LETTER **Communicated by Liam Paninski**

Active Data Collection for Efficient Estimation and Comparison of Nonlinear Neural Models

Christopher DiMattina
chris_dimattina@yahoo.com
Department of Neuroscience, Johns Hopkins University School of Medicine, Baltimore, MD 21205, U.S.A.

Kechen Zhang
kzhang4@jhmi.edu
Department of Biomedical Engineering, Johns Hopkins University School of Medicine, Baltimore, MD 21205, U.S.A.

The stimulus-response relationship of many sensory neurons is nonlinear, but fully quantifying this relationship by a complex nonlinear model may require too much data to be experimentally tractable. Here we present a theoretical study of a general two-stage computational method that may help to significantly reduce the number of stimuli needed to obtain an accurate mathematical description of nonlinear neural responses. Our method of active data collection first adaptively generates stimuli that are optimal for estimating the parameters of competing nonlinear models and then uses these estimates to generate stimuli online that are optimal for discriminating these models. We applied our method to simple hierarchical circuit models, including nonlinear networks built on the spatiotemporal or spectral-temporal receptive fields, and confirmed that collecting data using our two-stage adaptive algorithm was far more effective for estimating and comparing competing nonlinear sensory processing models than standard nonadaptive methods using random stimuli.

1 Introduction

Linear system identification methods are widely used to quantify the relationship between stimuli and neural responses in systems-level sensory neurophysiology (Wu, David, & Gallant, 2006), with the canonical example being the estimation of a linear receptive field from stimulus ensembles like random noise (Marmarelis & Marmarelis, 1978; Jones & Palmer, 1987; DiCarlo, Johnson, & Hsiao, 1998; Yu & Young, 2000) or natural stimuli

C. DiMattina is now at the Department of Electrical Engineering and Computer Science, Case Western Reserve University, Cleveland, OH 44106.

(Theunissen, Sen, & Doupe, 2000; David, Vinje, & Gallant, 2004). Since many sensory neurons have important nonlinear properties, various methods have been used to model nonlinear neurons, including recent examples such as quadratic analyses (Yu and Young, 2000; Rust, Schwartz, Movshon, & Simoncelli, 2005; Bandyopadhyay, Reiss, & Young, 2007; Chen, Han, Poo, & Dan, 2007), multilinear models (Ahrens, Linden, & Sahani, 2008; Ahrens, Paninski, & Sahani, 2008), and neural network models (Lau, Stanley, & Dan, 2002; Prenger, Wu, David, & Gallant, 2004; Cadieu et al., 2007). The vast majority of nonlinear modeling techniques are applied post hoc in offline analyses, and as a consequence, reliable parameter estimation often requires a large amount of data, and it is also impossible to directly test model predictions in real time during the course of an experiment.

In this study, we consider an online active learning approach to the problem of identifying nonlinear neurons, where stimuli are presented to sensory neurons in an adaptive manner, evolving with the neuron's response history so that the stimulus presented at each step is the most likely to be useful for recovering the network parameters given our current state of knowledge. This general approach has various names in the literature, such as optimal experimental design (Paninski, 2005; Benda, Gollisch, Machens, & Herz, 2007; Atkinson & Donev, 1992; Chaloner & Verdinelli, 1995), adaptive design optimization (Cavagnaro, Myung, Pitt, & Kujala, 2010), active learning (Cohn, Ghahramani, & Jordan, 1996), and query-based learning (Freund, Seung, Shamir, & Tishby, 1997).

Besides applications in machine learning (MacKay, 1992; Cohn et al., 1996; Freund et al., 1997; Sugiyama & Rubens, 2008) and psychophysics (Watson & Pelli, 1983; Kujala & Lukka, 2006), a recent study (Lewi, Butera, & Paninski, 2009; Lewi, Schneider, Woolley, & Paninski, 2011), has demonstrated an efficient information-theoretic algorithm for estimating generalized linear models (McCullagh & Nelder, 1989), also referred to as a linear-nonlinear Poisson (LNP) model, which has been used in many sensory neuroscience studies (Simoncelli, Paninski, Pillow, & Schwartz, 2004). The LNP models incorporate several useful biological features like spiking and spike history adaptation, and with gaussian inputs enjoy consistent estimation and log-convex likelihood functions so that they may be estimated without local minima (Paninski, 2004). Since the LNP model resembles a two-layer perceptron model, which has inherent computational limitations (Minsky & Papert, 1988), to model nonlinear sensory neurons in higher brain areas, a hierarchical network of LNP models would be needed (Schinkel-Bielefeld, David, Shamma, & Butts, 2010), but for such a network, the likelihood function is no longer guaranteed to always have a unique optimum.

In this letter, we focus on feedforward hierarchical neural networks or multilayer perceptrons (Rumelhart, Hinton, & McClelland, 1986), which are universal function approximators (Hornik, Stinchcombe, & White, 1989; Cybenko, 1989) with widespread applications in various disciplines.

Although these networks models are highly simplified, they are still biologically relevant as sensory processing models because of their resemblance to the general neural mechanisms, where complex response properties are built up from simpler responses at lower levels. In fact, these models have been used in numerous models of nonlinear sensory neurons (Zipser & Andersen, 1988; Lehky, Sejnowski, & Desimone, 1992; Riesenhuber & Poggio, 1999; Lau et al., 2002; Prenger et al., 2004; Cadieu et al., 2007; Hinton, 2010). Since optimal design methods for such models have never been applied to any online neurophysiological experiment, developing design algorithms based on these simplified models may have immediate benefit and might also help gain knowledge for handling more complicated models in the future. Because in practice neuroscientists often entertain several alternative hypotheses about the function of nonlinear sensory neurons, we will introduce a general two-stage procedure for estimating and comparing multiple competing models based on optimal experimental design in a Bayesian framework. In the model networks we analyze, we observe only the noisy activity of a neuron at the top, assuming that the activities of the neurons at the lower levels (hidden units) are unknown, not unlike the situation in most neurophysiological experiments. Since we know the true parameters of the model network, we will examine how accurately the model parameters are recovered from the stimulus-response data, as well as how accurately our method chooses the correct model from a set of candidate models.

2 Estimation of Nonlinear Network Models

In this section we use biologically inspired neural network models to show how we design stimuli to efficiently estimate the parameters of a single given model. The methods developed here are necessary for our consideration of comparison of multiple models in later sections.

2.1 Optimal Design Reduces Parameter Confounding: Center-Surround Network. We use the simple center-surround network model in Figure 1a to illustrate stimulus generation by optimal experimental design and demonstrate that this method permits accurate parameter recovery in neural networks, which are harder to identify using random noise stimuli due to the problem of continuous parameter confounding (DiMattina & Zhang, 2010). This generic circuit, with an excitatory neuron (E) and an inhibitory neuron (I), is inspired by the type II neuron with surround inhibition in the dorsal cochlear nucleus (Young & Davis, 2002). The response r of this network has Poisson distribution, and the mean response to input $\mathbf{x} = (x_1, \ldots, x_{21})$ is given by the rate model

$$f(\mathbf{x}, \boldsymbol{\theta}) = v_{\mathrm{E}} g_{\mathrm{E}} \left(w_{0\mathrm{E}} + \mathbf{w}_{\mathrm{E}} \cdot \mathbf{x} + v_{\mathrm{I}} g_{\mathrm{I}} \left(w_{0\mathrm{I}} + \mathbf{w}_{\mathrm{I}} \cdot \mathbf{x} \right) \right), \tag{2.1}$$

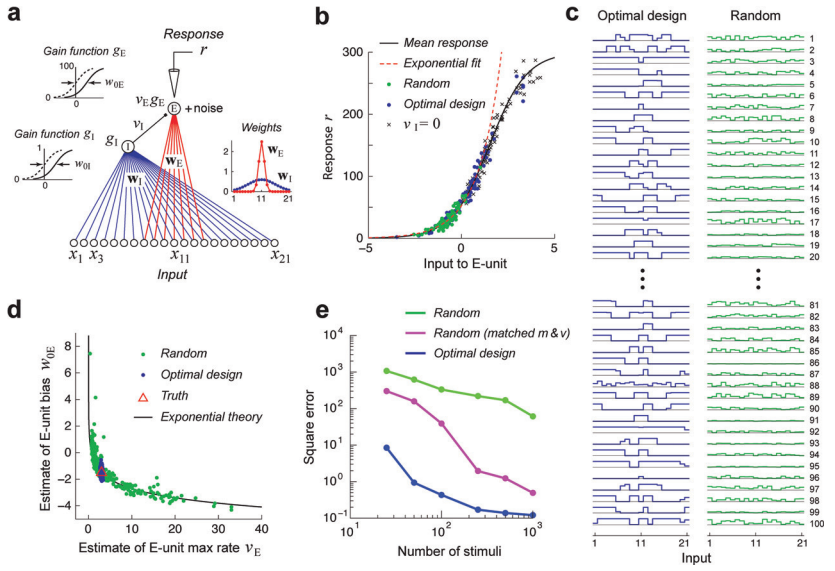


Figure 1: Optimally designed stimuli allow better estimation of neural network parameters than random stimuli. (a) The circuit implements center-surround inhibition via an interneuron (I) that inhibits the output (E) unit. Insets show the gain functions (left) and the input weights (right). (b) Optimally designed stimuli (left column in panel c) elicit a wide range of responses (blue dots). Random stimuli (right column in panel c) drive the responses (green dots) over a narrow region of the gain function (black curve), which is well approximated by an exponential (red dashed line); the response range increases after removing I-unit inhibition (black crosses). (c) Left column: Optimally designed stimulus sequence obtained by maximizing the D-optimal utility function (see Table 1). Right column: Random stimuli with independent amplitudes in the bins and random maximum amplitude between 0 and 1. (d) Parameter estimates (v_E, w_{0E}) attained from optimally designed stimuli (green dots) are clustered around the truth (red triangle), whereas the estimates from the random stimuli (blue dots) lie widely scattered along the curve given by equation 2.2 as predicted by the exponential confounding theory. See text for detail. Each dot represents the estimates from one experiment with 100 stimuli, and the experiment was repeated 300 times by Monte Carlo simulations. (e) Optimally designed stimuli lead to better estimates than the random stimuli, while the errors of all methods decrease with the number of stimuli. Constraining the random stimuli to have the same mean and variance as the optimally designed stimuli can improve performance, although they are still not as good as the optimally designed stimuli. Each dot represents the median square error of 50 Monte Carlo experiments.

Table 1: Summary of Parameters for Model in Figure 1.

Parameter	Description	Value
v_E	E-unit maximum rate	3
w_{0E}	E-unit bias	-1.5
A_E	E-unit weight amplitude	2.5
μ_E	E-unit weight center	11
σ_E	E-unit weight spread	1
v_I	I-unit output weight	-5
w_{0I}	I-unit bias	-2
A_I	I-unit weight amplitude	0.6
μ_I	I-unit weight center	11
σ_I	I-unit weight spread	5

where the gain functions are sigmoidal: $g_E(u) = 100/(1 + e^{-u})$ and $g_I(u) = 1/(1 + e^{-u})$, and the weights from the input layer are gaussian: $\mathbf{w}_E = A_E e^{(\mathbf{n}-\mu_E)^2/2\sigma_E^2}$ and $\mathbf{w}_I = A_I e^{(\mathbf{n}-\mu_I)^2/2\sigma_I^2}$ with $\mathbf{n} = (1, \ldots, 21)$. The model has 10 free parameters $\boldsymbol{\theta} \equiv (v_E, w_{0E}, A_E, \mu_E, \sigma_E, v_I, w_{0I}, A_I, \mu_I, \sigma_I)$, whose meanings and true values are given in Table 1 (see also Figure 1a).

These parameters can be estimated by fitting the model to any given stimulus-response data. Poorly chosen stimulus sets such as the random stimuli in Figure 1c yield poor parameter estimation, as shown by the wide scattering of the maximum-likelihood estimates of the output weight v_E and the bias w_{0E} in repeated simulations (see Figure 1d). This scattering is caused by the fact that only a very narrow range of the gain function is used, such that the effective gain function is approximately exponential. This leads to the confounding phenomenon: different combinations of parameters can lead to neural networks with almost identical input-output functionality (DiMattina & Zhang, 2010). More specifically, the strong inhibition of the I-unit (see Figures 1a and 1b) allows the random stimuli to elicit only small responses from the E-unit. When restricted to low activation, the E-unit gain function can be approximated by an exponential: $g_E(u) \approx Ae^{\alpha u}$ (see Figure 1b), so that we can rewrite equation 2.1 as $f = v_E g_E(w_{0E} + z) \approx Av_E e^{\alpha w_{0E}} e^{\alpha z}$. Thus, we can choose different values of parameters v_E and w_{0E} but still obtain nearly identical overall input-output transformation of the network, provided that parameters v_E and w_{0E} satisfy the equation

$$v_E e^{\alpha w_{0E}} = \text{const}. \tag{2.2}$$

In other words, the theory predicts that there will be confounds between the bias (w_{0E}) and output weight (v_E) of a hidden unit if the stimuli probe the gain functions only within the range where it is approximately exponential.

Continuous parameter confounding, where changing parameters continuously leave the outputs of a neural network unaltered, is a common phenomenon that takes many different forms (DiMattina & Zhang, 2010). The scattering due to the confounding should be reduced automatically by optimal experimental design because it generates stimuli that yield the most accurate parameter estimation. The optimally designed stimuli in Figure 1c are generated one by one by optimizing a utility function, and they lead to estimates showing very little spread around the truth in repeated experiments (see Figure 1d). A small number of optimally designed stimuli can be equally effective for parameter estimation as a much larger number of random stimuli (see Figure 1e). Note how the optimally designed stimuli contain a great deal of complex correlated structure (see Figure 1c) and how consecutive stimuli tend to be different from one another. Although for a given network, one may use intuitive or heuristic ideas to look for efficient stimuli, the optimal design method provides a principled approach to find the most efficient stimulus set for parameter estimation, automatically taking into account network architecture and response history.

Compared with the random stimuli where the amplitudes of different bins are completely uncorrelated, the optimally designed stimuli appear to have more complex and correlated structures, which are responsible for their effectiveness (see Figure 1c). As is clear from Figure 1b, the optimally designed stimuli do not always drive the network to the highest response. Instead, the outputs appear to span the entire range, from very small to very large. So an optimally designed stimulus is generally different from an optimal stimulus, which commonly refers to the stimulus that elicits the maximum response (DiMattina & Zhang, 2008). We have also used random stimuli whose amplitudes are uncorrelated but have segmented uniform distribution so as to match the mean (0.4357) and variance (0.2301) of the optimally designed stimuli. The performance was improved but still below that of the optimally designed stimuli. Here the segmented uniform distribution had a constant probability within intervals $[0, a]$ and $[b, 1]$, but 0 probability between a and b ($a = 0.0360$ and $b = 0.9727$).

The optimally designed stimuli in Figure 1 were generated by maximizing an expected utility $U_{n+1}^{(\mathrm{E})}(\mathbf{x})$ quantifying how useful we expect stimulus $\mathbf{x}$ to be for estimating the parameter set $\boldsymbol{\theta}$ (Chaloner & Verdinelli, 1995). The superscript E (for estimation phase) is used to distinguish from the comparison phase in later sections. At iteration $n+1$ ($n = 0, 1, 2, \ldots$), the stimulus is given by

$$\mathbf{x}_{n+1} = \arg\max_{\mathbf{x}} U_{n+1}^{(\mathrm{E})}(\mathbf{x}), \tag{2.3}$$

with

$$U_{n+1}^{(\mathrm{E})}(\mathbf{x}) = \int u_n^{(\mathrm{E})}(\mathbf{x} \mid \boldsymbol{\theta})\, p_n(\boldsymbol{\theta}) d\boldsymbol{\theta}, \tag{2.4}$$

Table 2: Some Utility Functions for Model Estimation.

Design	Utility $u^{(\mathrm{E})}(\mathbf{x} \mid \boldsymbol{\theta})$	Interpretation
D-optimal	$\det \mathbf{F}_{n+1}(\mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{x}, \boldsymbol{\theta})$	Inverse volume of covariance ellipsoid
A-optimal	$-\mathrm{trace}\, \mathbf{F}_{n+1}^{-1}(\mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{x}, \boldsymbol{\theta})$	Sum of minimum variance of parameters
Mutual information	$D_{\mathrm{KL}}[p(r \mid \mathbf{x}, \boldsymbol{\theta}), p(r \mid \mathbf{x})]$	Information between data and parameters
Square loss	$-\langle \lvert\lvert \boldsymbol{\theta} - \mathrm{E}[\boldsymbol{\theta} \mid \mathbf{x}, r] \rvert\rvert^2 \rangle_r$	Direct squared error

where $u_n^{(\mathrm{E})}(\mathbf{x} \mid \boldsymbol{\theta})$ denotes the conditional expected utility, which quantifies how useful $\mathbf{x}$ is for estimating $\boldsymbol{\theta}$ given the n previous observations $\{(\mathbf{x}_i, r_i)\}_{i=1}^n$ and true (unknown) parameter value $\boldsymbol{\theta}$. Here $p_n(\boldsymbol{\theta})$ is the posterior distribution of parameters, which depend implicitly on the data $\{(\mathbf{x}_i, r_i)\}_{i=1}^n$. The initial distribution $p_0(\boldsymbol{\theta})$ is the Bayesian prior.

