Hidden Markov Models for the Stimulus-Response Relationships of Multistate Neural Systems

Sean Escola
sean@neurotheory.columbia.edu
Center for Theoretical Neuroscience and Department of Psychiatry, Columbia University, New York, NY 10032, U.S.A.

Alfredo Fontanini
alfredo.fontanini@stonybrook.edu
Department of Neurobiology and Behavior, Stony Brook University, Stony Brook, NY 11794, U.S.A.

Don Katz
dbkatz@brandeis.edu
Department of Psychology, Brandeis University, Waltham, MA 02453, U.S.A.

Liam Paninski
liam@stat.columbia.edu
Center for Theoretical Neuroscience and Department of Statistics, Columbia University, New York, NY 10032, U.S.A.

Given recent experimental results suggesting that neural circuits may evolve through multiple firing states, we develop a framework for estimating state-dependent neural response properties from spike train data. We modify the traditional hidden Markov model (HMM) framework to incorporate stimulus-driven, non-Poisson point-process observations. For maximal flexibility, we allow external, time-varying stimuli and the neurons' own spike histories to drive both the spiking behavior in each state and the transitioning behavior between states. We employ an appropriately modified expectation-maximization algorithm to estimate the model parameters. The expectation step is solved by the standard forward-backward algorithm for HMMs. The maximization step reduces to a set of separable concave optimization problems if the model is restricted slightly. We first test our algorithm on simulated data and are able to fully recover the parameters used to generate the data and accurately recapitulate the sequence of hidden states. We then apply our algorithm to a recently published data set in which the observed neuronal ensembles displayed multistate behavior and show that inclusion of spike history information significantly improves the fit of the model. Additionally, we show that a simple reformulation of the state

space of the underlying Markov chain allows us to implement a hybrid half-multistate, half-histogram model that may be more appropriate for capturing the complexity of certain data sets than either a simple HMM or a simple peristimulus time histogram model alone.

1 Introduction

Evidence from recent experiments indicates that many neural systems may exhibit multiple, distinct firing regimes, such as tonic and burst modes in thalamus (for review, see Sherman, 2001) and UP and DOWN states in cortex (Anderson, Lampl, Reichova, Carandini, & Ferster, 2000; Sanchez-Vives & McCormick, 2000; Haider, Duque, Hasenstaub, Yu, & McCormick, 2007). It is reasonable to speculate that neurons in multistate networks that are involved in sensory processing might display differential firing behaviors in response to the same stimulus in each of the states of the system; indeed, Bezdudnaya et al. (2006) showed that temporal receptive field properties change between tonic and burst states for relay cells in rabbit thalamus. These results call into question traditional models of stimulus-evoked neural responses that assume a fixed, reproducible mechanism by which a stimulus is translated into a spike train. For the case of a time-varying stimulus (e.g., a movie), the neural response has often been modeled by the generalized linear model (GLM; Simoncelli, Paninski, Pillow, & Schwartz, 2004; Paninski, 2004; Truccolo, Eden, Fellows, Donoghue, & Brown, 2005; Paninski, Pillow, & Lewi, 2007) where spikes are assumed to result from a point process whose instantaneous firing rate λ_t at time t is given by

$$\lambda_t = f\left(\mathbf{k}^{\mathrm{T}}\mathbf{s}_t\right), \tag{1.1}$$

where f is a positive, nonlinear function (e.g., the exponential), $\mathbf{s}_t$ is the stimulus input at time t (which can also include spike history and interneuronal effects), and $\mathbf{k}$ is the direction in stimulus space that causes maximal firing (i.e., the preferred stimulus or receptive field of the neuron). Since $\mathbf{k}$ does not change with time, this model assumes that the response function of the neuron is constant throughout the presentation of the stimulus (i.e., the standard GLM is a single-state model that would be unable to capture the experimental results discussed above).

In this article, we propose a generalization of the GLM appropriate for capturing the time-varying stimulus-response properties of neurons in multistate systems. We base our model on the hidden Markov model (HMM) framework (Rabiner, 1989). Specifically, we model the behavior of each cell in each state n as a GLM with a state-dependent stimulus filter $\mathbf{k}_n$, where transitions from state to state are governed by a Markov chain whose transition probabilities may also be stimulus dependent. Our model is an extension of previous HMMs applied to neural data (Abeles et al., 1995;

Seidemann, Meilijson, Abeles, Bergman, & Vaadia, 1996; Jones, Fontanini, Sadacca, Miller, & Katz, 2007; Chen, Vijayan, Barbieri, Wilson, & Brown, 2009; Tokdar, Xi, Kelly, & Kass, 2009), and is thus an alternative to several of the recently developed linear state-space models (Brown, Nguyen, Frank, Wilson, & Solo, 2001; Smith & Brown, 2003; Eden, Frank, Barbieri, Solo, & Brown, 2004; Kulkarni & Paninski, 2007), which also attempt to capture more of the complexity in the stimulus-response relationship than is possible with a simple GLM.

To infer the most likely parameters of our HMM given an observed spike train, we adapt the standard Baum-Welch expectation-maximization (EM) algorithm (Baum, Petrie, Soules, & Weiss, 1970; Dempster, Laird, & Rubin, 1977) to point-process data with stimulus-dependent transition and observation densities. The E-step here proceeds via a standard forward-backward recursion, while the M-step turns out to consist of a separable set of concave optimization problems if a few reasonable restrictions are placed on the model (Paninski, 2004). The development of EM algorithms for the analysis of point-process data with continuous state-space models has been previously described (Chan & Ledolter, 1995; Smith & Brown, 2003; Kulkarni & Paninski, 2007; Czanner et al., 2008), as has the development of EM algorithms for the analysis of point-process data with discrete state-space models, albeit using Markov chain Monte Carlo techniques to estimate the E-step of the algorithm (Sansom & Thomson, 2001; Chen et al., 2009; Tokdar et al., 2009). Our algorithm, on the other hand, uses a discrete state-space model with inhomogeneous transition and observation densities and allows the posterior probabilities in the E-step to be computed exactly.

This article is organized as follows: Section 2 briefly reviews the basic HMM framework and associated parameter learning algorithm, and then develops our extension of these methods for stimulus-driven multistate neurons. We also introduce an extension that may be appropriate for data sets with spike trains that are triggered by an event (e.g., the beginning of a behavioral trial) but are not driven by a known time-varying stimulus. This extension results in a hybrid half-multistate, half-histogram model. Section 3 presents the results of applying our model and learning procedure to two simulated data sets meant to represent a thalamic relay cell with different tonic and burst firing modes and a cell in sensory cortex that switches between stimulus-attentive and stimulus-ignoring states. In section 4, we analyze a data set from rat gustatory cortex in which multistate effects have previously been noted (Jones et al., 2007), expanding the prior analysis to permit spike-history-dependent effects. Our results show that accounting for history dependence significantly improves the cross-validated performance of the HMM. In section 5 we conclude with a brief discussion of the models and results presented in this article in comparison to other approaches for capturing multistate neuronal behavior.

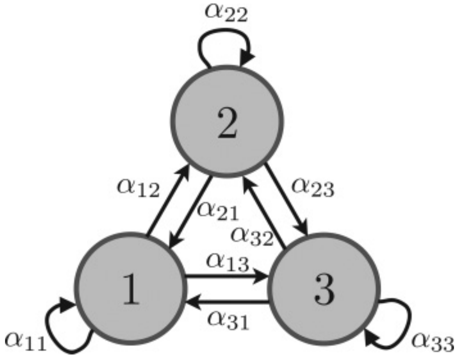

Figure 1: An example Markov chain with three states. At every time step, the system transitions from its current state n to some new state m (which could be the same state) by traveling along the edges of the graph according to the probabilities α_{nm} associated with each edge.

2 Methods

2.1 Hidden Markov Model Review. Before we present our modification of the HMM framework for modeling the stimulus-response relationship of neurons in multistate systems, we briefly review the traditional framework as described in Rabiner (1989). While sections 2.1.1 through 2.1.3 are not specific to neural data, we will note features of the model that we modify in later sections and introduce notation that we use throughout the article.

2.1.1 Model Introduction and Background. HMMs are described by two random variables at every point in time t: the state q_t and the emission y_t. Assuming that the state variable q_t can take on one of N discrete states $\{1, \ldots, N\}$ and makes a transition at every time step according to fixed transition probabilities (as shown in Figure 1 for $N = 3$), then the states form a homogeneous, discrete-time Markov chain defined by the following two properties. First,

$$p\big(q_t \mid \mathbf{q}_{[0:t-1]}, \mathbf{y}_{[0:t-1]}\big) = p(q_t \mid q_{t-1}), \tag{2.1}$$

or the future state is independent of past states and emissions given the present state (i.e., the Markov assumption). Thus, the sequence of states, $\mathbf{q} \equiv (q_0, \ldots, q_T)^{\mathrm{T}}$, evolves only with reference to itself, without reference to the sequence of emissions, $\mathbf{y} \equiv (y_0, \ldots, y_T)^{\mathrm{T}}$. Second,

$$\alpha_{nm} \equiv p(q_t = m \mid q_{t-1} = n) = p(q_s = m \mid q_{s-1} = n), \quad \forall t, s \in \{1, \ldots, T\}, \tag{2.2}$$

or the probability of transitioning from state n to state m is constant (homogeneous) for all time points. All homogeneous, discrete-time Markov chains can then be completely described by matrices $\boldsymbol{\alpha}$ with the constraints that $0 \leq \alpha_{nm} \leq 1$ and $\sum_{m=1}^{N} \alpha_{nm} = 1$. We will relax both the independence of state transition probabilities on past emissions (see equation 2.1) and the homogeneity assumption (see equation 2.2) in our adaptation of the model to allow for spike history dependence and dynamic state transition probabilities, respectively.

In another Markov-like assumption, the probability distributions of the emission variables do not depend on any previous (or future) state or emission given the current state,

$$p\big(y_t \mid \mathbf{q}_{[0:t]}, \mathbf{y}_{[0:t-1]}\big) = p(y_t \mid q_t), \tag{2.3}$$

another assumption we will relax. The traditional HMM framework assumes that the emission probability distributions, similar to the transition probability distributions, are time homogeneous. Thus, the emission probability distributions can be represented with matrices $\boldsymbol{\eta}$ that have the same constraints as the transition matrices: $0 \leq \eta_{nk} \leq 1$ and $\sum_{k=1}^{K} \eta_{nk} = 1$, where $\eta_{nk} \equiv p(y_t = k \mid q_t = n)$ for a system with K discrete emission classes $\{1, \ldots, K\}$.

The dependencies and conditional independencies of an HMM as encapsulated in the Markov assumptions, equations 2.1 and 2.3, can be easily captured in the graphical model shown in Figure 2a. As can be seen directly from the figure, the following factorized, complete log-probability distribution over the sequence of states and the sequence of emissions is the full, probabilistic description of an HMM:

$$\log p(\mathbf{y}, \mathbf{q}) = \log\left(p(q_0) \prod_{t=1}^{T} p(q_t \mid q_{t-1}) \prod_{t=0}^{T} p(y_t \mid q_t)\right) \tag{2.4}$$

or

$$\log p(\mathbf{y}, \mathbf{q} \mid \boldsymbol{\alpha}, \boldsymbol{\eta}, \boldsymbol{\pi}) = \log \pi_{q_0} + \sum_{t=1}^{T} \log \alpha_{q_{t-1} q_t} + \sum_{t=0}^{T} \log \eta_{q_t y_t}, \tag{2.5}$$

where the $N \times N$ matrix $\boldsymbol{\alpha}$ and the $N \times K$ matrix $\boldsymbol{\eta}$ are as defined above, and the N-element vector $\boldsymbol{\pi}$ is the initial state distribution ($\pi_n \equiv p(q_0 = n)$).

The parameters of the model $\boldsymbol{\alpha}$, $\boldsymbol{\eta}$, and $\boldsymbol{\pi}$ (or, collectively, $\boldsymbol{\theta}$) are learned from the data by maximizing the log likelihood. Unlike the sequence of emissions $\mathbf{y}$, which is known (e.g., experimentally measured), the sequence of states $\mathbf{q}$ in an HMM is unknown (thus, "hidden") and must be integrated out of the complete log-likelihood equation to yield the marginal log

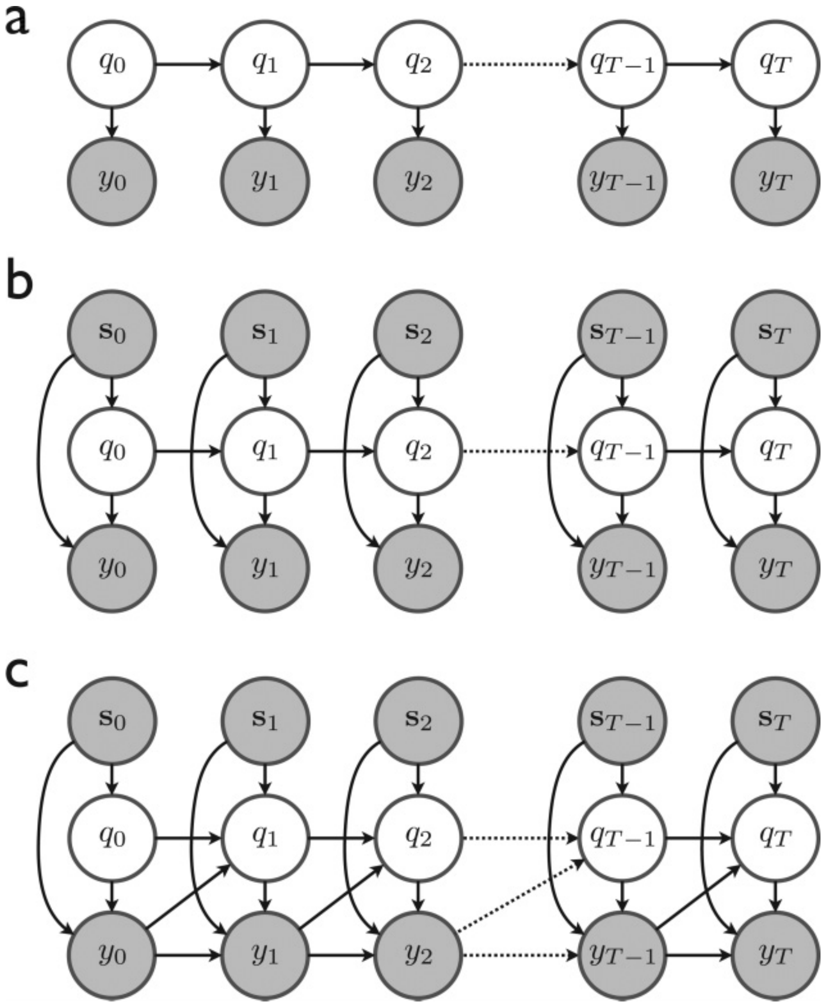

Figure 2: The graphical models corresponding to the HMMs discussed in the text. Each node is a random variable in the system, and the edges represent causal dependences. The hidden states $\{q_0, \ldots, q_T\}$ are the latent variables of the models and are represented with white nodes to denote this distinction. (a) The traditional HMM where the transition and emission probability distributions are homogeneous. (b) The stimulus-driven HMM where the inhomogeneous probability distributions are dependent on an external, time-varying stimulus. (c) The stimulus and history-driven HMM where the distributions are also dependent on the emission history (e.g., spike history) of the system.

likelihood:

$$\begin{aligned}L(\boldsymbol{\theta} \mid \mathbf{y}) &\equiv \log p(\mathbf{y} \mid \boldsymbol{\theta}) \\ &= \log \sum_{\mathbf{q}} p(\mathbf{y}, \mathbf{q} \mid \boldsymbol{\theta}) \\ &= \log \sum_{\mathbf{q}} \left(\pi_{q_0} \prod_{t=1}^{T} \alpha_{q_{t-1}q_t} \prod_{t=0}^{T} \eta_{q_t y_t} \right), \qquad (2.6)\end{aligned}$$

where the notation $L(\boldsymbol{\theta} \mid \cdot)$ expresses the log-likelihood as a function of the model parameters: $L(\boldsymbol{\theta} \mid \cdot) \equiv \log p(\cdot \mid \boldsymbol{\theta})$. The sum in equation 2.6 is over all possible paths along the hidden Markov chain during the course of the time series. The forward-backward algorithm allows a recursive evaluation of this likelihood, whose complexity is linear rather than exponential in T and is the topic of the next section.

2.1.2 The Forward-Backward Algorithm. In order to find the parameters that maximize the marginal log likelihood, we first need to be able to evaluate this likelihood efficiently. This is solved by the forward-backward algorithm (Baum et al., 1970), which also comprises the E-step of the Baum-Welch algorithm (EM for HMMs).

The forward-backward algorithm works in the following manner. First, the "forward" probabilities are defined as

$$a_{n,t} \equiv p\big(\mathbf{y}_{[0:t]}, q_t = n \mid \boldsymbol{\theta}\big), \qquad (2.7)$$

which is the probability of all of the emissions up to time t and the probability that at time t, the system is in state n. The forward probabilities can be calculated recursively by

$$a_{n,0} = \pi_n \eta_{n y_0} \qquad (2.8)$$

and

$$a_{n,t} = \left(\sum_{m=1}^{N} a_{m,t-1} \alpha_{mn} \right) \eta_{n y_t}, \qquad (2.9)$$

which involves $\mathcal{O}(T)$ computation. Marginalizing over the hidden state in the final forward probabilities yields the likelihood

$$p(\mathbf{y} \mid \boldsymbol{\theta}) = \sum_{n=1}^{N} a_{n,T}, \qquad (2.10)$$

the log of which is equivalent to equation 2.6.

To complete the algorithm, the "backward" probabilities are introduced as

$$b_{n,t} \equiv p\big(\mathbf{y}_{[t+1:T]} \mid q_t = n, \boldsymbol{\theta}\big), \tag{2.11}$$

which is the probability of all future emissions given that the state is n at time t. These can also be computed recursively by

$$b_{n,T} = 1 \tag{2.12}$$

and

$$b_{n,t} = \sum_{m=1}^{N} \alpha_{nm} \eta_{my_{t+1}} b_{m,t+1}, \tag{2.13}$$

which also involves linear time complexity in T.

It is now trivial to calculate the single and consecutive pairwise marginal probabilities of $p(\mathbf{q} \mid \mathbf{y}, \boldsymbol{\theta})$, the posterior distribution of the state sequence given the emission sequence, as

$$p(q_t = n \mid \mathbf{y}, \boldsymbol{\theta}) = \frac{a_{n,t} b_{n,t}}{p(\mathbf{y} \mid \boldsymbol{\theta})} \tag{2.14}$$

and

$$p(q_t = n, q_{t+1} = m \mid \mathbf{y}, \boldsymbol{\theta}) = \frac{a_{n,t} \alpha_{nm} \eta_{my_{t+1}} b_{m,t+1}}{p(\mathbf{y} \mid \boldsymbol{\theta})}. \tag{2.15}$$

Computing these marginals constitutes the E-step of EM, which is the subject of the next section.

2.1.3 HMM Expectation-Maximization. The EM algorithm (Dempster et al., 1977) is an iterative process for learning model parameters with incomplete data. During the E-step, the posterior distribution over the hidden variables given the data and the model parameters, $p(\mathbf{q} \mid \mathbf{y}, \boldsymbol{\theta}^i)$ is calculated, where $\boldsymbol{\theta}^i$ is the parameter setting during iteration i. During the M-step, the next setting of the parameters is found by maximizing the expected complete log likelihood with respect to the parameters, where the expectation is taken over the posterior distribution resulting from the E-step:

$$\boldsymbol{\theta}^{i+1} = \arg\max_{\boldsymbol{\theta}} \big\langle L(\boldsymbol{\theta} \mid \mathbf{y}, \mathbf{q}) \big\rangle_{p(\mathbf{q}|\mathbf{y},\boldsymbol{\theta}^i)}. \tag{2.16}$$

While EM is guaranteed to increase the likelihood with each iteration of the procedure,

$$L\left(\boldsymbol{\theta}^{i+1} \mid \mathbf{y}\right) \geq L\left(\boldsymbol{\theta}^{i} \mid \mathbf{y}\right), \tag{2.17}$$

it is susceptible to being trapped in local minima and may not converge as rapidly as other procedures (Salakhutdinov, Roweis, & Ghahramani, 2003).

By substituting the complete log likelihood for an HMM, equation 2.5, into the equation for the M-step, equation 2.16, it becomes clear why the forward-backward algorithm is the E-step for an HMM.

$$\begin{aligned}
\langle L(\boldsymbol{\theta} \mid \mathbf{y}, \mathbf{q})\rangle_{\hat{p}(\mathbf{q})} &= \left\langle \log \pi_{q_0} + \sum_{t=1}^{T} \log \alpha_{q_{t-1}q_t} + \sum_{t=0}^{T} \log \eta_{q_t y_t} \right\rangle_{\hat{p}(\mathbf{q})} \\
&= \langle \log \pi_{q_0} \rangle_{\hat{p}(\mathbf{q})} + \sum_{t=1}^{T} \langle \log \alpha_{q_{t-1}q_t} \rangle_{\hat{p}(\mathbf{q})} + \sum_{t=0}^{T} \langle \log \eta_{q_t y_t} \rangle_{\hat{p}(\mathbf{q})} \\
&= \langle \log \pi_{q_0} \rangle_{\hat{p}(q_0)} + \sum_{t=1}^{T} \langle \log \alpha_{q_{t-1}q_t} \rangle_{\hat{p}(q_{t-1},q_t)} \\
&\quad + \sum_{t=0}^{T} \langle \log \eta_{q_t y_t} \rangle_{\hat{p}(q_t)} \\
&= \sum_{n=1}^{N} \hat{p}(q_0 = n) \log \pi_n \\
&\quad + \sum_{t=1}^{T} \sum_{n=1}^{N} \sum_{m=1}^{N} \hat{p}(q_{t-1} = n, q_t = m) \log \alpha_{nm} \\
&\quad + \sum_{t=0}^{T} \sum_{n=1}^{N} \hat{p}(q_t = n) \log \eta_{n y_t},
\end{aligned} \tag{2.18}$$

where $\hat{p}(\mathbf{q})$ is used in place of $p(\mathbf{q} \mid \mathbf{y}, \boldsymbol{\theta}^i)$ to simplify notation. From equation 2.18, it is clear that although the complete posterior distribution over the sequence of states $p(\mathbf{q} \mid \mathbf{y}, \boldsymbol{\theta}^i)$ is not computed by the forward-backward algorithm, the only quantities needed during the M-step are the single and consecutive-pairwise marginal distributions given by equations 2.14 and 2.15.

