ARTICLE Communicated by Christian Machens

Sequential Optimal Design of Neurophysiology Experiments

Jeremy Lewi
jeremy@lewi.us
Bioengineering Graduate Program, Wallace H. Coulter Department of Biomedical Engineering, Laboratory for Neuroengineering, Georgia Institute of Technology, Atlanta, GA 30332, U.S.A. http://www.lewilab.org

Robert Butera
rbutera@ece.gatech.edu
School of Electrical and Computer Engineering, Laboratory for Neuroengineering, Georgia Institute of Technology, Atlanta, GA 30332, U.S.A.

Liam Paninski
liam@stat.columbia.edu
Department of Statistics and Center for Neurotheory, Columbia University, New York, NY 10027, U.S.A. http://www.stat.columbia.edu/∼liam

Adaptively optimizing experiments has the potential to significantly reduce the number of trials needed to build parametric statistical models of neural systems. However, application of adaptive methods to neurophysiology has been limited by severe computational challenges. Since most neurons are high-dimensional systems, optimizing neurophysiology experiments requires computing high-dimensional integrations and optimizations in real time. Here we present a fast algorithm for choosing the most informative stimulus by maximizing the mutual information between the data and the unknown parameters of a generalized linear model (GLM) that we want to fit to the neuron's activity. We rely on important log concavity and asymptotic normality properties of the posterior to facilitate the required computations. Our algorithm requires only low-rank matrix manipulations and a two-dimensional search to choose the optimal stimulus. The average running time of these operations scales quadratically with the dimensionality of the GLM, making real-time adaptive experimental design feasible even for high-dimensional stimulus and parameter spaces. For example, we require roughly 10 milliseconds on a desktop computer to optimize a 100-dimensional stimulus. Despite using some approximations to make the algorithm efficient, our algorithm asymptotically decreases the uncertainty about the model parameters at a rate equal to the maximum rate predicted by an asymptotic analysis. Simulation results show that picking stimuli by maximizing the mutual information can speed up convergence to the optimal values of the parameters by an order of magnitude compared

to using random (nonadaptive) stimuli. Finally, applying our design procedure to real neurophysiology experiments requires addressing the nonstationarities that we would expect to see in neural responses; our algorithm can efficiently handle both fast adaptation due to spike history effects and slow, nonsystematic drifts in a neuron's activity.

1 Introduction

In most neurophysiology experiments, data are collected according to a design that is finalized before the experiment begins. During the experiment, the data already collected are rarely analyzed to evaluate the quality of the design. These data, however, often contain information that could be used to redesign experiments to better test hypotheses (Fedorov, 1972; Chaloner & Verdinelli, 1995; Kontsevich & Tyler, 1999; Warmuth et al., 2003; Roy, Ghosal, & Rosenberger, in press). Adaptive experimental designs are particularly valuable in domains where data are expensive or limited. In neuroscience, experiments often require training and caring for animals, which can be time-consuming and costly. As a result of these costs, neuroscientists are often unable to conduct large numbers of trials using different subjects. The inability to collect enough data makes it difficult for them to investigate high-dimensional, complex neural systems. By using adaptive experimental designs, neuroscientists could potentially collect data more efficiently. In this article, we develop an efficient algorithm for optimally adapting the experimental design in one class of neurophysiology experiments.

A central question in neuroscience is understanding how neural systems respond to different inputs. For sensory neurons, the input might be sounds or images transduced by the organism's receptors. More generally, the stimulus could be a chemical or electrical signal applied directly to the neuron. Neurons often respond nonlinearly to these stimuli because their activity will typically adapt or saturate. We can model these nonlinearities by viewing a neuron's firing rate as a variable dependent on its past activity in addition to recent stimuli. To model the dependence on past stimuli and responses, we define the input as a vector comprising the current and recent stimuli, $\{\vec{x}_t, \vec{x}_{t-1}, \ldots, \vec{x}_{t-t_k}\}$, as well as the neuron's recent activity, $\{r_{t-1}, \ldots, r_{t-t_a}\}$ (Keat, Reinagel, Reid, & Meister, 2001; Truccolo, Eden, Fellows, Donoghue, & Brown, 2005). $\vec{x}_t$ and r_t denote the stimulus and firing rate at time t, respectively. When we optimize the input for time $t+1$, we can control only $\vec{x}_{t+1}$, as the rest of the components of the input (i.e., past stimuli and responses) are fixed. To distinguish the controllable and fixed components of the input, we use the subscripts x and f:

$$\vec{s}_t = \left[\vec{x}_t^T, \vec{s}_{f,t}^T\right]^T \tag{1.1}$$

$$\vec{s}_{x,t} = \vec{x}_t \tag{1.2}$$

$$\vec{s}_{f,t} = \left[\vec{x}_{t-1}^T, \ldots, \vec{x}_{t-t_k}^T, r_{t-1}, \ldots, r_{t-t_a}\right]^T. \tag{1.3}$$

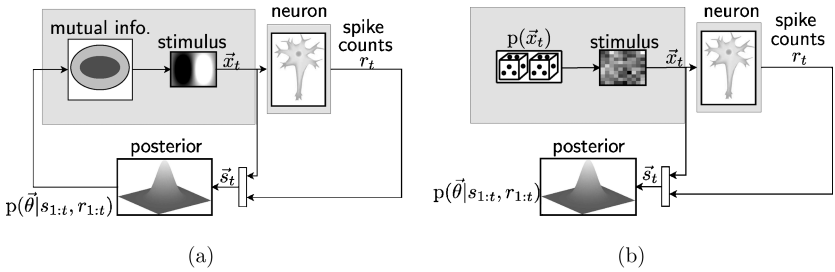

Figure 1: (a) Schematic of the process for designing information-maximizing (infomax) experiments. Stimuli are chosen by maximizing the mutual information between the data and the parameters. Since the mutual information depends on the posterior distribution on $\vec{\theta}$, the infomax algorithm updates the posterior after each trial. (b) Schematic of the typical independent and identically distributed (i.i.d.) design of experiments. Stimuli are selected by drawing i.i.d. samples from a distribution that is chosen before the experiment starts. An i.i.d. design does not use the posterior distribution to choose stimuli.

$\vec{s}_t$ is the input at time t. $\vec{s}_{f,t}$ is a vector comprising the past stimuli and responses on which the response at time t depends. t_k and t_a are how far back in time the dependence on the stimuli and responses stretches (i.e., if $t_k = 0$ and $t_a = 0$, then $\vec{s}_t = \vec{x}_t$). Not all models will include a dependence on past stimuli or responses; the values of t_k and t_a will depend on the model adopted for a particular experiment.

We can describe a model that incorporates all of these features by specifying the conditional distribution of the responses given the input. This distribution gives the probability of observing response r_t at time t given the input $\vec{s}_t$. We use a distribution as opposed to a deterministic function to specify the relationship between r_t and $\vec{s}_t$ because a neuron's response varies for repeated presentations of a stimulus. To simplify the model, we restrict our consideration to parametric distributions that lie in some space Θ. Each vector $\vec{\theta}$ denotes a particular model in this space. To fit a model, $p(r_t \mid \vec{s}_t, \vec{\theta})$, to a neuron, we need to find the best value of $\vec{\theta}$.

We estimate $\vec{\theta}$ by observing the neuron's response to various stimuli. For these experiments, the design is a procedure for picking the stimulus on each trial. The design can be specified as a probability distribution, $p(\vec{x}_t)$, from which we sample the stimulus on each trial. Nonrandom designs can be specified by putting all the probability mass on a single stimulus. A sequential design modifies this distribution after each observation. In contrast, the standard nonsequential approach is to fix this distribution before the experiment starts, and then select the stimulus on each trial by drawing independent and identically distributed (i.i.d.) samples from $p(\vec{x}_t)$. Figure 1 provides a schematic of the sequential approach we want to implement, as well as a diagram of the typical i.i.d. design.

We want to design our experiments to facilitate identification of the best model in Θ. Based on this objective, we define the optimal design for each trial as the design that provides the most information about $\vec{\theta}$. A natural metric for the informativeness of a design is the mutual information between the data and the model (Lindley, 1956; Bernardo, 1979; Watson & Pelli, 1983; Cover & Thomas, 1991; MacKay, 1992; Chaloner & Verdinelli, 1995; Paninski, 2005),

$$I(\{r_t, \vec{s}_t\}; \vec{\theta}) = \int p(r_t, \vec{s}_t, \vec{\theta}) \frac{\log p(r_t, \vec{s}_t, \vec{\theta})}{\log p(r_t, \vec{s}_t) p(\vec{\theta})} dr_t d\vec{s}_t d\vec{\theta}. \tag{1.4}$$

The mutual information measures how much we expect the experimental data to reduce our uncertainty about $\vec{\theta}$. The mutual information is a function of the design because it depends on the joint probability of the data, $p(r_t, \vec{s}_t)$, which obviously depends on how we pick the stimuli. We can determine the optimal design by maximizing the mutual information with respect to the marginal distribution $p(\vec{s}_{x,t} = \vec{x}_t)$.

Designing experiments by maximizing the mutual information is computationally challenging. The information we expect to gain from an experiment depends on what we have already learned from past observations. To extract the information from past observations, we need to compute the posterior distribution $p(\vec{\theta} \mid \{r_t, r_{t-1}, \ldots, r_1\}, \{\vec{s}_t, \vec{s}_{t-1}, \ldots, \vec{s}_1\})$ after each trial. Once we have updated the posterior, we need to use it to compute the expected information gain from future experiments; this requires a high-dimensional integration over the space Θ. Maximizing this integral with respect to the design requires a nonlinear search over the high-dimensional stimulus space, $\mathcal{X}$. In sensory neurophysiology, the stimulus space is high-dimensional because the stimuli tend to be complex, spatiotemporal signals like movies and sounds. The challenge of evaluating this high-dimensional integral and solving the resulting nonlinear optimization has impeded the application of adaptive experimental design to neurophysiology. In the worst case, the complexity of these operations will grow exponentially with the dimensionality of $\vec{\theta}$ and $\vec{s}_t$. For even moderately sized spaces, direct computation will therefore be intractable, particularly if we wish to adapt the design in a real-time application.

The main contribution of this article is to show how these computations can be performed efficiently when Θ is the space of generalized linear models (GLM) and the posterior distribution on $\vec{\theta}$ is approximated as a gaussian. Our solution depends on some important log-concavity and rank-one properties of our model. These properties justify the gaussian approximation of the posterior distribution and permit a rapid update after each trial. These properties also allow optimization of the mutual information to be approximated by a tractable two-dimensional problem that can be solved numerically. The solution to this 2D optimization problem depends on the

stimulus domain. When the stimulus domain is defined by a power constraint, we can easily find the nearly optimal design. For arbitrary stimulus domains, we present a general algorithm for selecting the optimal stimulus from a finite subset of stimuli in the domain. Our analysis leads to efficient heuristics for constructing this subset to ensure the resulting design is close to the optimal design.

Our algorithm facilitates estimation of high-dimensional systems because picking more informative designs leads to faster convergence to the best model of the neuron. In our simulations (see section 5.4), the optimal design converges more than an order of magnitude faster than an i.i.d. design. Our algorithm can be applied to high-dimensional, real-time applications because it reduces the complexity with respect to dimensionality from exponential to on average quadratic running time.

This article is organized as follows. In section 2, we present the GLM of neural systems. In section 3, we present an online method for computing a gaussian approximation of the posterior distribution on the GLM's parameters. In section 4, we show how the mutual information, $I(r_t; \vec{\theta} \mid \vec{s}_t)$, can be approximated by a much simpler, low-dimensional function. In section 5, we present the procedure for picking optimal stimuli and show some simulation results. In section 6, we generalize our basic methods to some important extensions of the GLM needed to handle more complicated experiments. In section 7, we show that our algorithm asymptotically decreases the uncertainty about $\vec{\theta}$ at a rate nearly equal to the optimal rate predicted by a general theorem on the rate of convergence of information maximizing designs (Paninski, 2005). We therefore conclude that this efficient (albeit approximate) implementation produces designs that are in fact asymptotically optimal. Simulations investigating the issue of model misspecification are presented in section 8. Finally, we discuss some limitations and directions for future work in section 9. To help the reader, we summarize in Table 1 the notation that we will use in the rest of the article.

2 The Parametric Model

For the model space, Θ, we choose the set of generalized linear models (GLM) (see Figure 2). The GLM is a tractable and flexible parametric family that has proven useful in neurophysiology (McCullagh & Nelder, 1989; Simoncelli, Paninski, Pillow, & Schwartz, 2004; Paninski, 2004; Truccolo et al., 2005; Paninski, Pillow, & Lewi, 2007). GLMs are fairly natural from a physiological point of view, with close connections to biophysical models such as the integrate-and-fire cell. Consequently, they have been applied in a wide variety of experimental settings (Brillinger, 1988, 1992; Chichilnisky, 2001; Theunissen et al., 2001; Paninski, Shoham, Fellows, Hatsopoulos, & Donoghue, 2004).

A GLM model represents a spiking neuron as a point process. The likelihood of the response, the number of spikes, depends on the firing rate, λ_t,

Table 1: Definitions of Symbols and Conventions Used Throughout the Article.

$\vec{x}_t$	Stimulus at time t
r_t	Response at time t
$\vec{s}_t = [\vec{s}_{x,t}^T, \vec{s}_{f,t}^T]^T$	Complete input at time t
$\vec{s}_{x,t}$	Controllable part of the input at time t
$\vec{s}_{f,t}$	Fixed part of the input at time t
$\boldsymbol{x_{1:t}} \triangleq \{\vec{x}_1, \ldots, \vec{x}_t\}$	Sequence of stimuli up to time t; boldface denotes a matrix.
$\boldsymbol{r_{1:t}} \triangleq \{r_1, \ldots, r_t\}$	Sequence of observations up to time t
$\boldsymbol{s_{1:t}} \triangleq \{\vec{s}_1, \ldots, \vec{s}_t\}$	Sequence of inputs up to time t
$E_\omega(\omega) = \int p(\omega)\omega \, d\omega.$	Expectation with respect to the distribution on the random variable denoted in the subscript
$H(p(\omega \mid \gamma)) \triangleq \int -p(\omega \mid \gamma) \log p(\omega \mid \gamma) \, d\omega.$	Entropy of the distribution $p(\omega \mid \gamma)$
$d = \dim(\vec{\theta})$	Dimensionality of the model
$p(\vec{\theta} \mid \vec{\mu}_t, \boldsymbol{C}_t)$	Gaussian approximation of the posterior distribution, $p(\vec{\theta} \mid \boldsymbol{s_{1:t}}, \boldsymbol{r_{1:t}})$; $(\vec{\mu}_t, \boldsymbol{C}_t)$ are the mean and covariance matrix, respectively

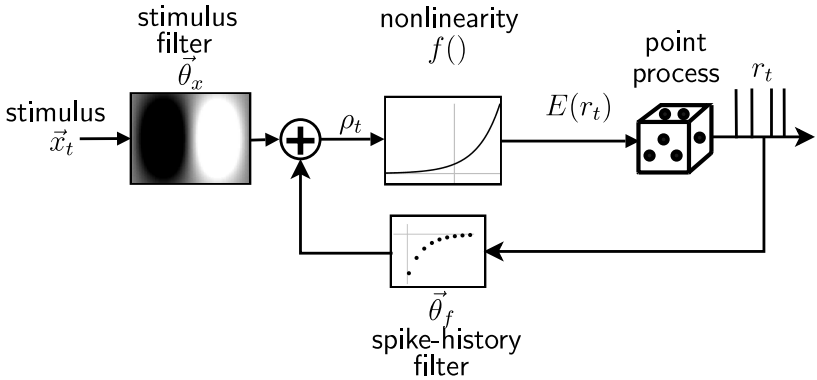

Figure 2: Diagram of a general linear model of a neuron. A GLM consists of a linear filter followed by a static nonlinearity. The output of this cascade is the estimated, instantaneous firing rate of a neuron. The unknown parameters $\vec{\theta} = [\vec{\theta}_x^T, \vec{\theta}_f^T]^T$ are the linear filters applied to the stimulus and spike history.

which is a nonlinear function of the input,

$$\lambda_t = E_{r_t \mid \vec{s}_t, \vec{\theta}}(r_t) = f\left(\vec{\theta}^T \vec{s}_t\right) = f\left(\vec{\theta}_x^T \vec{s}_{x,t} + \vec{\theta}_f^T \vec{s}_{f,t}\right). \tag{2.1}$$

As noted earlier, the response at time t depends on the current stimulus, $\vec{x}_t$, as well as past stimuli and responses. The inclusion of spike history in the

input means we can account for refractory effects, burstiness, and firing-rate adaptation (Berry & Meister, 1998; Keat et al., 2001; Paninski, 2004; Truccolo et al., 2005). As noted earlier, we use subscripts to distinguish the components that we can control from those that are fixed (see Table 1).

The parameters of the GLM are the coefficients of the filter, $\vec{\theta}$, applied to the input. $\vec{\theta}$ can be separated into two filters $\vec{\theta} = [\vec{\theta}_x^T, \ \vec{\theta}_f^T]^T$, which are applied to the variable and fixed components of the input, respectively. After filtering the input by $\vec{\theta}$, the output of the filter is pushed through a static nonlinearity, $f()$, known as the link function. The input-output relationship of the neuron is fully specified by the log likelihood of the response given the input and $\vec{\theta}$,

$$\log p\left(r_t \mid \vec{s}_t, \vec{\theta}\right) = \log \frac{e^{-\lambda_t dt}(\lambda_t dt)^{r_t}}{r_t!} \tag{2.2}$$

$$= r_t \log f(\vec{\theta}^T \vec{s}_t) - f(\vec{\theta}^T \vec{s}_t)\, dt + const. \tag{2.3}$$

dt is the length of the time window over which we measure the firing rate, r_t. The constant term is constant with respect to $\vec{\theta}$ but not r_t. In this article, we always use a Poisson distribution for the conditional likelihood, $p(r_t \mid \vec{s}_t, \vec{\theta})$, because it is the best one for modeling spiking neurons. However, by making some minor modifications to our algorithm, we can use it with other distributions in the exponential family (Lewi, Butera, & Paninski, 2007).

To ensure the maximum a posteriori (MAP) estimate of $\vec{\theta}$ is unique, we restrict the GLM so that the log likelihood is always concave. When $p(r_t \mid \vec{s}_t, \vec{\theta})$ is a Poisson distribution, a sufficient condition for concavity of the log likelihood is that the nonlinearity $f()$ is a convex and log concave function (Wedderburn, 1976; Haberman, 1977; McCullagh & Nelder, 1989; Paninski, 2004). $f()$ can be convex and log concave only if its contours are linear. When the contours are linear, we can, without loss of generality, assume that $f()$ is a function of a scalar variable, ρ_t. ρ_t is the result of applying the linear filter of the GLM to the input,

$$\rho_t = \vec{\theta}^T \vec{s}_t. \tag{2.4}$$

Since ρ_t is a scalar, $\vec{\theta}$ must be a vector and not a matrix. Convexity of $f()$ also guarantees that the nonlinearity is monotonic. Since we can always multiply $\vec{\theta}$ by negative 1 (i.e., flip our coordinate system), we can without loss of generality assume that f is increasing. Furthermore, we assume $f()$ is known, although this condition could potentially be relaxed. Knowing $f()$ exactly is not essential because previous work (Li & Duan, 1989; Paninski, 2004) and our own results, (see section 8) indicate that the parameters of a GLM can often be estimated, at least up to a scaling factor, even if the link function is incorrect.

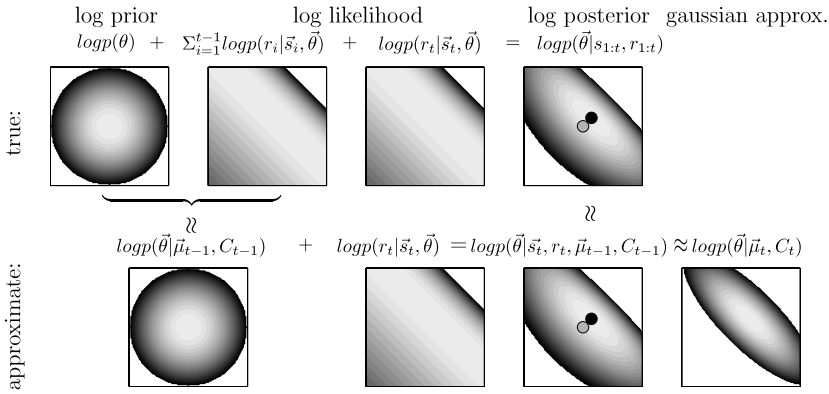

Figure 3: Schematic illustrating the procedure for recursively constructing the gaussian approximation of the true posterior; $\dim(\vec{\theta}) = 2$. The images are contour plots of the log prior, log likelihoods, log posterior, and log of the gaussian approximation of the posterior (see text for details). The key point is that since $p(r_t \mid \vec{s}_t, \vec{\theta})$ is one-dimensional with respect to $\vec{\theta}$, when we approximate the log posterior at time t using our gaussian approximation, $p(\vec{\theta} \mid \vec{\mu}_{t-1}, \boldsymbol{C}_{t-1})$, we need to do only a one-dimensional search to find the peak of the log posterior at time t. The gray and black dots in the figure illustrate the location of $\vec{\mu}_{t-1}$ and $\vec{\mu}_t$, respectively.

3 Representing and Updating the Posterior

Our first computational challenge is representing and updating the posterior distribution on the parameters, $p(\vec{\theta} \mid \boldsymbol{r}_{1:t}, \boldsymbol{s}_{1:t})$. We use a fast, sequential procedure for constructing a gaussian approximation of the posterior, (see Figure 3). This gaussian approximation leads to an update that is both efficient and accurate enough to be used online for picking optimal stimuli.

A gaussian approximation of the posterior is justified by the fact that the posterior is the product of two smooth, log-concave terms—the GLM likelihood function and the prior (which we assume to be gaussian, for simplicity). As a result, the log posterior is concave (i.e., it always curves downward) and can be well approximated by the quadratic expression for the log of a gaussian. Furthermore, the main result of Paninski (2005) is a central limit-like theorem for optimal experiments based on maximizing the mutual information. This theorem guarantees that asymptotically, the gaussian approximation of the posterior will be accurate.

We recursively construct a gaussian approximation to the posterior by first approximating the posterior using our posterior from the previous trial (see Figure 3). Since the gaussian approximation of the posterior at time $t-1$, $p(\vec{\theta} \mid \vec{\mu}_{t-1}, \boldsymbol{C}_{t-1})$, summarizes the information in the first $t-1$ trials, we can use this distribution to approximate the log posterior after the

tth trial,

$$\log p(\vec{\theta} \mid \boldsymbol{s}_{1:t}, \boldsymbol{r}_{1:t}) = \underbrace{\log p(\vec{\theta}) + \sum_{i=1}^{t-1} \log p(r_i \mid \vec{s}_i, \vec{\theta})}_{} + \log p(r_t \mid \vec{s}_t, \vec{\theta}) + const. \tag{3.1}$$

$$\approx \log p(\vec{\theta} \mid \vec{\mu}_{t-1}, \boldsymbol{C}_{t-1}) + \log p(r_t \mid \vec{s}_t, \vec{\theta}) + const. \tag{3.2}$$

$$\approx \log p(\vec{\theta} \mid \vec{\mu}_t, \boldsymbol{C}_t) + const. \tag{3.3}$$

We fit the log of a gaussian to the approximation of the log posterior in equation 3.2, using the Laplace method (Berger, 1985; MacKay, 2003). This recursive approach is much faster, albeit slightly less accurate, then using the Laplace method to fit a gaussian distribution to the true posterior. The running time of this recursive update is $O(d^2)$, whereas fitting a gaussian distribution to the true posterior is $O(td^3)$. Since t and d are large, easily $O(10^3)$, the computational savings of the recursive approach are well worth the slight loss of accuracy. If the dimensionality is low, $d = O(10)$, we can measure the error by using Monte Carlo methods to compute the Kullback-Leibler distance between the true posterior and our gaussian approximation. This analysis (results not shown) reveals that the error is small and rapidly converges to zero.

The mean of our gaussian approximation is the peak of equation 3.2. The key to rapidly updating our posterior is that we can easily compute the direction in which the peak of equation 3.2 lies relative to $\vec{\mu}_{t-1}$. Once we know the direction in which $\vec{\mu}_t$ lies, we just need to perform a one-dimensional search to find the actual peak. To compute the direction of $\vec{\mu}_t - \vec{\mu}_{t-1}$, we write out the gradient of equation 3.2,

$$\frac{d \log p(\vec{\theta} \mid \boldsymbol{r}_{1:t}, \boldsymbol{s}_{1:t})}{d\vec{\theta}} \approx \frac{\partial \log p(\vec{\theta} \mid \vec{\mu}_{t-1}, \boldsymbol{C}_{t-1})}{\partial \vec{\theta}} + \frac{\partial \log p(r_t \mid \vec{s}_t, \vec{\theta})}{\partial \vec{\theta}} \tag{3.4}$$

$$= -(\vec{\theta} - \vec{\mu}_{t-1})^T \boldsymbol{C}_{t-1}^{-1} + \left(\frac{r_t}{f(\rho_t)} - dt \right) \frac{df}{d\rho}\bigg|_{\rho_t} \vec{s}_t^T. \tag{3.5}$$

At the peak of the log posterior, the gradient equals zero, which means the first term in equation 3.5 must be parallel to $\vec{s}_t$. Since $\boldsymbol{C}_{t-1}$ is nonsingular, $\vec{\mu}_t - \vec{\mu}_{t-1}$ must be parallel to $\boldsymbol{C}_{t-1}\vec{s}_t$,

$$\vec{\mu}_t = \vec{\mu}_{t-1} + \Delta_t \boldsymbol{C}_{t-1} \vec{s}_t. \tag{3.6}$$

Δ_t is a scalar that measures the magnitude of the difference, $\vec{\mu}_t - \vec{\mu}_{t-1}$. We find Δ_t by solving the following one-dimensional equation using Newton's

method:

$$-\Delta_t + \left(\frac{r_t}{f(\rho_t)} - dt\right)\frac{df}{d\rho}\bigg|_{\rho_t=\vec{s}_t^T\vec{\mu}_{t-1}+\Delta_t\vec{s}_t^T\boldsymbol{C}_{t-1}\vec{s}_t} = 0. \tag{3.7}$$

This equation defines the location of the peak of the log posterior in the direction $\boldsymbol{C}_{t-1}\vec{s}_t$. Since the log posterior is concave, equation 3.7 is the solution to a one-dimensional concave optimization problem. Equation 3.7 is therefore guaranteed to have a single, unique solution. Solving this one-dimensional problem involves a single matrix-vector multiplication that requires $O(d^2)$ time.