There is a wide variety of choices for $u_n^{(\mathrm{E})}(\mathbf{x} \mid \boldsymbol{\theta})$ in the literature (Atkinson & Donev, 1992; Chaloner & Verdinelli, 1995), some of which are listed in Table 2. For this example we employ D-optimal design, which minimizes the expected volume of the covariance ellipsoid by maximizing the expected value of the determinant of the Fisher information matrix $\mathbf{F}_{n+1}(\mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{x}, \boldsymbol{\theta})$, which is updated recursively by

$$\mathbf{F}_{n+1}(\mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{x}, \boldsymbol{\theta}) = \mathbf{F}_n(\mathbf{x}_1, \ldots, \mathbf{x}_n, \boldsymbol{\theta}) + \frac{1}{\nu} \nabla_{\boldsymbol{\theta}} f(\mathbf{x}, \boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}} f(\mathbf{x}, \boldsymbol{\theta})^{\mathrm{T}} \quad (2.5)$$

(Atkinson & Donev, 1992), starting from an identity matrix $\mathbf{F}_0 = \mathbf{I}$. Here the variance $\nu = f(\mathbf{x}, \boldsymbol{\theta})$ for Poisson noise and $\nu = \sigma^2$ for gaussian noise. The utility $u_n^{(\mathrm{E})}(\mathbf{x} \mid \boldsymbol{\theta}) = \det \mathbf{F}_{n+1}(\mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{x}, \boldsymbol{\theta})$ for D-optimal design is equivalent to

$$u_n^{(\mathrm{E})}(\mathbf{x} \mid \boldsymbol{\theta}) = \frac{1}{\nu} \nabla_{\boldsymbol{\theta}} f(\mathbf{x}, \boldsymbol{\theta})^{\mathrm{T}} \mathbf{F}_n^{-1}(\mathbf{x}_1, \ldots, \mathbf{x}_n, \boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}} f(\mathbf{x}, \boldsymbol{\theta}) \quad (2.6)$$

(Atkinson & Donev, 1992). The example in Figure 1 is based on equations 2.3 to 2.6 with particle filter approximation of the posterior (see section 3.1.2 for further detail).

2.2 Three-Layer Subunit Network. In this section, we consider the network in Figure 2a, where the final response is a nonlinear combination of the outputs of two linear subunits that resemble the spatiotemporal or spectral-temporal receptive fields (STRF). Here the nonlinearity in the system comes from the gain functions in the network rather than from nonlinear preprocessing (David & Gallant, 2005; Ahrens, Paninski, & Sahani, 2008). The network performs a supralinear detection of a specific feature

conjunction (see Figure 2b). This network contains a large number of parameters because each pixel in the STRF weight patterns is an independent parameter. We use this example to illustrate another estimation utility function based on mutual information or entropy (see Table 2).

This network is equivalent to a three-layer perceptron (Rumelhart et al., 1986), whose mean response to stimulus $\mathbf{x}$ is given by

$$f(\mathbf{x}, \boldsymbol{\theta}) = h\left(\sum_{i=1}^{m} v_i g(\mathbf{w}_i^{\mathrm{T}}\mathbf{x} + w_{0i})\right), \tag{2.7}$$

where we have $m = 2$ hidden units with the sigmoidal gain function $g(u) = (1 + e^{-u})^{-1}$ and the output unit has an exponential gain function $h(u) = e^u$ (see Figure 2a, inset). The input weights $\mathbf{w}_1$ and $\mathbf{w}_2$ to the two hidden subunits define their preferred stimuli and are based on Gabor function

$$w_i(x, y) = A_i \exp\left(-\frac{u_i^2}{2\sigma_{ix}^2} - \frac{{v_i}^2}{2\sigma_{iy}^2}\right)\cos\left(\frac{2\pi u_i}{T_i} + \varphi_i\right), \tag{2.8}$$

where

$$u_i(x, y) = (x - \mu_{ix})\cos\theta_i + (y - \mu_{iy})\sin\theta_i, \tag{2.9}$$

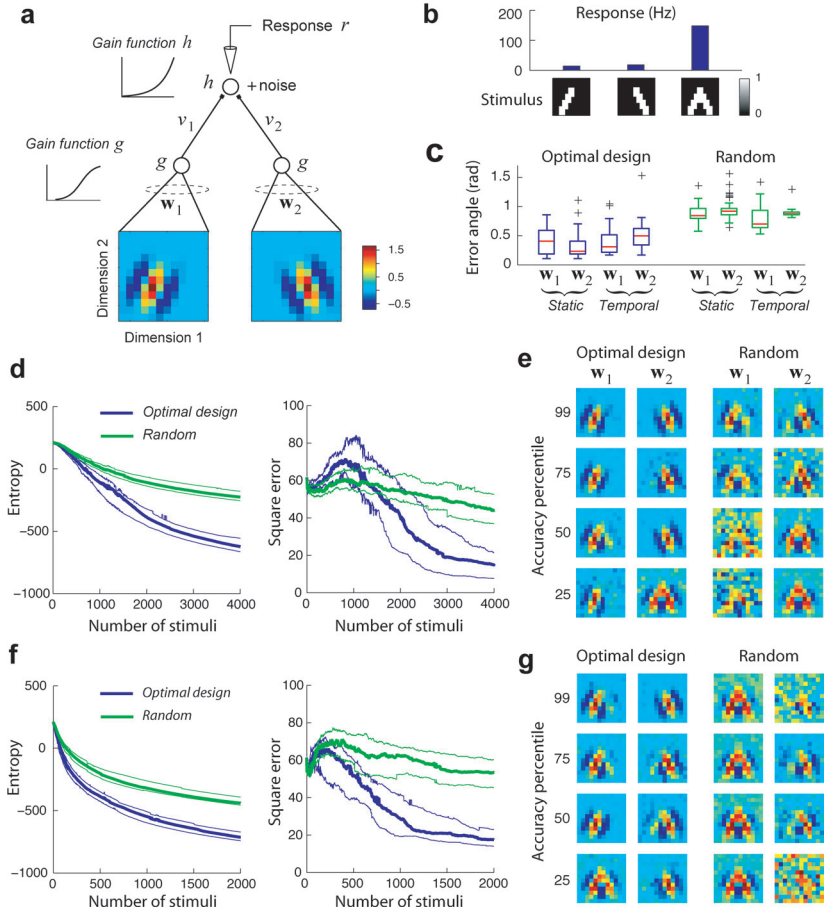
$$v_i(x, y) = -(x - \mu_{ix})\sin\theta_i + (y - \mu_{iy})\cos\theta_i, \tag{2.10}$$

with $i = 1$ or 2, and the parameters are given in Table 3. Vector $\mathbf{w}_1$ and $\mathbf{w}_2$ are made from discrete 12×12 sample of the Gabor patterns, then concatenated as column vectors, and finally normalized by $||\mathbf{w}_1|| = ||\mathbf{w}_2|| = 4$. The parameter set $\boldsymbol{\theta} = \{v_1, v_2, w_{01}, w_{02}, \mathbf{w}_1, \mathbf{w}_2\}$ contains 292 free parameters.

The representation of a posterior that evolves as new observations are made is fundamental to the Bayesian method. The posterior density was approximated by a particle filter for the example in Figure 1, which has a small number of parameters, but due to the high dimensionality of the model in Figure 2, we use an alternative approximation of the posterior as a sum of gaussians:

$$p_n(\boldsymbol{\theta}) = \sum_{j=1}^{K} \alpha_n^{(j)} \mathcal{N}\left(\boldsymbol{\theta} \mid \boldsymbol{\mu}_n^{(j)}, \boldsymbol{\Sigma}_n^{(j)}\right), \tag{2.11}$$

with means $\boldsymbol{\mu}_n^{(j)}$ and covariances $\boldsymbol{\Sigma}_n^{(j)}$. Given a novel stimulus-response observation, we approximate the Poisson noise by a gaussian model with variance equal to the mean, and then recursively update the posterior density using the extended Kalman filter equations (Alspach & Sorenson, 1972;



Haykin, 2001; Hering & Simandl, 2007). Following Paninski (2005), our expected utility function is based on mutual information or negative differential entropy:

$$\begin{aligned} U_{n+1}^{(\mathrm{E})}(\mathbf{x}) &= -\int H\left[p_{n+1}(\boldsymbol{\theta})\right] p(r \mid \mathbf{x}) dr \\ &\approx \ln\left(1 + \frac{1}{\nu}\mathbf{a}_n^{(k)}(\mathbf{x})^{\mathrm{T}}\boldsymbol{\Sigma}_n^{(k)}\mathbf{a}_n^{(k)}(\mathbf{x})\right), \end{aligned} \tag{2.12}$$

where H is the differential entropy of the posterior, $\mathbf{a}_n^{(k)}(\mathbf{x}) = \nabla_{\boldsymbol{\theta}} f(\mathbf{x}, \boldsymbol{\mu}_n^{(k)})$ is the sensitivity function (Cohn, 1996), $\nu = f(\mathbf{x}, \boldsymbol{\mu}_n^{(k)})$ for Poisson noise or $\nu = \sigma^2$ for gaussian noise, and the approximation in the second step follows from

equation 2.11 by assuming that the probability mass lies almost entirely on the highest peak $k \equiv \arg\max_j \{\alpha_n^{(j)}\}$. (See sections 3.2.1 and 3.2.2 for details.)

As shown in Figure 2, optimally designed stimuli that maximize equation 2.12 yield far more accurate estimates of the parameters than random stimuli (see Figures 2d and 2e). These stimuli were chosen from a finite set of stimuli generated using a heuristic in order to minimize computation time (section 3.2.3).

To extend our neural network formulation to time-varying stimulus $\mathbf{x}(t)$, we may describe the time-varying response $r(t)$ as a Poisson process with the time-varying rate given by

$$f(\mathbf{x}(t), \boldsymbol{\theta}) = h\left(\sum_{i=1}^{m} v_i g\left(\int_0^\infty \mathbf{w}_i^{\mathrm{T}}(\tau)\mathbf{x}(t-\tau)d\tau + w_{0i}\right)\right), \tag{2.13}$$

where the weight $\mathbf{w}_i(\tau)$ is the spatiotemporal kernel that defines the time-varying input to the ith subunit. In the special case of a single subunit ($m = 1$) and linear gain functions, equation 2.13 reduces to the well-known linear STRF model. This optimal design procedure is readily generalized to time-varying stimuli $\mathbf{x}(t)$ that give rise to time-varying responses $r(t)$. In Figures 2f and 2g, dimension 1 (horizontal axis) is given a temporal

---

Figure 2: Optimal design for nonlinear network with linear subunits. (a) Hierarchical network model of a hypothetical sensory neuron that is selective for a specific feature conjunction. Color images show the input weights (12×12 bins) of the two subunits. (b) The response of the network to the conjunction of stimulus features is nonlinear, that is, far greater than the sum of the responses to either stimulus feature alone. (c) Box plot of the angle between the true weight vector $\mathbf{w}_1$ (or $\mathbf{w}_2$) and its estimate obtained using various methods as explained in panels d–g. Here "static" corresponds to the case in panels d and e, and "temporal" corresponds to the case in panels f and g. In all cases, optimally designed stimuli yielded more accurate estimation than random stimuli. (d) Left: Optimally designed stimuli (blue curves) that minimize the posterior entropy perform much better than random stimuli (green curves) for any given number of stimuli. Thick lines indicate the median and thin lines indicate the 25th and 75th percentiles of 50 Monte Carlo experiments. Entropy was computed using the bound in expression 3.31. Right: The same data shown by the square error. (e) Estimated input weights of models with parameter estimation accuracy in the 99th, 75th, 50th, and 25th percentiles of the 50 Monte Carlo experiments, showing that the optimally designed stimuli can recover the weights more accurately than the random stimuli. (f) Same as in panel d but for time-varying stimuli and responses. In each of the 25 Monte Carlo experiments, each stimulus was repeated 20 times to attain a reliable estimate of the time-varying firing rate. (g) Same as in panel e but for spatiotemporal filters.

Table 3: Summary of Parameters for the Models in Figures 2 and 6.

Parameter	Description	Value
v_1, v_2	Subunit 1, 2 output weights	2.5, 2.5
w_{01}, w_{02}	Subunit 1, 2 bias	$-1.5, -2.5$
$\mathbf{w}_1, \mathbf{w}_2$	Subunit 1, 2 input weights	12×12 Gabor pattern
μ_{1x}, μ_{2x}	Gabor centers in x-dimension	7, 4
σ_{1x}, σ_{2x}	Gabor spreads in x-dimension	2, 2
μ_{1y}, μ_{2y}	Gabor centers in y-dimension	5, 5
σ_{1y}, σ_{2y}	Gabor spreads in y-dimension	1.8, 1.8
A_1, A_2	Gabor-amplitudes	1.7, 1.7
T_1, T_2	Gabor period	5, 5
φ_1, φ_2	Gabor phase shifts	0, 0
θ_1, θ_2	Gabor orientation	$3\pi/8, -3\pi/8$

Note: The weight vectors $\mathbf{w}_1$ and $\mathbf{w}_2$ are described by equation 2.8.

interpretation of 20 msec time interval per bin. Here both the weight patterns and the stimulus patterns have 12×12 bins each. We pad the beginning and end of a stimulus pattern with 0s and slide the weight bins across the padded pattern to generate a sequence of elicited responses. The corresponding sequence of static stimuli (the segment of stimulus pattern falling within the weight bins) can be used to update the Kalman filter as before. The only caveat is that because the instantaneous firing rate is computed by observing spike counts in a short time window, one may need to present several repetitions of the time-varying stimulus to get an accurate estimation of $r(t)$. Here averaging over 20 repeated stimulus presentations was used. We see in this case that the optimal design also performs better than random stimuli (see Figures 2f and 2g).

To quantify the error of estimation, we computed the angle between the true weight vector $\mathbf{w}_i$ and its estimate $\hat{\mathbf{w}}_i$, namely, angle $= \arccos(\mathbf{w}_i \cdot \hat{\mathbf{w}}_i / ||\mathbf{w}_i|| \, ||\hat{\mathbf{w}}_i||)$. As shown in Figure 2c, the error angles obtained from optimally designed stimuli were always significantly better than that obtained from random stimuli in all cases we tested (Wilcoxon rank-sum test: $p < 4 \times 10^{-15}$ in static cases and $p < 2 \times 10^{-5}$ in spatiotemporal cases).

We also performed another analysis in order to determine the extent to which the estimated weight vectors $\hat{\mathbf{w}}_1$ and $\hat{\mathbf{w}}_2$ lie in the plane spanned by the true vectors $\mathbf{w}_1$ and $\mathbf{w}_2$. We computed the projection $\boldsymbol{\pi}_i = (\hat{\mathbf{w}}_i \cdot \mathbf{u}_1)\mathbf{u}_1 + (\hat{\mathbf{w}}_i \cdot \mathbf{u}_2)\mathbf{u}_2$, where $\mathbf{u}_1$ and $\mathbf{u}_2$ are an orthonormal basis for span$(\mathbf{w}_1, \mathbf{w}_2)$, and then defined an index $p_i = ||\boldsymbol{\pi}_i|| \, / \, ||\hat{\mathbf{w}}_i||$. By definition, we have $0 \leq p_i \leq 1$, with $p_i = 1$ indicating that $\hat{\mathbf{w}}_i$ lies entirely in span$(\mathbf{w}_1, \mathbf{w}_2)$ and $p_i = 0$, indicating that $\hat{\mathbf{w}}_i$ is orthogonal to this plane. Applying this analysis to the

numerical experiments for the static case in Figure 2, we found that the median indices for the optimally designed stimuli ($p_1 = 0.98$, $p_2 = 0.98$) were significantly greater than that for the random stimuli ($p_1 = 0.90$, $p_2 = 0.93$; Wilcoxon rank-sum test, $p < 1 \times 10^{-7}$), indicating a tendency for the weights recovered by optimal design to lie closer to the plane spanned by the true weight vectors. However, because the network is nonlinear and the input signals weighted by $\mathbf{w}_1$ and $\mathbf{w}_2$ have to go through the nonlinear gain function g separately (see Figure 2a), it is generally not sufficient for the estimated weights to simply lie in the plane spanned by the true weight vectors in order to yield accurate predictions, as would be the case if the gain g were linear. In the next section, we verify directly that parameter estimation by optimal design provides more accurate predictions in novel situations compared to random stimuli.

2.3 Accurate Estimation Helps Accurate Prediction. As we see from the examples in Figures 1 and 2, optimally designed stimuli allow the parameters of the neural networks to be estimated more accurately than random stimuli. However, in experimental applications, the success of a model is measured by how well it predicts the responses to novel stimuli (Wu et al., 2006). A model with wrong parameter values might still make accurate predictions, especially when diverse values of confounded parameters may yield nearly identical input-output relations of neural networks, at least over a restricted set of stimuli (DiMattina & Zhang, 2010).

To test this possibility directly using the simple network shown in Figure 1a, we compared the predictions of the estimated models against the responses by the true model to a novel stimulus set comprising of 95 spot stimuli (see Figure 3a inset), each with a small localized activity bump 3 receptors wide, centered at location $i = 2, \ldots, 20$ and with amplitude of 0.2, 0.4, 0.6, 0.8, 1. All responses were computed without adding noise. As shown in Figure 3b, the median square error of $n = 300$ models estimated using optimally designed stimuli was significantly better than that of $n = 300$ models estimated using random stimuli (Wilcoxon rank-sum test, $p = 1.5 \times 10^{-95}$). Figure 3a shows two representative models whose square errors are equal to the medians of the two groups. The model estimated using optimal design does a nearly perfect job of predicting the true responses, whereas the model estimated using random stimuli makes poor predictions for large responses.

