

# Neurons as Monte Carlo Samplers: Bayesian Inference and Learning in Spiking Networks

Yanping Huang & Rajesh P. N. Rao

Department of Computer Science and Engineering

University of Washington, Seattle, USA

{huangyp, rao}@cs.washington.edu

## Abstract

We propose a new model of Bayesian computation in a two-layer recurrent spiking network. The lower layer sensory neurons receive noisy measurements of hidden world states. The higher layer neurons infer a posterior distribution over world states via Bayesian inference from spike trains generated by sensory neurons. We show how such a neuronal network with synaptic plasticity can implement a form of Bayesian inference similar to Monte Carlo methods such as particle filtering. Each spike in the population of inference neurons represents a sample of a particular hidden world state. The spiking activity across the neural population approximates the posterior distribution of hidden state. The model provides a functional explanation for the Poisson-like noise commonly observed in cortical responses. Uncertainties in spike times provide the necessary variability for sampling during inference. Unlike previous models, the hidden world state is not observed by the sensory neurons, and the temporal dynamics of the hidden state is unknown. We demonstrate how the network can learn the likelihood model as well as the transition probabilities underlying the dynamics using a spike-timing dependent Hebbian learning rule. Our results illustrate the ability of model to explain neurobiological data: (1) the network exhibits history dependent adaptation to light intensity, mimicking responses seen in the visual system; (2) neurons modeling area CA1 change the shapes of their receptive fields after learning in a manner consistent with experimental observations in rat hippocampal place cells.

**Keywords:** Bayesian inference, spiking networks, hidden Markov models, sequential Monte Carlo sampling, Hebbian learning, online expectation-maximization, stochastic approximation.

## 1 Introduction

Animals constantly face the problem of estimating unknown world states from ambiguous and noisy stimuli. For example, when inferring 3D structure from a 2D image, the neural system must choose one among many possible interpretations that are consistent with the projected 2D image. A mouse in a maze must estimate its current location indirectly from noisy sensory evidence such as whisker deflections, sight, and odor. In such situations, the brain needs to combine noisy sensory information with incomplete knowledge of the environment. An optimal way of combining such information is to use Bayesian inference, where the level of uncertainty for each possible state is represented as a probability distribution (Zemel et al. 2005). Behavioral and neuropsychophysical experiments (Knill & Richards 1996, Rao et al. 2002, Kording & Wolpert 2004, Doya et al. 2007, Wark et al. 2009) have suggested that the brain may indeed maintain such a representation and employ Bayesian inference in a great variety of tasks in perception, sensori-motor integration, and sensory adaptation. However, the neural implementation of such Bayesian models remains an open question.

Furthermore, the world is dynamic, putting a premium on the ability to actively anticipate upcoming events by learning the temporal dynamics of relevant states of the world. For example, when facing an approaching tennis ball, a player must not only estimate the current position of the ball, but also predict its trajectory by inferring the ball's velocity and acceleration, before deciding on the next stroke. The relevant hidden variables (e.g., velocity, acceleration) are not directly available but must be estimated from retinal images. Tasks such as these can be modeled using a hidden Markov model (Rabiner 1989), where the relevant states of the world are latent variables related to sensory observations via a likelihood model (determined by the *emission probability matrix*). The states themselves evolve over time in a Markovian manner, the dynamics being governed by a *transition probability matrix*.

In this article, we propose a new model of Bayesian computation in networks of

spiking neurons. We show how the time-varying posterior probability distribution for a hidden Markov model can be directly represented by mean spike counts in sub-populations of neurons, without invoking complicated decoding methods. Each spike in the posterior population is viewed as a Monte Carlo sample of a particular world state. The probability that a neuron’s membrane potential exceeds spiking threshold is shown to approximate the posterior probability of the preferred state encoded by the neuron. Neurons within the same sub-population encode the same preferred state. The resulting responses of model neurons exhibit a characteristic property of cortical neurons, namely, that the variance of spike count proportional to the mean. In this model, variability in spiking is not regarded as a nuisance but an integral feature that provides the variability necessary for sampling during inference.

We show that a population of leaky integrate-and-fire (LIF) neurons with short-term synaptic depression can perform approximate Bayesian inference similar to the Monte Carlo method of *particle filtering* (Doucet et al. 2001). The posterior spike distribution is recursively updated in a Bayesian manner by integrating feedforward spikes, whose firing probabilities represent the likelihood of sensory measurements, with recurrent spikes, which represent the previous posterior probability distribution. The model thus provides a concrete neural implementation of ideas previously suggested in (Hoyer et al. 2002, Lee & Mumford 2003, Paulin 2005). We assume that the sensory neurons do not have direct access to the hidden world state and only observe noisy stimuli. We illustrate how a spiking network can learn the parameters of a hidden Markov model by using a spike-timing based Hebbian learning rule to implement an online version of Expectation-Maximization(EM) algorithm.

## 2 Spiking Network Model

### 2.1 Hidden Markov models and grid-based filtering

We begin by considering a discrete-time hidden Markov process  $\{X_k, k \in \mathbb{N}\}$  such that

$$X_{k+1} | (X_k = x') \sim f(x|x'), \quad x, x' \in \mathbb{X}. \quad (1)$$

where  $f(x|x')$  is the transition probability density,  $\mathbb{X}$  is the state space of  $X_k$  (e.g., real-valued vectors),  $\mathbb{N}$  is the set of natural numbers, and “ $\sim$ ” denotes distributed according

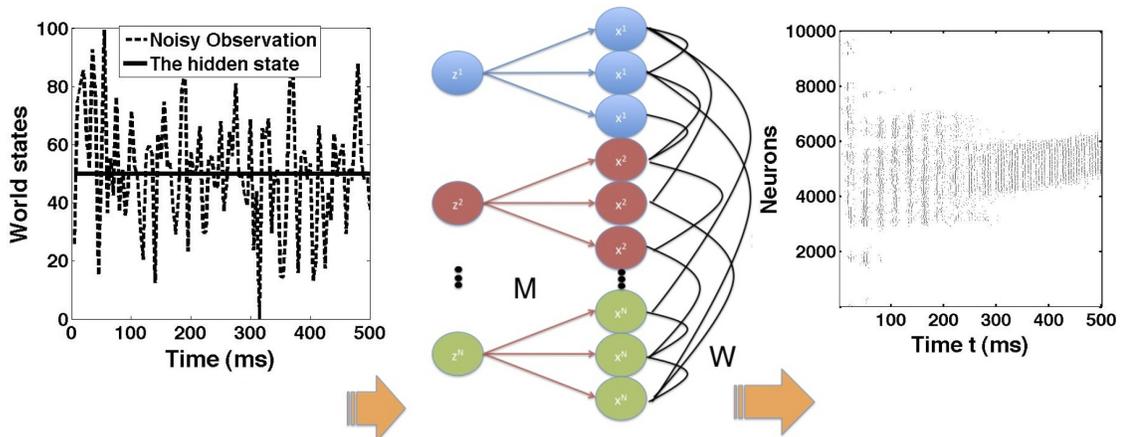


Figure 1: **Spiking Network Model for Sequential Monte Carlo Bayesian Inference.**

In this simple example illustrating the model, the input spikes to the two-layer network provide noisy and ambiguous observations while the output spikes of the inference neurons represent a Monte Carlo approximation to the posterior distribution of hidden state. There are 100 hidden states and the network utilizes 10,000 neurons for the Monte Carlo approximation, with each state preferred by a sub-population of 100 neurons.

to. The hidden world state  $X_k$  could correspond to an attribute of the real world, such as the mean light intensity of the visual stimulus, or the location of a rat in a maze. Animals are interested in estimating  $X_k$  by constructing its probability density function (pdf), also called the “belief state,” based only on noisy measurements or observations  $\{Z_k\}$ . The  $\{Z_k\}$  are assumed to be conditional independent given  $\{X_k\}$  and are governed by a likelihood function  $g$ :

$$Z_k | (X_k = x) \sim g(z|x), \quad z \in \mathbb{Z}. \quad (2)$$

It is not necessary for the animal to remember the complete history of observations  $\{Z_k\}$  to calculate the belief state. Instead, the belief state can be updated sequentially every time the sensory organs receive a new measurement. This procedure is called “filtering” in the engineering literature. From a Bayesian perspective, filtering corresponds to recursively calculating the belief state of  $X_k$  given the observations  $Z_{1:k}$  up to time  $k$ . When both  $f(x|x')$  and  $g(z|x)$  are given, the posterior pdf  $P(X_k|Z_{1:k})$  may be obtained recursively in two steps: a prediction step (Equation 3) and a measurement update (or

correction) step (Equation 4):

$$P(X_{k+1}|Z_{1:k}) = \int f(X_{k+1}|X_k)P(X_k|Z_{1:k})dx_k, \quad (3)$$

$$P(X_{k+1}|Z_{1:k+1}) = g(z_{k+1}|x_{k+1})P(X_{k+1}|Z_{1:k})/P(Z_{k+1}|Z_{1:k}). \quad (4)$$

The prediction equation 3 uses the previous belief state  $P(X_k|Z_{1:k})$  from time step  $k$  to produce a prior distribution of the state at time  $k + 1$ . When the new measurement  $Z_{k+1}$  becomes available, the update equation 4 modifies this prior density via Bayes' rule to obtain the posterior distribution  $P(X_{k+1}|Z_{1:k+1})$ . This process is repeated for each time step.

The two recursive equations above are the foundation for any exact or approximate solution to Bayesian filtering, including well-known examples such as Kalman filtering in the case of a linear Gaussian model and extended Kalman filtering for nonlinear Gaussian models. However, for most non-linear non-Gaussian models, closed-form solutions to equations 3 and 4 may be hard or impossible to compute. In such cases, numerical methods can be used to approximate the optimal Bayesian solution. For example, one can divide the state space  $\mathbb{X}$  into  $\mathcal{X}$  bins, each of which is centered at one of  $\{x^i \in \mathbb{X}, i = 1, \dots, \mathcal{X}\}$ . Such a grid-based method can be used to approximate the posterior density. Suppose the belief state at time  $k$  is given by:

$$P(X_k|Z_{1:k}) \simeq \sum_{i=1}^{\mathcal{X}} \omega_{k|k}^i \delta(X_k - x^i), \quad (5)$$

$$\begin{aligned} \omega_{k|k}^i &:= \int_{(x^{i-1}+x^i)/2}^{(x^{i+1}+x^i)/2} P(X_k|Z_{1:k})dx \\ &\simeq P(X_k = x^i|Z_{1:k}). \end{aligned} \quad (6)$$

where  $\delta(i)$  is the Kronecker delta function. The weights  $\omega_{k|k}^i$  denote the conditional probability of  $X_k$  at the  $i$ -th bin given observations up to time step  $k$ . If the grid is sufficient dense and the state space is continuous (or the state space is discrete and finite), one can compute these weights  $\{\omega_{k|k}^i\}$  at the center of each bin. Similarly, equations 3 and 4 can be re-written as

$$P(X_{k+1}|Z_{1:k}) \simeq \sum_{i=1}^{\mathcal{X}} \omega_{k+1|k}^i \delta(X_{k+1} - x^i), \quad (7)$$

$$P(X_{k+1}|Z_{1:k+1}) \simeq \sum_{i=1}^{\mathcal{X}} \omega_{k+1|k+1}^i \delta(X_{k+1} - x^i). \quad (8)$$

where

$$\omega_{k+1|k}^i = \sum_{j=1}^{\mathcal{X}} \omega_{k|k}^j f(x^i|x^j) \quad (9)$$

$$\omega_{k+1|k+1}^i = \frac{\omega_{k+1|k}^i g(Z_{k+1}|x^i)}{\sum_{j=1}^{\mathcal{X}} \omega_{k+1|k}^j g(Z_{k+1}|x^j)} \quad (10)$$

If the number of bins  $\mathcal{X}$  is large enough and both the dynamics model  $f$  and likelihood model  $g$  are known, the above equations provide the optimal Bayesian solution to equations 3 and 4.

## 2.2 Network architecture

We now show that the framework of grid-based filtering can be implemented in a two-layer spiking neural network as shown in Figure 1. Let  $\mathbf{s}_k$  denote the binary vector of activities at time  $k$  in the hidden-layer inference neurons. The following equation defines the dynamics of the network:

$$\mathbf{s}_k = \Phi(\mathbf{a}_k, \mathbf{b}_k) \quad (11)$$

where  $\Phi$  is the neuron’s response function,  $\mathbf{a}_k$  is the vector representing the inference neurons’ recurrent inputs, which are determined by the recurrent weight matrix  $W$  and  $\mathbf{s}_{k-1}$  from the previous time step, and  $\mathbf{b}_k$  is the vector representing feedforward inputs, which are determined by the feedforward weight matrix  $M$  and sensory measurement  $Z_k$ . How can Bayesian inference be achieved using the above dynamics? We approach this problem by first showing how this neural network can represent probability distributions.