Having found $\vec{\mu}_t$, we estimate the covariance matrix $\boldsymbol{C}_t$ of the posterior by forming the Taylor approximation of equation 3.2 about $\vec{\mu}_t$:

$$\log p(\vec{\theta} \mid \boldsymbol{r_{1:t}}, \boldsymbol{s_{1:t}}) \approx -\frac{1}{2}(\vec{\theta} - \vec{\mu}_t)^T \boldsymbol{C}_t^{-1}(\vec{\theta} - \vec{\mu}_t) + const. \tag{3.8}$$

$$-\boldsymbol{C}_t^{-1} = \frac{\partial^2 \log p(\vec{\theta} \mid \vec{\mu}_t, \boldsymbol{C}_t)}{d\vec{\theta}^2} \tag{3.9}$$

$$= \frac{\partial^2 \log p(\vec{\theta} \mid \vec{\mu}_{t-1}, \boldsymbol{C}_{t-1})}{\partial\vec{\theta}^2} + \frac{\partial^2 \log p(r_t \mid \vec{s}_t, \vec{\theta})}{\partial\vec{\theta}^2}. \tag{3.10}$$

The Laplace method uses the curvature of the log posterior as an estimate of the inverse covariance matrix. The larger the curvature, the more certain we are that our estimate $\vec{\mu}_t$ is close to the true parameters. The curvature, as measured by the second derivative, is the sum of two terms, equation 3.10. The first term approximates the information provided by the first $t-1$ observations. The second term measures the information in our latest observation, r_t. The second term is proportional to the Fisher information. By definition, the Fisher information is the negative of the second derivative of the log likelihood (Berger, 1985). The second derivative of the log likelihood provides an intuitive metric for the informativeness of an observation because a larger second derivative means that small differences in $\vec{\theta}$ produce large deviations in the responses. Hence, a large Fisher information means we can infer the parameters with more confidence.

To compute the Hessian, the matrix of partial second derivatives, of the log posterior, we need to sum only two matrices: $\boldsymbol{C}_{t-1}^{-1}$ and the Hessian of $\log p(r_t \mid \vec{s}_t, \vec{\theta})$. The Hessian of the log likelihood is a rank 1 matrix. We can therefore efficiently invert the Hessian of the updated log posterior in $O(d^2)$ time using the Woodbury matrix lemma (Henderson & Searle, 1981; Seeger, 2007). Evaluating the derivatives in equation 3.10 and using the Woodbury

lemma yields

$$\boldsymbol{C}_t = \left(\boldsymbol{C}_{t-1}^{-1} - \frac{\partial^2 \log p(r_t \mid \rho_t)}{\partial \rho^2} \vec{s}_t \vec{s}_t^T \right)^{-1} \tag{3.11}$$

$$= \boldsymbol{C}_{t-1} - \frac{\boldsymbol{C}_{t-1} \vec{s}_t D(r_t, \rho_t) \vec{s}_t^T \boldsymbol{C}_{t-1}}{1 + D(r_t, \rho_t) \vec{s}_t^T \boldsymbol{C}_{t-1} \vec{s}_t} \tag{3.12}$$

$$D(r_t, \rho_t) = -\frac{\partial^2 \log p(r_t \mid \rho)}{\partial \rho^2}\bigg|_{\rho_t}$$

$$= -\left(\frac{r_t}{f(\rho_t)} - dt \right) \frac{d^2 f}{d\rho^2}\bigg|_{\rho_t} + \frac{r_t}{(f(\rho_t))^2} \left(\frac{df}{d\rho}\bigg|_{\rho_t} \right)^2 \tag{3.13}$$

$$\rho_t = \vec{\theta}^T \vec{s}_t. \tag{3.14}$$

$D(r_t, \rho_t)$ is the one-dimensional Fisher information—the negative of the second derivative of the log likelihood with respect to ρ_t. In this equation, ρ_t depends on the unknown parameters, $\vec{\theta}$, because we would like to compute the Fisher information for the true parameters. That is, we would like to expand our approximation of the log posterior about $\vec{\theta}$. Since $\vec{\theta}$ is unknown, we use the approximation

$$\rho_t \approx \vec{\mu}_t^T \vec{s}_t \tag{3.15}$$

to compute the new covariance matrix. Since computing the covariance matrix is just a rank one update, computing the updated gaussian approximation requires only $O(d^2)$ computations. A slower but potentially more accurate update for small t would be to construct our gaussian by matching the first and second moments of the true posterior distribution using the expectation propagation algorithm (Minka, 2001; Seeger, Gerwinn, & Bethge, 2007).

Asymptotically under suitable regularity conditions, the mean of our gaussian is guaranteed to converge to the true $\vec{\theta}$. Consistency can be established by applying theorems for the consistency of estimators based on stochastic gradient descent (Fabian, 1978; Sharia, 2007). We used numerical simulations (data not shown) to verify the predictions of these theorems. To apply these theorems to our update, we must be able to restrict $\vec{\theta}$ to a closed and bounded space. Since all $\vec{\theta}$ corresponding to neural models would naturally be bounded, this constraint is satisfied for all biologically reasonable GLMs.

Our update uses the Woodbury lemma, which is unstable when $\boldsymbol{C}_t$ is close to being singular. When optimizing under a power constraint (see section 5.2), we can avoid using the Woodbury lemma by computing the eigendecomposition of the covariance matrix. Since we need to compute

the eigendecomposition in order to optimize the stimulus, no additional computation is required in this case. When the eigendecomposition was not needed for optimization, we usually found that the Woodbury lemma was sufficiently stable. However, a more stable solution in this case would have been to compute and maintain the Cholesky decomposition of the covariance matrix (Seeger, Steinke, & Tsuda, 2007).

4 Computing the Mutual Information

A rigorous Bayesian approach to sequential optimal experimental design is to pick the stimulus that maximizes the expected value of a utility function (Bernardo, 1979). Common functions are the mean squared error of the model's predictions (Fedorov, 1972; Cohn, Ghahramani, & Jordan, 1996; Schein, 2005), the entropy of the responses (Bates, Buck, Riccomagno, & Wynn, 1996), and the expected information gain (Lindley, 1956; Bernardo, 1979; MacKay, 1992; Chaloner & Verdinelli, 1995). A number of different quantities can be used to measure the expected information depending on whether the goal is prediction or inference. We are primarily interested in estimating the unknown parameters, so we measure expected information using the mutual information between $\vec{\theta}$ and the data $(\vec{s}_t, r_t)$. The mutual information measures the expected reduction in the number of models consistent with the data. Choosing the optimal design requires maximizing the mutual information, $I(\{\vec{s}_{t+1}, r_{t+1}\}; \vec{\theta} \mid \boldsymbol{s_{1:t}}, \boldsymbol{r_{1:t}})$, conditioned on the data already collected as a function of the design $p(\vec{x}_{t+1})$,

$$p_{opt}(\vec{x}_{t+1}) = \arg\max_{p(\vec{x}_{t+1})} I(\{\vec{s}_{t+1}, r_{t+1}\}; \vec{\theta} \mid \boldsymbol{s_{1:t}}, \boldsymbol{r_{1:t}}). \tag{4.1}$$

We condition the mutual information on the data already collected because we want to maximize the information given what we have already learned about $\vec{\theta}$.

Before diving into a detailed mathematical computation, we want to provide a less technical explanation of our approach. Before we conduct any trials, we have a set, Θ, of possible models. For any stimulus, each model in Θ makes a prediction of the response. To identify the best model, we should pick a stimulus that maximizes the disagreement between the predictions of the different models. In theory, we could measure the disagreement for any stimulus by computing the predicted response for each model. However, since the number of possible models is large, explicitly computing the response for each model is rarely possible.

We can compute the mutual information efficiently because once we pick a stimulus, we partition the model space, Θ, into equivalent sets with respect to the predicted response. Once we fix $\vec{s}_{t+1}$, the likelihood of the responses varies only with the projection $\rho_{t+1} = \vec{s}_{t+1}^T \vec{\theta}$. Hence, all models with the same value for ρ_{t+1} make the same prediction. Therefore, instead

of computing the disagreement among all models in Θ space, we only have to compute the disagreement between the models in these different subspaces; that is, at most, we have to determine the response for one model in each of the subspaces defined by $\rho_{t+1} = const.$

Of course, the mutual information also depends on what we already know about the fitness of the different models. Since our experiment provides no information about $\vec{\theta}$ in directions orthogonal to $\vec{s}_{t+1}$, our uncertainty in these directions will be unchanged. Therefore, the mutual information will only depend on the information we have about $\vec{\theta}$ in the direction $\vec{s}_{t+1}$; that is, it depends only on $p(\rho_{t+1} \mid \vec{s}_{t+1}, \vec{\mu}_t, \boldsymbol{C}_t)$ instead of our full posterior $p(\vec{\theta} \mid \vec{s}_{t+1}, \vec{\mu}_t, \boldsymbol{C}_t)$.

Furthermore, we have to evaluate the mutual information only for nonrandom designs because any optimal design $p_{opt}(\vec{x}_{t+1})$ must place all of its mass on the stimulus, $\vec{x}_{t+1}$, which maximizes the conditional mutual information $I(r_{t+1}; \vec{\theta} \mid \vec{s}_{t+1}, \boldsymbol{s_{1:t}}, \boldsymbol{r_{1:t}})$ (MacKay, 1992; Paninski, 2005). This property means we can focus on the simpler problem of efficiently evaluating $I(r_{t+1}; \vec{\theta} \mid \vec{s}_{t+1}, \boldsymbol{s_{1:t}}, \boldsymbol{r_{1:t}})$ as a function of the input $\vec{s}_{t+1}$.

The mutual information measures the reduction in our uncertainty about the parameters $\vec{\theta}$, as measured by the entropy,

$$\begin{aligned} &I(\vec{\theta}; r_{t+1} \mid \vec{s}_{t+1}, \boldsymbol{s_{1:t}}, \boldsymbol{r_{1:t}}) \\ &\quad = H(p(\vec{\theta} \mid \boldsymbol{s_{1:t}}, \boldsymbol{r_{1:t}})) - E_{\vec{\theta} \mid \vec{\mu}_t, \boldsymbol{C}_t} E_{r_{t+1} \mid \vec{\theta}, \vec{s}_{t+1}} H(p(\vec{\theta} \mid \boldsymbol{s_{1:t+1}}, \boldsymbol{r_{1:t+1}})). \end{aligned} \tag{4.2}$$

The first term, $H(p(\vec{\theta} \mid \boldsymbol{s_{1:t}}, \boldsymbol{r_{1:t}}))$, measures our uncertainty at time t. Since $H(p(\vec{\theta} \mid \boldsymbol{s_{1:t}}, \boldsymbol{r_{1:t}}))$ is independent of $\vec{s}_{t+1}$, we just need to minimize the second term, which measures how uncertain about $\vec{\theta}$ we expect to be after the next trial. Our uncertainty at time $t+1$ depends on the response to the stimulus. Since r_{t+1} is unknown, we compute the expected entropy of the posterior, $p(\vec{\theta} \mid \boldsymbol{s_{1:t+1}}, \boldsymbol{r_{1:t+1}})$, as a function of r_{t+1} and then take the average over r_{t+1} using our GLM to compute the likelihood of each r_{t+1} (MacKay, 1992; Chaloner & Verdinelli, 1995). Since the likelihood of r_{t+1} depends on the unknown model parameters, we also need to take an expectation over $\vec{\theta}$. To evaluate the probability of the different $\vec{\theta}$, we use our current posterior, $p(\vec{\theta} \mid \vec{\mu}_t, \boldsymbol{C}_t)$.

We compute the posterior entropy, $H(p(\vec{\theta} \mid \boldsymbol{s_{1:t+1}}, \boldsymbol{r_{1:t+1}}))$, as a function of r_{t+1} by first approximating $p(\vec{\theta} \mid r_{t+1}, \vec{s}_{t+1})$ as gaussian. The entropy of a gaussian is easy to compute (Cover & Thomas, 1991):

$$H(p(\vec{\theta} \mid \boldsymbol{s_{1:t+1}}, \boldsymbol{r_{1:t+1}})) \approx H(p(\vec{\theta} \mid \vec{\mu}_{t+1}, \boldsymbol{C}_{t+1})) \tag{4.3}$$

$$= \frac{1}{2} \log \mid \boldsymbol{C}_{t+1} \mid + const. \tag{4.4}$$

According to our update rule,

$$\boldsymbol{C}_{t+1} = \boldsymbol{C}_t - \frac{\boldsymbol{C}_t \vec{s}_{t+1} D(r_{t+1}, \rho_{t+1}) \vec{s}_{t+1}^T \boldsymbol{C}_t}{1 + D(r_{t+1}, \rho_{t+1}) \vec{s}_{t+1}^T \boldsymbol{C}_t \vec{s}_{t+1}} \tag{4.5}$$

$$\rho_{t+1} = \vec{\theta}^T \vec{s}_{t+1}. \tag{4.6}$$

As discussed in the previous section, the Fisher information depends on the unknown parameters. To compute the entropy, we treat the Fisher information,

$$J_{obs}(r_{t+1}, \vec{s}_{t+1}, \vec{\theta}) = -\frac{\partial^2 \log p(r_{t+1} \mid \rho_{t+1})}{\partial \rho^2} \vec{s}_{t+1} \vec{s}_{t+1}^T, \tag{4.7}$$

as a random variable since it is a function of $\vec{\theta}$. We then estimate our expected uncertainty as the expectation of $H(p(\vec{\theta} \mid \vec{\mu}_{t+1}, \boldsymbol{C}_{t+1}))$ with respect to $\vec{\theta}$ using the posterior at time t. The mutual information, equation 4.2, already entails computing an average over $\vec{\theta}$, so we do not need to introduce another integration.

This Bayesian approach to estimating the expected posterior entropy differs from the approach used to update our gaussian approximation of the posterior. To update the posterior at time t, we use the point estimate $\vec{\theta} \approx \vec{\mu}_t$ to estimate the Fisher information of the observation at time t. We could apply the same principle to compute the expected posterior entropy by using the approximation

$$\rho_{t+1} \approx \vec{\mu}_{t+1}^T \vec{s}_{t+1}, \tag{4.8}$$

where $\vec{\mu}_{t+1}$ is computed using equations 3.6 and 3.7. Using this approximation of ρ_{t+1} is intractable because we would need to solve for $\vec{\mu}_{t+1}$ numerically for each value of r_{t+1}. We could solve this problem by using the point approximation $\rho_{t+1} \approx \vec{\mu}_t^T \vec{s}_{t+1}$, which we can easily compute since $\vec{\mu}_t$ is known (MacKay, 1992; Chaudhuri & Mykland, 1993; Cohn, 1994). This point approximation means we estimate the Fisher information for each possible $(r_{t+1}, \vec{s}_{t+1})$ using the assumption that $\vec{\theta} \approx \vec{\mu}_t$. Unless $\vec{\mu}_t$ happens to be close to $\vec{\theta}$, there is no reason that the Fisher information computed assuming $\vec{\theta} \approx \vec{\mu}_t$ should be close to the Fisher information evaluated at the true parameters. In particular, at the start of an experiment when $\vec{\mu}_t$ is highly inaccurate, we would expect this point approximation to lead to poor estimates of the Fisher information. Similarly, we would expect this point approximation to fail for time-varying systems as the posterior covariance may no longer converge to zero asymptotically (see section 6.2). In contrast to using a point approximation, our approach of averaging the Fisher information with respect to $\vec{\theta}$ should provide much better estimates of the Fisher information when our uncertainty about $\vec{\theta}$ is high or when $\vec{\theta}$ is changing (Lindley, 1956; Chaloner & Verdinelli, 1995). Averaging the

expected information of $\vec{s}_{t+1}$ with respect to our posterior leads to an objective function, which takes into account all possible models. In particular, it means we favor inputs that are informative under all models with high probability as opposed to inputs that are informative only if $\vec{\theta} = \vec{\mu}_t$.

To compute the mutual information, equation 4.2, we need to evaluate a high-dimensional expectation over the joint distribution on $(\vec{\theta}, r_t)$. Evaluating this expectation is tractable because we approximate the posterior as a gaussian distribution and the log likelihood is one-dimensional. The one-dimensionality of the log likelihood means $\boldsymbol{C}_{t+1}$ is a rank 1 update of $\boldsymbol{C}_t$. Hence, we can use the the identity $\mid I + \vec{w}\vec{z}^T \mid = 1 + \vec{w}^T\vec{z}$ to evaluate the entropy at time $t+1$,

$$|\boldsymbol{C}_{t+1}| = |\boldsymbol{C}_t| \left| I - \frac{\vec{s}_{t+1} D(r_{t+1}, \rho_{t+1}) \vec{s}_{t+1}^T \boldsymbol{C}_t}{1 + D(r_{t+1}, \rho_{t+1}) \vec{s}_{t+1}^T \boldsymbol{C}_t \vec{s}_{t+1}} \right| \tag{4.9}$$

$$= |\boldsymbol{C}_t| \cdot \left(1 + D(r_{t+1}, \rho_{t+1}) \sigma_\rho^2\right)^{-1} \tag{4.10}$$

$$\sigma_\rho^2 = \vec{s}_{t+1}^T \boldsymbol{C}_t \vec{s}_{t+1}$$

$$\rho_{t+1} = \vec{s}_{t+1}^T \vec{\theta}.$$

Consequently,

$$E_{\vec{\theta} \mid \vec{\mu}_t, \boldsymbol{C}_t} E_{r_{t+1} \mid \vec{s}_{t+1}, \vec{\theta}} H(p(\vec{\theta} \mid \vec{\mu}_{t+1}, \boldsymbol{C}_{t+1})) \tag{4.11}$$

$$= -\frac{1}{2} E_{\vec{\theta} \mid \vec{\mu}_t, \boldsymbol{C}_t} E_{r_{t+1} \mid \vec{s}_{t+1}, \vec{\theta}} \log\left(1 + D(r_{t+1}, \rho_{t+1}) \sigma_\rho^2\right) + const. \tag{4.12}$$

We can evaluate equation 4.12 without doing any high-dimensional integration because the likelihood of the responses only depends on $\rho_{t+1} = \vec{s}_{t+1}^T \vec{\theta}$. As a result,

$$-\frac{1}{2} E_{\vec{\theta} \mid \vec{\mu}_{t+1}, \boldsymbol{C}_{t+1}} E_{r_{t+1} \mid \vec{\theta}, \vec{s}_{t+1}} \log\left(1 + D(r_{t+1}, \rho_{t+1}) \sigma_\rho^2\right)$$

$$= -\frac{1}{2} E_{\rho_{t+1} \mid \vec{\mu}_{t+1}, \boldsymbol{C}_{t+1}} E_{r_{t+1} \mid \rho_{t+1}} \log\left(1 + D(r_{t+1}, \rho_{t+1}) \sigma_\rho^2\right). \tag{4.13}$$

Since $\rho_{t+1} = \vec{\theta}^T \vec{s}_{t+1}^T$ and $p(\vec{\theta} \mid \vec{\mu}_t, \boldsymbol{C}_t)$ is gaussian, ρ_{t+1} is a one-dimensional gaussian variable with mean $\mu_\rho = \vec{\mu}_t^T \vec{s}_{t+1}$ and variance $\sigma_\rho^2 = \vec{s}_{t+1}^T \boldsymbol{C}_t \vec{s}_{t+1}$. The final result is a very simple, two-dimensional expression for our objective function,

$$I(r_{t+1}; \vec{\theta} \mid \vec{s}_{t+1}, \boldsymbol{s_{1:t}}, \boldsymbol{r_{1:t}})$$

$$\approx E_{\rho_{t+1} \mid \mu_\rho, \sigma_\rho^2} E_{r_{t+1} \mid \rho_{t+1}} \log\left(1 + D(r_{t+1}, \rho_{t+1}) \sigma_\rho^2\right) + const$$

$$\mu_\rho = \vec{\mu}_t^T \vec{s}_{t+1} \quad \sigma_\rho^2 = \vec{s}_{t+1}^T \boldsymbol{C}_t \vec{s}_{t+1}. \tag{4.14}$$

The right-hand side of equation 4.14 is an approximation of the mutual information because the posterior is not in fact gaussian.

Equation 4.14 is a fairly intuitive metric for rating the informativeness of different designs. To distinguish among models, we want the response to be sensitive to $\vec{\theta}$. The information increases with the sensitivity because as the sensitivity increases, small differences in $\vec{\theta}$ produce larger differences in the response, making it easier to identify the correct model. The information, however, also depends on the variability of the responses. As the variability of the responses increases, the information decreases because it is harder to determine which model is more accurate. The Fisher information, $D(r_{t+1}, \rho_{t+1})$, takes into account both the sensitivity and the variability. As the sensitivity increases, the second derivative of the log likelihood increases because the peak of the log likelihood becomes sharper. Conversely, as the variability increases, the log likelihood becomes flatter, and the Fisher information decreases. Hence, $D(r_{t+1}, \rho_{t+1})$ measures the informativeness of a particular response. However, information is valuable only if it tells us something we do not already know. In our objective function, σ_ρ^2 measures our uncertainty about the model. Since our objective function depends on the product of the Fisher information and our uncertainty, our algorithm will favor experiments providing large amounts of new information.

In equation 4.14 we have reduced the mutual information to a two-dimensional integration over ρ_{t+1} and r_{t+1}, which depends on $(\mu_\rho, \sigma_\rho^2)$. While 2D numerical integration is quite tractable, it could potentially be too slow for real-time applications. A simple solution is to precompute this function before training begins on a suitable 2D region of $(\mu_\rho, \sigma_\rho^2)$ and then use a lookup table during our experiments.

In certain special cases, we can further simplify the expectations in equation 4.14, making numerical integration unnecessary. One simplification is to use the standard linear approximation $\log(1 + x) = x + o(x)$ when $D(r_{t+1}, \rho_{t+1})\sigma_\rho^2$ is sufficiently small. Using this linear approximation, we can simplify equation 4.14 to

$$\begin{aligned} &E_{\rho_{t+1}\,|\,\mu_\rho,\sigma_\rho^2}\, E_{r_{t+1}\,|\,\rho_{t+1}}\, \log\left(1 + D(r_{t+1}, \rho_{t+1})\sigma_\rho^2\right) \\ &\quad \approx E_{\rho_{t+1}\,|\,\mu_\rho,\sigma_\rho^2}\, E_{r_{t+1}\,|\,\rho_{t+1}} D(r_{t+1}, \rho_{t+1})\sigma_\rho^2, \end{aligned} \tag{4.15}$$

which may be evaluated analytically in some special cases (see below). If $\vec{\theta}$ is constant, then this approximation is always justified asymptotically because the variance in all directions asymptotically converges to zero (see section 7). Consequently, $\sigma_\rho^2 \to 0$ as $t \to \infty$. Therefore, if $D(r_{t+1}, \rho_{t+1})$ is bounded, then asymptotically $D(r_{t+1}, \rho_{t+1})\sigma_\rho^2 \to 0$.

4.1 Special Case: Exponential Nonlinearity. When the nonlinear function $f()$ is the exponential function, we can derive an analytical approximation for the mutual information, equation 4.14, because the Fisher

information is independent of the observation. This special case is worth considering because the exponential nonlinearity has proved adequate for modeling several types of neurons in the visual system (Chichilnisky, 2001; Pillow, Paninski, Uzzell, Simoncelli, & Chichilnisky, 2005; Rust, Mante, Simoncelli, & Movshon, 2006). As noted in the previous section, the Fisher information depends on the variability and sensitivity of the responses to the model parameters. In general, the Fisher information depends on the response because we can use it to estimate the variability and sensitivity of the neuron's responses. For the Poisson model with convex and increasing $f()$,[1] a larger response indicates more variability but also more sensitivity of the response to ρ_{t+1}. For the exponential nonlinearity, the decrease in information due to increased variability and the increase in information due to increased sensitivity with the response cancel out, making the Fisher information independent of the response. Mathematically this means the second derivative of the log likelihood with respect to $\vec{\theta}$ is independent of r_{t+1},

$$D(r_{t+1}, \rho_{t+1}) = \exp(\rho_{t+1}). \tag{4.16}$$

By eliminating the expectation over r_{t+1} and using the linear approximation $\log(1 + x) = x + o(x)$, we can simplify equation 4.14:

$$\begin{aligned} & E_{\rho_{t+1} \mid \mu_\rho, \sigma_\rho^2} E_{r_{t+1} \mid \rho_{t+1}} \log\left(1 + D(r_{t+1}, \rho_{t+1})\sigma_\rho^2\right) \\ & = E_{\rho_{t+1} \mid \mu_\rho, \sigma_\rho^2} \log\left(1 + \exp(\rho_{t+1})\sigma_\rho^2\right) + const. && (4.17) \\ & = E_{\rho_{t+1}, \mu_\rho, \sigma_\rho^2} \log\left(1 + \exp(\rho_{t+1})\sigma_\rho^2\right) && (4.18) \\ & \approx E_{\rho_{t+1} \mid \mu_\rho, \sigma_\rho^2} \exp(\rho)\sigma_\rho^2. && (4.19) \end{aligned}$$

We can use the moment-generating function of a gaussian distribution to evaluate this expectation over ρ_{t+1}:

$$E_{\rho_{t+1} \mid \mu_\rho, \sigma_\rho^2} \exp(\rho_{t+1})\sigma_\rho^2 = \sigma_\rho^2 \exp\left(\mu_\rho + \frac{1}{2}\sigma_\rho^2\right). \tag{4.20}$$

Our objective function is increasing with μ_ρ and σ_ρ^2. In section 5.2, we show that this property makes optimizing the design for an exponential nonlinearity particularly tractable.

