data size M and mixture number K increase the complexity of the likelihood function, which slows down the rate of convergence.

algorithm, where we impose $K \leq k_*$ for some positive integer k_*, the computing time is of the order:

$$T_{MAP} = \mathcal{O}\left(\sum_{K=1}^{k_*} M \cdot K \cdot m_{M,K}\right)$$
$$\geq \mathcal{O}\left(M \cdot m_{M,1} \cdot \frac{k_*^2 + k_*}{2}\right). \tag{3.2}$$

Thus, the Bayesian MAP decoding takes more than $\mathcal{O}(M \cdot m_{M,1} \cdot k_*^2)$ time units, which grows quadratically in the maximum number of considered mixtures, k_*.

3.2 New Decoding Algorithm for Stimulus Mixtures: Cluster Decoding

Here we propose an alternative decoding algorithm for stimulus mixtures, which we will call *Cluster Decoding*. The idea is to avoid the computational complexities caused by the mixture distribution through a clustering algorithm. The settings are the same as for the MAP decoding. First we decode for each spike train d the optimal single stimulus without using probability mixtures, i.e., assuming that $\alpha_j = 1$ for some $j = 1, \ldots, K$, and $\alpha_i = 0$ for $i \neq j$. This yields M decoded stimuli, one for each of the M spike trains. For each spike train $i = 1, \ldots, M$, we define a characteristic vector $\boldsymbol{v}_i$ of size M with each element v_{ji}, $j = 1, \ldots, M$, being

$$v_{ji} = \begin{cases} \ell_{ji}, & \ell_{ji} \geq 0 \\ \exp(\ell_{ji}) - 1, & \ell_{ji} < 0 \end{cases}, \tag{3.3}$$

where ℓ_{ji} is the log-likelihood value of spike train i responding to the decoded stimulus j. The plot of function (3.3) is shown in Figure 3.1. The vectors for all i constitute a matrix, $[\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_M]$. The idea is that if two spike trains were generated by the same stimulus component, they will have similar characteristic vectors. When the log-likelihood value is less than 0, the original likelihood value, $\exp(\ell_{ji})$, will be used to avoid too large negative log-likelihood values.

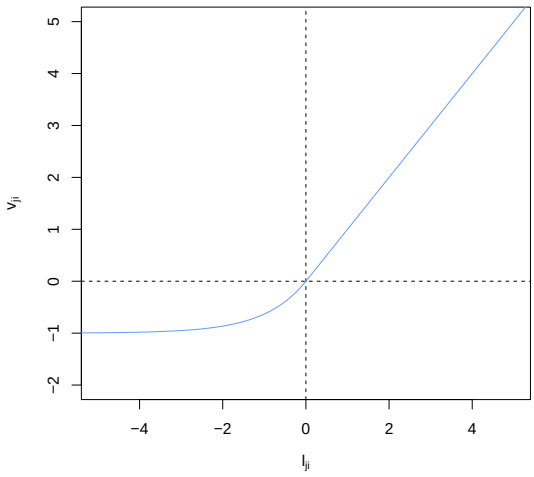


Figure 3.1: Plot of function (3.3). The x-axis is the log-likelihood value of the ith spike train responding to the stimulus decoded from the jth spike train, and the y-axis is the corresponding characteristic value.

Based on the characteristic vectors, the spike trains are then clustered into k categories, using unsupervised clustering algorithms. We employ k-means for clustering using Euclidean distances between characteristic vectors. The best cluster result of k-means is obtained by trying out different initial values and minimizing the cluster variance (Hastie et al., 2009, chpt. 14). The spike trains in the same cluster are then assumed to attend to the same stimulus, and therefore used together to decode the stimulus in the next step. This yields k decoded stimuli, not necessarily a subset of the first M decoded stimuli, providing the estimated stimulus mixture containing k components, $S_k = (S_k^1, S_k^2, \ldots, S_k^k)$, where the subscript k denotes the currently considered number of components.

The clustering is done for different numbers of categories k, and each will be assigned a score in order to obtain the best value of k. We use the BIC value as the score defined by

$$\mathrm{BIC}_k = -2\ell_k + kn_0 \log M_{ISI}, \tag{3.4}$$

where ℓ_k is the log-likelihood value of the optimal stimuli using k categories, n_0 is the base parameter number to describe a single stimulus ($n_0 = 4$ for the sinusoidal stimuli), and M_{ISI} is the total number of interspike intervals (ISIs) in all spike trains (instead of the number of spike trains because the likelihood function is based on the ISI probability density). The k with smallest score will be chosen. The flow of the proposed decoding algorithm is shown in the diagram of Figure 3.2.

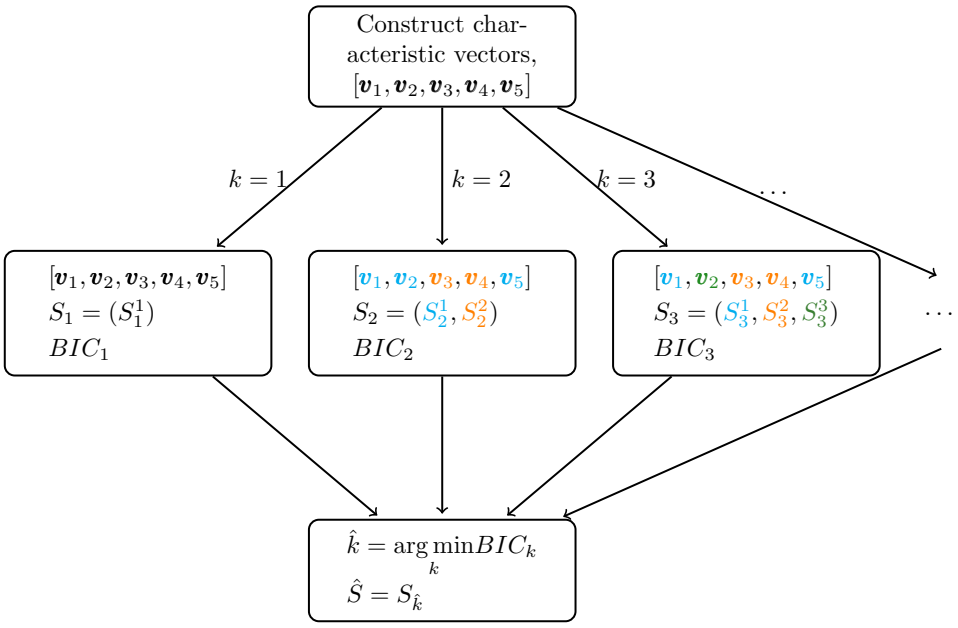


Figure 3.2: Flow diagram of the cluster decoding algorithm. In the blocks, different colors indicate clustered spike trains. In the first step shown in the upper panel, characteristic vectors are constructed by decoding each spike train. For illustration, the number of spike trains is $M = 5$. In the second step, shown in the middle panels, the M spike trains are clustered in $k = 1, 2, 3, \ldots$ clusters. For each cluster, a stimulus component is decoded based on the spike trains within the cluster. The BIC values are also calculated. Finally, in the lower panel, the k that minimizes the BIC value is chosen, and the corresponding stimuli are used as the decoding result.

Note that the classification of spike trains happens before the estimation of the stimuli. In the MAP decoding, the spike trains are classified to some already known or already estimated mixtures, whereas the cluster decoding algorithm first classifies the spike trains into unknown categories, then the categories are estimated.

Comparison of stability and computing times between cluster decoding and direct MAP decoding

The MAP decoding suffered two major problems when applied to probability mixtures: stability and complexity. With respect to stability, the cluster decoding algorithm does not include the probability mixture, so the optimization stage is more stable. However, the additional clustering stage could generate extra variance. As for the complexity, we now analyze the computing time for the new algorithm. During preparation when constructing the characteristic vectors, the time is $M \cdot m_{1,1} + M^2$, the sum of decoding each single spike train and calculating the characteristic vectors. Assume that the number of spike trains, M, is much smaller than any m values, which is most often the case – and if not fulfilled, the computing time is small and not an issue. Then we ignore M^2 and obtain the approximate computing time $M \cdot m_{1,1}$. The computing time for k-means can be ignored compared with the numerical computation of the ISI probabilities. The k-means algorithm clusters the M spike trains into k categories with number of spike trains $M_1, M_2, \ldots, M_k$, respectively, then the decoding time is $\sum_{i=1}^{k} M_i \cdot m_{M_i,1} \leq m_{M,1} \sum_{i=1}^{k} M_i =$

$m_{M,1} \cdot M$. Thus, the total time for the whole cluster decoding algorithm, trying out mixture numbers in the set $\{1, \ldots, k_*\}$, is

$$T_{cluster} \leq \mathcal{O}\left(M \cdot m_{1,1} + k_* \cdot m_{M,1} \cdot M\right). \tag{3.5}$$

Only when $k_* = 1$, i.e., when there is no mixture and only one single stimulus, the speed of the new cluster decoding algorithm is slower than the MAP decoding. In practice, when $k_* = 1$ the cluster decoding and the MAP decoding coincide, since we do not need clustering for one component. For $k_* > 1$, the cluster decoding is expected to be much faster. Furthermore, the difference in computing times grows rapidly with increasing k_*, the maximal number of allowed stimuli in the mixture distribution.

4 Decoding of Stochastic Stimulus Mixtures with Markov Switching

To model more natural stimuli that the neuronal system might be exposed to, we now let the stimulus be stochastic, and allow for the neurons to switch attention between stimulus also within a single trial. We discretize the time interval of a trial in smaller intervals of length v, and assume that the neurons can only switch attention between intervals, but will attend the same stimulus during any of these small intervals. Fiebelkorn et al. (2013) found that sustained attention naturally fluctuates with a periodicity of 4–8 Hz, thus, at most switching attention after $125\,ms$. In the simulations presented later, we set $v = 100\,ms$. Denote by C_n the index of the attended stimulus at the nth time point, $C_n \in \{1, \ldots, K\}, n = 1, \ldots, N$, such that $v\,N$ is the length of the total observation interval, and let S_n denote the stochastic realization of the attended stimulus at the nth time point. In the decoding algorithm, it is assumed that S_n is constant, thus approximating the stochastic stimulus process by a piecewise constant process. Assume the neuron switches attention between two consecutive time intervals following a Markov chain with transition probability matrix (TPM) $\mathbf{\Gamma}$. Denote the elements of $\mathbf{\Gamma}$ by λ_{kl} for $k, l = 1, \ldots, K$. Thus, $\lambda_{kl} = P(C_n = l \,|\, C_{n-1} = k)$ is the probability that at time n the attended stimulus is S^l, given that the neuron attended stimulus S^k at time $n - 1$.

The stochastic stimuli are described by Ornstein-Uhlenbeck (OU) processes. For a mixture of K stimuli $S = (S^1, \ldots, S^K)$, the kth stimulus component is governed by the stochastic differential equation (SDE):

$$dS^k(t) = [\beta^k - S^k(t)]dt + \gamma dW(t), \tag{4.1}$$

where β^k and γ are parameters, and $W(t)$ is a standard Wiener process. Only the drift parameter β^k is stimulus specific, the diffusion parameter γ is assumed the same for all stimuli in the mixture.

The parameters describing the stimulus are unknown, namely γ, $\beta = (\beta^1, \beta^2, \ldots, \beta^K)$ and the TPM $\mathbf{\Gamma}$, so that $\theta = (\gamma, \beta, \mathbf{\Gamma})$. For simplicity, the mixture number K is assumed to be known. If K is unknown, then the algorithm is run with different $k = 1, 2, \ldots$, and the k that minimizes the BIC is chosen. We focus on various Monte Carlo techniques for decoding, including the bootstrap filter, the auxiliary particle filter with parameter learning, fixed-lag and fixed-interval smoothing, etc; see Kantas et al. (2015) for a review of such methods. The goal is not only to decode which stimulus is attended, C_n, but also the stochastic realization of S_n for $n = 1, \ldots, N$. We will present on-line methods, where parameter estimates are updated sequentially as observations become available. We also explore smoothing techniques, where some delay is allowed before the stimulus is reported.

In the following sections, we first establish sequential Monte Carlo methods for decoding of single spike train data, then we discuss decoding of simultaneously recorded spike trains, and include extensions to continuous-time switches, for example following a Poisson process. In most of the simulations, we use the bursting response kernel, see Table 2.1. Finally, we include extensions of other spike response kernels.

4.1 State space model

We use a state-space model to describe the evolution of the stochastic stimuli. The state space is extended to not only include the stimuli S, but also the unknown stimulus-related parameters, which are included

for the construction of the decoding algorithms. The full states are then

$$\begin{aligned} \mathbf{\Gamma}_n &\quad \text{(TPM)} \\ C_n &\quad \text{(index of attended stimulus)} \\ \gamma_n &\quad \text{(common diffusion parameter of all stimuli)} \\ \beta_n = (\beta_n^1, \ldots, \beta_n^K) &\quad \text{(drift parameter of each stimulus)} \\ S_n = (S_n^1, \ldots, S_n^K) &\quad \text{(value of each stimulus)} \end{aligned} \tag{4.2}$$

The subscript n stands for the current time in the state evolution. Note that, even if $\mathbf{\Gamma}, \gamma$ and β are constant in model (4.1), the filters will at each time point update information regarding their value, and thus, they are allowed to change at each time point. Hopefully, they converge towards their true values as more spikes are used in the decoding algorithm. The propagation of states at time n is given by:

$$\begin{aligned} \lambda_{kl,n} &\sim N_{tr}(\lambda_{kl,n-1}, V_\lambda); \quad \sum_{l=1}^{K} \lambda_{kl,n} = 1, \lambda_{kl,n} \geq 0 \\ C_n &\sim \mathbf{\Gamma}(C_{n-1}); \qquad\qquad C_n \in \{1, \ldots, K\} \\ \gamma_n &\sim N_{tr}(\gamma_{n-1}, V_\gamma); \qquad\qquad \gamma_n > 0 \\ \beta_n^k &\sim N(\beta_{n-1}, V_\beta); \\ S_n^k &\sim N(M_n^k, V_n^k); \end{aligned} \tag{4.3}$$

for $k, l = 1, \ldots, K$. The state propagation is explained as follows. All elements of the TPM follow a truncated Gaussian distribution within $(0, 1)$ with variance V_λ, subject to the constraint that rows sum to 1. The index of the attended stimulus is sampled from a multinomial distribution given by row C_{n-1} of the TPM, $\mathbf{\Gamma}(C_{n-1})$. The parameters γ_n and β_n are updated using Gaussian distributions with variance V_γ and V_β, respectively. Since $\gamma_n > 0$, a positive truncated Gaussian distribution is used. The strength of each stimulus, S_n^k, is updated according to the OU model, following a Gaussian distribution with mean $M_n^k = (S_{n-1}^k - \beta_n^k)e^{-\Delta t} + \beta_n^k$ and variance $V_n^k = \gamma_n^2(1 - e^{-2\Delta t})/2$.