In the simple case of static $\boldsymbol{\alpha}$ and $\boldsymbol{\eta}$ matrices in a time-homogeneous HMM, it is possible to derive analytic solutions for the next parameter setting in each M-step. In the more general case, other techniques such as gradient ascent can be employed to maximize equation 2.18, as will be described below. However, the analytic solution of the parameter update for the initial state distribution $\boldsymbol{\pi}$ is still useful in the general case. This can be easily shown to be

$$\pi_n = \hat{p}(q_0 = n). \tag{2.19}$$

2.2 HMMs Modified for Stimulus-Driven Neural Response Data. We develop an HMM to model spike train data produced by neurons that transition between several hidden neuronal states. In the most general case, we assume that an external stimulus is driving the neurons' firing patterns within each state, as well as the transitions between states. We further extend the model to allow spike history effects such as refractory periods and burst activity. Although, for notational simplicity, we initially develop the model assuming that the data consist of a single spike train recorded from a single neuron, in section 2.2.5 we show that this framework can be easily extended to the multicell and multitrial setting.

2.2.1 Incorporating Point-Process Observations. In order to be relevant to neural spike train recordings, the traditional HMM framework must be modified to handle point-process data. We begin by redefining the emission matrices to be parameterized by rates λ_n. Thus, each row of $\boldsymbol{\eta}$ becomes the Poisson distribution corresponding to each state,

$$\eta_{ni} = \frac{(\lambda_n dt)^i \, e^{-\lambda_n dt}}{i!} \qquad i \in \{0, 1, 2, \ldots\}, \tag{2.20}$$

where λ_n is the nth state's firing rate, η_{ni} is the probability of observing i spikes during some time step given that the neuron is in state n, and dt is the time step duration (Abeles et al., 1995).

Similarly, for the development of our model that follows, it will be convenient to define the transition matrix $\boldsymbol{\alpha}$ in terms of rates. This extension is slightly more complicated because it is nonsensical to allow multiple transitions to occur from state n to state m during a single time step. Therefore, we use the following model:

$$\alpha_{nm} = \begin{cases} \dfrac{\lambda'_{nm} dt}{1 + \sum_{l \neq n} \lambda'_{nl} dt} & m \neq n \\[2ex] \dfrac{1}{1 + \sum_{l \neq n} \lambda'_{nl} dt} & m = n \end{cases}, \tag{2.21}$$

where λ'_{nm} is the "pseudo-rate" of transitioning from state n to state m. (Throughout the article, the $'$ notation is used to denote rates and parameters associated with transitioning as opposed to spiking. Here, for example, λ' is a transition rate, while λ is a firing rate.) This definition of $\boldsymbol{\alpha}$ is convenient because it restricts transitions to at most one per time step (i.e., if $m \neq n$) and guarantees that the rows of $\boldsymbol{\alpha}$ sum to one. Furthermore, in the limit of small dt, the pseudo-rates become true rates (i.e., the probabilities of transitioning become proportional to the rates):

$$dt \to 0 \quad \Longrightarrow \quad \alpha_{nm} \propto \lambda'_{nm}. \tag{2.22}$$

2.2.2 Incorporating Stimulus and Spike History Dependence. In our model we permit the spike trains to be dependent on an external, time-varying stimulus $\mathbf{S} \equiv (\mathbf{s}_1 \cdots \mathbf{s}_T)$, where $\mathbf{s}_t$ is the stimulus at time t. The vector $\mathbf{s}_t$ has a length equal to the dimensionality of the stimulus. For example, if the stimulus is a 10×10 pixel image patch, then $\mathbf{s}_t$ would be a 100-element vector corresponding to the pixels of the patch. In the general case, $\mathbf{s}_t$ can also include past stimulus information.

We incorporate stimulus dependence in our model by allowing the transition and firing rates to vary with time as functions defined by linear-nonlinear filterings of the stimulus $\mathbf{s}_t$. In this time-inhomogeneous model, we have

$$\lambda'_{nm,t} = g\big({\mathbf{k}'_{nm}}^{\mathrm{T}}\mathbf{s}_t + b'_{nm}\big) \tag{2.23}$$

and

$$\lambda_{n,t} = f\Big({\mathbf{k}_n}^{\mathrm{T}}\mathbf{s}_t + b_n\Big), \tag{2.24}$$

where $\mathbf{k}'_{nm}$ and $\mathbf{k}_n$ are linear filters that describe the neuron's preferred directions in stimulus space for transitioning and firing, respectively, and g and f are nonlinear rate functions mapping real scalar inputs to nonnegative scalar outputs. In the absence of a stimulus (i.e., when $\mathbf{s}_t = \mathbf{0}$), the bias terms b'_{nm} and b_n determine the background transitioning and firing rates as $g(b'_{nm})$ and $f(b_n)$ respectively. It is possible to simplify the notation by augmenting the filter and stimulus vectors according to

$$\mathbf{k} \leftarrow \begin{bmatrix} \mathbf{k} \\ b \end{bmatrix} \tag{2.25}$$

and

$$\mathbf{s}_t \leftarrow \begin{bmatrix} \mathbf{s}_t \\ 1 \end{bmatrix}. \tag{2.26}$$

Then equations 2.23 and 2.24 reduce to

$$\lambda'_{nm,t} = g\big({\mathbf{k}'_{nm}}^{\mathrm{T}}\mathbf{s}_t\big) \tag{2.27}$$

and

$$\lambda_{n,t} = f\left(\mathbf{k}_n{}^{\mathrm{T}}\mathbf{s}_t\right). \tag{2.28}$$

The $\mathbf{k}_n$ stimulus filters for firing are the N preferred stimuli or receptive fields associated with each of the N states of the neuron. In the degenerate case where $N = 1$, the model reduces to a standard GLM model, and $\mathbf{k}_1$ becomes the canonical receptive field. The $\mathbf{k}'_{nm}$ stimulus filters for transitioning are, by analogy, "receptive fields" for transitioning, and since there are $N(N - 1)$ of these, there are N^2 total transition and firing stimulus filters describing the full model. This stimulus-dependent HMM is represented graphically in Figure 2b.

The manner in which spike history dependence enters into the rate equations is mathematically equivalent to that of the stimulus dependence. First, to introduce some notation, let $\boldsymbol{\gamma}_t$ be the vector of the spike counts for each of the τ time steps prior to t:

$$\boldsymbol{\gamma}_t \equiv \left(y_{t-1}, \ldots, y_{t-\tau}\right)^{\mathrm{T}}. \tag{2.29}$$

Then the transition and firing rate equations are modified by additional linear terms as

$$\lambda'_{nm,t} = g\left(\mathbf{k}'_{nm}{}^{\mathrm{T}}\mathbf{s}_t + \mathbf{h}'_{nm}{}^{\mathrm{T}}\boldsymbol{\gamma}_t\right) \tag{2.30}$$

and

$$\lambda_{n,t} = f\left(\mathbf{k}_n{}^{\mathrm{T}}\mathbf{s}_t + \mathbf{h}_n{}^{\mathrm{T}}\boldsymbol{\gamma}_t\right), \tag{2.31}$$

where $\mathbf{h}'_{nm}$ and $\mathbf{h}_n$ are weight vectors or linear filters that describe the neuron's preferred spike history patterns for transitioning and firing, respectively. The effect of adding history dependence to the rate equations is captured in Figure 2c.

As in the case of the stimulus filters, there are N^2 history filters. Thus, adding history dependence introduces τN^2 additional parameters to the model, and if dt is much smaller than the maximal duration of history effects, τ can be large, which can lead to a significant increase in the number of parameters. One way to reduce the number of parameters associated with history dependence is to assume that the history filters are linear combinations of H fixed-basis filters $\{\mathbf{e}_1, \ldots, \mathbf{e}_H\}$ where $H < \tau$. These basis filters could, for example, be exponentials with appropriately chosen time constants. We can then define $\mathbf{h}$ to be the H-element vector of coefficients corresponding to the linear combination composing the history filter rather than the history filter itself. In this formulation, the spike history data vector $\boldsymbol{\gamma}_t$ is redefined as

$$\boldsymbol{\gamma}_t \equiv [\mathbf{e}_1 \cdots \mathbf{e}_H]^{\mathrm{T}} \begin{bmatrix} y_{t-1} \\ \vdots \\ y_{t-\tau} \end{bmatrix}, \tag{2.32}$$

while the transition and firing rate equations remain unchanged (equations 2.30 and 2.31 respectively).

Since either choice of spike history dependence simply adds linear terms to the rate equations and since either formulation of $\boldsymbol{\gamma}_t$ can be precomputed directly from the spike train $\mathbf{y}$ with equations 2.29 and 2.32, we can safely augment $\mathbf{k}$ and $\mathbf{s}_t$ with $\mathbf{h}$ and $\boldsymbol{\gamma}_t$, as in equations 2.25 and 2.26. Thus, for the remainder of this article, without loss of generality, we will consider only equations 2.27 and 2.28 for both history-dependent and history-independent models.

2.2.3 Summary of Model. We have redefined the standard HMM transition and emission matrices $\boldsymbol{\alpha}$ and $\boldsymbol{\eta}$ to be time-inhomogeneous matrices $\boldsymbol{\alpha}_t$ and $\boldsymbol{\eta}_t$ defined by rates λ'_t and λ_t, which in turn are calculated from linear-nonlinear filterings of the stimulus $\mathbf{s}_t$ and the spike history $\boldsymbol{\gamma}_t$. Specifically, the transition matrix in the final model is

$$\alpha_{nm,t} = \begin{cases} \dfrac{g\left(\mathbf{k}'_{nm}{}^{\mathrm{T}}\mathbf{s}_t\right) dt}{1+\sum_{l\neq n} g\left(\mathbf{k}'_{nl}{}^{\mathrm{T}}\mathbf{s}_t\right) dt} & m \neq n \\ \dfrac{1}{1+\sum_{l\neq n} g\left(\mathbf{k}'_{nl}{}^{\mathrm{T}}\mathbf{s}_t\right) dt} & m = n \end{cases}, \tag{2.33}$$

and the emission matrix is

$$\eta_{ni,t} = \frac{\left(f\left(\mathbf{k}_n{}^{\mathrm{T}}\mathbf{s}_t\right) dt\right)^i e^{-f\left(\mathbf{k}_n{}^{\mathrm{T}}\mathbf{s}_t\right) dt}}{i!} \qquad i \in \{0, 1, 2, \ldots\}. \tag{2.34}$$

Therefore, with N hidden states, the parameters of the model $\boldsymbol{\theta}$ are the $N(N-1)$ $\mathbf{k}'$ transition filters, the N $\mathbf{k}$ spiking filters, and the initial state distribution $\boldsymbol{\pi}$. Since the number of parameters grows quadratically with N, it may be desirable to consider reduced-parameter models in some contexts (see appendix A for discussion). The $\mathbf{k}$ filters are the state-specific receptive fields (and possible history filters) of the model neuron, while the $\mathbf{k}'$ filters are the "receptive fields" describing how the stimulus (and possibly spike history) influences the state transition dynamics.

2.2.4 Parameter Learning with Baum-Welch EM. In order to learn the model parameters from a spike train $\mathbf{y}$ given a stimulus $\mathbf{S}$, we employ Baum-Welch EM. The E-step remains completely unchanged by the modification to point-process, stimulus, and history-driven emission data. All references to $\boldsymbol{\alpha}$ and $\boldsymbol{\eta}$ in section 2.1.2 can simply be replaced by $\boldsymbol{\alpha}_t$ and $\boldsymbol{\eta}_t$ as defined in equations 2.33 and 2.34. For concreteness, we show the validity of the forward recursion, equation 2.9, under the full model:

$$\begin{aligned} a_{n,t} &\equiv p(\mathbf{y}_{[0:t]}, q_t = n \mid \mathbf{S}) \\ &= p(\mathbf{y}_{[0:t-1]}, q_t = n \mid \mathbf{S})\, p(y_t \mid q_t = n, \mathbf{y}_{[0:t-1]}, \mathbf{S}) \\ &= \left(\sum_{m=1}^{N} p(\mathbf{y}_{[0:t-1]}, q_{t-1} = m, q_t = n \mid \mathbf{S}) \right) p(y_t \mid q_t = n, \mathbf{y}_{[0:t-1]}, \mathbf{S}) \\ &= \left(\sum_{m=1}^{N} p(\mathbf{y}_{[0:t-1]}, q_{t-1} = m \mid \mathbf{S})\, p(q_t = n \mid q_{t-1} = m, \mathbf{y}_{[0:t-1]}, \mathbf{S}) \right) \\ &\quad \times p(y_t \mid q_t = n, \mathbf{y}_{[0:t-1]}, \mathbf{S}) \\ &= \left(\sum_{m=1}^{N} p(\mathbf{y}_{[0:t-1]}, q_{t-1} = m \mid \mathbf{S})\, p(q_t = n \mid q_{t-1} = m, \mathbf{y}_{[t-\tau:t-1]}, \mathbf{S}) \right) \\ &\quad \times p(y_t \mid q_t = n, \mathbf{y}_{[t-\tau:t-1]}, \mathbf{S}) \\ &= \left(\sum_{m=1}^{N} a_{m,t-1} \alpha_{mn,t} \right) \eta_{ny_t,t}. \end{aligned} \tag{2.35}$$

Through a similar calculation, the backward recursion can also be shown to be unchanged from equation 2.13.

The expression for the expected complete log likelihood (ECLL) that needs to be maximized during the M-step can be found by substituting the definitions of $\boldsymbol{\alpha}_t$ and $\boldsymbol{\eta}_t$ into equation 2.18:

$$\begin{aligned} \left\langle L(\boldsymbol{\theta} \mid \mathbf{y}, \mathbf{q}, \mathbf{S}) \right\rangle_{\hat{p}(\mathbf{q})} = & \sum_{n=1}^{N} \hat{p}(q_0 = n) \log \pi_n \\ & + \sum_{t=1}^{T} \sum_{n=1}^{N} \sum_{m=1}^{N} \hat{p}(q_{t-1} = n, q_t = m) \log \alpha_{nm,t} \\ & + \sum_{t=0}^{T} \sum_{n=1}^{N} \hat{p}(q_t = n) \log \eta_{ny_t,t}, \end{aligned} \tag{2.36}$$

where $\hat{p}(\cdot)$ now also depends on the stimulus $\mathbf{S}$: $\hat{p}(\cdot) = p(\cdot \mid \mathbf{y}, \mathbf{S}, \boldsymbol{\theta}^i)$. Since the parameters of $\boldsymbol{\pi}$, $\boldsymbol{\alpha}_t$, and $\boldsymbol{\eta}_t$ enter into the above expression in a separable manner, we can consider the three terms of equation 2.36 in turn and maximize each independent of the others. Maximizing the $\boldsymbol{\pi}$ term proceeds as before (see equation 2.19). For the $\boldsymbol{\alpha}_t$ term in the ECLL, we have

$$\begin{aligned}
&\sum_{t=1}^{T}\sum_{n=1}^{N}\sum_{m=1}^{N} \hat{p}(q_{t-1} = n, q_t = m) \log \alpha_{nm,t} \\
&= \sum_{t=1}^{T}\sum_{n=1}^{N} \left(\begin{array}{l} \displaystyle\sum_{m \neq n} \hat{p}(q_{t-1} = n, q_t = m) \log \frac{g\left({\mathbf{k}'_{nm}}^{\mathrm{T}}\mathbf{s}_t\right) dt}{1 + \sum_{l \neq n} g\left({\mathbf{k}'_{nl}}^{\mathrm{T}}\mathbf{s}_t\right) dt} \\ \displaystyle + \hat{p}(q_{t-1} = n, q_t = n) \log \frac{1}{1 + \sum_{l \neq n} g\left({\mathbf{k}'_{nl}}^{\mathrm{T}}\mathbf{s}_t\right) dt} \end{array} \right) \\
&\sim \sum_{t=1}^{T}\sum_{n=1}^{N} \left(\begin{array}{l} \displaystyle\sum_{m \neq n} \hat{p}(q_{t-1} = n, q_t = m) \log g\left({\mathbf{k}'_{nm}}^{\mathrm{T}}\mathbf{s}_t\right) \\ \displaystyle - \hat{p}(q_{t-1} = n) \log \left(1 + \sum_{l \neq n} g\left({\mathbf{k}'_{nl}}^{\mathrm{T}}\mathbf{s}_t\right) dt \right) \end{array} \right),
\end{aligned} \tag{2.37}$$

where we have made use of the identity $\sum_m \hat{p}(q_{t-1} = n, q_t = m) = \hat{p}(q_{t-1} = n)$. The $\boldsymbol{\eta}_t$ term reduces as

$$\begin{aligned}
&\sum_{t=0}^{T}\sum_{n=1}^{N} \hat{p}(q_t = n) \log \eta_{ny_t,t} \\
&= \sum_{t=0}^{T}\sum_{n=1}^{N} \hat{p}(q_t = n) \log \frac{\left(f\left({\mathbf{k}_n}^{\mathrm{T}}\mathbf{s}_t\right) dt\right)^{y_t} e^{-f\left({\mathbf{k}_n}^{\mathrm{T}}\mathbf{s}_t\right) dt}}{y_t!} \\
&\sim \sum_{t=0}^{T}\sum_{n=1}^{N} \hat{p}(q_t = n)\left(y_t \log f\left({\mathbf{k}_n}^{\mathrm{T}}\mathbf{s}_t\right) - f\left({\mathbf{k}_n}^{\mathrm{T}}\mathbf{s}_t\right) dt\right).
\end{aligned} \tag{2.38}$$

We employ gradient ascent methods to maximize equations 2.37 and 2.38 (see appendix B for the necessary gradients and Hessians). Unfortunately, in the general case, there is no guarantee that the ECLL has a unique maximum. However, if the nonlinearities g and f are chosen from a restricted set of functions, it is possible to ensure that the ECLL is concave and smooth with respect to the parameters of the model $\mathbf{k}'_{nm}$ and $\mathbf{k}_n$, and therefore each M-step has a global maximizer that can be easily found with a gradient ascent technique. The appropriate choices of g and f are discussed in section 2.2.6.

2.2.5 Modeling Multicell and Multitrial Data. One major motivation for the application of the HMM framework to neural data is that the hidden variable can be thought of as representing the overall state of the neural network from which the data are recorded. Thus, if multiple spike trains are simultaneously observed (e.g., with tetrodes or multielectrode arrays), an HMM can be used to model the correlated activity between the single units (under the assumption that each of the behaviors of the single units depends on the hidden state of the entire network as in Abeles et al., 1995; Seidemann et al., 1996; Gat, Tishby, & Abeles, 1997; Yu et al., 2006; Jones et al., 2007; Kulkarni & Paninski, 2007). Additionally, if the same stimulus is repeated to the same experimental preparation, the data collected on each trial can be combined to improve parameter estimation. In this section, we provide the extension of our stimulus- and history-dependent framework to the regime of data sets with C simultaneously recorded spike trains and R independent trials.

In the single-cell case, we considered the state-dependent emission probability $p(y_t \mid q_t, \mathbf{S})$, with y_t being the number of spikes in time-step t. We now consider the joint probability of the spiking behavior of all C cells at time t conditioned on state q_t, or $p(y_t^1, \ldots, y_t^C \mid q_t, \mathbf{S})$. We factorize

$$\begin{aligned} p\big(y_t^1, \ldots, y_t^C \mid q_t, \mathbf{S}\big) &= \prod_{c=1}^{C} p(y_t^c \mid q_t, \mathbf{S}) \\ &= \prod_{c=1}^{C} \eta^c_{q_t y_t^c}, \end{aligned} \tag{2.39}$$

where each cell-specific emission matrix $\boldsymbol{\eta}^c$ is defined according to the Poisson distribution as before (see equation 2.20):

$$\eta^c_{ni} = \frac{(\lambda_n^c\, dt)^i\, e^{-\lambda_n^c dt}}{i!} \qquad i \in \{0, 1, 2, \ldots\}, \tag{2.40}$$

with λ_n^c as the state- and cell-specific firing rate for cell c in state n, given the observed stimulus and past spike history. The time-varying rates also retain their definitions from the single-cell setting (see equation 2.28):

$$\lambda^c_{n,t} = f\big(\mathbf{k}_n^{c\,\mathrm{T}} \mathbf{s}_t\big). \tag{2.41}$$

Note that the number of transition filters for the multicell model is unchanged ($N^2 - N$), but that the number of spiking filters is increased from N to NC (i.e., there is one spiking filter per state per cell).