4.2 Linear Model. The optimal design for minimizing the posterior entropy of $\vec{\theta}$ for the standard linear model is a well-known result in the statistics and experimental design literature (MacKay, 1992; Chaloner &

[1] Recall that we can take $f()$ to be increasing without loss of generality.

Verdinelli, 1995). It is enlightening to rederive these results using the methods we have introduced so far and to point out some special features of the standard linear case.

The linear model is

$$r_t = \vec{\theta}^T \vec{s}_t + \epsilon, \tag{4.21}$$

with ϵ a zero-mean gaussian random variable with variance σ^2. The linear model is a GLM with a gaussian distribution for the conditional distribution and a linear link function:

$$\log p(r_t \mid \vec{s}_t, \vec{\theta}, \sigma^2) = -\frac{1}{2\sigma^2}(r_t - \vec{\theta}^T \vec{s}_t)^2 + const \tag{4.22}$$

$$= -\frac{1}{2\sigma^2} r_t^2 + \frac{1}{\sigma^2} \rho_t r_t - \frac{1}{2\sigma^2} \rho_t^2 + const. \tag{4.23}$$

For the linear model, the variability, σ^2, is constant. Furthermore, the sensitivity of the responses to the input and the model parameters is also constant. Consequently, the Fisher information is independent of both the response and the input (Chaudhuri & Mykland, 1993). Mathematically this means that the observed Fisher information $D(r_{t+1}, \rho_{t+1})$ is a constant equal to the reciprocal of the variance:

$$D(r_{t+1}, \rho_{t+1}) = \frac{1}{\sigma^2}. \tag{4.24}$$

Plugging $D(r_{t+1}, \rho_{t+1})$ into equation 4.14, we obtain the simple result:

$$E_{\vec{\theta} \mid \vec{\mu}_t, \boldsymbol{C}_t} E_{r_{t+1} \mid \vec{\theta}, \vec{s}_{t+1}} I(r_{t+1}; \vec{\theta} \mid \vec{s}_{t+1}, \boldsymbol{s_{1:t}}, \boldsymbol{r_{1:t}}) = \log\left(1 + \frac{\sigma_\rho^2}{\sigma^2}\right) + const. \tag{4.25}$$

Since σ^2 is a constant, we can increase the mutual information only by picking stimuli for which $\sigma_\rho^2 = \vec{s}_{t+1}^T \boldsymbol{C}_t \vec{s}_{t+1}$ is maximized. Under the power constraint, σ_ρ^2 is maximized when all the stimulus energy is parallel to the maximum eigenvector of $\boldsymbol{C}_t$, the direction of maximum uncertainty. μ_ρ does not affect the optimization at all. This property distinguishes the linear model from the exponential Poisson case described above. Furthermore, the covariance matrix $\boldsymbol{C}_t$ is independent of past responses because the true posterior is gaussian with covariance matrix:

$$\boldsymbol{C}_t^{-1} = \boldsymbol{C}_0^{-1} + \sum_{i=1}^{t} \frac{1}{\sigma^2} \vec{s}_i \vec{s}_i^T. \tag{4.26}$$

Consequently, the optimal sampling strategy can be determined a priori, without having to observe r_t or to make any corresponding adjustments in our sampling strategy (MacKay, 1992).

Like the Poisson model with an exponential link function, the linear model's Fisher information is independent of the response. However, for the linear model, the Fisher information is also independent of the model parameters. Since the Fisher information is independent of the parameters, an adaptive design offers no benefit because we do not need to know the parameters to select the optimal input. In contrast, for the Poisson distribution with an exponential link function, the Fisher information depends on the parameters and the input, even though it is independent of the responses. As a result, we can improve our design by adapting it as our estimate of $\vec{\theta}$ improves.

5 Choosing the Optimal Stimulus

The simple expression for the conditional mutual information, equation 4.14, means that we can find the optimal stimulus by solving the following simple program:

1. $$\left(\mu_\rho, \sigma_\rho^2\right)^* = \underset{(\mu_\rho,\sigma_\rho^2)\in\mathcal{R}_{t+1}}{\operatorname{argmax}} \; E_{\rho_{t+1}\mid\mu_\rho,\sigma_\rho^2} E_{r_{t+1}\mid\rho_{t+1}} \times \log\left(1 + D(r_{t+1}, \rho_{t+1})\sigma_\rho^2\right) \tag{5.1}$$

$$\mathcal{R}_{t+1} = \left\{\left(\mu_\rho, \sigma_\rho^2\right) : \mu_\rho = \vec{\mu}_t^T \vec{s}_{t+1} \;\&\; \sigma_\rho^2 = \vec{s}_{t+1}^T \boldsymbol{C}_t \vec{s}_{t+1},\ \forall \vec{s}_{t+1} \in \mathcal{S}_{t+1}\right\} \tag{5.2}$$

$$\mathcal{S}_{t+1} = \left\{\vec{s}_{t+1} : \vec{s}_{t+1} = \left[\vec{x}_{t+1}^T, \vec{s}_{f,t+1}^T\right]^T, \vec{x}_{t+1} \in \mathcal{X}_{t+1}\right\}. \tag{5.3}$$

2. Find $\vec{s}_{t+1}$ s.t $\mu_\rho^* = \vec{\mu}_t^T \vec{s}_{t+1}$ $\sigma_\rho^{2*} = \vec{s}_{t+1}^T \boldsymbol{C}_t \vec{s}_{t+1}$. (5.4)

$\mathcal{R}_{t+1}$ is the range of the mapping $\vec{s}_{t+1} \to (\mu_\rho, \sigma_\rho^2)$ corresponding to the stimulus domain, $\mathcal{X}_{t+1}$. Once we have computed $\mathcal{R}_{t+1}$, we need to solve a highly tractable 2D optimization problem numerically. The final step is to map the optimal $(\mu_\rho, \sigma_\rho^2)$ back into the input space. In general, computing $\mathcal{R}_{t+1}$ for arbitrary stimulus domains is the hardest step.

We first present a general procedure for handling arbitrary stimulus domains. This procedure selects the optimal stimulus from a set, $\hat{\mathcal{X}}_{t+1}$, which is a subset of $\mathcal{X}_{t+1}$. $\hat{\mathcal{X}}_{t+1}$ contains a finite number of inputs; its size will be denoted $\mid \hat{\mathcal{X}}_{t+1} \mid$. Picking the optimal input in $\hat{\mathcal{X}}_{t+1}$ is easy. We simply compute $(\mu_\rho, \sigma_\rho^2)$ for each $\vec{x}_{t+1} \in \hat{\mathcal{X}}_{t+1}$.

Picking the optimal stimulus in a finite set, $\hat{\mathcal{X}}_{t+1}$, is flexible and straightforward. The informativeness of the resulting design, however, is highly dependent on how $\hat{\mathcal{X}}_{t+1}$ is constructed. In particular, we want to ensure that with high probability, $\hat{\mathcal{X}}_{t+1}$ contains inputs in $\mathcal{X}_{t+1}$ that are nearly optimal.

If we could compute $\mathcal{R}_{t+1}$, then we could avoid the problem of picking a good $\hat{\mathcal{X}}_{t+1}$. One case in which we can compute $\mathcal{R}_{t+1}$ is when $\mathcal{X}_{t+1}$ is defined by a power constraint; that is, $\mathcal{X}_{t+1}$ is a sphere. Since we can compute $\mathcal{R}_{t+1}$, we can optimize the input over its full domain. Unfortunately, our method for computing $\mathcal{R}_{t+1}$ cannot be applied to arbitrary input domains.

5.1 Optimizing over a Finite Set of Stimuli. Our first method simultaneously addresses two issues: how to deal with arbitrary stimulus domains and what to do if the stimulus domain is ill defined. In general, we expect that more efficient procedures for mapping a stimulus domain into $\mathcal{R}_{t+1}$ could be developed by taking into account the actual stimulus domain. However, a generalized procedure is needed because efficient algorithms for a particular stimulus domain may not exist, or their development may be complex and time-consuming. Furthermore, for many stimulus domains (i.e., natural images), we have many examples of the stimuli but no quantitative constraints that define the domain. An obvious solution to both problems is to simply choose the best stimulus from a subset of examples, $\hat{\mathcal{X}}_{t+1}$.

The challenge with this approach is picking the set $\hat{\mathcal{X}}_{t+1}$. For the optimization to be fast, $|\hat{\mathcal{X}}_{t+1}|$ needs to be sufficiently small. However, we also want to ensure that $|\hat{\mathcal{X}}_{t+1}|$ contains an optimal or nearly optimal input. In principle, this second criterion means $\hat{\mathcal{X}}_{t+1}$ should contain a large number of stimuli evenly dispersed over $\mathcal{X}_{t+1}$. We can in fact satisfy both requirements because the informativeness of a stimulus depends on only $(\mu_\rho, \sigma_\rho^2)$. Consequently, we can partition $\mathcal{X}_{t+1}$ into sets of equally informative experiments based on the value of $(\mu_\rho, \sigma_\rho^2)$. When constructing $\hat{\mathcal{X}}_{t+1}$, there is no reason to include more than one input for each value of $(\mu_\rho, \sigma_\rho^2)$ because all of these inputs are equally informative. Hence, to ensure that $\hat{\mathcal{X}}_{t+1}$ contains a nearly optimal input, we just need its stimuli to span the two-dimensional $\mathcal{R}_{t+1}$ and not the much higher-dimensional space, $\mathcal{X}_{t+1}$.

Although $\vec{\mu}_t$ and $\boldsymbol{C}_t$ change with time, these quantities are known when optimizing $\vec{s}_{t+1}$. Hence, the mapping $\mathcal{S}_{t+1} \to \mathcal{R}_{t+1}$ is known and easy to evaluate for any stimulus. We can use this knowledge to develop simple heuristics for selecting inputs that tend to be dispersed throughout $\mathcal{R}_{t+1}$. We delay until sections 5.3 and 6.1 the presentation of the heuristics that we used in our simulations so that we can first introduce the specific problems and stimulus domains for which these heuristics are suited.

5.2 Power Constraint. Ideally, we would like to optimize the input over its full domain as opposed to restricting ourselves to a subset of inputs. Here we present a method for computing $\mathcal{R}_{t+1}$ when $\mathcal{X}_{t+1}$ is defined by the power constraint $\|\vec{x}_{t+1}\|_2 \leq m$.[2] This is an important stimulus domain because of

[2]We apply the power constraint to $\vec{x}_{t+1}$, as opposed to the full input $\vec{s}_{t+1}$. However, the power constraint could just as easily have been applied to the full input.

its connection to white noise, which is often used to study sensory systems (Eggermont, 1993; Cottaris & De Valois, 1998; Chichilnisky, 2001; Dayan & Abbot, 2001; Wu, David, & Gallant, 2006). Under an i.i.d. design, the stimuli sampled from $\mathcal{X}_{t+1} = \{\vec{x}_{t+1}: ||\vec{x}_{t+1}||_2 \leq m\}$ resemble white noise. The primary difference is that we strictly enforce the power constraint, whereas for white noise, the power constraint applies only to the average power of the input. The domain $\mathcal{X}_{t+1} = \{\vec{x}_{t+1} : | \, |\vec{x}_{t+1}| \, |_2 \leq m\}$ is also worth considering because it defines a large space that includes many important subsets of stimuli such as random dot patterns (DiCarlo, Johnson, & Hsiao, 1998).

Our main result is a simple, efficient procedure for finding the boundary of $\mathcal{R}_{t+1}$ as a function of a 1D variable. Our procedure uses the fact that $\mathcal{R}_{t+1}$ is closed and connected. Furthermore, for fixed μ_ρ, σ_ρ^2 is continuous on the interval between its maximum and minimum values. These properties of $\mathcal{R}_{t+1}$ mean we can compute the boundary of $\mathcal{R}_{t+1}$ by maximizing and minimizing σ_ρ^2 as a function of μ_ρ. $\mathcal{R}_{t+1}$ consists of all points on this boundary as well as the points enclosed by this curve (Berkes & Wiskott, 2005):

$$\begin{aligned}\mathcal{R}_{t+1} = \big\{(\mu_\rho, \sigma_\rho^2) : \big(&- m||\vec{\mu}_{x,t}||_2 + \vec{s}_{f,t+1}^T \vec{\mu}_{f,t}\big) \leq \mu_\rho \\ &\leq \big(m||\vec{\mu}_{x,t}||_2 + \vec{s}_{f,t+1}^T \vec{\mu}_{f,t}\big), \\ &\sigma^2_{\rho,\min}(\mu_\rho) \leq \sigma_\rho^2 \leq \sigma^2_{\rho,\max}(\mu_\rho)\big\} \end{aligned} \tag{5.5}$$

$$\sigma^2_{\rho,\max}(\mu_\rho) = \max_{\vec{x}_{t+1}} \sigma_\rho^2 \quad s.t \quad \mu_\rho = \vec{\mu}_t^T \vec{s}_{t+1} \quad \& \quad ||\vec{x}_{t+1}||_2 \leq m \tag{5.6}$$

$$\sigma^2_{\rho,\min}(\mu_\rho) = \min_{\vec{x}_{t+1}} \sigma_\rho^2 \quad s.t \quad \mu_\rho = \vec{\mu}_t^T \vec{s}_{t+1} \quad \& \quad ||\vec{x}_{t+1}||_2 \leq m. \tag{5.7}$$

By solving equations 5.6 and 5.7, we can walk along the curves that define the upper and lower boundaries of $\mathcal{R}_{t+1}$ as a function of μ_ρ. To move along these curves, we simply adjust the value of the linear constraint. As we walk along these curves, the quadratic constraint ensures that we do not violate the power constraint that defines the stimulus domain.

We have devised a numerically stable and fast procedure for computing the boundary of $\mathcal{R}_{t+1}$. Our procedure uses linear algebraic manipulations to eliminate the linear constraints in equations 5.6 and 5.7. To eliminate the linear constraint, we derive an alternative quadratic expression for σ_ρ^2 in terms of $\vec{x}_{t+1}$,

$$\sigma_\rho^2 = \vec{x}_{t+1}^T A \vec{x}_{t+1} + \vec{b}(\alpha)^T \vec{x}_{t+1} + d(\alpha). \tag{5.8}$$

Here we discuss only the most important points regarding equation 5.8; the derivation and definition of the terms are provided in appendix A. The linear term of this modified quadratic expression ensures that the value of this expression is independent of the projection of $\vec{s}_{t+1}$ on $\vec{\mu}_{t+1}$. The constant

term ensures that the value of this expression equals the value of σ_ρ^2 if we forced the projection of $\vec{s}_{t+1}$ on $\vec{\mu}_t$ to μ_ρ. Maximizing and minimizing σ_ρ^2 subject to linear and quadratic constraints is therefore equivalent to maximizing and minimizing this modified quadratic expression with just the quadratic constraint.

To maximize and minimize equation 5.8 subject to the quadratic constraint $||\vec{x}_{t+1}||_2 \leq m$, we use the Karush-Kuhn-Tucker (KKT) conditions. For these optimization problems, it can be proved that the KKT are necessary and sufficient (Fortin, 2000). To compute the boundary of $\mathcal{R}_{t+1}$ as a function of μ_ρ, we need to solve the KKT for each value of μ_ρ. This approach is computationally expensive because for each value of μ_ρ, we need to find the value of the Lagrange multiplier by finding the root of a nonlinear function. We have devised a much faster solution based on computing μ_ρ as a function of the Lagrange multiplier; the details are in appendix A. This approach is faster because to compute μ_ρ as a function of the Lagrange multiplier, we need only find the root of a 1D quadratic expression.

To solve the KKT conditions, we need the eigendecomposition of A. Computing the eigendecomposition of A is the most expensive operation and, in the worst case, requires $O(d^3)$ operations. A, however, is a rank 2 perturbation of $\boldsymbol{C}_t$, equation A.11. When these perturbations are orthogonal to some of the eigenvectors of $\boldsymbol{C}_t$, we can reduce the number of computations needed to compute the eigendecomposition of $\boldsymbol{C}_t$ by using the Gu-Eisenstat algorithm (Gu & Eisenstat, 1994), as discussed in the next section. The key point is that we can on average compute the eigendecomposition in $O(d^2)$ time.

Having computed $\mathcal{R}_{t+1}$, we can perform a 2D search to find the pair $(\mu_\rho, \sigma_\rho^2)^*$, which maximizes the mutual information, thereby completing step 1 in our program. To finish the program, we need to find an input $\vec{s}_{t+1}$ such that $\vec{\mu}_t^T \vec{s}_{t+1} = \mu_\rho^*$ and $\vec{s}_{t+1}^T \boldsymbol{C}_t \vec{s}_{t+1} = \sigma_\rho^{2*}$. We can easily find one solution by solving a one-dimensional quadratic equation. Let $\vec{s}_{\min}$ and $\vec{s}_{\max}$ denote the inputs corresponding to $(\mu_\rho^*, \sigma_{\rho\,\min}^2)$ and $(\mu_\rho^*, \sigma_{\rho\,\max}^2)$, respectively. These inputs are automatically computed when we compute the boundary of $\mathcal{R}_{t+1}$. To find a suitable $\vec{s}_{t+1}$, we find a linear combination of these two vectors that yields σ_ρ^{2*}:

$$\text{find} \quad \gamma \quad \text{s.t} \quad \sigma_\rho^{2*} = \vec{s}_{t+1}(\gamma)^T \boldsymbol{C}_t \vec{s}_{t+1}(\gamma) \tag{5.9}$$

$$\vec{s}_{t+1}(\gamma) = (1-\gamma)\vec{s}_{\min}(\mu_\rho{}^*) + \gamma \vec{s}_{\max}(\mu_\rho{}^*) \quad \gamma \in [0,1]. \tag{5.10}$$

All $\vec{s}_{t+1}(\gamma)$ necessarily satisfy the power constraint because it defines a convex set, and $\vec{s}_{t+1}(\gamma)$ is a linear combination of two stimuli in this set. Similar reasoning guarantees that $\vec{s}_{t+1}(\gamma)$ has projection μ_ρ^* on $\vec{\mu}_t$. Although this $\vec{s}_{t+1}(\gamma)$ maximizes the mutual information with respect to the full stimulus domain under the power constraint, this solution may not be unique. Finding γ completes the optimization of the input under the power constraint.

In certain cases, we can reduce the two-dimensional search over $\mathcal{R}_{t+1}$ to an even simpler one-dimensional search. If the mutual information is monotonically increasing in σ_ρ^2, then we need to consider only $\sigma^2_{\rho,\max}(\mu_\rho)$ for each possible value of μ_ρ. Consequently, a one-dimensional search over $\sigma^2_{\rho,\max}(\mu_\rho)$ for $\mu_\rho \in [-m||\vec{\mu}_{x,t}||_2 + \vec{s}^T_{f,t+1}\vec{\mu}_{f,t}, m||\vec{\mu}_{x,t}||_2 + \vec{s}^T_{f,t+1}\vec{\mu}_{f,t}]$ is sufficient for finding the optimal input. A sufficient condition for guaranteeing that the mutual information increases with σ_ρ^2 is convexity of $E_{r_{t+1}\,|\,\rho_{t+1}} \log(1 + D(r_{t+1}, \rho_{t+1})\sigma_\rho^2)$ in ρ_{t+1} (see appendix B). An important example satisfying this condition is $f(\rho_{t+1}) = \exp(\rho_{t+1})$, which satisfies the convexity condition because

$$\frac{\partial^2 \log\left(1 + D(r_{t+1}, \rho_{t+1})\sigma_\rho^2\right)}{\partial \rho_{t+1}^2} = \frac{\exp(\rho_{t+1})\sigma_\rho^2}{\left(1 + \exp(\rho_{t+1})\sigma_\rho^2\right)^2} > 0. \tag{5.11}$$

5.3 Heuristics for the Power Constraint. Although we can compute $\mathcal{R}_{t+1}$ when $\mathcal{X}_{t+1} = \{\vec{x}_{t+1} : ||\vec{x}_{t+1}||_2 \le m\}$, efficient heuristics for picking subsets of stimuli are still worth considering. If the size of the subset of stimuli is small enough, then computing $(\mu_\rho, \sigma_\rho^2)$ for each stimulus in the subset is usually faster than computing $\mathcal{R}_{t+1}$ for the entire stimulus domain. Since we can set the size of the set to any positive integer, by decreasing the size of the set we can sacrifice accuracy, in terms of finding the optimal stimulus, for speed.

We developed a simple heuristic for constructing finite subsets of $\mathcal{X}_{t+1} = \{\vec{x}_{t+1} : ||\vec{x}_{t+1}||_2 \le m\}$ by taking linear combinations of the mean and maximum eigenvector. To construct a subset, $\hat{\mathcal{X}}_{ball,t+1}$, of the closed ball, we use the following procedure:

1. Generate a random number, ω, uniformly from the interval $[-m, m]$, where m^2 is the stimulus power.
2. Generate a random number, ϕ, uniformly from the interval $[-\sqrt{m^2 - \omega^2}, \sqrt{m^2 - \omega^2}]$.
3. Add the input $\vec{x}_{t+1} = \omega \frac{\vec{\mu}_{x,t}}{||\vec{\mu}_{x,t}||_2} + \phi \vec{g}_\perp$ to $\hat{\mathcal{X}}_{ball,t+1}$, where $\vec{g}_\perp = \frac{\vec{g}_{\max} - \frac{\vec{\mu}^T_{x,t}}{||\vec{\mu}_{x,t}||_2}\vec{g}_{\max}}{||\vec{g}_{\max} - \frac{\vec{\mu}^T_{x,t}}{||\vec{\mu}_{x,t}||_2}\vec{g}_{\max}||_2}$. $\vec{g}_{\max}$ is the maximum eigenvector of $\boldsymbol{C}_{x,t}$.

This procedure tends to produce a set of stimuli that are dispersed throughout $\mathcal{R}_{t+1}$. By varying the projection of $\vec{x}_{t+1}$ along the MAP, the heuristic tries to construct a set of stimuli for which the values of μ_ρ are uniformly distributed on the valid interval. Similarly, by varying the projection of each stimulus along the maximum eigenvector, we can adjust the value of σ_ρ^2 for each stimulus. Unfortunately, the subspace of the stimulus domain spanned by the mean and max eigenvector may not contain the stimuli that map to the boundaries of $\mathcal{R}_{t+1}$. Nonetheless, since this heuristic produces

stimuli that tend to be dispersed throughout $\mathcal{R}_{t+1}$, we can usually find a stimulus in $\hat{\mathcal{X}}_{ball,t+1}$ that is close to being optimal.

When the mutual information is increasing with σ_ρ^2, we can easily improve this heuristic. In this case, the optimal stimulus always lies on the sphere $\mathcal{X}_{t+1} = \{\vec{x}_{t+1} : ||\vec{x}_{t+1}||_2 = m\}$. Therefore, when constructing the stimuli in a finite set, we should pick only stimuli that are on this sphere. To construct such a subset, $\hat{\mathcal{X}}_{heur,t+1}$, we use the heuristic above except we set $\phi = \sqrt{m^2 - \omega^2}$. Since the mutual information for the exponential Poisson model is increasing with σ_ρ^2, our simulations for this model will always use $\hat{\mathcal{X}}_{heur,t+1}$ as opposed to $\hat{\mathcal{X}}_{ball,t+1}$.

We could also have constructed subsets of the stimulus domain, $\hat{\mathcal{X}}_{iid,t+1}$, by uniformly sampling the ball or sphere. Unfortunately, this process produces sets that rarely contain highly informative stimuli, particularly in high dimensions. Since the uniform distribution on the sphere is radially symmetric, $E_{\vec{x}_{t+1}}(\mu_\rho) = 0$ and the covariance matrix of $\vec{x}_{t+1}$ is diagonal with entries $\frac{E_{\vec{x}_{t+1}}(||\vec{x}_{t+1}||_2^2)}{d}$. As a result, the variance of μ_ρ, $||\vec{\mu}_t||_2^2 \frac{E_{\vec{x}_{t+1}}(||\vec{x}_{t+1}||_2^2)}{d}$ decreases as $1/d$, ensuring that for high-dimensional systems, the stimuli in $\hat{\mathcal{X}}_{iid,t+1}$ have μ_ρ close to zero with high probability (see Figure 4). Uniformly sampling the ball or sphere therefore does a poor job of selecting stimuli that are dispersed throughout $\mathcal{R}_{t+1}$. As a result, $\hat{\mathcal{X}}_{iid,t+1}$ is unlikely to contain stimuli close to being maximally informative.

5.4 Simulation Results. We tested our algorithm using computer simulations that roughly emulated typical neurophysiology experiments. The main conclusion of our simulations is that using our information-maximizing (infomax) design, we can reduce by an order of magnitude the number of trials needed to estimate $\vec{\theta}$ (Paninski, 2005). This means we can increase the complexity of neural models without having to increase the number of data points needed to estimate the parameters of these higher-dimensional models. Furthermore, our results show that we can perform the computations fast enough—between 10 m and 1 sec depending on $\dim(\vec{x}_{t+1})$—that our algorithm could be used online, during an experiment, without requiring expensive or custom hardware.