The likelihood of the spike train given the parameters is obtained from the encoding model. Let $d_n = (t_1, \ldots, t_{L_n})$ denote the spike train within the duration of the nth interval, where it can happen that d_n is empty if no spikes were fired. Since the intervals are short, we need to take into account boundary effects, i.e., the time from the left boundary of the interval to the first spike, and the time from the last spike to the right boundary. Let T_b and T_e denote the beginning and the end of the interval, respectively. Then if d_n is non-empty, $T_b \leq t_1 < \cdots < t_{L_n} \leq T_e$. The likelihood of d_n is then

$$\begin{aligned} p(d_n | S_n^{C_n}, S_{n-1}^{C_{n-1}}, \mathcal{H}_{T_b}) = &\prod_{l=2}^{L_n} g(t_l | S_n^{C_n}, \mathcal{H}_{t_{l-1}}) && \text{(complete ISIs inside the interval)} \\ &\times g(t_1 | S_n^{C_n}, S_{n-1}^{C_{n-1}}, \mathcal{H}_{T_b}) && \text{(left boundary)} \\ &\times \left[1 - \int_{t_{L_n}}^{T_e} g(\tau | S_n^{C_n}, \mathcal{H}_{t_{L_n}}) d\tau \right] && \text{(survival probability at right boundary)} \end{aligned} \tag{4.4}$$

If there are no spikes in the interval, the likelihood is given by the survival probability:

$$p(d_n | S_n^{C_n}, S_{n-1}^{C_{n-1}}, \mathcal{H}_{T_b}) = 1 - \int_{T_b}^{T_e} g(\tau | S_n^{C_n}, S_{n-1}^{C_{n-1}}, \mathcal{H}_{T_b}) d\tau. \tag{4.5}$$

4.1.1 A bootstrap particle filter

Particle filtering approximates the posterior mean using I particles, where each particle is a sample from the state space at all time points, where we write $S_{n,i}$ for the value of S_n for particle i, and likewise for the other state variables. Then the stimulus at time n is approximated by the empirical distribution of the particles, $\hat{S}_n = \sum_{i=1}^{I} S_{n,i} \bar{w}_{n,i} \approx \int S_n p(S_n | d_{1:n}) dS_n$, where the $\bar{w}_{n,i}$ is the normalized weight of

particle i satisfying $\bar{w}_{n,i} \propto p(S_{1:n,i}|d_{1:n})$ and $\sum_i^I \bar{w}_{n,i} = 1$. Here, $d_{1:n} = (d_1, \ldots, d_n)$ and likewise for $S_{1:n,i}$.

Using the state evolution and the likelihood of observation, a bootstrap particle filter (BF) is formulated in Algorithm 4.1. In this particle filter, each particle has the attended target C_n as a state, and only the information about the attended stimulus is used to calculate the weights. In the first step at $n = 1$, the states are initialized by sampling from uniform distributions. The attention state C is sampled from a discrete uniform distribution containing indexes of all K stimuli, $U\{1, \cdots, K\}$, and the other states are sampled from continuous uniform distributions, whose intervals are given in the Result section.

In this filter and the subsequent filters, we resample particles using systematic resampling, which is conducted as follows. Denote by U_j, for $j = 0, 1, \ldots, I-1$, a total of I random grid variables. A uniform variable $\bar{U}$ is sampled from $U(0, 1]$. The grid variables follow

$$U_j = \frac{j + \bar{U}}{I}, \quad j = 0, 1, \ldots, I - 1. \tag{4.6}$$

The number of duplicates for particle i, $i = 1, 2, \ldots, I$, after resampling is

$$W_i = \left| \left\{ j; U_j \in (\sum_{l=1}^{i-1} \bar{w}_l, \sum_{l=1}^{i} \bar{w}_l], j = 0, 1, \ldots, I - 1 \right\} \right|, \tag{4.7}$$

i.e., the number of grid variables that fall into the ith increment of the cumulative sum of the normalized weights. It follows that $\sum_i^I W_i = I$ and $W_i \geq 0$ for $i = 1, 2, \ldots, I$. Afterwards, we set the weight of all resampled particles to $1/I$.

Algorithm 4.1 Bootstrap particle filter, BF

Initialization: at $n = 1$

1: **for** particle $i = 1, \ldots, I$ **do**
2: Set elements of $\mathbf{\Gamma}$ to $1/K$
3: $C_{1,i} \sim U\{1, \ldots, K\}$; $\gamma_{1,i} \sim U(0, \max_\gamma)$; $\beta_{1,i}^k \sim U(0, \max_\beta)$; $S_{1,i}^k \sim U(0, \max_S)$, $k = 1, \ldots, K$
4: Calculate the weights, $w_i = p(d_1|S_{1,i}^{C_{1,i}})$
5: **end for**
6: Calculate normalized weights, $\bar{w}_i = w_i / \sum_i w_i$

Iteration: for $n = 2, \ldots, N$

7: Resample particles (systematic resampling)
8: **for** particle $i = 1, \ldots, I$ **do**
9: Propagate states: first $\mathbf{\Gamma}_{n,i}$, then $C_{n,i}$, $\gamma_{n,i}$, $\beta_{n,i}$, and finally, $S_{n,i}$, from distributions (4.3)
10: Calculate the weights, $w_i = p(d_n|S_{n,i}^{C_{n,i}}, S_{n-1,i}^{C_{n-1,i}}, \mathcal{H}_{n-1})$
11: **end for**
12: Calculate normalized weights, $\bar{w}_i = w_i / \sum_i w_i$
13: Estimate attended stimulus, $\hat{S}_n = \sum_{i=1}^I \bar{w}_i S_{n,i}^{C_{n,i}}$

4.1.2 Auxiliary particle filter with parameter estimation

In the bootstrap filter, the resampling weights are calculated from the past observation. A more reasonable idea is to calculate the weights based on the current observation. In the auxiliary particle filter (APF) (Pitt and Shephard, 1999), the resampling relies on auxiliary variables, for example, the likelihood of the current observation conditional on the expected states:

$$u_n = w_{n-1} p(d_n|\mu_n^{C_n}), \tag{4.8}$$

where

$$\mu_n = E(S_n^{C_n}|S_{n-1}^{C_n}, \theta_{n-1}). \tag{4.9}$$

The idea is that the resampling based on the current observation provides particles that are distributed more closely to the posterior at the following time point. Therefore, the weights degenerate less and the effective number of particles is larger.

The stimulus model contains fixed hyperparameters θ that are estimated using artificial propagation, which introduces information loss over time (Liu and West, 2001). To overcome this, we propagate the hyperparameter γ_n using kernel smoothing as proposed by Liu and West (2001). The propagation of γ_n follows the Gaussian distribution

$$\gamma_{n+1} \sim N(\psi\gamma_n + (1-\psi)\bar{\gamma}_n, h^2 v_n), \tag{4.10}$$

where $\bar{\gamma}_n$ and v_n are the mean and the variance of the posterior $p(\gamma|d_{1:n})$, evaluated from particles at time n. In practice, we use a truncated version of the Gaussian distribution in (4.10) since the parameter γ is positive. The constants $\psi = (3\delta-1)/2\delta$ and $h^2 = 1-\psi^2$ are evaluated using a discount factor $\delta \in (0,1]$, typically around $0.95-0.99$ (Liu and West, 2001). For the parameters $\mathbf{\Gamma}_n$ and β_n, which depend on each stimulus component, we use the same propagation distribution as before, because of label switching in mixture models (Fearnhead, 2004; Stephens, 2000). It is difficult to evaluate the posterior of elements of $\mathbf{\Gamma}_n$ and β_n because each particle can label each component differently.

The APF with kernel smoothing of parameters is formulated in Algorithm 4.2.

Algorithm 4.2 Auxiliary particle filter with kernel smoothing, APF

Initialization: at $n = 1$

for particle $i = 1, \ldots, I$ **do**
  Set elements of $\mathbf{\Gamma}$ to $1/K$
  $C_{1,i} \sim U\{1, \ldots, K\}$; $\gamma_{1,i} \sim U(0, \max_\gamma)$; $\beta^k_{1,i} \sim U(0, \max_\beta)$; $S^k_{1,i} \sim U(0, \max_S)$, $k = 1, \ldots, K$
end for

Iteration: for $n = 2, \ldots, N$

for particle $i = 1, \ldots, I$ **do**
  Propagate $\mathbf{\Gamma}_{n,i}$ and then $C_{n,i}$
  Calculate $\mu^{C_{n,i}}_{n,i} = E(S^{C_{n,i}}_{n,i} | S^{C_{n,i}}_{n-1,i}, \theta^{C_{n,i}}_{n-1,i})$
  Calculate the first-stage weight, $u_i = w_i p(d_n | \mu^{C_{n,i}}_{n,i}, S^{C_{n-1,i}}_{n-1,i}, \mathcal{H}_{n-1})$
end for
Resample particles (systematic resampling) using $\{u_i\}$, giving a new set of particles $\mathcal{N}$
for particle $j \in \mathcal{N}$ **do**
  propagate $\gamma_{n,j}$ using (4.10), then $\beta_{n,j}$, and finally $S_{n,j}$
  Evaluate the weight, $w_j = p(d_n | S^{C_{n,j}}_{n,j}, S^{C_{n-1,j}}_{n-1,j}, \mathcal{H}_{n-1}) / p(d_n | \mu^{C_{n,j}}_{n,j}, S^{C_{n-1,j}}_{n-1,j}, \mathcal{H}_{n-1})$
end for
Normalize weights and output estimate

4.1.3 Particle filtering with marginal likelihood

In Algorithms 4.1 and 4.2 we use the attended target C as a hidden state, and the weights are evaluated conditional on C. Alternatively, we can marginalize out C in each particle, and use all $S = (S^1, \ldots, S^K)$ to calculate the marginal likelihood as the weight:

$$\begin{aligned} p(d_n|\mathcal{H}_{n-1}) &= \sum_{j=1}^{K} p(d_n|C_n = j, \mathcal{H}_{n-1}) P(C_n = j|\mathcal{H}_{n-1}) \\ &= \sum_{j=1}^{K} \left(\sum_{i=1}^{K} P(C_{n-1} = i|\mathcal{H}_{n-1}) \lambda_{ij,n} \right) p(d_n|C_n = j, \mathcal{H}_{n-1}). \end{aligned} \tag{4.11}$$

Here we suppress the dependency of S for readability in the term $P(d_n|S_n, S_{n-1}, \mathcal{H}_{n-1})$ as well as other relevant terms. Also note that in the marginal probability we depend on all stimuli S instead of a component given by C as in Eq. (4.4). The probability $p(C_{n-1} = i|\mathcal{H}_{n-1})$, conditional on the history up to the previous interval, $\mathcal{H}_{n-1}$, is calculated recursively at each time step by Bayes' theorem:

$$\begin{aligned} P(C_{n-1} = i|\mathcal{H}_{n-1}) &\propto p(d_{n-1}|C_{n-1} = i, \mathcal{H}_{n-2}) P(C_{n-1} = i|\mathcal{H}_{n-2}) \\ &= p(d_{n-1}|C_{n-1} = i, \mathcal{H}_{n-2}) \sum_{k=1}^{K} P(C_{n-2} = k|\mathcal{H}_{n-2}) \lambda_{ki,n-1}. \end{aligned} \tag{4.12}$$

Due to label switching, each particle could label the stimulus components differently. It is then difficult to output the correct results (Fearnhead, 2004). Here we use a simple method. The stimuli in each particle are sorted first, then the posterior mean is calculated for the sorted stimuli. The hope is that after sorting, each particle relabels the components in the same order. The algorithm of a bootstrap particle filter with marginal likelihood is formulated in Algorithm 4.3.

For single spike trains, we cannot decode all components of the stimulus mixture because only one is attended at a time. Therefore marginal likelihood is less appealing for single spike train decoding. However, if we have multiple independent observations at each time point, marginal likelihood will be more appropriate.

Algorithm 4.3 Bootstrap particle filter with marginal likelihood, mBF

Initialization: at $n = 1$

for particle $i = 1, \ldots, I$ **do**
  Set elements of $\mathbf{\Gamma}$ to $1/K$
  $\gamma_{1,i} \sim U(0, \max_\gamma)$; $\beta_{1,i}^k \sim U(0, \max_\beta)$; $S_{1,i}^k \sim U(0, \max_S)$, $k = 1, \ldots, K$
  Calculate the weights, $w_i = p(d_1|S_{1,i})$
end for
Calculate normalized weights, $\bar{w}_i = w_i / \sum_i w_i$

Iteration: for $n = 2, \ldots, N$

Resample particles (systematic resampling)
for particle $i = 1, \ldots, I$ **do**
  Propagate states: first $\mathbf{\Gamma}_{n,i}$, then $\gamma_{n,i}$, $\beta_{n,i}$ and finally $S_{n,i}$ from distributions (4.3)
  Calculate the weights, $w_i = p(d_n|S_{n,i}, S_{n-1,i}, \mathcal{H}_{n-1})$
end for
Calculate normalized weights, $\bar{w}_i = w_i / \sum_i w_i$
Estimate all $S_n = (S_n^1, \ldots, S_n^K)$ using $\hat{S}_n^k = \sum_{i=1}^N \bar{w}_i S_{n,i}^k$ on sorted stimulus components

4.1.4 Auxiliary particle filtering with parameter estimation and marginal likelihood

The idea of APF and parameter learning using kernel smoothing can also be applied to the particle filter with marginal likelihood. We calculate the first-stage weights using marginal likelihood:

$$u_n = w_{n-1} p(d_n|\mu_n), \tag{4.13}$$

where μ_n is the expectation of all components of S_n:

$$\mu_n = E(S_n|S_{n-1}, \theta_{n-1}). \tag{4.14}$$

The calculation of the marginal likelihood $p(d_n|\mu_n)$ follows the same way as in equations (4.11) and (4.12). Again, only the propagation of the common parameter γ_n is done using the kernel smoothing method by Liu and West (2001) due to label switching. The algorithm is formulated in Algorithm 4.4.