Learning the parameters of this multicell model via Baum-Welch EM is essentially unchanged. For the E-step, all references to η_{ny_t} in the expressions for the forward and backward recursions as presented in section 2.1.2 are simply replaced with the product $\prod_c \eta^c_{ny^c_t}$ to give the corresponding expressions for the multicell setting. For the M-step, we note that the complete log likelihood for the multicell model is modified from equation 2.5 only in the final term:

$$L(\boldsymbol{\theta} \mid \mathbf{Y}, \mathbf{q}) = \log \pi_{q_0} + \sum_{t=1}^{T} \log \alpha_{q_{t-1}q_t} + \sum_{c=1}^{C} \sum_{t=0}^{T} \log \eta^c_{q_t y^c_t}, \tag{2.42}$$

where $\mathbf{Y} \equiv (\mathbf{y}^1 \cdots \mathbf{y}^C)$. Thus, the emission component of the ECLL (see equation 2.38) becomes

$$\begin{aligned} &\sum_{c=1}^{C} \sum_{t=0}^{T} \sum_{n=1}^{N} \hat{p}(q_t = n) \log \eta^c_{ny^c_t,t} \\ &\sim \sum_{c=1}^{C} \sum_{t=0}^{T} \sum_{n=1}^{N} \hat{p}(q_t = n)\big(y^c_t \log f\big({\mathbf{k}^c_n}^{\mathrm{T}} \mathbf{s}_t\big) - f\big({\mathbf{k}^c_n}^{\mathrm{T}} \mathbf{s}_t\big)\, dt\big). \end{aligned} \tag{2.43}$$

Since the cell-specific filters $\mathbf{k}^c_n$ enter into equation 2.43 in a separable manner, the parameters for each cell can again be learned independently by climbing the gradient of the cell-specific component of the ECLL. Thus, the gradient and Hessian given in appendix B can be used in the multicell setting without modification.

To allow for multiple trials, we note that if each of the R trials is independent of the rest, then the total log likelihood for all of the trials is simply the sum of the log likelihoods for each of the individual trials. Thus, to get the total log likelihood, the forward-backward algorithm is run on each trial r separately, and the resultant trial-specific log likelihoods are summed. The M-step is similarly modified, as the total ECLL is again just the sum of the trial-specific ECLLs:

$$\begin{aligned} &\big\langle L\big(\boldsymbol{\theta} \mid \mathbf{Y}^1, \mathbf{q}^1, \ldots, \mathbf{Y}^R, \mathbf{q}^R\big)\big\rangle_{\hat{p}(\mathbf{q}^1,\ldots,\mathbf{q}^R)} \\ &\quad = \sum_{r=1}^{R} \sum_{n=1}^{N} \hat{p}(q^r_0 = n) \log \pi_n + \sum_{n=1}^{N} \sum_{r=1}^{R} \sum_{t=1}^{T} \sum_{m=1}^{N} \hat{p}(q^r_{t-1} = n, q^r_t = m) \log \alpha_{nm,t} \\ &\qquad + \sum_{c=1}^{C} \sum_{r=1}^{R} \sum_{t=0}^{T} \sum_{n=1}^{N} \hat{p}(q^r_t = n) \log \eta^c_{ny^{c,r}_t,t}, \end{aligned} \tag{2.44}$$

where $\hat{p}(q^r_t)$ and $\hat{p}(q^r_{t-1}, q^r_t)$ are the trial-specific single- and consecutive-pairwise marginals of the posterior distribution over the state sequence given by the forward-backward algorithm applied to each trial during the E-step, and $y^{c,r}_t$ is the number of spikes by cell c in the tth time step of the rth trial. The M-step update for the start-state distribution (see equation 2.19) is modified as

$$\pi_n = \frac{1}{R}\sum_{r=1}^{R} \hat{p}(q_0^r = n). \tag{2.45}$$

To update the transition and spiking filters, gradient ascent is performed as in the single trial setting, except that the trial-specific gradients and Hessians for each filter are simply summed to give the complete gradients and Hessians. Note that the order of the sums in equation 2.44 represents the fact that the parameters that determine the transitioning behavior away from each state n are independent of each other, as are the parameters that determine the spiking behavior for each cell c, and so these sets of parameters can be updated independently during each M-step.

2.2.6 Guaranteeing the Concavity of the M-step. As Paninski (2004) noted for the standard GLM model, the ECLL in equation 2.38 depends on the model parameters through the spiking nonlinearity f via a sum of terms involving $\log f(u)$ and $-f(u)$. Since the sum of concave functions is concave, the concavity of the ECLL will be ensured if we constrain $\log f(u)$ to be a concave function and $f(u)$ to be a convex function of its argument u. Example nonlinearities that satisfy these log concavity and convexity constraints include the exponential, rectified linear, and rectified power law functions.

The concavity constraints for the transition rate nonlinearity are significantly more stringent. From equation 2.37, we see that g enters into the ECLL in two logarithmic forms: $\log g(u)$ and $-\log(1 + \sum_i g(u_i)\,dt)$, where each u_i is a function of the parameters corresponding to a single transition (e.g., $\mathbf{k}'_{nm}$). The first term gives rise to the familiar constraint that g must be log concave. To analyze the second term, we consider the limiting case where $g(u_j)dt \gg 1$ and $g(u_j) \gg g(u_i), \forall i \neq j$ (which will be true for some setting of the parameters that compose the u_i). Then the second logarithmic term reduces to $-\log g(u_j)$, which introduces the necessary condition that g must be log convex. Explicit derivation of the second derivative matrix of $-\log(1 + \sum_i g(u_i)dt)$ confirms that the log convexity of the nonlinearity is sufficient to guarantee that this matrix is negative-definite for all values of the u_i (i.e., not just in the limiting case). The only functions that are both log concave and log convex are those that grow exponentially, and thus, if the transition nonlinearity is the exponential function (if $g(u) = e^u$), the concavity of the M-step will be guaranteed.[1]

[1] In general, any function of the form e^{cu+d} satisfies log concavity and log convexity in u. But for our model where $u = \mathbf{k}^{\mathrm{T}}\mathbf{s}_t + b$, the parameters c and d can be eliminated by scaling the filter $\mathbf{k}$ and changing the bias term b. Thus, we can restrict ourselves to consider only the nonlinearity e^u without loss of generality.

2.2.7 Using a Continuous-Time Model. It is possible to adapt our model to a continuous-time rather than a discrete-time framework (see appendix C for details). This has several potential advantages. First, as the derivation of the continuous-time M-step reveals, the stringent requirement that the transition rate nonlinearity g must be the exponential can be relaxed while maintaining the concavity of the ECLL. This significantly increases the flexibility of the model. More important, however, the continuous-time implementation may require less computation and memory storage. During the E-step in the discrete-time case, the forward and backward probabilities are calculated for every time step t. When one considers the fact that the vast majority of time steps for a reasonable choice of dt ($\leq$10 ms) are associated with the trivial "no-spike" emission even for neurons with relatively high firing rates, it becomes obvious why a continuous-time framework might potentially be advantageous since it is possible to numerically integrate the forward and backward probabilities from spike time to spike time. Using an ODE solver to perform this integration is effectively the same as using adaptive time stepping where dt is modified to reflect the rate at which the probabilities are changing. This can result in significantly fewer computations per iteration than in the discrete-time case. Additionally, it is necessary to store the marginal probabilities of the posterior distribution (for eventual use during the M-step) only at the time points where the ordinary differential equation (ODE) solver chooses to evaluate them, which is likely to be many fewer total points. Although the results we present below were all obtained using a discrete-time algorithm, for the reasons just mentioned, implementation of a continuous-time model may be more appropriate for certain large data sets, specifically those with highly varying firing rates where a single time step would be either too computationally expensive or would result in a loss of the finely grained structure in the data.

2.3 Hybrid Peristimulus Time Histogram and Hidden Markov Models. As a demonstration of how the framework introduced in this article can be extended to more appropriately suit certain data sets, in this section we introduce modifications that allow the modeling of state-dependent firing rates when the rates are not being driven by an explicit time-varying stimulus, but simultaneously are not time homogeneous. Many experimental data sets consist of multiple trials that are triggered by an event and exhibit interesting event-triggered dynamics (see, e.g., the data discussed in section 4). Assuming these dynamics evolve in a state-dependent manner, the ability to model such inhomogeneous systems with HMMs is potentially useful. It is important to note, however, that the models that we introduce in sections 2.3.1 and 2.3.2, while mathematically very similar to those already presented, differ philosophically in that they are not specifically motivated by a simple neural mechanism. In the previous models, firing rate changes are driven by the time-varying stimulus. In these models, though they allow the capture of firing rate changes, the genesis of these inhomogeneities is not explicitly defined (although these models are designed to provide a snapshot of the dynamics of an underlying neural network from which a data set is recorded).

2.3.1 The Trial-Triggered Model. As our first example, we consider a model in which the transition and firing rates depend on the time t since the beginning of the trial (in addition to the hidden state q_t and, possibly, spike history effects). For simplicity, we assume that no time-varying stimulus is present, although incorporating additional stimulus terms is straightforward. Explicitly, we can model the transition and firing rates as $\lambda'_{nm,t} = g([\mathbf{k}'_{nm}]_t)$ and $\lambda^c_{n,t} = f([\mathbf{k}^c_n]_t + \mathbf{h}^{c\,\mathrm{T}}_n \boldsymbol{\gamma}^c_t)$, respectively, where the spike history effects $\mathbf{h}^{c\,\mathrm{T}}_n \boldsymbol{\gamma}^c_t$ are as defined in section 2.2.2, $[\mathbf{k}]_t$ is the tth element of $\mathbf{k}$, and the filters $\mathbf{k}'_{nm}$ and $\mathbf{k}^c_n$ are now time-varying functions of length T (see Kass & Ventura, 2001; Frank, Eden, Solo, Wilson, & Brown, 2002; Kass, Ventura, & Cai, 2003; Czanner et al., 2008; Paninski et al., 2009, for discussion of related models). In principle, the filter elements can take arbitrary values at each time t, but clearly estimating such arbitrary functions given limited data would lead to overfitting. Thus, we may either represent the filters in a lower-dimensional basis set (as we discussed with $\mathbf{h}_n$ in section 2.2.2), such as a set of splines (Wahba, 1990), or we can take a penalized maximum likelihood approach to obtain smoothly varying filters (where the difference between adjacent filter elements $[\mathbf{k}]_t$ and $[\mathbf{k}]_{t+1}$ must be small), or potentially combine these two approaches. (For a full treatment of a penalized maximum likelihood approach to this "trial-triggered" model, see appendix D.)

A convenient feature of using the smoothness penalty formulation is that the model then completely encapsulates the homogeneous HMM with static firing rates in each state. If the smoothness penalties are set such that the difference between adjacent filter elements is constrained to be essentially zero, then the spiking and transition filters will be flat, or, equivalently, the model will become time homogeneous. In the opposite limit, as discussed above, if the penalties are set such that the differences between adjacent elements can be very large, then we revert to the standard maximum likelihood setting, where overfitting is ensured. Thus, by using model selection approaches for choosing the setting of the penalty parameter (e.g., with cross-validation or empirical Bayes as in Rahnama Rad and Paninski, 2010), it is possible to determine the optimal level of smoothness required of the spiking and transitioning filters.

It is clear that this proposed trial-triggered model is a hybrid between a standard peristimulus time histogram-based (PSTH) model and a time-homogeneous HMM. Although we have been able to reformulate this model to fit exactly into our framework, it is illustrative to consider the model, rather than as an N-state time-inhomogeneous model with an unrestricted transition matrix (as in Figure 1), as an NT-state time-homogeneous model with a restricted state-space connectivity. In this interpretation, state n_t is associated with the $N-1$ transition rates $\lambda'_{n_t m_{t+1}} \equiv g(k'_{n_t m_{t+1}})$ for $m \neq n$

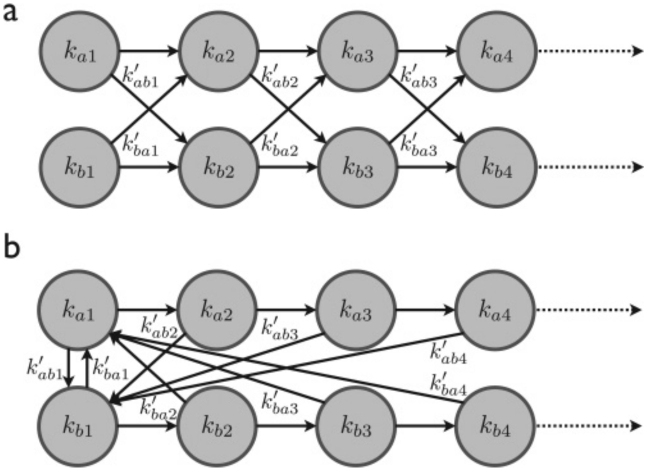

Figure 3: The time-expanded Markov chains representing the trial-triggered and transition-trial models. (a) The trial-triggered model. At time step t, the neuronal ensemble is in some state n_t, where its firing rates are determined by the parameters $\{k^1_{n_t}, \ldots, k^C_{n_t}\}$. At the following time step, the system is forced to move one step to the right (to the column corresponding to time $t+1$) and change rows depending on the transition rates given by the $N-1$ parameters $k'_{n_t m_{t+1}}$ for $m \neq n$. The firing and transition rates associated with each row of states change gradually with increasing t due to the application of smoothness priors (see appendix D). Note that there are no self-transitions in this model; whether the state changes rows or not, at every time step, it moves one column to the right. (b) The transition-triggered model. The firing rates are associated with each state as in a, but the model must now either advance along a row or transition back to the first column of states. Therefore, after the first such transition, the time-step t and the depth in the Markov chain τ become decoupled. This allows the intra state dynamics to evolve from the time that the neuron enters a state (or, more accurately, a row of states) rather than from the time of the onset of the trial.

leading to the states available at time $t+1$.[2] In other words, the transition matrix is sparse with nonzero entries only between states corresponding to adjacent time steps. Note that this model is exactly the same as before, merely represented in a different way. Conceptually, a small state space with dynamic firing and transition rates has been replaced by a large state space with static rates. A schema of the Markov chain underlying the trial-triggered model is given in Figure 3a. Each row of states in the figure

[2] Note that no parameter is needed for the transition from state n_t to n_{t+1}, as this is the default behavior of the system in the absence of any other transition (see equation 2.21).

corresponds to what was a single state in the previous representation of the model, and each column corresponds to a time step t. Due to the restricted nature of the state-space connectivity (i.e., the few nonzero entries in the transition matrix), the system will always be in a state of the tth column at time t.

2.3.2 The Transition-Triggered Model. Another possible extension of this framework is illustrated in Figure 3b. The idea is to couple the dynamics of the system to the times at which the state transitions rather than the start of the trial. This model structure is closely connected to the "semi-Markov" models and other related models described previously (Sansom & Thomson, 2001; Guédon, 2003; Fox, Sudderth, Jordan, & Willsky, 2008; Chen et al., 2009; Tokdar et al., 2009), as we will discuss further below. In this model, transitions that result in a change of row reset the system to the first column of the state space, as opposed to the trial-triggered model, where all transitions move the system to the next column. In this transition-triggered model, we label the current state as n_τ, which is the τth state in the nth row of states. Note that the index τ is only equal to the time-step t prior to the first transition back to the first column of states. Subsequently, τ, which can be thought of as the depth of the current state in the state-space cascade, will reflect the time since the last transition, not the time since the onset of the trial, exactly as desired. The model parameters $\mathbf{k}_n$ and $\mathbf{k}'_{nm}$ can now be thought of as state-dependent peritransition time histograms (PTTHs) for spiking and transitioning (rather than PSTHs) due to the decoupling of τ from t. Note that each state n_τ is associated with N transition rates $\lambda'_{n_\tau m_0} \equiv g(k'_{n_\tau m_0})$ where m may equal n (unlike in the trial-triggered case, where each state was associated with $N-1$ transition rates) because we permit transitions back to the start of the current row. Additionally, recall that when the trial-triggered model was reformulated as having NT-states rather than N-states, the model became time homogeneous. For the transition-triggered model, however, since τ and t are decoupled, the firing rates for each state n_τ are no longer time homogeneous. A consequence is that the time complexity of the associated Baum-Welch learning algorithm becomes $\mathcal{O}(T^2)$ rather than $\mathcal{O}(T)$. For a full treatment of the transition-triggered model, see appendix E. Results from the analysis of real data using this model appear elsewhere (Escola, 2009).

3 Results with Simulated Data

In this section, we apply our algorithm to simulated data sets to test its ability to appropriately learn the parameters of the model.

3.1 Data Simulation. In our trials with simulated data, the stimuli used to drive the spiking of the model neurons are time correlated gaussian white noise stimuli with spatially independent and identically distributed (i.i.d.)

pixels. More specifically, the intensity of the stimulus at each pixel was given by an independent autoregressive process of order 1 with a mean of 0, a variance of 1, an autocorrelation of 200 ms, and a time step of 2 ms.

In order to generate simulated spike trains (via equation 2.28), we used the firing rate nonlinearity,

$$f(u) = \begin{cases} e^u & u \leq 0 \\ 1 + u + \frac{1}{2}u^2 & u > 0 \end{cases}. \tag{3.1}$$

This function f is continuous and has continuous first and second derivatives, thus facilitating learning in gradient algorithms. Furthermore, the properties of convexity and log concavity are also maintained, guaranteeing that the ECLL has a unique maximum (recall section 2.2.6). The nonlinearity g governing the transitioning behavior is selected to be the exponential function (also per section 2.2.6).

3.2 A Tonic and Burst Two-State Model.

We tested our learning algorithm on spike trains generated from a model representing tonic and burst thalamic relay cells. Experimental studies such as those reviewed in Sherman (2001) have shown that relay cells exhibit two distinct modes of firing. In the tonic mode (hereafter referred to as the tonic state), interspike intervals (ISIs) are approximately distributed according to an exponential distribution, suggesting that spikes are more or less independent and that a Poisson firing model is reasonable. In the burst state, neighboring spikes are highly correlated (they tend to occur in bursts), as indicated by a vastly different ISI distribution (Ramcharan, Gnadt, & Sherman, 2000), and thus any reasonable model must capture these correlations. To do so, we employed different spike history filters for the two states.

If the tonic state history filter $\mathbf{h}_t$ were the zero vector (where the subscripts t and b refer to the tonic and burst states, respectively), then tonic state spikes during a constant stimulus would be independent, leading to an exactly exponential ISI distribution. Instead we chose the history filter shown in Figure 4a, which has a large, negative value for the most recent time step, followed by small, near-zero values for earlier time steps. This negative value models the intrinsic refractoriness of neurons by strongly reducing the probability of a subsequent spike one time step (2 ms) after a preceding spike (recall how the spike history affects the firing rate according to equation 2.31). The resulting ISI distribution (in light gray in Figure 5) has low probability density for short intervals due to the imposed refractoriness, but it is otherwise essentially exponential.

The burst-state history filter $\mathbf{h}_b$ (see Figure 4b) has a similar negative value for the most recent time step and thus also models refractoriness, but it has strong, positive values for the previous two time steps. This has the effect of raising the probability of a spike following an earlier spike, and thus

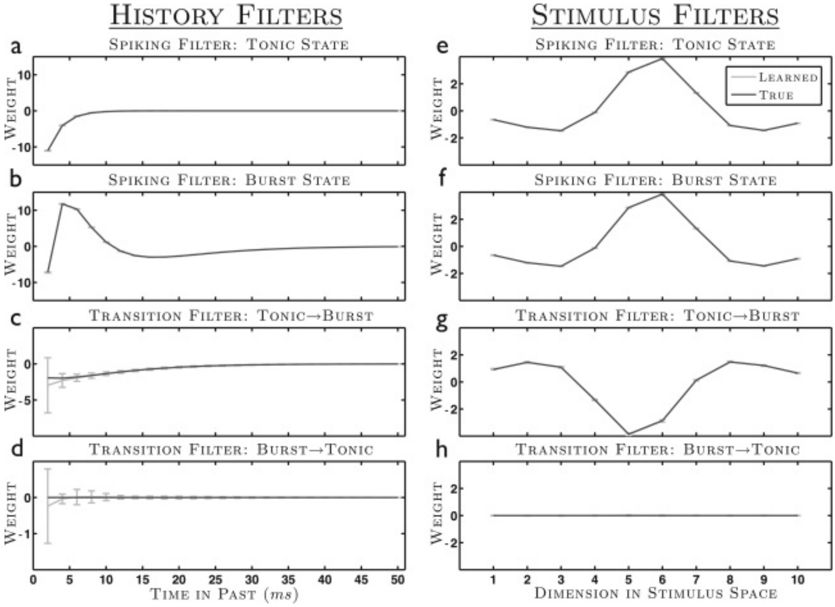

Figure 4: The true and learned stimulus and history filters of the tonic and burst thalamic relay cell described in section 3.2. For this model, the preferred stimulus for spiking is the same for both states. The history filters are actually parameterized by the coefficients of three exponential basis functions with 2, 4, and 8 ms time constants. For ease of visual interpretation, the filters, rather than the underlying parameters, are shown. All true parameter values (in dark gray) fall within the ±1 σ error bars (in light gray). Means and errors were calculated over 100 learning trials, each with a unique stimulus/spike train pair generated according to section 3.1. Parameters were initialized randomly from zero mean, unit variance gaussians. By visual inspection, all 100 trials converged to seemingly correct solutions (i.e., local minima were not encountered). As discussed in the text, the larger error bars shown for the history filter weights at 2 ms in the past reflect the fact that the data contain little information about the filter weights at this time resolution. (a) Spiking filter $\mathbf{h}_t$. (b) Spiking filter $\mathbf{h}_b$. (c) Transition filter $\mathbf{h}'_{tb}$. (d) Transition filter $\mathbf{h}'_{bt}$. (e) Spiking filter $\mathbf{k}_t$. (f) Spiking filter $\mathbf{k}_b$. (g) Transition filter $\mathbf{k}'_{tb}$. (h) Transition filter $\mathbf{k}'_{bt}$.

encourages bursting. Furthermore, the filter returns to negative values for more distant time steps, which tends to cause gaps between bursts, another known neurophysiological feature. The resulting ISI distribution (in dark gray in Figure 5) has the signature bimodal shape of bursting (Ramcharan et al., 2000).