To explain intuitively why the parameters obtained from random stimuli have yielded poorer predictions, we notice that diverse values of confounded parameters can mimic the same true input-output relationship only with respect to a restricted stimulus set (here, the random stimuli), which drives the neuron over a small dynamic range so that the gain function (a sigmoid) is well approximated by a specific function (here an exponential). For novel stimuli that can drive the neuron over a much wider range of firing rate, the approximation is no longer valid, and the actual

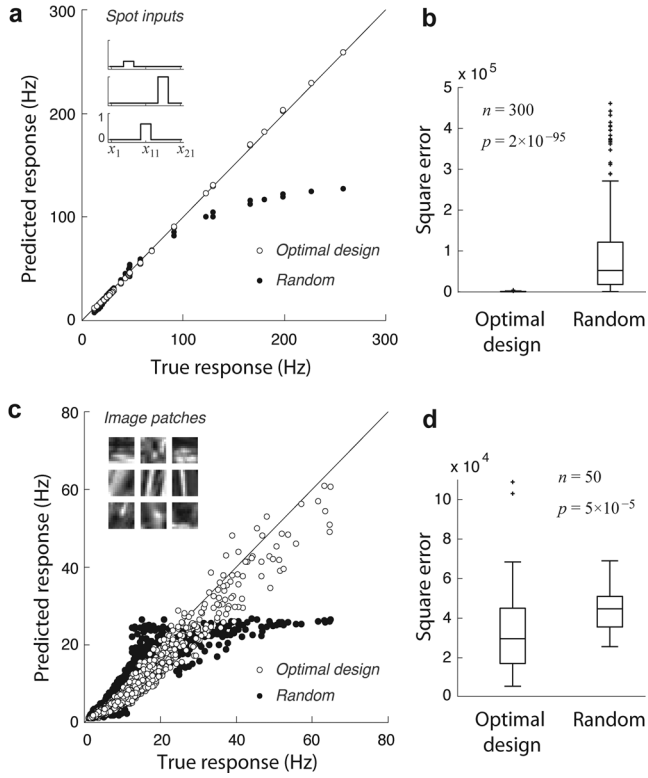


Figure 3: Accurate parameter estimation is important for accurately predicting the responses to novel stimuli. (a) The true model is as shown in Figure 1a. When tested on a set of novel spot stimuli (inset), a representative model obtained from optimally designed stimuli made predictions much closer to the truth than a representative model obtained from random stimuli. Each of the two representative models was selected as the one with median square error from $n = 300$ models obtained from repeated trials. (b) Box plot showing that the models ($n = 300$) estimated using optimally designed stimuli do a significantly better job of predicting the responses than the models ($n = 300$) estimated using random stimuli. (c) Same as panel a, but for the true model shown in Figure 2a, and with novel test stimuli being a set of natural image patches (examples shown in inset). (d) Same as panel b, but for the model in Figure 2a.

responses follow the fixed sigmoidal gain function, which now deviates significantly from the approximating function (e.g., exponential), which allows parameter confounding. Now diverse values of the parameters can no longer lead to the same input-output relationship and, hence, the poor predictions. For example, the saturation in the predicted responses in

Figure 3a (random stimuli) is due to underestimation of the output weight and overestimation of the confounded bias, so that the model predicts, wrongly, that responses should saturate at large input. By contrast, the parameters obtained from optimally designed stimuli are close to the truth and therefore should make good predictions for any novel stimuli.

Similarly, for the network model in Figure 2a, we interpreted it as a visual neuron and tested it on a novel set of 3000 natural image patches (see Figure 3c inset) taken from an existing image database (van Hateren & van der Schaaf, 1998). As shown in Figure 3d, the median square error of $n = 50$ models estimated using optimal design was significantly less than that of $n = 50$ models estimated using random stimuli (Wilcoxon rank-sum test, $p = 5.3 \times 10^{-5}$). Figure 3c shows two representative models with median squared errors. Although both models can account for small-amplitude responses, the model estimated using random stimuli systematically underpredicts large responses, whereas the model estimated using optimal design does a much better job. Based on these results, we conclude that in our examples in Figures 1 and 2, accurate estimation of parameters allows better predictions in novel situations.

2.4 Choice of Utility Functions. In the examples shown in Figures 1 and 2, we made use of two different utility functions for optimal stimulus design. Here we briefly point out how they differ and justify our use of each function, while leaving their derivations to the next section.

The D-optimal utility function, equation 2.6, used for the example in Figure 1 is based on the asymptotic approximation of the posterior density as a gaussian whose variance is the inverse Fisher information matrix (see the next section). Utility functions like D-optimal and A-optimal design (see Table 2) based on the asymptotic gaussian assumption permit faster computation since they do not involve integration over possible responses (Muller & Parmigiani, 1995). The D-optimality criterion has been widely used for its simplicity and ease of implementation (Atkinson & Donev, 1992), and we chose it for the same reason. By contrast, the mutual information criterion (Paninski, 2005; Lewi et al., 2009) and square error criteria (Muller & Parmigiani, 1995) directly optimize the stimulus with respect to the current posterior density and do not rely on approximations, although their evaluation requires integration over all possible responses and are thus computationally more expensive. We adopt the mutual information criterion for the example in Figure 2 and use a gaussian sum approximation under several simplifying assumptions in order to speed up computation. Although various optimality criteria look distinctive, they are often asymptotically related (Atkinson & Donev, 1992; Chaloner & Verdinelli, 1995) and therefore do not present mutually exclusive choices. Although our choice and implementation of utility functions appear adequate for our problems, it is not our intention in this letter to show that they are better than

other alternatives. More systematic comparison would be helpful in future studies.

3 Implementation of Optimal Design

In this section, we present the details of the implementation of optimal experimental design for the examples illustrated in Figures 1 and 2.

3.1 Center-Surround Model Implementation

3.1.1 Derivation of Utility Function. The center-surround network in Figure 1 has Poisson distributed responses, and the mean response to stimulus $\mathbf{x}$ is specified by the tuning function $f(\mathbf{x}, \boldsymbol{\theta})$. Let the first n observations of the stimulus-response data be $\mathcal{D}_n = \{(\mathbf{x}_1, r_1), \ldots, (\mathbf{x}_n, r_n)\}$. The likelihood of the data is given by

$$p(\mathcal{D}_n \mid \boldsymbol{\theta}) = \prod_{i=1}^{n} \frac{f(\mathbf{x}_i, \boldsymbol{\theta})^{r_i}}{r_i!} e^{-f(\mathbf{x}_i, \boldsymbol{\theta})}. \tag{3.1}$$

Alternatively, if the response has gaussian noise, the likelihood becomes

$$p(\mathcal{D}_n \mid \boldsymbol{\theta}) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} e^{-(r_i - f(\mathbf{x}_i, \boldsymbol{\theta}))^2/2\sigma^2}. \tag{3.2}$$

The posterior of the parameters follows Bayes' rule,

$$p(\boldsymbol{\theta} \mid \mathcal{D}_n) = \frac{p(\mathcal{D}_n \mid \boldsymbol{\theta}) p_0(\boldsymbol{\theta})}{\int p(\mathcal{D}_n \mid \boldsymbol{\theta}) p_0(\boldsymbol{\theta}) d\boldsymbol{\theta}}, \tag{3.3}$$

where $p_0(\boldsymbol{\theta})$ is the prior. We write the posterior as $p_n(\boldsymbol{\theta}) \equiv p(\boldsymbol{\theta} \mid \mathcal{D}_n)$.

The optimally designed stimuli in Figure 1 is based on the D-optimality criterion (Atkinson & Donev, 1992), which chooses at each step the stimulus $\mathbf{x}$ that optimizes the expected utility,

$$U_{n+1}^{(\mathrm{E})}(\mathbf{x}) = \int |\mathbf{F}_{n+1}(\mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{x}, \boldsymbol{\theta})| p_n(\boldsymbol{\theta}) d\boldsymbol{\theta}, \tag{3.4}$$

where $\mathbf{F}_{n+1}(\mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{x}, \boldsymbol{\theta})$ is the Fisher information matrix and $|\cdot|$ denotes the determinant. The Fisher information matrix is defined by

$$\mathbf{F}_n(\mathbf{x}_1, \ldots, \mathbf{x}_n, \boldsymbol{\theta}) = -\left\langle \nabla_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}} \ln p(\mathcal{D}_n \mid \boldsymbol{\theta}) \right\rangle, \tag{3.5}$$

where the average $\langle\cdot\rangle$ is over all possible responses of all n observations, $\nabla_{\boldsymbol{\theta}} = \partial/\partial\boldsymbol{\theta}$, and $p(\mathcal{D}_n \mid \boldsymbol{\theta})$ is given by either equation 3.1 for Poisson noise or equation 3.2 for gaussian noise. The rationale behind equation 3.4 is that the posterior, equation 3.3, is asymptotically a normal distribution whose covariance is given by the inverse Fisher information matrix (van der Vart, 1998). Therefore, intuitively, a set of stimuli that maximizes the Fisher information should minimize the uncertainty, or spread, of our parameter estimates obtained from the posterior parameter distribution.

Since we assume in equations 3.1 and 3.2 that the responses to different stimulus presentations are independent, the Fisher information matrix is additive with respect to the observations and obeys the recursive formula,

$$\mathbf{F}_{n+1}(\mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{x}, \boldsymbol{\theta}) = \mathbf{F}_n(\mathbf{x}_1, \ldots, \mathbf{x}_n, \boldsymbol{\theta}) + \frac{1}{\nu}\nabla_{\boldsymbol{\theta}} f(\mathbf{x}, \boldsymbol{\theta})\nabla_{\boldsymbol{\theta}} f(\mathbf{x}, \boldsymbol{\theta})^{\mathrm{T}}, \quad (3.6)$$

where $\nu = f(\mathbf{x}, \boldsymbol{\theta})$ for Poisson noise and $\nu = \sigma^2$ for gaussian noise (Kay, 1993). Since

$$|\mathbf{A} + \mathbf{v}\mathbf{v}^{\mathrm{T}}| = |\mathbf{A}|\left(1 + \mathbf{v}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{v}\right) \quad (3.7)$$

holds for an arbitrary invertible matrix $\mathbf{A}$ and column vector $\mathbf{v}$ (Harville, 1997), we obtain

$$\begin{aligned} |\mathbf{F}_{n+1}(\mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{x}, \boldsymbol{\theta})| &= |\mathbf{F}_n(\boldsymbol{\theta})| \\ &\quad \times \left(1 + \frac{1}{\nu}\nabla_{\boldsymbol{\theta}} f(\mathbf{x}, \boldsymbol{\theta})^{\mathrm{T}}\mathbf{F}_n^{-1}(\boldsymbol{\theta})\nabla_{\boldsymbol{\theta}} f(\mathbf{x}, \boldsymbol{\theta})\right), \end{aligned} \quad (3.8)$$

where $\mathbf{F}_n(\boldsymbol{\theta}) \equiv \mathbf{F}_n(\mathbf{x}_1, \ldots, \mathbf{x}_n, \boldsymbol{\theta})$ (Atkinson & Donev, 1992). Thus, maximizing the utility function in equation 3.4 is equivalent to maximizing the following utility function,

$$U_{n+1}^{(\mathrm{E})}(\mathbf{x}) = \int \frac{1}{\nu}\left[\nabla_{\boldsymbol{\theta}} f(\mathbf{x}, \boldsymbol{\theta})^{\mathrm{T}}\mathbf{F}_n^{-1}(\boldsymbol{\theta})\nabla_{\boldsymbol{\theta}} f(\mathbf{x}, \boldsymbol{\theta})\right] p_n(\boldsymbol{\theta}) d\boldsymbol{\theta}, \quad (3.9)$$

which was used for the example in Figure 1.

3.1.2 Particle Filter Approximation. To practically evaluate our D-optimal utility function, equation 3.9, we make use of a particle filter representation (Gordon, Salmond, & Smith, 1993; Carpenter, Clifford, & Fearnhead, 1999; Gramacy & Polson, 2009) of the continuous posterior $p_n(\boldsymbol{\theta})$ defined on

particles $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N\}$, and we denote this discrete density by $p_n(\boldsymbol{\theta}_i)$ with $\sum_{i=1}^{N} p_n(\boldsymbol{\theta}_i) = 1$. Now the utility function 3.9 is approximated by

$$U_{n+1}^{(\mathrm{E})}(\mathbf{x}) = \sum_{i=1}^{N} \left(\frac{1}{\nu} \nabla_{\boldsymbol{\theta}} f(\mathbf{x}, \boldsymbol{\theta}_i)^{\mathrm{T}} \mathbf{F}_n^{-1}(\boldsymbol{\theta}_i) \nabla_{\boldsymbol{\theta}} f(\mathbf{x}, \boldsymbol{\theta}_i) \right) p_n(\boldsymbol{\theta}_i), \tag{3.10}$$

where $\nu = f(\mathbf{x}, \boldsymbol{\theta}_i)$ for Poisson noise and $\nu = \sigma^2$ for gaussian noise. We used $\nu = f(\mathbf{x}, \boldsymbol{\theta}_i)$ and $N = 250$ particles and a uniform prior $p_0(\boldsymbol{\theta})$ with the initial set of particles ($p_0(\boldsymbol{\theta}_i) = 1/N$) drawn uniformly from $\boldsymbol{\theta} \pm |\boldsymbol{\theta}|/2$. (For comparison, we also used $\boldsymbol{\theta} \pm |\boldsymbol{\theta}|$ as a broader prior distribution, and the results of the optimal design method were similar, with a moderate increase of the median square error from 0.19 to 0.32. In contrast, the median square error of estimation using random stimuli increased from 8.1 to 141.) Since the continuous prior $p_0(\boldsymbol{\theta})$ was uniform over the range of permissible particle locations, it did not figure into the recursive Bayesian update equation 3.3. For the results in Figure 1d, we performed 300 Monte Carlo experiments with 100 stimuli per experiment (either random or optimally designed by optimizing equation 3.10). At the end of each experiment, a final estimate was attained by maximizing the log-likelihood function,

$$L(\mathcal{D}_n \mid \boldsymbol{\theta}) = -\sum_{i=1}^{n} \frac{1}{2r_i} (r_i - f(\mathbf{x}, \boldsymbol{\theta}_i))^2 - \ln \sqrt{2\pi r_i}, \tag{3.11}$$

which follows from approximating equation 3.1 with a gaussian whose variance is equal to the mean. The optimization uses the Matlab function *fmincon*, with search bounds $\boldsymbol{\theta} \pm 10|\boldsymbol{\theta}|$, and the search initialized at the particle with the highest probability:

$$\boldsymbol{\theta}_0 = \arg\max_{\boldsymbol{\theta}_i} p_n(\boldsymbol{\theta}_i). \tag{3.12}$$

As new stimulus-response observations were obtained, we updated the particle filter weights recursively by

$$p_{n+1}(\boldsymbol{\theta}_i \mid \mathcal{D}_{n+1}) = \frac{1}{Z} p((r_{n+1}, \mathbf{x}_{n+1}) \mid \boldsymbol{\theta}_i) p_n(\boldsymbol{\theta}_i \mid \mathcal{D}_n), \tag{3.13}$$

where Z is a normalizing constant. Since a particle filter tends to degenerate to a small number of effective particles quantified by $N_{\mathrm{eff}} = 1/\sum_{i=1}^{N} p_n(\boldsymbol{\theta}_i)^2$, after every 20 stimulus presentations, we obtained a new set of particles for the filter by resampling the current continuous posterior density $p_n(\boldsymbol{\theta} \mid \mathcal{D}_n)$ by Markov chain Monte Carlo (MCMC) sampling using the random-walk Metropolis algorithm (Gilks, Richardson, & Spiegelhalter, 1995; Liu, 2001). Because of the periodic updating, the particle locations were not fixed in

the long run. Our target is the posterior in equation 3.3, and our proposal distribution for the kth step in the chain is a gaussian $\mathcal{N}(\boldsymbol{\theta} \mid \boldsymbol{\theta}_k, \rho\mathbf{I})$, where the mean $\boldsymbol{\theta}_k$ is the kth accepted particle and $\mathbf{I}$ is the identity matrix. The chain is initialized at $\boldsymbol{\theta}_0$ as in equation 3.12. To keep the acceptance rate for the chain near the ideal 20% (Gilks et al., 1995), we used $\rho = 0.1$, doubling ρ whenever the acceptance rate from the previous MCMC resampling of the continuous posterior exceeded 50% and halving it when it went below 1%. Intuitively, if our acceptance rate is too high (or low), then our proposal distribution is not broad (or narrow) enough.