Algorithm 4.4 Auxiliary particle filter with kernel smoothing and marginal likelihood, mAPF

Initialization: at $n = 1$

for particle $i = 1, \ldots, I$ **do**
  Set elements of $\mathbf{\Gamma}$ to $1/K$
  $\gamma_{1,i} \sim U(0, \max_\gamma)$; $\beta^k_{1,i} \sim U(0, \max_\beta)$; $S^k_{1,i} \sim U(0, \max_S)$, $k = 1, \ldots, K$
  Calculate the weights, $w_i = p(d_1|S_{1,i})$
end for

Iteration: for $n = 2, \ldots, N$

for particle $i = 1, \ldots, I$ **do**
  Calculate $\mu_{n,i} = E(S_{n,i}|S_{n-1,i}, \theta_{n-1,i})$
  Calculate the first-stage weight, $u_i = w_i p(d_n|\mu_{n,i}, S_{n-1,i}, \mathcal{H}_{n-1})$
end for

Resample particles (systematic resampling) using $\{u_i\}$, giving a new set of particles $\mathcal{N}$

for particle $j \in \mathcal{N}$ **do**
  propagate $\gamma_{n,j}$ using (4.10), then $\beta_{n,j}$, $\mathbf{\Gamma}_{n,j}$ and finally $S_{n,j}$
  Evaluate the weight, $w_j = p(d_n|S_{n,j}, S_{n-1,j}, \mathcal{H}_{n-1})/p(d_n|\mu_{n,j}, S_{n-1,j}, \mathcal{H}_{n-1})$
end for

Normalize weights and output estimate based on sorted stimulus components

4.2 Decoding From Multiple Spike Trains

Now we consider multiple neurons simultaneously recorded in one trial providing multiple spike trains. Since stochastic stimuli contain inevitable noise and are not reproducible by repetitions in real applications, all estimates of the stimuli depend entirely on the spike trains from one trial. Thus, the attentional behavior of the simultaneously recorded neurons is of great importance for understanding the full information of stimuli.

For multiple, simultaneously recorded spike trains we consider two opposing hypotheses for visual search in neuronal attention, namely the serial and the parallel processing. In serial processing, all stimuli are processed sequentially. The neural interpretation is that all neurons attend to the same stimulus at the same time, and switch to another all together. Therefore, all spike trains would have similar spiking patterns. On the contrary, in parallel processing, stimuli are processed in parallel. Each neuron attends its own stimulus and can switch to another stimulus independently of the other neurons. The spike trains are then distinct from each other.

For stimulus decoding using particle methods, serial processing essentially means an increase of the sample size at each time point, making the decoding more accurate. However, it only decodes the attended stimulus at any time, and the data contain no information about the other stimuli at that time point. For M spike trains, $D = \{d^{(m)}|m = 1, \ldots, M\}$, the likelihood function with the serial processing assumption within a small interval is then

$$p(D_n \,|\, S_n^{C_n}, S_{n-1}^{C_{n-1}}, \{\mathcal{H}_{n-1}^{(m)}\}_{m=1,\ldots,M}) = \prod_{m=1}^{M} p(d_n^{(m)}|S_n^{C_n}, S_{n-1}^{C_{n-1}}, \mathcal{H}_{n-1}^{(m)}). \tag{4.15}$$

The right hand side is evaluated using expression (4.4).

In parallel processing each spike train has its own attended stimulus. Stimulus decoding can then estimate multiple components of the mixture. Each single stimulus is decoded independently using Algorithms 4.1 or 4.2, which produces estimates of each neuron's attended stimulus at each time point, and then the results from all spike trains give an empirical distribution of the stimulus mixture at each time point. Then we run cluster analysis at each time point in one-dimensional space based on the estimates of stimuli. Since there are outliers (see Result section), we apply k-medoids clustering (Kaufman and Rousseeuw, 2009; Hastie et al., 2009, chpt. 14) using the square root of Euclidean distance as the dissimilarity measure. The k-medoids clustering is preferred over k-means because k-medoids can be more robust against outliers (Hastie et al., 2009). Furthermore, the square root of the Euclidean distance puts less weight on extreme outliers than the Euclidean distance. Finally, we use the median of each cluster as the estimate for each component of the stimulus mixture.

Another decoding method for parallel processing is to exploit the marginal likelihood since we have multiple independent observations. Now each particle can decode all stimulus components, and all decoded components will be used for the output estimation. When calculating the weights, we need the likelihood, which is the product of the marginal likelihoods of all spike trains:

$$p(D_n|S_n, S_{n-1}, \{\mathcal{H}_{n-1}^{(m)}\}_{m=1,\dots,M}) = \prod_{m=1}^{M} p(d_n^{(m)}|S_n, S_{n-1}, \mathcal{H}_{n-1}^{(m)}), \tag{4.16}$$

and the right hand side is evaluated using equation (4.11).

Adjusting auxiliary variables for large data size. In Algorithms 4.2 and 4.4 based on APF for population decoding, the auxiliary variables are calculated using the likelihood, which can take extreme values if the sample size is large, e.g., when the data contain multiple spike trains. The consequence is that only few particles with extreme weight values survive the resampling, reducing the posterior variance and leading to the degeneracy of parameter learning (Rios and Lopes, 2013; Carvalho et al., 2010). To slow down the degeneracy, we use the geometric mean of the likelihood value over the number of spike trains, $\widetilde{p}(D_n|\mu_n, S_{n-1}, \{\mathcal{H}_{n-1}^{(m)}\}_{m=1,\dots,M}) = \left(\prod_{m=1}^{M} p(d_n^{(m)}|\mu_n, S_{n-1}, \mathcal{H}_{n-1}^{(m)})\right)^{1/M}$, when calculating the auxiliary variables in Algorithms 4.2 and 4.4.

4.3 Particle smoothing

The above online algorithms return estimates of stimuli by approximating the filtering probability conditional on the observation up to the current time, $p(S_{1:n}|D_{1:n})$. An alternative is offline methods that make use of future observations or the entire data set when estimating the stimuli at a certain time point. This posterior is referred to as the smoothing distribution. A full-length smoothing reports the posterior of the stimulus at any time n conditional on all observations over $1:N$, $p(S_n|D_{1:N})$, but we can also apply partial smoothing when only certain delays are allowed. Say we need to report the stimulus after a delay of n^* time points, then we can decode the stimulus at time n using partial smoothing, $p(S_{n-n^*}|D_{1:n})$. Thus, filtering does real-time online decoding, while smoothing does semi-online decoding with some delay or offline decoding after the full observation. Here we pursue the semi-online decoding allowing a delay of n^* before reporting the stimulus. Two smoothing methods have been tried, the fixed-lag smoothing and the fixed-interval smoothing (Doucet et al., 2000).

In the fixed-lag smoothing estimates, we simply marginalize the filtering probability $p(S_{1:n}|D_{1:n})$ at time $n - n^*$, estimating S_{n-n^*} with the current weights when calculating the posterior mean, $\hat{S}_{n-n^*} = \sum_{i=1}^{N} S_{n-n^*,i}\bar{w}_{n,i}$. This requires additional memory to store the history of S.

In fixed-interval smoothing we apply the forward-filtering backward-smoothing algorithm, and calculate the smoothing distribution $p(S_{n-n^*}|D_{1:n})$ for the desired time $n - n^*$ instead of using the joint filtering distribution $p(S_{1:n}|D_{1:n})$. The smoothing distribution $p(S_{n-n^*}|D_{1:n})$ is obtained using recursive backward smoothing from n after a full forward filtering up to n (Doucet et al., 2000); see Appendix B.

4.4 Continuous-time switching

All the decoding algorithms assume that neuronal attention is fixed within intervals of duration 100 ms, and only switches between two intervals. To test how strong this assumption is, we also simulate spike trains with continuous-time switching, i.e., the attentional switching does not need to take place exactly between two intervals. One example is that the switching follows a Poisson process, which is what we use in this paper. If this is the case, then decoding with discretization will be less accurate. However, if the switching rate is sufficiently low such that the average inter-switch interval is much longer than a discretized interval, the Poisson attentional switching is well approximated by the approach based on discretization.

A fixed TPM on discretized time points approximates the Poisson switching model well due to the memoryless property of the Poisson process. However, since the TPM is updated at each time point as

latent states, the model is easy to extend to non-Poissonian switching allowing for memory effects by adapting the TPM for a specific model. This is not pursued here.

5 Results

Throughout the following examples, we use the parameters for the LIF encoding model shown in Table 5.1. Figure 5.1 illustrates some realizations of spike trains generated from the encoding model using different response kernels and stimuli.

Table 5.1: Parameters of the LIF encoding model used in the simulations.

Parameter	Value	Explanation
a	100	decay rate in LIF model
x^-	0	reflecting boundary of Fokker-Planck equation
x_{th}	1	firing threshold of potential
x_0	0.4	reset potential
μ	0.5	resting potential
σ	1	diffusion parameter in LIF model
η_{burst}	(50, 25, 40, 15)	burst response kernel
η_{decay}	(0, 0, 2, 0.5)	decay response kernel
η_{delay}	(20, 8, 50, 15)	delay response kernel
Δt	0.002s	time discretization in numerical solution
Δx	0.02	potential discretization in numerical solution
n^*	10 intervals	time delay for particle smoothing (10 intervals = 1s)

5.1 Deterministic Mixtures

The MAP decoding and the cluster decoding algorithm for mixtures are applied to constant and sinusoidal mixtures using the bursting response kernel. In all the following results for deterministic mixtures, ten spike trains of length 5s are simulated, illustrating $M = 10$ (non-simultaneous) recordings of spike trains responding to the same stimuli in a real experiment. The simulation is repeated 20 times, and each simulated data set is used for the decoding analysis. In the simulation study, for the maximal mixture number k_* we always use the true mixture number plus one, $k_* = K + 1$. We try out the possible numbers in $\{1, 2, \ldots, k_*\}$ and select the one with the minimum BIC.

The decoding performance can be evaluated by comparing the estimation of stimulus strength with the true strength. In addition, the classification of spike trains for the stimulus mixture can also be evaluated. We define two statistics. Denote by P_n the ratio of the number of decodings that correctly predict the true mixture number, over the total number of decodings. Denote by P_c the ratio of the decodings that make correct clustering of all spike trains, over the number of decodings with correct mixture number calculated in the previous step. These statistics are numbers between 0 and 1, and the closer to 1, the more correct the classification.

Figure 5.2 shows the decoding using the MAP (upper panel) and the cluster algorithm (lower panel) for constant stimuli containing 2 or 3 components, with the statistics P_n and P_c shown in the top-left corner. The true stimuli are shown as lines in different colors, and the estimates are plotted as histograms. Performances of the two algorithms are comparable, for both spike train classification and stimulus estimation, even if the cluster algorithm performs slightly better, as indicated by the P_n and P_c statistics. When the two stimuli are very close (middle column panel), the classification becomes difficult, but we still obtain good estimation of the strength. Table 5.2 lists the stimulus and attentional weight parameters used for simulation.

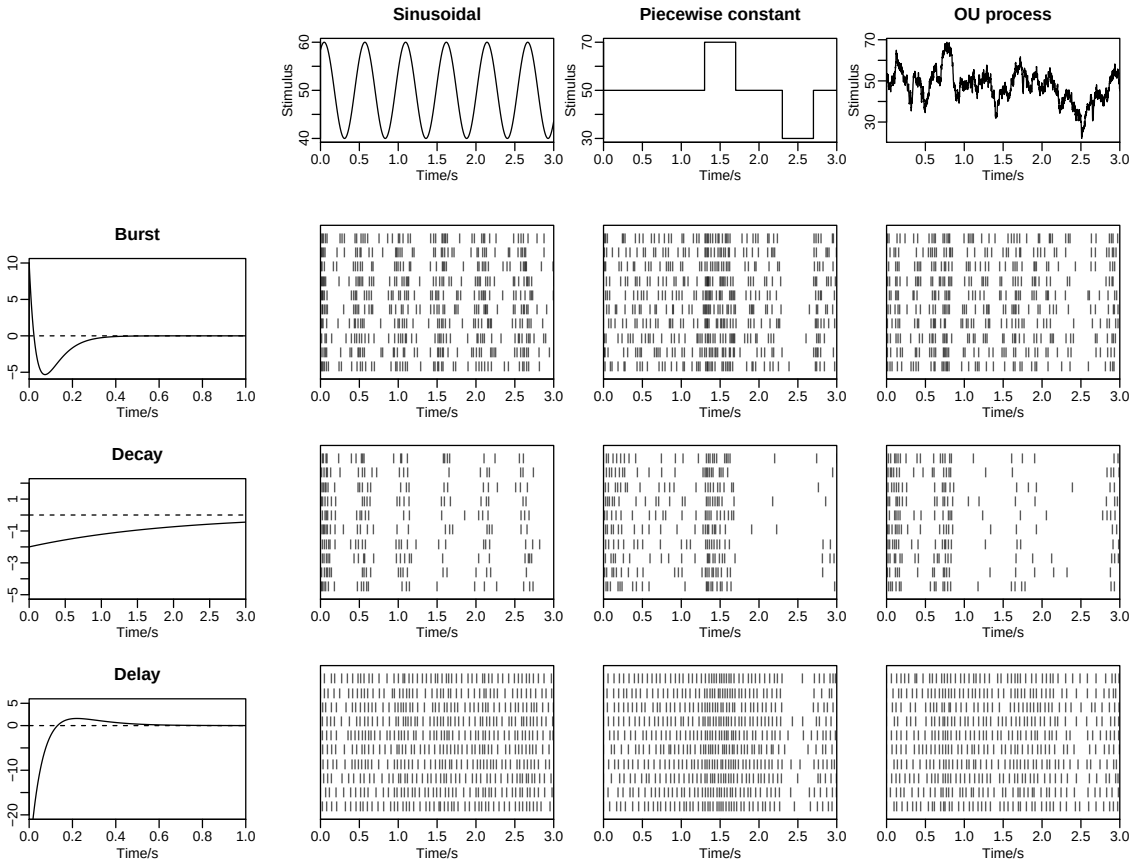


Figure 5.1: Realizations of spike trains. The left panel shows the three response kernels. The top panel shows different types of stimuli. Then spike trains are shown in each combination of response kernel and stimulus.