A reasonable choice for the history filter for the transition from the tonic state to the burst state ($\mathbf{h}'_{tb}$) consists of negative values for the several time

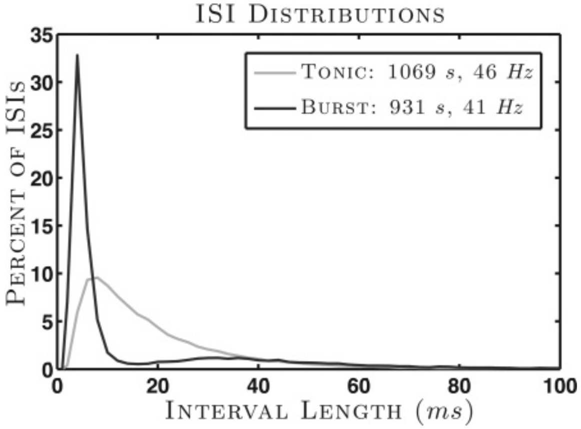

Figure 5: Interspike interval (ISI) distributions calculated for the tonic and burst states of the model neuron described in section 3.2 over 2000 s of simulated data. It is clear that the tonic state ISI is essentially exponential, excluding the refractory effects at small interval lengths. The burst state ISI has a sharp peak at very short intervals, followed by a reduction in interval probability at medium interval lengths. This pattern represents bursts separated by longer periods of silence, the physiological criteria for bursting. Total state dwell times and mean state firing rates are given in the legend.

steps preceding the transition. This is because bursts tend to follow periods of relative quiescence (Wang et al., 2007), and, with this choice of $\mathbf{h}'_{tb}$ (see Figure 4c),[3] the model neuron will prefer to transition to the burst state when there has not been a recent spike. We chose the history filter for the reverse transition ($\mathbf{h}'_{bt}$) to be the zero vector (see Figure 4d), and thus spike history does not affect the return to the tonic state from the burst state. To reduce the model dimensionality, the history filters were defined by the coefficients of three exponential basis functions with time constants 2, 4, and 8 ms (recall the discussion in section 2.2.2).

The stimulus filters for spiking for both states ($\mathbf{k}_t$ and $\mathbf{k}_b$; see Figures 4e and 4f, respectively) were chosen to be identical, following experimental evidence that the spatial component of the preferred stimulus does not change regardless of whether a relay cell is firing in the tonic or burst

[3] Comparing Figures 4c to 4a and 4b, one might conclude that $\mathbf{h}'_{tb}$ is relatively insignificant due to the fact that the magnitudes of its weights are much less than those of $\mathbf{h}_t$ and $\mathbf{h}_b$. Recall, however, that the nonlinearity for transitioning g grows exponentially, while the nonlinearity for spiking f grow quadratically, so small-magnitude filter weights can still have pronounced effects on the transition rate.

regime (Bezdudnaya et al., 2006).[4] The spiking bias terms were set such that the background firing rates were 45 Hz in both states: $f(b_t) = f(b_b) = 45$ Hz.

To choose the stimulus filter for the transition from the tonic state to the burst state ($\mathbf{k}'_{tb}$; see Figure 4g), we used a similar line of reasoning as in the choice of the corresponding history filter. Since bursts tend to follow periods of quiescence, we selected as this transition filter the negative of the spiking filter. Thus, the antipreferred stimulus would drive the cell into the burst state, where the preferred stimulus could then trigger bursting. This is reasonable from a neurophysiological point of view by noting that voltage recordings from patched cells have shown hyperpolarized membrane potentials immediately prior to bursts (Sherman, 2001; Wang et al., 2007) and that an antipreferred stimulus would be expected to hyperpolarize a neuron through push-pull inhibition. The stimulus filter for the reverse transition $\mathbf{k}'_{bt}$, as with $\mathbf{h}'_{bt}$, was chosen to be the zero vector (see Figure 4h). Thus, the return to the tonic state in this model is governed solely by the background transition rate. The bias terms b'_{tb} and b'_{bt} were set such that the background transition rates were 3 Hz and 7 Hz, respectively, for the tonic→burst and the burst→tonic transitions. When the neuron is presented with a stimulus, however, due to the variance of ${\mathbf{k}'_{tb}}^{\mathrm{T}}\mathbf{s}_t$ and the effects of the nonlinearity g, the average resultant rates are roughly equal for both transitions (approximately 7 Hz), and thus the model neuron spends about the same amount of time in each state.

When generating spike trains using these parameters, we changed the model slightly so as to restrict the number of spikes allowed per time step to be either zero or one. Specifically, we changed the emission matrix defined in equation 2.34 to be

$$\eta_{ny_t,t} = \begin{cases} e^{-\lambda_{n,t}dt} & y_t = \text{no spike} \\ 1 - e^{-\lambda_{n,t}dt} & y_t = \text{spike} \end{cases}. \tag{3.2}$$

This corresponds to thresholding the Poisson spike counts to form a Bernoulli (binary) process: when the Poisson spike count is greater than zero, we record a one for the Bernoulli process. Note that this Bernoulli formulation converges to the original Poisson formulation in the limit of small dt. Conveniently, the nonlinearity f has the same concavity constraints under this Bernoulli model as in the original Poisson model (see appendix F for proof).

[4] These experiments also show that the temporal component of the preferred stimulus differs between the two states, which we could model by including multiple time slices in the stimulus filters. For simplicity and reduction of parameters, we ignore the temporal differences in our model.

Using this spiking model and the parameter settings described above, we generated 2000 s spike trains as test data. Before iterating our learning algorithm, the filters and biases were initialized to random values drawn from zero mean, unit variance gaussians, while the initial state distribution $\boldsymbol{\pi}$ was initialized from an N-dimensional uniform distribution and then normalized to sum to 1. Learning proceeded according to the Baum-Welch EM algorithm described in sections 2.1.2, 2.1.3, and 2.2.4, with Newton-Raphson optimization used to perform the update of the parameters during the M-step (see section F.1 for the gradient and Hessian of the Bernoulli model). Considerable experimentation with the learning procedure suggested that except perhaps for the first one or two iterations of EM when the parameters are far from their correct values, a single Newton-Raphson step was sufficient to realize the parameter maximum for each M-step (i.e., the ECLL was very well approximated by a quadratic function). For these parameters and this amount of data, learning generally converged in about 200 to 300 iterations, which requires about 30 minutes of CPU time on an Apple 2.5 GHz dual-core Power Mac G5 with 3 GB of RAM running MATLAB.

Learning was repeated for 100 trials, each with a unique stimulus/spike train pair and a unique random initialization. By visual inspection, all trials appeared to avoid local minima and converged to reasonable solutions. The results for the history and stimulus filters (without bias terms) are shown in Figure 4. The $\pm 1\ \sigma$ error ranges for the bias terms (expressed in rate space) are 44.5 to 45.5 Hz, 44.6 to 45.4 Hz, 2.5 to 3.3 Hz, and 6.5 to 7.4 Hz, for b_t, b_b, b'_{tb}, and b'_{bt}, respectively. All true filter and bias parameters fall within the $\pm 1\ \sigma$ error ranges, suggesting that parameter learning was successful. The larger-than-average error bars for the weights of the transition history filters at 2 ms in the past (see Figures 4c and 4d) reflect the fact that spike trains contain little information about the dependence of the state transitioning on the spike history at very short timescales. The estimation of the consecutive-pairwise marginal probabilities of the posterior distribution of the state sequence (see equation 2.15) calculated by the forward-backward algorithm (see section 2.1.2) is not able to temporally localize the transitions to within a 2 ms precision even if the true parameters are used for the estimation. Therefore, one would need to average over a great deal more data to infer the dependence at this timescale than at slightly longer timescales. If more data were used to estimate the parameters, these error bars would be expected to decrease accordingly.

Although the parameter values appear to be learned appropriately, they are not learned perfectly. To understand the implication of these deviations, data generated using the true parameters can be compared to those generated using a sample learned parameter set. Rather than compare spike trains directly, it is sufficient to compare instantaneous firing rates, since the rate is a complete descriptor of Bernoulli (or Poisson) firing statistics. Figure 6a shows the instantaneous firing rates of two sample simulations

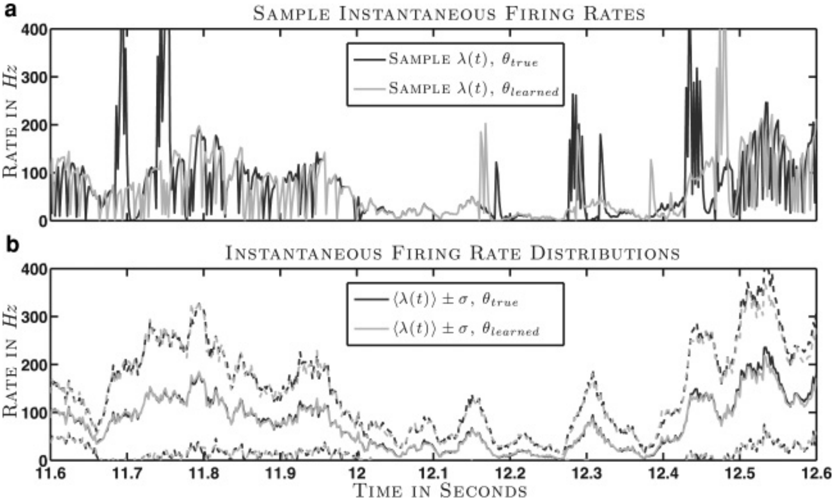

Figure 6: Instantaneous firing rate samples and distributions calculated using the true parameter values and a set of learned parameters for the tonic and burst model neuron discussed in section 3.2 during an illustrative 1 s time window of stimulus data. (a) The dark and light gray traces are the instantaneous firing rates of two sample simulations of the model, the former using the true parameters and the latter using the learned parameters. The two sample simulations differ significantly due to the fact that spike history affects the instantaneous firing rate. (b) The solid and dashed dark gray lines are, respectively, the mean and ±1 σ deviations of the instantaneous firing rate estimated from 1000 repeated simulations using the true parameters. The analogous mean and deviations estimated using the learned parameters are shown in light gray. The similarity of the two distributions confirms that learning was successful. The fact that the means and variances are conserved despite highly divergent individual sample firing rates suggests that the average rate over some window of time is a better descriptor of the behavior of the neuron than the instantaneous rate.

of the model using the same stimulus but two different parameter sets.[5] The most striking feature is how different the two traces seem from each other. This is because spikes in the two traces are very rarely coincident, and the spike history dependence dramatically alters the firing rates during the several milliseconds following a spike. This is apparent from the many dips in the firing rates to near-zero values (immediately after spikes), followed by relatively quick rebounds to the purely stimulus-evoked firing rate (the rate given by a spike history independent model). Also noticeable are the

[5] To remove any potential bias, the learned parameter set was not trained on the stimulus used to create Figure 6.

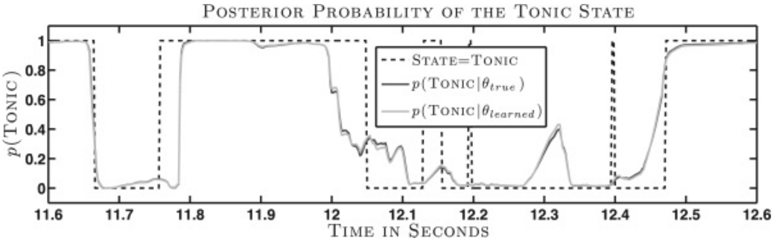

Figure 7: The posterior probability of the hidden state calculated using true and learned parameter values for the tonic and burst model during the same 1 s time window of stimulus data as in Figure 6. The dotted trace indicates when the model neuron was in the tonic state during the simulation corresponding to the sample shown in dark gray in Figure 6a (recall that for simulated data, the true state sequence is known). The dark gray trace is the posterior probability of the tonic state using the true parameters (as calculated with the forward-backward algorithm described in section 2.1.2), while the light gray trace corresponds to the posterior probability using the learned parameters. The similarity between the two posterior probability traces confirms that the learned parameters are as effective as the true parameters in recovering the hidden state sequence.

huge jumps in the firing rate corresponding to times when the neuron has been simulated to be in the burst state and is actively bursting.

The distributions of the instantaneous firing rates calculated over 1000 model simulations for both the true parameters and the set of learned parameters are shown in Figure 6b. Despite the fact that individual trials such as those shown in Figure 6a can differ significantly, the means and $\pm 1\,\sigma$ deviations are almost identical between the two distributions, confirming that the two parameter sets (true and learned) produce identical behavior in the model neuron. In other words, the interparameter set firing rate variability is no more than the intraparameter set firing rate variability.

To additionally evaluate the estimation performance, in Figure 7, we compare the posterior probability of the hidden state variable at each time step with the true state sequence. The trace corresponding to the posterior probability calculated using the learned parameters is essentially the same as that calculated using the true parameters, suggesting that both sets of parameters are equally able to extract all the information about the sequence of states that exists in the spike train. The difference between the true state sequence and the posterior probability calculated using the true parameters represents the intrinsic uncertainty in the system, which we cannot hope to remove. However, over 2000 s of stimulus/spike train data, the percentage of time steps when the true state was predicted with a posterior probability greater than 0.5 was 92%. These results support the fidelity of the learning procedure and suggest that it may be possible to use this method to recapitulate an unknown state sequence.

3.3 An Attentive/Ignoring Two-State Model. We also tested our learning algorithm on spike trains generated from a simulated neuron with two hidden states corresponding to stimulus-driven spiking and stimulus-ignoring spiking, respectively. This model could be interpreted to represent a neuron in primary sensory cortex. The attentive state would correspond to times when the synaptic current into the neuron is predominantly derived from thalamic afferents, and thus when the neuron's spiking behavior would be highly correlated with the sensory stimulus. The ignoring state would be associated with times when recurrent activity in the local cortical column or feedback activity from higher cortical areas overwhelms the inputs and drives the neuron's spiking in a stimulus-independent manner.

The ignoring state can be represented by setting the stimulus filter for spiking in that state to be zero for all elements except for the bias term: $\mathbf{k}_i = \left(\mathbf{0}^{\mathrm{T}}, b_i\right)^{\mathrm{T}}$, where the subscript i indicates the ignoring state (and a the attentive state). The stimulus filters of the model—$\mathbf{k}_a$, $\mathbf{k}_i$, $\mathbf{k}'_{ai}$, and $\mathbf{k}'_{ia}$—are shown in Figure 8 (history effects are ignored for this simple model). The forms of these filters are arbitrary choices (with the exception of $\mathbf{k}_i$), and the magnitudes of the filter values were chosen to be of the same order of magnitude as the zero mean, unit variance stimulus. The bias terms were set such that the background firing and transition rates in both states were 45 Hz and 0.1 Hz, respectively, which resulted in mean firing and transition rates in the presence of a stimulus of about 50 Hz and 9 Hz, respectively, due to the effects of the nonlinearities. Note that the original Poisson spiking model was used to generate the data for this example.

Learning proceeded as in the previous example and was repeated for 100 trials, each with a unique 2000 s stimulus/spike train pair and a unique random parameter initialization. By visual inspection, all trials appeared to avoid local minima and converged to reasonable solutions. The results for the filter parameters (without biases) are summarized in Figure 8. The $\pm 1\ \sigma$ error ranges for the bias terms (expressed in rate space) are 44.6 Hz to 45.4 Hz, 44.8 Hz to 45.2 Hz, 0.04 Hz to 0.13 Hz, and 0.07 Hz to 0.12 Hz, for b_a, b_i, b'_{ai}, and b'_{ia}, respectively. All true filter and bias parameters fall within the $\pm 1\ \sigma$ error ranges; thus, parameter learning was successful. For comparison purposes, the linear filter of a standard GLM (i.e., one-state) model was also learned. The resulting filter (shown with the dotted line in Figure 8a) differs significantly from the underlying stimulus filter for spiking $\mathbf{k}_a$ and seems to represent some combination of $\mathbf{k}_a$ and $\mathbf{k}_i$ (i.e., the two spiking filters), as well as $\mathbf{k}'_{ia}$, the transition filter that drives the neuron into the attentive state so that it can subsequently be driven to fire by the stimulus acting through $\mathbf{k}_a$.

As is shown in Figure 6b for the previous example, the distributions of the instantaneous firing rates calculated over many simulations of the

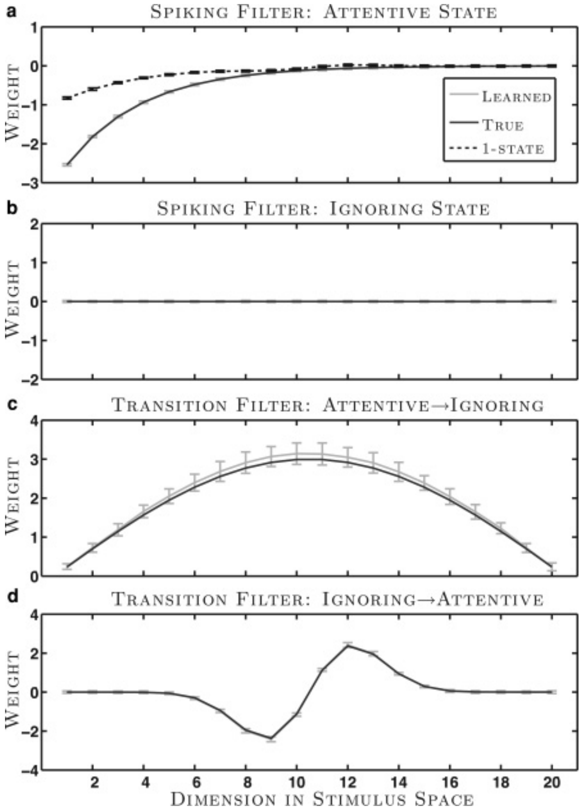

Figure 8: The true and learned stimulus filters constituting the parameters of the two-state attentive/ignoring neuron described in section 3.3. The conventions are the same as in Figure 4. The filter resulting from learning a standard GLM model is shown with the dotted line. (a) Spiking filter $\mathbf{k}_a$. (b) Spiking filter $\mathbf{k}_i$. (c) Transition filter $\mathbf{k}'_{ai}$. (d) Transition filter $\mathbf{k}'_{ia}$.

attentive/ignoring model for both the true parameters and a set of learned parameters can be compared; again, the means and ±1 σ deviations are almost identical between the two distributions, confirming that the two parameter sets (true and learned) produce identical behavior in the model neuron (data not shown). Analysis of the inferred posterior probability of the hidden state variable at each time step compared with the true state sequence further confirms the success of learning. The posterior probabilities resulting from the true parameters and a set of learned parameters are nearly identical, suggesting that the learning procedure was as successful as possible (data not shown). Over 2000 s of data, the correlation coefficient between the true state sequence and the inferred posterior probability was

0.91, while the percentage of time steps when the true state was predicted with a posterior probability greater than 0.5 was 95%.

4 Multistate Data in Rat Gustatory Cortex

Jones et al. (2007) have presentd an analysis of multielectrode data collected from gustatory cortex during the delivery of tastants—solutions of sucrose (sweet), sodium chloride (salty), citric acid (sour), and quinine (bitter)—to the tongues of awake, restrained rats. Each tastant was applied 7 to 10 times during each recording session, with water washouts of the mouth between trials. Across all recording sessions and all four tastants, the data consist of 424 trials, where each trial composes the 2.5 s of multielectrode spiking data immediately following the application of a tastant. Note that different sets of cells (varying in number from 6 to 10) were isolated during each recording session, and so only the trials corresponding to the same session and tastant pair can be considered to be samples from the same neural process.[6] In the initial analysis of the data, after directly inspecting the spike raster plots over multiple trials, it was realized that when multiple cells tended to change their firing rates during the course of a trial, they tended to do so simultaneously on a given trial but that this transition time often differed between trials. Thus, the choice was made to perform a simple HMM analysis to model these data. Specifically, a four-state model with constant state-dependent firing rates for each cell and constant transition rates between all pairs of states was fit to the data. Couching this previous model in our current notation, the stimulus filters $\mathbf{k}_n^c$ and $\mathbf{k}'_{nm}$ reduce to the background firing and transition rates b_n^c and b'_{nm}, respectively, with all history filters equal to $\mathbf{0}$. Note that these data conform to the multicell multitrial paradigm introduced in section 2.2.5.

4.1 Results from Spike History Dependent Models. While this data set does not have a known external time-varying stimulus and thus determining preferred stimuli for firing and transitioning is not possible, we provide a brief analysis extending the model presented in Jones et al. (2007) to include one aspect of the framework developed in section 2.2: the modeling of spike history effects. Note that in Chen et al. (2009), the authors also include spike history dependence in their model of UP and DOWN states during slow-wave sleep.

Unlike the case of simulated data, we do not know the true state-dependent firing rates, transition rates, or history filters, and so rather than

[6] In Jones et al. (2007) and in the analysis presented here, the trials from each session and tastant pair are assumed to be i.i.d. samples, which could be a false assumption due to, for example, changing motivational and arousal factors. However, a previous analysis investigated this issue and found that to a first approximation, the neural spiking behavior remains stationary over the course of a recording session (Fontanini & Katz, 2006).

comparing the learned model parameters to the true model parameters as is done in the previous section, we instead evaluate the cross-validated log likelihood of the data, an unbiased measure of the goodness of fit, over several model classes. The model class with the highest cross-validated log likelihood provides the best fit to the data. We compute the cross-validated log likelihood using leave-one-out cross-validation as follows. For every trial, we fit an HMM to the remaining trials of the same session and tastant pair and then evaluate the log likelihood of the left-out trial on the trained model. The sum of these log likelihoods for all 424 trials equals the total cross-validated log likelihood for the particular model class in question.