Our first simulation used our algorithm to learn the receptive field of a visually sensitive neuron. The simulation tested the performance of our algorithm with a high-dimensional input space. We took the neuron's receptive field to be a Gabor function as a proxy model of a V1 simple cell (Ringach, 2002). We generated synthetic responses by sampling equation 2.3 with $\vec{\theta}$ set to a 40×40 Gabor patch. The nonlinearity was the exponential function.

Plots of the posterior means (recall these are equivalent to the MAP estimate of $\vec{\theta}$) for several designs are shown in Figure 5. The results show that all infomax designs do better than an i.i.d. design, and an infomax design that optimizes over the full domain of the input, $\mathcal{X}_{t+1} = \{\vec{x}_{t+1} : ||\vec{x}_{t+1}||_2 = m\}$,

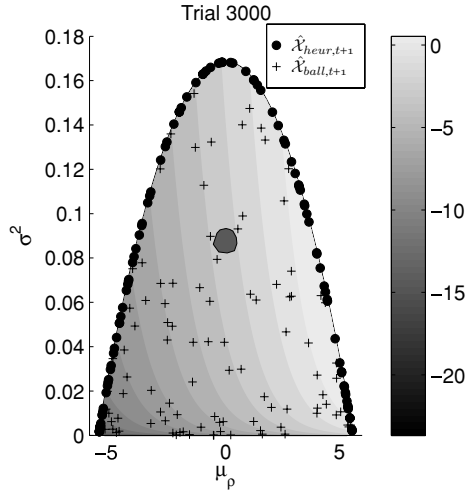

Figure 4: A plot showing $\mathcal{R}_{t+1}$, equation 5.5. The gray scale indicates the objective function, the log of equation 4.20. The dots and crosses show the points corresponding to the stimuli in $\hat{\mathcal{X}}_{heur,t+1}$ and $\hat{\mathcal{X}}_{ball,t+1}$ respectively. The dark gray region centered at $\mu_\rho = 0$ shows the region containing all stimuli in $\hat{\mathcal{X}}_{iid,t+1}$. To make the points easy to see, we kept the size of $\hat{\mathcal{X}}_{heur,t+1}$ and $\hat{\mathcal{X}}_{ball,t+1}$ small: $|\hat{\mathcal{X}}_{heur,t+1}| = |\hat{\mathcal{X}}_{heur,t+1}| = 100$, $|\hat{\mathcal{X}}_{iid,t+1}| = 10^4$. The points on the boundary corresponding to the largest and smallest values of μ_ρ correspond to stimuli that are parallel and antiparallel to $\vec{\mu}_t$. The posterior used to compute these quantities was the posterior after 3000 trials for the Gabor simulation described in the text. The posterior was taken from the design, which picked the optimal stimulus in $\mathcal{X}_{t+1}$ (i.e., $\vec{\mu}_t$ is the image shown in the first row and third column of Figure 5).

does much better than choosing the best stimulus in a subset constructed by uniformly sampling $\mathcal{X}_{t+1}$.

The results in Figures 5 and 6 show that if we choose the optimal stimulus from a finite set, then intelligently constructing the set is critical to achieving good performance. We compared two approaches for creating the set when $\mathcal{X}_{t+1} = \{\vec{x}_{t+1} : |\,|\vec{x}_{t+1}|\,|_2 = m\}$. The first approach selected a set of stimuli, $\hat{\mathcal{X}}_{iid,t+1}$, by uniformly sampling $\mathcal{X}_{t+1}$. The second approach constructed a set $\hat{\mathcal{X}}_{heur,t+1}$ for each trial using the heuristic presented in section 5.3. Picking the optimal stimulus in $\hat{\mathcal{X}}_{heur,t+1}$ produced much better estimates of $\vec{\theta}$ than picking the optimal stimulus in $\hat{\mathcal{X}}_{iid,t+1}$. In particular, the design using $\hat{\mathcal{X}}_{heur,t+1}$ converged to $\vec{\theta}$ nearly as fast as the design that optimized over the full stimulus domain, $\mathcal{X}_{t+1}$. These results show that using $\hat{\mathcal{X}}_{heur,t+1}$ is more efficient than reusing the same set of stimuli for all trials. To achieve comparable results using $\hat{\mathcal{X}}_{iid,t+1}$, we would have to increase the number of stimuli by several orders of magnitude. Consequently, the added cost of

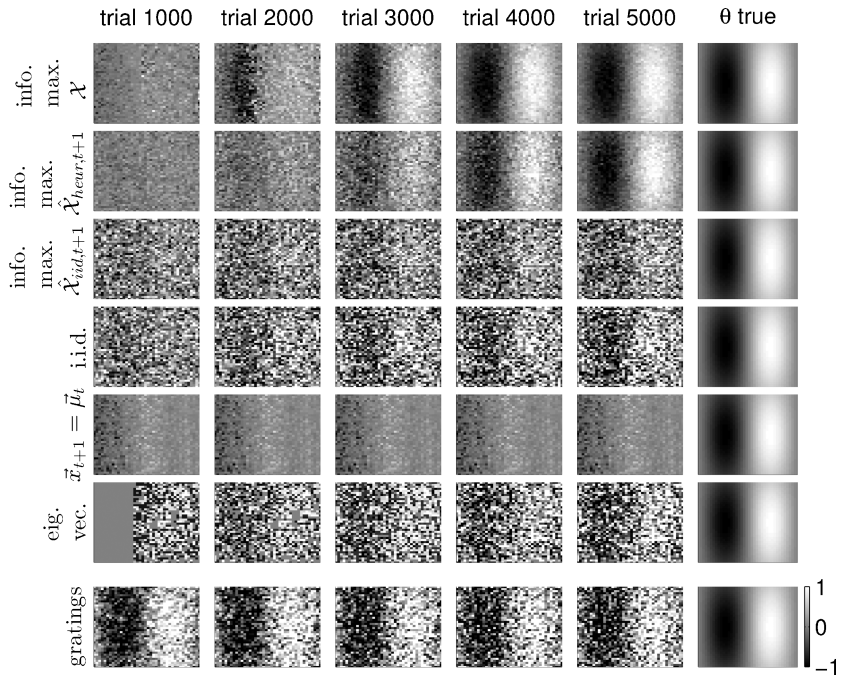

Figure 5: The receptive field, $\vec{\mu}_t$, of a simulated neuron estimated using different designs. The neuron's receptive field $\vec{\theta}$ was the 40×40 Gabor patch shown in the last column (spike history effects were set to zero for simplicity, $\vec{\theta}_f = 0$). The stimulus domain was defined by a power constraint $\mathcal{X}_{t+1} = \{\vec{x}_{t+1} : \|\vec{x}_{t+1}\|_2 = m\}$. The top three rows show the MAP if we pick the optimal stimulus in $\mathcal{X}_{t+1}$, $\hat{\mathcal{X}}_{heur,t+1}$, and $\hat{\mathcal{X}}_{iid,t+1}$ respectively. $\hat{\mathcal{X}}_{heur,t+1}$, and $\hat{\mathcal{X}}_{iid,t+1}$ contained 1000 stimuli. The final four rows show the results for an i.i.d. design, a design that set $\vec{x}_{t+1} = \vec{\mu}_t$, a design that set the stimulus to the maximum eigenvector of C_t, and a design that used sinusoidal gratings with random spatial frequency, orientation, and phase. Selecting the optimal stimulus in $\mathcal{X}_{t+1}$ or $\hat{\mathcal{X}}_{heur,t+1}$ leads to much better estimates of $\vec{\theta}$ using fewer stimuli than the other methods.

constructing a new stimulus set after each trial is more than offset by our ability to use fewer stimuli compared to using a constant set of stimuli.

We also compared the infomax designs to the limiting cases where we put all stimulus energy along the mean or maximum eigenvector (see Figures 5 and 6). Putting all energy along the maximum eigenvector performs nearly as well as an i.i.d. design. Our update, equation 3.12, ensures that if the stimulus is an eigenvector of C_t, the updated covariance matrix is the result of shrinking the eigenvalue corresponding to that eigenvector. Consequently, setting the stimulus to the maximum eigenvector ends up scanning through the different eigenvectors on successive trials. The resulting sequence of

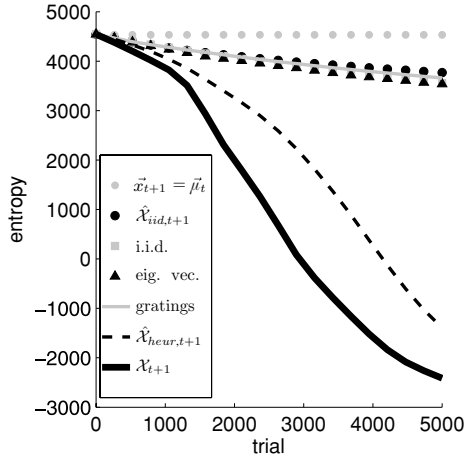

Figure 6: The posterior entropies for the simulations shown in Figure 5. Picking the optimal input from $\mathcal{X}_{t+1}$ decreases the entropy much faster than restricting ourselves to a subset of $\mathcal{X}_{t+1}$. However, if we pick a subset of stimuli using our heuristic, then we can decrease the entropy almost as fast as when we optimize over the full input domain. Note that the gray squares corresponding to the i.i.d. design are being obscured by the black triangles.

stimuli is statistically similar to that of an i.i.d. design because the stimuli are highly uncorrelated with each other and with $\vec{\theta}$. As a result, both methods generate similar marginal distributions $p(\vec{\theta}^T \vec{s}_{t+1})$ with sharp peaks at 0. Since the Fisher information of a stimulus under the power constraint varies only with $\rho_{t+1} = \vec{\theta}^T \vec{s}_{t+1}$, both methods pick stimuli that are roughly equally informative. Consequently, both designs end up shrinking the posterior entropy at similar rates.

In contrast, making the stimulus on each trial parallel to the mean leads to a much slower initial decrease of the posterior entropy. Since our initial guess of the mean is highly inaccurate, $\rho_{t+1} = \vec{\theta}^T \vec{s}_{t+1}$ is close to zero, resulting in a small value for the Fisher information. Furthermore, sequential stimuli end up being highly correlated. As a result, we converge very slowly to the true parameters.

We also evaluated a design that used sinusoidal gratings as the stimuli. In Figure 5, this design produces an estimate of $\vec{\theta}$ that already has the basic inhibitory and excitatory pattern of the receptive field after just 1000 trials. However, on the remaining trials $\vec{\mu}_t$ improves very little. Figure 6 shows that this design decreases the entropy at roughly the same rate as the i.i.d. design. The reason the coarse structure of the receptive field appears after so few trials is that the stimuli have a large amount of spatial correlation. This spatial correlation among the stimuli induces a similar correlation among

the components of the MAP and explains why the coarse inhibitory and excitatory pattern of the receptive field appears after so few trials. However, it also makes it difficult to estimate the higher-resolution features of $\vec{\theta}$, which is why $\vec{\mu}_t$ does not improve much between 1000 and 5000 trials.

Similar results to Figure 5 in Paninski (2005) used a brute force computation and optimization of the mutual information. The computation in Paninski (2005) was possible only because $\vec{\theta}$ was assumed to be a Gabor function specified by just three parameters (the 2D location of its center and its orientation). Similarly, the stimuli were constrained to be Gabor functions. Our simulations did not assume that $\vec{\theta}$ or $\vec{x}_{t+1}$ was Gabor. $\vec{x}_{t+1}$ could have been any 40×40 image with power m^2. Attempting to use brute force in this high-dimensional space would have been hopeless. Our results show that a sequential optimal design allows us to perform system identification in high-dimensional spaces that might otherwise be tractable only by making strong assumptions about the system.

The fact that we can pick the stimulus to increase the information about the parameters, $\vec{\theta}_x$, that determine the dependence of the firing rate on the stimulus is unsurprising. Since we are free to pick any stimulus, by choosing an appropriate stimulus we can distinguish among different values of $\vec{\theta}_x$. Our GLM, however, can also include spike history terms. Since we cannot fully control the spike history, a reasonable question is whether infomax can improve our estimates of the spike history coefficients, $\vec{\theta}_f$. Figure 7 shows the results of a simulation characterizing the receptive field of a neuron whose response depends on its past spiking. The unknown parameter vector, $\vec{\theta} = [\vec{\theta}_x^T, \vec{\theta}_f^T]^T$, consists of the stimulus coefficients $\vec{\theta}_x$, which were a 1D Gabor function, and the spike history coefficients, $\vec{\theta}_f$, which were inhibitory and followed an exponential function. The nonlinearity was the exponential function.

The results in Figure 7 show that an infomax design leads to better estimates of both $\vec{\theta}_x$ and $\vec{\theta}_f$. Figure 7 shows the MAPs of both methods on different trials, as well as the mean squared error (MSE) on all trials. In Figure 7, the MSE increases on roughly the first 100 trials because the mean of the prior is zero. The data collected on these early trials tend to increase the magnitude of $\vec{\mu}_t$. Since the true direction of $\vec{\theta}$ is still largely unknown, the increase in the magnitude of $\vec{\mu}_t$ tends to increase the MSE.

By converging more rapidly to the stimulus coefficients, the infomax design produces a better estimate of how much of the response is due to $\vec{\theta}_x$, which leads to better estimates of $\vec{\theta}_f$. The size of this effect is measured by the correlation between $\vec{\theta}_x$ and $\vec{\theta}_f$, which is given by $\boldsymbol{C}_{x,f}$ in equation A.3. Consider a simple example where the first entry of $\boldsymbol{C}_{x,f}$ is negative and the remaining entries are zero. In this example, θ_{x_1} and θ_{f_1} (the first components of $\vec{\theta}_x$ and $\vec{\theta}_f$, respectively) would be anticorrelated. This value of $\boldsymbol{C}_{x,f}$ roughly means that the log posterior remains relatively constant if we increase θ_{x_1} but decrease θ_{f_1}. If we knew the value of θ_{x_1}, then we would

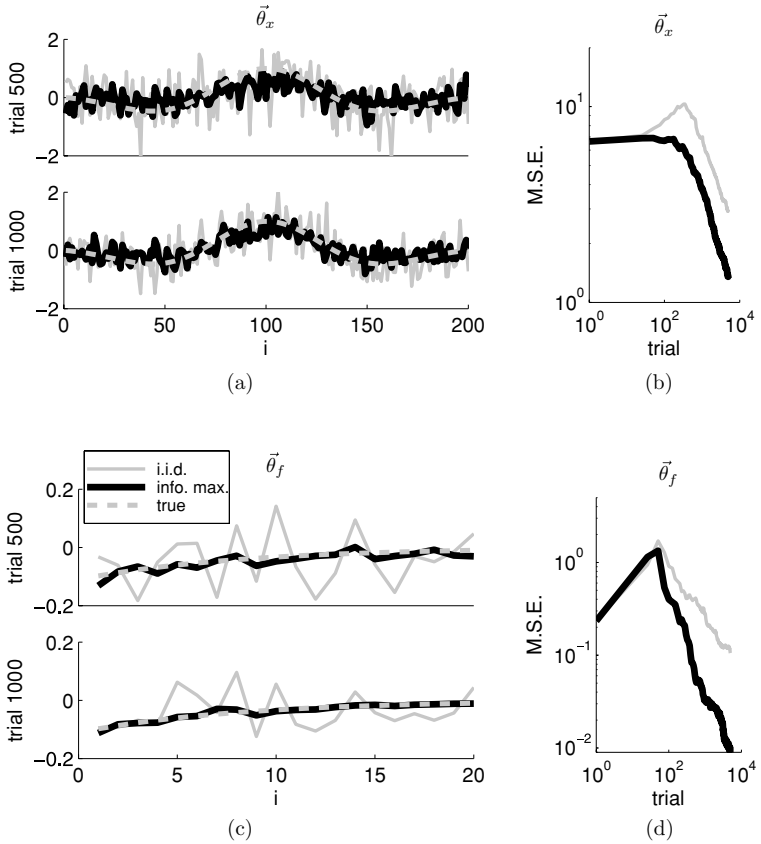

Figure 7: A comparison of parameter estimates using an infomax design versus an i.i.d. design for a neuron whose conditional intensity depends on both the stimulus and the spike history. (a) Estimated stimulus coefficients $\vec{\theta}_x$, after 500 and 1000 trials for the true model (dashed gray), infomax design (solid black) and an i.i.d. design (solid gray). (b) MSE of the estimated stimulus coefficients for the infomax design (solid black line) and i.i.d. design (solid gray line). (c) Estimated spike history coefficients, $\vec{\theta}_f$, after 500 and 1000 trials. (d) MSE of the estimated spike history coefficients.

know where along this line of equal probability the true parameters were located. As a result, increasing our knowledge about θ_{x_1} also reduces our uncertainty about θ_{f_1}.

5.4.1 Running Time. Our algorithm is suited to high-dimensional, real-time applications because it reduces the exponential complexity of choosing

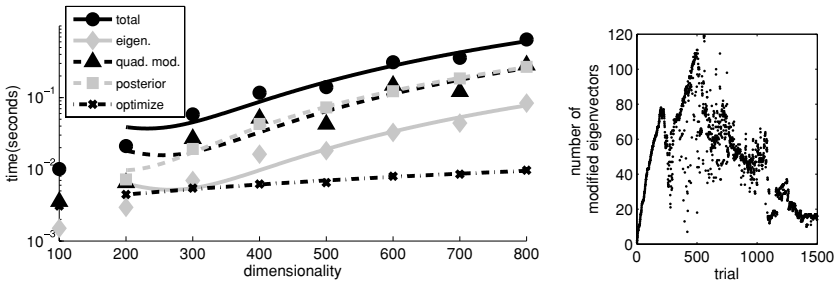

Figure 8: (a) Running time of the four steps that must be performed on each iteration as a function of the dimensionality of $\vec{\theta}$. The total running time as well as the running times of the eigendecomposition of the covariance matrix (eigen.), eigendecomposition of A in equation A.11 (quadratic modification), and posterior update were well fit by polynomials of degree 2. The time required to optimize the stimulus as a function of λ was well fit by a line. The times are the median over many iterations. (b) The running time of the eigendecomposition of the posterior covariance on average grows quadratically because many of our eigenvectors remain unchanged by the rank 1 perturbation. We verified this claim empirically for one simulation by plotting the number of modified eigenvectors as a function of the trial. The data are from a 20×10 Gabor simulation.

the optimal design to on average quadratic and at worst cubic running time. We verified this claim empirically by measuring the running time for each step of the algorithm as a function of the dimensionality of $\vec{\theta}$, Figure 8a.[3] These simulations used a GLM with an exponential link function. This nonlinearity leads to a special case of our algorithm because we can derive an analytical approximation of our objective function, equation 4.20, and only a one-dimensional search in $\mathcal{R}_{t+1}$ is required to find the optimal input. These properties facilitate implementation but do not affect the complexity of the algorithm with respect to d. Using a lookup table instead of an analytical expression to estimate the mutual information as a function of $(\mu_\rho, \sigma_\rho^2)$ would not change the running time with respect to d because $\mathcal{R}_{t+1}$ is always 2D. Similarly, the increased complexity of a full 2D search compared to a 1D search in $\mathcal{R}_{t+1}$ is independent of d.

The main conclusion of Figure 8a is that the complexity of our algorithm on average grows quadratically with the dimensionality. The solid black line shows a polynomial of degree 2 fitted to the total running time. We also measured the running time of the four steps that make up our

[3]These results were obtained on a machine with a dual core Intel 2.80GHz XEON processor running Matlab.

algorithm: (1) updating the posterior, (2) computing the eigendecomposition of the covariance matrix, (3) modifying the quadratic form for σ_ρ^2 to eliminate the linear constraint (i.e., finding the eigendecomposition of A in equation A.11) and (4) finding the optimal stimulus. The solid lines indicate fitted polynomials of degree 1 for optimizing the stimulus and degree 2 for the remaining curves. Optimizing the stimulus entails searching along the upper boundary of $\mathcal{R}_{t+1}$ for the optimal pair $(\mu_\rho^*, \sigma_\rho^{2*})$ and then finding an input that maps to $(\mu_\rho^*, \sigma_\rho^{2*})$. The running time of these operations scales as $O(d)$ because computing $\sigma_{\rho,\max}^2$ as a function of λ requires summing d terms, equation A.17. When $\vec{\theta}$ was 100 dimensions, the total running time was about 10 ms, which is within the range of tolerable latencies for many experiments. Consequently, these results support our conclusion that our algorithm can be used in high-dimensional, real-time applications.

When we optimize under the power constraint, the bottleneck is computing the eigendecomposition. In the worst case, the cost of computing the eigendecomposition will grow as $O(d^3)$. Figure 8a, however, shows that the average running time of the eigendecomposition grows only quadratically with the dimensionality. The average running time grows as $O(d^2)$ because most of the eigenvectors remain unchanged after each trial. The covariance matrix after each trial is a rank 1 perturbation of the covariance matrix from the previous trial, and every eigenvector orthogonal to the perturbation remains unchanged. A rank 1 update can be written as

$$M' = M + \vec{z}\vec{z}^T, \tag{5.12}$$

where M and M' are the old and perturbed matrices, respectively. Clearly, any eigenvector, $\vec{g}$, of M orthogonal to the perturbation, $\vec{z}$, is also an eigenvector of M' because

$$M'\vec{g} = M\vec{g} + \vec{z}\vec{z}^T\vec{g} = M\vec{g} = c\vec{s}, \tag{5.13}$$

where c is the eigenvalue corresponding to $\vec{g}$.

If the perturbation leaves most of our eigenvectors and eigenvalues unchanged, then we can use the Gu-Eisenstat algorithm to compute fewer than d eigenvalues and eigenvectors, thereby achieving on average quadratic running time (Gu & Eisenstat, 1994; Demmel, 1997; Seeger, 2007). Asymptotically, we can prove that the perturbation is correlated with at most two eigenvectors (see section 7). Consequently, asymptotically we need to compute at most two new eigenvectors on each trial. These asymptotic results, however, are not as relevant for the actual running time as empirical results. In Figure 8b, we plot, for one simulation, the number of eigenvectors that are perturbed by the rank 1 modification. On most trials, fewer than d eigenvectors are perturbed by the update. These results rely to some extent on the fact that our prior covariance matrix was white and hence

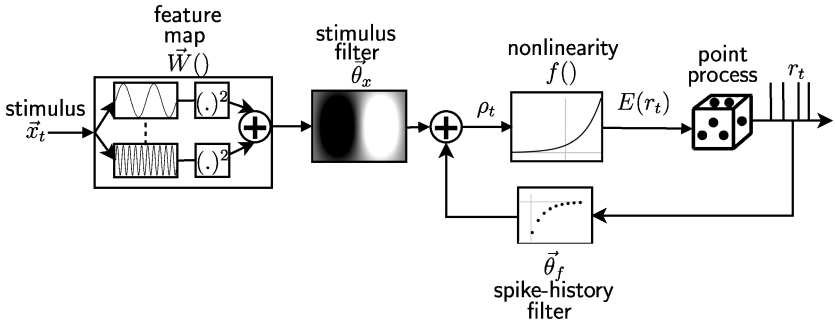

Figure 9: A GLM in which we first transform the input into some feature space defined by the nonlinear functions $W_i(\vec{x}_t)$—in this case, squaring functions.

had only one distinct eigenvalue. On each subsequent iteration, we can reduce the multiplicity of this eigenvalue by at most one. Our choice of prior covariance matrix therefore helps us manage the complexity of the eigendecomposition.

6 Important Extensions

In this section we consider two extensions of the basic GLM that expand the range of neurophysiology experiments to which we can apply our algorithm: handling nonlinear transformations of the input and dealing with time-varying $\vec{\theta}$. In both cases, our method for picking the optimal stimulus from a finite set requires only slight modifications. Unfortunately, our procedure for picking the stimulus under a power constraint will not work if the input is pushed through a nonlinearity.

6.1 Input Nonlinearities. Neurophysiologists routinely record from neurons that are not primary sensory neurons. In these experiments, the input to a neuron is a nonlinear function of the stimulus due to the processing in earlier layers. To make our algorithm work in these experiments, we need to extend our GLM to model the processing in these earlier layers. The extended model shown in Figure 9 is a nonlinear-linear-nonlinear (NLN) cascade model (Wu et al., 2006; Ahrens, Paninski, & Sahani, 2008; Paninski et al., 2007). The only difference from the original GLM is how we define the input:

$$\vec{s}_t = \left[W_1(\vec{x}_t), \ldots, W_{n_w}(\vec{x}_t), r_{t-1}, \ldots, r_{t-t_a}\right]^T. \tag{6.1}$$

The input now consists of nonlinear transformations of the stimulus. The nonlinear transformations are denoted by the functions W_i. These functions map the stimulus into feature space a simple example being the case where

the functions W_i represent a filter bank. n_w denotes the number of nonlinear basis functions used to transform the input. For convenience, we denote the output of these transformations as $\vec{W}(\vec{x}_t) = [W_1(\vec{x}_t), \ldots, W_{n_w}(\vec{x}_t)]^T$. As before, our objective is picking the stimulus that maximizes the mutual information about the parameters, $\vec{\theta}$. For simplicity, we have assumed that the response does not depend on past stimuli, but this assumption could easily be dropped.