Table 5.2: Stimulus parameters used in Figure 5.2.

Panel	Left		Middle		Right		
Stimulus index	1	2	1	2	1	2	3
Strength	66	71	70	71	66	56	71
Weight	0.4	0.6	0.4	0.6	0.3	0.2	0.5

Now we conduct decoding of both algorithms for sinusoidal stimuli. In many cases, the decoding of sinusoidal stimuli suffers from identification problems, because both the stimuli and the bursting kernel will cause oscillations in the spiking dynamics. The numerical optimization can easily choose stimuli that can generate similar spiking patterns but are far from the true stimuli. Thus, we also tried to fix the frequency (the parameter s_2) of the stimuli, assuming it known.

Figure 5.3 shows the decoding using the MAP and the cluster algorithms for sinusoidal stimuli. The left three column panels use mixtures of two stimuli with different parameters while the right panel uses a mixture of three. In this figure we compare different algorithms and settings, where the MAP decoding without fixing s_2 is shown in the top panel, the MAP decoding fixing s_2 in the middle upper panel, the cluster decoding without fixing s_2 in the middle lower panel, and finally the cluster decoding fixing s_2 in the bottom panel. The decoding performance for both algorithms is much improved when the frequency parameter s_2 is fixed. Before fixing s_2, the MAP decoding tends to make too fluctuating estimates while the cluster decoding tends to make too flat estimates, and both work poorly. Comparing the results after fixing s_2, we find the cluster decoding achieves better accuracy and stability than MAP, for both stimulus strength estimation and spike train classification. Focusing on the different stimulus settings in the upper middle and the bottom panels, for two similar stimuli we again see poor classification of

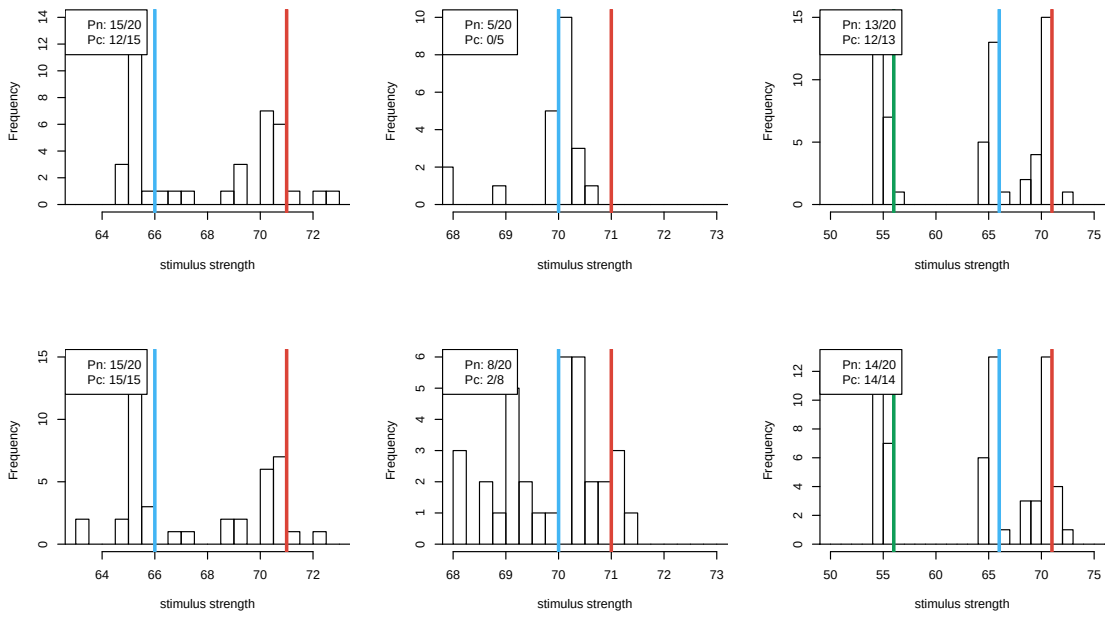


Figure 5.2: The MAP and the cluster decoding algorithms applied to constant stimulus mixtures. The upper panel shows the results of the MAP decoding and the lower panel shows the cluster decoding. Blue, red, and green solid lines show the true stimuli. Histograms show the decoding results. The P_n and P_c ratios are classification statistics; see the main text. The closer these numbers are to 1, the more correct the classification.

spike trains but accurate estimation of the stimulus strength and shape. By contrast, when two stimuli are different, classification can perform better even though we cannot always obtain accurate stimulus estimation (left figure in the lower middle panel). The parameters used in the simulation are shown in Table 5.3.

Table 5.3: Stimulus parameters used in Figure 5.3.

Panel	Left		Left middle		Right middle		Right		
Stimulus Index	1	2	1	2	1	2	1	2	3
s_1	20	10	10	12	10	12	20	20	10
s_2	8	8	8	8	8	8	8	8	8
s_3	0	1	1	0	0	0	0	2	1
s_4	60	60	60	60	60	60	60	60	60
Weight	0.4	0.6	0.4	0.6	0.5	0.5	0.3	0.4	0.3

Finally, we compare the efficiency of the MAP and the cluster decoding algorithms in Figure 5.4, where we show the elapsed time in seconds for a decoding estimation. We merge the simulation studies shown above into four groups for both constant and sinusoidal stimuli: MAP decoding for two stimuli (MAP2) and three stimuli (MAP3), cluster decoding for two stimuli (Cluster2) and three stimuli (Cluster3). For example, Cluster2 for sinusoidal stimuli includes the simulations with the cluster decoding algorithm using all three types of parameter settings, both with and without fixing s_2. The figure clearly shows that the cluster decoding is significantly faster, and it also suffers much less from increasing the number of stimuli.

The performance of MAP could potentially be improved by employing the Expectation Maximization (EM) algorithm rather than using the marginal likelihood here. However, the EM iterations will spend much longer time than the marginal likelihood method, considering especially the unknown number of

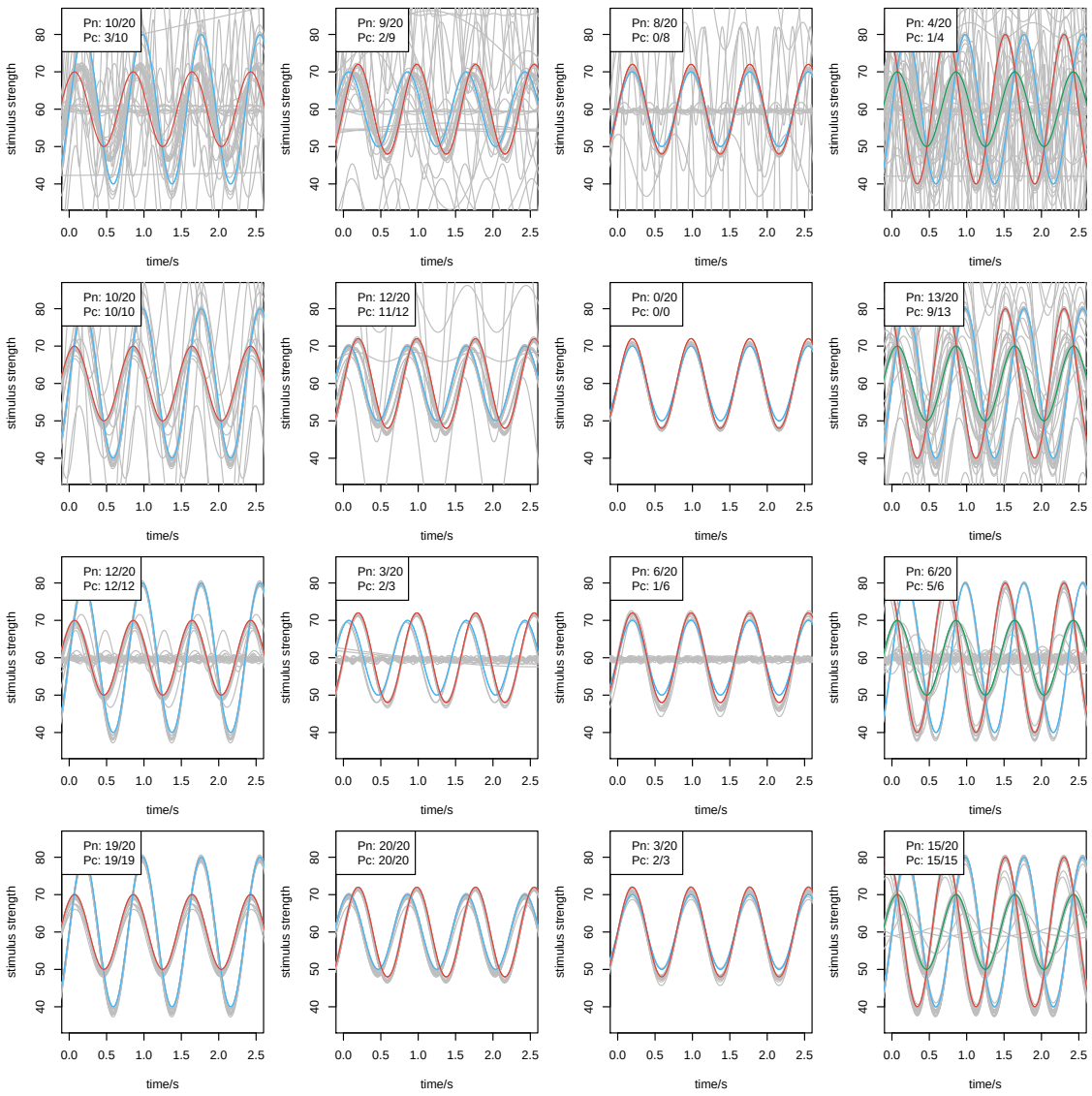


Figure 5.3: The MAP and the cluster decoding algorithms on sinusoidal stimuli with 2 *or* 3 *components. Blue, red, and green solid lines show the true stimuli. Light grey lines show the decoding results. The four row panels from top to bottom represent respectively MAP without fixing s_2, MAP fixing s_2, cluster decoding without fixing s_2, and cluster decoding fixing s_2. Different column panels represent different stimulus parameters shown in Table 5.3. The P_n and P_c ratios are classification statistics; see the main text. The closer these numbers are to 1, the more correct the classification.*

stimuli K. We do not pursue it here.

5.2 Stochastic Mixtures

In each of $M = 10$ trials we simulate K new stimuli according to the OU model. Each spike train is generated using the simulated stimuli within the period $[1, 6]$s (a period of 5s after 1s burn-in). The time step size of generating the stimulus is 0.01s. We then decode the stochastic mixtures from the spike trains.

The root mean squared deviation (RMSD) between true and decoded stimuli is used to evaluate the performance. Since the stochastic stimuli are simulated with steps of 0.01s and we approximate the

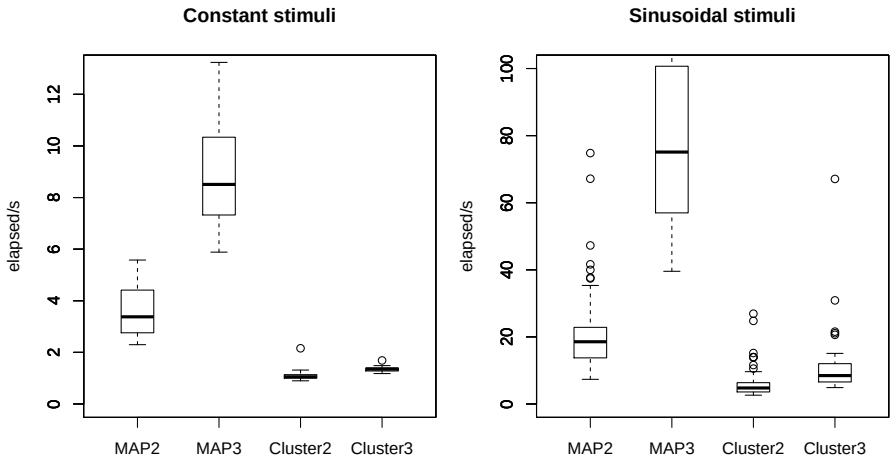


Figure 5.4: Boxplots of elapsed time for decoding simulations. All simulation studies in Figures 5.2 and 5.3 are included and merged into groups. The labels on x-axis represent the decoding algorithm and the number of stimuli; see the main text.

stochastic process with a discretized piecewise constant function with steps of 0.1s, the RMSD will always be greater than 0. To take this into account, a relative root mean square deviation (rRMSD) is used to measure the decoding accuracy:

$$\text{rRMSD} = \frac{\sqrt{\frac{1}{10N}\sum_{n=1}^{N}\sum_{l=1}^{10}(\hat{S}_n - S_{n,l})^2}}{\sqrt{\frac{1}{10N}\sum_{n=1}^{N}\sum_{l=1}^{10}(\hat{S}^*_n - S_{n,l})^2}}. \tag{5.1}$$

where N is the number of discretized intervals, $S_{n,l}$ denotes the true stimulus, different for each n and l, $\hat{S}_n$ is the prediction of the stimulus and $\hat{S}^*_n$ is an artificial stimulus that minimizes the RMSD, $\hat{S}^*_n = \frac{1}{10}\sum_{l=1}^{10} S_{n,l}$. Then the best achievable value of rRMSD is 1.

The effective sample size (ESS) measures the weight degeneracy of the sequential Monte Carlo methods. The ESS at time n for I particles is given by

$$(N_{eff})_n = \frac{1}{\sum_{i=1}^{I}(\bar{w}_{n,i})^2}. \tag{5.2}$$

If the weights are evenly spread, then $(N_{eff})_n = I$ takes its maximum value. The smaller ESS is, the less effective are the particles in representing the distribution.

The performance of different particle methods are compared using rRMSD, ESS and the trace of parameter learning over time.

We tried stimulus mixtures of $K = 1, 2$ and 3 components. A mixture of 1 component implies that the neuron's attention is fixed at the single stimulus. We set the TPM for the mixture of two to

$$\mathbf{\Gamma}_2 = \begin{bmatrix} 0.8 & 0.2 \\ 0.2 & 0.8 \end{bmatrix}, \tag{5.3}$$

and for the mixture of three to

$$\mathbf{\Gamma}_3 = \begin{bmatrix} 0.5 & 0.2 & 0.3 \\ 0.3 & 0.5 & 0.2 \\ 0.2 & 0.3 & 0.5 \end{bmatrix}. \tag{5.4}$$

Table 5.4 shows the β parameters used for each component and the common γ values for each mixture.