Jones et al. (2007) showed that using HMMs provides a more favorable fit to the data than using peristimulus time histograms (PSTHs), the traditional way of analyzing data such as these. It was argued that a possible explanation for this improved performance is that if the cortex does follow a series of computational steps after the application of the stimulus but does not complete each step in the same amount of time from trial to trial, then at any given poststimulus time, the state of the animal, and thus the firing rates of the recorded cells, may be significantly different on differing trials. By averaging over trials, as in the calculation of the PSTH, these differences are smeared into mean firing rates that may not be similar to the true rates of any single trial. A multistate model, on the other hand, since it allows different switching times from trial to trial, can preserve these differences and more accurately model the experimental data. Thus, in this article, we also fit PSTHs to these data to compare the cross-validated log likelihoods. We use the following Poisson-spiking PSTH model:

$$\lambda_t^c = f\left([\mathbf{k}^c]_t + \mathbf{h}^{c\mathrm{T}}\boldsymbol{\gamma}_t^c\right) \qquad t \in \{0, \ldots, T\}, \tag{4.1}$$

where $y_t^c \sim \mathrm{Poisson}(\lambda_t^c)$ as before. The length T filter $\mathbf{k}^c$ is fit by penalized maximum likelihood, exactly as discussed in section 2.3.1.[7]

Figure 9 shows the comparison of the cross-validated log likelihoods for the HMM and PSTH models with and without the inclusion of history effects. Since the number of spike trains in the data set varies across recording sessions and since the firing rates vary significantly across session and tastant pairs, we normalize the cross-validated log likelihoods by dividing by the number of cells and subtracting off the log likelihoods derived from a homogeneous Poisson model (i.e., to capture differences in firing rates).

[7] In fact, this Poisson-spiking PSTH model is exactly a one-state trial-triggered model whose parameters (which consists of the C vectors $\{\mathbf{k}^1, \ldots, \mathbf{k}^C\}$) can be estimated using the same algorithm developed for the trial-triggered model (see appendix D). Specifically, estimation is accomplished by a single M-step (since there are no latent variables in the model).

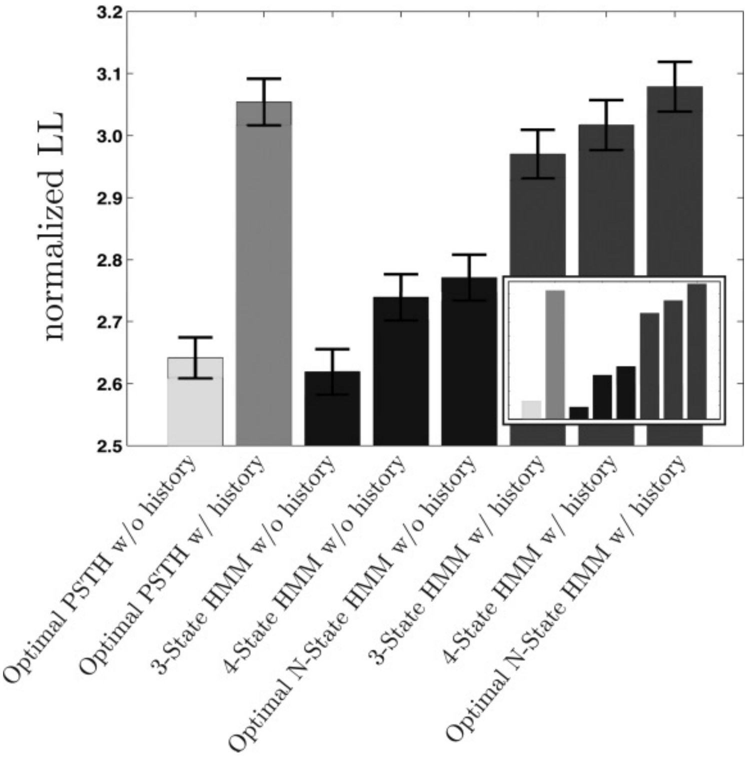

Figure 9: The performance of the several model classes described section 4.1, as measured by the normalized cross-validated log likelihoods of the proposed models. The normalization procedure is detailed in the text. While HMMs seem to outperform PSTHs in general (although not always, as in bar 6), the inclusion or exclusion of history effects, not the choice of HMM versus PSTH, seems to be the primary determinant of the value of the cross-validated log likelihood for the specific model class. Inset: The raw unnormalized results (the sum of the cross-validated log likelihoods for each of the 424 trials in the full data set). Note that the relative positions of the bars are preserved under the normalization procedure.

This allows a comparison of trials across session and tastant pairs and, thus, the calculation of meaningful error bars. Note that the relative heights of the normalized data in the figure are unchanged when compared to the raw data (see the figure inset).

The normalization procedure is given as follows. First, we define the normalized log likelihoods for each trial as

$$\mathrm{LL}^r_{\mathrm{norm}} \equiv \frac{1}{C_r}\left[L\left(\hat{\boldsymbol{\theta}}^r_{\mathrm{proposed}} \mid \mathbf{Y}^r\right) - L\left(\hat{\boldsymbol{\theta}}^r_{\mathrm{Poisson}} \mid \mathbf{Y}^r\right)\right], \tag{4.2}$$

where C_r is the number of cells composing the data of left-out trial r, $\mathbf{Y}^r$ consists of the spike trains of trial r, and $\hat{\boldsymbol{\theta}}^r$ is the maximum likelihood solution learned using the remaining trials from the same session and tastant pair as trial r. The Poisson models are simple fits of homogeneous firing rates to each cell: $\lambda^c = f(b^c)$. Then, if N_{cells} refers to the total number of spike trains across every combination of trial and tastant session (3872 spike trains in total) and if N_{trials} refers to the total number of trials over every session and tastant pair (424 trials), the sample means and sample standard errors shown in the figure are calculated from the set of values $\{\text{LL}^r_{\text{norm}}\}$, where each value in the set is included in the sample C_r times:

$$\langle \text{LL} \rangle = \frac{1}{N_{\text{cells}}} \sum_{r=1}^{N_{\text{trials}}} C_r \left(\text{LL}^r_{\text{norm}} \right), \tag{4.3}$$

and

$$\text{std. err.} = \frac{1}{\sqrt{N_{\text{cells}}}} \sqrt{\frac{1}{N_{\text{cells}} - 1} \sum_{r=1}^{N\text{trials}} C_r \left(\text{LL}^r_{\text{norm}} - \langle \text{LL} \rangle \right)^2}. \tag{4.4}$$

As in section 3.2, the history filters for each cell were parameterized by the coefficients of exponential basis functions with time constants of 2, 4, and 8 ms. For simplicity, no interneuronal cross-coupling spike history terms were included in this analysis. Since the true value of N, the number of hidden states, is unknown when using experimental data, we calculate the cross-validated log likelihoods for both $N = 3$ and $N = 4$ across the entire data set and then calculate the "optimal" cross-validated log-likelihood by summing the higher of the two values for each session and tastant pair. The choices of three and four states are empirically driven: for $N < 3$, the cross-validated log likelihood falls off significantly, while for $N > 4$, we found that the system is typically not inferred to spend at least some period of time in every state. Additionally, $N = 4$ was the choice that Jones et al. (2007) made.

The figure confirms the result in Jones et al. (2007) that HMMs fit the data better than PSTHs, but more strikingly draws attention to the importance of correctly modeling spike history effects. In terms of maximizing the cross-validated log likelihood, whether or not to include history dependence in the model far outstrips the choice of PSTH versus HMM. To understand why history dependence seems to outweigh the importance of using an HMM, it is helpful to consider the behavior of the log likelihood under a simple Bernoulli model with parameter p; in our context, of course, p would model the probability of a spike in a single time step. The Fisher information for this model is $\frac{1}{p(1-p)}$, which implies that the log likelihood is much more sensitive to perturbations around $p \approx 0$ than for larger values of p. The spike history filters serve mainly to capture intrinsic neuronal refractoriness and thus allow p to be correctly set to near zero values in time bins immediately following a spike. This evidently has a larger impact on the total log likelihood than does the choice of PSTH versus HMM (which can be thought to modulate p around less sensitive regions of parameter space when the true probability of spiking is far from $p = 0$). Thus, the magnitudes in the figure are somewhat artifactual. A history-ignoring model is clearly wrong, but not necessarily more wrong than the choice of PSTH over HMM.

Sample history filters as determined by both the history-dependent PSTH and HMM for one of the cells in the data set are shown in Figure 10a. The two model classes determine nearly identical history filters, with a large negative component at short post spike times accounting for neuronal refractoriness. Figures 10b–10d show instantaneous firing rates for PSTHs and HMMs with and without history dependence. Note that when history effects are modeled, the baseline firing rates of the cell are higher than in the history-ignoring models. This suggests that the history-ignoring models must reduce their baseline firing rates to account for the net reduction in firing rate introduced by neuronal refractoriness (thus keeping the area under the firing rate curves constant). However, by doing so, they fail to capture the correct background rate.

As we noted, Jones et al. (2007), argued that HMMs offer improved performances over PSTHs due to their ability to identify transition times. This avoids the need to average across trials and model the data with intermediate firing rates that may never be realized on any actual trial. Comparing the PSTH-determined firing rates in Figure 10b with the HMM-determined rates in Figures 10c and 10d reveals this phenomenon. In the PSTH models, the firing rates increase smoothly during the length of the trial, whereas they make discrete jumps when determined by HMMs.

4.2 Results from the Trial-Triggered Model. In a preliminary analysis of these data with the trial-triggered model presented in section 2.3.1, we found that the best setting of the smoothness penalty, as determined by the cross-validated log likelihood, is to require that the difference between adjacent firing rate values (e.g., $[\mathbf{k}_n]_t$ and $[\mathbf{k}_n]_{t+1}$) be essentially zero. This causes the trial-triggered model to degenerate to a time-homogenous HMM (albeit with spike history effects) or, in other words, almost exactly the model just presented.[8] While this degeneration is initially surprising, it does agree with

[8] Although cross-validation chooses a smoothness penalty for the spiking filters that yields time homogeneous spiking, the optimal degree of smoothness for the transitioning filters does result in time-inhomogeneous transitioning. This result indicates that the true state dwell times are not exponentially distributed (i.e., that there is some reliability in the transitioning behavior of the system from trial to trial). Thus, the trial-triggered model marginally outperforms the model presented in section 4.1 by capturing this inhomogeneity. Further analysis with the semi-Markov transition-triggered model described in section 2.3.2 will appear elsewhere (Escola, 2009).

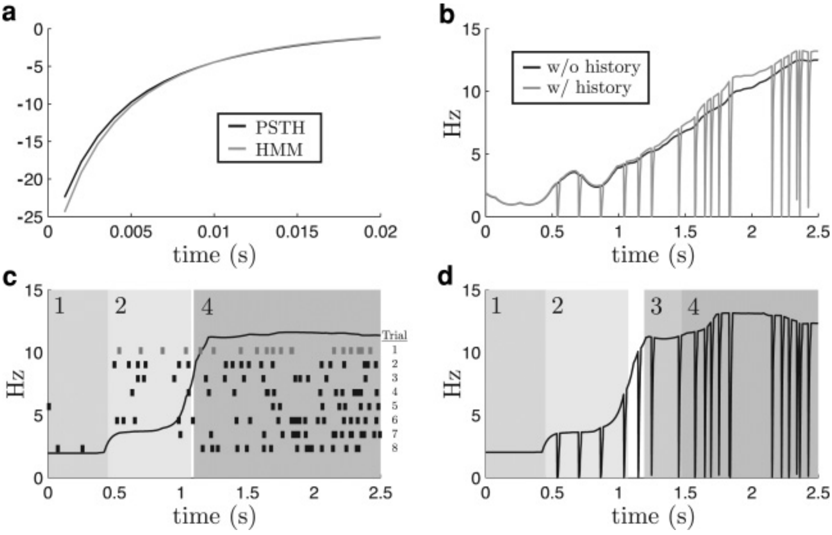

Figure 10: (a) The fitted history filters for a sample cell from a single session and tastant pair. The filters capture the neuron's intrinsic refractoriness and do not vary significantly by model class. (b) The instantaneous firing rates λ_t from a single trial for the same sample cell shown in *a* as determined by the with- and without-history PSTH models (see equation 4.1). The dips in the lighter trace reflect the refractoriness of the neuron following spikes due to the fitted history filter (the darker filter in *a*). (c) The instantaneous firing rate and predicted hidden state sequence for the same cell from the same trial shown in *b* on a four-state HMM without history dependence. The black trace is determined by weighting the state-specific firing rates by the posterior distribution of the state sequence: $\langle \lambda_t \rangle = \sum_n \hat{p}(q_t = n)\lambda_{n,t}$. The solid backgrounds indicate time periods when the posterior probability of one of the states exceeds 0.5. Note that for this particular trial from this session and tastant pair, the system is never predicted to be in state 3. (d) As in panel *c*, but for a history-dependent HMM. The neuron's intrinsic refractoriness (determined by the lighter filter in *a*) is clearly captured. Note that the firing rates and state sequences shown in *b, c,* and *d* are for a left-out trial evaluated on models trained on the remaining trials so as to avoid bias. The raster plot in *c* shows the neuron's firing pattern across the eight trials of the session and tastant pair used to create this figure. The lighter-colored raster corresponds to the trial from which the instantaneous firing rates and state sequences in *b, c,* and *d* are calculated. These spike times correspond exactly to the dips in the firing rates shown for the history-dependent models. Although the firing rate trajectories between the HMM and PSTH models are notably different, the accurate modeling of neuronal refractoriness (by means of spike history filters) dominates the goodness of fit of the models (see Figure 9).

the interpretation of Jones et al. (2007) as to why a multistate model more accurately reflects the data from trial to trial than does a standard PSTH model. Recall the argument that if a cortical computation (in this case, tastant recognition) requires a neural network to progress through a sequence of states with homogeneous state-dependent firing rates, and if the time needed to complete each step of the sequence varies from trial to trial, then a PSTH will smear out these state-dependent rates and result in average firing rates that do not reflect the true rates observed on actual trials. Now imagine if the true structure of the data is that each state *n* is defined not by a set of homogeneous firing rates, but, rather, by a set of time-varying firing rate trajectories that the cells of the network follow each time the system visits state *n*. Then if the time at which the system transitions to state *n* varies from trial to trial, the trial-triggered model will smear out these state-dependent trajectories (in the same way that a standard PSTH model will smear out state-dependent homogeneous firing rates) due to the fact that the parameters of the trial-triggered model, the state-dependent PSTHs, are locked to the onset time of the trial, not the onset time of the current state. Since the trial-triggered framework is unable to account for firing rate trajectories that evolve from the state onset times, this may explain why the cross-validated log likelihood is maximal for the setting of the smoothness penalty that forces the model to have unchanging firing rates.

5 Discussion

5.1 Previous HMMs Applied to Spike Trains. Our model can be considered to be a direct extension of previous applications of HMMs to spike-train data discussed in Abeles et al. (1995), Seidemann et al. (1996), and Gat et al. (1997). In these earlier models, the goal was to predict which task a monkey was performing during the recording of a spike train by comparing the likelihoods of the spike train tested on two different HMMs—one trained on only trials corresponding to task 1 and the other trained on trials corresponding to task 2. These are similar to the model recently described in Jones et al. (2007) and discussed in detail in section 4 for predicting tastants from recordings of neurons in gustatory cortex. The major difference between these models and ours, and thus the reason that ours may be considered an extension, is the lack of an external time-varying stimulus and spike history dependence. Thus, both the transition rates and the firing rates in their models were time homogeneous and the size of the parameter set significantly smaller (because rates are defined by single parameters rather than stimulus and history filters). In the degenerate case where no stimulus is present and history effects are ignored, the state-dependent firing and transition rates are constant, and our model becomes identical to these previous models.

Although we specifically consider stimulus-driven sensory neurons in this article, the model can also represent the relationship between spike trains and time-varying behavioral variables (e.g., hand position in a motor task). With no change to the model as presented in section 2, we could infer the hidden state sequence and optimal "motor filters" from paired spike train and behavior data (Kemere et al., 2008). A related approach is developed in Wu et al. (2004) where they assume that hand position data (location, velocity, and acceleration in two dimensions) evolve according to an autoregressive model. The graphical representation of their model is essentially identical to Figure 2b, except that they have arrows between adjacent stimuli (or, in their case, position vectors) to reflect the fact that one stimulus affects the next.[9] Thus, while the focus of our work can been seen to be the identification of the filters of the model and the recapitulation of the hidden state sequence, these goals are closely related to models that seek to identify either the behavior or stimulus (the two are mathematically identical) encoded by spike trains.

5.2 Alternative Current Techniques. There has been a good deal of recent work representing the stimulus-response relationship of neurons with linear state-space models, which, since they also employ latent variables that alter the stimulus-dependent spiking behavior, are similar in spirit to our model. In Smith and Brown (2003), for example, there is a one-dimensional state variable given by $q_t = \rho q_{t-1} + \mathbf{k}'^{\mathrm{T}}\mathbf{s}_t + \beta\varepsilon(t)$, where ρ is the correlation between adjacent time steps and $\beta\varepsilon(t)$ represents gaussian noise with variance β^2. We have otherwise modified their notation to match our own. Given the state variable q_t, the firing rate is $\lambda_t = f(q_t + b)$ with bias b. This model is similar to ours in that the state dynamics are stimulus dependent (i.e., through the filter $\mathbf{k}'$), but, conditioned on the state employ homogeneous firing rates.

Similarly, in Frank et al. (2002), Eden et al. (2004), Czanner et al. (2008), and Kulkarni and Paninski (2007), the state-variable vectors evolve according to homogeneous gaussian dynamics, but the state-conditioned firing rates are stimulus dependent. For example, the rates can be given by $\lambda_t = f(\mathbf{k}_t{}^{\mathrm{T}}\mathbf{s}_t)$ where the stimulus filter $\mathbf{k}_t$ itself is the state variable, or by $\lambda_t = f(\mathbf{k}^{\mathrm{T}}\mathbf{s}_t + \mathbf{g}^{\mathrm{T}}\mathbf{q}_t)$, where $\mathbf{g}^{\mathrm{T}}\mathbf{q}_t$ is meant to represent some unmeasured input current (i.e. not stimulus derived). See also Wu, Kulkarni, Hatsopoulos, and Paninski (2009) for a related approach and Paninski et al. (2009) for further examples.

The continuous state-space formulation has a number of potential advantages and disadvantages compared to the discrete HMM approach. On the one hand, the state dynamics in the discrete setting are nonlinear, and

[9] They also remove the arrows between the stimuli and the hidden state variables, meaning that they have homogeneous transition rates.

the state dynamics and the spiking behavior are driven by the stimulus in a flexible, nonlinear manner. On the other hand, the number of parameters in the discrete HMM approach depends quadratically on the number of states (although techniques exist to limit the dimensionality of the model, as discussed in appendix A), and the transitioning is assumed to occur on a fast timescale (within a single time step of length dt), while in many cases, the true transitioning behavior may be slower.

An interesting continuous state-space model that employs nonlinear dynamics and may serve as a middle ground between the linear continuous models and HMMs is given in Yu et al. (2006). The authors propose a model with a stochastic recurrent nonlinear neural network as the hidden variable. Such networks, as is known from the classic network literature, may have multiple stable attractor states, and these could correspond to the discrete states of an HMM. In addition, recent work in the network literature indicates that it may be possible to design networks that implement a desired Markov chain (Jin, 2009).

While we expect even richer models to be developed as computational power increases and experimental evidence for multistate neurons grows, we believe that the model presented in this article nicely complements these other available models and offers an alternative for the analysis of the stimulus-reponse relationship. Ultimately, however, the appropriate model for a particular data set is dictated by the data themselves and can be determined by evaluating the different approaches mentioned in this section under the rules of model selection. By contributing to the library of point-process models, we hope to provide an additional tool that may be the most appropriate for certain data sets.

5.3 Additional Applications of the HMM Framework.

Another interesting application for which the framework developed in the article may be appropriate involves the modeling of neural systems that have been shown to have stimulus-specific adaptation (Blake & Merzenich, 2002; Borst, Flanagin, & Sompolinsky, 2005; Maravall, Petersen, Fairhall, Arabzadeh, & Diamond, 2007), in which optimal linear filters of neural encoding models change with certain stimulus attributes. Such stimulus dependencies have previously been discussed in theoretical contexts in, for example, Ahrens, Linden, and Sahani (2008) and Hong, Lundstrom, and Fairhall (2008), but the stimulus-dependent HMM might provide an alternative model for similar phenomena. In such a model, the stimulus-response relationship in each state could be thought to define, in a piecewise fashion, a portion of the overall relationship, and the state dynamics would thus reflect the dynamics of adaptation.