### 2.2.1 Neural representation of probability distributions

Similar to the idea of grid-based filtering, we first divide the inference neuron population into  $\mathcal{X}$  sub-populations.  $\mathbf{s} = \{s_l^i, i = 1, \dots, \mathcal{X}, l = 1, \dots, \mathcal{L}\}$ .  $s_l^i(k) = 1$  if there is a spike in the  $l$ -th neuron of the  $i$ -th sub-population at time step  $k$ .  $s_l^i(k) = 0$  otherwise. Each sub-population of  $\mathcal{L}$  neurons share the same preferred world state, there being  $\mathcal{X}$  such sub-populations representing each of  $\mathcal{X}$  preferred states. One can, for example, view a neuron sub-population as a cortical column, within which neurons encode similar features (Ecker et al. 2010). A spike, generated by a neuron whose preferred world

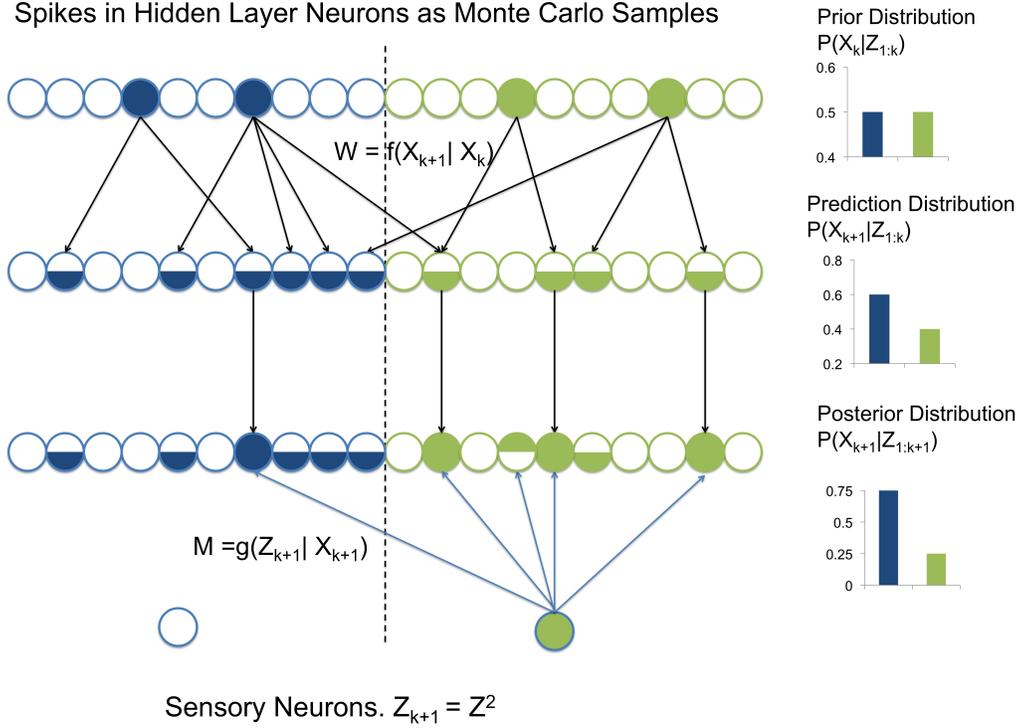


Figure 2: **Graphical Representation of spike distribution propagation.** Here,  $\mathcal{X} = \mathcal{Z} = 2$  and  $\mathcal{L} = 10$ . At time  $k$ , spikes (shown as filled circles in the top row) in the posterior population represent the distribution  $P(X_k|Z_{1:k})$ . With recurrent weights  $W \propto f(X_{k+1}|X_k)$ , spiking neurons send EPSPs to their neighbors and make them partially activated (shown as half-filled circles in the second row). The distribution of partially activated neurons is a Monte-Carlo approximation to the prediction distribution  $P(X_{k+1}|Z_{1:k})$ . When a new observation  $Z_{k+1}$  arrives, sensory input neurons send feedforward EPSPs to the inference neurons using synaptic weights  $M = g(Z|X)$ . The inference neurons at time  $k + 1$  fire only if they receive both recurrent and feedforward inputs. With the firing probability proportional to the product of prediction probability  $P(X_{k+1}|Z_{1:k})$  and observation likelihood  $g(Z_{k+1}|X_{k+1})$ , the spike distribution at time  $k + 1$  again represents the updated posterior  $P(X_{k+1}|Z_{1:k+1})$ .

state is  $x^i$  at time  $k$ , represents an independent Monte Carlo sample (particle) from the posterior probability  $P(X_k = x^i | Z_k)$ . Neural variability can thus be interpreted as arising naturally due to sampling (Hoyer et al. 2002). As depicted in the raster plot of Figure 1, the distribution of spikes across the entire inference layer population is a Monte-Carlo approximation to the current posterior distribution:

$$n_{k|k}^i := \sum_{l=1}^{\mathcal{L}} s_l^i(k) \propto \omega_{k|k}^i \quad (12)$$

$$N_k = \sum_{i=1}^{\mathcal{X}} n_{k|k}^i \quad (13)$$

where  $n_{k|k}^i$  is the number of spiking neurons in the  $i$ th sub-population at time  $k$ , which can also be regarded as the instantaneous firing rate for sub-population  $i$ .  $N_k$  is the total spike count in the inference layer population. The set  $\{n_{k|k}^i\}$  represents the unnormalized conditional probabilities of  $X_k$ , so that  $P(X_k = x^i | Z_{1:k}) = \omega_{k|k}^i = n_{k|k}^i / N_k$ .

With the above neural representation of probability distributions, we next show how suitable neural dynamics and synaptic weights can be chosen such that the spike distribution  $n_{k|k}^i / N_k$  will propagate according to equation 10, as illustrated in the example in figure 2. We tackle the problem of learning the synaptic weights in a later section.

### 2.2.2 Bayesian inference with stochastic synaptic transmission

To implement the prediction equation 3 in a spiking network, we require that the recurrent weights between the inference neurons encode the transition probabilities:  $W_{ij} = f(x^j | x^i) / C_W$ , where  $C_W$  is a scaling constant. We define the recurrent weight  $W_{ij}$  to be the synaptic release probability between  $i$ -th neuron sub-population and  $j$ -th neuron sub-population in the inference layer. Each neuron that spikes at time step  $k$  will randomly evoke, with probability  $W_{ij}$ , one recurrent excitatory post-synaptic potential (EPSP) at time step  $k + 1$ , after some network delay. We define the number of recurrent EPSPs received by neuron  $l$  in the  $j$ -th sub-population as  $a_l^j$ . Thus,  $a_l^j$  is the sum of  $N_k$  independent (but not identically distributed) Bernoulli trials:

$$a_l^j(k+1) = \sum_{i=1}^{\mathcal{X}} \sum_{l'=1}^{\mathcal{L}} \epsilon_{l'}^i s_{l'}^i(k), \quad \forall l = 1 \dots \mathcal{L}. \quad (14)$$

where  $P(\epsilon_l^i = 1) = W_{ij}$  and  $P(\epsilon_l^i = 0) = 1 - W_{ij}$ . The sum  $a_l^j$  follows the so-called ‘‘Poisson binomial’’ distribution (Hodges & Cam 1960) and in the limit approaches the

Poisson distribution:

$$P(a_l^j(k+1) \geq 1) \simeq 1 - \exp\left(-\sum_i W_{ij} n_{k|k}^i\right) \quad (15)$$

$$\simeq \sum_i W_{ij} n_{k|k}^i = \frac{N_k}{C_W} \omega_{k+1|k}^i \quad (16)$$

where the absolute difference between the two sides of equation 15 is bounded by  $3\sqrt[3]{\frac{\max_i f_{ij}}{C_W}}$ . Higher order terms involving  $\frac{N_k}{C_W}$  are discarded in the approximation of equation 16. Detailed analysis of the distribution of  $a_l^j$  is provided in appendix A.

Let  $n_{k+1|k}^j$  be the number of neurons in  $j$ -th sub-population receiving one or more recurrent EPSPs. Then, we have

$$\begin{aligned} E[n_{k+1|k}^j | \{n_{k|k}^i\}] &= \mathcal{L} \sum_{i=1}^x W_{ij} n_{k|k}^i \\ &= \mathcal{L} \frac{N_k}{C_W} \omega_{k+1|k}^i \end{aligned} \quad (17)$$

$$\text{Var}[n_{k+1|k}^j | \{n_{k|k}^i\}] \simeq \mathcal{L} \frac{N_k}{C_W} \omega_{k+1|k}^i \quad (18)$$

Thus, the prediction probability in equation 7 is represented by the expected number of neurons that receive recurrent inputs, as shown in figure 2.

In the model, recurrent inputs alone are not strong enough to make the inference neurons fire – these inputs leave the neurons partially activated. We can view these partially activated neurons as the “proposed” samples drawn from the prediction density  $P(X_{k+1}|X_k)$ . To correct the prediction distribution based on the current observation, these proposed samples are accepted with a probability proportional to the observation likelihood  $P(Z_{k+1}|X_{k+1})$  when the new measurement  $Z_{k+1}$  becomes available. This implements a form of “rejection sampling” used in sequential Monte Carlo algorithms (Doucet et al. 2001). Neurally, this is implemented by feedforward inputs from sensory neurons (which receive  $Z_{k+1}$ ) causing neurons to spike when coincident with recurrent inputs. Thus, the inference neurons act as coincidence detectors which fire if and only if both recurrent and sensory inputs are received:

$$s_l^j(k+1) = \text{sgn}(a_l^j(k+1) \times b_l^j(k+1)) \quad (19)$$

where the sign function  $\text{sgn}(x) = 1$  only when  $x > 0$ . The feedforward input  $b_l^j$  represents the number of EPSPs caused by sensory inputs. Equation 19 defines the

output of an abstract model neuron. In section 4 we show that such abstract model neurons can be implemented using leaky-integrate-and-fire (LIF) dynamics.

Note that  $P(s_l^j(k+1) = 1) \propto P(X_{k+1} = x^j | Z_{1:k+1})$  if and only if  $P(b_l^j(k+1) = 1) \propto g(Z_{k+1} | X_{k+1} = x^j)$ . In other words, the hidden-layer inference neurons will spike with probability proportional to the updated posterior distribution if and only if the feedforward input  $\{b_l^j\}$  arrives with probability proportional to the likelihood of observations. We now examine what feedforward weight matrix  $M$  between the sensory neurons and inference neurons achieves such a requirement.

The noisy measurement  $Z_{k+1}$  is not directly observed by the inference neurons, but sensed through an array of  $\mathcal{Z}$  sensory neurons, whose receptive fields are centered at  $z^i \in \mathbb{Z}, i = 1, \dots, \mathcal{Z}$ . We assume for simplicity that receptive fields of sensory neurons do not overlap with each other (appendix B discusses the more general overlapping case). Again we define the feedforward weight  $M_{ij}$  to be the synaptic release probability between sensory neuron  $i$  and inference neurons in the  $j$ -th sub-population. A spiking sensory neuron  $i$  causes an EPSP in a neuron in the  $j$ -th sub-population with probability  $M_{ij}$ . When  $Z_{k+1} = z^i$  arrives, the sensory neuron centered at  $z^i$  emits a spike at time  $k$ , causing a feedforward EPSP in each of its post-synaptic neurons with probability proportional to the likelihood:

$$P(b_l^i(k+1) = 1) = g(Z_{k+1} | x^i) / C_M \quad (20)$$

where  $C_M$  is a scaling constant such that  $M_{ij} = g(Z_{k+1} = z^i | x^j) / C_M$ .

Finally, an inference neuron fires a spike at time  $k+1$  if and only if it receives both recurrent and sensory inputs. The corresponding firing probability is then the product of the probabilities of the two inputs:

$$\begin{aligned} P(s_l^i(k+1) = 1) &= P(a_l^i(k+1) \geq 1) P(b_l^i(k+1) \geq 1) \\ &= \frac{N_k}{C_W C_M} P(X_{k+1} | Z_{1:k}) g(Z_{k+1} | X_{k+1}) \\ &\propto P(X_{k+1} | Z_{1:k+1}) \end{aligned} \quad (21)$$

Let  $n_{k+1|k+1}^i$  be the number of spikes in  $i$ -th sub-population at time  $k + 1$ ,

$$n_{k+1|k+1}^i = \sum_{l=1}^{\mathcal{L}} s_l^i(k+1) \quad (22)$$

$$\begin{aligned} E[n_{k+1|k+1}^i | \{n_{k|k}^i\}] &= \mathcal{L} \frac{N_k}{C_W C_M} g(Z_{k+1}|x^i) \omega_{k+1|k}^i \\ &= \mathcal{L} \frac{N_k}{C_W C_M} P(Z_{k+1}|Z_{1:k}) \omega_{k+1|k+1}^i \end{aligned} \quad (23)$$

$$\begin{aligned} \text{Var}[n_{k+1|k+1}^i | \{n_{k|k}^i\}] &= \sum_{l=1}^{\mathcal{L}} [\text{Var}(a_l^i) \text{Var}(b_l^i) + \text{Var}(a_l^i) E(b_l^i)^2 + \text{Var}(b_l^i) E(a_l^i)^2] \\ &\simeq \mathcal{L} \frac{N_k}{C_W C_M} g(Z_{k+1}|x^i) \omega_{k+1|k}^i \end{aligned} \quad (24)$$

Equation 23 ensures that the expected spike distribution at time  $k + 1$  is a Monte Carlo approximation to the updated posterior probability  $P(X_{k+1}|Z_{1:k+1})$ . It also determines how many neurons are activated at time  $k + 1$ . To keep the number of spikes at different time steps relatively constant, the scaling constant  $C_W$  and the number of neurons  $\mathcal{L}$  could be of the same order of magnitude: for example,  $C_W = \mathcal{L}$ . Note that approximations in equations 16, 18 and 24 become exact when  $\frac{N_k}{C_W} \rightarrow 0$ . This implies a form of sparse coding: although the number of neurons in the network may be large, only a small fraction of neurons are activated.