Table 5.4: Stimulus parameters, β and γ, of the stochastic stimulus mixtures using OU processes.

Mixture number	one	two		three		
Stimulus index	1	1	2	1	2	3
β	70	65	75	60	70	80
γ	20	20		20		

During initialization, the values of γ, β and the stimulus strength S are uniformly sampled from $U(0, 40)$, $U(0, 200)$ and $U(0, 200)$, respectively. The variances for the algorithmic updating of Γ, γ and β are $V_\lambda = 0.01$, $V_\gamma = 1$ and $V_\beta = 4$, respectively. For the AFP algorithm with kernel smoothing, we use $\delta = 0.95$. Throughout the experiments, the number of particles is $I = 500$. The delay time for particle smoothing is $n^* = 10$ intervals equal to 1s.

To represent various types of algorithms in single neuron and population decoding, we use the notation explained as follows. An algorithm is denoted by a unified term

$$\texttt{\{,i,m\} \{BF,APF\} \{,g\} - \{F,lag,FB\}} \ . \tag{5.5}$$

A possible prefix `i` or `m` stands for the individual decoding or the marginal likelihood decoding in parallel processing. The main term BF or APF means the filtering algorithm. A possible suffix of `g` stands for using the geometric mean for the likelihood value. Finally, the last part represents whether we use filtering (F), fixed-lag smoothing (lag) or fixed-interval smoothing with the forward-backward algorithm (FB).

5.2.1 Single Spike Train

In single spike train experiments, the decoding trials are repeated 50 times. In each trial new stimuli are generated and one spike train is simulated following the stimulus mixture. Then all decoding is conducted only on this single spike train.

Figure 5.5 illustrates decoding examples for single spike trains using the online BF. Shown in the figure are single spike trains and the corresponding decoding results (left) together with kernel smoothing approximations of the posterior distributions (middle and right) at selected time points (dashed lines in left figures), using stochastic mixtures of 1, 2 and 3 components in the upper, middle and lower row panels. In Figure 5.6 are shown decoding examples for two stimuli, using online filtering, fixed-lag smoothing and fixed-interval smoothing with a delay of $n^* = 10$ for the upper, middle and lower row panels. The same spike train is used for the three methods.

Boxplots of rRMSD values from 50 repetitions are shown in Figure 5.7. Various combinations of three filtering methods (online filtering, fixed-lag smoothing and fixed-interval smoothing), two particle methods (BF and APF) and three component sizes ($K = 1, 2$ and 3) are tried. The decoding performance tends to be better when there are less number of stimulus components and when we use delayed smoothing rather than online filtering. The benefit of APF is not observed for $K = 1$ and $K = 2$, but becomes notable when $K = 3$.

Figure 5.8 shows the ESS of different particle methods for different number of components. The ESS is calculated for all time steps, so the boxplots cover 2500 samples for all 50 repetitions at all 50 time steps. The ESS of APF outperforms BF only when $K = 3$. When $K = 2$, the medians of APF and BF are comparable but the variance of BF is smaller. When K gets larger, the weight degeneracy quickly becomes a problem for BF, but the weights are less sensitive to K for APF. This finding here corresponds to the finding in the rRMSD plots in Figure 5.7.

Finally, in Figure 5.9 we show examples of the time trajectory of parameter learning for γ, the diffusion parameter in the OU model of the stimuli. Parameter learning converges faster using APF when there is more than one stimulus, but the learning is not as fast as the parameter degeneracy (observed and explained in the following population decoding).

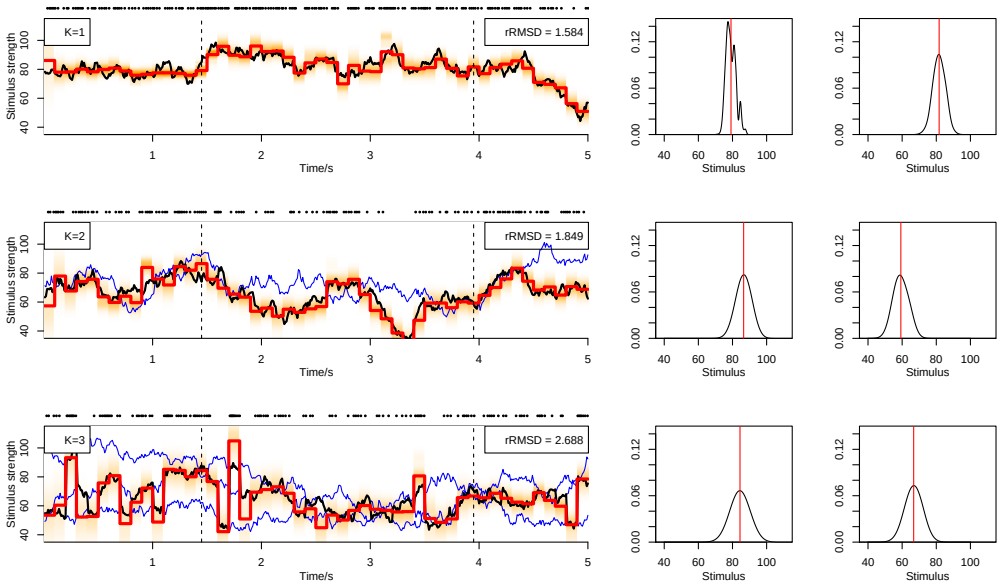


Figure 5.5: Decoding of stochastic stimulus mixtures using BF with filtering from a single spike train responding to stimulus mixtures containing 1 (upper panel), 2 (middle panel) or 3 (lower panel) components. Blue curves show all the stimulus components in the mixture, and the black curve switching between the blue curves indicate the attended stimulus. Red piecewise-constant lines show the decoding results as the posterior mean, with each constant interval being 100 ms long. The light red shaded area indicates the posterior distribution at each time step. The spike train is plotted above each decoding figure as sequences of dots. The rRMSD values are shown on the top-right corner of each figure. In the right side of each panel, the empirical posterior distributions at selected time points indicated by dashed lines in the left panels are shown, computed from weighted kernel density smoothing using the particles. The red vertical line indicates the posterior mean, i.e., the decoding estimates shown in the left panels.

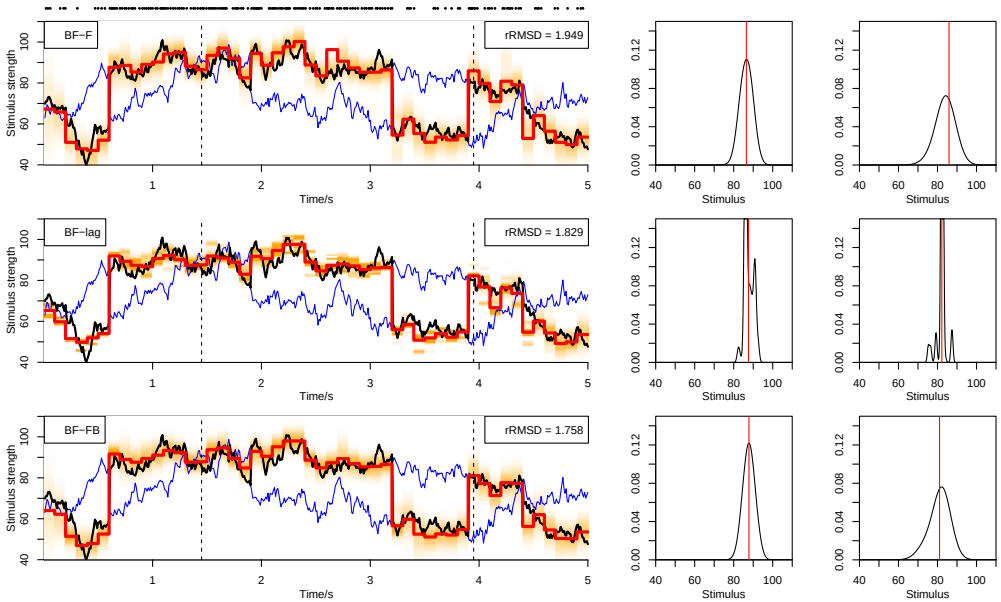


Figure 5.6: Decoding of stochastic stimulus mixtures from a single spike train by BF with filtering, BF-F (upper panel), fixed-lag smoothing, BF-lag (middle panel) and fixed-interval smoothing, BF-FB (lower panel). The three panels show the decoding of the same spike train. See caption of figure 5.5 for explanation.

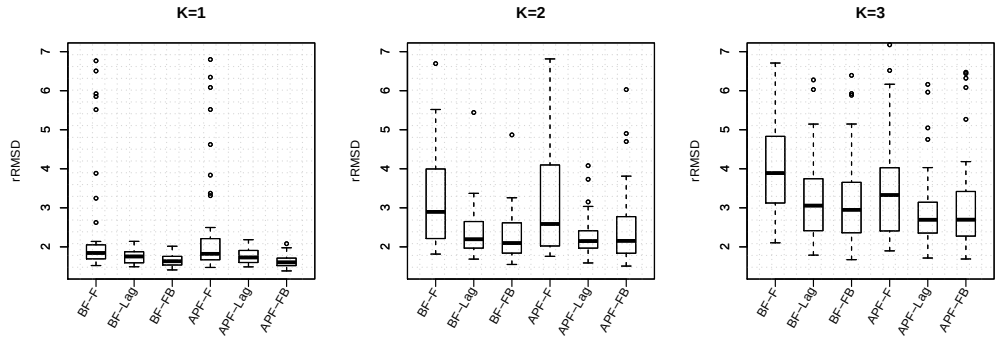
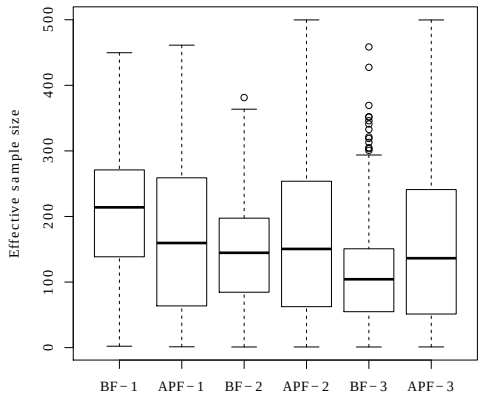


Figure 5.7: The rRMSD values of decoding stochastic mixtures with $K = 1$, 2 and 3 components using different particle methods, calculated from 50 repetitions. In the labels of the x-axis, F: filtering, Lag: fixed-lag smoothing, FB: fixed-interval smoothing using the forward-backward algorithm. For example, APF-Lag means using APF and reporting estimates using fixed-lag smoothing.

Figure 5.8: ESS of BF and APF with $K = 1, 2, 3$ stimuli, shown in boxplots for 2500 samples of 50 repetitions at 50 time steps. The labels in the x-axis show the number of stimuli. For example, APF-2 means using APF with 2 stimuli.



5.2.2 Multiple Spike Trains

In population decoding of multiple spike trains, we use a mixture of two stimuli also of length 5s. In each trial we simulate new stimuli and 20 simultaneous spike trains, and we conduct 50 repetitions. Population decoding assumes either serial processing or parallel processing.

A decoding example following serial processing is shown in figure 5.10. The figure compares filtering, fixed-lag smoothing and fixed-interval smoothing, all using BF. In the top of the figure are shown the 20 spike trains used for decoding, which follow similar spiking patterns because all of them attend to the same stimulus assuming serial processing.

A decoding example following parallel processing is shown in Figure 5.11. Spike trains can be quite distinct due to different attended stimuli. All stimuli can be simultaneously decoded at each time point. Two decoding methods are used. First we apply individual decoding of each spike train, obtaining 20 estimates which are clustered into two categories. The median of each category is the final estimate. The histograms to the right show how the 20 estimates are distributed at two selected time points. Sometimes one category contains less estimates than the other. This occurs when the two components are different in strength and most spike trains happen to attend to one stimulus component, or when the two components have similar strength and outliers form a second category. A category with few estimates is marked by a red color and stars if $\leq 5\%$ of the total size. Starred estimates should be ignored to avoid the effect of outliers and the other category will be used as the decoding result for both components. The stars at 4.9s in the middle panel captures a situation where the two stimuli are

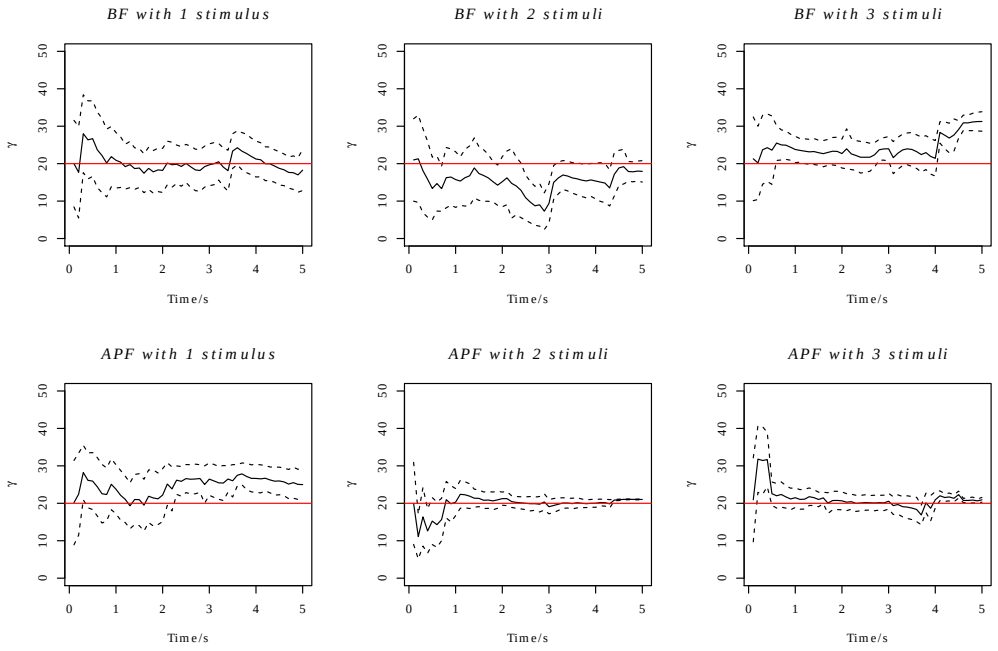


Figure 5.9: Examples of parameter learning of γ over time. The solid line is the mean of 500 particles, and dashed lines show $\pm$ the standard deviation. The red lines are the true values.