NLN models are frequently used to explain how sensory systems process information. In vision, for example, MT cells can be modeled as a GLM whose input is the output of a population of V1 cells (Rust et al., 2006). In this model, V1 is modeled as a population of tuning curves whose output is divisively normalized. Similarly in audition, cochlear processing is often represented as a spectral decomposition using gammatone filters (de Boer & de Jongh, 1978; Patterson et al., 1992; Lewicki, 2002; Smith & Lewicki, 2006). NLN models can be used to model this spectral decomposition of the auditory input, as well as the subsequent integration of information across frequency (Gollisch, Schutze, Benda, & Herz, 2002). One of the most important NLN models in neuroscience is the energy model. In vision, energy models are used to explain the spatial invariance of complex cells in V1 (Adelson & Bergen, 1985; Dayan & Abbot, 2001). In audition, energy models are used to explain frequency integration and phase insensitivity in auditory processing (Gollisch et al., 2002; Carlyon & Shamma, 2003).

Energy models integrate information by summing the energy of the different input signals. The expected firing rate is a nonlinear function of the integrated energy,

$$E(r_t) = f\left(\sum_i (\vec{\phi}^{i,T}\vec{x}_t)^2\right). \tag{6.2}$$

Each linear filter, $\vec{\phi}^i$, models the processing in an earlier layer or neuron. For simplicity, we present the energy model assuming the firing rate does not depend on past spiking. As an example of the energy model, consider a complex cell. In this model, each $\vec{\phi}^i$ models a simple cell. The complex cell then sums the energy of the outputs of the simple cells.

Energy models are an important class of models compatible with the extended GLM shown in Figure 9. To represent an energy model in our framework, we need to express energy integration as an NLN cascade. We start by expressing the energy of each channel as a vector matrix multiplication by introducing the matrices Q^i,

$$(\vec{\phi}^{i,T}\vec{x}_t)^2 = \vec{x}_t^T\vec{\phi}^i\vec{\phi}^{i,T}\vec{x}_t = \vec{x}_t^T Q^i\vec{x}_t. \tag{6.3}$$

The right-hand side of this expression has more degrees of freedom than our original energy model unless we restrict Q^i to be a rank 1 matrix. Letting

$Q = \sum_i Q^i$, we can write the energy model as

$$E(r_t) = f\left(\vec{x}_t^T Q \vec{x}_t\right) = f\left(\sum_{i,j} Q_{i,j} x_{i,t} x_{j,t}\right) \tag{6.4}$$

$$Q = \sum_i \vec{\phi}^i \vec{\phi}^{i,T},$$

where $x_{i,t}$ denotes the ith component of $\vec{x}_t$. This model is linear in the matrix coefficients $Q_{i,j}$ and the products of the stimulus components $x_{i,t}x_{j,t}$. To obtain a GLM, we use the input nonlinearity, $\vec{W}$, to map $\vec{x}_t$ to the vector $[x_{1,t}x_{1,t}, \ldots, x_{i,t}x_{j,t}, \ldots]^T$. The parameter vector for the energy model is the matrix Q rearranged as a vector $\vec{\theta} = [Q_{1,1}, \ldots, Q_{i,j}, \ldots]^T$, which acts on feature space not stimulus space.

Using the functions, W_i, to project the input into feature space does not affect our strategy for picking the optimal stimulus from a finite set. We simply have to compute $\vec{W}(\vec{x}_{t+1})$ for each stimulus before projecting it into $\mathcal{R}_{t+1}$ and computing the mutual information. Our solution for optimizing the stimulus under a power constraint, however, no longer works for two reasons. First, a power constraint on $\vec{x}_{t+1}$ does not in general translate into a power constraint on the values of $\vec{W}(\vec{x}_{t+1})$. As a result, we cannot use the algorithm of section 5.2 to find the optimal values of $\vec{W}(\vec{x}_{t+1})$. Second, assuming we could find the optimal values of $\vec{W}(\vec{x}_{t+1})$, we would need to invert $\vec{W}$ to find the actual stimulus. For many nonlinearities, the energy model being one example, $\vec{W}$ is not invertible.

To estimate the parameters of an energy model, we use our existing update method to construct a gaussian approximation of the posterior in feature space, $p(\vec{\theta} \mid \vec{\mu}_t, \boldsymbol{C}_t)$. We can then use the MAP to estimate the input filters $\vec{\phi}^i$. The first step is to rearrange the terms of the mean, $\vec{\mu}_t$, as a matrix, $\hat{Q}$. We then estimate the input filters, $\vec{\phi}^i$, by computing the singular value decomposition (SVD) of $\hat{Q}$. If $\hat{Q}$ converges to the true value, then the subspace corresponding to its nonzero singular values should equal the subspace spanned by the true filters, $\vec{\phi}^i$.

Since we can optimize the design only with respect to a finite set of stimuli, we devised a heuristic for making this set more dispersed throughout $\mathcal{R}_{t+1}$. For the energy model,

$$\mu_\rho = \vec{\mu}_t^T \vec{s}_{t+1} \tag{6.5}$$

$$= \sum_{i=1}^{n_w} \mu_{i,t} W_i(\vec{x}_{t+1}) \tag{6.6}$$

$$= \vec{x}_{t+1}^T \hat{Q} \vec{x}_{t+1} \tag{6.7}$$

$$\hat{Q}_{i,j} = \mu_{i+(j-1)\cdot\dim(\vec{x}),t}, \tag{6.8}$$

where $\mu_{i,t}$ is the ith component of $\vec{\mu}_t$. r_t in this example has no dependence on past responses; hence, we do not need to sum over the past responses to compute μ_ρ (i.e., $t_a = 0$). $\hat{Q}$ is just the MAP, $\vec{\mu}_t$, rearranged as a $\dim(\vec{x}) \times \dim(\vec{x})$ matrix. We construct each stimulus in $\hat{\mathcal{X}}_{heur,t+1}$ as follows:

1. We randomly pick an eigenvector, $\vec{v}$, of $\hat{Q}$ with the probability of picking each eigenvector being proportional to the relative energy of the corresponding eigenvalue.
2. We pick a random number, ω, by uniformly sampling the interval $[-m, m]$, where m^2 is the maximum allowed stimulus power.
3. We choose a direction, $\vec{\omega}$, orthogonal to $\vec{v}$ by uniformly sampling the $d - 1$ unit sphere orthogonal to $\vec{v}$.
4. We add the stimulus,

$$\vec{x} = \omega\vec{v} + \sqrt{m^2 - \omega^2}\vec{\omega}, \tag{6.9}$$

to $\hat{\mathcal{X}}_{heur,t+1}$.

This heuristic works because for the energy model, $\rho_{t+1} = \vec{x}_{t+1}^T Q \vec{x}_{t+1}$ measures the energy of the stimulus in feature space. For this model, feature space is defined by the eigenvectors of Q. Naturally, if we want to increase ρ_{t+1}, we should increase the energy of the stimulus along one of the basis vectors of feature space. The eigenvectors of $\hat{Q}$ are our best estimate for the basis vectors of feature space. Hence, μ_ρ, the expected value of ρ_{t+1}, varies linearly with the energy of the input along each eigenvector of $\hat{Q}$, equation 6.7.

The effectiveness of our heuristic is illustrated in Figure 10. This figure illustrates the mapping of stimuli into $\mathcal{R}_{t+1}$ space for stimulus sets constructed using our heuristic, $\hat{\mathcal{X}}_{heur,t+1}$, and stimulus sets produced by uniformly sampling the sphere, $\hat{\mathcal{X}}_{iid,t+1}$. Our heuristic produces a set of stimuli that is more spread out on the range of μ_ρ. As a result, $\hat{\mathcal{X}}_{heur,t+1}$ contains more informative stimuli than $\hat{\mathcal{X}}_{iid,t+1}$.

6.1.1 Auditory Simulation. We applied these estimation and optimization procedures to a simulation of an auditory neuron. We modeled the neuron using an energy model. For simplicity, our hypothetical neuron received input from just two neurons in earlier layers. We modeled these input neurons as gammatone filters that were identical except for a 90 degree difference in phase (de Boer & de Jongh, 1978; Patterson et al., 1992). We generated spikes by sampling a conditional Poisson process whose instantaneous, conditional firing rate was set by equation 6.2 with $Q_{true} = \vec{\phi}^1\vec{\phi}^{1,T} + \vec{\phi}^2\vec{\phi}^{2,T}$, $\vec{\phi}^1$ and $\vec{\phi}^2$ being the gammatone filters, and $f(\rho_{t+1}) = \exp(\rho_{t+1})$. We estimated the parameters, Q, using an i.i.d. and two infomax designs. The i.i.d. design uniformly sampled the stimulus from the sphere $\|\vec{x}_{t+1}\|^2 = m^2$. The two infomax designs picked the optimal stimulus in a subset of stimuli drawn from the sphere. In one case, this set was constructed using our heuristic, while in the other case, it was constructed by uniformly sampling the sphere.

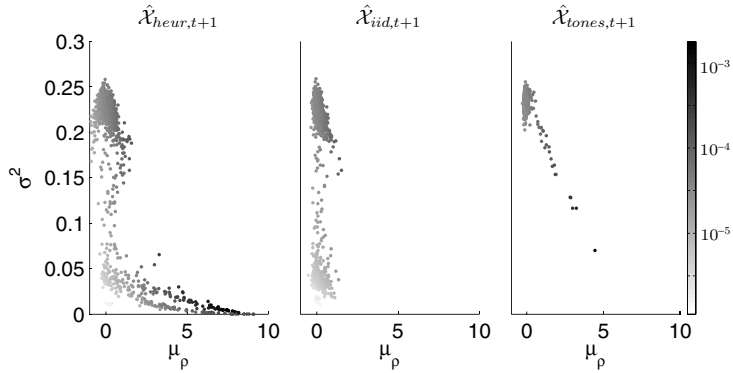

Figure 10: Plot shows the mapping of different stimulus sets into $\mathcal{R}_{t+1}$ after 500 trials. $\hat{\mathcal{X}}_{heur,t+1}$ consists of 1000 stimuli selected using the heuristic described in the text. $\hat{\mathcal{X}}_{iid,t+1}$ consists of 1000 stimuli randomly sampled from the sphere $||\vec{x}_{t+1}||_2 = m$. $\hat{\mathcal{X}}_{tones}$ is a set of 1000 pure tones with random phase and frequency, and power equal to m^2. All mappings were computed using the same posterior, which was taken from the simulation that picked the optimal stimulus in $\hat{\mathcal{X}}_{heur,t+1}$ on each trial. The shading of the dots is proportional to the mutual information of each input, equation 4.20. The plots show that $\hat{\mathcal{X}}_{heur,t+1}$ contains more informative stimuli than $\hat{\mathcal{X}}_{iid,t+1}$ and $\hat{\mathcal{X}}_{tones}$ and that the stimuli in $\hat{\mathcal{X}}_{heur,t+1}$ are more dispersed in $(\mu_\rho, \sigma^2_\rho)$ space.

The results of our simulations are shown in Figure 11. When finding the MAP of $\vec{\theta}$, we restricted $\vec{\mu}_t$ such that the corresponding matrices, $\hat{Q}$, were symmetric but not necessarily rank 2. The rank 2 restriction is unnecessary because the number of linear filters can be recovered from the number of nonzero singular values of $\hat{Q}$. To show how well the true gammatone filters can be estimated from the principal components of $\hat{Q}$, we show in Figure 11 the reconstruction of $\vec{\phi}^1$ and $\vec{\phi}^2$ using the first two principal components of $\hat{Q}$ that is, the linear combination of the projections of each filter along the first two principal components.

Figures 11 and 12 show that when the optimal stimulus in $\hat{\mathcal{X}}_{heur,t+1}$ is picked, the MAP converges more rapidly to the true gammatone filters. In Figure 11, the design that uses pure tones as the inputs appears to produce good estimates of the filters. These results, however, are somewhat misleading. Since these inputs are restricted to tones, the inputs that cause the neuron to fire are highly correlated. As a result, the estimated receptive field is biased by the correlations in the input. Since gammatone filters are similar to sine waves, in some sense, this bias means that using pure tones will rapidly produce a coarse estimate of the gammatone filters. However, since the pure tones are highly correlated, it is difficult to remove

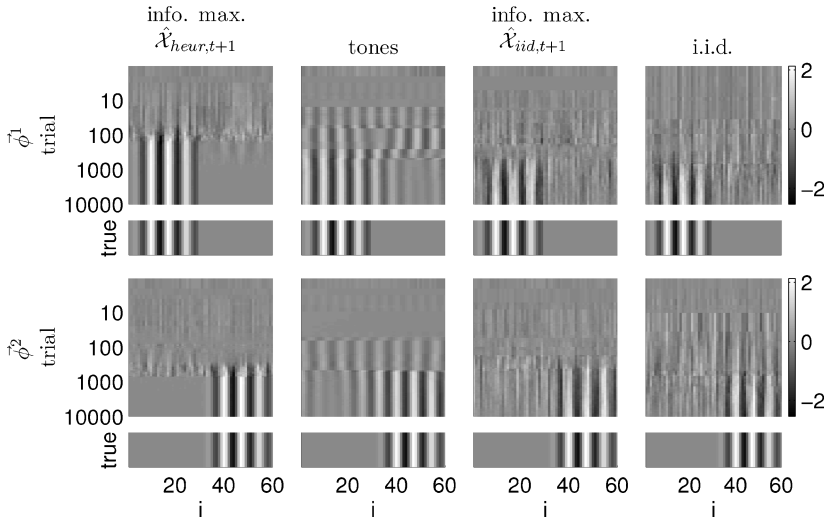

Figure 11: Simulation results for the hypothetical auditory neuron described in the text. Simulated responses were generated using equation 6.2 with $\vec{\phi}^1$ and $\vec{\phi}^2$ being gammatone filters. These filters were identical except for the phase, which differed by 90 degrees. The results compare an i.i.d. design, two infomax designs, and a design using pure tones. The two infomax designs picked the optimal stimulus in the sets $\hat{\mathcal{X}}_{heur,t+1}$ and $\hat{\mathcal{X}}_{iid,t+1}$ respectively; both sets contained 1000 inputs. The i.i.d. design picked the input by uniformly sampling the sphere $||\vec{x}_{t+1}||_2 = m$. The pure tones had random frequency and phase but power equal to m^2. To illustrate how well $\vec{\phi}^1$ and $\vec{\phi}^2$ can be estimated, we plot the reconstruction of $\vec{\phi}^1$ and $\vec{\phi}^2$ using the first two principal components of the estimated Q. The infomax design using a heuristic does much better than an i.i.d. design. For the infomax design, the gammatone structure of the two filters is evident starting around 100 and 500 trials, respectively. By 1000 trials, the infomax design using $\hat{\mathcal{X}}_{heur,t+1}$ has essentially converged to the true parameters, whereas for the i.i.d. design, the gammatone structure is starting to be revealed only after 1000 trials.

these correlations from the estimated receptive field and resolve the finer structure of the filters. This behavior is evident in Figure 12, which shows that after 1000 trials, the MSE for the pure tones design does not decrease as fast as for the alternative designs.

Also evident in the infomax results is the exploitation-exploration trade-off (Kaelbling, Littman, & Moore, 1996). To increase the information about one of the expected filters, we need to pick stimuli that are correlated with this filter. Since the input filters are orthogonal and the stimulus power is constrained, we can only effectively probe one filter at a time.

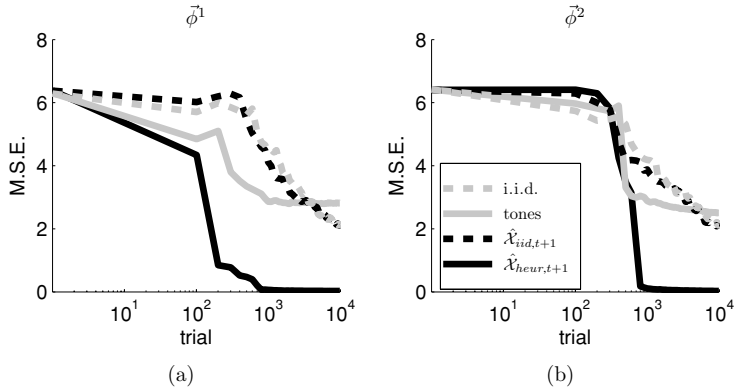

Figure 12: The MSE of the estimated filters shown in Figure 11. (a) MSE of $\vec{\phi}^1$. (b) MSE of $\vec{\phi}^2$. The solid black and dashed black lines show the results for designs that picked the optimal stimulus in $\hat{\mathcal{X}}_{heur,t+1}$ and $\hat{\mathcal{X}}_{iid,t+1}$, respectively. The solid gray line is for pure tones. The dashed gray line is for an i.i.d. design.

The exploitation-exploration trade-off explains why on trials 100 to 500, the estimate of the first filter improves much more than the second filter. On these trials, the algorithm exploits its knowledge of the first filter rather than searching for other filters. After roughly 500 trials, exploring becomes more rewarding than exploiting our estimate of $\vec{\phi}^1$. Hence, the infomax design picks stimuli orthogonal to the first gammatone filter, which eventually leads to our finding the second filter.

6.2 Time-Varying $\vec{\theta}$. Neural responses often change slowly over the course of an experiment due to changes in the health, arousal, or attentive state of the preparation (Lesica & Stanley, 2005). If we knew the underlying dynamics of $\vec{\theta}$, we could try to model these changes. Unfortunately, incorporating arbitrary, nonlinear dynamical models of $\vec{\theta}$ into our information-maximizing strategy is nontrivial because we would have to compute and maximize the expectation of the mutual information with respect to the unobserved changes in $\vec{\theta}$. Furthermore, even when we expect that $\vec{\theta}$ is varying systematically, we often have very little a priori knowledge about these dynamics. Therefore, instead of trying to model the actual changes in $\vec{\theta}$, we simply model the fact that the changes in $\vec{\theta}$ will cause our uncertainty about $\vec{\theta}$ to increase over time in the absence of additional observations. We can capture this increasing uncertainty by assuming that after each trial $\vec{\theta}$ changes in some small and unknown way (Ergun, Barbieri, Eden, Wilson, & Brown, 2007),

$$\vec{\theta}_{t+1} = \vec{\theta}_t + \vec{w}_t, \tag{6.10}$$

where $\vec{w}_t$ is normally distributed with a known mean and covariance matrix, Π. Using this simple model, we can factor into our optimization the loss of information about $\vec{\theta}$ due to its unobserved dynamics. Our use of gaussian noise can be justified using a maximum entropy argument. Since the gaussian distribution maximizes the entropy for a particular mean and covariance, we are in some sense overestimating the loss of information due to changes in $\vec{\theta}$. As a result, our uncertainty no longer converges to zero even asymptotically. This is the key property that our model must capture to ensure that our infomax algorithm will pick optimal stimuli. If we assume $\vec{\theta}$ is constant, then we would underestimate our uncertainty and, by extension, the amount of new information each stimulus would provide. Consequently, the infomax algorithm would do a poor job of picking the optimal stimulus.

To update the posterior and choose the optimal stimulus, we use the procedures described in sections 3 and 5. The only difference due to a time-varying $\vec{\theta}$ is that the covariance matrix of $p(\vec{\theta}_{t+1} \mid \boldsymbol{s}_{1:t+1}, \boldsymbol{r}_{1:t+1})$ is in general no longer just a rank 1 modification of the covariance matrix of $p(\vec{\theta}_t \mid \boldsymbol{s}_{1:t}, \boldsymbol{r}_{1:t})$. Therefore, we cannot use the rank 1 update to compute the eigendecomposition. However, since we may not have any a priori knowledge about the direction of changes in $\vec{\theta}$, it is often reasonable to assume $\vec{w}_t$ has mean zero and white covariance matrix, $\Pi = cI$. In this case, the eigenvectors of $\boldsymbol{C}_t + \Pi$ are those of $\boldsymbol{C}_t$, and the eigenvalues are $c_i + c$ where c_i is the ith eigenvalue of $\boldsymbol{C}_t$; in this case, our methods may be applied without modification. In cases where we expect that $\vec{\theta}$ varies systematically, we could try to model those dynamics more accurately by selecting an appropriate mean and covariance matrix for $\vec{w}_t$.

Figure 13 shows the results of using an infomax design to fit a GLM to a neuron whose receptive field drifts nonsystematically with time. The receptive field was a one-dimensional Gabor function whose center moved according to a random walk (we have in mind a slow random drift of eye position during a visual experiment). Although only the center of $\vec{\theta}$ moved, we still modeled changes in $\vec{\theta}$ using equation 6.10. The results demonstrate the benefits of using an infomax design to estimate a time-varying $\vec{\theta}$. Although we cannot reduce our uncertainty below a level determined by Π, the infomax design can still improve our estimate of $\vec{\theta}$ compared to using random stimuli.

7 Asymptotically Optimal Design

Our simulation results have shown that our algorithm can decrease our uncertainty more rapidly than an i.i.d. design. Naturally we would also like to know how well we do compared to the truly optimal design. To efficiently maximize $I(r_{t+1}; \vec{\theta} \mid \boldsymbol{s}_{1:t+1}, \boldsymbol{r}_{1:t})$, we approximated the posterior as a gaussian distribution. We would like to know how much this approximation costs us. In this section, we use an asymptotic analysis to investigate this question.

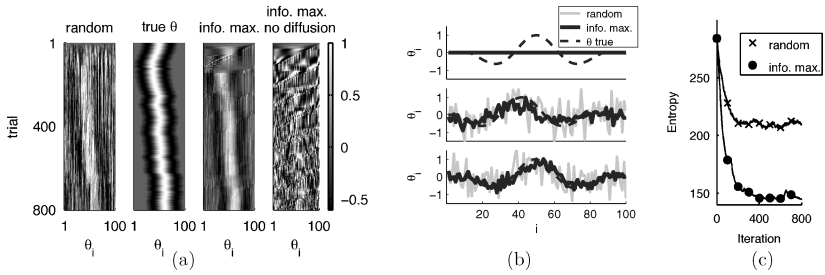

Figure 13: Estimating the receptive field when $\vec{\theta}$ is not constant. (a) The posterior means $\vec{\mu}_t$ and true $\vec{\theta}_t$ plotted after each trial. $\vec{\theta}$ was 100-dimensional, with its components following a Gabor function. To simulate slow drifts in eye position, the center of the Gabor function was moved according to a random walk between trials. We modeled the changes in $\vec{\theta}$ as a random walk with a white covariance matrix, Π, with variance .01. In addition to the results for random and information-maximizing stimuli, we also show the $\vec{\mu}_t$ estimated using stimuli chosen to maximize the information under the (mistaken) assumption that $\vec{\theta}$ was constant. Each row of the images plots $\vec{\mu}_t$ using intensity to indicate the value of the different components. (b) Details of the posterior means $\vec{\mu}_t$ on selected trials. (c) Plots of the posterior entropies as a function of trial number; once again, we see that information-maximizing stimuli constrain the posterior of $\vec{\theta}$ more effectively. The infomax design selected the optimal stimulus from the sphere $||\vec{x}_{t+1}||_2 = m$. The i.i.d. design picked stimuli by uniformly sampling this sphere.

The basis of this section is a central-limit-like theorem for infomax designs proved in Paninski (2005). This theorem states that asymptotically, the infomax design decreases our uncertainty at the same rate as a design that maximizes the expected Fisher information. This theorem uses the fact that the posterior of the infomax design is asymptotically normal, with mean and covariance

$$\vec{\mu}_t \xrightarrow{p} \vec{\theta} \tag{7.1}$$

$$(1/t)\boldsymbol{C}_t^{-1} \xrightarrow{p} E_{\vec{x}}(J_{\exp}(\vec{\theta}, \vec{x})) \tag{7.2}$$

$$p_{opt}(\vec{x}) = \arg\max_{p(\vec{x})} \log \mid E_{\vec{x}}(J_{\exp}(\vec{\theta}, \vec{x})) \mid . \tag{7.3}$$

Here the convergence, denoted by p, is in probability. $J_{\exp}$ is the expected Fisher information (evaluated at the true parameters). The expectation over $\vec{x}$ is with respect to the distribution $p_{opt}(\vec{x})$; the lack of the temporal subscript on $\vec{x}$ means the distribution is independent of time. p_{opt} represents an experimental design that picks the stimulus by sampling the stimulus distribution that maximizes the expected Fisher information, equation 7.3. This

design is nonadaptive (i.e., independent of the data already observed) because, unlike our infomax design, $p_{opt}(\vec{x})$ is independent of the posterior at time t. Asymptotically, the infomax design decreases our uncertainty at the same rate as p_{opt} because our uncertainty at time t is our prior uncertainty minus the information in the observations. As $t \to \infty$, the contribution of the prior information to the posterior entropy becomes negligible since we are dealing with an infinite series. Consequently, as $t \to \infty$, minimizing the posterior entropy becomes equivalent to maximizing the rate at which information is acquired, that is, the expected information of each observation, equation 7.3. Even though the infomax design is asymptotically equivalent to $p_{opt}(\vec{x})$, we cannot use $p_{opt}(\vec{x})$ instead of the infomax design in actual experiments because to compute $p_{opt}(\vec{x})$, we need to know $\vec{\theta}$.

The limit theorem for infomax designs, equations 7.1 and 7.2, holds only if the order of the trials does not matter as $t \to \infty$ (Paninski, 2005). Consequently, we can apply equation 7.2 only to situations where r_t depends on only the current stimulus, that is, $\vec{s}_t = \vec{x}_t$. Hence, in the remainder of this section, we use $\vec{x}_t$ instead of $\vec{s}_t$.

$p_{opt}(\vec{x})$ is the maximizer of a concave function over the convex set of valid stimulus distributions $p(\vec{x})$. Finding $p_{opt}(\vec{x})$ is closely related to "D-optimality" in the experimental design literature (Fedorov, 1972). Since the log determinant is concave, finding $p_{opt}(\vec{x})$ should be numerically stable because there are no local optima. In reality, numerical approaches become impractical when the stimulus domain is large. However, approximate approaches are still feasible; for example, we could search for the best p within some suitably chosen lower-dimensional subspace of the $(|\mathcal{X}| - 1)$-dimensional set of all possible $p(\vec{x})$.