To evaluate the utility function 3.10, we maintained a set of Fisher information matrices at each particle, and after each new stimulus $\mathbf{x}$, we updated the Fisher information matrix at each particle recursively using equation 3.6. After obtaining a new set of particles using the MCMC resampling procedure described above, we used all available stimuli $\mathbf{x}_1, \ldots, \mathbf{x}_n$ collected up to that point to compute the Fisher information matrix for each new particle. Since there are 10 parameters to estimate (see Table 1), at least 10 independent observations are needed to make our Fisher information matrix full rank, so at the start of the experiment, we added an identity matrix $\mathbf{I}$ to each Fisher information matrix until we had collected 20 data points (twice the number of parameters in Table 1) in order to avoid matrix degeneracy. To speed up computation, we evaluated equation 3.10 using a reduced posterior defined on the $N_{\text{eff}}/2$ particles with the largest posterior probability, using at least 10 and at most 50 particles.

The main goal of the example in Figure 1 was to demonstrate the usefulness of optimal design as a method for overcoming the problem of continuous parameter confounding, which can make accurate parameter estimation difficult in hierarchical neural network models (DiMattina & Zhang, 2010). Although the numerical methods used in this example were adequate for this network model, they may not generalize well to problems in higher dimensions. In particular, the particle filter methods used for the 10-dimensional problem in Figure 1 can potentially break down in higher dimensions, as many studies have shown that the number of particles needed to accurately represent the posterior density grows exponentially with the parameter space dimension (Snyder, Bengtsson, Bickel, & Anderson, 2008; Bengtsson, Bickel, & Li, 2008; Bickel, Li, & Bengtsson, 2008). MCMC was a simple resampling technique that was easy to implement for our problem in Figure 1, and it allowed us to track the posterior with a small number of particles so the optimization of equation 3.10 was reasonably fast (2–3 seconds per stimulus on a 2.4 GHz quad core PC). However, since the evaluations of the posterior density required by MCMC depend on all previous observations $\mathcal{D}_n$ up to that point, the MCMC resampling procedure gets slower in direct proportion to the number of stimuli that have been generated so far. Thus, for implementation of online experiments, the particle filter methods with MCMC are limited by the dimension of the parameter space and the number of stimuli to be generated. For models with many

parameters, such as the network in Figure 2a, we suggest using a gaussian-sum extended Kalman filter (EKF) method combined with optimization by heuristic search, as explained in the next section. This method worked faster (about 1 second per stimulus) in higher dimensions (about 300 in our examples), even though it may also become unreliable when the linearization in EKF is inaccurate or the prior is poorly chosen (Haykin, 2001).

3.2 Three-Layer Subunit Model Implementation

3.2.1 Posterior Approximation by Gaussian Sum. We write the posterior density of model parameters given the data $\mathcal{D}_n = \{(\mathbf{x}_1, r_1), \ldots, (\mathbf{x}_n, r_n)\}$ as $p_n(\boldsymbol{\theta}) \equiv p_n(\boldsymbol{\theta} \mid \mathcal{D}_n)$, and approximate it by a sum of gaussians (Alspach & Sorenson, 1972; Hering & Simandl, 2007):

$$p_n(\boldsymbol{\theta}) = \sum_{j=1}^{K} \alpha_n^{(j)} p_n^{(j)}(\boldsymbol{\theta}), \tag{3.14}$$

where

$$p_n^{(j)}(\boldsymbol{\theta}) \equiv \mathcal{N}\left(\boldsymbol{\theta} \mid \boldsymbol{\mu}_n^{(j)}, \boldsymbol{\Sigma}_n^{(j)}\right), \tag{3.15}$$

and the mixing weights $\alpha_n^{(j)} \geq 0$ with $\sum_{j=1}^{K} \alpha_n^{(j)} = 1$ so that equation 3.14 integrates to unity. For the prior distribution $p_0(\boldsymbol{\theta})$, we assume that $\alpha_0^{(j)} = 1/K$, $\boldsymbol{\Sigma}_0^{(j)} = \kappa \mathbf{I}$ (with $\kappa = 1/4$), and $\boldsymbol{\mu}_0^{(j)}$ is drawn randomly from a gaussian $\mathcal{N}(\boldsymbol{\theta} \mid \boldsymbol{\nu}, \kappa \mathbf{I})$, where components of $\boldsymbol{\nu}$ were set to -2 for hidden unit biases and 0 otherwise.

Approximating the Poisson neural response by a gaussian whose variance is equal to the mean (Dean, 1981) allows us to employ the EKF formalism (Alspach & Sorenson, 1972; Haykin, 2001) to update the parameters of the gaussian sum, given each new observation. The gaussian approximation for response r is

$$p(r \mid \mathbf{x}, \boldsymbol{\theta}) = \mathcal{N}\left(r \mid f(\mathbf{x}, \boldsymbol{\theta}), \sigma^2\right), \tag{3.16}$$

where the tuning function $f(\mathbf{x}, \boldsymbol{\theta})$ is the mean response to stimulus $\mathbf{x}$, and the variance $\sigma^2 = f(\mathbf{x}, \boldsymbol{\theta})$.

Given the posterior density $p_n(\boldsymbol{\theta})$ in equation 3.14, near the jth gaussian peak $\boldsymbol{\mu}_n^{(j)}$, we can linearly approximate the tuning function by

$$f(\mathbf{x}, \boldsymbol{\theta}) \approx b_n^{(j)} + \mathbf{a}_n^{(j)\mathrm{T}} \boldsymbol{\theta}, \tag{3.17}$$

where

$$\mathbf{a}_n^{(j)} = \nabla_{\boldsymbol{\theta}} f(\mathbf{x}, \boldsymbol{\mu}_n^{(j)}), \qquad b_n^{(j)} = f(\mathbf{x}, \boldsymbol{\mu}_n^{(j)}) - \mathbf{a}_n^{(j)\mathrm{T}} \boldsymbol{\mu}_n^{(j)}. \tag{3.18}$$

By ignoring all other gaussian bumps except for the jth one, the probability in equation 3.16 becomes approximately

$$p_n^{(j)}(r \mid \mathbf{x}, \boldsymbol{\theta}) = \mathcal{N}(r \mid b_n^{(j)} + \mathbf{a}_n^{(j)\mathrm{T}} \boldsymbol{\theta}, \sigma^2). \tag{3.19}$$

Assuming that the gaussian bumps in equation 3.14 have little overlap, we can approximate equation 3.16 as

$$p_n(r \mid \mathbf{x}, \boldsymbol{\theta}) = \sum_{j=1}^{K} \alpha_n^{(j)} p_n^{(j)}(r \mid \mathbf{x}, \boldsymbol{\theta}), \tag{3.20}$$

where subscript n indicates that the posterior $p_n(\boldsymbol{\theta})$ with n data points has been given.

Using Bayes' rule, as new data $(\mathbf{x}, r) \equiv (\mathbf{x}_{n+1}, r_{n+1})$ are observed, we update the means and variances of each of the gaussian bumps in the approximating sum, equation 3.14, separately and then update the relative weights $\alpha_n^{(j)}$ of each bump. Using Bayes' rule,

$$p_{n+1}^{(j)}(\boldsymbol{\theta} \mid \mathbf{x}, r) = \frac{p_n^{(j)}(r \mid \mathbf{x}, \boldsymbol{\theta}) p_n^{(j)}(\boldsymbol{\theta})}{p_n^{(j)}(r \mid \mathbf{x})}, \tag{3.21}$$

and the fact that the product of two gaussians is a gaussian, we have

$$p_{n+1}^{(j)}(\boldsymbol{\theta} \mid \mathbf{x}, r) = \mathcal{N}\left(\boldsymbol{\theta} \mid \boldsymbol{\mu}_{n+1}^{(j)}, \boldsymbol{\Sigma}_{n+1}^{(j)}\right) \tag{3.22}$$

with

$$\boldsymbol{\mu}_{n+1}^{(j)} = \boldsymbol{\mu}_n^{(j)} + \frac{1}{\sigma^2} \boldsymbol{\Sigma}_{n+1}^{(j)} \mathbf{a}_n^{(j)} \left(r - f(\mathbf{x}, \boldsymbol{\mu}_n^{(j)})\right), \tag{3.23}$$

$$\boldsymbol{\Sigma}_{n+1}^{(j)} = \left(\boldsymbol{\Sigma}_n^{(j)^{-1}} + \frac{1}{\sigma^2} \mathbf{a}_n^{(j)} \mathbf{a}_n^{(j)\mathrm{T}}\right)^{-1} = \boldsymbol{\Sigma}_n^{(j)} - \frac{\boldsymbol{\Sigma}_n^{(j)} \mathbf{a}_n^{(j)} \mathbf{a}_n^{(j)\mathrm{T}} \boldsymbol{\Sigma}_n^{(j)}}{\sigma^2 + \mathbf{a}_n^{(j)\mathrm{T}} \boldsymbol{\Sigma}_n^{(j)} \mathbf{a}_n^{(j)}}, \tag{3.24}$$

together with a succinct form for $p_n^{(j)}(r \mid \mathbf{x})$, which is useful for many of our calculations (Bishop, 2006):

$$p_n^{(j)}(r \mid \mathbf{x}) = \mathcal{N}\left(r \mid f(\mathbf{x}, \boldsymbol{\mu}_n^{(j)}), \sigma^2 + \mathbf{a}_n^{(j)\mathrm{T}} \boldsymbol{\Sigma}_n^{(j)} \mathbf{a}_n^{(j)}\right). \tag{3.25}$$

In the last step of equation 3.24, we applied the Woodbury matrix lemma (Harville, 1997) in order to avoid matrix inversion. After updating the means and variances of each gaussian bump, we update the mixing weights by

$$\alpha_{n+1}^{(j)} = \frac{\alpha_n^{(j)} p(r \mid \mathbf{x}, \boldsymbol{\mu}_{n+1}^{(j)})}{\sum_{j=1}^{K} \alpha_n^{(j)} p(r \mid \mathbf{x}, \boldsymbol{\mu}_{n+1}^{(j)})}. \tag{3.26}$$

These results are similar to those of the Kalman filter (Kalman, 1960) and its extensions (Alspach & Sorenson, 1972; Brown, Frank, Tang, Quirk, & Wilson, 1998; Haykin, 2001; Hering & Simandl, 2007).

In the time-varying example shown in Figures 2f and 2g, we assumed 10 gaussian bumps. In the example, which estimates additive models having $m = 1, 2, 3$ hidden subunits, we used $K = 3, 10, 10$ gaussians, respectively, in our approximation in equation 3.14.

One known weakness of the EKF is that it can fail to accurately track the posterior density due to accumulation of error (Haykin, 2001). This problem could in principle be rectified by periodic resampling of the posterior density in order to define a more accurate approximation of the posterior, as we did for the particle filter approximation used in Figure 1. In our application of the EKF, we did not use a resampling step but simply applied the standard recursive updating procedure outlined above, which we found to be adequate for our examples. This is probably because unlike a particle filter model, even when there is only a single gaussian bump, it may still be a reasonably accurate representation of the true posterior (see the examples in section 3.3).

3.2.2 Derivation of Utility Function with Gaussian Sum Approximation. For the estimation phase of the experiments illustrated in Figures 2 and 6, we design the stimulus by minimizing an information-theoretic estimation utility function (Paninski, 2005), which is the expected value of the posterior entropy (see equation 3.32). Here we compute this utility function and reduce it to a simpler form (see equation 3.38) using the gaussian sum approximation considered above.

Let the posterior be given by a gaussian sum as in equation 3.14. First, we compute its entropy:

$$H[p_n(\boldsymbol{\theta})] = -\int p_n(\boldsymbol{\theta}) \ln p_n(\boldsymbol{\theta}) d\boldsymbol{\theta} \tag{3.27}$$

$$= -\sum_{j=1}^{K} \alpha_n^{(j)} \int p_n^{(j)}(\boldsymbol{\theta}) \ln \left(\sum_{k=1}^{K} \alpha_n^{(k)} p_n^{(k)}(\boldsymbol{\theta}) \right) d\boldsymbol{\theta} \tag{3.28}$$

$$\leq -\sum_{j=1}^{K} \alpha_n^{(j)} \int p_n^{(j)}(\boldsymbol{\theta}) \ln \left(\alpha_n^{(j)} p_n^{(j)}(\boldsymbol{\theta}) \right) d\boldsymbol{\theta} \tag{3.29}$$

$$= \sum_{j=1}^{K} \alpha_n^{(j)} H \left[p_n^{(j)}(\boldsymbol{\theta}) \right] - \sum_{j=1}^{K} \alpha_n^{(j)} \ln \alpha_n^{(j)} \tag{3.30}$$

$$= \frac{1}{2} \sum_{j=1}^{K} \alpha_n^{(j)} \ln |\boldsymbol{\Sigma}_n^{(j)}| - \sum_{j=1}^{K} \alpha_n^{(j)} \ln \alpha_n^{(j)}. \tag{3.31}$$

The inequality in step 3.29 holds because the logarithm is a monotonically increasing function and each $\alpha_n^{(k)} p_n^{(k)}(\boldsymbol{\theta}) \geq 0$. The inequality 3.29 becomes approximately an equality when there is little overlap between different gaussian bumps $p_n^{(j)}$ and $p_n^{(k)}$ so that all cross-terms with $j \neq k$ vanish. (See section 3.3 for more discussion of this approximation.) The last step, equation 3.31, follows from the entropy $H \left[p_n^{(j)}(\boldsymbol{\theta}) \right] = \frac{1}{2} \ln |\boldsymbol{\Sigma}_n^{(j)}| + C$ for a gaussian (see equation 3.15), and the constant C does not affect our optimization and is ignored in equation 3.31. In our simulations, instead of minimizing the true entropy, we minimize the upper bound, equation 3.31.

Expression 3.30 has an information-theoretic interpretation: The second term is the entropy of the random variable j, which measures our uncertainty about which gaussian bump is correct, and the first term is a weighted sum of the conditional entropy of parameter $\boldsymbol{\theta}$ for each gaussian bump j, which measures the uncertainty of each gaussian bump. In optimal design, it is desirable to minimize both terms.

We are now ready to define our utility function as the expected value of the entropy of the subsequent posterior $p_{n+1}(\boldsymbol{\theta} \mid \mathbf{x}, r) \equiv p_{n+1}(\boldsymbol{\theta} \mid \mathcal{D}_n, \mathbf{x}, r)$:

$$U_{n+1}^{(E)}(\mathbf{x}) = -\int H \left[p_{n+1}(\boldsymbol{\theta} \mid \mathbf{x}, r) \right] p(r \mid \mathbf{x}) \, dr \tag{3.32}$$

$$\leq -\int \left(\frac{1}{2} \sum_{j=1}^{K} \alpha_{n+1}^{(j)} \ln |\boldsymbol{\Sigma}_{n+1}^{(j)}| - \sum_{j=1}^{K} \alpha_{n+1}^{(j)} \ln \alpha_{n+1}^{(j)} \right) p(r \mid \mathbf{x}) \, dr \tag{3.33}$$

$$\approx -\frac{1}{2} \int \left(\ln |\boldsymbol{\Sigma}_{n+1}^{(k)}| \right) p(r \mid \mathbf{x}) \, dr \,, \tag{3.34}$$

where step 3.33 follows from the bound 3.31 and the approximation in step 3.34 follows from the empirical observation that, with the gaussian sum approximation, at most times nearly all of the probability mass is focused on a single gaussian bump, say, with index k such that $\alpha_n^{(k)} \approx 1$ while $\alpha_n^{(j)} \approx 0$ for all other $j \neq k$. As the experiment progresses, the preferred

bump may change, and there are brief periods where the mass is somewhat evenly distributed between two or more bumps, but for the majority of trials, this approximation is valid. (See section 3.3 for an explicit test of this approximation.)

Using equation 3.24 and the matrix identity equation 3.7, we obtain

$$\ln\left|\mathbf{\Sigma}_{n+1}^{(k)}\right| = \ln\left|\mathbf{\Sigma}_{n}^{(k)}\right| - \ln\left(1+\frac{1}{\sigma^2}\mathbf{a}_n^{(k)\mathrm{T}}\mathbf{\Sigma}_n^{(k)}\mathbf{a}_n^{(k)}\right). \tag{3.35}$$

Since the first term on the right-hand side does not depend on $\mathbf{x}$ or r, maximizing equation 3.34 is equivalent to maximizing a new utility function:

$$U_{n+1}^{(\mathrm{E})}(\mathbf{x}) = \int \ln\left(1+\frac{1}{\sigma^2}\mathbf{a}_n^{(k)\mathrm{T}}\mathbf{\Sigma}_n^{(k)}\mathbf{a}_n^{(k)}\right) p\,(r \mid \mathbf{x})\,dr \tag{3.36}$$

$$= \ln\left(1+\frac{1}{\sigma^2}\mathbf{a}_n^{(k)\mathrm{T}}\mathbf{\Sigma}_n^{(k)}\mathbf{a}_n^{(k)}\right). \tag{3.37}$$

The last step obtains because for gaussian distribution, the Fisher information is independent of r (Kay, 1993), and therefore all the variables in the logarithm do not depend on r, including $\mathbf{a}_n^{(k)}$ by equation 3.18, $\mathbf{\Sigma}_n^{(k)}$ by equation 3.24 (recursive relation), and $\sigma^2 = f(\mathbf{x}, \boldsymbol{\mu}_n^{(k)})$. Thus maximizing equation 3.37 is equivalent to maximizing the final utility function,

$$U_{n+1}^{(\mathrm{E})}(\mathbf{x}) = \frac{1}{\sigma^2}\mathbf{a}_n^{(k)\mathrm{T}}\mathbf{\Sigma}_n^{(k)}\mathbf{a}_n^{(k)}, \tag{3.38}$$

which was used for the examples in Figures 2 and 3 with the simplifying assumption of constant σ^2.