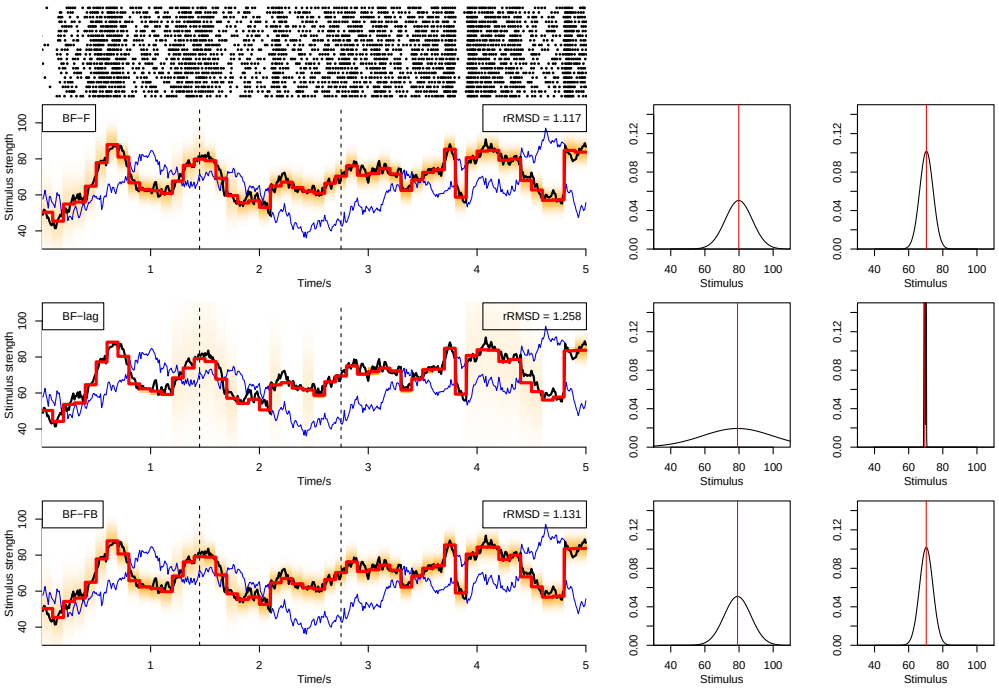


Figure 5.10: Decoding from 20 spike trains on a stimulus mixture with two components assuming serial processing. Decoding is done by BF with online filtering (upper middle panel), fixed-lag smoothing (lower middle panel) and fixed-interval smoothing (lower panel).

close. The second method for parallel population decoding is to use marginal likelihood. All stimulus components are decoded due to multiple independent observations at each time point, shown in the lower panel.

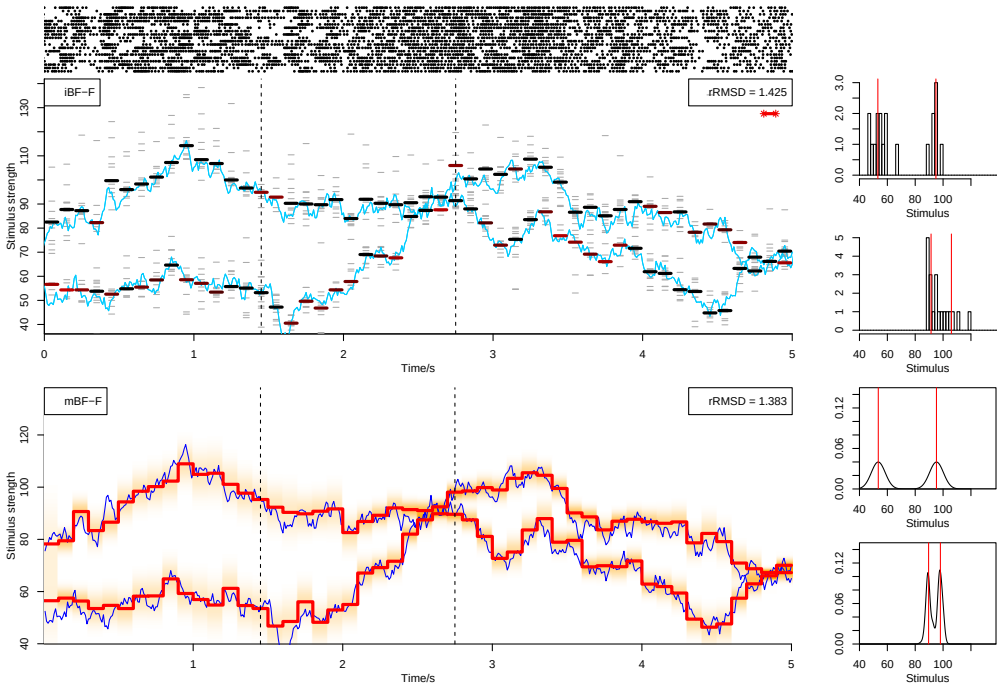


Figure 5.11: Decoding from 20 spike trains using BF assuming parallel processing. In the top panel 20 spike trains are shown. In the middle panel is shown the method using individual decoding and clustering. Short gray bars show the individual decoding results of stimulus at each time point from 20 spike trains. Thick bars show the medians of clustered categories. A more red color of the thick bars means less number of estimates inside the corresponding category. We mark by two stars if less than or equal to 5% (in this case, $5\% \times 20 = 1$). The histograms to the right show the distribution of 20 estimates with red lines indicating the medians. Blue curves show the true stimuli. In the lower panel is shown BF with marginal likelihood. For graphical reasons, we plot the two dimensional posterior estimation of the two stimuli in one dimension. For both decoding methods assuming parallel processing, all stimulus components are decoded at each time point.

In Figure 5.12 the rRMSD from 50 repetitions of different methods are shown as boxplots. Population decoding using multiple spike trains generally performs better than single spike train decoding. For serial processing, APF performs worse than BF, and for parallel processing APF performs as well as or better than BF, judging from rRMSD results. For both serial and parallel population processing methods, smoothing yields little or no improvement over filtering. However, the exception is the individual decoding methods for parallel processing, of course, since they are based on decoding of single stimuli. Indeed, significant improvement is observed when using smoothing instead of filtering for iBF and iAPF. The reason for the performances of BF, APF, filtering and smoothing can be partly found from the ESS values shown in Figure 5.13. Most notably, the ESS values are much smaller than the ESS values of single spike train results (Figure 5.8), due to extreme weights for larger sample sizes. This can lead to inaccurate approximations of the marginalization in fixed-lag smoothing and the integrals in the forward-backward algorithm. The smoothing performance is more affected by the small ESS than filtering. Furthermore, for serial processing BF has better ESS with higher median and smaller variance than APF, whereas for parallel processing, APF has better ESS. This explains the different performances of BF and APF in serial and parallel processing in Figure 5.12. Finally, regarding using geometric means, we do not observe much improvement of APFg and mAFPg over APF and mAPF. Using geometric means have positive effects since the ESS's are larger and the parameter degeneracy slows down (Figure 5.14) with APFg and mAFPg. However, the geometric mean changes the resulting posterior distribution and introduces a bias.

In Figure 5.14, parameter learning of γ is plotted for different methods. The APF algorithm for serial population decoding suffers from parameter degeneracy. Parameter degeneracy of APF with kernel smoothing (Liu and West, 2001) under large sample sizes has been reported in previous studies (Rios and Lopes, 2013), which is a phenomenon where the parameter distribution quickly becomes narrow or

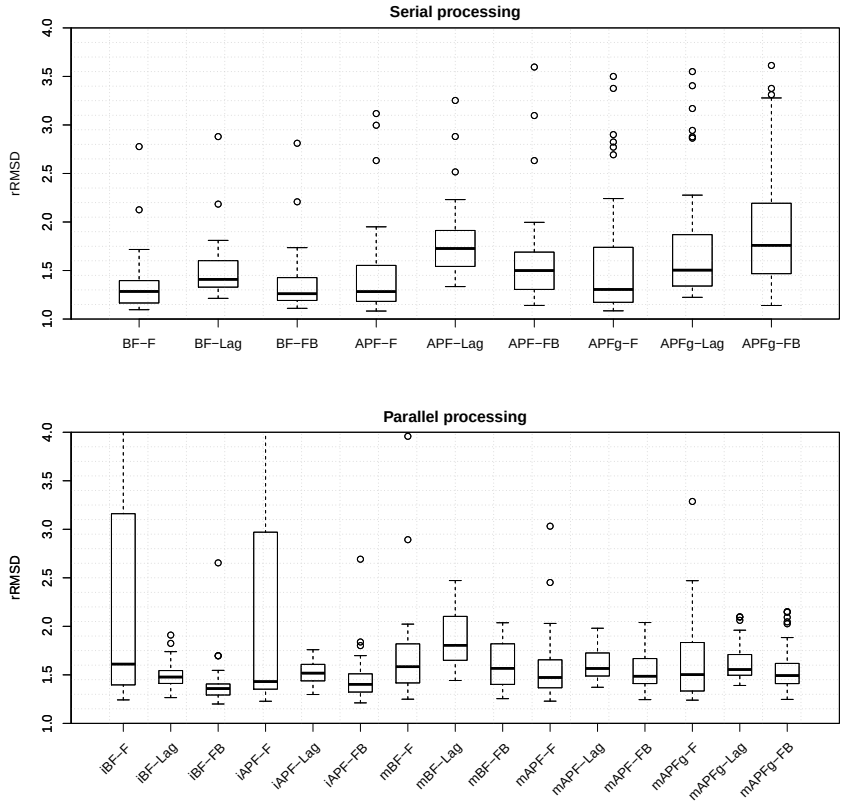
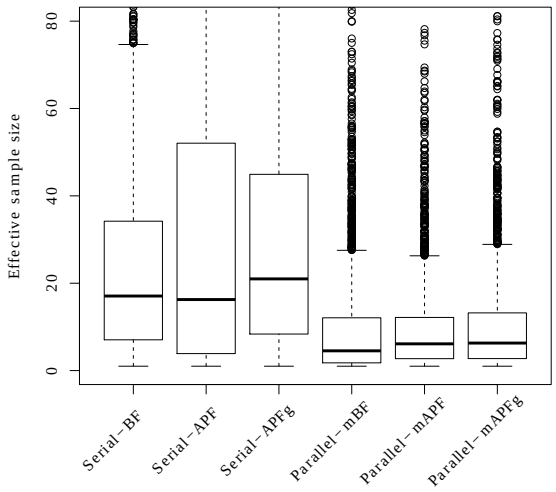


Figure 5.12: The rRMSD values using different particle methods for serial and parallel processing, calculated from 50 repetitions. In the labels of the x-axis, APFg: APF with geometric mean, iBF: individual decoding using BF, iAPF: individual decoding using APF, mBF: BF with marginal likelihood, mAPF: APF with marginal likelihood, mAFPg: APF with marginal likelihood and geometric mean. For example, APFg-FB means using APF with geometric mean, and reporting estimates using fixed-interval smoothing by the forward-backward algorithm.

Figure 5.13: ESS using different methods in serial and parallel processing, shown in box-plots for 2500 samples of 50 repetitions at 50 time steps. The labels in the x-axis show the methods used. For example, parallel-mAPFg means using mAPF with geometric mean for parallel processing.



collapses to a Dirac delta function. If parameter learning degenerates too fast before it receives sufficient data to achieve a good estimate, the parameter can be fixed at values far from the true one, reducing the decoding accuracy. Using the geometric mean slows down the degeneracy for serial processing. Other parameter learning methods have previously been studied using sufficient statistics, which may avoid the

degeneracy problem (Rios and Lopes, 2013; Carvalho et al., 2010); it is not pursued here. For particle filtering with marginal likelihood on parallel population decoding, there is not a large difference between APF and BF in terms of degeneracy.

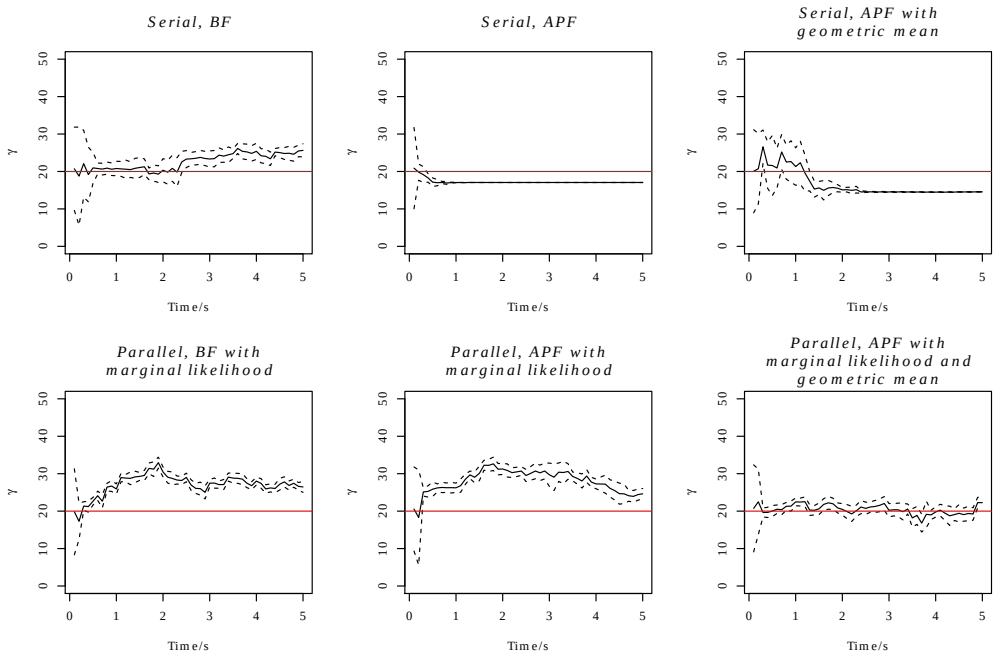


Figure 5.14: Examples of parameter learning of γ over time. The solid line is the mean of 500 particles, and dashed lines show $\pm$ the standard deviation. The red lines are the true value.

5.3 Approximating continuous-time switching

Here we simulate the attentional switching in continuous time following a Poisson process. With the same setup and methods as above, we conduct the population decoding with parallel processing. In Figure 5.15 is shown the decoding result of parallel population decoding, and in Figure 5.16 are shown two examples of single spike train decoding selected from the 20 spike trains in Figure 5.15. The posterior distribution to the right are taken from the switching time indicated by dashed lines. With a low Poisson switching rate, the decoding accuracy is not severely affected for parallel population decoding. For single spike train decoding, the estimate at switching times tend to be somewhere between the two values before and after the switch (first spike train in the upper panel in Figure 5.16), but sometimes the estimation can be far from the true stimulus (second spike train at 0.8s in the lower panel in Figure 5.16).