5.4 Hidden Semi-Markov Models and the Transition-Triggered Model.

The transition-triggered model introduced in section 2.3.2 falls into the category of hidden semi-Markov models (see Sansom & Thomson, 2001, for an introduction to HSMMs). These models replace exponentially distributed state dwell times with dwell times drawn from other distributions with support on the nonnegative real axis (e.g., the gamma distribution). Thus, the models are no longer purely Markovian. That is, knowing the current state of the system at time t does not provide the maximum possible information about future states of the system; the elapsed time in the current state is also needed, which is analogous to keeping track of both t and τ (where τ, as introduced in section 2.3.2, is the current depth in the state-space cascade of Figure 3b). Although two previous papers have presented the use of HSMMs for the analysis of spike train data (Chen et al., 2009; Tokdar et al., 2009), the transition-triggered model differs significantly from these in two major respects. First, by introducing the use of an expanded state-space with NT instead of N states, we are able to adapt the Baum-Welch algorithm to do exact inference of the hidden state distribution (as described in appendix D), while previous approaches employed Markov chain Monte Carlo (MCMC) techniques for inference. Both methods have a time complexity of $\mathcal{O}(T^2)$ per iteration. Second, and more important, the transition-triggered model can capture state-dependent firing-rate dynamics via the peritransition time histograms (PTTHs), which parameterize the model. These PTTHs allow the spiking behavior of the model to be dependent on the time since the most recent transition in addition to stimulus and spike history–driven effects. Previous authors have focused on accounting for nonexponentially distributed state dwell times, but have used traditional spiking models in each state (which, for Chen et al., 2009, does include the modeling of history effects, but, is otherwise determined only by the state-dependent background firing rates). Again, further work incorporating the strengths of each approach should be fruitful.

Appendix A: Reduced Parameter Models

If the dimensionality of the stimulus (including the augmentation for the bias term and any history dependence) is D, then the total number of parameters corresponding to the transition and spiking filters of our model is DN^2 (with an additional $N-1$ for $\boldsymbol{\pi}$). Since this number grows quadratically with N, it will become unfeasibly expensive, with respect to both computational time and data demands, to fit the parameters of a model with even a modest number of states and a modest stimulus dimensionality. Thus, it is reasonable to consider techniques for reducing the number of parameters and thereby control for this quadratic dependence on N.

The obvious target for such a reduction is in the parameters that define the transition matrix $\boldsymbol{\alpha}_t$, since these are the ones that grow quadratically. The total collection of transition parameters ($\mathbf{k}'_{nm} : \forall n, \forall m \neq n$) can be thought to constitute a rank 3 tensor $\mathbf{K}'$ of dimensionality $D \times N \times N-1$. Thus $\mathbf{K}'_{dnm}$

corresponds to the dth element of the transition filter $\mathbf{k}'_{nm}$. Ahrens, Paninski, and Sahani (2008) developed a formalism for the approximation of a full-rank tensor such as $\mathbf{K}'$ by the sum of lower-rank tensors. In general, there are many ways to decompose a full-rank tensor into low-rank components, which can result in reduced parameter sets as small as $D + 2N - 1$ for the most restricted case (i.e., if each dimension of $\mathbf{K}'$ is considered to vary independent of the others).

One example reduction that has a neurobiological interpretation operates on each row of the transition matrix independently. Consider $\mathbf{K}'_n \equiv (\mathbf{k}'_{n1} \cdots \mathbf{k}'_{nN})$, the $D \times N - 1$ matrix composed of all filters corresponding to transitions away from state n. This matrix can be approximated as

$$\mathbf{K}'_n \approx \sum_{i=1}^{R} \mathbf{k}'_{ni} \mathbf{w}_{ni}^T, \tag{A.1}$$

where R is the desired rank of the approximation to $\mathbf{K}'_n$, which is at most the lesser of D and $N - 1$. Each individual transition filter $\mathbf{k}'_{nm}$ is thus a linear combination of the R filters $\mathbf{k}'_{ni}$ with the mixing coefficients given by the weight vectors $\mathbf{w}_{ni}$. The directions in stimulus space given by the $\mathbf{k}'_{ni}$ filters can be thought of as the set of stimulus triggers (or single trigger, in the case that $R = 1$) for transitioning away from state n, while the $\mathbf{w}_{ni}$ weight vectors dictate which transition is most likely given the trigger.[10] While this reduced model certainly has more restricted state dynamics, it may reasonably capture the behavior of some multistate neural systems. The total number of parameters in this reduced model is $NR(D + N - 1)$.

Another example is essentially the converse of the last. In this case, we consider $\mathbf{K}'_m \equiv (\mathbf{k}'_{1m}, \ldots, \mathbf{k}'_{Nm})$, the matrix consisting of all filters corresponding to transitions into state m. If the parameter reduction is performed as before, the interpretation is that one or more triggers $\mathbf{k}'_{mi}$ (the number of which depends on R) drive the system into state m, and the responsiveness to a given trigger when the system is in state n depends on the weight vectors $\mathbf{w}_{mi}$.

A caveat when using these techniques is that the ECLL maximized during each M-step is no longer concave in the model parameters as described in section 2.2.6. However, the ECLL is concave in the $\mathbf{k}'$ filters while holding the $\mathbf{w}$ weights constant and concave in the $\mathbf{w}$ weights while holding the $\mathbf{k}'$ filters constant. Thus, the M-step may be maximized by alternatively maximizing the ECLL with respect to each parameter subset separately until the procedure converges. Unfortunately, there is no way to guarantee that the globally optimal solution for each M-step is found when using these

[10] Since the weights $\mathbf{w}_{ni}$ can be either positive or negative, each $\mathbf{k}'_{ni}$ filter actually corresponds to two potential stimulus triggers: $\mathbf{k}'_{ni}$ and $-\mathbf{k}'_{ni}$.

low-rank models (i.e., local optima may exist). However, empirical results in Ahrens, Paninski, et al. (2008) suggest that this is not a serious problem.

Besides low-rank models, other parameter-reduction techniques are possible. The simplest is to assume that some of the transitions are stimulus independent, for which one might have some a priori biological evidence. In this case, these filters can be removed from the model and the corresponding transitions fit with homogeneous rates. Ultimately, one may even wish to eliminate certain transitions altogether and define those rates to be zero.

A more robust and general approach with many desirable properties is to add a prior distribution over the parameters that can control overfitting of the data even in the case of very high-dimensional parameter spaces. Essentially such priors reduce the number of parameters in the model from the allowable maximum to the effective subset needed to represent the data appropriately. Parameters that are not in this effective subset have their values dictated by their prior distributions. In this case, the learned maximum a posteriori parameter setting is that which maximizes the product of the prior and the likelihood rather than the likelihood alone. It is convenient to choose priors that do not affect the concavity of the ECLL (i.e., those that are log concave with respect to the parameters).

Appendix B: Gradient Ascent of the Expected Complete Log Likelihood

The solution for the M-step for each iteration of EM—the parameter setting that maximizes the ECLL—is not analytically tractable. However, given the concavity constraints for the nonlinearities g and f, we know that a unique solution exists, and thus that it may be found by ascending the gradient of this likelihood. Although numerical gradient techniques are possible, maximization is much more rapid if the gradient and the Hessian (second-derivative matrix) have analytic solutions, which permits the application of Newton-Raphson optimization.

Since the ECLL for our model decomposes into terms that depend on the transition matrix $\boldsymbol{\alpha}_t$ and those that depend on the emission matrix $\boldsymbol{\eta}_t$, we can optimize the two parameter sets independently. For the transition-dependent terms, the ECLL decomposes further into terms corresponding to all possible transitions that originate from the same state n (i.e., the parameters of each row of $\boldsymbol{\alpha}_t$ can be optimized independently). Thus, we have

$$\begin{aligned}&\left\langle L(\mathbf{k}'_{nm} \mid \mathbf{y}, \mathbf{q}, \mathbf{S})\right\rangle_{\hat{p}(\mathbf{q})}\\ &\quad\sim \sum_{t=1}^{T} \left(\begin{array}{l} \displaystyle\sum_{m \neq n} \hat{p}(q_{t-1} = n, q_t = m) \log g\left(\mathbf{k}'_{nm}{}^{\mathrm{T}} \mathbf{s}_t\right) \\ -\, \hat{p}(q_{t-1} = n) \log\left(1 + \sum_{l \neq n} g\left(\mathbf{k}'_{nl}{}^{\mathrm{T}} \mathbf{s}_t\right) dt\right) \end{array} \right),\end{aligned} \tag{B.1}$$

and the following gradient:

$$\vec{\nabla}\big(\mathbf{k}'_{nm}\big) = \sum_{t=1}^{T} g'\big(\mathbf{k}'_{nm}{}^{\mathrm{T}}\mathbf{s}_t\big) \times \left(\frac{\hat{p}(q_{t-1}=n, q_t=m)}{g\big(\mathbf{k}'_{nm}{}^{\mathrm{T}}\mathbf{s}_t\big)} - \frac{\hat{p}(q_{t-1}=n)\,dt}{1+\sum_{l\neq n} g\big(\mathbf{k}'_{nl}{}^{\mathrm{T}}\mathbf{s}_t\big)\,dt} \right)\mathbf{s}_t. \tag{B.2}$$

The Hessian can be further broken down into two types of matrices depending on whether the second derivative is taken with respect to the same filter as that from which the first derivative was calculated ($\mathbf{k}'_{nm}$) or with respect to another filter corresponding to a different transition in the same row of $\boldsymbol{\alpha}_t$ (e.g., $\mathbf{k}'_{no}$). For the former case, we have

$$H(\mathbf{k}'_{nm}, \mathbf{k}'_{nm}) = \sum_{t=1}^{T} \left(\begin{array}{l} \dfrac{\hat{p}(q_{t-1}=n, q_t=m)\Big[g\big(\mathbf{k}'_{nm}{}^{\mathrm{T}}\mathbf{s}_t\big)g''\big(\mathbf{k}'_{nm}{}^{\mathrm{T}}\mathbf{s}_t\big) - g'\big(\mathbf{k}'_{nm}{}^{\mathrm{T}}\mathbf{s}_t\big)^2\Big]}{g\big(\mathbf{k}'_{nm}{}^{\mathrm{T}}\mathbf{s}_t\big)^2} \\ - \dfrac{\hat{p}(q_{t-1}=n)\,dt\Big[\big(1+\sum_{l\neq n} g\big(\mathbf{k}'_{nl}{}^{\mathrm{T}}\mathbf{s}_t\big)\,dt\big)g''\big(\mathbf{k}'_{nm}{}^{\mathrm{T}}\mathbf{s}_t\big) - g'\big(\mathbf{k}'_{nm}{}^{\mathrm{T}}\mathbf{s}_t\big)^2\,dt\Big]}{\big(1+\sum_{l\neq n} g\big(\mathbf{k}'_{nl}{}^{\mathrm{T}}\mathbf{s}_t\big)\,dt\big)^2} \end{array} \right)\mathbf{s}_t\mathbf{s}_t{}^{\mathrm{T}}, \tag{B.3}$$

and for the latter case,

$$H(\mathbf{k}'_{nm}, \mathbf{k}'_{no}) = \sum_{t=1}^{T} \frac{\hat{p}(q_{t-1}=n)g'\big(\mathbf{k}'_{nm}{}^{\mathrm{T}}\mathbf{s}_t\big)g'\big(\mathbf{k}'_{no}{}^{\mathrm{T}}\mathbf{s}_t\big)\,dt^2}{\big(1+\sum_{l\neq n} g\big(\mathbf{k}'_{nl}{}^{\mathrm{T}}\mathbf{s}_t\big)\,dt\big)^2}\mathbf{s}_t\mathbf{s}_t{}^{\mathrm{T}}. \tag{B.4}$$

Since the concavity constraints on g require that it be the exponential function (see section 2.2.6), the complexity of equations B.2, B.3, and B.4 is considerably reduced. For example, g and g' cancel in the gradient, and the first term of the $H(\mathbf{k}'_{nm}, \mathbf{k}'_{nm})$ is equal to zero.

For the spiking-dependent terms in the ECLL, the parameters for spiking in each state are again independent and can be optimized separately. Thus, we have

$$\big\langle L(\mathbf{k}_n \mid \mathbf{y}, \mathbf{q}, \mathbf{S})\big\rangle_{\hat{p}(\mathbf{q})} \sim \sum_{t=0}^{T} \hat{p}(q_t=n)(y_t \log f(\mathbf{k}_n{}^{\mathrm{T}}\mathbf{s}_t) - f(\mathbf{k}_n{}^{\mathrm{T}}\mathbf{s}_t)\,dt), \tag{B.5}$$

which yields the following gradient

$$\vec{\nabla}(\mathbf{k}_n) = \sum_{t=0}^{T} \hat{p}(q_t = n)\left(y_t \frac{f'\left(\mathbf{k}_n{}^{\mathrm{T}}\mathbf{s}_t\right)}{f\left(\mathbf{k}_n{}^{\mathrm{T}}\mathbf{s}_t\right)} - f'\left(\mathbf{k}_n{}^{\mathrm{T}}\mathbf{s}_t\right) dt \right)\mathbf{s}_t \tag{B.6}$$

and Hessian

$$H(\mathbf{k}_n, \mathbf{k}_n) = \sum_{t=0}^{T} \hat{p}(q_t = n) \times \left(y_t \frac{f\left(\mathbf{k}_n{}^{\mathrm{T}}\mathbf{s}_t\right) f''\left(\mathbf{k}_n{}^{\mathrm{T}}\mathbf{s}_t\right) - f'\left(\mathbf{k}_n{}^{\mathrm{T}}\mathbf{s}_t\right)^2}{f\left(\mathbf{k}_n{}^{\mathrm{T}}\mathbf{s}_t\right)^2} - f''\left(\mathbf{k}_n{}^{\mathrm{T}}\mathbf{s}_t\right) dt \right)\mathbf{s}_t\mathbf{s}_t{}^{\mathrm{T}}, \tag{B.7}$$

where, again, depending on the choice of f, these formulas may simplify considerably.

Thus, since analytic formulations for the gradient and Hessian can be found for all parameters, Newton-Raphson optimization can be used to solve the M-step.

Appendix C: Continuous-Time HMMs

In order to adapt the Baum-Welch algorithm to continuous time, we first find the instantaneous values of the transition and emission matrices $\boldsymbol{\alpha}_t$ and $\boldsymbol{\eta}_t$ as $dt \to 0$, where t is now a real number rather than the time-step index. For the off-diagonal terms of $\boldsymbol{\alpha}_t$, we have

$$\lim_{dt \to 0} \alpha_{nm,t} = \lim_{dt \to 0} \frac{\lambda'_{nm,t}\, dt}{1 + \sum_{l \neq n} \lambda'_{nl,t} dt} = \lambda'_{nm,t} dt \qquad m \neq n, \tag{C.1}$$

where we use the notation $\lim_{x \to 0} f(x) = g(x)$ to indicate that $f(x) = g(x) + o(x)$ for values of x near zero. To take the limit of the diagonal terms of $\boldsymbol{\alpha}_t$, we use the fact that the Taylor expansion of $f(x) = \frac{1}{1+x}$ for small values of x yields $f(x) \approx 1 - x$:

$$\lim_{dt \to 0} \alpha_{nn,t} = \lim_{dt \to 0} \frac{1}{1 + \sum_{l \neq n} \lambda'_{nl,t} dt} = 1 - \sum_{l \neq n} \lambda'_{nl,t} dt. \tag{C.2}$$

The resulting probability distribution is consistent as expected (i.e., $\sum_m \alpha_{nm,t} = 1$). If we define a rate matrix $\mathbf{R}$ as

$$R_{nm,t} = \begin{cases} \lambda'_{nm,t} & m \neq n \\ -\sum_{l \neq n} \lambda'_{nl,t} & m = n \end{cases}, \tag{C.3}$$

then $\boldsymbol{\alpha}_t$ can be written as

$$\boldsymbol{\alpha}_t = \mathbf{I} + \mathbf{R}_t \, dt, \tag{C.4}$$

where $\mathbf{I}$ is the identity matrix.

In the limit of small dt, there will never be more than one spike per time step, and so the $\boldsymbol{\eta}_t$ matrix reduces from the description in equation 2.20 to a simple two-column matrix (i.e., the Poisson distribution becomes a binary distribution) as follows:

$$\lim_{dt \to 0} \eta_{ni,t} = \lim_{dt \to 0} \frac{(\lambda_{n,t} dt)^i \, e^{-\lambda_{n,t} dt}}{i!} \qquad i \in \{0, 1, 2, \ldots\} \tag{C.5}$$

becomes

$$\lim_{dt \to 0} \eta_{n0,t} = \lim_{dt \to 0} e^{-\lambda_{n,t} dt} = 1 - \lambda_{n,t} dt \tag{C.6}$$

and

$$\lim_{dt \to 0} \eta_{n1,t} = \lim_{dt \to 0} (\lambda_{n,t} dt) \, e^{-\lambda_{n,t} dt} = \lambda_{n,t} dt, \tag{C.7}$$

where we use the linear terms of the Taylor expansion to make the approximation that $e^{-x} \approx 1 - x$ for values of x near 0. Consistency is again assured (i.e., $\eta_{n0,t} + \eta_{n1,t} = 1$).

C.1 The E-Step. Next, we extend the forward-backward algorithm to continuous time. From equation 2.9 and assuming that y_t is the "no-spike" emission, we have

$$\begin{aligned} a_{n,t} &= \eta_{n0,t} \left(\sum_{m=1}^{N} \alpha_{mn,t} a_{m,t-dt} \right) \\ &= (1 - \lambda_{n,t} dt) \left(\sum_{m=1}^{N} \alpha_{mn,t} a_{m,t-dt} \right), \end{aligned} \tag{C.8}$$

which can be written in matrix form as

$$\begin{aligned}\mathbf{a}_t &= \left(\mathbf{I} - \mathrm{diag}(\boldsymbol{\lambda}_t)\, dt\right) \boldsymbol{\alpha}_t{}^{\mathrm{T}} \mathbf{a}_{t-dt}\\ &= \left(\mathbf{I} - \mathrm{diag}(\boldsymbol{\lambda}_t)\, dt\right) (\mathbf{I} + \mathbf{R}_t dt)^{\mathrm{T}}\, \mathbf{a}_{t-dt}\\ &= \mathbf{a}_{t-dt} + \left(\mathbf{R}_t - \mathrm{diag}(\boldsymbol{\lambda}_t)\right)^{\mathrm{T}} \mathbf{a}_{t-dt} dt + o\left(dt^2\right).\end{aligned} \tag{C.9}$$

For small dt, this yields the linear differential equation

$$\dot{\mathbf{a}}_t = \left(\mathbf{R}_t - \mathrm{diag}(\boldsymbol{\lambda}_t)\right)^{\mathrm{T}} \mathbf{a}_t. \tag{C.10}$$

This equation holds from spike to spike (i.e., for all the dt's when the emission is in fact "no spike"). If t_{i-1} and t_i are consecutive spike times, a numerical ODE solver can be used to determine $\mathbf{a}_{t_i}$ given $\mathbf{a}_{t_{i-1}}$ by using equation C.10. Determining the update at the spike times is similar:

$$\begin{aligned}\mathbf{a}_{t_i} &= \left(\mathrm{diag}(\boldsymbol{\lambda}_{t_i})\, dt\right) \boldsymbol{\alpha}_{t_i}{}^{\mathrm{T}} \mathbf{a}_{t_i-dt}\\ &= \left(\mathrm{diag}(\boldsymbol{\lambda}_{t_i})\, dt\right) (\mathbf{I} + \mathbf{R}_t dt)^{\mathrm{T}}\, \mathbf{a}_{t_i-dt}\\ &= \mathrm{diag}(\boldsymbol{\lambda}_{t_i}) \mathbf{a}_{t_i-dt} dt + o\left(dt^2\right).\end{aligned} \tag{C.11}$$

By taking the limit as $dt \to 0$ and dropping the dt multiplier,[11] we get the spike time update as

$$\mathbf{a}_{t_{i+}} = \mathrm{diag}(\boldsymbol{\lambda}_{t_i}) \mathbf{a}_{t_{i-}}, \tag{C.12}$$

where $\mathbf{a}_{t_{i-}}$ is the forward probability vector before the spike (i.e., the result of the integration from t_{i-1} to t_i using equation C.10) and $\mathbf{a}_{t_{i+}}$ is the vector after the spike. Finally, $\mathbf{a}_0$ is initialized as $\boldsymbol{\pi}$.

The backward probabilities are adapted to continuous time in a similar manner. From equation 2.13 and assuming the no-spike emission,

$$\begin{aligned}b_{n,t-dt} &= \sum_{m=1}^{N} \alpha_{nm,t} \eta_{m0,t} b_{m,t}\\ &= \sum_{m=1}^{N} \alpha_{nm,t} (1 - \lambda_{n,t} dt) b_{m,t},\end{aligned} \tag{C.13}$$

[11] In a continuous-time model, the probability of any given spike train with specific spike times is zero. Thus, we discard these zero multipliers since they are independent of the model parameters and merely add a constant (albeit infinite) component to the log likelihood.

which, in matrix form, becomes

$$\begin{aligned}\mathbf{b}_{t-dt} &= \boldsymbol{\alpha}_t \left(\mathbf{I} - \mathrm{diag}(\boldsymbol{\lambda}_t)\, dt\right) \mathbf{b}_t \\ &= (\mathbf{I} + \mathbf{R}_t dt) \left(\mathbf{I} - \mathrm{diag}(\boldsymbol{\lambda}_t)\, dt\right) \mathbf{b}_t \\ &= \mathbf{b}_t - \left(\mathrm{diag}(\boldsymbol{\lambda}_t) - \mathbf{R}_t\right) \mathbf{b}_t dt + o\left(dt^2\right),\end{aligned} \tag{C.14}$$

yielding the differential equation

$$\dot{\mathbf{b}}_t = \left(\mathrm{diag}(\boldsymbol{\lambda}_t) - \mathbf{R}_t\right) \mathbf{b}_t. \tag{C.15}$$

The spike time update follows exactly as before:

$$\mathbf{b}_{t_i -} = \mathrm{diag}(\boldsymbol{\lambda}_{t_i}) \mathbf{b}_{t_i +}. \tag{C.16}$$

The initialization of the backward probabilities remains unchanged from equation 2.12: $\mathbf{b}_T = \mathbf{1}$.