In our derivation of the final utility function, equation 3.38, we have made use of several crude approximations for the sake of computational tractability. In our numerical simulations, we treat the response variance σ^2 as a constant, while in general, it depends on the stimulus $\mathbf{x}$. We assume minimal overlap between different gaussian bumps in order to obtain an upper bound, equation 3.29, for the entropy, and we minimize this bound rather than the entropy itself. We further assume that the majority of the weight lies on a single gaussian, which was approximately true for our examples but might break down for other examples. We return to these issues with concrete examples in section 3.3.

3.2.3 Stimulus Generation. Direct numerical optimization of the utility function equation 3.38 in the $2 \times 12^2 = 288$ dimensional stimulus space for the examples in this study (see Figures 2 and 6) is computationally intensive and may not be practical during real-time online experiments. Therefore,

we follow Lewi et al. (2009) in developing a heuristic search over a restricted set of stimuli.

Previous work shows that in order to accurately estimate the parameters of a neural network, it is desirable to present stimuli that drive the hidden units of the network over the full range of their gain function in order to prevent continuous parameter confounding (DiMattina & Zhang, 2010). This fact is visible from the example in Figure 1 (see panel b). Therefore, the main motivation for our heuristic will be to generate a finite set of stimuli that we expect to drive each of the hidden unit gain functions over their full range, given our current estimates of the weights of each hidden unit.

In our example, the neural networks have multiple hidden subunits, and we wish to drive each one of these subunits over their full range in all possible combinations of high and low activities for different subunits, given our current parameter estimates $\boldsymbol{\mu}_n^{(k)}$ as the most likely (kth) gaussian peak. Suppose we have m hidden subunits. From $\boldsymbol{\mu}_n^{(k)}$ we extract the current estimates of the weight vectors $\mathbf{w}_1, \ldots, \mathbf{w}_m$ for each of the subunits to form the matrix

$$\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_m, \mathbf{w}], \tag{3.39}$$

where $\mathbf{w}$ is a random column vector, which is picked differently for each of the stimuli we generate. To obtain a new set of orthonormal vectors, we apply Gram-Schmidt to equation 3.39 and obtain $Q = [\mathbf{q}_1, \ldots, \mathbf{q}_m, \mathbf{q}_{m+1}]$, whose columns span the same space as those of $\mathbf{W}$ but enjoy orthonormality. Each stimulus we generate is a linear combination given by

$$\mathbf{x} = c_1\mathbf{q}_1 + \cdots + c_m\mathbf{q}_m + c_{m+1}\mathbf{q}_{m+1}, \tag{3.40}$$

where coefficients c_i are chosen randomly as follows. For a network with m hidden units, we take N_m evenly spaced values $\omega_1, \ldots, \omega_{N_m}$ from the interval $[-E, E]$ and put $c_i = \omega_{l_i}/m$ for $i = 1, \ldots, m$, where ω_{l_i} is picked randomly from the list $\omega_1, \ldots, \omega_{N_m}$. We set the last coefficient as $c_{m+1} = \sqrt{E^2 - \sum_{i=1}^{m} c_i^2}$ so that the stimulus $\mathbf{x}$ is normalized to have Euclidean norm E.

Clearly this heuristic will generate stimuli that are various linear combinations of the columns of $\mathbf{Q}$ and all have power E^2. The vectors $\mathbf{q}_1, \ldots, \mathbf{q}_m$ span the same space as the weight vectors $\mathbf{w}_1, \ldots, \mathbf{w}_m$, while vector $\mathbf{q}_{m+1}$ also contains information about the random vector $\mathbf{w}$. If our current parameter estimate $\boldsymbol{\mu}_n^{(k)}$ is reasonably close to the true parameters, this stimulus set should drive the hidden units of the network over a wide range of their gain functions, and in all possible combinations of high and low activity across the population of hidden units. We optimize the utility function over this finite stimulus set. As shown by a special example in the next section, this heuristic method can sometimes yield parameter estimation nearly as accurate as that obtained by full optimization.

In the example in Figure 2, the true model has $m = 2$ hidden units. We used $N_2 = 10$ evenly spaced values from the interval $[-E, E]$. The total number of stimuli generated was $N_2^2 = 100$. Since our particular example assumes an amplitude constraint rather than a power constraint, as a final processing step we set all positive components of each of our stimuli to $x_{\max} = 1$ and all negative components to 0 so that the stimulus amplitude lies in the range $[0, x_{\max}]$. Generating each stimulus took about 1 second on a Dell Inspiron desktop (2.4 GHz quad core).

For model comparison (see section 4), we also considered models with $m = 1$ hidden unit or $m = 3$ hidden units. In these two cases, the numbers of evenly spaced values were $N_1 = 50$ and $N_3 = 5$, while the total numbers of stimuli were $N_1^1 = 50$ and $N_3^3 = 125$. For the time-varying example in Figures 2f and 2g, we used $N_2^2 = 100$ stimuli.

3.3 Testing Numerical Approximations Using Toy Problems. The examples presented in the preceding sections used several simplifying assumptions and numerical approximations that were not tested explicitly. In this section, we will do the tests on the low-dimensional toy models shown in Figures 4a and 5a, which are easier to analyze, at least numerically.

3.3.1 Testing Gaussian Sum Approximations and Utility Function Optimization. Figure 4a illustrates the simplest possible three-layer network, having a single unit in the input, hidden, and output layers. We will use this model to examine the optimization of the utility function 3.38 and assess the validity of both our gaussian sum approximation 2.11 and our gaussian sum entropy approximation 3.29. This model has a one-dimenisional input space $x \in [-5, 5]$, permitting us to easily visualize and globally optimize the utility function. The response of this model has Poisson distribution, with its mean given by

$$f(x, \boldsymbol{\theta}) = K v g(wx + w_0), \tag{3.41}$$

where $g(u) = 1/(1 + e^{-u})$ is the gain function, and $K = 50$ and $w_0 = -2$ are fixed so that our model parameters are $\boldsymbol{\theta} = (v, w)$, with the true values given by $\boldsymbol{\theta} = (1, 2)$. Given the mean value $\boldsymbol{\mu} = (v, w)^{\mathrm{T}}$ of the tallest peak in the current gaussian sum posterior, it is simple to compute

$$\nabla_{\boldsymbol{\theta}} f(x, \boldsymbol{\mu}) = K \begin{bmatrix} g(wx + w_0) \\ vxg'(wx + w_0) \end{bmatrix}. \tag{3.42}$$

Let the covariance matrix of this gaussian be given by $\boldsymbol{\Sigma} = \{\sigma_{ij}\}$ $(i, j = 1, 2)$, and we may write the expected utility of the next stimulus x (see equation 3.38) as

$$U^{(\mathrm{E})}(x) = \sigma_{11} g^2 + 2\sigma_{12} vxg'g + \sigma_{22} \left(vxg'\right)^2, \tag{3.43}$$

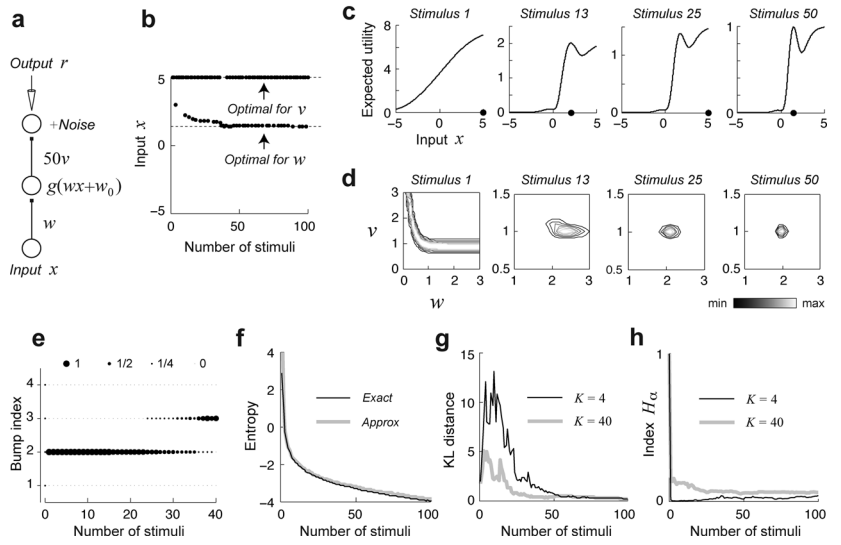


Figure 4: Testing numerical methods using a simple toy model with a single hidden unit. (a) A simple neural network model having a single hidden unit and a single input. (b) Input stimulus $x \in [-5, 5]$ is chosen to optimize the expected utility function in equation 3.38. As the experiment progresses, the utility is alternately optimized at one of two locations. (c) The shapes of the utility function 3.38 at several stages of the experiments. The locations that maximize the utility functions (black dots on the x-axis) correspond to the stimuli plotted in panel b. (d) True posterior distribution sampled by Markov chain Monte Carlo method at several stages of the experiments. (e) The weights on each of the $K = 4$ gaussian bumps in the gaussian sum approximations during early iterations of experiments are represented by the diameters of the dots. (f) Median entropy of 100 Monte Carlo trials computed exactly using equation 3.28 (black curve) or approximately using the upper-bound equation 3.29 (gray curve). The medians are very similar, and therefore the bound is quite tight. Curves of the 25th and 75th percentiles are too close to the medians and are not shown here to reduce clutter. (g) Median KL distance between the true posterior and the gaussian sum approximation over 50 Monte Carlo trials with $K = 4$ or 40 gaussians. (h) Median of the entropy index H_α (see equation 3.45) for the gaussian sum weights over 50 Monte Carlo trials. Even when the number of gaussians in the sum is increased by an order of magnitude, the entropy decreases rapidly toward 0, indicating that most weight is on a single bump.

where $g \equiv g(wx + w_0)$ and unity response noise variance is assumed in equation 3.38.

We can gain some intuition for how stimuli x may be chosen to optimize the utility by examining equation 3.43 in detail. Suppose that we

have nearly complete certainty of w so that $\sigma_{22} \approx 0$, and little covariance between w and v so that $\sigma_{12} \approx 0$. Then equation 3.43 is approximately $U^{(\mathrm{E})}(x) = \sigma_{11} g(wx + w_0)^2$, which is maximized for positive weight $w > 0$ and monotonically increasing g when input x assumes the largest permissible value ($x = 5$). Similarly, assuming that we have nearly complete certainty of v and negligible covariance ($\sigma_{11}, \sigma_{12} \approx 0$), we may approximate equation 3.43 by $U^{(\mathrm{E})}(x) = \sigma_{22}(vxg'(wx + w_0))^2$. We maximize $U^{(\mathrm{E})}(x)$ by maximizing $(xg'(wx + w_0))^2$, and find $x = 1.3797$, using the true parameters. (Another local maximum at $x = -0.5474$ has a smaller peak.)

From the considerations above, we might expect the utility function to be maximized at two distinct locations ($x = 5$ or $x = 1.3797$) as the experiment progresses and the parameter estimates converge to the truth. Figure 4b shows that this is exactly what happens in simulation (dashed lines indicating the two theoretical locations). Figure 4c shows the utility function (see equation 3.43) at four different iterations of the procedure, and the input that maximizes the utility function is indicated by the black dot.

Next, to test whether the bound in equation 3.29 is a good approximation for gaussian sum entropy, we evaluated this bound and the exact expression 3.28 by Monte Carlo integration and made a direct comparison between the two over 100 repeated trials. As shown in Figure 4f, the medians of the two methods are very similar, with the approximate entropy being slightly higher on average, which we would expect because our approximation is actually a bound. In this example, the bound is relatively tight and therefore may serve as a reasonable approximation. In general, we minimize the upper bound 3.31 of the entropy instead of the true entropy 3.27. The bound approaches the true entropy when there is little overlap among different gaussian bumps. Even when the bound is not tight, minimizing an upper bound may still be useful.

Finally, we directly verify the accuracy of our gaussian sum representation (see equation 2.11) of the true posterior density by computing at each iteration the Kullback-Leibler (KL) divergence between the two. The true posterior was sampled by the MCMC method described in section 3.1.2. As shown by Figure 4g, for 100 repeated experiments, the median KL divergence between the two distributions goes to 0 as the experiment proceeds, demonstrating the eventual convergence of our gaussian sum approximation (see equation 2.11) to the true posterior.

3.3.2 Testing the Heuristic Search Method. Here we will use the network model with two hidden units (see Figure 5a) to test our heuristic search method, as used for the example in Figure 2 and described in section 3.2.3, and compare it against direct optimization of the utility function 3.38. This model has Poisson-distributed responses with the mean given by

$$f(\mathbf{x}, \boldsymbol{\theta}) = h\left(v_1 g\left(\mathbf{w}_1^{\mathrm{T}}\mathbf{x} + w_{01}\right) + v_2 g\left(\mathbf{w}_2^{\mathrm{T}}\mathbf{x} + w_{02}\right)\right), \tag{3.44}$$

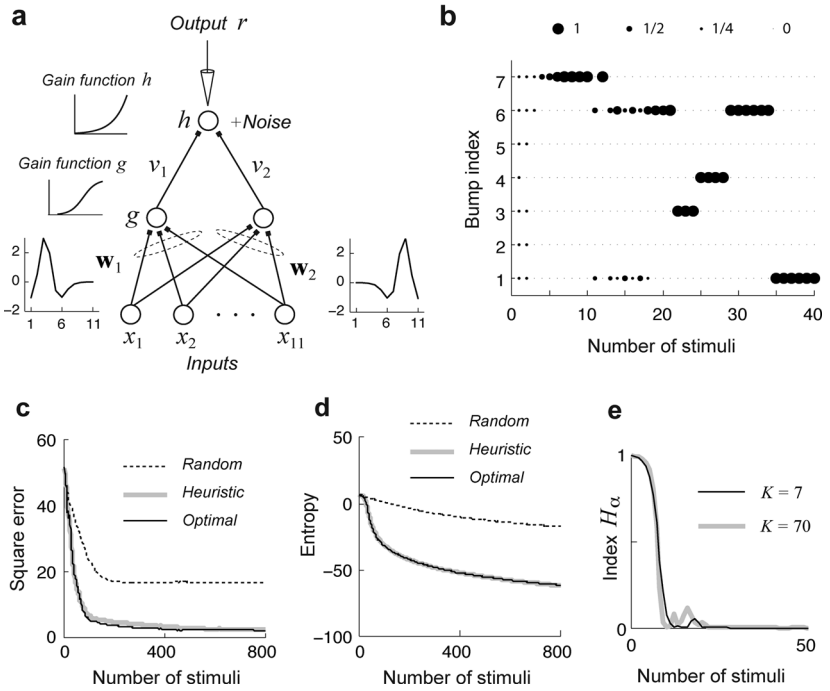


Figure 5: Testing numerical methods using another toy model with two hidden units. (a) Simple neural model having two hidden units. Top insets: Hidden unit and output unit gain functions. Bottom insets: True input weights. (b) Plots of the weights on each of the $K = 7$ gaussian bumps in the gaussian sum approximations during early iterations of experiments. (c) Median square errors over 100 Monte Carlo trials using either direct optimization over full stimulus space (black curve) or optimization over a finite set generated using our heuristic (gray curve) are about the same, and both are much better than that using random stimuli (dashed curves). Curves denoting the 25th and 75th percentiles are very close to the medians and not shown here. (d) Same as panel c but for entropy. (e) The entropy index H_α (see equation 3.45) vanishes quickly regardless of the number of gaussian bumps used ($K = 7$ or 70), indicating the eventual dominance by a single bump. The results shown are the medians from 50 Monte Carlo trials.

where $h(u) = e^u$, $g(u) = 1/(1 + e^{-u})$, and each input x_i in $\mathbf{x} = (x_1, \ldots, x_{11})$ lies in the range $[0, 1]$. The parameter set for this model is $\boldsymbol{\theta} = \{v_1, v_2, w_{01}, w_{02}, \mathbf{w}_1, \mathbf{w}_2\}$, with the true parameter values given by $v_1 = v_2 = 3$, $w_{01} = w_{02} = -1$, and $\mathbf{w}_1, \mathbf{w}_2$, as shown in the insets at the bottom of Figure 5a.

Direct optimization of the utility function 3.38 was implemented by performing an exhaustive search over the set of all $2^{11} = 2048$ possible stimuli, where each bin is either 1 or 0, and then using the best stimulus in this discrete set as the initial search point for numerical maximization in the full stimulus space by constraint optimization (Matlab fmincon). We also initiated the optimization procedure from random starting points in the full stimulus space, and the results always seemed to be close to binary stimulus patterns (bins with activities below 0.5 averaged to 0.0073, while those above 0.5 averaged to 0.9925, $N = 800$ stimuli). This observation justifies our exhaustive search method described above.

As shown in Figure 5c, over 100 repeated numerical experiments, the median square errors attained using our heuristic method were very similar to those attained by direct optimization, and both were substantially less than that attained using random stimuli. Figure 5d shows similar results with entropy measure. As an additional control, we optimized our entropy criterion 3.38 over a finite set of random stimuli, attaining better results than presenting random stimuli but significantly worse results than optimizing over an equally large set of stimuli generated by our heuristic (not shown). The Wilcoxon rank-sum test failed to find any significant difference between the results (squared error or entropy) attained by direct optimization and our heuristic method at stimulus numbers 50, 100, 200, and 800. Therefore, in this example, optimizing over a finite set of stimuli generated by our heuristic method performs nearly as well as optimization in the full stimulus space.