5.4 Decoding with the delay and decay kernel

In the above analysis, we have been using the burst response kernel which generates rhythmic and oscillatory bursting spiking patterns. Now we also try parallel population decoding using the decay and the delay kernel, shown in Figures 5.17 and 5.18, respectively. Again we use the same setup and methods. For the delay kernel, good performance is achieved, comparable with the burst kernel. For our current specification of the decay kernel, the spiking rate decreases greatly over time and we have to use stronger stimulus, but there are still long ISIs (e.g. in the middle region from 2s to 4s) which reduce the decoding accuracy.

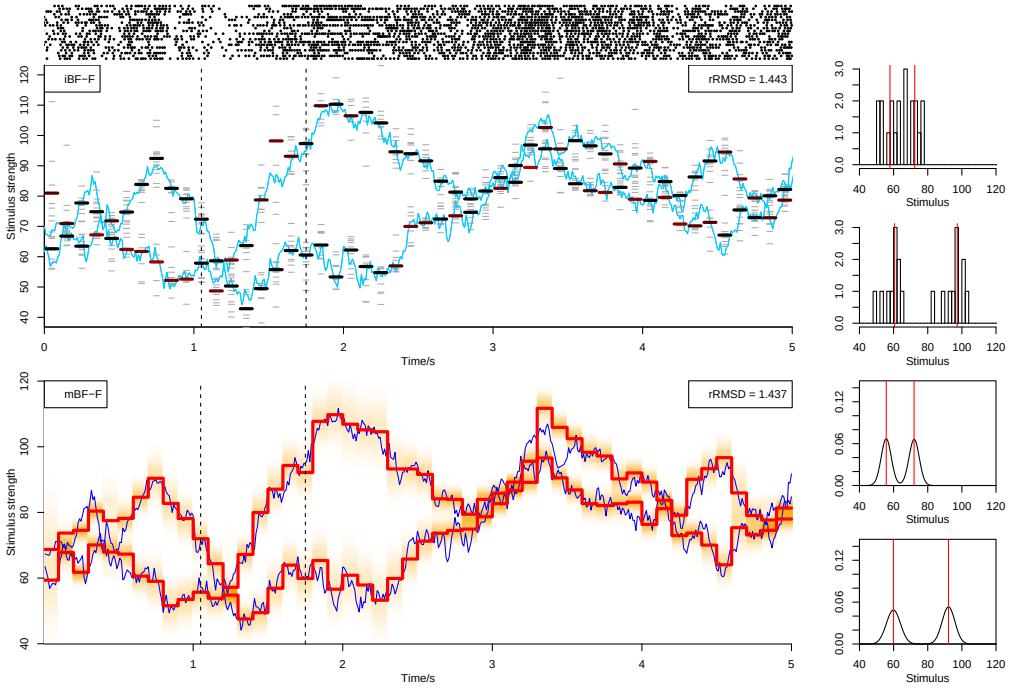


Figure 5.15: Decoding from 20 spike trains using BF assuming parallel processing. In each spike train, neuronal attention switches at continuous times following a Poisson process.

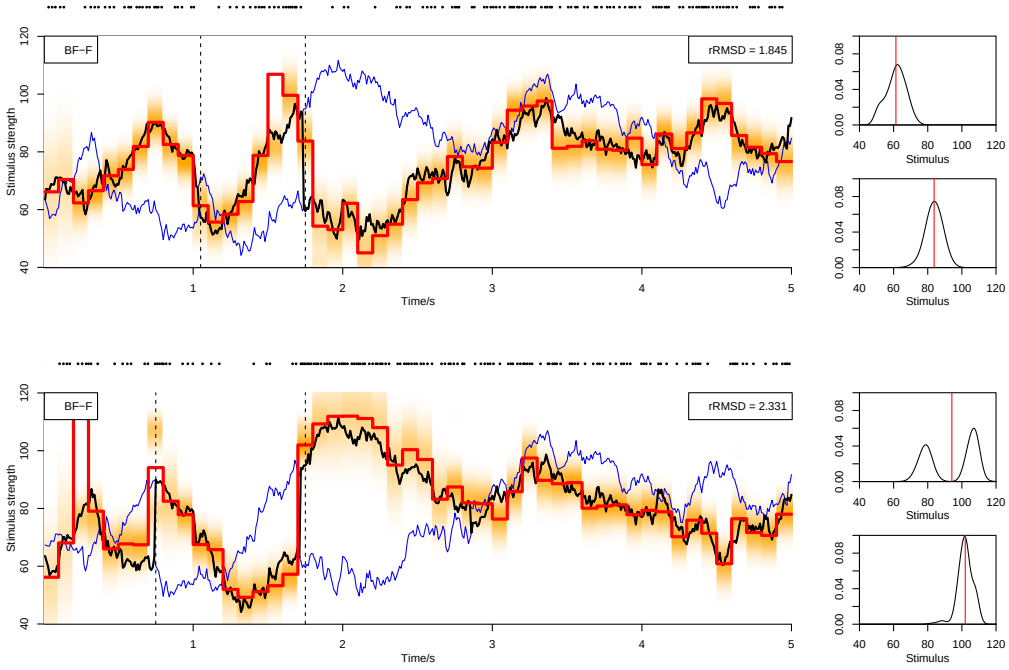


Figure 5.16: Decoding of two example single spike trains selected from Figure 5.15 using BF. Neuronal attention switches at continuous times following a Poisson process. Example switching times are indicated by dashed lines.

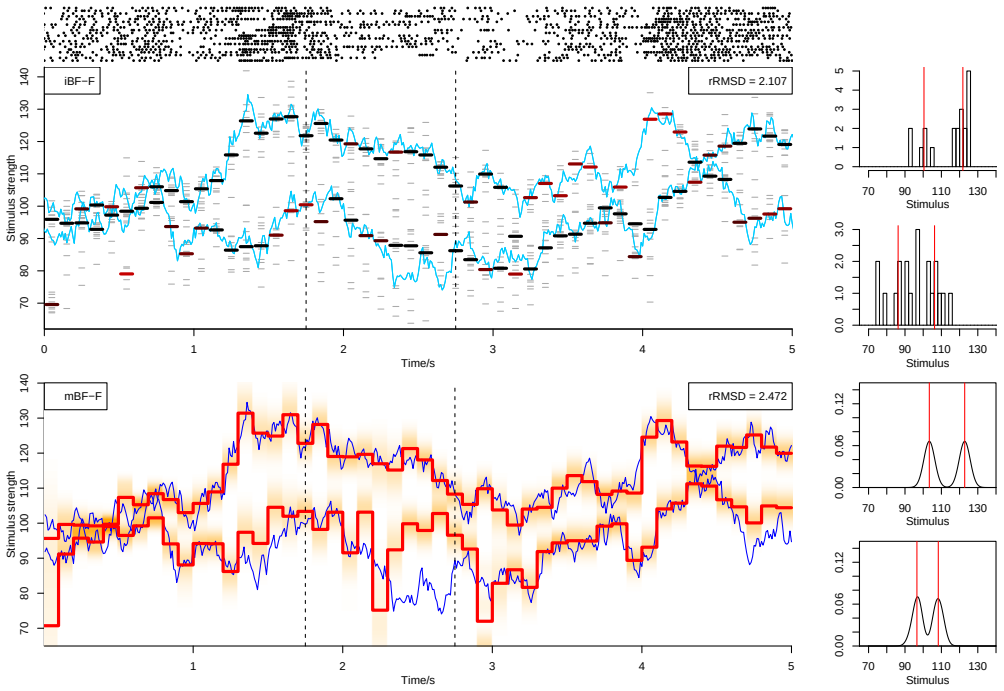


Figure 5.17: Decoding from 20 spike trains using BF assuming parallel processing. The decay response kernel is used in the LIF model.

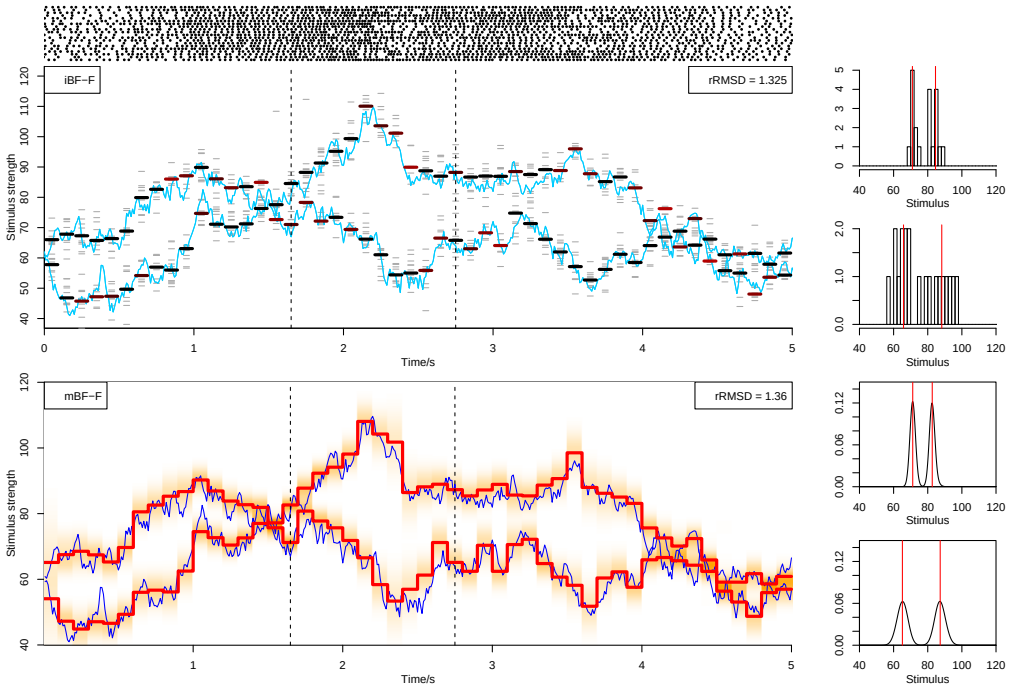


Figure 5.18: Decoding from 20 spike trains using BF assuming parallel processing. The delay response kernel is used in the LIF model.

6 Discussion

We have shown how to decode mixtures of multiple stimuli in the framework of visual attention under the hypothesis of probability mixing, which assumes the neuron responds to only one single stimulus at any

time. The opposing hypothesis is response averaging (Reynolds and Heeger, 2009), which assumes the neuron responds to a weighted average of the mixture. In this case, the decoding of each single stimulus would be much harder or impossible due to the difficulty in identifying each single stimulus based on the estimate of the weighted average, and information of individual stimulus characteristics would not be identifiable. This is an argument for why the neural system probably follows the probability-mixing hypothesis, as we also show in Li et al. (2016b).

The case of deterministic mixtures expands the capability of decoding to mixtures of an unknown number of components. The new proposed cluster decoding algorithm that first clusters the spike trains into k categories and then decodes without probability mixtures, is more efficient and stable than direct MAP decoding with weights as nuisance parameters. The decoding with an unknown number of components can potentially be applied to decoding of complicated visual scenes, to find out how many components the neurons would treat the visual scene as.

When decoding stochastic mixtures, we successfully decode the attended stimulus component using a single spike train or using population data under serial processing. Using population data under parallel processing enables us to obtain information of all stimulus components. Various types of particle methods are employed and compared. Interestingly, we find that the more complicated techniques using APF and kernel-based parameter learning do not necessarily perform better than basic methods using BF, and smoothing, conditional on more observations, does not necessarily perform better than filtering. This is related to sample size and model complexity.

For a limited number of particles (500 in our case), smoothing performance is closely related to ESS and how extreme the weights are. If the sample size is increased, weights become extreme and ESS decreases. After $n^* = 10$ times of resampling, the values $\{S^i_{n-n^*}; i = 1, \ldots, I\}$ used in fixed-lag smoothing only contain very few or only one unique value, so the accuracy will be affected. The forward-backward algorithm is also affected because the backward sweep requires the integration using the past particles. Therefore, for a large sample size smoothing can perform worse than filtering.

The performance of APF compared with BF has previously been studied; see e.g. Johansen and Doucet (2008); Douc et al. (2009); Whiteley and Johansen (2010). APF applies new proposal weights to resample particles by an early introduction of subsequent distributions, as a variance reduction approach: the estimation variance is reduced if we achieve a good prediction of subsequent weights and thus larger ESS. When the sample size is large, distributions become narrow and the first-stage weight in APF cannot provide good prediction of the subsequent distribution; meanwhile, the more complicated two-stage numerical calculations under a limited particle size could yield more variance and bias. Therefore, the variance reduction can perform worse for a large sample size. When the model is more complicated, so are the prior and transition distributions of the states. It becomes difficult for BF to have good samples with a limited number of particles. APF, on the other hand, gains advantage by introducing the subsequent states information, and therefore suffers less from the increased model complexity than BF. Increased model complexity also makes the distributions less narrow under a large sample size due to higher dimensions. In summary, APF is more favored for smaller sample sizes and more complex models. In our case, population decoding contains a bigger sample size than single spike train decoding. Increasing the stimulus number K yields higher dimensions and thus a more complex model. With the same K, parallel processing with mAPF (using full stimulus information) has larger dimensions than serial processing with APF (using partial stimulus information).

In our simulations of parallel processing, the stimulus number K is much less than the number of simultaneously recorded spike trains, and each stimulus component has sufficiently large probability to be attended. Consequently, at all time points each component is likely to be attended by some neurons and we decode all stimulus components. If, on the other hand, K is too large, or the probability of attending to one of more components is very small, the decoded stimuli will not likely form as many as K clusters. In that case we could try out different K values for the clustering analysis, and report the k^* which minimizes the BIC. This means that among all K stimuli, k^* are most likely attended by the recorded neurons and we decode those k^* attended stimuli.