Fortunately, when the stimulus domain is defined by a power constraint, there exists a semianalytical solution for p_{opt}. The complexity of this solution turns out to be independent of the dimensionality of the stimulus $\vec{x}$. We derive this result in the next section. In section 7.3, we present results showing that our infomax designs converge to the limiting design. These results show that our implementation is asymptotically optimal, despite the approximations we have made for numerical efficiency.

These asymptotic results allow us to quantify the relative efficiency of the infomax design compared to an i.i.d. design. For an i.i.d. design, equations 7.1 and 7.2 still hold, under appropriate conditions, provided we take the expectation in equation 7.2 with respect to the distribution, $p_{iid}(\vec{x})$, from which stimuli are selected on each trial (van der Vaart, 1998). As a result we can use equation 7.2 to compute and compare the asymptotic performance of our infomax design and $p_{iid}(\vec{x})$. In this section, the stimulus distribution $p_{iid}(\vec{x})$ refers to a uniform distribution on the sphere $||\vec{x}||_2 = m$.

7.1 Asymptotically Optimal Design Under a Power Constraint. In this section, we discuss the problem of finding $p_{opt}(\vec{x})$ under the power constraint $||\vec{x}||_2 \leq m$. This turns out to be surprisingly tractable: in particular,

we may reduce this apparently infinite-dimensional problem to a two-dimensional optimization problem that we can easily solve numerically.

Without loss of generality, we choose a coordinate system in which $\vec{x}$ is aligned with $\vec{\theta}$: $\theta_i = 0 \; \forall i \neq 1$. Using this parameterization, we may write our objective function as

$$F(p(\vec{x})) = \log \mid E_{\vec{x}} J_{\exp}(\vec{\theta}, \vec{x}) \mid \tag{7.4}$$

$$= \log \left| E_{x_1} \left(E_{r \mid x_1} D(r, x_1\theta_1) E_{x_2,\ldots,x_d \mid x_1} \left(\vec{x}\vec{x}^T \right) \right) \right|. \tag{7.5}$$

Recall the subscripts of $\vec{x}$ denote its components. The second integral above is just the correlation matrix of $\vec{x}$ taken over the stimulus distribution conditioned on x_1. A simple symmetry argument, along with the log concavity of the determinant, establishes that we may always find a spherically symmetric distribution $p(x_2, \ldots, x_d \mid x_1)$, which maximizes F for some $p(x_1)$ (the proof is in appendix E).

If we consider only spherically symmetric $p(x_2, \ldots, x_d \mid x_1)$, we can easily evaluate the inner integral in equation 7.5:

$$E_{\vec{x}_2,\ldots,x_d \mid x_1} \vec{x}\vec{x}^T = \begin{bmatrix} x_1^2 & \\ & \frac{1}{d-1} \left(E_{||\vec{x}||_2 \mid x_1} ||\vec{x}||_2^2 - x_1^2 \right) I_{\dim(\vec{x})-1} \end{bmatrix}, \tag{7.6}$$

where $I_{\dim(\vec{x})-1}$ is the $\dim(\vec{x}) - 1$-dimensional identity matrix. Using this result, we can easily evaluate the log determinant of the asymptotic covariance matrix:

$$\log \mid E_{\vec{x}} J_{\exp}(\vec{\theta}, \vec{x}) \mid = \log E_{x_1} E_{r \mid x_1} D(r, x_1\theta_1) x_1^2 + (d-1) \log E_{x_1} E_{r \mid x_1} D(r, x_1\theta_1) \frac{E_{||\vec{x}||_2^2 \mid x_1} ||\vec{x}||_2^2 - x_1^2}{d-1}. \tag{7.7}$$

To maximize the second term under a power constraint, $p(||\vec{x}||_2 \big| x_1)$ should have all its support on $||\vec{x}||_2 = m$. Since we also know $p_{opt}(x_2, \ldots, x_d \mid x_1)$ is spherically symmetric, $p_{opt}(x_2, \ldots, x_d \mid x_1)$ is just a uniform distribution on the $d-1$-dimensional sphere of radius $\sqrt{m^2 - x_1^2}$. To find the optimal distribution on x_1, we solve

$$p_{opt}(x_1) = \arg\max_{p(x_1)} \left[\log \phi + (d-1) \log \left(\frac{m^2\beta - \phi}{d-1} \right) \right] \tag{7.8}$$

$$\phi = E_{x_1} (E_{r \mid x_1} D(r, x_1\theta_1) {x_1}^2) \tag{7.9}$$

$$\beta = E_{x_1} (E_{r \mid x_1} D(r, x_1\theta_1)).$$

This objective function depends on $p(x_1)$ only through the two scalars ϕ and β, each of which is simply a linear projection of $p(x_1)$. As a result, we can always find a $p_{opt}(x_1)$ supported on just two values of x_1.[4] Thus, we have reduced our objective function to

$$\begin{aligned}
&\log \phi + (d-1)\log\left(\frac{m^2\beta - \phi}{d-1}\right) \\
&\quad = \log\left(w E_{r\mid y_1} D(r, y_1\theta_1) y_1^2 + (1-w) E_{r\mid y_2} D(r, y_2\theta_1) y_2^2\right) \\
&\qquad + (d-1)\log\left(w E_{r\mid y_1} D(r, y_1\theta_1)\left(m^2 - y_1^2\right)\right. \\
&\qquad \left. + (1-w) E_{r\mid y_2} D(r, y_2\theta_1)\left(m^2 - y_2^2\right)\right) + const., \qquad (7.10)
\end{aligned}$$

which has just three unknown parameters: the two support points (y_1, y_2) of $p(x_1)$, where $-m \le y_1 \le y_2 \le m$, and the relative probability mass on these support points (w here denotes the mass on the point y_1). w can be computed analytically as a function of (y_1, y_2) by setting the derivative of equation 7.10 with respect to w to zero. As a result, solving for the best values of (y_1, y_2, w) requires a simple two-dimensional numerical search over all pairs (y_1, y_2). In practice, we have found that the optimal $p(x_1)$ has support on a single point, $y_1 = y_2$, which reduces our problem to a one-dimensional search. While we cannot prove that this reduction holds in general, we can prove that it holds asymptotically as we increase d.

To prove that $p_{opt}(x_1)$ converges to a distribution with support on a single point as $d \to \infty$, we show that for any $(y1, y2)$, the optimal weight on y_1 asymptotically tends to $w = 0$ or $w = 1$. For any $(y1, y2)$, we compute w by setting the derivative of equation 7.10 with respect to w to 0:

$$w = \frac{bd\, y_2^2\left(a m^2 - b m^2 - a y_1^2 + b y_2^2\right) - a b m^2 y_1^2 + a b m^2 y_2^2}{d\left(b y_2^2 - a y_1^2\right)\left(a m^2 - b m^2 - a y_1^2 + b y_2^2\right)} \qquad (7.11)$$

$$a = E_{r\mid y_1} D(r, y_1\theta_1) \qquad (7.12)$$

$$b = E_{r\mid y_2} D(r, y_2\theta_2). \qquad (7.13)$$

Now, whenever the above equation yields $w \in [0, 1]$, that w is the optimal weight on y_1. If w is outside this interval, then $w = 0$ or $w = 1$, depending on which of these two values maximizes equation 7.10.

[4] Suppose we can find some optimal distribution $q(x_1)$ supported on more than two points. We can simply change $q(x_1)$ without changing our objective function by moving in some direction orthogonal to the two projections ϕ and β of $q(x_1)$. We may continue moving until we hit the boundary of the simplex of acceptable $q(x_1)$ (i.e., until $q(x_1) = 0$ for some value of x_1). By iterating this argument, we may reduce the number of points for which $p_{opt}(x_1) > 0$ down to two.

We can easily evaluate the limit of w as $d \to \infty$:

$$\lim_{d\to\infty} w = \frac{by_2^2}{by_2^2 - ay_1^2}. \tag{7.14}$$

b and a are positive because the Fisher information is always positive. Furthermore, $y_2 > y_1$ by assumption. These facts ensure that

$$\lim_{d\to\infty} w \leq 0 \quad \text{or} \quad \lim_{d\to\infty} w \geq 1. \tag{7.15}$$

In either case, the optimal weight ends up being $w = 1$ or $w = 0$, so the optimal distribution has support on only a single point as $d \to \infty$.

7.2 Relative Efficiency of the Infomax Design. We can quantify the relative efficiency of the infomax design to the i.i.d. design by computing the ratio of the asymptotic variances: the ratio of the dotted gray lines to the dotted black lines in Figures 14 and 15. The ratio

$$\frac{\sigma_{iid}^2(\vec{\omega})}{\sigma_{info}^2(\vec{\omega})} \triangleq \frac{\vec{\omega}^T \boldsymbol{C}_{iid}\vec{\omega}}{\vec{\omega}^T \boldsymbol{C}_{info}\vec{\omega}}, \tag{7.16}$$

measures how much faster the infomax design decreases the variance in direction $\vec{\omega}$ (a unit vector) than the i.i.d. design. $\boldsymbol{C}_{info}$ and $\boldsymbol{C}_{iid}$ are the asymptotic covariance matrices that come from equation 7.2:

$$C_{info} = (E_{p_{opt}(\vec{x})}(J_{\exp}(\vec{\theta}, \vec{x})))^{-1} \quad C_{iid} = (E_{p_{iid}(\vec{x})}(J_{\exp}(\vec{\theta}, \vec{x})))^{-1}. \tag{7.17}$$

We know from section 7.1 that for both designs, one eigenvector of $E_{\vec{x}}(J_{\exp}(\vec{\theta}, \vec{x}))$ is parallel to $\vec{\theta}$ and has an eigenvalue of $E_{\vec{x}}E_{r\,|\,\vec{x}}D(r, \vec{x}^T\vec{\theta})\frac{(\vec{x}^T\vec{\theta})^2}{||\vec{\theta}||_2^2}$. The remaining eigenvectors of $E_{\vec{x}}(J_{\exp}(\vec{\theta}, \vec{x}))$ all have an eigenvalue of $E_{\vec{x}}E_{r\,|\,\vec{x}}D(r, \vec{x}^T\vec{\theta})(m^2 - \frac{(\vec{x}^T\vec{\theta})^2}{||\vec{\theta}||_2^2})$. These results lead to simple expressions for $\sigma^2(\vec{\omega})$ for both designs,

$$\sigma^2(\vec{\omega}_{\|}) = \left(E_{\vec{x}}E_{r\,|\,\vec{x}}D(r, \vec{x}^T\vec{\theta})\frac{(\vec{x}^T\vec{\theta})^2}{||\vec{\theta}||_2^2}\right)^{-1} \tag{7.18}$$

$$\sigma^2(\vec{\omega}_{\perp}) = \left(E_{\vec{x}}E_{r\,|\,\vec{x}}D(r, \vec{x}^T\vec{\theta})\frac{m^2 - \frac{(\vec{x}^T\vec{\theta})^2}{||\vec{\theta}||_2^2}}{d-1}\right)^{-1}, \tag{7.19}$$

where $p(\vec{x})$ depends on whether we are computing $\sigma_{info}^2(\vec{\omega})$ or $\sigma_{iid}^2(\vec{\omega})$. $\vec{\omega}_{\|}$ is a unit vector parallel to $\vec{\theta}$, and $\vec{\omega_{\perp}}$ is a unit vector orthogonal to $\vec{\theta}$. Using these

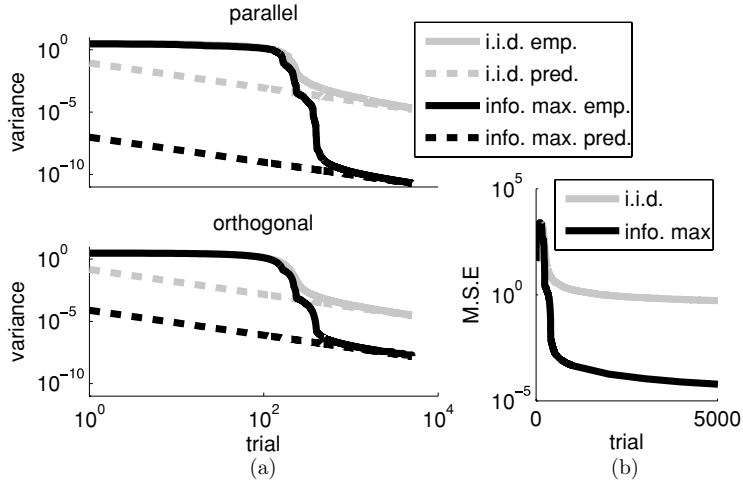

Figure 14: Comparison of the empirical posterior covariance matrix to the asymptotic variance predicted by equation 7.2. Despite our approximations, the empirical covariance matrix under an infomax design converged to the predicted value. (a) The top axis shows the variance in the direction of the posterior mean. The bottom axis is the geometric mean of the variances in directions orthogonal to the mean; asymptotically the variances in these directions are equal. The unknown $\vec{\theta}$ was a 11×15 Gabor patch. Stimuli were selected under the power constraint using an i.i.d. or infomax design. (b) The mean squared error between the empirical variance and the asymptotic variance.

expressions for $\sigma^2(\vec{\omega})$, we can compute the efficiency, $\frac{\sigma^2_{iid}(\vec{\omega})}{\sigma^2_{info}(\vec{\omega})}$, numerically for any nonlinearity. For the exponential Poisson model, we can derive some illustrative analytical results about the scaling of $\frac{\sigma^2_{iid}(\vec{\omega})}{\sigma^2_{info}(\vec{\omega})}$ with respect to d and $||\vec{\theta}||_2$.

For the exponential nonlinearity,

$$\frac{\sigma^2_{iid}(\vec{\omega}_{\parallel})}{\sigma^2_{info}(\vec{\omega}_{\parallel})} = \frac{E_{p_{opt}(\vec{x})} \exp(\vec{x}^T\vec{\theta}) \frac{(\vec{x}^T\vec{\theta})^2}{||\vec{\theta}||_2^2}}{E_{p_{iid}(\vec{x})} \exp(\vec{x}^T\vec{\theta}) \frac{(\vec{x}^T\vec{\theta})^2}{||\vec{\theta}||_2^2}} \tag{7.20}$$

$$\frac{\sigma^2_{iid}(\vec{\omega}_{\perp})}{\sigma^2_{info}(\vec{\omega}_{\perp})} = \frac{E_{p_{opt}(\vec{x})} \exp(\vec{x}^T\vec{\theta}) \left(m^2 - \frac{(\vec{x}^T\vec{\theta})^2}{||\vec{\theta}||_2^2}\right)}{E_{p_{iid}(\vec{x})} \exp(\vec{x}^T\vec{\theta}) \left(m^2 - \frac{(\vec{x}^T\vec{\theta})^2}{||\vec{\theta}||_2^2}\right)} \tag{7.21}$$

Naturally, both $\sigma^2_{iid}(\vec{\omega})$ and $\sigma^2_{info}(\vec{\omega})$ increase with d because as the dimensionality increases, we collect fewer observations in each direction for a fixed number of trials. Hence, as d increases, the variance increases.

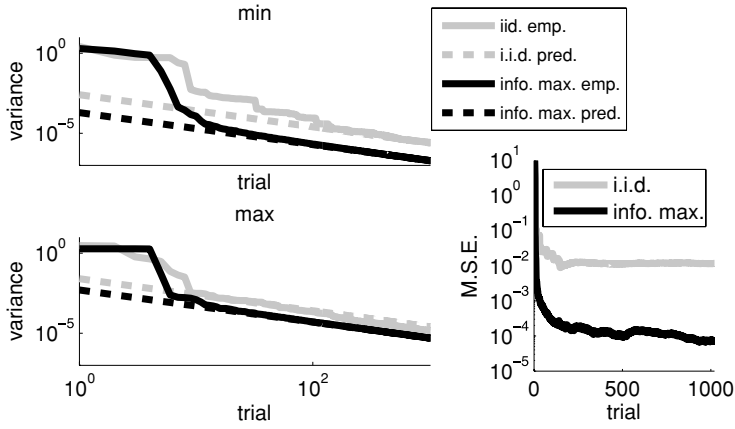

Figure 15: Comparison of the empirical variance of the posterior in our simulations to the asymptotic variance predicted based on the central limit theorem. The infomax design picked the optimal stimulus from a small number of stimuli (see text for details). (a) The axes compare the minimum eigenvalue and maximum eigenvalue of the asymptotic covariance matrix to the empirical variance in the direction of the corresponding eigenvalue. (b) A plot of the MSE between the empirical variance and the asymptotic variance.

Since the information of any stimulus depends on ρ_t, we would expect that the infomax design would become more efficient as d increases. Intuitively, as d increases, the probability of an i.i.d. design picking a direction that is highly correlated with $\vec{\theta}$ decreases because the variance of $(\vec{x}^T\vec{\theta})$ decreases linearly with d (see section 5.3). In contrast, the infomax design can use knowledge of $\vec{\theta}$ to ensure ρ_t is large with high probability even as the dimensionality grows.

We can in fact show that $\frac{\sigma^2_{iid}(\vec{\omega}_\|)}{\sigma^2_{info}(\vec{\omega}_\|)}$ is asymptotically linear in d. The d^{-1} scaling of the variance of $(\vec{x}^T\vec{\theta})$ for the i.i.d. design means that $\sigma^2_{iid}(\vec{\omega}_\|)$ and $\sigma^2_{iid}(\vec{\omega}_\perp)$ increase linearly with d.[5] For the i.i.d. design, each stimulus is equally likely. Therefore, the number of observations in any direction should decrease linearly with d. As a result, the variance in any direction increases linearly with d.

[5]For the i.i.d. design, $p(\vec{x}^T\frac{\vec{\theta}}{||\vec{\theta}||_2})$ has mean zero and variance m^2/d (see section 5.3 and Paninski, 2005; note that Paninski mistakenly had a scaling of d^{-2} here instead of the correct rate of d^{-1}). This result ensures that $\vec{x}^T\frac{\vec{\theta}}{||\vec{\theta}||_2}$ converges to zero at the rate $d^{-1/2}$. Since the power of $\vec{x}$ is constrained and the variance of $\vec{x}^T\frac{\vec{\theta}}{||\vec{\theta}||_2}$ decreases as $1/d$, it follows that both $\sigma^2_{iid}(\vec{\omega}_\|)$ and $\sigma^2_{iid}(\vec{\omega}_\perp)$ increase linearly with d.

In contrast, the infomax design can use the exponential increase of the Fisher information with $\vec{x}^T \frac{\vec{\theta}}{||\vec{\theta}||_2}$ to produce a slower increase of σ^2_{info} with d. To analyze the infomax design, we use the fact that as $d \to \infty$, $p_{opt}(x_1)$ converges to a distribution that has support on a single point, x_1. Furthermore, we can easily show (see appendix F) that as $d \to \infty$, x_1 converges to a constant away from 0 and m. This result means that $\sigma^2_{info}(\vec{\omega}_\parallel)$ is constant asymptotically with d, while $\sigma^2_{info}(\vec{\omega}_\perp)$ increases linearly with d. Since $\sigma^2_{iid}(\vec{\omega}_\parallel)$ scales linearly with d and $\sigma^2_{info}(\vec{\omega}_\parallel)$ is asymptotically constant with respect to d, the relative efficiency of the infomax design in direction $\vec{\omega}_\parallel$ increases linearly with d:

$$\frac{\sigma^2_{iid}(\vec{\omega}_\parallel)}{\sigma^2_{info}(\vec{\omega}_\parallel)} = O(d). \tag{7.22}$$

In directions orthogonal to $\vec{\theta}$, the relative efficiency of the infomax design is constant with respect to d because $\sigma^2_{iid}(\vec{\omega}_\perp)$ and $\sigma^2_{info}(\vec{\omega}_\perp)$ both increase linearly with d:

$$\frac{\sigma^2_{iid}(\vec{\omega}_\perp)}{\sigma^2_{info}(\vec{\omega}_\perp)} = O(1). \tag{7.23}$$

These results are also plotted in Figure 16. The important conclusion is that as d increases, we can reduce our uncertainty about $\vec{\theta}$ by a factor of d by using an infomax design as opposed to an i.i.d. design.

We can also consider the effect of increasing $||\vec{\theta}||_2$ for the exponential Poisson model. For this model, increasing $||\vec{\theta}||_2$ is roughly equivalent to increasing the signal-to-noise ratio because the Fisher information increases exponentially with $||\vec{\theta}||_2$. The infomax design can take advantage of the increase in the Fisher information by putting more stimulus energy along $\vec{\theta}$. For the i.i.d. design, most stimuli are orthogonal or nearly orthogonal to $\vec{\theta}$. Therefore, we would expect an increase in $||\vec{\theta}||_2$ to produce a much smaller decrease in the variances for the i.i.d. design than for the infomax design.

We can easily show that $\sigma^2_{i.i.d.}(\vec{\omega})/\sigma^2_{info}(\vec{\omega})$ increases at least exponentially with $||\vec{\theta}||_2$ by assuming that $p_{opt}(x_1)$ is supported on a single point, x_1. As we showed earlier, this assumption is always valid in the limit $d \to \infty$. By taking the limit of x_1 as $||\vec{\theta}||_2 \to \infty$ (see appendix F), we can show that x_1 converges to m. In contrast, for the i.i.d. design, the probability of $\vec{x}^T \frac{\vec{\theta}}{||\vec{\theta}||_2}$ being close to m is bounded away from 1. These differences in the marginal distribution of $p(\vec{x}^T \frac{\vec{\theta}}{||\vec{\theta}||_2})$ for the i.i.d. and infomax design imply that the ratios in equations 7.20 and 7.21 grow exponentially with $||\vec{\theta}||_2$.

7.3 Convergence to the Asymptotically Optimal Covariance Matrix. We can verify whether our design converges to the asymptotic design by

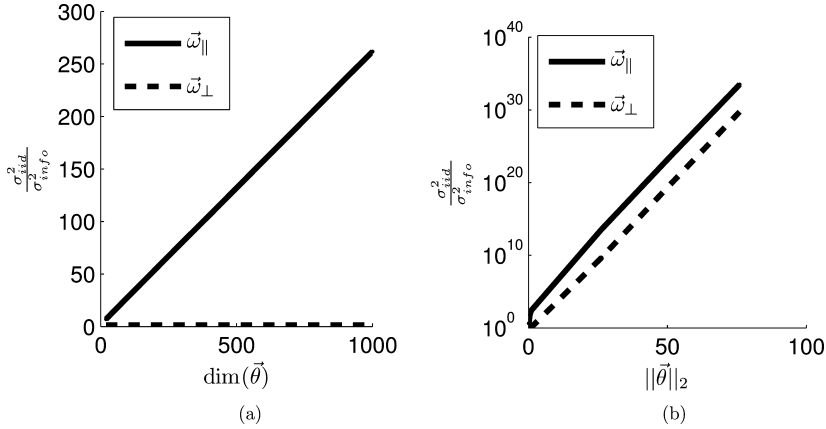

Figure 16: We measure the relative efficiency of the infomax design to the i.i.d. as the ratio of the variances, equation 7.16, for the exponential Poisson model. (a) $\frac{\sigma^2_{iid}(\vec{\omega})}{\sigma^2_{info}(\vec{\omega})}$ as a function of the dimensionality of $\vec{\theta}$. The ratio is computed with $\vec{\omega}$ set to a unit vector in the direction of $\vec{\theta}$ and a direction orthogonal to $\vec{\theta}$. The infomax design decreases the variance in the direction of $\vec{\theta}$ faster than the i.i.d. design by a factor that increases linearly with d. $\frac{\sigma^2_{iid}(\vec{\omega}_\perp)}{\sigma^2_{info}(\vec{\omega}_\perp)}$ has a value greater than one and is relatively flat with respect to d. Consequently, as d increases, the infomax design becomes more efficient at reducing the variance in the direction of $\vec{\theta}$ but not in directions orthogonal to $\vec{\theta}$. The stimulus domain was the unit sphere. The magnitude of $\vec{\theta}$ was also set to one. (b) $\frac{\sigma^2_{iid}(\vec{\omega})}{\sigma^2_{info}(\vec{\omega})}$ as a function of the magnitude of $\vec{\theta}$ when $dim(\vec{\theta}) = 1000$. The graph shows that the infomax design becomes exponentially more efficient than the i.i.d. design as we increase $||\vec{\theta}||_2$. The stimulus domain was again the unit sphere.

testing whether the covariance matrix of the posterior converges to the value predicted by equation 7.2 (see Figures 14 and 15). If the covariance matrix does not converge, then we conclude that our design is not decreasing our uncertainty as fast as the asymptotically optimal design.

Since the complexity of computing p_{opt} under a power constraint is independent of the dimensionality, we were able to perform this analysis for the high-dimensional Gabor results presented earlier. The symmetry of $p(x_2, \ldots, x_d \mid x_1)$ for the optimal and i.i.d. designs means the asymptotic covariance matrix has a simple structure: one eigenvector is parallel to $\vec{\theta}$, and the eigenvalues corresponding to all of the other eigenvectors (which are orthogonal to $\vec{\theta}$) are equal. Therefore, we just plot and compare the variance in the direction $\vec{\theta}$ and the geometric mean of the variances in directions orthogonal to $\vec{\theta}$.

We also wanted to test our infomax design when we picked the stimulus from a finite set. We chose a low five-dimensional example with just

100 stimuli to make computing p_{opt} numerically tractable. When $\vec{x}_{t+1}$ is restricted to a finite set, the asymptotic covariance matrix is no longer diagonal with directions orthogonal to $\vec{\theta}$ having equal variance. Therefore, in Figure 15, we compare the maximum and minimum eigenvalues of the asymptotic covariance matrix to the empirical variance in these directions. We also plot the MSE between the empirical and asymptotic covariance matrices. For comparison, we also computed the asymptotic variance for an i.i.d. design.