### 2.3 Poisson variability and convergence results

In this section, we briefly discuss some convergence results for Bayesian filtering using the proposed spiking network. Equations 23 and 24 imply that the spike distribution  $\{n_{k|k}^j\}$  is approximately proportional to the true pdf  $P(X_k = x^j | Z_{1:k})$  at time  $k$ . Suppose the true distribution is known only at initial time 1:  $\hat{P}_1^i = \omega_{1|1}^i$ . We would like to investigate how the mean and variance of  $\hat{P}_k^i$  vary over time. To simplify the analysis, we let  $\mathcal{L} = C_W$  and  $C_M = P(Z_{k+1}|Z_{1:k})$  so that

$$\begin{aligned} E[N_{k+1} | \{n_{k|k}^j\}] &= \text{Var}[N_{k+1} | \{n_{k|k}^j\}] \\ &= \frac{\mathcal{L}}{C_W C_M} P(Z_{k+1}|Z_{1:k}) E[N_k] = E[N_k] = N_1. \end{aligned} \quad (25)$$

where  $N_1$  is the initial spike count. Given the previous distribution  $\{\hat{P}_k^i\}$ , equations 23 and 24 can be rewritten as:

$$E[\hat{P}_{k+1}^j | \{\hat{P}_k^j\}] = \frac{g(Z_{k+1}|x^j)}{P(Z_{k+1}|Z_{1:k})} \sum_{i=1}^{\mathcal{X}} f(x^j|x^i) \hat{P}_k^i \quad (26)$$

$$\begin{aligned} \text{Var}[\hat{P}_{k+1}^j | \{\hat{P}_k^j\}] &= \frac{\text{Var}[n_{k+1|k+1}^i | \{n_{k|k}^i\}] - N_k E^2[\hat{P}_{k+1}^j | \{\hat{P}_k^j\}]}{N_k^2 + N_k} \\ &\simeq \frac{1}{N_k} (E[\hat{P}_{k+1}^j | \{\hat{P}_k^j\}] - E^2[\hat{P}_{k+1}^j | \{\hat{P}_k^j\}]) \end{aligned} \quad (27)$$

The approximation holds when  $N_k$  is large so that  $N_k \simeq N_k + 1$ . Marginalizing over  $\{\hat{P}_k^j\}$  we obtain the recursive update equation for  $E[\hat{P}_{k+1}^j]$  and  $\text{Var}[\hat{P}_{k+1}^j]$  using the law of total expectation and the law of total variance:

$$\begin{aligned} E[\hat{P}_{k+1}^j] &= \frac{g(Z_{k+1}|x^j)}{P(Z_{k+1}|Z_{1:k})} \sum_{i=1}^{\mathcal{X}} f(x^j|x^i) E[\hat{P}_k^i] \quad (28) \\ \text{Var}[\hat{P}_{k+1}^j] &= E[\text{Var}[\hat{P}_{k+1}^j | \{\hat{P}_k^j\}]] + \text{Var}[E[\hat{P}_{k+1}^j | \{\hat{P}_k^j\}]] \\ &= \frac{E[\hat{P}_{k+1}^j] - E^2[\hat{P}_{k+1}^j]}{N_1} + \frac{g^2(Z_{k+1}|x^j)}{P^2(Z_{k+1}|Z_{1:k})} \times \text{Var}\left[\sum_{i=1}^{\mathcal{X}} f(x^j|x^i) \hat{P}_k^i\right] \\ &\simeq \frac{E[\hat{P}_{k+1}^j] - E^2[\hat{P}_{k+1}^j]}{N_1} + \eta_k^j \text{Var}\left[\sum_{i=1}^{\mathcal{X}} f(x^j|x^i) \hat{P}_k^i\right] \end{aligned} \quad (29)$$

where  $\eta_k^j = g^2(Z_{k+1}|x^j)/P^2(Z_{k+1}|Z_{1:k})$ . The variance  $\text{Var}[\hat{P}_{k+1}^j]$  can be partitioned into two parts. The first part represents the variance from current time step. The second part represents the variance from previous time step, but weighted by the coefficient  $\eta_k^j$ .

Since the initial distribution  $\omega_1^j$  is known, the solution to equation 28 is easy to obtain:

$$E[\hat{P}_k^j] = \omega_{k|k}^j \quad (30)$$

Thus,  $\hat{P}_k^j$  is an unbiased estimator of true posterior probability  $\omega_{k|k}^j$ . However, the closed-form solution for the variance update equation 29 is generally intractable, except for some special forms of  $f$ . For example, consider a uniform transition model where

$f(x^j|x^i) = 1/\mathcal{X}$ . Since the  $\{\hat{P}_k^i\}$  are negatively correlated, we have:

$$\begin{aligned} \text{Var}[\hat{P}_2^j] &= \frac{1}{N_1}(E[\hat{P}_2^j] - E^2[\hat{P}_2^j]) + 0 \\ \text{Var}[\hat{P}_3^j] &\leq \frac{1}{N_1}\{(E[\hat{P}_3^j] - E^2[\hat{P}_3^j]) + \frac{\eta_2^j}{\mathcal{X}^2}(1 - \sum_i E^2[\hat{P}_2^i])\} \\ &\dots \\ \text{Var}[\hat{P}_k^j] &\leq \frac{1}{N_1}\{(E[\hat{P}_k^j] - E^2[\hat{P}_k^j]) + \frac{\eta_{k-1}^j}{\mathcal{X}^2}(1 - \sum_i E^2[\hat{P}_{k-1}^i]) + O(\frac{1}{\mathcal{X}^4})\} \end{aligned} \quad (31)$$

Equation 31 has several implications. First, when the state space  $\mathbb{X}$  is large, we can ignore the higher order terms of  $\mathcal{X}$ . The variance of the estimator becomes:

$$\text{Var}[\hat{P}_k^j] \approx \frac{1}{N_1}(E[\hat{P}_k^j] - E^2[\hat{P}_k^j]) \quad (32)$$

Equivalently, from equations 25 and 32 the variance of the spike count  $n_{k|k}^j = N_k \hat{P}_k^j$  can be shown to have the following form:

$$\text{Var}[n_{k|k}^j] = N_1 E[\hat{P}_k^j] + E[\hat{P}_k^j] - E^2[\hat{P}_k^j] \simeq E[n_{k|k}^j] \quad (33)$$

The variance of neural response is roughly proportional to the mean when the transition model is uniform. This is consistent with experimental results showing that spike count variances grow in proportion to spike count means (Dean 1981, Tolhurst et al. 1983). Thus, rather than representing noise, Poisson variability in the model occurs as a natural consequence of sampling and sparse coding (equation 16). Second, the variance  $\text{Var}[\hat{P}_k^j] \propto 1/N_1$ . Therefore  $\text{Var}[\hat{P}_k^j] \rightarrow 0$  as  $N_1 \rightarrow \infty$ , showing that  $\hat{P}_k^j$  is a consistent estimator of  $\omega_{k|k}^j$ .

In general, when the transition model is arbitrary, numerical methods are needed to study the relationship between the variance of  $\hat{P}_k^j$  and  $k$ . In Figure 3, we test whether the above two implications still hold for random transition models. The state space is finite,  $\mathbb{X} = \{1, 2, \dots, 20\}$ ,  $Z_k \sim N(X_k, 5)$ ,  $\mathcal{L} = C_W = 25 \times N_1$ . Ideally,  $C_M(k) = P(Z_{k+1}|Z_{1:k}) = \sum_j g(Z_{k+1}|x^j)P(x^j|Z_{1:k})$ . However, animals may not be able to calculate  $P(Z_{k+1}|Z_{1:k})$ . A biologically plausible alternative is to utilize inhibition in the network to determine  $C_W$ . For example,  $C_W$  could be made time dependent, e.g.,  $C_W(k+1) = 10 * N_k/N_1$ , resulting in a form of divisive inhibition (Chance & Abbott 2000). If the overall neural activity is weak at time  $k$ , then the global inhibition regulating  $M$  is decreased to allow more spikes at time  $k+1$ .

For the experiments, elements in the transition matrix  $f(x^j|x^i)$  were first uniformly drawn from  $[0, 1]$ , and then normalized to ensure  $\sum_j f(x^j|x^i) = 1$ . In Figure 3(a-c), we examine equation 32 for different initial spike count values:  $N_1 = 10^2, 10^3$  and  $10^4$ . Each data point represents  $\text{Var}[\hat{P}_k^j]$  along the vertical axis and  $E[\hat{P}_k^j] - E^2[\hat{P}_k^j]$  along the horizontal axis, calculated over 100 trials with the same random transition matrix  $f$ , and  $k = 1, \dots, 10, j = 1, \dots, 20$ . The solid lines represent a least squares power law fit to the data:  $\text{Var}[\hat{P}_k^j] = C_V * (E[\hat{P}_k^j] - E^2[\hat{P}_k^j])^{C_E}$ . For 100 different random transition matrices  $f$ , the means of the exponential term  $C_E$  were 1.2863, 1.13, and 1.037, with standard deviations 0.13, 0.08, and 0.03 respectively, for  $N_1 = 100$  and  $\mathcal{X} = 4, 20$ , and 100. The mean of  $C_E$  continues to approach 1 when  $\mathcal{X}$  is increased, as shown in figure 3(d). Since  $\text{Var}[\hat{P}_k^j] \propto (E[\hat{P}_k^j] - E^2[\hat{P}_k^j])$  implies  $\text{Var}[n_{k|k}^j] \propto E[n_{k|k}^j]$ , these results suggest that arbitrary transition models still preserve the Poisson variability.

The term  $C_V$  represents the scaling constant for the variance. Figure 3(e) shows that the mean of  $C_V$  over 100 different transition matrices  $f$  (over 100 different trials with the same  $f$ ) is inversely proportional to initial spike count  $N_1$ , with power law fit  $C_V = 1.77N_1^{-0.9245}$ . This indicates that the relation  $\text{Var}[\hat{P}_k^j] \propto 1/N_1$  (equation 31) still approximately holds no matter what the dynamics model  $f$  is.

The bias between estimated and true posterior probability can be calculated as:  $\text{bias}(f) = \frac{1}{\mathcal{X}\mathcal{K}} \sum_{i=1}^{\mathcal{X}} \sum_{k=1}^{\mathcal{K}} (E[\hat{P}_k^i] - \omega_{k|k}^i)^2$ . The relationship between the mean of the bias (over 100 different  $f$ ) versus initial count  $N_1$  is shown in figure 3(f). Since the precision of the estimator  $\hat{P}_k^j$  is limited by  $N_1$ , we also have an inverse proportionality between bias and  $N_1$ . Therefore, as the figure shows, for arbitrary  $f$ , the estimator  $\hat{P}_k^j$  remains a consistent estimator of  $\omega_{k|k}^j$ .

In summary, we have proposed a spiking network model that approximates Bayesian filtering using spikes as Monte Carlo samples of probability distributions. We assume that the transition (dynamics) and emission (observation) models are known and encoded in the recurrent weights  $W$  and feedforward weights  $M$ , respectively. In addition, we assume that the network employs a sparse coding strategy: the total neuronal activity  $N_k$  at any time step is small compared to the number of neurons  $\mathcal{L}$  in a sub-population. The model does not put any constraints on the particular form of the probability density over hidden world state. When the state space  $\mathbb{X}$  is discrete, the spiking network provides the optimal Bayesian solution when the above assumptions hold.

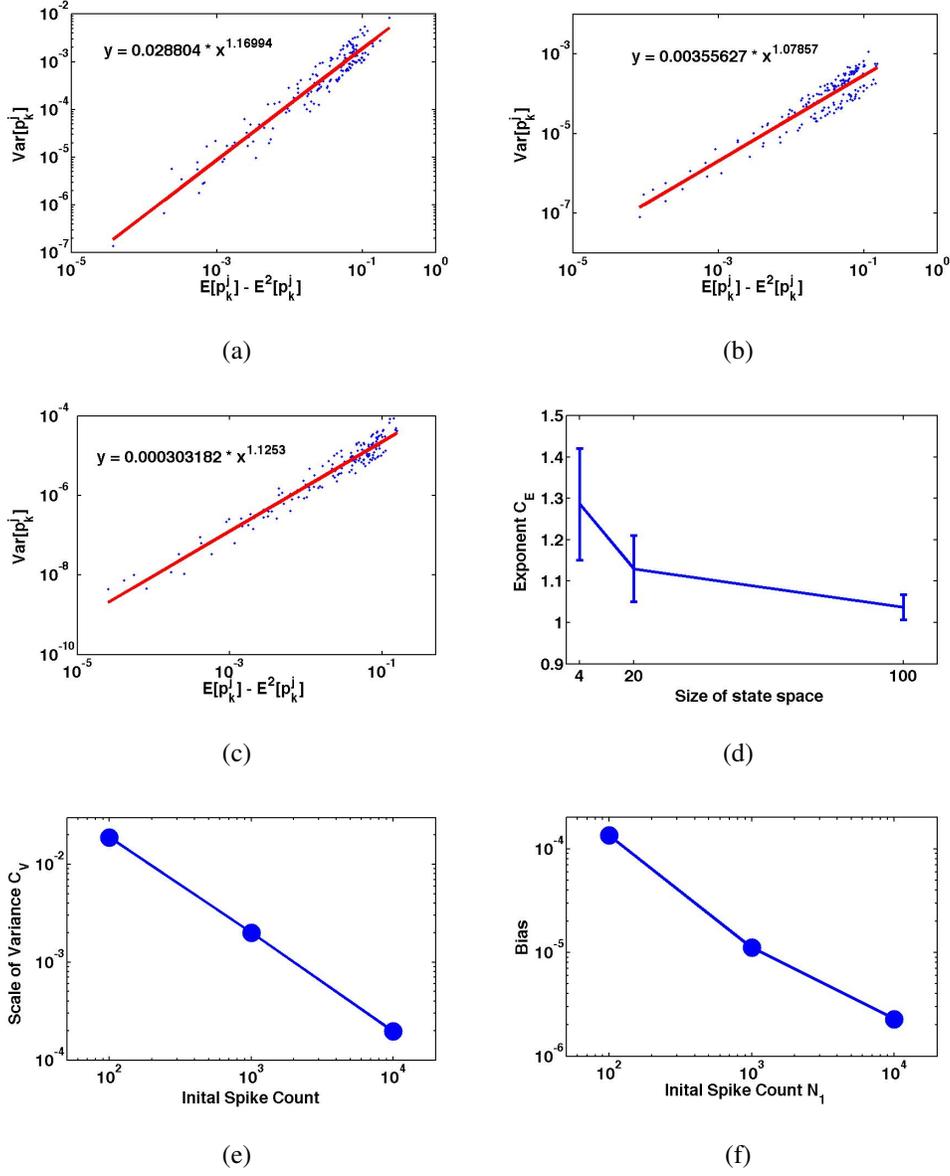


Figure 3: **Variance versus mean of estimator for different initial spike counts.** (a)  $N_1 = 100$ , (b)  $N_1 = 1000$ , (c)  $N_1 = 10,000$ . Each data point represents the variance of the estimator  $\hat{P}_k^i$  (vertical axis) and “mean”  $E[\hat{P}_k^i] - E^2[\hat{P}_k^i]$  (horizontal axis) over 100 different trials with the same transition matrix  $f$ , for  $i = 1, \dots, 20$  and  $k = 2, \dots, 10$ . The solid lines are least-square power law fits  $\text{Var}[\hat{P}_k^i] = C_V * (E[\hat{P}_k^i] - E^2[\hat{P}_k^i])^{C_E}$  to different data sets, with coefficients  $(C_V, C_E)$  shown in the legend. (d) The mean of the exponential term  $C_E$  over 100 different transition matrices  $f$  approaches 1 as  $\mathcal{X}$  increases. (e) & (f) The mean of  $C_V$  decreases as  $N_1$  increases, as does the bias between the mean of the estimator and true posterior probability  $\frac{1}{\mathcal{K}\mathcal{X}} \sum_{i,k} (E[\hat{P}_k^i] - \omega_{k|k}^i)^2$ .