Appendix

A Probability of ISIs

Suppose the membrane potential x resets to x_0 at time 0, and the spike time $t > 0$. We use the following notation:

$$\begin{aligned} f(x,t|S,\mathcal{H}_{t-}) &\text{ (time-evolving probability density of voltage)} \\ F(x,t|S,\mathcal{H}_{t-}) &\text{ (time-evolving cumulative distribution of voltage)} \\ g(t|S,\mathcal{H}_{t-}) &\text{ (probability density of spiking at } t\text{, i.e., PDF of the ISI)} \\ G(t|S,\mathcal{H}_{t-}) &\text{ (cumulative distribution of spiking at } t\text{, i.e., CDF of the ISI)} \end{aligned}$$

All the above probabilities depend on the stimulus S and the spike history up to the previous spike, $\mathcal{H}_{t-}$. In the following, we suppress S and $\mathcal{H}_{t-}$ in the notation for readability.

The probability that the neuron has not yet fired at time t, $1 - G(t)$, is equivalent to the probability that the potential has not yet reached x_{th}, $F(X_{th}, t)$. Thus, the probability density of an ISI is

$$g(t) = -\frac{\partial}{\partial t}F(x_{th},t) = -\frac{\partial}{\partial t}\int_{-\infty}^{x_{th}} f(x',t)dx'. \tag{A.1}$$

The transition probability density with a resetting threshold follows the Fokker-Planck equation, defined by the following partial differential equation (PDE):

$$\partial_t f(x,t) = -\partial_x(b(t)f(x,t)) + \frac{\sigma^2}{2}\partial_{xx}^2 f(x,t), \tag{A.2}$$

with absorbing boundary condition $f(x_{th}, t) = 0$ and initial condition $f(x, 0) = \delta(x - x_0)$. For numerical reasons, we also approximate by setting a reflecting boundary condition at a small value $x = x^-$, where the flux equals 0.

Now we formulate a PDE based on the CDF, $F(x, t)$ (Li et al., 2016a; Iolov et al., 2014; Hurn et al., 2005). Plugging $f(x, t) = \partial_x F(x, t)$ into (A.2) gives

$$\partial_t\partial_x F(x,t) = -\partial_x\left[b(x,t)\partial_x F(x,t) - \frac{\sigma^2}{2}\partial_x\partial_x F(x,t)\right]. \tag{A.3}$$

Integrating both sides with respect to x yields

$$\partial_t F(x,t) = -b(x,t)\partial_x F(x,t) + \frac{\sigma^2}{2}\partial_{xx}^2 F(x,t) + C(t). \tag{A.4}$$

At the lower reflecting boundary $x = x^-$, we have $F(x^-, t) = 0$ and thus $\partial_t F(x, t)|_{x=x^-} = 0$. The flux equals 0, so

$$\begin{aligned} J(x^-,t) &= -b(x^-,t)f(x^-,t) + \frac{\sigma^2}{2}\partial_x f(x,t)|_{x=x^-} \\ &= -b(x,t)\partial_x F(x,t)|_{x=x^-} + \frac{\sigma^2}{2}\partial_{xx}^2 F(x,t)|_{x=x^-} \\ &= 0. \end{aligned} \tag{A.5}$$

Thus, $C(t) = 0$, and we obtain the PDE for $F(x, t)$:

$$\partial_t F(x,t) = -b(x,t)\partial_x F(x,t) + \frac{\sigma^2}{2}\partial_{xx}^2 F(x,t), \tag{A.6}$$

with boundary conditions $\partial_x F(x_{th}, t) = 0$, $F(x^-, t) = 0$, and initial condition $F(x, 0) = H(x - x_0)$, where $H(\cdot)$ is the Heaviside step function.

The PDE is solved numerically using Crank-Nicholson finite difference method by discretizing time and potential with grid size Δt and Δx.

B Forward-Filtering Backward-Smoothing

Suppose a general hidden Markov process $X_{1:T}$ and observations $Y_{1:T}$. The smoothing distribution at time t can be expressed using

$$\begin{aligned}
&p(X_t|Y_{1:T}) \\
=&p(X_t|Y_{1:t,Y_{t+1:T}}) \\
=&\frac{p(Y_{t+1:T}|X_t,Y_{1:t})p(X_t|Y_{1:t})}{p(Y_{t+1:T}|Y_{1:t})} \\
=&p(X_t|Y_{1:t})\int\frac{p(X_{t+1}|X_t)p(Y_{t+1:T}|X_{t+1},Y_{1:t})}{p(Y_{t+1:T}|Y_{1:t})}dX_{t+1} \\
=&p(X_t|Y_{1:t})\int p(X_{t+1}|X_t)\frac{p(X_{t+1}|Y_{1:T})}{p(X_{t+1}|Y_{1:t})}dX_{t+1} \\
=&p(X_t|Y_{1:t})\int p(X_{t+1}|X_t)\frac{p(X_{t+1}|Y_{1:T})}{\int p(X_{t+1}|X_t)p(X_t|Y_{1:t})dX_t}dX_{t+1}.
\end{aligned} \tag{B.1}$$

Approximating the integrals using I particles, the smoothing weight of particle i is

$$\bar{w^*}_{t,i} \approx \bar{w}_{t,i}\sum_j \frac{p(X_{t+1,j}|X_{t,i})\bar{w^*}_{t+1,j}}{\sum_l p(X_{t+1,j}|X_{t,l})\bar{w}_{t,l}}, \tag{B.2}$$

where $\bar{w}_{t,i}$ is the normalized filtering weight at time t for particle i, which is calculated using the bootstrap filter and auxiliary particle filter algorithms introduced in the main text.

The transition distribution denoted by $p(X_{t+1,j}|X_{t,i})$ varies depending on what we use for the states X_t. In our case, if we include the attention state C, i.e. $X = (\mathbf{\Gamma}, C, \gamma, \beta, S)$, then for β and S we only need β^C and S^C. Otherwise, if $X = (\mathbf{\Gamma}, \gamma, \beta, S)$ when applying the marginal likelihood, then all β and S should be used. The calculation of the transition density $p(X_{t+1,j}|X_{t,i})$ follows the propagation given in 4.3.

Following equation (B.2), to obtain the smoothing weights we do normal filtering to have the filtering weights at all time points, then run backward smoothing using (B.2) recursively to calculate the smoothing weights at each time point.

References

Brockmeier, A. J., Choi, J. S., Kriminger, E. G., Francis, J. T., and Principe, J. C. (2014). Neural decoding with kernel-based metric learning. *Neural computation*, 26(6):1080–1107.

Brockwell, A. E., Rojas, A. L., and Kass, R. (2004). Recursive bayesian decoding of motor cortical signals by particle filtering. *Journal of Neurophysiology*, 91(4):1899–1907.

Bundesen, C. and Habekost, T. (2008). Principles of visual attention: Linking mind and brain.

Bundesen, C., Habekost, T., and Kyllingsbæk, S. (2005). A neural theory of visual attention: bridging cognition and neurophysiology. *Psychological Review*, 112(2):291.

Burkitt, A. N. (2006). A review of the integrate-and-fire neuron model: I. homogeneous synaptic input. *Biological cybernetics*, 95(1):1–19.

Carvalho, C., Johannes, M. S., Lopes, H. F., and Polson, N. (2010). Particle learning and smoothing. *Statistical Science*, 25(1):88–106.

Dayan, P. and Abbott, L. F. (2001). *Theoretical neuroscience : computational and mathematical modeling of neural systems*. Computational neuroscience. MIT Press, Cambridge (Mass.), London. Autre tirage: 2005 pour l'édition brochée.

Douc, R., Moulines, E., and Olsson, J. (2009). Optimality of the auxiliary particle filter. *Probability and Mathematical Statistics*, 29(1):1–28.

Doucet, A., Godsill, S., and Andrieu, C. (2000). On sequential monte carlo sampling methods for bayesian filtering. *Statistics and computing*, 10(3):197–208.

Eichhorn, J., Tolias, A., Zien, A., Kuss, M., Weston, J., Logothetis, N., Schölkopf, B., and Rasmussen, C. E. (2003). Prediction on spike data using kernel algorithms. In *Advances in neural information processing systems*, page None.

Fearnhead, P. (2004). Particle filters for mixture models with an unknown number of components. *Statistics and Computing*, 14(1):11–21.

Fiebelkorn, I. C., Saalmann, Y. B., and Kastner, S. (2013). Rhythmic sampling within and between objects despite sustained attention at a cued location. *Current Biology*, 23(24):2553–2558.

Fific, M., Nosofsky, R. M., and Townsend, J. T. (2008). Information-processing architectures in multidimensional classification: A validation test of the systems factorial technology. *Journal of Experimental Psychology: Human Perception and Performance*, 34(2):356.

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

Gerstner, W., Van Hemmen, J. L., and Cowan, J. D. (1996). What matters in neuronal locking? *Neural computation*, 8(8):1653–1676.

Hastie, T., Tibshirani, R., Friedman, J., Hastie, T., Friedman, J., and Tibshirani, R. (2009). *The elements of statistical learning*, volume 2. Springer, New York.

Hurn, A., Jeisman, J., and Lindsay, K. (2005). Ml estimation of the parameters of sdes by numerical solution of the fokker-planck equation. In *MODSIM 2005: International Congress on Modelling and Simulation: Advances and Applications for Management and Decision Making*, pages 849–855.

Iolov, A., Ditlevsen, S., and Longtin, A. (2014). Fokker–planck and Fortet equation-based parameter estimation for a leaky integrate-and-fire model with sinusoidal and stochastic forcing. *The Journal of Mathematical Neuroscience*, 4(1):4.

Johansen, A. M. and Doucet, A. (2008). A note on auxiliary particle filters. *Statistics & Probability Letters*, 78(12):1498–1504.

Kantas, N., Doucet, A., Singh, S. S., Maciejowski, J., Chopin, N., et al. (2015). On particle methods for parameter estimation in state-space models. *Statistical science*, 30(3):328–351.

Kass, R. E., Eden, U. T., and Brown, E. N. (2014). *Analysis of neural data*. Springer.

Kaufman, L. and Rousseeuw, P. J. (2009). *Finding groups in data: an introduction to cluster analysis*, volume 344. John Wiley & Sons.

Kelly, R. and Lee, T. S. (2003). Decoding V1 neuronal activity using particle filtering with Volterra kernels. In *Advances in neural information processing systems*, page None.

Kistler, W., Gerstner, W., and Hemmen, J. (1997). Reduction of the Hodgkin-Huxley equations to a single-variable threshold model. *Neural Computation*, 9(5):1015–1045.

Koyama, S., Eden, U. T., Brown, E. N., and Kass, R. E. (2010). Bayesian decoding of neural spike trains. *Annals of the Institute of Statistical Mathematics*, 62(1):37–59.

Lebedev, M. A. and Nicolelis, M. A. (2006). Brain–machine interfaces: past, present and future. *TRENDS in Neurosciences*, 29(9):536–546.

Li, K., Bundesen, C., and Ditlevsen, S. (2016a). Responses of leaky integrate-and-fire neurons to a plurality of stimuli in their receptive fields. *The Journal of Mathematical Neuroscience*, 6(1):1.

Li, K., Kozyrev, V., Kyllingsbæk, S., Treue, S., Ditlevsen, S., and Bundesen, C. (2016b). Neurons in primate visual cortex alternate between responses to multiple stimuli in their receptive field. *Submitted*.

Liu, J. and West, M. (2001). Combined parameter and state estimation in simulation-based filtering. In *Sequential Monte Carlo methods in practice*, pages 197–223. Springer.

Nobre, K. and Kastner, S. (2013). *The Oxford handbook of attention*. Oxford University Press.

Paninski, L., Ahmadian, Y., Ferreira, D. G., Koyama, S., Rad, K. R., Vidne, M., Vogelstein, J., and Wu, W. (2010). A new look at state-space models for neural data. *Journal of computational neuroscience*, 29(1-2):107–126.

Paninski, L., Pillow, J., and Lewi, J. (2007). Statistical models for neural encoding, decoding, and optimal stimulus design. *Progress in brain research*, 165:493–507.

Pillow, J. W., Ahmadian, Y., and Paninski, L. (2011). Model-based decoding, information estimation, and change-point detection techniques for multineuron spike trains. *Neural Computation*, 23(1):1–45.

Pitt, M. K. and Shephard, N. (1999). Filtering via simulation: Auxiliary particle filters. *Journal of the American statistical association*, 94(446):590–599.

Reynolds, J. H. and Heeger, D. J. (2009). The normalization model of attention. *Neuron*, 61(2):168–185.

Rieke, F. (1999). *Spikes: exploring the neural code*. MIT press.

Rios, M. P. and Lopes, H. F. (2013). The extended liu and west filter: Parameter learning in markov switching stochastic volatility models. In *State-Space Models*, pages 23–61. Springer.

Sacerdote, L. and Giraudo, M. T. (2013). *Stochastic Biomathematical Models with Applications to Neuronal Modeling*, volume 2058, chapter Stochastic Integrate and Fire Models: A Review on Mathematical Methods and Their Applications, pages 99–148. Lecture Notes in Mathematics series (Biosciences subseries), Springer, New York.

Shoham, S., Paninski, L. M., Fellows, M. R., Hatsopoulos, N. G., Donoghue, J. P., and Normann, R. A. (2005). Statistical encoding model for a primary motor cortical brain-machine interface. *Biomedical Engineering, IEEE Transactions on*, 52(7):1312–1322.

Stephens, M. (2000). Dealing with label switching in mixture models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62(4):795–809.

Townsend, J. T. (1990). Serial vs. parallel processing: Sometimes they look like tweedledum and tweedledee but they can (and should) be distinguished. *Psychological Science*, 1(1):46–54.

Truccolo, W., Eden, U. T., Fellows, M. R., Donoghue, J. P., and Brown, E. N. (2005). 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.

Waldert, S., Pistohl, T., Braun, C., Ball, T., Aertsen, A., and Mehring, C. (2009). A review on directional information in neural signals for brain-machine interfaces. *Journal of Physiology-Paris*, 103(3):244–254.

Warland, D. K., Reinagel, P., and Meister, M. (1997). Decoding visual information from a population of retinal ganglion cells. *Journal of Neurophysiology*, 78(5):2336–2350.

Whiteley, N. and Johansen, A. M. (2010). Recent developments in auxiliary particle filtering. *Barber, Cemgil, and Chiappa, editors, Inference and Learning in Dynamic Models. Cambridge University Press*, 38:39–47.

Wu, W., Gao, Y., Bienenstock, E., Donoghue, J. P., and Black, M. J. (2006). Bayesian population decoding of motor cortical activity using a kalman filter. *Neural computation*, 18(1):80–118.