In the figures, the variances are relatively flat at the beginning because of the one-dimensionality of our GLM and the flatness of our prior. Since the one-dimensional GLM collects information in only one direction, we need to make d observations in order to decrease our initial uncertainty in all directions. Until we make d observations, the probability of the stimuli being correlated with $\vec{\theta}$ is low, and the variance in this direction remains high.

The main point of these figures is that our design does converge to the asymptotically optimal design. Furthermore, we see that maximizing the information decreases the variance much faster than an i.i.d. design. This is the expected result based on a theorem in Paninski (2005) that ensures the posterior entropy of an infomax design will in general be asymptotically no greater than that of an i.i.d. design. Infomax does better whenever the limiting design $p_{opt}(\vec{x})$ depends on $\vec{\theta}$, as this ensures there is not a single distribution that simultaneously maximizes the efficiency for all (a priori unknown) values of $\vec{\theta}$. For our GLM with a conditional Poisson (see Figure 2), the Fisher information depends on the stimulus and $\vec{\theta}$. Therefore, the optimal design cannot be determined a priori.

8 Misspecified Models

We used simulations to investigate the performance of the infomax algorithm when the link function, $f()$, is incorrect. The two primary questions we are interested in are whether the estimated $\vec{\theta}$ converges to the true value and how fast the uncertainty decreases compared to using i.i.d. stimuli.

A well-known result is that the parameters of a GLM can be estimated up to a scaling factor even if the link function is misspecified, provided the input distribution, $p(\vec{s}_{t+1})$, is elliptically symmetric (Li & Duan, 1989; Paninski, 2004). A distribution is elliptically symmetric if there exists a matrix A such that stimuli lying on the ellipse defined by $\| A\vec{s}_{t+1} \|_2 = const$ are equally likely. Our infomax design does not in general produce elliptically symmetric stimulus distributions because the 1D Fisher information, $D(r_{t+1}, \rho_{t+1})$, is not symmetric about $\rho_{t+1} = 0$. As a result, maximizing the mutual information leads to a marginal distribution $p(\rho_{t+1} = \vec{\mu}_t^T \vec{s}_{t+1})$ that is not symmetric about zero. We would therefore expect the infomax design to produce a biased estimate of $\vec{\theta}$ if the model is misspecified. This bias is due to an inevitable trade-off between efficiency and robustness. Ultimately, the only way to reduce the number of data points we need to fit a model is by

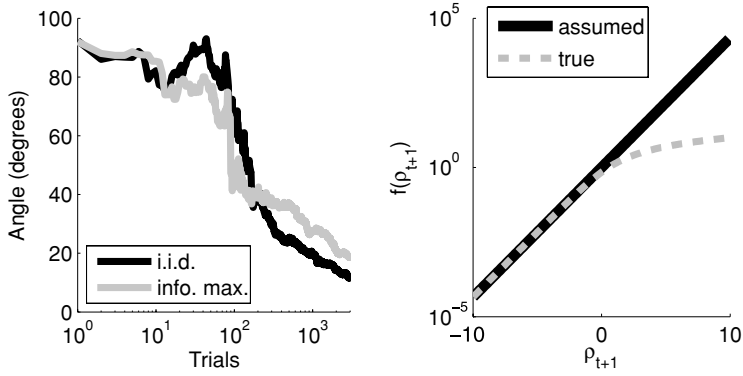

Figure 17: Effect of model misspecification. Infomax stimuli were selected using the wrong nonlinearity. The results compare the accuracy of the estimated $\vec{\theta}$ using i.i.d. stimuli versus infomax stimuli. Since the parameters can at best be estimated up to a scaling factor, (a) shows the angle between the estimated parameters and their true value. (b) Plot of the expected firing rate as a function of ρ_{t+1} for the true and assumed nonlinearities. The true nonlinearity was $f(\rho_{t+1}) = \log(1 + \exp(\vec{\theta}^T \vec{s}_{t+1}))$, while the assumed nonlinearity was $f(\rho_{t+1}) = \exp(\vec{\theta}^T \vec{s}_{t+1})$.

making assumptions about the model. These assumptions make it possible to infer the response function without observing the responses to every possible input. Stronger assumptions allow us to estimate the model using fewer data points. However, stronger assumptions increase the risk that our assumed model will be incorrect, which will bias our estimate of $\vec{\theta}$. We can make our design more robust by weakening our assumptions, for example, by using an elliptically symmetric design, but at the expense of being less efficient than the infomax design.

Nonetheless, our simulations showed that the estimates produced by the infomax design were comparable and sometimes better than those produced with i.i.d. data when the link function was misspecified. Figures 17 and 18 show the results for two different nonlinearities. In Figure 17, the simulated data were generated using the nonlinearity $f(\rho_{t+1}) = \log(1 + \exp(\vec{\theta}^T \vec{s}_{t+1}))$. The infomax design, however, assumed the nonlinearity was $f(\rho_{t+1}) = \exp(\vec{\theta}^T \vec{s}_{t+1})$. In this case, the assumed nonlinearity differs significantly from the true nonlinearity. In particular, for large ρ_{t+1}, the true nonlinearity is approximately linear in ρ_{t+1}. As a result, for the true model, the Fisher information is decreasing for very large ρ_{t+1} because the sensitivity of the response to the input is constant, but the variability of the response increases with ρ_{t+1}. Under the assumed model, however, the Fisher information is increasing with ρ_{t+1}. Consequently, the infomax design does a

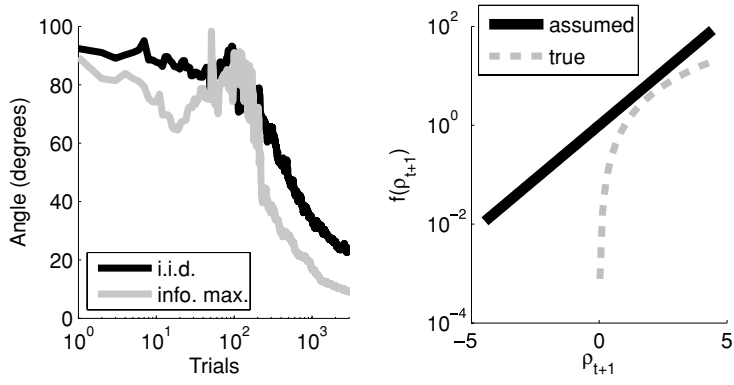

Figure 18: Same plots as in Figure 17 except here, the true nonlinearity was $f(\rho_{t+1}) = (\lfloor \vec{\theta}^T \vec{s}_{t+1} \rfloor^+)^2$ ($\lfloor\rfloor^+$ denotes half-wave rectification) and the assumed nonlinearity was $f(\rho_{t+1}) = \exp(\vec{\theta}^T \vec{s}_{t+1})$.

poor job of picking optimal stimuli. Nonetheless, using an infomax design leads to estimates that are nearly as good as those obtained with an i.i.d. design.

In Figure 18, the responses were simulated using the nonlinearity $f(\rho_{t+1}) = (\lfloor \vec{\theta}^T \vec{s}_{t+1} \rfloor^+)^2$ ($\lfloor\rfloor^+$ denotes half-wave rectification). The infomax design, however, took the nonlinearity to be $f(\rho_{t+1}) = \exp(\vec{\theta}^T \vec{s}_{t+1})$. As a result, even though the infomax design miscalculates the Fisher information, it correctly predicts that the Fisher information is increasing with ρ_{t+1}. Consequently, the infomax design produced smaller errors in the estimated $\vec{\theta}$. Even though the predicted mutual information is inaccurate, it is close enough to the true value that we can on average pick more informative stimuli than using an i.i.d. design.

9 Discussion

Previous work (MacKay, 1992; Chaloner & Verdinelli, 1995; Paninski, 2005) established a rigorous Bayesian framework for optimal sequential experimental design based on mutual information. Our work is a practical implementation suitable for high-dimensional, near-real-time applications using GLMs. Our algorithm depends on certain log concavity and asymptotic normality properties that models of neural systems often possess.

Our algorithm uses several ideas that are frequently employed in experimental design. The mutual information as a design criterion has been proposed by many authors (Lindley, 1956; Bernardo, 1979; Fedorov, 1972; MacKay, 1992; Paninski, 2005). To evaluate the mutual information, we use a normal approximation of the posterior. While we rely on a theorem due to

Paninski (2005), which proves asymptotic normality for the mutual information criterion, similar results concerning the asymptotics of sequential designs exist in the statistics literature (Wu, 1985; Chaudhuri & Mykland, 1993; Rosenberger & Hu, 2002). Furthermore, evaluating complicated, high-dimensional integrals by first approximating the function using an easily integrable function is a basic numerical quadrature technique. In addition to normality, we also rely on the structure of the GLM to facilitate the required computations. Sequential design has been successfully applied to GLMs before, but primarily with low-dimensional input spaces (Paninski, 2005; Roy et al., in press). The logistic model in particular has received a great deal of attention because it is frequently used for classification (Kontsevich & Tyler, 1999; Gilad-Bachrach, Navot, & Tishby, 2005; Schein, 2005; Roy et al., in press). Compared to our algorithm, previous algorithms for sequential design with GLMs do not scale nearly as well in high dimensions (Chaudhuri & Mykland, 1993; McLeish, 1999).

Optimal experimental design is also closely related to problems in optimal control (Movellan, 2005; Todorov, 2006) and reinforcement learning (Kaelbling et al., 1996; Boutilier, Dean, & Hanks, 1999). In reinforcement learning, the goal is to find the set of actions that maximize an agent's reward. Since the payoff of different actions is usually unknown a priori, the agent must simultaneously learn the payoffs of different actions while maximizing the reward. One important difference between our work and most formulations of reinforcement learning is that our reward signal, the mutual information, is not provided by the system being studied. Unlike most external reward signals, the payoff of $(\vec{x}_t, r_t)$ is highly dependent on the agent because the informativeness of any observation depends on the agent's existing knowledge.

9.1 Optimal Design in Neurophysiology. The application of sequential design to neurophysiology is not new (Benda, Gollisch, Machens, & Herz, 2007). A common approach to stimulus optimization in neurophysiology is to use model-free, finite-difference methods to measure the gradient of an objective function with respect to small perturbations in the stimulus (Foldiak, 2001; Gollisch et al., 2002; Edin, Machens, Schutze, & Herz, 2004; Machens, Gollisch, Kolesnikova, & Herz, 2005; O'Connor, Petkov, & Sutter, 2005). The firing rate and stimulus reconstruction error are two objective functions frequently optimized with this approach. Maximizing the firing rate is typically used to find a neuron's "preferred stimulus," which by definition is the stimulus that maximizes the firing rate of the neuron (Nelken, Prut, Vaadia, & Abeles, 1994; deCharms, Blake, & Merzenich, 1998; Foldiak, 2001; Zhang, Anderson, & Young, 2004; O'Connor et al., 2005). There is a natural connection between our objective function and maximizing the firing rate because, given our convexity conditions on $f()$, the preferred stimulus is closely related to $\vec{\theta}$. When encoding in sensory systems is studied, natural objective functions are the mutual information between the stimulus and

response (Machens et al., 2005) and the stimulus reconstruction error (Edin et al., 2004). These metrics are used to find stimuli that can be reconstructed with high fidelity from the neural responses.

An advantage of a finite-difference approach to stimulus adaptation is that an explicit model of the input-output function of a neuron is often unnecessary (Foldiak, 2001; Gollisch et al., 2002; O'Connor et al., 2005). However, these methods generally assume that the objective function with respect to the stimulus is fairly constant on successive trials. As a result, these methods can be highly susceptible to firing rate adaptation. In contrast, our method estimates the information using a model of the neuron's behavior. Our method is therefore highly dependent on the suitability of the GLM. However, since we can explicitly model adaptation and other potential nonstationarities, we automatically take their impact on the informativeness of different designs into account when optimizing our design.

9.2 Future Work. In most experiments, neurophysiologists are interested in how well we can model the neuron after all the data have been collected. We can measure the utility of the data set as the mutual information between all observations and $\vec{\theta}$, $I(\{\boldsymbol{r}_{1:t}, \vec{\theta}\} \mid \boldsymbol{x}_{1:t})$ where t represents the total number of trials. Unfortunately, there is no guarantee that a design based on maximizing $I(r_{t+1} \mid \vec{\theta}, \vec{x}_{t+1}, \boldsymbol{s}_{1:t}, \boldsymbol{r}_{1:t})$, will also maximize $I(\{\boldsymbol{r}_{1:t}, \vec{\theta}\} \mid \boldsymbol{x}_{1:t})$. When we pick stimuli by maximizing $I(r_{t+1} \mid \vec{\theta}, \vec{x}_{t+1}, \boldsymbol{s}_{1:t}, \boldsymbol{r}_{1:t})$, we ignore any effect $\vec{x}_{t+1}$ has on future trials. Ignoring future trials (i.e., using a greedy algorithm) simplifies the optimization problem. Greedy optimization, however, can be suboptimal because $\vec{x}_{t+1}$ can restrict the experiments we can conduct in future trials (Dasgupta, 2005). If the neuron's response depends on past stimuli or responses, then the choice of $\vec{x}_{t+1}$ will obviously constrain the input on trials after $t + 1$. Consequently, using a greedy algorithm limits our ability to optimize the experimental design to learn the neuron's dependence on past stimuli or responses (i.e., $\vec{\theta}_f$). Our algorithm can increase the information obtained about $\vec{\theta}_f$ only by exploiting the correlation between $\vec{\theta}_f$ and $\vec{\theta}_x$. In contrast, if we select a set of ordered stimuli to present on the next several trials, then we can directly control the entire stimulus history of the last trial in this sequence. We can also attempt to control the responses that are part of the input on the last trial. For these reasons, selecting a set of ordered stimuli allows us to change our design to maximize the information about the unknown parameters in a more direct fashion than greedy optimization.

Nongreedy optimization is more challenging than maximizing $I(r_{t+1} \mid \vec{\theta}, \vec{x}_{t+1}, \boldsymbol{s}_{1:t}, \boldsymbol{r}_{1:t})$. One of the primary challenges of nongreedy optimization is that the number of remaining trials is usually unknown because neurophysiologists will continue gathering data as long as the neuron is responding in a normal fashion. Assuming we pick some finite, arbitrary value for the number of remaining trials, the complexity of choosing the most informative sequence of stimuli will grow exponentially with the

number of trials because the dimensionality of the input and output spaces grows exponentially with the length of the sequence. The inclusion of spike history effects introduces additional complexity because the trials are no longer independent. Despite these challenges, nongreedy optimization is worth pursuing because if we can optimally learn spike history dependence, then we can begin to learn the structure of neural networks. To learn network structure, we simply modify the input of the GLM model so that a neuron's firing rate depends on the spiking of other neurons. Efficiently probing the network structure requires generating maximally informative patterns of stimuli and network activity. Generating these patterns requires nongreedy optimization because we can only influence future spiking, not past spiking.

Another extension that we are pursuing is how to incorporate more realistic priors. In our current algorithm, we can only represent prior beliefs as a gaussian prior on $\vec{\theta}$. This representation of prior knowledge is not flexible enough to represent the assumptions that are frequently adopted in real experiments. For example, we cannot represent the knowledge that $\vec{\theta}$ is sparse (Seeger, Gerwinn, & Bethge, 2007), low-rank (Depireux, Simon, Klein, & Shamma, 2001; Linden, Liu, Sahani, Schreiner, & Merzenich, 2003), or in some parametric family. In the near future, we hope to exploit knowledge that $\vec{\theta}$ lies in some parametric family of functions to help regularize our estimate of $\vec{\theta}$ in the absence of data, thereby improving the optimization of the stimuli.

Ultimately the goal of both improvements, nongreedy optimization and more refined priors, is to permit experiments that can help us understand the complex, nonlinear behavior of real neurons. These extensions will build on the solid mathematical framework we have developed in this article. We plan to apply this methodology to real experimental data in the near future.

Appendix A: Computing $\mathcal{R}_{t+1}$ Under the Power Constraint

In section 5.2 we outlined the procedure for computing $\mathcal{R}_{t+1}$ when $\mathcal{X} = \{\vec{x}_{t+1} : ||\vec{x}_{t+1}||_2 \leq m\}$. We find the boundary of $\mathcal{R}_{t+1}$ by maximizing and minimizing σ_ρ^2, equations 5.6 and 5.7, as a function of μ_ρ. To solve these optimization problems, we use the Karush-Kuhn-Tucker (KKT) conditions.

Since we can vary only $\vec{x}_{t+1} = \vec{s}_{x,t+1}$, we rewrite σ_ρ^2 as

$$\sigma_\rho^2 = \vec{s}_{t+1}^T \boldsymbol{C}_t \vec{s}_{t+1} \tag{A.1}$$

$$= \vec{s}_{x,t+1}^T \boldsymbol{C}_x \vec{s}_{x,t+1} + 2\vec{s}_{f,t+1}^T \boldsymbol{C}_{fx} \vec{s}_{x,t+1} + \vec{s}_{f,t+1}^T \boldsymbol{C}_f \vec{s}_{f,t+1}, \tag{A.2}$$

using the block matrix form for $\boldsymbol{C}_t$:

$$\boldsymbol{C}_t = \left[\begin{array}{c|c} \boldsymbol{C}_x & \boldsymbol{C}_{xf} \\ \hline \boldsymbol{C}_{fx} & \boldsymbol{C}_f \end{array}\right]. \tag{A.3}$$

To find the limits of σ_ρ^2 as a function of μ_ρ, we need to solve

$$\max \sigma_\rho^2 = \max_{\vec{s}_{x,t+1}} \vec{s}_{x,t+1}^T \boldsymbol{C}_x \vec{s}_{x,t+1} + 2\vec{s}_{f,t+1}^T \boldsymbol{C}_{fx} \vec{s}_{x,t+1} + \vec{s}_{f,t+1}^T \boldsymbol{C}_f \vec{s}_{f,t+1} \quad \text{(A.4)}$$

$$\min \sigma_\rho^2 = \min_{\vec{s}_{x,t+1}} \vec{s}_{x,t+1}^T \boldsymbol{C}_x \vec{s}_{x,t+1} + 2\vec{s}_{f,t+1}^T \boldsymbol{C}_{fx} \vec{s}_{x,t+1} + \vec{s}_{f,t+1}^T \boldsymbol{C}_f \vec{s}_{f,t+1} \quad \text{(A.5)}$$

$$\text{s.t} \quad \mu_\rho = \vec{s}_{t+1}^T \vec{\mu}_t \quad \|\vec{s}_{x,t+1}\,|\,|_2 \le m. \quad \text{(A.6)}$$

We can compute the limits of σ_ρ^2 as a function of μ_ρ by introducing two Lagrange multipliers to enforce the linear and quadratic constraints, respectively. Using a Lagrange multiplier to enforce the linear constraint, however, leads to a numerically unstable solution. A more stable approach is to use linear algebraic manipulations to derive an equivalent expression for σ_ρ^2 for which the linear constraint always holds. We start by rewriting μ_ρ as a 1D function of, α, the projection of $\vec{s}_{x,t+1}$ along the mean:

$$\mu_\rho = \alpha ||\vec{\mu}_{x,t}||_2 + \vec{\mu}_{f,t}^T \vec{s}_{f,t+1}. \quad \text{(A.7)}$$

To enforce the linear constraint, we first subtract from $\vec{s}_{x,t+1}$ its projection along $\vec{\mu}_{x,t}$ and then add to it a vector of length α in the direction of $\vec{\mu}_{x,t}$:

$$\vec{s}_{x,t+1}^{'} \stackrel{\triangle}{=} \vec{s}_{x,t+1} - \frac{\vec{\mu}_{x,t}^T \vec{s}_{x,t+1}}{||\vec{\mu}_{x,t}||_2^2} \vec{\mu}_{x,t} + \frac{\alpha}{||\vec{\mu}_{x,t}||_2} \vec{\mu}_{x,t}. \quad \text{(A.8)}$$

To enforce the linear constraint, we compute σ_ρ^2 by substituting $\vec{s}_{x,t+1}^{'}$ for $\vec{s}_{x,t+1}$ and then expanding using equation A.8:

$$\sigma_\rho^2 = \vec{s}_{x,t+1}^{'\,T} \boldsymbol{C}_t \vec{s}_{x,t+1}^{'} \quad \text{(A.9)}$$

$$= \vec{s}_{x,t+1}^T A \vec{s}_{x,t+1} + \vec{b}(\alpha)^T \vec{s}_{x,t+1} + d(\alpha) \quad \text{(A.10)}$$

$$A = \boldsymbol{C}_x - \frac{1}{2}\vec{v}\vec{v}^T + \frac{1}{2}\vec{u}\vec{u} \quad \text{(A.11)}$$

$$\vec{v} = \frac{-\vec{\mu}_{x,t}^T \boldsymbol{C}_x \vec{\mu}_{x,t} + 2||\vec{\mu}_x||_2^2}{2||\vec{\mu}_{x,t}||_2^3} \vec{\mu}_{x,t} + \boldsymbol{C}_x \frac{\vec{\mu}_{x,t}}{||\vec{\mu}_{x,t}||_2} \quad \text{(A.12)}$$

$$\vec{u} = \frac{-\vec{\mu}_{x,t}^T \boldsymbol{C}_x \vec{\mu}_{x,t} - 2||\vec{\mu}_x||_2^2}{2||\vec{\mu}_{x,t}||_2^3} \vec{\mu}_{x,t} + \boldsymbol{C}_x \frac{\vec{\mu}_{x,t}}{||\vec{\mu}_{x,t}||_2} \quad \text{(A.13)}$$

$$\vec{b}(\alpha) = 2\alpha \boldsymbol{C}_x \frac{\vec{\mu}_{x,t}}{||\vec{\mu}_{x,t}||_2} - 2\alpha(\vec{\mu}_{x,t}^T \boldsymbol{C}_x \vec{\mu}_{x,t}) \frac{\vec{\mu}_{x,t}}{||\vec{\mu}_{x,t}||_2^3}$$

$$+ 2(\boldsymbol{C}_{xf} \vec{s}_{f,t+1}) - 2(\vec{\mu}_{x,t}^T \boldsymbol{C}_{xf} \vec{s}_{f,t+1}) \frac{\vec{\mu}_{x,t}}{||\vec{\mu}_{x,t}||_2^2} \quad \text{(A.14)}$$

$$d(\alpha) = \alpha^2 \frac{(\vec{\mu}_{x,t}^T \boldsymbol{C}_x \vec{\mu}_{x,t})}{||\vec{\mu}_{x,t}||_2^2} + 2\alpha \frac{\vec{\mu}_{x,t}^T}{||\vec{\mu}_{x,t}||_2} \boldsymbol{C}_{xf} \vec{s}_{f,t+1} + \vec{s}_{f,t+1}^T \boldsymbol{C}_f \vec{s}_{f,t+1}. \quad \text{(A.15)}$$

The most important property of these quantities is that A is a rank 2 perturbation of $\boldsymbol{C}_x$ such that $\vec{\mu}_{x,t}^T A \vec{\mu}_{x,t} = 0$. As a result, one of the eigenvectors of A is parallel to $\vec{\mu}_{x,t}$ and has an eigenvalue of zero. Geometrically, equation A.10 defines the intersection of the ellipses defined by $\vec{s}_{t+1}^T \boldsymbol{C}_t \vec{s}_{t+1} = const$ with the plane defined by the linear constraint, $\mu_\rho = const$. Since equation A.10 is constant with respect to $\vec{\mu}_{x,t}^T \vec{s}_{x,t+1}^T$, we can always find a global maximum and minimum of σ_ρ^2 with $\vec{\mu}_{x,t}^T \vec{s}_{x,t+1} = 0$. Therefore, we can drop the linear constraint and just optimize equation A.10 under the power constraint $\|\vec{s}_{x,t+1}\|_2^2 \leq m^2 - \alpha^2$. Once we have found the optimal $\vec{\mu}_{x,t}$, we compute $\vec{s}'_{x,t+1}$. $\vec{s}_{x,t+1}$ satisfies the linear constraint while still maximizing or minimizing σ_ρ^2.

Optimizing a quadratic expression with a quadratic constraint is a well-studied optimization problem known as the trust region subproblem (TRS) (Fortin, 2000; Berkes & Wiskott, 2005). For the TRS, the KKT conditions are both necessary and sufficient (Fortin, 2000). Therefore, we can find all local minima and maxima by solving the KKT conditions.