### 3 On-line parameter learning

In the previous section, we assumed that the model parameters, i.e., the transition probabilities  $f(X_{k+1}|X_k)$  and the emission probabilities  $h(Z_k|X_k)$ , are known. In this section, we describe how these parameters  $\theta = \{f, g\}$  can be learned from noisy observations  $\{Z_k\}$ . Traditional methods to estimate model parameters are based on the Expectation-Maximization (EM) algorithm (Dempster et al. 1977), which maximizes the (log) likelihood of the unknown parameters  $\log P_\theta(Z_{1:k})$  given a set of observations collected previously. However, such an “off-line” approach is biologically implausible because (1) it requires animals to store all of the observations before learning, and (2) evolutionary pressures dictate that animals update their belief over  $\theta$  sequentially any time a new measurement becomes available.

We therefore propose an on-line estimation method where observations are used for updating parameters as they become available and then discarded. Our approach is based on recursively calculating the sufficient statistics of  $\theta$  using stochastic approximation algorithms and the Monte Carlo method. We explore how animals can implement this on-line learning algorithm in a spiking network, where changes in synaptic weights are subject to Hebbian learning rules.

#### 3.1 General framework

In this section, we describe a general framework for on-line learning of parameters in non-linear non-Gaussian state space models, following (Andrieu et al. 2005, Cappe & Moulines 2009). Again,  $\{X_k, k \in \mathbb{N}\}$  is a hidden Markov process, with the additional assumption that it is stationary and ergodic: as before,  $X_{k+1}|X_k \sim f_\theta(X_{k+1}|X_k)$  and  $\{Z_k\}$  are the observations with emission probabilities  $Z_k|X_k \sim g_\theta(Z_k|X_k)$ . We would like to find the parameters  $\theta$  that maximize the log likelihood:  $\log P_\theta(Z_{1:k}) = \sum_{t=1}^k \log P_\theta(Z_t|Z_{t-1})$ .

We first show how the traditional off-line EM algorithm (Dempster et al. 1977) accomplishes this goal in an iterative manner. In iteration  $k$ , the EM algorithm approximates the joint log-likelihood  $\log P_\theta(X, Z)$  using the *expected* value over the hidden data  $X$  based on the current estimate of parameters  $\theta_k$  (E-step):

$$Q(\theta, \theta_k) = E_{\theta_k}[\log P_\theta(X, Z)|Z] \quad (34)$$

Then the value of  $\theta$  that *maximizes*  $Q(\theta, \theta_k)$  is found in the M-step. This gives rise to the new estimate:

$$\theta_{k+1} = \arg \max Q(\theta, \theta_k). \quad (35)$$

Maximizing  $Q(\theta, \theta_k)$  is equivalent to increasing the marginalized log likelihood  $\log P_\theta(Z_{1:k})$ , since their gradient terms coincide (Dempster et al. 1977):

$$E_\theta \nabla_\theta \log P_\theta(X, Z)|Z = \nabla_\theta \log P_\theta(Z), \quad (36)$$

Starting from an initial guess  $\theta_0$ , the EM algorithm generates a sequence of estimates  $\{\theta_k\}$ , which converge to the true parameter  $\theta^*$  under some regularity conditions (Wu 1983).

To perform on-line parameter estimation, we aim to produce a new estimate  $\theta_{k+1}$  when the observation  $Z_k$  becomes available, where  $\theta_{k+1}$  maximizes the function

$$\begin{aligned} Q(\theta, \theta_k) &= E_{\theta_k}[\log P_\theta(X_{1:k}, Z_{1:k})|Z_{1:k}] \\ &= E_{\theta_k}[\sum_{t=1}^k \log P_\theta(X_t, Z_t|X_{t-1})|Z_{1:k}] \\ &= E_{\theta_k}[\sum_{t=1}^k \log(f_\theta(X_t|X_{t-1})g_\theta(Z_t|X_t))|Z_{1:k}] \end{aligned} \quad (37)$$

In general,  $Q(\theta, \theta_k)$  and its derivative  $\nabla_\theta Q(\theta, \theta_k)$  are difficult to estimate because they are functions of the complete data  $\{X_{1:k}, Z_{1:k}\}$ . Equation 37 is only of theoretical interest unless the unknown parameter  $\theta$  can be estimated, without any loss of information, from a function of the complete data that has much lower dimension, the so-called sufficient statistic for  $\theta$ . As an example, suppose the likelihood  $P_\theta(X_t, Z_t|X_{t-1}) = f_\theta(X_t|X_{t-1})g_\theta(Z_t|X_t)$  belongs to an exponential family (Casella & Berger 2001):

$$P_\theta(X_t, Z_t|X_{t-1}) \propto P_\theta(T) = \exp[\psi(\theta) \cdot T(X_t, Z_t, X_{t-1}) - A(\theta)] \quad (38)$$

where  $T(X_t, Z_t, X_{t-1})$  is a complete sufficient statistic for parameter  $\theta$ , and  $\psi$  and  $A$  are arbitrary functions of  $\theta$ . All inference about  $\theta$  depends only on

$$\hat{T}(\theta_k) = k^{-1} E_{\theta_k}[\sum_{t=1}^k T(X_t, Z_t, X_{t-1})|Z_{1:k}], \quad (39)$$

which is the expected sufficient statistic of the joint distribution  $P(X_{1:k}, Z_{1:k})$ . The expectation  $E_{\theta_k}(\cdot|Z_{1:k})$  is taken with respect to the posterior distribution  $P_{\theta_k}(X_{1:k}|Z_{1:k})$  based on the current  $\theta_k$ .

An online EM algorithm can be obtained by approximating the expected sufficient statistic  $\hat{T}(\theta_k)$  using the stochastic approximation (or Robbins-Monro) procedure (Robbins & Monro 1951):

$$\hat{T}(\theta_k) \simeq \gamma_k E_{\theta_{k-1}}(T(X_{k-1}, Z_k, X_k)|Z_k) + (1 - \gamma_k)\hat{T}(\theta_{k-1}), \quad (40)$$

where the learning rate  $\gamma_k$  is a decreasing function of  $k$ . Equation 40 enables us to combine new observations  $Z_k$  with the previous estimate  $\hat{T}(\theta_{k-1})$  sequentially. When the learning rate is small  $\gamma_k \rightarrow 0$  such that  $\theta_k$  changes slowly, the approximation in equation 40 becomes exact. In general, convergence is guaranteed when  $\sum_{k=1}^{\infty} \gamma_k = \infty$  and  $\sum_{k=1}^{\infty} \gamma_k^2 < \infty$ . Note that if  $\gamma_k = 1/k$ ,  $\hat{T}_k$  is simply the running average of  $T$ .

In summary, the online EM algorithm based on the sufficient statistic can be rewritten as:

**E-step**  $\hat{T}(\theta_k) = \gamma_k E_{\theta_{k-1}}(T(X_{k-1}, Z_k, X_k)|Z_k) + (1 - \gamma_k)\hat{T}(\theta_{k-1})$

**M-step**  $\theta_{k+1} = \arg \max_{\theta} P_{\theta}(\hat{T}_k)$ , which is the unique solution to the equation

$$\nabla_{\theta} \psi(\theta) \cdot \hat{T}_k = \nabla_{\theta} A(\theta).$$

### 3.2 Learning transition and emission probabilities

For a discrete hidden Markov model, the unknown parameters  $\theta$  consist of the transition matrix  $f_{ij} = f(x^j|x^i)$  and the emission probability matrix  $g_{ij} = g(z^j|x^i)$ . Recall that for the spiking network in Section 2.2, we defined  $M^k$  and  $W^k$  as the feed-forward and recurrent weights respectively at time step  $k$ . In this section, we introduce Hebbian learning rules (based on equation 40) for the synaptic weights  $M^k$  and  $W^k$  such that  $M^k$  and  $W^k$  become consistent estimators of  $f$  and  $g$  respectively as  $k \rightarrow \infty$ .

Recall that the population of inference neurons in the model maintains a Monte-Carlo approximation of the posterior distribution  $P_{\theta_k}(X_k|Z_{1:k})$  over the hidden state  $X_k$ , given observations up to time  $k$ . However, the expectation  $E_{\theta_{k-1}}(T(X_{k-1}, Z_k, X_k)|Z_k)$  in equation 40 is taken with respect to the *smoothed* distribution

$$P_{\theta_k}(X_{k-1}, X_k|Z_{1:k}) = P_{\theta_k}(X_{k-1}|X_k, Z_{1:k})P(X_k|Z_{1:k}), \quad (41)$$

which is the product of the posterior distribution and the distribution of hidden state at the previous time step  $k-1$  given the observations  $Z_{1:k}$ . Such a retrospective distribution cannot be implemented in a two-layer spiking network such as the one described above. Therefore, we employ an approximation to equation 40:

$$\begin{aligned}\hat{T}(\theta_k) &\simeq \gamma_k \times \sum_{X_k, X_{k-1}} T(X_{k-1}, Z_k, X_k) P(X_k | Z_{1:k}, \theta_{k-1}) P(X_{k-1} | Z_{1:k-1}, \theta_{k-1}) \\ &\quad + (1 - \gamma_k) \times \hat{T}(\theta_{k-1}) \\ &\simeq \gamma_k \times \sum_{n=1}^{N_k} \sum_{n'=1}^{N_{k-1}} T(\hat{x}_{k-1}^{n'}, Z_k, \hat{x}_k^n) / (N_k \times N_{k-1}) + (1 - \gamma_k) \times \hat{T}(\theta_{k-1})\end{aligned}\quad (42)$$

where  $\{\hat{x}_k^n\}$  and  $\{\hat{x}_{k-1}^{n'}\}$  are Monte-Carlo samples drawn from posterior distributions  $P(X_{k-1} | Z_{1:k-1}, \theta_{k-1})$ ,  $P(X_k | Z_{1:k}, \theta_k)$ , respectively.

The sufficient statistic for  $g$  given the current estimator  $M^k = \hat{g}_k$  can be written as

$$T(X_k, Z_k | g) = \delta(X_k = x^j, Z_k = z^i) \quad (43)$$

$$\hat{T}(M^k) = \gamma_k E_{M^{k-1}}(\delta(X_k = x^j, Z_k = z^i) | Z_k) + (1 - \gamma_k) \hat{T}(M^{k-1}) \quad (44)$$

The expectation in the first term can be further approximated by Monte Carlo sampling of spikes:

$$E_{M^{k-1}}(\delta(X_t = x^j, Z_t = z^i) | Z_k) = \frac{n_{k|k}^j}{N_k} \times \frac{\tilde{n}^i(k)}{\sum_i \tilde{n}^i(k)}, \quad (45)$$

where  $n_{k|k}^j$  is the number of post-synaptic spikes in the  $j$ -th sub-population of inference neurons,  $N_k = \sum_j n_{k|k}^j$ , and  $\tilde{n}^i(k)$  is the number of pre-synaptic spikes in  $i$ -th sub-population of sensory neurons at time  $k$ .

The corresponding M-Step is given by:

$$\hat{g}_k = M^k = \frac{\hat{T}(M_{ij}^k)}{\sum_i \hat{T}(M_{ij}^k)} \quad (46)$$

Combining equations 44 and 46, we derive a local Hebbian learning rule for  $M^k$ :

$$\begin{aligned}M_{ij}^k &= \frac{\gamma_k n_{k|k}^j}{N_k} \times \frac{\tilde{n}^i(k)}{\sum_i \tilde{n}^i(k)} + \left(1 - \frac{\gamma_k n_{k|k}^j}{N_k}\right) \times M_{ij}^{k-1} \\ \frac{M_{ij}^k - M_{ij}^{k-1}}{\gamma_k^M} &= -M_{ij}^{k-1} + \frac{\tilde{n}^i(k)}{\sum_i \tilde{n}^i(k)}, \quad \text{when } n_{k|k}^j > 0,\end{aligned}\quad (47)$$

where the effective learning rate  $\gamma_k^M = \gamma_k \frac{n_{k|k}^j}{N_k}$  is proportional to the post-synaptic activity  $n_{k|k}^j$  in the inference layer population. A higher value of  $n_{k|k}^j / N_k$  represents a higher posterior belief for the world state  $X^j$ , resulting in faster learning.

Similarly, the transition probability matrix  $f$  can be learned by estimating its sufficient statistics:

$$T(X_k, Z_t, X_{k-1}|f_{ij}) = \delta(X_k = x^j, X_{k-1} = x^i) \quad (48)$$

Equation 40 can then be implemented as

$$\begin{aligned} \hat{T}(W^k) &= \gamma_k E_{W^{k-1}}(\delta(X_{k-1} = x^i, X_k = x^j)|Z_k) + (1 - \gamma_k)\hat{T}(W^{k-1}) \\ &= \gamma_k \times \frac{n_{k-1|k-1}^i}{N_{k-1}} \times \frac{n_{k|k}^j}{N_k} + (1 - \gamma_k) \times W_{ij}^{k-1} \end{aligned} \quad (49)$$

The corresponding M-step also has the form:

$$W_{ij}^k = \frac{\hat{T}(W_{ij}^k)}{\sum_{j=1}^{\mathcal{X}} T(W_{ij}^k)} \quad (50)$$

Combining equations 49 and 50 we derive a local Hebbian learning rule for  $M^k$ :

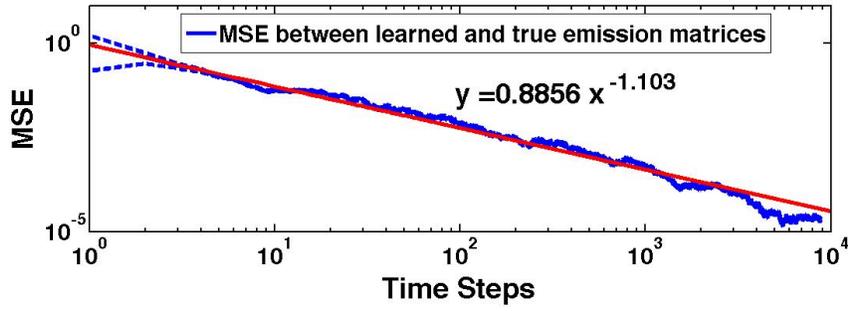
$$\begin{aligned} W_{ij}^k &= \gamma_k \frac{n_{k-1|k-1}^i}{N_{k-1}} \times \frac{n_{k|k}^j}{N_k} + (1 - \gamma_k \frac{n_{k-1|k-1}^i}{N_{k-1}}) \times W_{ij}^{k-1} \\ \frac{W_{ij}^k - W_{ij}^{k-1}}{\gamma_k^W} &= -W_{ij}^{k-1} + \frac{n_{k|k}^j}{N_k}, \quad \text{when } n_{k-1|k-1}^i > 0, \end{aligned} \quad (51)$$

where the effective learning rate  $\gamma_k^W = \gamma_k \frac{n_{k-1|k-1}^i}{N_{k-1}}$  is proportional to the pre-synaptic activity  $n_{k-1|k-1}^i$  in the inference layer population.