3.3.3 Varying the Number of Gaussian Bumps. One free variable in the specification of the gaussian sum approximation (see equation 2.11) to the posterior is the number K of gaussians. In general, increasing K improves the quality of the approximation while making the EKF update step slower. In the examples studied in Figure 2, we simply set K to keep the procedure fast enough to use in real experiments (≈ 1 second per stimulus). In this section, we use the two toy models in Figures 4a and 5a to study the effects of increasing the number of gaussian bumps. We find that even when K is increased by an order of magnitude, the majority of the weight still lies on a single bump or a few bumps, possibly due to the asymptotically near-gaussian shape of the true posterior as the trial progresses.

We quantify the extent to which the weight is concentrated on a single gaussian with the index

$$H_\alpha = -\frac{1}{\ln K} \sum_{j=1}^{K} \alpha_n^{(j)} \ln \alpha_n^{(j)}, \tag{3.45}$$

where $\alpha_n^{(j)}$ is the weight on the jth gaussian bump for stimulus number n. Note that we always have $0 \leq H_\alpha \leq 1$. The minimum $H_\alpha = 0$ occurs when

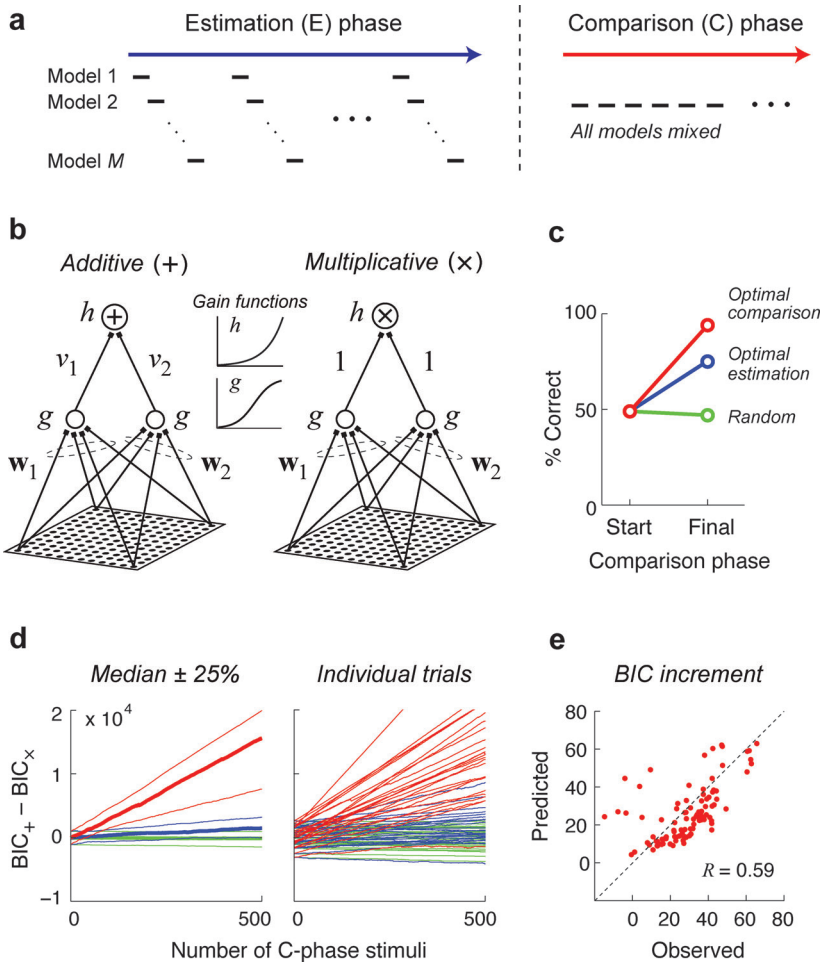
all of the weight in the gaussian sum lies on a single bump ($\alpha_n^{(k)} = 1$ for some k, $\alpha_n^{(j)} = 0$ for all $j \neq k$). The maximum $H_\alpha = 1$ occurs when the weight is evenly distributed among all of the bumps ($\alpha_n^{(j)} = 1/K$ for all j).

For the two-dimensional example shown in Figure 4a, we can directly visualize the posterior density and find that in most cases, the posterior resembles a unimodal gaussian density after a small number of stimuli (see the examples in Figure 4d). We have also visualized the weight distribution in the gaussian sum and found that at any given time, the majority of weight is often located on a single bump or split between only a few bumps, with the preferred bump changing during the early phases of the experiment, but ultimately settling on a single gaussian bump (see Figures 4e and 5b). Consistent with the intuitive observations above, we find that for both toy models, the entropy index H_α decreases quickly to nearly 0, even when the number K of gaussian bumps varies by an order of magnitude (see Figures 4h and 5e). Taken together, these results suggest a degenerate gaussian sum approximation with a single dominating bump and provide a justification for our single-bump approximation used in section 3.2.

4 Estimating and Comparing Multiple Neural Networks

4.1 A Two-Stage Procedure for Online Experiments. Neuroscientists often entertain multiple possibilities when considering the appropriate model to describe sensory neurons. Therefore, it is of interest to develop methods that use active data collection for the dual goals of estimating the parameters of multiple models and discriminating between these models. In general the best stimuli for model estimation may not be the most useful for model discrimination (Nelson, 2005), and so very few studies to date have addressed the issue of how to choose stimuli adaptively in order to combine these two distinct goals (Sugiyama & Rubens, 2008). One recent study (Cavagnaro et al., 2010) presents an information-theoretic approach to model discrimination based on choosing stimuli that most greatly decrease the entropy of a distribution defined on the space of possible models. However, in this method, estimating the relative probability of each model requires integration over the unknown parameters of that model, which can be computationally demanding for high-dimensional models like those considered here.

In light of these considerations, we propose a two-phase active data collection procedure illustrated schematically in Figure 6a for identifying the best model in a set of $M \geq 2$ candidate models. In the first or estimation, phase (E-phase), optimal design is used to generate stimuli that are most useful for estimating the parameters of the candidate models, with the best stimulus for each model being presented in turn. After the parameters of each model have been estimated, we start the second, or comparison,



phase (C-phase) during which stimuli are generated that are optimal for discriminating the competing models. Although this procedure is by no means the only possible way to combine estimation and comparison, breaking the experiment into sequential phases ensures that we attain a good estimate of the parameters of each model before we find stimuli to best discriminate among them. Several optimal design methods for model comparison have been considered in the literature (Atkinson & Donev, 1992), and in sections 4.2.1 and 4.2.2 we derive criteria based on likelihood and mutual information that are applicable to comparisons of multiple models. The remainder of this letter illustrates the application of this procedure to hypothetical neurophysiology experiments.

4.1.1 Estimation Phase. We want to simultaneously estimate the parameters of M candidate network models. For example, index $m = 1, 2, \ldots, M$ could be the number of hidden units in a neural network. For each model m, we obtain an optimal estimation stimulus by optimizing the expected utility function $U_{n+1}^{(\mathrm{E},m)}(\mathbf{x})$, which depends on the posterior density $p_n(\boldsymbol{\theta}^{(m)})$ as in equation 2.4. Other than the superscript m, the procedure is the same as before. We design the stimuli for optimally estimating each model in turn (see Figure 6a).

4.1.2 Comparison Phase. Once parameter estimates $\hat{\boldsymbol{\theta}}_1, \ldots, \hat{\boldsymbol{\theta}}_M$ are attained for the M candidate models by maximum likelihood, we compute the likelihood of each model using the Bayes information criterion (BIC), which measures how well each model fits the data while penalizing model complexity (Schwarz, 1978; MacKay, 1992). For a model m having K_m free parameters, we use

$$\mathrm{BIC}_m = \ln p\left(\mathcal{D}_n \mid \hat{\boldsymbol{\theta}}_m\right) - \frac{K_m}{2} \ln n, \tag{4.1}$$

where $\mathcal{D}_n = \{(\mathbf{x}_1, r_1), \ldots, (\mathbf{x}_n, r_n)\}$ denotes the set of data used to estimate the model. We select the model m with the highest BIC_m.

Figure 6: A two-phase procedure for estimating and comparing competing nonlinear models using active data collection. (a) To estimate and compare competing models (e.g., neural networks of varying complexity), first in the estimation phase, we design stimuli that are optimal for estimating each of the candidate models in turn, and next in the comparison phase, we generate stimuli that are optimal for distinguishing these models. (b) The additive network model is similar to that in Figure 2a, whereas the multiplicative model assumes that the subunit activities are combined multiplicatively instead of additively. The additive model (+) is assumed to be the true model. (c) Percentage of correct choices of the two models in panel b before and after the comparison phase. With optimal comparison stimuli presented, we almost always picked the correct model (red), whereas we were essentially at chance at the start of the comparison phase and did not improve with random stimuli (green) and improved only somewhat by continuing with optimal estimation stimuli (blue). In each condition, 100 Monte Carlo experiments were performed. (d) The same data as in panel c, showing that the differential evidence ($\mathrm{BIC}_+ - \mathrm{BIC}_\times$) in favor of the true model increases the fastest with the optimal comparison stimuli (red), only moderately with the optimal estimation stimuli (blue), and hardly at all with the random stimuli (green). Left: Median differential evidence is shown by the thick lines. Thin lines denote 25th and 75th percentiles. Right: Differential evidence for a random sample of 25 individual trials. (e) BIC increment was observed as the slope of ($\mathrm{BIC}_+ - \mathrm{BIC}_\times$), as exemplified by the red lines in panel d, for 100 individual trials. The results were consistent with the predictions by equation 4.31.

Table 4: Some Utility Functions for Model Comparison.

Design	Utility $u^{(C)}(\mathbf{x} \mid m)$	Interpretation
Likelihood	$D_{KL}\left[p(r \mid \mathbf{x}, m),\, p(r \mid \mathbf{x},\, j \neq m)\right]$	Relative likelihood of true model
Mutual information	$D_{KL}\left[p(r \mid \mathbf{x}, m),\, p(r \mid \mathbf{x})\right]$	Model space entropy

Since the data set $\mathcal{D}_n$ used to estimate the models may not be optimally suited for discriminating among competing models, we recommend collecting additional data optimized for model comparison during a second experimental phase (C-phase), during which we design stimuli by optimizing a comparison utility function $U^{(C)}(\mathbf{x})$. We refer to the stimulus $\mathbf{x}$ that maximizes $U^{(C)}(\mathbf{x})$ as the optimal comparison stimulus.

There are numerous ways to define optimal stimuli for comparing M competing models (see Table 4). When $M = 2$ with gaussian noise, the utility function based on expected change in likelihood in favor of the true model can be written as

$$U^{(C)}(\mathbf{x}) = \left(f_1(\mathbf{x}) - f_2(\mathbf{x})\right)^2, \tag{4.2}$$

where $f_m(\mathbf{x})$ is the mean response of model m to stimulus $\mathbf{x}$, using the current parameter estimate $\hat{\boldsymbol{\theta}}_m$. The stimulus that maximizes this function elicits the most different responses by the two models. In the case of Poisson noise, this utility becomes

$$U^{(C)}(\mathbf{x}) = \left(f_1(\mathbf{x}) - f_2(\mathbf{x})\right)\left(\ln f_1(\mathbf{x}) - \ln f_2(\mathbf{x})\right), \tag{4.3}$$

which is invariant when subscripts 1 and 2 are switched. (See section 4.2.1 for derivation.) Alternatively, one may also define a more general utility function for comparing any number of competing models ($M \geq 2$) using model space entropy:

$$U^{(C)}(\mathbf{x}) = \sum_{m=1}^{M} P_0(m) D_{KL}\left(p(r \mid \mathbf{x}, m),\, p(r \mid \mathbf{x})\right), \tag{4.4}$$

where $P_0(m)$ is the a priori probability for model m to be true and D_{KL} denotes the Kullback-Leibler divergence (see section 4.2.2).

In the hypothetical experiment in Figure 6b for testing two competing hypotheses about the subunit integration mechanism (additive versus multiplicative), the data were generated by the true additive model. Our candidate models include both the additive model (see Figure 6b, left) and

an incorrect multiplicative model (see Figure 6b, right), which is described by

$$f_{\times}(\mathbf{x}, \boldsymbol{\theta}) = h\left(\alpha \prod_{i=1}^{m} g(\mathbf{w}_i^{\mathrm{T}}\mathbf{x} + w_{0i})\right), \tag{4.5}$$

where α is an output scaling parameter, $m = 2$ is the number of hidden units, and the gains g and h are the same as the additive model in equation 2.7. The parameter set $\boldsymbol{\theta} = \{\alpha, w_{01}, w_{02}, \mathbf{w}_1, \mathbf{w}_2\}$ contains 291 parameters. We used $K = 10$ gaussians in our approximation of the posterior in equation 3.14.

After the two candidate models have been estimated using optimally designed stimuli, the accuracy of model selection is still close to chance (see Figure 6c, "Start"). In the comparison phase, using random noise stimuli does not rectify this situation (green); using stimuli optimized for model estimation (i.e., continuing the same procedure as in the estimation phase) only partially rectifies this (blue); but using stimuli optimized for model comparison (with the criterion in equation 4.2) leads to nearly perfect model selection (red). Consistent with this result, the differential evidence ($\mathrm{BIC}_{+} - \mathrm{BIC}_{\times}$) in favor of the true model increases sharply for the optimal comparison stimuli but not for the random stimuli and only slightly for the optimal estimation stimuli (see Figure 6d). Note that for many individual experiments with optimal comparison stimuli, the differential evidence changes from negative to positive (see Figure 6d, right), meaning that an incorrect initial preference for the multiplicative model is corrected to a final preference for the additive model.

We have also used our two-stage procedure for estimating and comparing models like the one in Figure 2a having $m = 1, 2$, or 3 hidden units. This analysis yielded similar results (not shown). See section 4.3 for further analysis on model comparison with the BIC.

4.2 Derivation of Model Comparison Utility Functions. This section contains the mathematical derivation of the model comparison utility functions used in the examples discussed in the preceding section.

4.2.1 Likelihood-Based Utility Function. In this section we derive the general likelihood-based utility function as given by equations 4.18 and 4.19. This function includes equations 4.2 and 4.3 as special cases.

Given two candidate models equally likely a priori and observations $\mathcal{D}_n = \{(\mathbf{x}_1, r_1), \ldots, (\mathbf{x}_n, r_n)\}$ generated by one of the two models, we may compare the models by taking the difference of the log likelihoods,

$$\lambda_{12} = \ln p(\mathcal{D}_n \mid 1) - \ln p(\mathcal{D}_n \mid 2), \tag{4.6}$$

where model 1 is preferred when $\lambda_{12} > 0$ and model 2 is preferred when $\lambda_{12} < 0$. In most experiments, the data set $\mathcal{D}_n$ is not collected with the goal

of maximizing $|\lambda_{12}|$. Here we consider how to collect additional data $(\mathbf{x}, r)$ that maximize the expected change in $|\lambda_{12}|$ in order to tip the scales more solidly in favor of one model or the other.

If model 1 is the true model, then the expected change $\Delta\lambda_{12}$ caused by a new stimulus $\mathbf{x}$ is given approximately by

$$\Delta\lambda_{12}(\mathbf{x}) = \int p(r \mid \mathbf{x}, \hat{\boldsymbol{\theta}}^{(1)}) \ln \frac{p(r \mid \mathbf{x}, \hat{\boldsymbol{\theta}}^{(1)})}{p(r \mid \mathbf{x}, \hat{\boldsymbol{\theta}}^{(2)})} dr, \tag{4.7}$$

where $\hat{\boldsymbol{\theta}}^{(1)}$ and $\hat{\boldsymbol{\theta}}^{(2)}$ are the final model parameter estimates attained from data $\mathcal{D}_n$ (Burnham & Anderson, 2002). Maximizing equation 4.7 with respect to $\mathbf{x}$ will tip the scales most strongly in favor of the true model (model 1). We readily recognize equation 4.7 as the Kullback-Leibler (KL) divergence (Kullback & Leibler, 1951; Cover & Thomas, 2006), which is denoted $D_{\mathrm{KL}}[p(r \mid \mathbf{x}, \hat{\boldsymbol{\theta}}^{(1)}), p(r \mid \mathbf{x}, \hat{\boldsymbol{\theta}}^{(2)})]$. A symmetric argument shows that if model 2 is true, the expected change in likelihood in favor of model 2 is given by $D_{\mathrm{KL}}[p(r \mid \mathbf{x}, \hat{\boldsymbol{\theta}}^{(2)}), p(r \mid \mathbf{x}, \hat{\boldsymbol{\theta}}^{(1)})]$. Since we do not know which model is true, we arrive at the expected utility function

$$U^{(\mathrm{C})}(\mathbf{x}) = \frac{1}{2} u^{(\mathrm{C})}(\mathbf{x} \mid 1) + \frac{1}{2} u^{(\mathrm{C})}(\mathbf{x} \mid 2), \tag{4.8}$$

where the conditional expected utility functions are given by

$$u^{(\mathrm{C})}(\mathbf{x} \mid 1) = D_{\mathrm{KL}}\left[p(r \mid \mathbf{x}, \hat{\boldsymbol{\theta}}^{(1)}), p(r \mid \mathbf{x}, \hat{\boldsymbol{\theta}}^{(2)})\right], \tag{4.9}$$

$$u^{(\mathrm{C})}(\mathbf{x} \mid 2) = D_{\mathrm{KL}}\left[p(r \mid \mathbf{x}, \hat{\boldsymbol{\theta}}^{(2)}), p(r \mid \mathbf{x}, \hat{\boldsymbol{\theta}}^{(1)})\right]. \tag{4.10}$$

We readily observe that equations 4.8 to 4.10 define a symmetrized KL divergence. The optimal comparison stimulus is obtained by maximizing $U^{(\mathrm{C})}(\mathbf{x})$.