As in the discrete-time case, the log likelihood is calculated as

$$L(\boldsymbol{\theta} \mid \mathbf{y}, \mathbf{S}) \equiv \log p(\mathbf{y} \mid \mathbf{S}, \boldsymbol{\theta}) = \log \sum_{n=1}^{N} a_{n,T}, \tag{C.17}$$

and the individual marginal distributions of $\hat{p}(\mathbf{q})$ are given by equation 2.14:

$$\hat{p}(q_t = n) = \frac{a_{n,t} b_{n,t}}{p(\mathbf{y} \mid \mathbf{S}, \boldsymbol{\theta})}. \tag{C.18}$$

Note that although the forward and backward probabilities are discontinuous at the spike times, the marginal probabilities are continuous at all times t. It is clear from equation C.18 that the marginal probabilities are continuous between spike times (since the forward and backward probabilities are), while at the spike times, we have

$$\hat{p}(q_{t_i -} = n) = \frac{a_{n,t_i -} b_{n,t_i -}}{p(\mathbf{y} \mid \mathbf{S}, \boldsymbol{\theta})} = \frac{a_{n,t_i -} \lambda_{n,t_i} b_{n,t_i +}}{p(\mathbf{y} \mid \mathbf{S}, \boldsymbol{\theta})} = \frac{a_{n,t_i +} b_{n,t_i +}}{p(\mathbf{y} \mid \mathbf{S}, \boldsymbol{\theta})} = \hat{p}(q_{t_i +} = n). \tag{C.19}$$

Rather than using the consecutive-pairwise marginals as in the discrete-time case, in the continuous-time framework, expected instantaneous transition rates $r_{n\to m,t}$ given $\mathbf{y}$, $\mathbf{S}$, and $\boldsymbol{\theta}$ are needed:

$$\begin{aligned}
r_{n\to m,t} &= \lim_{dt\to 0} \frac{p(q_t = m \mid q_{t-dt} = n, \mathbf{y}, \mathbf{S}, \boldsymbol{\theta})}{dt} \\
&= \lim_{dt\to 0} \frac{p(q_t = m, q_{t-dt} = n \mid \mathbf{y}, \mathbf{S}, \boldsymbol{\theta})}{p(q_{t-dt} = n \mid \mathbf{y}, \mathbf{S}, \boldsymbol{\theta})\, dt} \\
&= \lim_{dt\to 0} \frac{\frac{\cancel{a_{n,t-dt}}\alpha_{nm,t}\eta_{my_t,t}b_{m,t}}{\cancel{p(\mathbf{y}|\mathbf{S},\boldsymbol{\theta})}}}{\frac{\cancel{a_{n,t-dt}}b_{n,t-dt}}{\cancel{p(\mathbf{y}|\mathbf{S},\boldsymbol{\theta})}}dt} \\
&= \lim_{dt\to 0} \frac{\alpha_{nm,t}\eta_{my_t,t}b_{m,t}}{b_{n,t-dt}dt},
\end{aligned} \tag{C.20}$$

where we substitute the single and pairwise marginals with equations 2.14 and 2.15. Notice that the forward probability $a_{n,t-dt}$ cancels out of equation C.20. This reflects the Markovian nature of the model. The expected rate $r_{n\to m,t}$ is the transition rate to m assuming that the current state is n. The Markov assumption states that given the present, the past and future are independent. In other words, assuming some assignment of the current state, the forward probabilities (which reflect the past) do not affect the expected transition rates (which reflect predictions about the future). At the nonspike times, equation C.20 becomes

$$\begin{aligned}
r_{n\to m,t} &= \lim_{dt\to 0} \frac{\lambda'_{nm,t}\cancel{dt}(1 - \cancel{\lambda_{m,t}dt})b_{m,t}}{b_{n,t-dt}\cancel{dt}} \\
&= \lambda'_{nm,t} \cdot \frac{b_{m,t}}{b_{n,t}},
\end{aligned} \tag{C.21}$$

and at spike times,

$$\begin{aligned}
r_{n\to m,t_i-} &= \lim_{dt\to 0} \frac{\lambda'_{nm,t_i}\cancel{dt} \cdot \lambda_{m,t_i}dt \cdot b_{m,t_i+}}{b_{n,t_i-}\cancel{dt}} \\
&= \lim_{dt\to 0} \frac{\lambda'_{nm,t_i} \cdot \lambda_{m,t_i}\cancel{dt} \cdot b_{m,t_i+}}{\lambda_{n,t_i}\cancel{dt} \cdot b_{n,t_i+}} \\
&= r_{n\to m,t_i+} \cdot \frac{\lambda_{m,t_i}}{\lambda_{n,t_i}}.
\end{aligned} \tag{C.22}$$

equations C.21 and C.22 have an intuitive explanation. Between spikes, $r_{n\to m,t}$ (the expected rate of transition from state n to state m) is equal to $\lambda'_{nm,t}$ (the rate given by the stimulus and the current parameter settings of the model) scaled by the ratio of the probabilities of the future given that the current state is m versus n. In other words, if the remainder of the spike train can be better explained by having the neuron in state m than in state n at time t, the rate should be increased beyond $\lambda'_{nm,t}$; otherwise, it should be reduced. At the spike times, the additional information of knowing that a spike occurred further scales the expected transition rate by the ratio of the firing rates between the two states, which is equal to the ratio of the probabilities of firing in each state. As is obvious from equations C.21 and C.22, the expected transition rates $r_{n\rightarrow m,t}$ are discontinuous at the spike times where they jump by a factor of $\frac{\lambda_{m,t_i}}{\lambda_{n,t_i}}$ but continuous between spikes.

C.2 The M-Step. In order to maximize the parameters during the M-step, the ECLL must be modified to the continuous-time framework. The update of the initial state distribution $\boldsymbol{\pi}$ is unchanged (see equation 2.19). As with the discrete-time case, we can consider the $\boldsymbol{\alpha}_t$ and $\boldsymbol{\eta}_t$ dependent terms separately. From equation 2.18, we have

$$\begin{aligned}
&\sum_{t=dt}^{T}\sum_{n=1}^{N}\sum_{m=1}^{N}\hat{p}(q_{t-dt}=n,q_t=m)\log\alpha_{nm,t}\\
&=\sum_{t=dt}^{T}\sum_{n=1}^{N}\left(\begin{array}{l}\sum_{m\neq n}\hat{p}(q_{t-dt}=n,q_t=m)\log\left(\lambda'_{nm,t}dt\right)\\ +\ \hat{p}(q_{t-dt}=n,q_t=n)\log\left(1-\sum_{l\neq n}\lambda'_{nl,t}dt\right)\end{array}\right)\\
&=\sum_{t=dt}^{T}\sum_{n=1}^{N}\left(\begin{array}{l}\sum_{m\neq n}\hat{p}(q_{t-dt}=n)r_{n\rightarrow m,t}dt(\log\lambda'_{nm,t}+\cancel{\log dt})\\ -\ \hat{p}(q_{t-dt}=n)\sum_{l\neq n}\lambda'_{nl,t}dt\end{array}\right)\\
&\sim\sum_{n=1}^{N}\sum_{m\neq n}\int_0^T\hat{p}(q_t=n)\left(r_{n\rightarrow m,t}\log\lambda'_{nm,t}-\lambda'_{nm,t}\right)dt
\end{aligned}\tag{C.23}$$

and

$$\begin{aligned}
&\sum_{t=dt}^{T}\sum_{n=1}^{N}\hat{p}(q_t=n)\log\eta_{ny_t,t}\\
&=\sum_{n=1}^{N}\left(\begin{array}{l}\sum_{i\in\text{spikes}}\hat{p}(q_{t_i}=n)\log\left(\lambda_{n,t_i}\cancel{dt}\right)\\ +\sum_{i\notin\text{spikes}}\hat{p}(q_{t_i}=n)\log\left(1-\lambda_{n,t_i}dt\right)\end{array}\right)\\
&\sim\sum_{n=1}^{N}\left(\sum_{i\in\text{spikes}}\hat{p}(q_{t_i}=n)\log\lambda_{n,t_i}-\int_0^T\hat{p}(q_t=n)\lambda_{n,t}dt\right).
\end{aligned}\tag{C.24}$$

The integrals in equations C.23 and C.24 have to be evaluated piecewise from spike time to spike time since the probabilities are not smooth at these times. In order to capitalize on the increased computational efficiency of the continuous-time framework, it would not be desirable to need to calculate the marginal posterior probabilities and expected instantaneous transition rates at times for which they were not calculated and stored during the E-step. If the numerical integration procedure calls for values at unstored times, they can be approximated by linearly interpolating the values for the two closest stored times.

As with the discrete-time case, $\langle L(\boldsymbol{\theta} \mid \mathbf{y}, \mathbf{q}, \mathbf{S})\rangle_{\hat{p}(\mathbf{q})}$ is maximized by gradient ascent. To guarantee the concavity of the ECLL, the constraints on the spiking nonlinearity remain the same as before (f must be convex and log concave) since $\lambda_{n,t} = f\left(\mathbf{k}_n{}^{\mathrm{T}}\mathbf{s}_t\right)$ and each $\lambda_{n,t}$ enters into equation C.24 as a log and a negative. A desirable feature of the continuous-time model is that these constraints are also enough to guarantee concavity with respect to g as opposed to the more stringent requirement from the discrete-time case that g had to be exponential. Equation C.23 depends on g only through $\log \lambda'_{nm,t}$ and $-\lambda'_{nm,t}$, where $\lambda'_{nm,t} = g\left(\mathbf{k}'_{nm}{}^{\mathrm{T}}\mathbf{s}_t\right)$.

Appendix D: The Trial-Triggered Model

The trial-triggered model is a hybrid HMM–PSTH for the modeling of the evolution of the firing rates of cells recorded from a multistate neural network in the absence of an external time-varying stimulus. This hybrid model can be cast in the framework developed in this article by defining the "stimulus" $\mathbf{s}_t$ to be the tth standard basis vector in $\mathbb{R}^T$, where T is the number of time steps in the trial. The spiking and transitioning filters $\mathbf{k}_n$ and $\mathbf{k}'_{nm}$ are thus length T, and each of the filter elements is used only once (i.e., the dot products $\mathbf{k}^{\mathrm{T}}\mathbf{s}_t$ yield the tth filter elements $[\mathbf{k}]_t$ thus giving the model as defined in section 2.3.1). The vectors $f(\mathbf{k}_n)$ and $g(\mathbf{k}'_{nm})$ are thus PSTHs for spiking in state n and for transitioning from state n to state m, respectively.

This model is guaranteed to overfit in the maximum likelihood setting, but we can combat this issue by using smoothness priors for each of the spiking and transitioning filters:

$$p(\mathbf{k}_n^c) \propto \exp\left\{-\frac{1}{2\sigma^2 dt}\sum_{t=0}^{T-1}\left(\left[\mathbf{k}_n^c\right]_{t+1} - \left[\mathbf{k}_n^c\right]_t\right)^2\right\} \tag{D.1}$$

and

$$p(\mathbf{k}'_{nm}) \propto \exp\left\{-\frac{1}{2\sigma'^2 dt}\sum_{t=1}^{T-1}\left(\left[\mathbf{k}'_{nm}\right]_{t+1} - \left[\mathbf{k}'_{nm}\right]_t\right)^2\right\}, \tag{D.2}$$

where σ and σ' are hyperparameters determining the degree of smoothness required by the spiking and transitioning filters, respectively.

Learning the parameters of this smoothed model involves maximizing the log posterior distribution of the parameters given the data to find the maximum a posteriori ($\boldsymbol{\theta}_{\mathrm{MAP}}$), rather than the maximum likelihood, solution:

$$\begin{aligned}\boldsymbol{\theta}_{\mathrm{MAP}} &= \arg\max_{\boldsymbol{\theta}} \Big[\log p(\boldsymbol{\theta} \mid \mathbf{Y}) \Big] \\ &= \arg\max_{\boldsymbol{\theta}} \Big[\log p(\mathbf{Y} \mid \boldsymbol{\theta}) + \log p(\boldsymbol{\theta}) \Big] \\ &= \arg\max_{\boldsymbol{\theta}} \left[L(\boldsymbol{\theta} \mid \mathbf{Y}) + \sum_{n,c} \log p(\mathbf{k}_n^c) + \sum_{n,m} \log p\big(\mathbf{k}'_{nm}\big) \right]. \qquad \text{(D.3)}\end{aligned}$$

One advantage of the maximum a posteriori setting is that regardless of the choice of the hyperparameters σ and σ', $\boldsymbol{\theta}_{\mathrm{MAP}}$ is guaranteed to converge to the true parameters of the system with sufficient data (i.e., the likelihood term in equation D.3 will dominate the terms corresponding to the prior distributions). This is in contrast to other smoothing techniques (e.g., binning or using a gaussian filter), which will always coarsen the estimate despite the quantity of data. With fewer data points, however, as in the data set analyzed in section 4, the optimal values of σ and σ' can be found by searching over these hyperparameters and finding the peak of the cross-validated log likelihood.

During the E-step, the first term of equation D.3 (the log likelihood) is computed using the forward-backward algorithm exactly as in the standard HMM setting, and the other terms are computed using the definitions of the priors in equations D.1 and D.2. The M-step is modified slightly in that the maximization occurs over the expected complete log posterior (ECLP) rather than over the ECLL. The ECLP is given as

$$\begin{aligned}\big\langle \log p(\boldsymbol{\theta} \mid \mathbf{Y}, \mathbf{q}) \big\rangle_{\hat{p}(\mathbf{q})} &\sim \big\langle \log p(\mathbf{Y}, \mathbf{q} \mid \boldsymbol{\theta}) + \log p(\boldsymbol{\theta}) \big\rangle_{\hat{p}(\mathbf{q})} \\ &\sim \big\langle L(\boldsymbol{\theta} \mid \mathbf{Y}, \mathbf{q}) \big\rangle_{\hat{p}(\mathbf{q})} + \log p(\boldsymbol{\theta}). \qquad \text{(D.4)}\end{aligned}$$

In other words, the sum of the ECLL (given by equation 2.44) and the log prior gives the ECLP. Note that from the definitions given in equations D.1 and D.2, the log priors are quadratic forms and thus have trivial gradients and Hessians, which can be simply summed to those of the ECLL (see appendix B) to get the gradients and Hessians of the ECLP as needed to perform the parameter updates for each M-step. As before, the

Newton-Raphson method is used to solve the M-step:

$$\mathbf{k}^{i+1} = \mathbf{k}^{i} - H\big(\mathbf{k}^{i}, \mathbf{k}^{i}\big)^{-1} \vec{\nabla}\big(\mathbf{k}^{i}\big), \tag{D.5}$$

where $\vec{\nabla}(\mathbf{k}^i)$ and $H(\mathbf{k}^i, \mathbf{k}^i)$ are the gradient and Hessian of the ECLP evaluated at the current parameter setting $\mathbf{k}^i$. Since the dimension of $\mathbf{k}$ is T and given that the complexity of the inversion of a $T \times T$ matrix is $\mathcal{O}(T^3)$, it would appear that the convenient linear dependence on T is no longer preserved for the trial-triggered model. However, due to the decomposition of the ECLL into independent state- and cell-specific terms (see equation 2.44) and due to the nature of the smoothness priors (see equations D.1 and D.2), the Hessian of the ECLP is a sparse, banded matrix[12] for which $\mathcal{O}(T)$ algorithms exist to perform the update given in equation D.5 (Paninski et al., 2009).

As with the other models discussed in this article, the trial-triggered model can also account for spike history effects. Assuming that the transitioning behavior is not history dependent and that the history dependence of the spiking behavior is not state dependent—reasonable assumptions that limit the number of parameters required for the inclusion of history effects—then the full descriptions of the transitioning and firing rates become

$$\lambda'_{nm,t} = g([\mathbf{k}'_{nm}]_t) \tag{D.6}$$

and

$$\lambda^{c}_{n,t} = f\big(\big[\mathbf{k}^{c}_{n}\big]_t + \mathbf{h}^{\mathrm{T}}_{n} \boldsymbol{\gamma}^{c}_{t}\big), \tag{D.7}$$

where $\mathbf{h}_n$ and $\boldsymbol{\gamma}^c_t$ are defined as in section 2.2.2.

Appendix E: The Transition-Triggered Model

The transition-triggered model solves the limitation of the trial-triggered model (see appendix D) by decoupling the evolution of the state-dependent firing rates from the time of the onset of the trial as illustrated in Figure 3b. Specifically, when the system is in state n_τ (the τth state in the nth row of states), the restricted state-space connectivity permits transitions to the next state only in the current row of states ($n_{\tau+1}$) or to one of the start states m_0, where m may equal n. This topology decouples the time-step t from the

[12] Specifically, the sub-Hessians corresponding to the transition filter updates can be shown to have N unique bands (the main diagonal, $N-1$ upper subdiagonals, and $N-1$ symmetric lower subdiagonals) where N is the number of states, while the sub-Hessians corresponding to the spiking filters prove to be tridiagonal.

depth in the state-space cascade τ and thus permits the state-dependent filters to evolve from the time of the last "state" transition (i.e., transition to the begin of a row of states). These filters can be thought of as peritransition time histograms (PTTHs) rather than PSTHs, as in the trial-triggered case. The spiking and transitioning rates are thus given as

$$\lambda'_{n_\tau m_0} = g\left(k'_{n_\tau m_0}\right) \tag{E.1}$$

and

$$\lambda^c_{n_\tau,t} = f\left(k^c_{n_\tau} + \mathbf{h}_n^{\mathrm{T}}\boldsymbol{\gamma}^c_t\right), \tag{E.2}$$

where, unlike in the trial-triggered model, the state-specific firing rates are no longer time homogeneous.

Learning the parameters of the transition-triggered model is more difficult than learning those of the other models discussed in this article. Unlike in the trial-triggered case, there is no formulation of the model that uses some kind of "stimulus" as an indicator to select the appropriate filter elements at each time step, and so the full NT-state system must be considered, albeit with certain simplifications due to the restricted structure of the transition matrix. First, we note that the forward recursion (see equation 2.9) has different expressions for states with $\tau = 0$ and $\tau > 0$. In the latter case, any state n_τ for $\tau > 0$ can have been reached only on time-step t if the system was in state $n_{\tau-1}$ at time $t-1$ (as is clear from the state-space connectivity given in Figure 3b). Thus, for these states, the forward recursion simplifies as

$$a_{n_\tau,t} = a_{n_{\tau-1},t-1}\alpha_{n_{\tau-1}n_\tau}\eta_{n_\tau y_t} \qquad 0 < \tau \le t, \tag{E.3}$$

where, in the multicell setting, $\eta_{n_\tau y_t}$ will become $\prod_c \eta^c_{n_\tau y^c_t}$. The states with $\tau = 0$, on the other hand, can have been reached on time step t from any state with $\tau < t$, which follows from the fact that the farthest depth into the state-space cascade that is achievable $t-1$ time steps following the trial onset is $\tau - 1$. For these states, the forward recursion becomes

$$a_{n_0,t} = \left(\sum_{m=1}^{N}\sum_{\tau<t} a_{m_\tau,t-1}\alpha_{m_\tau n_0}\right)\eta_{n_\tau y_t}. \tag{E.4}$$

As always, the likelihood is computed from the forward probabilities at time T:

$$p(\mathbf{Y} \mid \boldsymbol{\theta}) = \sum_{n,\tau} a_{n_\tau,T}. \tag{E.5}$$

Note that the computational complexity of the forward recursion as well as the storage requirement for the forward probabilities are now both $\mathcal{O}(T^2)$, whereas they had previously been $\mathcal{O}(T)$. This significant loss in efficiency is the cost of a transition-triggered model. In order to have the behavior of the system triggered on the transition times rather than the elapsed time since the trial onset, the forward recursion must keep track of both the transition index τ and the trial index t, thus squaring the complexity (although it would be possible to reduce this quadratic dependence if the maximum value of τ were restricted to be less than T).

The backward recursion is updated as

$$b_{n_\tau,t} = \left[\sum_{m=1}^{N} \alpha_{n_\tau m_0} \eta_{m_0 y_{t+1}} b_{m_0,t+1}\right] + \alpha_{n_\tau n_{\tau+1}} \eta_{n_{\tau+1} y_{t+1}} b_{n_{\tau+1},t+1}, \tag{E.6}$$

where, as with the forward recursion, transitions to the start states in the first column of Figure 3b must be treated differently from the transitions along the rows of the state-space. The sum in equation E.6 deals with the former set of transitions (where all states m_0 are reachable from any state n_τ), while the final term deals with the latter transitions (where only state $n_{\tau+1}$ is reachable from n_τ). Again, the time and storage complexities of this recursion are $\mathcal{O}(T^2)$.

The single marginal probabilities of $p(\mathbf{q} \mid \mathbf{Y}, \boldsymbol{\theta})$ are calculated as before (see equation 2.14):

$$\hat{p}(q_t = n_\tau) = \frac{a_{n_\tau,t} b_{n_\tau,t}}{p(\mathbf{Y} \mid \boldsymbol{\theta})}. \tag{E.7}$$

However, only the following subset of the consecutive-pairwise marginals is needed, as will be shown:

$$\hat{p}(q_t = n_\tau, q_{t+1} = m_0) = \frac{a_{n_\tau,t} \alpha_{n_\tau m_0} \eta_{m_0 y_{t+1}} b_{m_0,t+1}}{p(\mathbf{Y} \mid \boldsymbol{\theta})}. \tag{E.8}$$

Again the complexity of the calculations of the marginals is $\mathcal{O}(T^2)$.