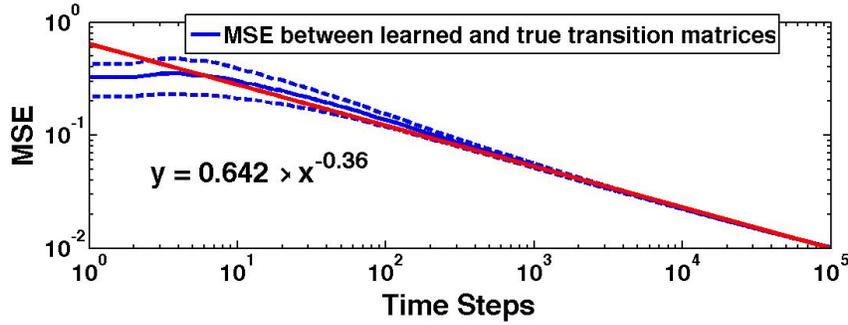
### 3.3 Numerical Experiments

Learning both emission and transition probability matrices at the same time using the online EM algorithm with stochastic approximation is very difficult because there are many local minima in the likelihood function. To simplify the task, we divide the learning process into two phases. The first phase involves learning the emission probability  $g$  when the hidden world state is stationary, *i.e.*,  $W_{ij} = f_{ij} = \delta_{ij}$ . This corresponds to learning the observation model of static objects at the center of gaze before learning the dynamics  $f$  of objects. After an observation model  $g$  is learned, we relax the stationary constraint, and allow the spiking network to update the recurrent weights  $W$  to learn the arbitrary transition probability  $f$ .

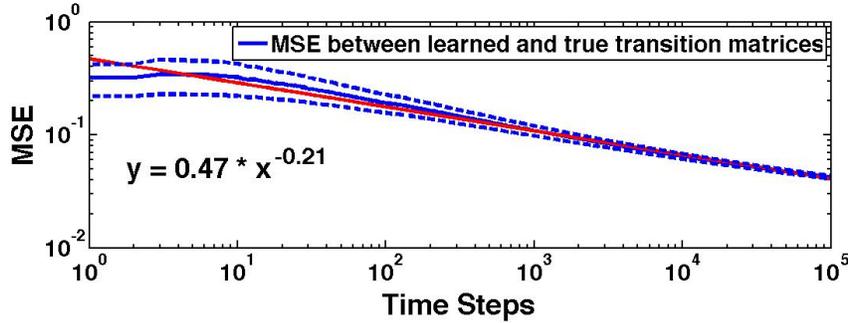
Figure 4 illustrates the performance of learning rules (47) and (51) for a discrete HMM with  $\mathcal{X} = 4$  and  $\mathcal{Z} = 12$ .  $X$  and  $Z$  values are spaced equally apart:  $X \in$



(a)



(b)



(c)

Figure 4: **Performance of the Hebbian learning rules.** (a) The mean square error (MSE) between the learned  $M^k$  and the true emission probability  $g$  as a function of the number of time steps  $k$ . The blue solid line shows the average MSE over trials with different  $g$ . The initial estimator  $M^0$  was randomly chosen. The dotted lines show  $\pm 1$  standard deviation. The red straight line is the power law fit  $y = ax^b$  to the average MSE. (b) MSE between learned  $W^k$  and true transition matrix  $f$  when the observation noise  $\sigma_Z$  is low, after the emission model  $g$  has been learned. (c) MSE between learned  $W^k$  and true transition matrix  $f$  when the observation noise  $\sigma_Z$  is high. As expected, learning is slower when the noise level of the observations is larger.

$\{1, \dots, 4\}$  and  $Z \in \{\frac{2}{3}, 1, \frac{4}{3}, \dots, 4\frac{1}{3}\}$ . The transition probability matrix  $f$  then involves  $4 \times 4 = 16$  parameters and the emission probability matrix  $g$  involves  $12 \times 4 = 48$  parameters.

In figure 4(a), we examine the performance of learning rule 47 for the feedforward weights  $M^k$ , with fixed transition matrix  $f_{ij} = \delta_{ij}$ . The true emission probability matrix has the form  $g_{.j} = P(Z_k | X_k = x^j) \sim N(x^j, \sigma_Z^2)$ . Each column of  $g$  is a Gaussian with observation noise  $\sigma_Z$ . The solid blue curve shows the average MSE between the learned feedforward weights  $M^k$  and the true emission probability matrix  $g$  over trials with different  $g$ , with  $MSE(k) = \sqrt{\sum_{ij} (M_{ij}^k - g_{ij})^2}$ . The dotted lines show  $\pm 1$  standard deviation for MSE based on 10 different trials.  $\sigma_Z$  varied from trial to trial and was drawn uniformly between 0.2 and 0.4, representing different levels of observation noises. The initial spike distribution was uniform  $n_{0|0}^i = n_{0|0}^j, \forall i, j = 1 \dots, \mathcal{X}$  and the initial estimate  $M_{i,j}^0 = \frac{1}{\mathcal{X}}$ . The learning rate was set to  $\gamma_k = \frac{1}{k}$ , although a small constant learning rate such as  $\gamma_k = 10^{-5}$  also gives rise to similar learning results.

A notable feature in figure 4(a) is that the average MSE exhibits a fast power-law decrease. The red solid line in figure 4(a) represents the power-law fit to the average MSE:  $MSE(k) \propto k^{-1.1}$ . Furthermore, the standard deviation of MSE approaches zero as  $k$  grows large. Figure 4(a) thus shows the asymptotic convergence of equation (47) irrespective of the  $\sigma_Z$  of the true emission matrix  $g$ .

We next examined the performance of learning rule 51 for the recurrent weights  $W^k$ , given the learned emission probability matrix  $g$  (the true transition probabilities  $f$  are unknown to the network). The initial estimator  $W_{ij}^0 = \frac{1}{\mathcal{X}}$ . Performance was evaluated by calculating the mean square error  $MSE(k) = \sqrt{\sum_{ij} (W_{ij}^k - f_{ij})^2}$  between the learned recurrent weight  $W^k$  and the true  $f$ . Different randomly chosen transition matrices  $f$  were tested. The average MSE and standard deviation over trials with different  $f$  are displayed in blue solid and dotted lines respectively in figure 4(b) and figure 4(c).

When  $\sigma_Z = 0.04$ , the observation noise is  $\frac{0.04}{1/3} = 12\%$  of the separation between two observed states. Hidden state identification in this case is relatively easy. The red solid line in figure 4(b) represents the power-law fit to the average MSE:  $MSE(k) \propto k^{-0.36}$ . Furthermore the standard deviation of MSE approaches zero as  $k$  grows large, indicating asymptotic convergence of equation 51 irrespective of the form of the true transition matrix  $f$ . Similar convergence results can still be obtained for higher  $\sigma_Z$ , e.g.,

$\sigma_Z = 0.4$  (figure 4(c)). In this case, hidden state identification is much more difficult as the observation noise is now 1.2 times the separation between two observed states. This difficulty is reflected in a slower asymptotic convergence rate, with a power-law fit  $MSE(k) \propto k^{-0.21}$ , as indicated by the red solid line in figure 4(c). In the extreme case when  $\sigma_Z = 1$ , hidden state identification becomes impossible due to high observation noise, causing the online learning rule (51) to fail.

## 4 LIF Implementation and Results

In this section, we demonstrate that the network model can be implemented using leaky integrate-and-fire neurons, which are commonly used to model CNS neurons. Model parameters are chosen to reflect those reported for biological neurons.

Figure 5 shows the dynamics of an example neuron. Let  $v_i$  be the membrane potential of a neuron whose preferred state is  $x^i$ .

$$\tau_m \frac{dv_i}{dt} = -v_i + R \times (I^S(t) + I^R(t)) \quad (52)$$

where  $\tau_m$  is the membrane time constant and  $R$  is the input resistance. The neuron spikes when  $v_i(t) > v_{th}$ . Note that the time variable  $t$  is continuous, while the HMM time variable  $k$  is discrete. Suppose the size of the HMM time step is  $\Delta_{hmm}$ . We define  $Z_t = Z_k$  and  $X_t = X_k$  if  $(k-1)\Delta_{hmm} < t \leq k\Delta_{hmm}$ .  $n_{k|k}^i$  represents the spike count in the time interval  $((k-1)\Delta_{hmm}, k\Delta_{hmm}]$  over neurons in the  $i$ -th sub-population. If an LIF neuron in the  $i$ -th sub-population fires at time  $t$ ,  $(k-1)\Delta_{hmm} < t \leq k\Delta_{hmm}$ , then it evokes a recurrent EPSP in the  $j$ -th sub-population at time  $t' = t + \Delta_{hmm}$ , with probability  $W_{ij}$ . A neuron also receives sensory EPSPs, whose arrival probability is proportional to  $M_{ij}dt$ .  $I^R(t)$  and  $I^S(t)$  represent the accumulated recurrent and sensory inputs respectively. Using the notation  $\iota = R$  or  $S$ , we have:

$$I^\iota(t) = \frac{\alpha^\iota}{\tau_\iota} \sum_{\nu=1}^{\nu} \exp(-(t - t_\nu^\iota)/\tau_\iota) \Theta(t - t_\nu^\iota) \quad (53)$$

where  $\tau_\iota$  is the synaptic time constant,  $\alpha^\iota$  is the amplitude of synaptic input, and  $\{t_1^\iota, \dots, t_\nu^\iota\}$  are the arrival times of pre-synaptic spikes. The Heaviside step function  $\Theta(t)$  ensures causality.

The normalized EPSP evoked by one input spike (either sensory or recurrent) mimics the effect of an ‘alpha’ synapse:

$$\epsilon^l(\tau) \propto \frac{\exp(-\tau/\tau_m) - \exp(-\tau/\tau_l)}{\tau_m - \tau_l} \Theta(\tau); \quad \tau = t - t_\nu^l; \tau_m > \tau_l \quad (54)$$

with  $\max_\tau \epsilon(\tau) = 1$ . The synaptic constants  $\tau_l$  are smaller than the membrane time constants  $\tau_m$  (Gerstner & Kistler 2002, Shadlen & Newsome 1994), e.g.,  $\tau_l = 1\text{ms}$  and  $\tau_m = 8\text{ms}$ . Thus, one can drop the dependence of  $v_i(t)$  on the arrival times of past spikes except for the most recent sensory and recurrent spikes.

Let  $t_\zeta^t = \max\{t_\zeta^t | t_\zeta^t < t\}$  and  $t_0 = \min(t_\zeta^R, t_\zeta^S)$ . Then:

$$v_i(t) = v_i(t_0) + \alpha^R \epsilon^R(t - t_\zeta^R) + \alpha^S \epsilon^S(t - t_\zeta^S); \quad t_0 \leq t < \min(t_{\zeta+1}^R, t_{\zeta+1}^S) \quad (55)$$

For the model to perform Bayesian filtering correctly, the LIF neuron should fire when there is coincident recurrent and sensory input and minimize firing for a sequence of spikes of one type. Due to the sparseness of the network, the proportionality constants  $C_W$  and  $C_M$  can be chosen such that interspike intervals between two input spikes of the same type are much greater than the membrane time constant:  $t_{\zeta+1}^t - t_\zeta^t \gg \tau_m$ . This helps reduce the probability of spiking for multiple spikes of the same type. We also choose  $\alpha^R$  and  $\alpha^S$  such that  $\max(v_i(t)) > v_{\text{th}}$  only when  $|t_\zeta^R - t_\zeta^S| \leq \Delta_{\text{cd}}$ , where  $\Delta_{\text{cd}}$  is the coincidence detection window. We then obtain a LIF model neuron that fires only if it receives both sensory and recurrent inputs within  $\Delta_{\text{cd}}$ . Finally, we require that  $\Delta_{\text{cd}} < \Delta_{\text{hmm}}$  to ensure that the neuron’s spiking probability is proportional to the product of likelihood  $P(Z_k | X_k)$  and the prediction probability  $P(X_k | Z_{1:k-1})$ , which in turn is proportional to the posterior probability  $P(X_k | Z_{1:k})$ .

The simple model above can be extended to handle the case of multiple spikes of the same type (the cases where  $t_{\zeta+1}^t - t_\zeta^t$  is small) by adding the mechanism of short-term synaptic depression (STSD) to the model. STSD usually occurs in cortical neurons due to depletion of synaptic vesicles (Zucker & Regehr 2002). The amplitude of the  $\hat{\zeta}$ -th input in the presence of rapid STSD can be modeled by (Tsodyks & Markram 1997):  $\alpha_\zeta^t = \alpha_{\text{max}}^t [1 - \exp(-(t_\zeta^t - t_{\zeta-1}^t)/\tau_m)]$ . Then the maximum response to two successive

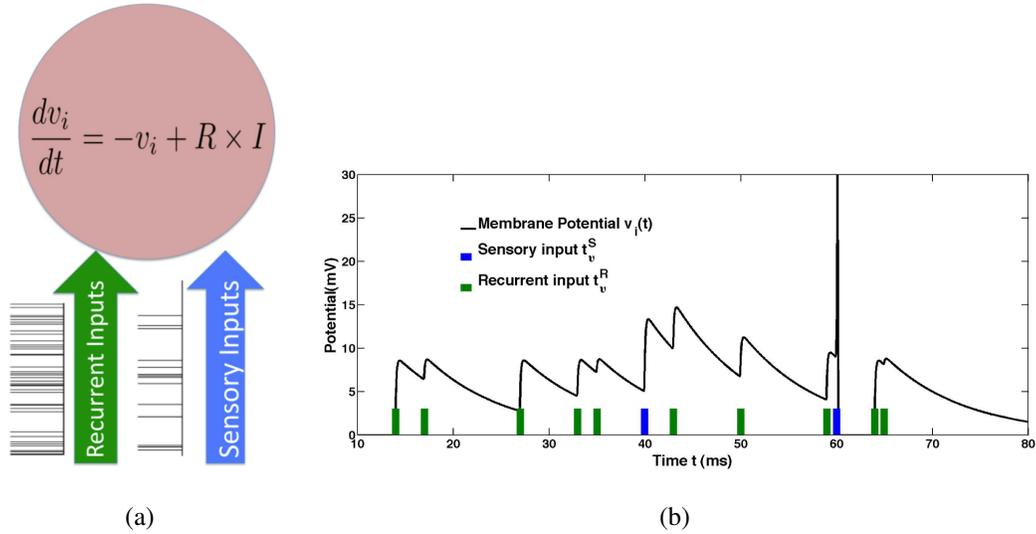


Figure 5: **Model LIF neuron.** (a) LIF neuron receiving inputs from recurrent and sensory synapses. (b) The black curve shows an example trajectory of the membrane potential. Green and blue bars represent the arrival times of recurrent and sensory spikes respectively.