In the special case where both models have gaussian noise with fixed variance σ^2, equation 4.8 reduces to the simple and intuitive form

$$U^{(\mathrm{C})}(\mathbf{x}) = \frac{1}{2\sigma^2} \left(f_1(\mathbf{x}) - f_2(\mathbf{x})\right)^2, \tag{4.11}$$

where $f_1(\mathbf{x}) \equiv f^{(1)}(\mathbf{x}, \hat{\boldsymbol{\theta}}^{(1)})$ and $f_2(\mathbf{x}) \equiv f^{(2)}(\mathbf{x}, \hat{\boldsymbol{\theta}}^{(2)})$, with $f^{(m)}(\mathbf{x}, \boldsymbol{\theta}^{(m)})$ being the average response of model m to stimulus $\mathbf{x}$. Equation 4.11 is the same as equation 4.2 when the constant factor $1/2\sigma^2$ is ignored. Thus, the best stimulus for discriminating two models is the stimulus for which their predictions of the model response differ the most, and a sum-of-squares

criterion similar to this one has been used in classic work to define designs that are T-optimal (Atkinson & Donev, 1992; Atkinson & Fedorov, 1975a, 1975b).

When both models have Poisson noise, we have $p(r \mid \mathbf{x}, \hat{\boldsymbol{\theta}}^{(i)}) = f_i(\mathbf{x})^r \exp(-f_i(\mathbf{x}))/r!$ with $i = 1, 2$. Now equation 4.9 becomes

$$u^{(\mathrm{C})}(\mathbf{x} \mid 1) = \sum_{r=0}^{\infty} p(r \mid \mathbf{x}, \hat{\boldsymbol{\theta}}^{(1)}) \ln \frac{p(r \mid \mathbf{x}, \hat{\boldsymbol{\theta}}^{(1)})}{p(r \mid \mathbf{x}, \hat{\boldsymbol{\theta}}^{(2)})} \tag{4.12}$$

$$= \sum_{r=0}^{\infty} p(r \mid \mathbf{x}, \hat{\boldsymbol{\theta}}^{(1)}) \left(r \ln \frac{f_1(\mathbf{x})}{f_2(\mathbf{x})} + f_2(\mathbf{x}) - f_1(\mathbf{x}) \right) \tag{4.13}$$

$$= f_1(\mathbf{x}) \ln \frac{f_1(\mathbf{x})}{f_2(\mathbf{x})} + f_2(\mathbf{x}) - f_1(\mathbf{x}), \tag{4.14}$$

where the last step follows from $\sum_{r=0}^{\infty} p(r \mid \mathbf{x}, \hat{\boldsymbol{\theta}}^{(1)}) r = f_1(\mathbf{x})$. A similar formula for $u^{(\mathrm{C})}(\mathbf{x} \mid 2)$ in equation 4.10 can be obtained with switched subscripts 1 and 2. Thus, equation 4.8 becomes

$$\begin{aligned} U^{(\mathrm{C})}(\mathbf{x}) &= \frac{1}{2} \left(f_1(\mathbf{x}) \ln \frac{f_1(\mathbf{x})}{f_2(\mathbf{x})} + f_2(\mathbf{x}) \ln \frac{f_2(\mathbf{x})}{f_1(\mathbf{x})} \right) \\ &= \frac{1}{2} \left(f_1(\mathbf{x}) - f_2(\mathbf{x}) \right) \ln \frac{f_1(\mathbf{x})}{f_2(\mathbf{x})}. \end{aligned} \tag{4.15}$$

This is the same as equation 4.3 when the factor 1/2 is ignored.

We now extend our results to arbitrarily many models, with each model m having a prior probability $P_0(m)$ to be the true model. The main idea is that if model $m \in \{1, \ldots, M\}$ is true, we want to find the stimulus $\mathbf{x}$ that maximizes the expected increase in the likelihood of model m while minimizing that for all other models $j \neq m$. We choose $\mathbf{x}$ to maximize the expression

$$\Delta\lambda_m = \langle \ln p(r \mid \mathbf{x}, m) - \ln p(r \mid \mathbf{x}, j \neq m) \rangle_{r|m}, \tag{4.16}$$

where $p(r \mid \mathbf{x}, m) \equiv p(r \mid \mathbf{x}, \hat{\boldsymbol{\theta}}^{(m)})$ and the notation $\langle \cdot \rangle_{r|m}$ denotes the expectation with respect to r assuming that model m is true, and we define

$$p(r \mid \mathbf{x}, j \neq m) = \frac{1}{Z} \sum_{j \neq m} P_0(j) p(r \mid \mathbf{x}, j), \tag{4.17}$$

with $Z = \sum_{j \neq m} P_0(j)$. Following an argument like that given above for two models, we can write the utility function for model comparison as

$$U^{(\mathrm{C})}(\mathbf{x}) = \sum_{m=1}^{M} u^{(\mathrm{C})}(\mathbf{x} \mid m) P_0(m), \tag{4.18}$$

where

$$u^{(\mathrm{C})}(\mathbf{x} \mid m) = D_{\mathrm{KL}}\left[p(r \mid \mathbf{x}, m),\, p(r \mid \mathbf{x},\, j \neq m)\right]. \tag{4.19}$$

This finishes the derivation of the final utility function. For two models ($M = 2$) with equally likely prior, equations 4.18 and 4.19 reduce to equations 4.8 to 4.10.

4.2.2 Information-Theoretic Utility Function. Now we consider the utility function for model comparison based on model space entropy, as given by equation 4.4. For a set of M candidate models, let the probability for model $m = 1, \ldots, M$ to be true be given by $P_0(m)$. In a recent study, Cavagnaro et al. (2010) proposed that one may reduce the uncertainty about the model by choosing a stimulus $\mathbf{x}$ that maximizes the mutual information between the stimulus and the unknown model m. This criterion may be written as

$$U^{(\mathrm{C})}(\mathbf{x}) = \sum_{m=1}^{M} P_0(m) \int\int p(r \mid \mathbf{x}, \boldsymbol{\theta}) p_0(\boldsymbol{\theta}) \ln \frac{p(m \mid \mathbf{x}, r)}{P_0(m)} dr d\boldsymbol{\theta} \tag{4.20}$$

$$= \sum_{m=1}^{M} P_0(m) \int p(r \mid \mathbf{x}, m) \ln \frac{p(m \mid \mathbf{x}, r)}{P_0(m)} dr\,, \tag{4.21}$$

where the second step obtains because the quantity in the argument of the logarithm does not depend on $\boldsymbol{\theta}$. Plugging the Bayes' rule $p(m \mid \mathbf{x}, r) = p(r \mid \mathbf{x}, m) P_0(m) / p(r \mid \mathbf{x})$ into equation 4.21 yields the final utility function as in equation 4.4:

$$U^{(\mathrm{C})}(\mathbf{x}) = \sum_{m=1}^{M} P_0(m) u^{(\mathrm{C})}(\mathbf{x} \mid m), \tag{4.22}$$

with

$$u^{(\mathbf{C})}(\mathbf{x} \mid m) = D_{\mathrm{KL}}(p(r \mid \mathbf{x}, m),\, p(r \mid \mathbf{x})). \tag{4.23}$$

The mutual information model comparison criterion in Table 4 (see equation 4.23) is analogous to the mutual information estimation criterion (see Table 2), with the unknown model m taking the place of the unknown parameter $\boldsymbol{\theta}$.

4.3 Model Comparison with the BIC. In general, none of the candidate models considered in an experiment will be identical to the actual underlying process that generates the data. Even the apparently best model will in general be wrong, a problem known as model misspecification (Burnham & Anderson, 2002). Therefore, it is sensible to view our method not as positively identifying the "true" model in the absolute sense, but rather demonstrating that one model is better than another. In the following, we evaluate the expected BIC difference and the expected BIC increment per trial under gaussian noise assumption and verify that in this situation, the BIC method will generally choose the model closest to the true model in the sense of least-square error. We show that the theoretical results are compatible with our numerical simulations using the optimal comparison stimuli.

Consider two models f_1 and f_2, neither of which is identical to the unknown true model F, which produces responses with gaussian noise having fixed variance σ^2 so that

$$r = F(\mathbf{x}) + \epsilon\,, \tag{4.24}$$

where $\epsilon \sim N(0, \sigma^2)$. Given a data set of n stimuli $\mathbf{x}_1, \ldots, \mathbf{x}_n$ and their elicited responses $r_1, \ldots, r_n$ generated by equation 4.24, we write the BIC for models f_1 and f_2 as

$$\mathrm{BIC}_1 = -\frac{1}{2\sigma^2}\sum_{i=1}^{n}(r_i - f_1(\mathbf{x}_i))^2 - \frac{K_1}{2}\ln n\,, \tag{4.25}$$

$$\mathrm{BIC}_2 = -\frac{1}{2\sigma^2}\sum_{i=1}^{n}(r_i - f_2(\mathbf{x}_i))^2 - \frac{K_2}{2}\ln n\,, \tag{4.26}$$

where K_1 and K_2 are the numbers of free parameters in the two models. Consider

$$\begin{aligned}\mathrm{BIC}_2 - \mathrm{BIC}_1 &= \frac{1}{2\sigma^2}\sum_{i=1}^{n}\left[2r_i\,(f_2(\mathbf{x}_i) - f_1(\mathbf{x}_i)) + f_1(\mathbf{x}_i)^2 - f_2(\mathbf{x}_i)^2\right] \\ &\quad + \frac{K_1 - K_2}{2}\ln n\,.\end{aligned} \tag{4.27}$$

Taking the expectation $\langle\ \rangle$ of equation 4.28 over the responses generated by the true model and using the fact that $\langle r_i \rangle = F(\mathbf{x}_i)$ by equation 4.24, we obtain

$$\langle \mathrm{BIC}_2 - \mathrm{BIC}_1 \rangle = \frac{E_1 - E_2}{2\sigma^2} + \frac{K_1 - K_2}{2}\ln n\,, \tag{4.28}$$

where

$$E_1 = \sum_{i=1}^{n} \left| F(\mathbf{x}_i) - f_1(\mathbf{x}_i) \right|^2, \quad E_2 = \sum_{i=1}^{n} \left| F(\mathbf{x}_i) - f_2(\mathbf{x}_i) \right|^2. \tag{4.29}$$

When f_1 and f_2 are of similar complexity, namely, $K_1 \approx K_2$, whichever of the two models that has the smaller discrepancy with F as measured by squared error over all stimuli will be the model preferred by the BIC. When the two models are of different complexity, the last term in equation 4.28 has the effect of biasing the preference toward the simpler model.

Next, we show how the slope of the differential $(\mathrm{BIC}_2 - \mathrm{BIC}_1)$ with respect to the number of stimuli is related to how well the candidate models approximate the true model $F(\mathbf{x})$. After having collected n data points, we now collect an additional data point $(\mathbf{x}, r) \equiv (\mathbf{x}_{n+1}, r_{n+1})$ for model comparison. It follows from equations 4.28 that the expected increment for this last data point should be

$$\begin{aligned} \langle \Delta (\mathrm{BIC}_2 - \mathrm{BIC}_1) \rangle &= \frac{|F(\mathbf{x}) - f_1(\mathbf{x})|^2 - |F(\mathbf{x}) - f_2(\mathbf{x})|^2}{2\sigma^2} \\ &\quad + \frac{K_1 - K_2}{2} \ln \frac{n+1}{n}. \end{aligned} \tag{4.30}$$

We can ignore the second term when n is large or when the two models have similar complexity ($K_1 \approx K_2$). Ignoring the second term in equation 4.31 and assuming that model 2 is true, namely, $F(\mathbf{x}) = f_2(\mathbf{x})$, we obtain

$$\langle \Delta (\mathrm{BIC}_2 - \mathrm{BIC}_1) \rangle \approx \frac{|f_1(\mathbf{x}) - f_2(\mathbf{x})|^2}{2\sigma^2}. \tag{4.31}$$

This equation is maximized for the $\mathbf{x}$ that best distinguishes the predictions of models 1 and 2, which is consistent with the utility function equation 4.2 or equation 4.11 as derived earlier.

Applying equation 4.31 to predict the increment $\Delta (\mathrm{BIC}_+ - \mathrm{BIC}_\times)$ for comparing the additive and multiplicative models in Figure 6b yielded reasonably good predictions, as we can see from Figure 6e (correlation coefficient $R = 0.59$ for 100 trials). Equation 4.31 is justified here because the two models have almost the same number of parameters ($K_+ = 292$ and $K_\times = 291$). For each data point in Figure 6e, the predicted value was based on equation 4.31, averaged over all 500 stimuli used for each individual trial, while the observed value was taken as the slope of each red line like those shown in Figure 6d (right panel). These lines are very close to straight lines (median correlation coefficient $R = 0.9999$ for 100 trials). We also applied equation 4.28 to directly predict the final value of $(\mathrm{BIC}_+ - \mathrm{BIC}_\times)$ at the end

of the comparison phase. The result was similar to that in Figure 6e with correlation coefficient $R = 0.58$ for 100 trials (not shown).

5 Discussion

With recent advances in computing power, adaptive stimulus generation has become a potentially powerful tool for sensory neurophysiology (Benda et al., 2007). Adaptive methods have been applied experimentally to maximize the firing rate response of a sensory neuron (Harth & Tzanakou, 1974; Nelken, Pruta, Vaadiaa, & Abeles, 1994; Bleeck, Patterson, & Winter, 2003; O'Connor, Petkov, & Sutter, 2005; Yamane, Carlson, Bowman, Wang, & Connor, 2008), to find stimulus ensembles that maximize the mutual information between stimuli and responses (Machens, 2002; Machens, Gollisch, Kolesnikova, & Herz, 2005), and to find stimuli that are optimally designed to estimate the parameters of some assumed response model (Lewi et al., 2009). Our study complements previous work on optimal experimental design for model estimation by considering its application to hierarchical nonlinear neural models and the problem of online model comparison.

Optimal experimental design methods have been used in various disciplines including statistics and psychology (Atkinson & Donev, 1992; Myung, 2000; Pitt, Myung, & Zhang, 2002; Wang & Simoncelli, 2008; Cavagnaro et al., 2010), but rarely in neuroscience besides the recent application to estimating a generalized linear model (Lewi et al., 2009, 2011). Furthermore, the problem of model comparison has rarely been studied within sensory systems neuroscience (Vladusich, Lucassen, & Cornelissen, 2006). We have proposed a general two-stage computational procedure for adaptively generating stimuli online that are optimal for estimating and distinguishing competing nonlinear models. The method may potentially facilitate the use of nonlinear mathematical models for quantifying sensory neurons by reducing the number of stimuli needed in neurophysiological experiments. Although our demonstration in this letter has been focused on hierarchical networks, the method may generalize to other input-output systems as well.

One can uniquely recover the parameters of a neural network correctly only if the model is identifiable (Bellman & Astrom, 1961; Bamber & van Santen, 2000); that is, two networks with different parameters need to respond differently to some stimuli. However, even if the two networks have distinct stimulus-response properties, when a limited set of noisy data is used for network parameter estimation, as is always the case in practice, we may still observe a continuum of network parameters giving rise to nearly identical input-output functionality, making correct identification very difficult (DiMattina & Zhang, 2010). The adaptive stimulus design approach may provide a practical method for alleviating the problem of parameter confounding when estimating the parameters of a nonlinear network.

We have illustrated the usefulness of our procedure by identifying a generic center-surround network and a more complicated neuron model that generates nonlinear responses by integrating responses from multiple linear subunits (see Figure 2), as inspired by some real neurons (Lau et al., 2002; Prenger et al., 2004; Rust et al., 2005). This model resembles the spectral-temporal or spatiotemporal receptive fields, but with additional nonlinear operations downstream. This example uses stimuli with several hundreds of dimensions, typical of those employed in physiological studies (Wu et al., 2006).

Previous work has considered the identification of a class of nonlinear models known as generalized linear models or linear-nonlinear Poisson (LNP) models by optimal experimental design (Paninski, 2004). Our work complements these studies by studying the identification of standard nonlinear neural network models (Rumelhart et al., 1986), which are universal function approximators whose nonlinearity arises entirely from the connectionist network architecture. Such hierarchical models have been shown to be useful for describing neuronal responses in higher areas of the brain, like the ventral visual pathway (Riesenhuber & Poggio, 1999; Cadieu et al., 2007; Hinton, 2010). It would be useful to extend our analysis to include the additional forms of biological nonlinearity that are incorporated in the LNP class of models.