The M-step is also somewhat updated from before. The transition-dependent term of the ECLL (see equation 2.37) becomes

$$\begin{aligned} &\sum_{r=1}^{R} \sum_{t=1}^{T} \left\langle \log \alpha_{q_{t-1}^r q_t^r} \right\rangle_{\hat{p}(\mathbf{q}^r)} \\ &\quad \sim \sum_{r=1}^{R} \sum_{t=1}^{T} \sum_{n=1}^{N} \sum_{\tau=0}^{t-1} \left(\begin{array}{l} \sum_{m=1}^{N} \hat{p}(q_{t-1}^r = n_\tau, q_t^r = m_0) \log \lambda'_{n_\tau m_0} \\ -\ \hat{p}(q_{t-1}^r = n_\tau) \log \left(1 + \sum_l \lambda'_{n_\tau l_0} dt\right) \end{array} \right), \end{aligned} \tag{E.9}$$

where the sum over τ reflects the fact that the depth in the state-space τ can never exceed time-step t, and the sum over m the fact that transitions to the start states are parameterized (e.g., $k'_{n_\tau m_0}$ parameterizes the transition from n_τ to m_0). Recall that the transitions along the rows of the state-space (e.g., from n_τ to $n_{\tau+1}$) are not parameterized and are determined by the residual probability (see equation 2.33). By rearranging the sums of t and τ, the expression takes a more convenient form:

$$\sum_{r=1}^{R}\sum_{t=1}^{T}\left\langle \log \alpha_{q^r_{t-1} q^r_t} \right\rangle_{\hat{p}(\mathbf{q}^r)}$$

$$\sim \sum_{n=1}^{N}\sum_{r=1}^{R}\sum_{\tau=0}^{T-1}\left(\begin{array}{l} \displaystyle\sum_{m=1}^{N}\left[\sum_{t=\tau+1}^{T} \hat{p}(q^r_{t-1} = n_\tau, q^r_t = m_0)\right] \log \lambda'_{n_\tau m_0} \\ \displaystyle -\left[\sum_{t=\tau+1}^{T} \hat{p}(q^r_{t-1} = n_\tau)\right] \log\left(1 + \textstyle\sum_l \lambda'_{n_\tau l_o} dt\right) \end{array}\right). \tag{E.10}$$

It is clear from equation E.10 that the only consecutive-pairwise marginals that are needed are those involving transitions to the first column of states (given by equation E.8). Furthermore, by moving the sum over time-step t to be the inner sum, it is clear that the consecutive-pairwise marginals themselves are not needed, but rather the sum over t of these marginals for the transitions between every state n_τ and each of the states in the first column m_0. Although computation of these summed pairwise marginals still requires $\mathcal{O}(T^2)$ time, the storage requirement is only $\mathcal{O}(T)$. However, the emission-dependent term shows that the full $\mathcal{O}(T^2)$ set of single marginals still needs to be stored for the update of the spiking parameters:

$$\sum_{r=1}^{R}\sum_{c=1}^{C}\sum_{t=0}^{T}\left\langle \log \eta^c_{q^r_t y^{c,r}_t, t} \right\rangle_{\hat{p}(\mathbf{q}^r)}$$

$$= \sum_{r=1}^{R}\sum_{c=1}^{C}\sum_{t=0}^{T}\sum_{n=1}^{N}\sum_{\tau \le t} \hat{p}(q^r_t = n_\tau) \log \frac{\left(\lambda^c_{n_\tau, t} dt\right)^{y^{c,r}_t} e^{-\lambda^c_{n_\tau, t} dt}}{y^{c,r}_t !}$$

$$\sim \sum_{c=1}^{C}\sum_{r=1}^{R}\sum_{t=0}^{T}\sum_{n=1}^{N}\sum_{\tau \le t} \hat{p}(q^r_t = n_\tau)\left(y^{c,r}_t \log \lambda^c_{n_\tau, t} - \lambda^c_{n_\tau, t} dt\right). \tag{E.11}$$

Note that this expression is unchanged from equation 2.38 except for the additional sums.

Newton-Raphson optimization can again be employed to update the parameters of the transition-triggered model, as the gradient and Hessian of the ECLL are essentially unchanged from the standard setting (albeit with the additional sums present in equations E.10 and E.11). As with the trial-triggered model, the inversion of the Hessian mandated by the optimization procedure can be computed efficiently (i.e., in $\mathcal{O}(T)$ time) by exploiting the banded structure of the Hessian. Since the computation required to construct the Hessian during each M-step is $\mathcal{O}(T^2)$, it is clear that the Newton-Raphson update itself is not a computational bottleneck in this setting. Note that as with the trial-triggered model, smoothness priors (see equations D.1 and D.2) are employed to prevent overfitting.

Appendix F: Concavity Constraints for the Bernoulli Spiking Model

If the emission model is modified to guarantee that no more than one spike occurs per time step, then it is necessary to reevaluate the sufficient conditions for concavity of the M-step. Recall from equation 2.38 that the nonlinearity f entered into the ECLL as both its logarithm and its negative. Therefore, the sufficient and necessary conditions for guaranteeing concavity were that $\log f$ needed to be concave and f needed to be convex. In a Bernoulli spiking model, the $\boldsymbol{\eta}_t$ term in equation 2.36 becomes

$$\begin{aligned}\left\langle L(\boldsymbol{\theta} \mid \mathbf{y}, \mathbf{q}, \mathbf{S})\right\rangle_{\hat{p}(\mathbf{q})} &\sim \sum_{t=0}^{T}\sum_{n=1}^{N} \hat{p}(q_t = n)\log \eta_{ny_t,t} \\ &\sim \sum_{n=1}^{N}\left(\begin{array}{l}\displaystyle\sum_{t\in\text{spikes}} \hat{p}(q_t = n)\log p(y_t = 1 \mid q_t = n) \\ \displaystyle\quad + \sum_{t\notin\text{spikes}} \hat{p}(q_t = n)\log p(y_t = 0 \mid q_t = n)\end{array}\right).\end{aligned} \tag{F.1}$$

The actual dependence of equation F.1 on the nonlinearity f is determined by the definition of $p(y_t = 0)$ and $p(y_t = 1)$. One reasonable choice is to have the probability of not spiking be the same as in the Poisson spiking model. The probability of having a spike is then equal to the probability of one or more spikes in the Poisson model. This model has the desirable property that in the limit of small dt, the Bernoulli formulation converges to the Poisson formulation. Thus, we have

$$p(y_t = 0 \mid q_t = n) \equiv \text{Poisson}(0 \mid \lambda_{n,t}dt) = e^{-f(\mathbf{k}_n{}^{\mathrm{T}}\mathbf{s}_t)\,dt} \tag{F.2}$$

and

$$p(y_t = 1 \mid q_t = n) \equiv \sum_{i=1}^{\infty} \text{Poisson}(i \mid \lambda_{n,t}dt) = 1 - e^{-f(\mathbf{k}_n{}^{\mathrm{T}}\mathbf{s}_t)\,dt}, \tag{F.3}$$

where we have used equation 2.34 and the fact that the probabilities must sum to 1. Substituting these definitions into equation F.1 gives

$$\begin{aligned}\left\langle L(\boldsymbol{\theta} \mid \mathbf{y}, \mathbf{q}, \mathbf{S})\right\rangle_{\hat{p}(\mathbf{q})} \sim & \sum_{n=1}^{N} \sum_{t\in\text{spikes}} \hat{p}(q_t = n)\log\left(1 - e^{-f(\mathbf{k}_n{}^{\mathrm{T}}\mathbf{s}_t)\,dt}\right) \\ & - \sum_{t\notin\text{spikes}} \hat{p}(q_t = n) f\left(\mathbf{k}_n{}^{\mathrm{T}}\mathbf{s}_t\right) dt.\end{aligned} \tag{F.4}$$

To guarantee the concavity of the M-step, both terms in equation F.4 must be concave in $\mathbf{k}_n$. The second term involves a negative f as before in equation 2.38, and so f must be convex. If we assume that the other original constraint also holds—that $\log f$ is concave—then we can show that the first term will also be concave. From the concavity of $\log f$, we have

$$\begin{aligned}(\log f)'' &\leq 0 \\ \left(\frac{f'}{f}\right)' &\leq 0 \\ \frac{ff'' - (f')^2}{f^2} &\leq 0 \\ -(f')^2 &\leq -ff''.\end{aligned} \tag{F.5}$$

The second derivative of the first term in equation F.4 can be expanded as follows:

$$\begin{aligned}\left(\log\left(1 - e^{-f dt}\right)\right)'' &= \left(\frac{e^{-f dt} f' dt}{1 - e^{-f dt}}\right)' \\ &= \left(\frac{f' dt}{e^{f dt} - 1}\right)' \\ &= \frac{(e^{f dt} - 1) f'' dt - e^{f dt} (f')^2 dt^2}{(e^{f dt} - 1)^2} \\ &\leq \frac{(e^{f dt} - 1) f'' dt - e^{f dt} f f'' dt^2}{(e^{f dt} - 1)^2} \\ &= \frac{e^{f dt} f'' dt (1 - e^{-f dt} - f dt)}{(e^{f dt} - 1)^2} \\ &\leq 0,\end{aligned} \tag{F.6}$$

where we have used the result from equation F.5, the convexity of f (i.e., $f'' \geq 0$), and the fact that $1 - e^{u} + u \leq 0$ for all values of u. Thus, we have shown that for the Bernoulli spiking model as defined by equations F.2 and F.3, convexity and log concavity of the nonlinearity f are sufficient conditions to guarantee the concavity of the M-step.

F.1 Gradient and Hessian of the Bernoulli Spiking Model. The analytic formulas for the gradient and Hessian required to maximize the ECLL during each M-step must be redetermined for the Bernoulli spiking model. From equation F.4, we derive the following gradient,

$$\vec{\nabla}(\mathbf{k}_n) = \sum_{t \in \text{spikes}} \hat{p}(q_t = n) \frac{f'\big(\mathbf{k}_n{}^{\mathrm{T}}\mathbf{s}_t\big)\,dt}{e^{f(\mathbf{k}_n{}^{\mathrm{T}}\mathbf{s}_t)\,dt} - 1}\mathbf{s}_t - \sum_{t \notin \text{spikes}} \hat{p}(q_t = n) f'\big(\mathbf{k}_n{}^{\mathrm{T}}\mathbf{s}_t\big)\,dt\,\mathbf{s}_t, \tag{F.7}$$

and Hessian,

$$\begin{aligned} H(\mathbf{k}_n, \mathbf{k}_n) = & \sum_{t \in \text{spikes}} \hat{p}(q_t = n) \\ & \times \frac{\big(e^{f(\mathbf{k}_n{}^{\mathrm{T}}\mathbf{s}_t)\,dt} - 1\big) f''(\mathbf{k}_n{}^{\mathrm{T}}\mathbf{s}_t)\,dt - e^{f(\mathbf{k}_n{}^{\mathrm{T}}\mathbf{s}_t)\,dt}\big[f'(\mathbf{k}_n{}^{\mathrm{T}}\mathbf{s}_t)\,dt\big]^2}{\big(e^{f(\mathbf{k}_n{}^{\mathrm{T}}\mathbf{s}_t)\,dt} - 1\big)^2}\mathbf{s}_t\mathbf{s}_t{}^{\mathrm{T}} \\ & - \sum_{t \notin \text{spikes}} \hat{p}(q_t = n) f''(\mathbf{k}_n{}^{\mathrm{T}}\mathbf{s}_t)\,dt\,\mathbf{s}_t\mathbf{s}_t{}^{\mathrm{T}}. \end{aligned} \tag{F.8}$$

Acknowledgments

S.E. acknowledges the NIH Medical Scientist Training Program, the Columbia University M.D.-Ph.D. Program, and the Leon Levy Foundation for supporting this research. L.P. is supported by an NSF CAREER award, an Alfred P. Sloan Research Fellowship, and a McKnight Scholar award. We also thank K. Miller, B. Bialek, L. Abbott, M. Eisele, T. Toyoizumi, X. Pitkow, and J. Pillow for many helpful discussions. We additionally thank our reviewer for pointing out the potential application of our work outlined in section 5.3.

References

Abeles, M., Bergman, H., Gat, I., Meilijson, I., Seidemann, E., Tishby, N., et al. (1995). Cortical activity flips among quasi-stationary states. *Proceedings of the National Academy of Sciences of the United States of America, 92*, 8616–8620.

Ahrens, M. B., Linden, J. F., & Sahani, M. (2008). Nonlinearities and contextual influences in auditory cortical responses modeled with multilinear spectrotemporal methods. *Journal of Neuroscience, 28*(8), 1929–1942.

Ahrens, M. B., Paninski, L., & Sahani, M. (2008). Inferring input nonlinearities in neural encoding models. *Network, 19,* 35–67.

Anderson, J., Lampl, I., Reichova, I., Carandini, M., & Ferster, D. (2000). Stimulus dependence of two-state fluctuations of membrane potential in cat visual cortex. *Nature Neuroscience, 3,* 617–621.

Baum, L., Petrie, T., Soules, G., & Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Annals of Mathematical Statistics, 41,* 164–171.

Bezdudnaya, T., Cano, M., Bereshpolova, Y., Stoelzel, C. R., Alonso, J.-M., & Swadlow, H. A. (2006). Thalamic burst mode and inattention in the awake LGND. *Neuron, 49,* 421–432.

Blake, D. T., & Merzenich, M. M. (2002). Changes of AI receptive fields with sound density. *Journal of Neurophysiology, 88*(6), 3409–3420.

Borst, A., Flanagin, V. L., & Sompolinsky, H. (2005). Adaptation without parameter change: Dynamic gain control in motion detection. *Proceedings of the National Academy of Sciences of the United States of America, 102*(17), 6172–6176.

Brown, E. N., Nguyen, D. P., Frank, L. M., Wilson, M. A., & Solo, V. (2001). An analysis of neural receptive field plasticity by point process adaptive filtering. *Proceedings of the National Academy of Sciences of the United States of America, 98,* 12261–12266.

Chan, K., & Ledolter, J. (1995). Monte Carlo EM estimation for time series models involving counts. *Journal of the American Statistical Association, 90,* 242–252.

Chen, Z., Vijayan, S., Barbieri, R., Wilson, M. A., & Brown, E. N. (2009). Discrete- and continuous-time probabilistic models and algorithms for inferring neuronal up and down states. *Neural Computation, 21,* 1797–1862.

Czanner, G., Eden, U., Wirth, S., Yanike, M., Suzuki, W., & Brown, E. (2008). Analysis of between-trial and within-trial neural spiking dynamics. *Journal of Neurophysiology, 99,* 2672–2693.

Dempster, A., Laird, N., & Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39, 1–38.

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

Escola, S. (2009). *Markov chains, neural responses, and optimal temporal computations*. Unpublished doctoral dissertation, Columbia University.

Fontanini, A., & Katz, D. B. (2006). State-dependent modulation of time-varying gustatory responses. *Journal of Neurophysiology, 96,* 3183–3193.

Fox, E. B., Sudderth, E. B., Jordan, M. I., & Willsky, A. S. (2008). An HDP-HMM for systems with state persistence. In *Proc. International Conference on Machine Learning*. Piscataway, NJ: IEEE Press.

Frank, L., Eden, U., Solo, V., Wilson, M., & Brown, E. (2002). Contrasting patterns of receptive field plasticity in the hippocampus and the entorhinal cortex: An adaptive filtering approach. *J. Neurosci., 22*(9), 3817–3830.

Gat, I., Tishby, N., & Abeles, M. (1997). Hidden Markov modeling of simultaneously recorded cells in the associative cortex of behaving monkeys. *Network: Computation in Neural Systems, 8,* 297–322.

Guédon, Y. (2003). Estimating hidden semi-Markov chains from discrete sequences. *Journal of Computational and Graphical Statistics, 12,* 604–639.

Haider, B., Duque, A., Hasenstaub, A. R., Yu, Y., & McCormick, D. A. (2007). Enhancement of visual responsiveness by spontaneous local network activity in vivo. *Journal of Neurophysiology, 97,* 4186–4202.

Hong, S., Lundstrom, B. N., & Fairhall, A. L. (2008). Intrinsic gain modulation and adaptive neural coding. *PLoS Computational Biology, 4*(7), e1000119.

Jin, D. (2009). Generating variable birdsong syllable sequences with branching chain networks in avian premotor nucleus HVC. *Phys. Rev. E, 80,* 051902.

Jones, L., Fontanini, A., Sadacca, B., Miller, P., & Katz, D. (2007). Natural stimuli evoke dynamic sequences of states in sensory cortical ensembles. *Proceedings of the National Academy of Sciences, 104,* 18772–18777.

Kass, R., & Ventura, V. (2001). A spike-train probability model. *Neural Comp., 13,* 1713–1720.

Kass, R., Ventura, V., & Cai, C. (2003). Statistical smoothing of neuronal data. *Network: Computation in Neural Systems, 14,* 5–15.

Kemere, C., Santhanam, G., Yu, B. M., Afshar, A., Ryu, S. I., Mẽng, T. H., et al. (2008). Detecting neural-state transitions using hidden Markov models for motor cortical prostheses. *J. Neurophysiol., 100,* 2441–2452.

Kulkarni, J. E., & Paninski, L. (2007). Common-input models for multiple neural spike-train data. *Network (Bristol, England), 18,* 375–407.

Maravall, M., Petersen, R. S., Fairhall, A. L., Arabzadeh, E., & Diamond, M. E. (2007). Shifts in coding properties and maintenance of information transmission during adaptation in barrel cortex. *PLoS Biology, 5*(2), e19.

Paninski, L. (2004). Maximum likelihood estimation of cascade point-process neural encoding models. *Network: Computation in Neural Systems, 15,* 243–262.

Paninski, L., Ahmadian, Y., Ferreira, D. G., Koyama, S., Rahnama Rad, K., Vidne, M., Vogelstein, J., et al. (2009). A new look at state-space models for neural data. *Journal of Computational Neuroscience, 29,* 107–126.

Paninski, L., Pillow, J., & Lewi, J. (2007). Statistical models for neural encoding, decoding, and optimal stimulus design. In P. Cisek, T. Drew, and J. Kalaska (Eds.), *Computational neuroscience: Progress in brain research*. Amsterdam: Elsevier.

Rabiner, L. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE, 77,* 257–286.

Rahnama Rad, K., & Paninski, L. (2010). Efficient estimation of two-dimensional firing rate surfaces via gaussian process methods. *Network: Computation in Neural Systems, 21,* 142–168.

Ramcharan, E., Gnadt, J., & Sherman, S. (2000). Burst and tonic firing in thalamic cells of unanesthetized, behaving monkeys. *Visual Neuroscience, 17,* 55–62.

Salakhutdinov, R., Roweis, S., & Ghahramani, Z. (2003). Optimization with EM and expectation-conjugate-gradient. *In Proceedings of the Twentieth International Conference on Machine Learning*. Menlo Park, CA: AAAI Press.

Sanchez-Vives, M. V., & McCormick, D. A. (2000). Cellular and network mechanisms of rhythmic recurrent activity in neocortex. *Nature Neuroscience, 3,* 1027–1034.

Sansom, J., & Thomson, P. (2001). Fitting hidden semi-Markov models to breakpoint rainfall data. *Journal of Applied Probability, 38,* 142–157.

Seidemann, E., Meilijson, I., Abeles, M., Bergman, H., & Vaadia, E. (1996). Simultaneously recorded single units in the frontal cortex go through sequences of discrete and stable states in monkeys performing a delayed localization task. *Journal of Neuroscience, 16,* 752–768.

Sherman, S. M. (2001). Tonic and burst firing: Dual modes of thalamocortical relay. *Trends in Neurosciences, 24,* 122–126.

Simoncelli, E., Paninski, L., Pillow, J., & Schwartz, O. (2004). Characterization of neural responses with stochastic stimuli. In M. Gazzaniga (Ed.), *The cognitive neurosciences* (3rd ed.). Cambridge, MA: MIT Press.

Smith, A. C., & Brown, E. N. (2003). Estimating a state-space model from point process observations. *Neural Computation, 15,* 965–991.

Tokdar, S., Xi, P., Kelly, R. C., & Kass, R. E. (2009). Detection of bursts in extracellular spike trains using hidden semi-Markov point process models. *Journal of Computational Neuroscience, 29,* 203–212.

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

Wahba, G. (1990). *Spline models for observational data*. Philadelphia: SIAM.

Wang, X., Wei, Y., Vaingankar, V., Wang, Q., Koepsell, K., Sommer, F. T., et al. (2007). Feedforward excitation and inhibition evoke dual modes of firing in the cat's visual thalamus during naturalistic viewing. *Neuron, 55,* 465–478.

Wu, W., Black, M. J., Mumford, D., Gao, Y., Bienenstock, E., & Donoghue, J. P. (2004). Modeling and decoding motor cortical activity using a switching Kalman filter. *IEEE Transactions on Bio-Medical Engineering, 51,* 933–942.

Wu, W., Kulkarni, J., Hatsopoulos, N., & Paninski, L. (2009). Neural decoding of goal-directed movements using a linear statespace model with hidden states. *IEEE Trans. Neural Syst. Rehabil. Eng., 17,* 370–378.

Yu, B., Afshar, A., Santhanam, G., Ryu, S., Shenoy, K., & Sahani, M. (2006). Extracting dynamical structure embedded in neural activity. In Y. Weiss, B. Schölkopf, & J. Platt (Eds.), *Advances in neural information processing systems, 18* (pp. 1545–1552). Cambridge, MA: MIT Press.

Received August 30, 2007; accepted September 16, 2010.