EPSPs from the same synapse can be simplified as follows:

$$\begin{aligned}
 v_i(t) &= v_i(t_{\hat{\zeta}-1}^t) + \alpha_{\hat{\zeta}}^t \epsilon^t (t - t_{\hat{\zeta}}^t) \\
 &\leq \alpha_{\max}^t \epsilon^t (t_{\hat{\zeta}}^t - t_{\hat{\zeta}-1}^t) + \alpha_{\hat{\zeta}}^t \epsilon^t (t - t_{\hat{\zeta}}^t) \\
 &\leq \alpha_{\max}^t \exp(-(t_{\hat{\zeta}}^t - t_{\hat{\zeta}-1}^t)/\tau_m) + \alpha_{\max}^t [1 - \exp(-(t_{\hat{\zeta}}^t - t_{\hat{\zeta}-1}^t)/\tau_m)] = \alpha_{\max}^t
 \end{aligned} \tag{56}$$

As a result, even if a model neuron receives more than one input spikes from either sensory or recurrent synapses in a short period of time, successive spikes will not further depolarize the membrane potential due to short-term synaptic depression. Such a voltage saturation effect has also been experimentally observed at pyramidal-pyramidal cell connections in adult rat neocortex (e.g., figure 6 of (Thomson 1997)).

In figure 5, we show an example trajectory of the membrane potential  $v_i(t)$ . The model parameters were chosen to be consistent with those reported in typical CNS neurons (Gerstner & Kistler 2002, Shadlen & Newsome 1994): refractory time period = 2.5ms,  $\tau_m = 8.33$ ms,  $\tau_R = \tau_S = 1$ ms,  $R = 138.8$  M  $\Omega$ ,  $\alpha^R = \alpha^S = 0.6$ nA and  $v_{th} = 15$ mV. The model neuron fires only if the sensory and recurrent inputs arrive within a time window  $\Delta_{cd} = 0.6$ ms. Note that short-term synaptic depression guarantees that the neuron will not fire even when the interspike interval between two recurrent spikes

is 1 ms. Thus we have an LIF model neuron that is equivalent to the binary neuron described in section 2.2

In the following two sections, we illustrate how a network of such LIF neurons can perform Bayesian inference for two different tasks and compare the simulation results with biological data.

## 4.1 Static World State: An Example from Sensory Adaptation

We first consider the special case where the dynamics of the hidden state is static and where Bayesian filtering reduces to Kalman filtering. We relate this abstract model to neural data and show how the network introduced above for Bayesian inference can explain the data.

Let  $X_k \in \mathbb{R}$  be the mean light intensity (luminance) of a static visual stimulus,  $X_k = x^0, \forall k, 1 \leq k \leq K_0$ . The measurements  $Z_k \in \mathbb{R}$  are the intensities of the time-varying noisy stimulus observed by the retina, with standard deviation (contrast)  $\sigma_Z$ :  $(Z_k - X_k) \sim N(0, \sigma_Z^2)$ . The estimated mean and variance of the posterior distribution over  $X_k$ , given past inputs, can be described using a Kalman filter (Russell & Norvig 2003):

$$E[X_k] = \frac{E[X_{k-1}] \times \sigma_Z^2 + Z_k \times \text{Var}[X_{k-1}]}{\text{Var}[X_{k-1}] + \sigma_Z^2} \quad (57)$$

$$\frac{1}{\text{Var}[X_k]} = \frac{1}{\sigma_Z^2} + \frac{1}{\text{Var}[X_{k-1}]} \quad (58)$$

Equation 57 has an intuitive explanation: the mean at time  $k$  is the weighted average of the previous mean  $E[X_{k-1}]$  and the current observation  $Z_k$ , each weight corresponding to the variance of the other component. Thus, if there is more noise in the sensory input (higher  $\sigma_Z^2$ ), more weight is given to the previous mean  $E[X_{k-1}]$ , and vice versa. Also, from equation 58, we have  $\frac{\text{Var}[X_k]}{\text{Var}[X_{k-1}]} = \frac{\sigma_Z^2}{\sigma_Z^2 + \text{Var}[X_{k-1}]} < 1$ . Thus, the variance of  $X_k$  decreases with time  $k$ , and will eventually converge to zero as  $k \rightarrow \infty$ .

Now consider the situation where the hidden variable  $X_k$  is suddenly switched to another state after time step  $K_0$ :  $X_k = x^1$  for  $k > K_0$ . Since  $X_k$  is hidden and the system is unaware of this change, the system continues to apply equations 57 and 58 for  $k > K_0$ . Thus, starting with mean  $E[X_{K_0}]$  and variance  $\text{Var}[X_{K_0}]$ , and combining

equations 57 and 58, we obtain :

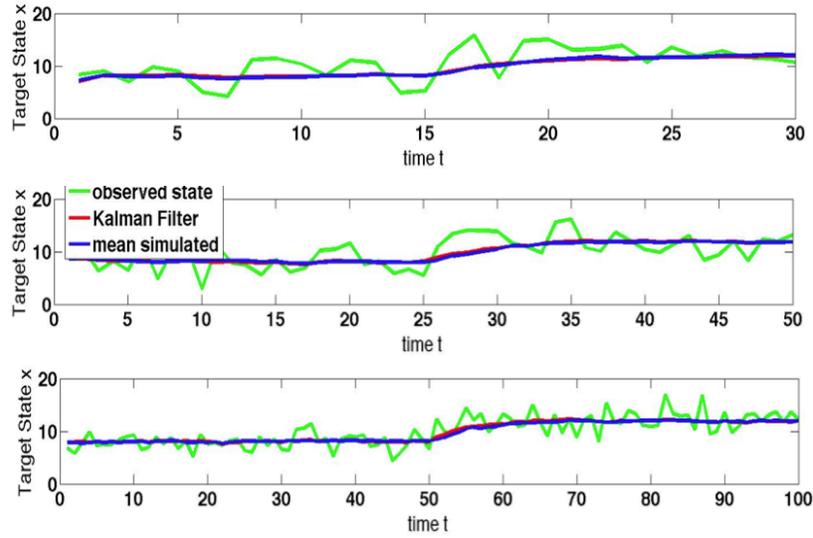
$$\begin{aligned}
E[X_k] &= E[X_{k-1}] \times \frac{\text{Var}[X_k]}{\text{Var}[X_{k-1}]} + Z_k \times \frac{\text{Var}[X_k]}{\sigma_Z^2} \\
&= \left( E[X_{k-2}] \frac{\text{Var}[X_{k-1}]}{\text{Var}[X_{k-2}]} + Z_{k-1} \frac{\text{Var}[X_{k-1}]}{\sigma_Z^2} \right) \times \frac{\text{Var}[X_k]}{\text{Var}[X_{k-1}]} + Z_k \times \frac{\text{Var}[X_k]}{\sigma_Z^2} \\
&= E[X_{k-2}] \times \frac{\text{Var}[X_k]}{\text{Var}[X_{k-2}]} + (Z_{k-1} + Z_k) \times \frac{\text{Var}[X_k]}{\sigma_Z^2} \\
&\dots \\
&= E[X_{K_0}] \times \frac{\text{Var}[X_k]}{\text{Var}[X_{K_0}]} + \sum_{s=K_0+1}^k Z_s \times \frac{\text{Var}[X_k]}{\sigma_Z^2} \tag{59}
\end{aligned}$$

$$\begin{aligned}
\frac{1}{\text{Var}[X_k]} &= \frac{k - K_0}{\sigma_Z^2} + \frac{1}{\text{Var}[X_{K_0}]} \\
\text{Var}[X_k] &= \frac{\sigma_Z^2 \text{Var}[X_{K_0}]}{\sigma_Z^2 + (k - K_0) \text{Var}[X_{K_0}]} \tag{60}
\end{aligned}$$

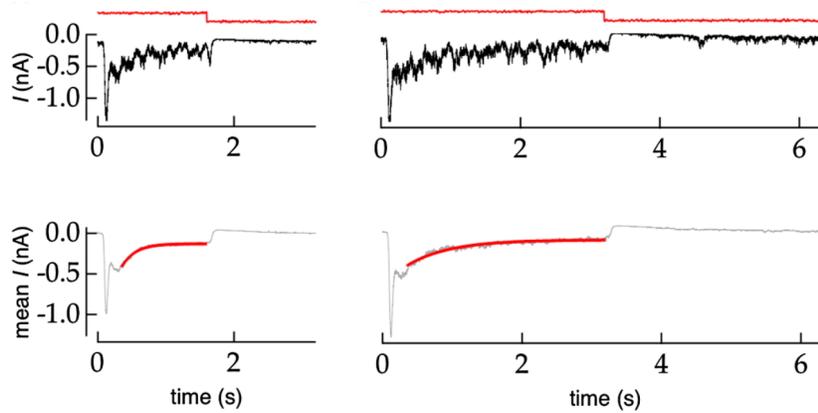
We see that the mean  $E[X_k]$  is a weighted average of the prior mean  $E[X_{K_0}]$  and the new observations  $\{Z_s\}$ . If  $\text{Var}[X_{K_0}]$  is small, more weight is given to the prior estimate  $E[X_{K_0}]$ . The prior estimates  $E[X_{K_0}]$  and  $\text{Var}[X_{K_0}]$  are determined by the time of transition  $K_0$ . For example, when the initial variance  $\text{Var}[X_0] = \infty$ , we have  $\frac{1}{\text{Var}[X_0]} = 0$  and  $\text{Var}[X_k] = \sigma_Z^2/k$  from equation 58 initial state  $x^0$ , the lesser  $\text{Var}[X_{K_0}]$  becomes as the system accumulates more evidence for  $x^0$ . Thus, when the state is changed at time step  $K_0 + 1$ , it takes longer for  $E[X_k]$  to converge to the new state.

Figure 6(a) shows three examples of temporal evolutions of  $E[X_k]$  (red traces) for different values for  $K_0$  (note the different scales on the time axis). All three trajectories display a form of exponential-like dynamics after  $K_0$ , with a half-life  $\approx K_0$ .

The phenomena discussed above can be interpreted as sensory adaptation, a key property exhibited by the brain. Efficient coding of the sensory world requires that the brain optimally estimate and adapt to the statistics of its sensory inputs (Barlow 1990). In the example above, this corresponds to the estimation of  $X_k$  from noisy observations  $Z_k$ . A switch from  $x^0$  to  $x^1$  is equivalent to an abrupt change in luminance of the environment, e.g, a sudden exposure to bright daylight when coming out of a dark movie theatre. The model above suggests that the time course of ‘‘adaptation’’ of  $E[X_k]$  to the new state  $x^1$  is determined by the duration of the prior state  $x^0$ . This is consistent with previous observations that the dynamics of the adaptation process could be dependent on stimulus history (Fairhall et al. 2001, Wark et al. 2009). Figure 6(b) shows the mean



(a)



(b)

Figure 6: **Sensory Adaptation and Bayesian Filtering.** (a) The hidden state (luminance) was switched from one value to another at specific time instants (time step 15, 25, and 50 respectively in the plots). The green curve represents the noisy stimuli  $Z_t$  available to the system, the red curve shows the estimation of  $X_t$  using the Kalman filter equation 57, and the blue curve displays the posterior mean  $\sum_{i=1}^{\mathcal{X}} x^i \hat{p}_k^i$  computed from the spiking LIF network model. Note the similarity in the time course of adaptation across different time scales (different scales on time axis for the three plots). (b) Above: Time course of excitatory synaptic input to a retinal ganglion cell (black trace) in response to a single cycle of stimulus (red trace). Below: Mean synaptic current over approximately 50 trials as above. The embedded red curve is the exponential fit to the adaptation. Compare with the red and blue curves in (a). (Plots in (b) are from (Wark et al. 2009))

synaptic current to an ON retinal ganglion cell (RGC) elicited by periodic switches between low luminance and high luminance stimuli. The time course of adaptation is dependent on the switching period  $K$ . The longer the retina is exposed to the low luminance environment, the slower the time course of adaptation to the high luminance. Wark et al. (2009) argued that the neural response in RGC encodes the mean of the posterior distribution of the visual stimulus  $X_k$ . They hypothesized that sensory adaptation involves Bayesian inference of stimulus parameters and suggested that the visual system may employ a form of Kalman filter.

The spiking network model we have proposed can be used to model sensory adaptation phenomena such as those reported by Wark et al. (2009). Sensory neurons in the model measure the noisy light intensity  $Z_k$ . The inference layer LIF neurons combine this sensory likelihood information with recurrent inputs to obtain a grid-based approximation  $\{\hat{P}_k^i\}$  of the posterior distribution of luminance  $X_k$ . If the model network correctly implements Bayesian filtering, one would expect the posterior mean  $\sum_{i=1}^{\mathcal{X}} x^i \hat{p}_k^i$  approximates the predictions from a Kalman filter. This is indeed the case which can be seen in figure 6(a) (blue trace).