One general problem faced by the active data collection methods is the adaptation of neural responses (Carandini, Heeger, & Senn, 2002; Ulanovsky, Las, Farkas, & Nelken, 2004; Wehr and Zador, 2005; Asari & Zador, 2009; David, Mesgarini, Fritz, & Shamma, 2009). For example, one recent study utilizing a genetic algorithm to optimize neural responses to three-dimensional visual shapes observed a general decrease in the maximum neural firing rates as the experiment progressed, most likely due to repeated presentations of similar stimuli (Yamane et al., 2008). The methods presented here may potentially be subject to the same limitations caused by neural adaptation and therefore are likely to be applicable only to a subset of neurons that do not show strong adaptive effects, at least under some stimulus conditions. On the other hand, we observe that consecutive stimuli generated by optimal design often differ from one another (see Figure 1c) and also tend to drive a neuron over a wide range of firing rate rather than only toward the maximum firing rate (see Figure 1b). As a consequence, in the presence of stimulus-specific adaptation, the optimally designed stimuli might be more robust than adaptive methods that seek to drive a neuron to its maximum firing rate because in the latter cases, the stimuli might be restricted to an even smaller region in the stimulus space (Harth & Tzanakou, 1974; Nelken et al., 1994; Bleeck et al., 2003; O'Connor et al., 2005). Lewi et al. (2009) have considered some forms of parameter drifts in their experimental design. For future research, it might be useful to further develop biologically realistic models of the adaptation processes and take their effects into account when stimuli are designed.

In reality, we seldom know the true structure of the underlying network but may have several alternative and competing candidates. We have shown that stimuli designed to optimally distinguish different models work better at finding the correct model than both nonadaptive random stimuli and adaptive stimuli designed for estimation only (see Figure 6). One valid criticism of this approach is that it is unlikely that the exactly correct model will be in the set of candidate models, a problem known as model misspecification (Burnham & Anderson, 2002). While this is a general problem of the system identification approach, we have shown that with well-chosen data, the BIC procedure will choose the model that is closest to the truth in a well-defined sense (see section 4.3).

Optimally designed stimuli depend on the mathematical models of the stimulus-response relationship as well as the actual neuronal responses. These stimuli cannot be precomputed and have to be generated on the fly during neurophysiological experiments. For practical applications, the most time-consuming part of our algorithms is the optimization procedures, which require the evaluation of complicated functions in high-dimensional spaces. However, when the heuristic methods described in this letter are used, stimuli of similar complexity to those used in neurophysiology experiments (over a 100 stimulus dimensions) can be generated in about 1 second on a desktop PC (2.4 GHz quad core), showing that our method is already within the timescale of feasible neurophysiological experiments. As the power of computers keeps increasing, adaptive stimulus generation may eventually become a standard method in systems neuroscience for estimating and comparing ever more complex models in online experiments.

Acknowledgments

We thank Giovanni Parmigiani for advice and discussion and two anonymous reviewers for helpful suggestions. C.D. also thanks his postdoctoral advisor, Michael S. Lewicki, for his generous support for this project. This work was supported by grant NSF IIS-0827695.

References

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

Ahrens, M. B., Paninski, L., & Sahani, M. (2008). Inferring input nonlinearities in neural encoding models. *Network, 21*, 35–67.

Alspach, D. & Sorenson, H. (1972). Nonlinear Bayesian estimation using gaussian sum approximation. *IEEE Trans. Auto. Contr., 17*, 439–448.

Asari, H., & Zador, A. M. (2009). Long-lasting context dependence constrains neural encoding models in rodent auditory cortex. *J. Neurophysiol., 102*, 2638–2656.

Atkinson, A. C., & Donev, A. N. (1992). *Optimum experimental designs*. Oxford: Clarendon Press.

Atkinson, A. C., & Fedorov, V. V. (1975a). The design of experiments for discriminating between two models. *Biometrika, 62*, 57–70.

Atkinson, A. C., & Fedorov, V. V. (1975b). Optimal design: Experiments for discriminating between several models. *Biometrika, 62*, 289–303.

Bamber, D., & van Santen, J.P.H. (2000). How to assess a model's testability and identifiability. *J. Math. Psychol., 44*, 20–40.

Bandyopadhyay, S., Reiss, L. A., & Young, E. D. (2007). Receptive fields for dorsal cochlear nucleus neurons at multiple sound levels. *J. Neurophysiol., 98*, 3505–3515.

Bellman, R., & Astrom, K. J. (1961). On structural identifiability. *Mathematical Biosciences, 7*, 329–339.

Benda, J., Gollisch, T., Machens, C. K., & Herz, A. V. (2007). From response to stimulus: Adaptive sampling in sensory physiology. *Curr. Opin Neurobiol., 17*, 430–436.

Bengtsson, T., Bickel, P., & Li, B. (2008). Curse-of-dimensionality revisited: Collapse of the particle filter in very large scale systems. In D. Nolan & T. Speed (Eds.), *Probability and statistics: Essays in honor of David A. Freedman* (pp. 316–334). Beechwood, OH: Institute of Mathematical Statistics.

Bickel, P., Li, B., & Bengtsson, T. (2008). Sharp failure rates for the bootstrap particle filter in high dimensions. In B. Clarke & S. Ghosal (Eds.), *Pushing the limits of contemporary statistics: Contributions in honor of Jayanta K. Ghosh* (pp. 318–329). Beechwood, OH: Institute of Mathematical Statistics.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Berlin: Springer.

Bleeck, S., Patterson, R. D., & Winter, I. M. (2003). Using genetic algorithms to find the most effective stimulus for sensory neurons. *Journal of Neuroscience Methods, 125*, 73–82.

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

Burnham, K. P., & Anderson, D. R. (2002). *Model selection and multimodel inference* (2nd ed.). Berlin: Springer.

Cadieu, C., Kouh, M., Pasupathy, A., Connor, C. E., Riesenhuber, M., & Poggio, T. (2007). A model of v4 shape selectivity and invariance. *J. Neurophysiol., 98*, 1733–1750.

Carandini, M., Heeger, D. J., & Senn, W. (2002). A synaptic explanation of suppression in visual cortex. *Journal of Neuroscience, 22*, 10053–10065.

Carpenter, J., Clifford, P., & Fearnhead, P. (1999). Improved particle filter for nonlinear problems. *IEE Proceedings–F, 146*, 2–7.

Cavagnaro, D. R., Myung, J. I., Pitt, M. A., & Kujala, J. V. (2010). Adaptive design optimization: A mutual information based approach to model discrimination in cognitive science. *Neural Comput., 22*, 887–905.

Chaloner, K., & Verdinelli, I. (1995). Bayesian experimental design: A review. *Stat. Sci., 10*, 273–304.

Chen, X., Han, F., Poo, M. M., & Dan, Y. (2007). Excitatory and suppressive receptive field subunits in awake monkey primary visual cortex. *Proc. Nat. Acad. Sci., 104*, 19120–19125.

Cohn, D. A. (1996). Neural network exploration using optimal experimental design. *Neural Netw., 9*, 1071–1083.

Cohn, D. A., Ghahramani, Z., & Jordan, M. I. (1996). Active learning with statistical models. *J. Artif. Intell. Res., 4*, 129–145.

Cover, T. M., & Thomas, J. A. (2006). *Elements of information theory* (2nd ed.). Hoboken, NJ: Wiley.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems, 2*, 303–314.

David, S. V., & Gallant, J. L. (2005). Predicting neuronal responses during natural vision. *Network, 16*, 239–260.

David, S. V., Mesgarini, N., Fritz, J. B., & Shamma, S. A. (2009). Rapid synaptic depression explains nonlinear modulation of spectro-temporal tuning in primary auditory cortex by natural stimuli. *J. Neurosci., 29*, 3374–3386.

David, S. V., Vinje, W. E., & Gallant, J. L. (2004). Natural stimulus statistics alter the receptive field stucture of v1 neurons. *J. Neurosci., 24*, 6991–7006.

Dean, A. F. (1981). The variability of discharge of simple cells in the cat striate cortex. *Exp. Brain Res., 44*, 437–440.

DiCarlo, J. J., Johnson, K. O., & Hsiao, S. S. (1998). Structure of receptive fields in area 3b of primary somatosensory cortex in the alert monkey. *J. Neurosci., 18*, 2626–2645.

DiMattina, C., & Zhang, K. (2008). How optimal stimuli for sensory neurons are constrained by network architecture. *Neural Comput., 20*, 668–708.

DiMattina, C., & Zhang, K. (2010). How to modify a neural network gradually without changing its input-output functionality. *Neural Comput., 22*, 1–47.

Freund, Y., Seung, H. S., Shamir, E., & Tishby, N. (1997). Selective sampling using the query by committee algorithm. *Machine Learning, 28*, 133–168.

Gilks, W. R., Richardson, S., & Spiegelhalter, D. (1995). *Markov chain Monte Carlo in practice*. London: Chapman & Hall/CRC.

Gordon, N. J., Salmond, D. J., & Smith, A. F. M. (1993). Novel approach to nonlinear/non-gaussian Bayesian state estimation. *IEE Proceedings–F, 140*, 107–113.

Gramacy, R. B., & Polson, N. G. (2009). *Particle learning of gaussian process models for sequential design and optimization.* http://arxiv.org/pdf/0909.5262.

Harth, E., & Tzanakou, E. (1974). Alopex: A stochastic method for determining visual receptive fields. *Vision Research, 14*, 1475–1482.

Harville, D. A. (1997). *Matrix algebra from a statistician's perspective*. Berlin: Springer.

Haykin, S. (Ed.). (2001). *Kalman filtering and neural networks*. Hoboken, NJ: Wiley.

Hering, P., & Simandl, M. (2007). Gaussian sum approach with optimal experimental design forneural network. In *Ninth IASTED Conference on Signal and Image Processing* (pp. 425–430). ACTA Press.

Hinton, G. E. (2010). Learning to represent visual input. *Philosophical Transactions of the Royal Society B, 356*, 177–184.

Hornik, K., Stinchcombe, M., & White, H. (1989). Multi-layer feed-forward neural networks are universal approximators. *Neural Netw., 2*, 359–366.

Jones, J. P., & Palmer, L. A. (1987). The two-dimensional spatial structure of simple receptive fields in cat striate cortex. *J. Neurophysiol., 58*, 1187–1211.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME—Journal of Basic Engineering, 82* (Series D), 35–45.

Kay, S. M. (1993). *Fundamentals of statistical signal processing: Estimation theory*. Upper Saddle River, NJ: Prentice Hall.

Kujala, J. V., & Lukka, T. J. (2006). Bayesian adaptive estimation: The next dimension. *J. Math. Psychol., 50*, 369–389.

Kullback, S., & Leibler, R. A. (1951). On information and sufficiency. *Ann. Math. Stat., 22*, 79–86.

Lau, B., Stanley, G. B., & Dan, Y. (2002). Computational subunits of visual cortical neurons revealed by artificial neural networks. *Proc. Natl. Acad. Sci. USA, 99*, 8974–8979.

Lehky, S. R., Sejnowski, T. J., & Desimone, R. (1992). Predicting responses of nonlinear neurons in monkey striate cortex to complex patterns. *J. Neurosci., 12*, 3568–3581.

Lewi, J., Butera, R., & Paninski, L. (2009). Sequential optimal design of neurophysiology experiments. *Neural Comput., 21*, 619–687.

Lewi, J., Schneider, D. M., Woolley, S. M., & Paninski, L. (2011). Automating the design of informative sequences of sensory stimuli. *J. Comput. Neurosci., 30*, 181–200.

Liu, J. S. (2001). *Monte Carlo strategies in scientific computing*. Berlin: Springer.

Machens, C. K. (2002). Adaptive sampling by information maximization. *Phys. Rev. Lett., 88*, 228104.

Machens, C. K., Gollisch, T., Kolesnikova, O., & Herz, A. V. (2005). Testing the efficiency of sensory coding with optimal stimulus ensembles. *Neuron, 47*, 447–456.

MacKay, D. J. C. (1992). Information based objective functions for active data collection. *Neural Comput., 4*, 448–472.

Marmarelis, P. Z., & Marmarelis, V. Z. (1978). *Analysis of physiological systems: The white-noise approach*. New York: Plenum Press.

McCullagh, P., & Nelder, J. A. (1989). *Generalized linear models*. London: Chapman Hall/CRC.

Minsky, M. L., & Papert, S. A. (1988). *Perceptrons: An introduction to computational geometry* (Exp. ed.). Cambridge, MA: MIT Press.

Muller, P., & Parmigiani, G. (1995). Optimal design via curve fitting of Monte Carlo experiments. *J. Amer. Stat. Assn., 90*, 1322–1330.

Myung, J. (2000). The importance of complexity in model selection. *J. Math. Psychol., 44*, 190–204.

Nelken, I., Pruta, Y., Vaadiaa, E., & Abeles, M. (1994). In search of the best stimulus: An optimization procedure for finding efficient stimuli in the cat auditory cortex. *Hearing Research, 72*, 237–253.

Nelson, J. (2005). Finding useful questions: On Bayesian diagnosticity, probability, impact, and information gain. *Psychol. Rev., 112*, 979–999.

O'Connor, K. N., Petkov, C. I., & Sutter, M. L. (2005). Adaptive stimulus optimization for auditory cortical neurons. *J. Neurophysiol., 94*, 4051–4067.

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

Paninski, L. (2005). Asymptotic theory of information-theoretic experimental design. *Neural Comput., 17*, 1480–1507.

Pitt, M. A., Myung, J., & Zhang, S. (2002). Toward a method of selecting among computational models of cognition. *Psychol. Review, 3*, 472–491.

Prenger, R., Wu, M. C., David, S. V., & Gallant, J. L. (2004). Nonlinear V1 responses to natural scenes revealed by neural network analysis. *Neural Netw., 17*, 663–679.

Riesenhuber, M., & Poggio, T. (1999). Hierarchical models of object recognition in cortex. *Nat. Neurosci., 2*, 1019–1025.

Rumelhart, D. E., Hinton, G. E., & McClelland, J. L. (1986). Learning internal representations by error propagation. In D. E. Rumelhart & J. L. McClelland (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition*. Cambridge, MA: MIT Press.

Rust, N. C., Schwartz, O., Movshon, J. A., & Simoncelli, E. P. (2005). Spatiotemporal elements of macaque V1 receptive fields. *Neuron, 46*, 945–956.

Schinkel-Bielefeld, N., David, S. V., Shamma, S. A., & Butts, D. A. (2010). Identification of excitation and inhibition in the auditory cortex using nonlinear modeling. In *Frontiers in Systems Neuroscience: Conference Abstract, COSYNE 2010*. Lausanne, Switzerland: Frontiers Media.

Schwarz, G. (1978). Estimating the dimension of a model. *Ann. Stat., 6*, 461–464.

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

Snyder, C., Bengtsson, T., Bickel, P., & Anderson, J. (2008). Obstacles to high-dimensional particle filtering. *Monthly Weather Review, 136*, 4629–4640.

Sugiyama, M., & Rubens, N. (2008). A batch ensemble approach to active learning with model selection. *Neural Netw., 21*, 1278–1286.

Theunissen, F. E., Sen, K., & Doupe, A. J. (2000). Spectral-temporal receptive fields of nonlinear auditory neurons obtained using natural sounds. *J. Neurosci., 20*, 2315–2331.

Ulanovsky, N., Las, L., Farkas, D., & Nelken, I. (2004). Multiple time scales of adaptation in auditory cortex neurons. *Journal of Neuroscience, 24*, 10440–10453.

van der Vart, A. W. (1998). *Asymptotic statistics*. Cambridge: Cambridge University Press.

van Hateren, J. H., & van der Schaaf, A. (1998). Independent component filters of natural images compared with simple cells in primary visual cortex. *Proc. Roy. Soc. B: Biol. Sci., 265*, 359–366.

Vladusich, T., Lucassen, M. P., & Cornelissen, F. W. (2006). Do cortical neurons process luminance or contrast to encode surface properties? *J. Neurophysiol., 95*, 2638–2649.

Wang, Z., & Simoncelli, E. P. (2008). Maximum differentiation (MAD) competition: A methodology for comparing computational models of perceptual quantitites. *J. Vis., 8*, 1–13.

Watson, A. B., & Pelli, D. G. (1983). Quest: A Bayesian adaptive psychometric method. *Percept. Psychophys., 33*, 113–120.

Wehr, M., & Zador, A. M. (2005). Synaptic mechanisms of forward suppression in rat auditory cortex. *Neuron, 47*, 437–445.

Wu, M. C., David, S. V., & Gallant, J. L. (2006). Complete functional characterization of sensory neurons by system identification. *Ann. Rev. Neurosci., 29*, 477–505.

Yamane, Y., Carlson, E. T., Bowman, K. C., Wang, Z., & Connor, C. E. (2008). A neural code for three-dimensional shape in macaque inferotemporal cortex. *Nat. Neurosci, 11*, 1352–1360.

Young, E. D., & Davis, K. A. (2002). Circuitry and function of the dorsal cochlear nucleus. In D. Oertel, R. R. Fay, & A. N. Popper (Eds.), *Integrative functions of the mammalian auditory pathway*. Berlin: Springer.

Yu, J. J., & Young, E. D. (2000). Linear and nonlinear pathways of spectral information transmission in the cochlear nucleus. *Proc. Natl. Acad. Sci. USA, 97*, 11780–11786.

Zipser, D., & Andersen, R. A. (1988). A back-propagation programmed network that simulates the response properties of a subset of posterior parietal neurons. *Nature, 331*, 679–684.

Received August 31, 2010; accepted March 3, 2011.