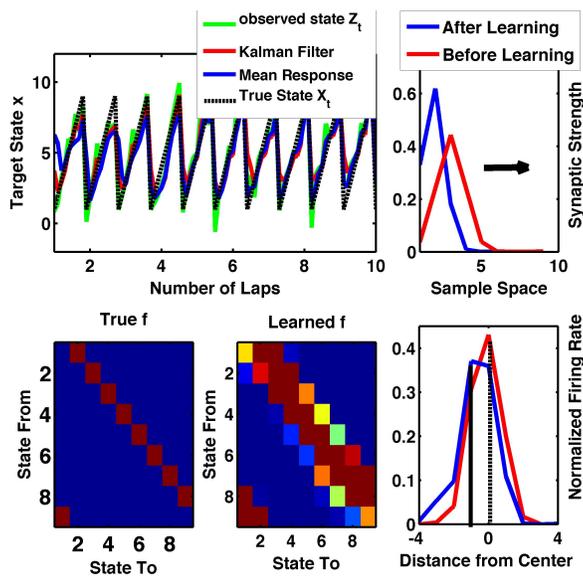
## 4.2 Dynamic World State: Adaptation in Hippocampal Place Cells

Consider an experiment where a rat moves along a linear track. Let  $X_t \in \Omega = 1, \dots, N$  be the position of the rat along the track. The motion is deterministic such that the transition probability matrix  $f$  defined by  $\delta(i, i+1)$  for  $1 \leq i < N$ , with a reset to the start position upon reaching the end of the track. The matrix  $f$  is unknown and the measurement of position  $X_t$  is noisy:  $Z_t = X_t + \eta_t$ , where  $Z_t$  is the observable input to the sensory system and  $\eta_t$  is white noise with variance  $\sigma_Z^2$ . The initial recurrent weights (at time 0) are set to be zero mean Gaussian with width  $\sigma_{\text{prior}}$ , i.e.,  $W_{ij}(0) = \exp(-(i-j)^2/(2\sigma_{\text{prior}}^2))$ , a biased estimator of  $f$ .

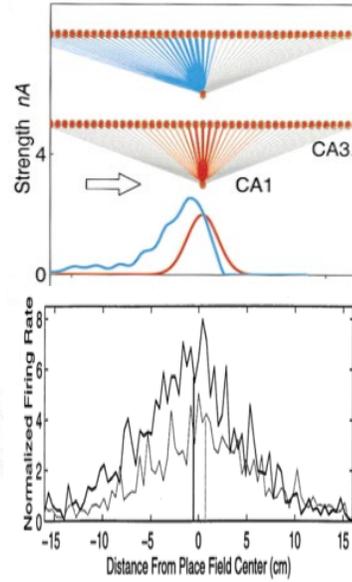
Figure 7(a) shows the recurrent weights  $W(t)$  learned using equation 51 after 10 laps. The synaptic weights  $\mathbf{W}_j$  become asymmetric and their centers show a backward shift after learning.<sup>1</sup> A similar backward shift has been reported in rat hippocampal

---

<sup>1</sup>The recurrent weights  $W$  in the model need not necessarily correspond to a single set of synaptic weights in the hippocampus but could instead capture the effect of a larger multi-synaptic loop such as the hippocampal-entorhinal network.



(a)



(b)

Figure 7: **Adaptation in Hippocampal Place Cells.** (a) Upper left: estimated  $X_k$ , noisy observation  $Z_k$  and the prediction from Kalman filter are shown in blue, green and red, respectively. Upper right: comparison between the learned  $W_3(T)$  with the initial  $W_3(0)$  after 10 laps. Bottom left: true transition matrix  $f$ . Bottom middle: learned recurrent weight matrix  $W(T)$ . Bottom right: Normalized firing rate during the first and the last lap. Model parameters:  $N = 9$ ,  $\sigma_Z = 0.1 \times N$  and  $\sigma_{\text{prior}} = 0.1 \times N$  (b) Top: (Figure from (Mehta et al. 2000)) Computational Model of CA3→CA1 network. The synaptic weight matrix shifts backward as the rat moves forward. Bottom: (Figure from (Mehta et al. 1997)) Histograms of firing rates in place cells recorded from rats during the first and the last lap. The center of the place field shifted backwards after learning.

place cells (Mehta et al. 2000), as shown in figure 7(b).

## 5 Related Work

This article makes contributions to two areas: neural models of Bayesian inference and learning models for online parameter estimation. We review previous work in these two areas below.

There have been a number of models of probabilistic inference in biological neural networks. The Boltzmann machine (Hinton & Sejnowski 1983, Sejnowski 1986) is perhaps the earliest example of a neural network capable of probabilistic inference. Similar to our model, Boltzmann machines employ a sampling based inference technique that allows them to learn an internal probabilistic model from the observations. Our model differs from the Boltzmann machine in the underlying generative model. Our model can represent probabilistic state transitions and can implement the state-space dynamics of arbitrary hidden Markov models. A recurrent neural network capable of statistical inference in hidden Markov models was first suggested by Bridle (1990). One limitation of Bridle’s model, known as the Alpha-net, was the assumption that the network could multiply arbitrary probabilities. In contrast, the inference performed in our model requires binary AND operations, which can be more easily implemented in a population of neurons.

The idea of representing probability distributions using populations of neurons originated in early work on basis function networks (Anderson & Essen 1990, Deneve & Pouget 2001) and distributional population coding (Zhang et al. 1998, Zemel et al. 1998, Zemel & Dayan 1999, Wu et al. 2003) models. In the basis function approach, probability distributions are decomposed into linear combinations of basis functions, which are proportional to the measurable tuning functions of neurons. Due to its additive nature, the probability distributions that can be represented by this approach cannot be sharper than the component distributions. In contrast, the model proposed in this article can approximate probability distributions of any shape as a sampled distribution. Distributional population coding (DPC) uses a generative model to encode a probability distribution in a population of neurons. DPC requires a sophisticated non-neural decoding mechanism to recover the distribution from the neural population response,

compared to the straightforward readout of the distribution from the spiking network proposed in this article.

A number of neural models for Bayesian inference of hidden world state have been proposed in recent years. Rao (2004) proposed a model in which the firing rates of a population of neurons approximate the log probabilities of the time-varying posterior distribution of hidden states, given noisy observations, for an arbitrary hidden Markov model. Beck & Pouget (2007) extended Rao's work using nonlinear recurrent networks for exact inference, with firing rates in a population directly proportional to posterior probabilities. Rao (2005) proposed a nonlinear network model for implementing belief propagation for Bayesian inference in arbitrary graphical models. Models based on predictive coding (Rao & Ballard 1997, Rao 1999), basis function networks (Deneve et al. 2007) and line attractor networks (Wilson & Finkel 2009) have been proposed for implementing the Kalman filter, which assumes all distributions are Gaussian and the dynamics is linear. More recently, Bobrowski et al. (2008, 2009) proposed a spiking network model that can compute the optimal posterior distribution in continuous time. One limitation of these models is that the model parameters (the emission probability and transition probability matrix) are assumed to be known a priori, whereas those model parameters are learned using a form of Hebbian learning in the model proposed here. Probabilistic population codes (Ma et al. 2006, Beck et al. 2008) (PPC) provide an alternative way to estimating the probability distribution of the hidden state in populations of neurons. The PPC model exploits neural variability to turn products in Bayesian computations into sums without the need for a log likelihood representation. However, the PPC approach assumes a static world state  $X$ . Deneve (2008*a,b*) proposed a model for inference and learning based on the dynamics of a single neuron but assuming a binary world state.

The model for learning we have proposed builds on prior work by Andrieu et al. (2005), Mongillo & Deneve (2008), and Cappe & Moulines (2009), Cappe (2009). The online algorithm used in our model for estimating HMM parameters involves three levels of approximation. The first level involves performing a stochastic approximation to estimate the expected complete-data sufficient statistics over the joint distribution of all hidden states and observations. Cappe & Moulines (2009) showed that under some mild conditions, such an approximation produces a consistent, asymptotically efficient

estimator of the true parameters. The second approximation comes from the use of filtered rather than smoothed posterior distributions in equation (40). Although the convergence reported in section 3.3 is encouraging, a rigorous proof of convergence remains to be shown. The asymptotic convergence rate using only the filtered distribution is about one third the convergence rate obtained from the algorithms described by Mongillo & Deneve (2008) and Cappe & Moulines (2009), where the smoothed distribution is used. The third approximation results from Monte-Carlo sampling of the posterior distribution in equation (42). As discussed in section 2.3, the Monte Carlo approximation converges in the limit of large numbers of spikes.

## 6 Discussion

We have proposed a two-layer spiking network model that implements Bayesian inference and learning. The model encodes the posterior distribution of hidden world states as a sampled distribution represented by spikes across a neural population. Neural variability in spiking arises naturally as a consequence of sampling necessary for inference. Our model embraces many biological properties that are frequently observed in CNS neurons, such as leaky integrate-and-fire dynamics, short-term synaptic depression, and spike-time dependent Hebbian plasticity.

The model we have proposed assumes an underlying hidden Markov model (HMM) for processing sensory information. This assumption implies that the sensory system makes noisy observations  $Z$  of the external world at discrete time steps (corresponding to the time steps of the HMM), and updates its belief over hidden world state  $X$  each time a new observation is made. The mechanism of coincidence detection in inference neurons provides a way of bridging the gap between the discrete time steps in the HMM and continuous time in a neural network. The arrival of sensory EPSPs at time  $t_i$  mark the onset of the  $i$ -th HMM epoch. The inference neurons then compute the posterior belief by combining the current observation and prior belief before time step  $t_{i+1}$ . This implies that the coincidence detection window should be less than the length of one HMM epoch, requiring relatively precise timing and low temporal variability in the sensory observations. The brain's ability to transmit temporal information with high precision and low variability has been studied by a number of researchers Kara et al.

(2000), Wang et al. (2010). In particular, Wang et al. (2010) found that the output firing rate is a highly nonlinear function of the number of synchronous synaptic events. This supports the assumption in the model that recurrent or feed-forward inputs alone are not sufficient to cause an inference neuron to spike: the coincidence of the two inputs is required to make spiking highly likely.

The model suggests that, contrary to the commonly held view, variability in spiking does not reflect “noise” in the nervous system but captures the animal’s uncertainty about the outside world. This suggestion is similar to previous models linking firing rate variability to probabilistic representations (Hoyer et al. 2002, Ma et al. 2006) but differs in the emphasis on spike-based representations and time-varying inputs. In our model, a probability distribution over a finite sample space is represented by spike counts in neural sub-populations. Treating spikes as random samples requires that neurons in a pool of identical cells fire independently. This hypothesis is supported by a recent experimental finding by Ecker et al. (2010), who report that nearby neurons with similar orientation tuning and common inputs show little or no correlation in activity. Our model offers a functional explanation for the existence of such decorrelated neuronal activity in the cortex.

We showed that spike counts of neuronal subpopulations in the model are unbiased estimators of the desired posterior probabilities, with variance that is linear with respect to the mean when plotted on a log-log plot. Since neurons in the same sub-population fire independently, the spike count  $n_{k|k}^i$  can be viewed as the firing rate of a single cell averaged over multiple independent trials. Thus, our model can account for the Poisson variability observed in cortical cells (Dean 1981, Tolhurst et al. 1983). One can also interpret  $n_{k|k}^i$  as the temporal firing rate. Instead of considering  $L$  neurons in  $\mathcal{X}$  different sub-populations, we could consider only  $\mathcal{X}$  neurons but divide the HMM time step of size  $\Delta$ ) into  $\mathcal{L}$  time bins, each having size  $\Delta/\mathcal{L}$ . In such a model, a neuron spikes if and only if it receives both sensory and recurrent inputs in the same time bin, with the duration of a spike being less than  $\Delta/\mathcal{L}$ .  $n_{k|k}^i$  then becomes the spike count of neuron  $i$  in HMM time step  $k$ , and spikes are propagated as before, with the same recurrent weights  $W_{ij}$  and feedforward weights  $M_{ij}$ . Such a neural implementation requires much less neurons, but demands a long period of time to compute the spike count  $n_{k|k}^i$ . However, the temporal versus spatial representations are not necessarily

mutually exclusive because both temporal and spatial spike variability provide the trial-to-trial variability required by Monte Carlo sampling.

Unlike many previous models of cortical computation, our model treats synaptic transmission between neurons as a stochastic process rather than a deterministic event. This acknowledges the inherent stochastic nature of neurotransmitter releases and bindings. Synapses between neurons usually have only a small number of vesicles available and a limited number of post-synaptic receptors near the release sites. Recent physiological studies (Nimchinsky et al. 2004) have shown that only 3 NMDA receptors open on average per release during synaptic transmission. These observations lend support to the view espoused by the model that synapses should be treated as probabilistic computational units rather than as simple scalar parameters.

## References

- Anderson, C. & Essen, D. V. (1990), *Computational intelligence: Imitating life*, New York: IEEE Press, chapter Neurobiological computational systems.
- Andrieu, C., Doucet, A. & Tadic, V. (2005), ‘Online parameter estimation in general state-space models’, *Proceedings of the 44th Conference on Decision and Control* pp. 332–337.
- Barlow, H. (1990), A theory about the functional role and synaptic mechanism of visual aftereffects, in C. Blakemore, ed., ‘Vision: Coding and Efficiency’, Cambridge University Press, pp. 363–375.
- Beck, J., Ma, W., Kiani, R., Hanks, T., Churchland, A., Roitman, J., Shadlen, M., Latham, P. & Pouget, A. (2008), ‘Bayesian decision making with probabilistic population codes’, *Neuron* **60**(6), 1142–1145.
- Beck, J. & Pouget, A. (2007), ‘Exact inferences in a neural implementation of a hidden markov model’, *Neural Computation* **19**(5), 1344–1361.
- Bobrowski, O., Meir, R. & Eldar, Y. (2009), ‘Bayesian filtering in spiking neural networks: noise adaptation and multisensory integration’, *Neural Computation* **21**(5), 1277–1320.

- Bobrowski, O., Meir, R., Shoham, S. & Eldar, Y. (2008), ‘A neural network implementation optimal state estimation based on dynamic spike train decoding’, *Neural information procession systems* **20**, 145–152.
- Bridle, J. (1990), ‘Alpha-nets: A recurrent “neural” network architecture with a hidden markov model interpretation’, *Speech Communication* **9**(1).
- Cappe, O. (2009), ‘Online em algorithm for hidden markov models’.  
**URL:** <http://www.citebase.org/abstract?id=oai:arXiv.org:0908.2359>
- Cappe, O. & Moulines, E. (2009), ‘Online em algorithm for laten data models’.
- Casella, G. & Berger, R. (2001), *Statistical Inference; 2nd edition*, Duxbury Press.
- Chance, F. S. & Abbott, L. F. (2000), ‘Divisive inhibition in recurrent networks’, *Network* **11**, 119–129.
- Dean, A. (1981), ‘The variability of discharge of simple cells in the cat striate cortex.’, *Experimental Brain Research* **44**, 437–440.
- Dempster, A., Laird, N. & Rubin, D. (1977), ‘Maximum likelihood from incomplete data via the em algorithm’, *J. ROy. Statist. Soc. Ser. B* **39**(1), 1–38.
- Deneve, S. (2008a), ‘Bayesian spiking neurons i: Inference’, *Neural Computation* **20**, 91–117.
- Deneve, S. (2008b), ‘Bayesian spiking neurons ii: Learning’, *Neural Computation* **20**, 118–145.
- Deneve, S., J.R.Duhamel & Pouget, A. (2007), ‘Optimal sensorimotor integration in recurrent cortical networks: a neural implementation of kalman filters’, *J. Neurosic* **27**(21), 5744–5756.
- Deneve, S. & Pouget, A. (2001), ‘Bayesian estimation by interconnected neural networks’, *Society of Neuroscience Abstracts* **27**(237.11).
- Doucet, A., de Freitas, N. & Gordon, N. (2001), *Sequential Monte Carlo methods in practice*, Springer-Verlag.

- Doya, K., Ishii, S., Pouget, A. & Rao, R. P. N. (2007), *Bayesian Brain: Probabilistic Approaches to Neural Coding*, Cambridge, MA: MIT Press.
- Ecker, A. S., Berens, P., Keliris, G., Bethge, M., Logothetis, N. K. & Tolias, A. S. (2010), ‘Decorrelated neuronal firing in cortical microcircuits’, *Science* **327**(5965), 584–587.
- Fairhall, A., Lewen, G., Bialek, W. & de Ruyter van Steveninck, R. (2001), ‘Efficiency and ambiguity in an adaptive neural code’, *Nature* **412**(23), 787–792.
- Gerstner, W. & Kistler, W. (2002), *Spiking Neuron Models. Single Neurons, Populations, Plasticity*, Cambridge University Press.
- Hinton, G. & Sejnowski, T. (1983), ‘Optimal perceptual inference’, *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, .
- Hodges, J. L., J. & Cam, L. L. (1960), ‘The poisson approximation to the poisson binomial distribution’, *The Annals of Mathematical Statistics* **31**(3), 737–740.
- Hoyer, P. O., Hyrinen, A. & Arinen, A. H. (2002), ‘Interpreting neural response variability as monte carlo sampling of the posterior’, *Advances in Neural Information Processing Systems* .
- Kara, P., Reinagel, P. & Reid, R. (2000), ‘Low response variability in simultane’, *Neuron* **27**(3), 635–646.
- Knill, D. & Richards, W. (1996), *Perception as Bayesian inference*, Cambridge University Press.
- Kording, K. & Wolpert, D. (2004), ‘Bayesian integration in sensorimotor learning’, *Nature* **427**, 244–247.
- Lee, T. & Mumford, D. (2003), ‘Hierarchical bayesian inference in the visual cortex’, *Journal of the Optical Society of America* **20**, 1434–1448.
- Ma, W., Beck, J., Latham, P. & Pouget, A. (2006), ‘Bayesian inference with probabilistic population codes’, *Nature Neuroscience* **9**(11), 1432–1438.

- Maass, W. (2000), 'On the computational power of winner-take-all', *Neural Computation* **12**(11).
- Mehta, M., Barnes, C. & McNaughton, B. (1997), 'Experience-dependent, asymmetric expansion of hippocampal place field', *PNAS US* **94**, 8918–8921.
- Mehta, M., Quirk, M. & Wilson, M. (2000), 'Experience-dependent asymmetric shape of hippocampal receptive fields', *Neuron* **25**(3), 707–715.
- Mongillo, G. & Deneve, S. (2008), 'Online learning with hidden markov models', *Neural Computation* **20**, 1706–1716.
- Nimchinsky, E., Yasuda, R., Oertner, T. & Svoboda, K. (2004), 'The number of glutamate receptors opened by synaptic stimulation in single hippocampal spines', *J Neurosci* **24**, 2054–2064.
- Paulin, M. G. (2005), 'Evolution of the cerebellum as a neuronal machine for bayesian state estimation', *J. Neural Eng.* **2**, S219–S234.
- Rabiner, L. (1989), 'A tutorial on hidden markov models and selected applications in speech recognition', *Proceedings of the IEEE* **77**(2), 257–286.
- Rao, R. (1999), 'An optimal estimation approach to visual perception and learning', *Vision Research* **39**(11).
- Rao, R. (2004), 'Bayesian computation in recurrent neural circuits', *Neural Computation* **16**(1), 1–38.
- Rao, R. (2005), 'Bayesian inference and attentional modulation in the visual cortex', *Neuroreport* **16**(16), 1843–1848.
- Rao, R. & Ballard, D. (1997), 'Dynamic model of visual recognition predicts neural response properties in the visual cortex.', *Neural Computation* **9**(4).
- Rao, R. P. N., Olshausen, B. A. & Lewicki, M. S. (2002), *Probabilistic Models of the Brain: Perception and Neural Function*, Cambridge, MA: MIT Press.
- Robbins, H. & Monro, S. (1951), 'A stochastic approximation method', *Ann. Math. Statist* **22**(3), 400–407.

- Russell, S. J. & Norvig, P. (2003), *Artificial Intelligence: A Modern Approach*, Pearson Education.
- Sejnowski, T. J. (1986), 'Higher-order boltzmann machines', *AIP Conference Proceedings* **151**(1).
- Shadlen, M. & Newsome, W. (1994), 'Noise, neural codes and cortical organization', *Current Opinion in Neurobiology* **4**(4), 569–579.
- Thomson, A. (1997), 'Activity-dependent properties of synaptic transmission at two classes of connections made by rat neocortical pyramidal axons in vitro.', *J. Physiol* **502**, 131–147.
- Tolhurst, D., Movshon, J. A. & Dean, A. F. (1983), 'The statistical reliability of signals in single neurons in cat and monkey visual cortex.', *Vision Research* **23**, 775–785.
- Tsodyks, M. & Markram, H. (1997), 'The neural code between neocortical pyramidal neurons depends on neurotransmitter release probability', *Proceedings of the National Academy of Sciences of the United States of America* **94**(2), 719–723.
- Wang, H., Spencer, D., Fellous, J. & Sejnowski, T. (2010), 'Synchrony of thalamocortical inputs maximizes cortical reliability', *Science* **328**(5974), 106–109.
- Wark, B., Farihall, A. L. & Rieke, F. (2009), 'Timescales of inference in visual adaptation', *Neuron* **61**, 750–761.
- Wilson, R. & Finkel, L. (2009), 'A neural implementation of the kalman filter', *NIPS* .
- Wu, C. (1983), 'On the convergence properties of the em algorithm', *Ann. Statist* **11**(1), 95–103.
- Wu, S., Chen, D., Niranjan, M. & Amari, S. (2003), 'Sequential bayesian decoding within a population of neurons', *Neural Computation* **15**.
- Zemel, R., Dayan, P. & Pouget, A. (1998), 'Probabilistic interpretation of population codes', *Neural Computation* **10**(2).
- Zemel, R., Huys, Q., Natarajan, R. & P. Dayan (2005), 'Probabilistic computation in spiking populations.', *Advances in Neural Information Processing* **17**, 1609–1616.

Zemel, R. S. & Dayan, P. (1999), ‘Distributional population codes and multiple motion models’, *Advances in neural information processing system* **11**.

Zhang, K., Ginzburg, I., McNaughton, B. & T.J.Sejnowski (1998), ‘Interpreting neuronal population activity by reconstruction: A unified framework with application to hippocampal place cells.’, *Journal of Neuroscience* **16**(22).

Zuker, R. S. & Regehr, W. (2002), ‘Short-term synaptic plasticity’, *Annual Review of Physiology* **64**, 355–405.

## A Probability Distribution of the Synaptic Inputs $a_l^j$

The synaptic input  $a_l^j$  is the number of EPSPs received by the  $l$ -th posterior neuron in the  $j$ -th sub-population. Since the recurrent network in the posterior population is fully connected, each spiking neuron that fired in the previous time step will attempt to send an EPSP to its neighbors with success probability  $W_{ij}$ . Therefore,  $a_l^j(k+1)$  can be view as the sum of  $N_k$  independent, but not identically distributed Bernoulli trials. Dropping all unnecessary indices, we have

$$a = \sum_{m=1}^N \epsilon_m \quad (61)$$

where each binary random variable  $\epsilon_m$  has a success probability  $P(\epsilon_m = 1) = P_m$ .  $P_m = W_{ij}$  when  $\epsilon_m$  represents neurotransmitter release from cells in sub-population  $i$  to sub-population  $j$ .  $a$  has the so-called ‘‘Poisson binomial’’ distribution.  $P(a)$  can be approximated by a Poisson distribution  $P_\lambda$  where  $\lambda = \sum_m p_m$ . In the familiar case  $\epsilon_m$  are i.i.d,  $P_m = p$  for all  $m$ , and  $a$  will have the exact  $P_\lambda$  distribution with  $\lambda = Np$ .

Let  $Y$  be a random variable that follows the Poisson distribution with  $E(Y) = \sum_m P_m$  and let

$$D = \sup_u |P(a \geq u) - P(Y \geq U)| \quad (62)$$

be the maximum absolute different between the two cumulative probability distributions. Hodges et al. (Hodges & Cam 1960) showed that  $D \leq 2 \sum p_m^2$  and  $D \geq 3\sqrt[3]{\alpha}$  where  $\alpha = \max_m P_m$ .

In our network implementation,  $P_m = \frac{1}{C_W} f_{ij}$ . Therefore  $\sum P_m^2 \leq \frac{N_k}{C_W^2} \alpha$ . Since  $C_W$  should have the same order as the network size  $\mathcal{L}$ , the approximation becomes exact as

$\frac{N_k}{C_W^2} \rightarrow 0$ , which corresponds to a sparse spiking network (large  $C_W$  and  $\mathcal{L}$ ) with finite energy budget (finite  $N_k$ ). In this case,  $a$  has the distribution

$$P(a = u) = \frac{u^\lambda}{u!} \exp(-\lambda) \quad (63)$$

$\lambda = \sum P_m \leq \frac{N_k}{C_W} \alpha$ . Since  $N_k$  is finite, we have  $\lambda^2 \rightarrow 0$ .  $P(a \geq 1) = 1 - \exp(-\lambda) \rightarrow \lambda$ . This corresponds to equation 16 in the text.  $\lambda^2 \rightarrow 0$  also implies that  $P(a > 1) \rightarrow 0$ . The probability that the neuron receives more than one EPSP vanishes in the sparse network. This mechanism is similar to a winner-take-all (WTA) (Maass 2000) network, where multiple pre-synaptic neurons compete to activate one post-synaptic neuron.

## B Sensory neurons

The noisy measurement  $Z_{k+1}$  is not directly observed by the inference neurons, but sensed through another array of  $\mathcal{Z}$  sensory neurons, whose receptive fields are centered at  $z^i \in \mathbb{Z}, i = 1, \dots, \mathcal{Z}$ . Each sensory neuron  $i$  generates a Poisson spike train, with intensity proportional to  $h_i(Z_{k+1})$ . The probability that the  $i$ -th sensory neuron fires at time  $k + 1$  is proportional to  $h_i(Z_{k+1})$ , if the window of coincidence detection is small. Again we define the feedforward weight  $M_{ij}$  to be the neurotransmitter release probability between sensory neuron  $i$  and inference neurons in the  $j$ -th sub-population. A spiking sensory neuron  $i$  sends an EPSP to inference neurons in the  $j$ -th sub-population with probability  $M_{ij}$ . Therefore, as in equation 16, the probability that neurons in the  $j$ -th sub-population receive feedforward inputs at time  $k + 1$  can be approximated by

$$P(b_l^j(k+1) \geq 1) \simeq \sum_{i=1}^{\mathcal{Y}} M_{ij} h_i(Z_{k+1}) = g(Z_{k+1} = z^i | x^j) \quad (64)$$

Let  $G = \{G_{ij} = g(Z_{k+1} = z^i | x^j)\}$  be a  $\mathcal{Y}$  by  $\mathcal{X}$  matrix, and  $H = \{H_{ii'} = h_i(Z_{k+1} = z^{ii'})\}$  be a  $\mathcal{Y}$  by  $\mathcal{Y}$  matrix. Equation 64 implies  $G \propto H \times M$ . Therefore, the  $\mathcal{Y}$  by  $\mathcal{X}$  feedforward weight matrix  $M \propto H^{-1} \times G$ .

Typical choices of the ‘‘tuning curve’’ function  $h_i$  for sensory neurons are radial basis functions with a peak value at the center of the receptive field  $z^i$ , e.g., Gaussian  $h_i(Z_{k+1}) = h(Z_{k+1} - z^i) = \exp[-(Z_{k+1} - z^i)^2 / \sigma_Z^2]$  or cosine tuning functions. However, for simplicity, we may require only one sensory neuron responds exclusively to  $Z_{k+1}$  at time  $k + 1$  such that  $h_i(Z_{k+1}) = \chi(\frac{z^i + z^{i+1}}{2} \leq Z_{k+1} \leq \frac{z^i + z^{i+1}}{2})$ , where  $\chi(\cdot)$

denotes the indicator function. In this case, receptive fields of sensory neurons do not overlap with each other,  $H$  is an identity matrix, and  $M \propto G$ . Moreover, the approximation in equation 64 becomes exact. When  $Z_{k+1} \approx z^i$  arrives, only one sensory neuron centered at  $z^i$  is activated and fires one spike at time  $k$ . This pre-synaptic neuron then sends one feedforward EPSP randomly to every post-synaptic inference neuron with probability proportional to the likelihood:

$$P(b_l^i(k+1) = 1) = g((Z_{k+1}|x^i)/C_M) \quad (65)$$

where  $C_M$  is another scaling constant such that  $M_{ij} = g((Z_{k+1} = z^i|x^j)/C_M)$ .