Before we compute the KKT conditions, we transform our coordinates using the eigenbasis of A:

$$A = \boldsymbol{G}_t \Lambda_t \boldsymbol{G}_t^T \quad \vec{y}_{t+1} = \boldsymbol{G}_t^T \vec{s}_{x,t+1} \quad \vec{w}_t(\alpha) = \boldsymbol{G}_t^T \vec{b}(\alpha). \quad \text{(A.16)}$$

This transformation simplifies the expression for σ_ρ^2 because the value of σ_ρ^2 does not depend on interactions between the components of $\vec{y}_{t+1}$,

$$\max_{\vec{y}_{t+1}} \sigma_\rho^2 = \max_{\vec{y}_{t+1}} \sum_i c_{i,t} \left(y_{i,t+1} + \frac{w_{i,t}(\alpha)}{2c_{i,t}} \right)^2 - \frac{w_{i,t}^2(\alpha)}{4c_{i,t}} \quad \text{(A.17)}$$

$$\min_{\vec{y}_{t+1}} \sigma_\rho^2 = \min_{\vec{y}_{t+1}} \sum_i c_{i,t} \left(y_{i,t+1} + \frac{w_{i,t}(\alpha)}{2c_{i,t}} \right)^2 - \frac{w_{i,t}^2(\alpha)}{4c_{i,t}} \quad \text{(A.18)}$$

$$\text{s.t } ||\vec{y}_{t+1}||_2^2 \leq m^2 - \alpha^2, \quad \text{(A.19)}$$

where c_i denotes the ith eigenvalue of A. To enforce the power constraint, we introduce a Lagrange multiplier:

$$\sum_i c_{i,t} \left(y_{i,t+1} + \frac{w_{i,t}(\alpha)}{2c_{i,t}} \right)^2 - \frac{w_{i,t}^2(\alpha)}{4c_{i,t}} - \lambda y_{i,t+1}^2. \quad \text{(A.20)}$$

All local minima and maxima of σ_ρ^2 must either have a gradient equal to zero or else be located on the boundary. These necessary conditions, the

first-order KKT conditions, result in a system of d equations for the gradient of σ_ρ^2 with respect to $\vec{y}_{t+1}$:

$$2y_{i,t+1}(c_{i,t} - \lambda) = -w_i(\alpha) \quad \forall i. \tag{A.21}$$

When $\lambda \neq c_{i,t}$, we can solve the first-order KKT for $y_{i,t+1}$:

$$y_{i,t+1} = \frac{-w_{i,t}(\alpha)}{2(c_{i,t} - \lambda)}. \tag{A.22}$$

For a point not on the boundary to be a local maximum (minimum), the function must be concave (convex) at that point. These conditions, the second-order KKT conditions, can be checked by looking at the sign of the second derivative of σ_ρ^2 with respect to $\vec{y}_{t+1}$. For $\sigma^2_{\rho,\max}$, the second-order conditions are

$$c_{i,t} - \lambda \leq 0 \quad \forall i. \tag{A.23}$$

Therefore, $\sigma^2_{\rho,\max}$ must occur with $\lambda \geq c_{\max}$, where $c_{\max}$ is the maximum eigenvalue. The corresponding conditions for the local minima are

$$c_{i,t} - \lambda \geq 0 \quad \forall i, \tag{A.24}$$

that is, $\sigma^2_{\rho,\min}$ must occur for $\lambda \leq c_{\min} = 0$.

By solving the KKT conditions as a function of λ, we can find the points $(\mu_\rho, \sigma_\rho^2)$ corresponding to the boundary of $\mathcal{R}_{t+1}$. In this section, we assume the eigenvalues $c_{i,t}$ of $\boldsymbol{G}_t$, equation A.16, are sorted in increasing order. Hence, $y_{d,t+1}$ is the projection of the stimulus along the maximum eigenvector of $\boldsymbol{G}_t$. We also use $c_{\max,t}$ to denote the maximum eigenvalue. We refer to the set of $(\mu_\rho, \sigma_\rho^2)$ that solve the KKT conditions as $\mathcal{B}$. We divide $\mathcal{B}$ into subsets, denoted by subscripts, based on the corresponding value of the Lagrange multiplier for the points in that subset.

Since the second-order KKT conditions for $\sigma^2_{\rho,\max}$ are satisfied only if $\lambda \geq c_{\max,t}$, the set $\mathcal{B}_{\lambda=c_{\max,t}} \cup \mathcal{B}_{\lambda>c_{\max,t}}$ must contain all $(\mu_\rho, \sigma_\rho^2)$ corresponding to $\sigma^2_{\rho,\max}$. We can easily find all points in $\mathcal{B}_{\lambda>c_{\max,t}}$, as follows:

1. For $\lambda > c_{\max,t}$, compute $y_{i,t+1}$ in terms of α by plugging λ into equation A.22.
2. Find α by solving $\sum_i y^2_{i,t+1} = m^2 - \alpha^2$.
3. If $\alpha \in [-m, m]$ then compute $(\mu_\rho, \sigma_\rho^2) \in \mathcal{B}_{\lambda>c_{\max}}$.

We find α in step 2 by using the fact that the power constraint is always satisfied with equality for any local maximum of σ_ρ^2 because the eigenvalues are positive (Fortin, 2000). Hence, we can always increase σ_ρ^2 without

changing μ_ρ by increasing the energy of the stimulus along an eigenvector orthogonal to the mean. If the solution in step 2 satisfies $\alpha \in [-m, m]$, then the corresponding stimulus, $\vec{y}_{t+1}(\lambda, \alpha)$, is a local maximum of σ_ρ^2.

The set $\mathcal{B}_{\lambda=c_{\max}}$ is nonempty only if $w_{d,t}(\alpha) = 0$. If the maximum eigenvalue has a multiplicity greater than one, then this condition must hold for the projection of $\vec{s}_{x,t+1}$ along all eigenvectors corresponding to the maximum eigenvalue; otherwise, it is impossible to satisfy the first-order optimality conditions, equation A.22. Therefore, a simple test can tell us if we have to consider this harder case. To test for and find solutions at $\lambda = c_{\max}$, we consider two cases: (1) there is a finite number of α such that $w_{d,t}(\alpha) = 0$, and (2) $w_{d,t}(\alpha) = 0 \,\forall\, \alpha$.

The first case is easy. Since we set $\lambda = c_{\max,t}$, we can find α by solving $w_{d,t}(\alpha) = 0$. We can then compute all components of the stimulus except $y_{d,t+1}$ by plugging α and λ into equation A.22. Since $\sigma^2_{\rho,\max}$ is increasing with the stimulus power, we set $y_{d,t+1}$ so that the power constraint is satisfied with equality:

$$y_{d,t+1}^2 = m^2 - \alpha^2 - \sum_{i=1}^{d-1} y_{i,t+1}(c_{\max}, \alpha)^2. \tag{A.25}$$

If a real solution for $y_{d,t+1}$ exists, then the corresponding pair $(\mu_\rho, \sigma_\rho^2)$ is in $\mathcal{B}_{\lambda=c_{\max}}$.

The second case, $w_{d,t}(\alpha) = 0 \,\forall\, \alpha$, is more complicated because setting $\lambda = c_{\max,t}$ does not completely determine α. We find $(\mu_\rho, \sigma_\rho^2) \in \mathcal{B}_{\lambda=c_{\max}}$ as follows:

1. Vary $y_{d,t+1}^2$ on the interval $[0, m^2]$, and for each value evaluate steps 2 to 4.
2. Use $\lambda = c_{\max,t}$ and equation A.22 to compute $y_{i,t+1}$ for $1 \le i < d$ in terms of α.
3. Compute α by solving equation A.25 using the results from steps 1 and 2.
4. If $\alpha \in [-m, m]$, then compute $(\mu_\rho, \sigma_\rho^2) \in \mathcal{B}_{\lambda=c_{\max}}$.

If the maximum eigenvector has multiplicity greater than one, then in step 1, we simply vary the energy in the eigenspace of the maximum eigenvector. We can distribute the energy any way we like because the value of σ_ρ^2 is invariant to the distribution of the energy among the maximum eigenvectors. Since the KKT conditions are necessary and sufficient, the union $\mathcal{B}_{\lambda=c_{\max}} \cup \mathcal{B}_{\lambda>c_{\max}}$ contains all the points on the upper boundary of $\mathcal{R}_{t+1}$.

Since the second-order KKT conditions for $\sigma^2_{\rho,\min}$ are satisfied only for $\lambda \le 0$, all points on the lower boundary of $\mathcal{R}_{t+1}$ must be in $\mathcal{B}_{\lambda<0} \cup \mathcal{B}_{\lambda=0}$. We can easily find the points in $\mathcal{B}_{\lambda=0}$ as follows:

1. Let

$$\Phi = \left\{ \alpha : \sum_i y_{i,t+1}(\alpha)^2 = \sum_i \frac{w_{i,t}^2}{4c_{i,t}^2} \leq m^2 - \alpha^2 \,\&\, \alpha \in [-m, m] \right\}. \tag{A.26}$$

2. For each $\alpha \in \Phi$, compute $\vec{y}_{t+1}(\alpha)$ by plugging $\lambda = 0$ and α into equation A.22.
3. For each $\vec{y}_{t+1}(\alpha)$ and $\alpha \in \Phi$, compute $(\mu_\rho, \sigma_\rho^2) \in \mathcal{B}_{\lambda=0}$.

Clearly, equation A.18 is minimized by setting $y_{i,t+1}(\alpha) = -\frac{w_{i,t}}{2c_{i,t}}$. Unfortunately, this solution may not satisfy the power constraint for all values of α. The above procedure finds the values of α for which $y_{i,t+1}(\alpha) = -\frac{w_{i,t}}{2c_{i,t}}$ does not violate the power constraint.

The points in $\mathcal{B}_{\lambda<0}$ correspond to the values of α for which $y_{i,t+1}(\alpha) = -\frac{w_{i,t}}{2c_{i,t}}$ violates the power constraint. We can find the corresponding value of $\sigma^2_{\rho,\min}$ for these points as follows:

1. Vary λ on the interval $(-\infty, c_{\min})$.
2. For each λ, find α by solving $\sum_i y_{i,t+1}^2 = m^2 - \alpha^2$.
3. For each real α found in step 2, compute $\vec{y}_{t+1}$ by plugging λ and α into equation A.22.
4. Compute $(\mu_\rho, \sigma_\rho^2) \in \mathcal{B}_{\lambda<0}$.

Taken together, these procedures find all local maxima and minima of σ_ρ^2 as a function of μ_ρ. Consequently, $\mathcal{R}_{t+1}$ is the largest set of $(\mu_\rho, \sigma_\rho^2)$ enclosed by the points in $\mathcal{B}_{\lambda<0} \cup \mathcal{B}_{\lambda=0} \cup \mathcal{B}_{\lambda=c_{\max,t}} \cup \mathcal{B}_{\lambda>c_{\max,t}}$.

Numerically, this parameterization of the boundary is very stable. In particular, errors in small eigenvalues, $c_{i,t}$, will not cause problems provided $c_{\max,t}$ is not close to zero. As long as $c_{\max,t}$ is large relative to the smallest eigenvalues, σ_ρ^2 will be nearly invariant to errors in small eigenvalues. Consequently, the border of $\mathcal{R}_{t+1}$ will be insensitive to errors in the small eigenvalues. When all eigenvalues are close to zero, the lower and upper boundaries of $\mathcal{R}_{t+1}$ approach $\sigma_\rho^2(\mu_\rho) = 0$, and the solution remains stable.

To summarize, we can rapidly and stably compute the boundary of $\mathcal{R}_{t+1}$ by solving the KKT conditions as a function of the Lagrange multiplier. The most expensive operation is obtaining the eigendecomposition of A, which in the worst case is $O(d^3)$. However, as discussed in section 5.4.1, the average running time of computing the eigendecomposition of A scales as $O(d^2)$ in practice.

Appendix B: Proof of Convexity Condition

We now prove the lemma used in section 5.2 to establish conditions under which the mutual information is increasing with σ_ρ^2:

Lemma. *If $x \sim N(\mu, \sigma^2)$ and $g(x, \sigma^2)$ is,*

1. *convex in x and*
2. *increasing in σ^2*

then $E_x g(x, \sigma_\rho^2)$ is increasing in σ^2.

Proof. We start by defining the following change of variables,

$$y = \frac{x - \mu}{\sigma}, \tag{B.1}$$

where σ is the positive square root of σ^2. Using this change of variables,

$$E_x g(x, \sigma^2) = \int_{-\infty}^{\infty} g(x, \sigma^2) \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) dx \tag{B.2}$$

$$= \int_{-\infty}^{\infty} g(y\sigma + \mu, \sigma^2) \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{y^2}{2}\right) dy. \tag{B.3}$$

To show the expected value of $g()$ is increasing with σ^2, we need to show that the derivative with respect to σ^2 is positive:

$$\frac{d E_x g(x, \sigma^2)}{d\sigma^2} = \int_0^{\infty} \frac{\exp(-\frac{1}{2}y^2)}{\sqrt{2\pi}} [\big(g_{\sigma^2}(y\sigma + \mu, \sigma^2) + g_{\sigma^2}(-y\sigma + \mu, \sigma^2)\big)$$

$$+ \frac{y}{2\sigma}\big(g_x(y\sigma + \mu, \sigma^2) - g_x(-y\sigma + \mu, \sigma^2)\big)] > 0 \tag{B.4}$$

$$g_x(x, \sigma^2) = \frac{\partial g(x, \sigma^2)}{\partial x} \quad g_{\sigma^2}(x, \sigma^2) = \frac{\partial g(x, \sigma^2)}{\partial \sigma^2}. \tag{B.5}$$

Since $g(x, \sigma^2)$ is increasing with σ^2, $g_{\sigma^2}(y\sigma + \mu, \sigma^2)$ is always positive. The difference $g_x(y\sigma + \mu, \sigma^2) - g_x(-y\sigma + \mu, \sigma^2)$ is always positive because $g(x, \sigma^2)$ is convex in x. Therefore, $\frac{d E_x g(x, \sigma^2)}{d\sigma^2}$ is positive, which guarantees $E_x g(x, \sigma^2)$ is monotonically increasing in σ^2.

Appendix C: Generalization of the Power Constraint

We can easily modify our solution for optimizing the stimulus under the power constraint, section 5.2, so that we can choose the stimulus from an ellipsoid with arbitrary center and radii. In this case, the stimulus domain is defined as

$$\vec{s}_{t+1} = \vec{s}_{c,t+1} + \vec{s}_{r,t+1} \quad \vec{s}_{r,t+1}^T M \vec{s}_{r,t+1} \leq m^2, \tag{C.1}$$

where M is a symmetric, positive semidefinite matrix that defines the extent of the ellipsoid, and $\vec{s}_c$ defines the center of the ellipsoid. Unlike our initial power constraint, this generalization no longer maps to a well-defined physical constraint.

Computing the feasible region in $(\mu_\rho, \sigma_\rho^2)$ space under these constraints requires only slight modifications to the procedure already described. As before, we just need to compute the maximum and minimum of σ_ρ^2 as a function of μ_ρ, where

$$\sigma_\rho^2 = \vec{s}_{r,t+1}^T A \vec{s}_{r,t+1} + (2\vec{s}_{c,t+1}^T A + \vec{b})^T \vec{s}_{r,t+1} + \vec{s}_{c,t+1}^T A \vec{s}_{c,t+1} + \vec{b}^T \vec{s}_{c,t+1} + d. \tag{C.2}$$

We can easily eliminate the matrix M from our quadratic constraint by rotating and scaling $\vec{s}_{r,t+1}$ using the eigenvalues and eigenvectors of M:

$$M = \boldsymbol{G}_M \Lambda_M \boldsymbol{G}_M^T \quad \vec{y}_{r,t+1} = \Lambda^{1/2} \boldsymbol{G}_M^T \vec{s}_{r,t+1}. \tag{C.3}$$

In the new coordinate system, the quadratic constraint becomes $||\vec{y}_r||_2 \leq m$. Therefore, we can compute the feasible region in $(\mu_\rho, \sigma_\rho^2)$ space exactly as before. Computing the eigendecomposition of M does not affect the time complexity of our algorithm because it can be computed before the experiment starts.

Appendix D: Minimizing the MSE of $\vec{\theta}$

The mean squared error (MSE) of the parameters provides an alternative metric for our uncertainty about $\vec{\theta}$. The MSE is advantageous if we care about some components of $\vec{\theta}$ more than others. In this case, we can use the weighted MSE to represent our priorities. This alternative objective function leads to only a slightly modified optimization problem, which can be solved using essentially the same procedure.

The primary difference from maximizing the mutual information is that our objective function depends on the trace of the covariance matrix instead of the determinant. The MSE is

$$E_{\vec{\theta}\,|\,\{\boldsymbol{x}_{1:t}, \boldsymbol{r}_{1:t}\}}(||\vec{\theta} - \vec{\theta}_o||_2^2) = E_{\vec{\theta}\,|\,\{\boldsymbol{x}_{1:t}, \boldsymbol{r}_{1:t}\}}(\vec{\theta}^T\vec{\theta}) - 2\vec{\theta}_o^T E_{\vec{\theta}\,|\,\{\boldsymbol{x}_{1:t}, \boldsymbol{r}_{1:t}\}}(\vec{\theta}) + \vec{\theta}_o^T\vec{\theta}_o \tag{D.1}$$

$$= E_{\vec{\theta}\,|\,\{\boldsymbol{x}_{1:t}, \boldsymbol{r}_{1:t}\}}(\vec{\theta}^T\vec{\theta}) - 2\vec{\theta}_o^T\vec{\mu}_t + const, \tag{D.2}$$

where $\vec{\theta}_o$ is the true value of $\vec{\theta}$. Since $\vec{\theta}_o$ is unknown, the best we can do is estimate the MSE by taking the expectation with respect to our current

posterior,

$$E_{\vec{\theta}\,|\,\{\boldsymbol{x_{1:t}}, \boldsymbol{r_{1:t}}\}}(||\vec{\theta} - \vec{\theta}_o||_2^2) \approx E_{\vec{\theta}\,|\,\vec{\mu}_t, \boldsymbol{C}_t}\left(\vec{\theta}^T\vec{\theta} - \vec{\theta}^T\vec{\mu}_t\right) + const \tag{D.3}$$

$$= Tr(\boldsymbol{C}_t) + const, \tag{D.4}$$

where Tr is the trace.

To optimize the accuracy of the predicted responses, we pick the stimulus that will minimize the MSE once we add that stimulus and its response to our training set. Since $\boldsymbol{C}_{t+1}$ depends on the unknown observation, r_{t+1}, we compute $\boldsymbol{C}_{t+1}$ as a function of r_{t+1} and then take the expectation over the responses. The expected MSE if we pick $\vec{s}_{t+1}$ is

$$E_{\vec{\theta}} E_{r_{t+1}\,|\,\vec{s}_{t+1}, \vec{\theta}} Tr(\boldsymbol{C}_{t+1})$$

$$= E_{\rho_{t+1}} E_{r_{t+1}\,|\,\rho_{t+1}} (Tr\,\boldsymbol{C}_{t+1}) \tag{D.5}$$

$$= E_{\rho_{t+1}} E_{r_{t+1}\,|\,\rho_{t+1}} Tr\left(\boldsymbol{C}_t - \frac{\boldsymbol{C}_t\vec{s}_{t+1} D(r_{t+1}, \rho_{t+1})\vec{s}_{t+1}^T\boldsymbol{C}_t}{1 + D(r_{t+1}, \rho_{t+1})\vec{s}_{t+1}^T\boldsymbol{C}_t\vec{s}_{t+1}}\right) \tag{D.6}$$

$$= Tr(\boldsymbol{C}_t) - E_{\rho_{t+1}} E_{r_{t+1}\,|\,\rho_{t+1}} \frac{D(r_{t+1}, \rho_{t+1})\vec{s}_{t+1}^T\boldsymbol{C}_t\boldsymbol{C}_t\vec{s}_{t+1}}{1 + D(r_{t+1}, \rho_{t+1})\vec{s}_{t+1}^T\boldsymbol{C}_t\vec{s}_{t+1}} + const. \tag{D.7}$$

The expected MSE is very similar to $I(r_{t+1}; \vec{\theta}\,|\,\vec{x}_{t+1}, \boldsymbol{r_{1:t}}, \boldsymbol{x_{1:t}})$. The primary difference is that it depends on an additional scalar quantity, $\vec{s}_{t+1}^T\boldsymbol{C}_t\boldsymbol{C}_t\vec{s}_{t+1}$. Nonetheless, we can continue to pick the stimulus from a finite set using the methods presented in section 5.1.

Appendix E: Spherical Symmetry of $p_{opt}(x_2, \ldots, x_d\,|\,x_1)$

To derive the optimal asymptotic design in section 7.1, we used the fact that there always exists an optimal $p(x_2, \ldots, x_d\,|\,x_1)$ that is spherically symmetric. Here we prove this claim using a proof by contradiction. Let us assume that some distribution $\hat{p}(\vec{x}) = \hat{p}(x_1)\hat{p}(x_2, \ldots, x_d\,|\,x_1)$ with nonsymmetric $\hat{p}(x_2, \ldots, x_d\,|\,x_1)$ maximizes our objective function $F()$. We will show that we can construct a spherically symmetric $p^*(x_2, \ldots, x_d\,|\,x_1)$ such that $F(p^*(\vec{x}))$ is never smaller than $F(\hat{p}(\vec{x}))$. We can construct a spherically symmetric distribution by taking an average of $\hat{p}(x_2, \ldots, x_d\,|\,x_1)$ over all possible rotations Ψ_R. We define these rotations as

$$\Psi_R p(\vec{x}) = p(R\vec{x}) \tag{E.1}$$

$$R = \begin{bmatrix} 1 & 0 \\ 0 & R_{d-1} \end{bmatrix}, \tag{E.2}$$

where R_{d-1} is a $d-1$ orthonormal matrix. Since all directions orthogonal to $\vec{\theta}$ are equally informative, F is invariant to these transformations:

$$F(\Psi_R p(\vec{x})) = \log \left| \int D(r, x_1\theta_1)\vec{x}\vec{x}^T p(R\vec{x})\, d\vec{x} \right| \tag{E.3}$$

$$= \log \left| \int D(r, x_1\theta_1) R^T \vec{x}'\vec{x}'^T R p(\vec{x}')\, d\vec{x}' \right| \tag{E.4}$$

$$= 2\log |R| + F(p(\vec{x})) \tag{E.5}$$

$$= F(p(\vec{x})). \tag{E.6}$$

Here $\vec{x}'$ is the new stimulus after applying the transformation $\vec{x}' = R\vec{x}$. The last equality is true because for an orthonormal matrix, the determinant is 1. $p^*(\vec{x})$ is the average of $\hat{p}(\vec{x})$ over all possible transformations Ψ_R:

$$p^*(\vec{x}) = E_{\Psi_R}(\Psi_R(\hat{p}(\vec{x})). \tag{E.7}$$

Since F is concave, Jensen's inequality guarantees $F(p^*(\vec{x}))$ is never smaller than $F(\hat{p}(\vec{x}))$:

$$F(p^*(\vec{x})) = F(E_{\Psi_R}\Psi_R\hat{p}(\vec{x})) \geq E_{\Psi_R}F(\Psi_R\hat{p}(\vec{x})) = F(\hat{p}(\vec{x})). \tag{E.8}$$

The last equality is obviously true since $F(\Psi_R\hat{p}(\vec{x})) = F(\hat{p}(\vec{x}))$.

Appendix F: Support of $p_{opt}(\vec{x})$

In section 7.2, we derived some analytical results regarding the relative efficiency of the infomax to i.i.d. designs for the exponential Poisson model. These results use the fact that we can compute analytically the optimal support point when the marginal distribution $p_{opt}(x_1 = \vec{x}^T \frac{\vec{\theta}}{||\vec{\theta}||_2})$ is supported on a single point. To compute the optimal support point, x_1, we set $p_{opt}(x_1)$ to a distribution with support only on x_1. We then find the value of x_1, which maximizes equation 7.10 by setting the derivative of the equation equal to zero. The derivative of equation 7.10 with respect to x_1 is the cubic polynomial:

$$h(x_1) = -d||\vec{\theta}||_2 x_1^3 - 2dx_1^2 + dm^2||\vec{\theta}||_2 x_1 + 2m^2. \tag{F.1}$$

We can easily show that $h(x_1)$ only has one root in the interval $(0, m)$, and this root is the optimal value of x_1. To prove $h(x_1)$ has two negative roots,

we compute the second derivative of $h(x_1)$:

$$\frac{d^2h(x_1)}{dx_1^2} = -6d||\vec{\theta}||_2 x_1 - 4d. \tag{F.2}$$

Since the second derivative of $h(x_1)$ is negative for $x_1 \geq 0$, $h(x_1)$ is concave for $x_1 \geq 0$. This fact ensures that $h(x_1)$ can have at most two positive roots. However, since $h(0) = 2m^2$, $h(x_1)$ can in fact have only one positive root, which means that the other two roots are negative or zero. The positive root of $h(x_1)$ must lie in the interval $(0, m)$ because $h(m) = -2(d-1)m^2$, which is negative for all $d > 1$. To show that the positive root is the optimal value of x_1, we show that $x_1 \leq 0$ cannot be optimal. $x_1 = 0$ is not optimal because if the stimuli are orthogonal to $\vec{\theta}$, then we never collect any information in the direction of $\vec{\theta}$. We can easily rule out $x_1 < 0$ by computing the Fisher information:

$$\log \left| E_{\vec{x}} \exp(x_1||\vec{\theta}||_2)\vec{x}\vec{x}^T \right| = d x_1||\vec{\theta}||_2 + \log x_1^2 + (d-1)\log(m^2 - x_1^2). \tag{F.3}$$

Clearly if x_1 is negative, we can increase this expression by multiplying x_1 by negative one. So the optimal x_1 must be in the interval $(0, m)$. By using the cubic formula, we can obtain an analytical, albeit complicated, expression for x_1. In certain limiting cases, however, much simpler expressions for x_1 can be derived.

We can easily compute the limit of x_1 as $d \to \infty$. To compute the limit, we divide both sides of the equation $h(x_1) = 0$ by d and take the limit:

$$\lim_{d\to\infty} \frac{h(x_1)}{d} = x_1(-||\vec{\theta}||_2 x_1^2 - 2x_1 + m^2||\vec{\theta}||_2). \tag{F.4}$$

The roots of this polynomial are

$$x_1 = 0 \quad \& \quad x_1 = \frac{-1 \pm \sqrt{1 + ||\vec{\theta}||_2^2 m^2}}{||\vec{\theta}||_2}. \tag{F.5}$$

We showed earlier that the optimal value of x_1 must be greater than zero. So as $d \to \infty$, x_1 converges to the positive root, which is a constant away from 0 and m.

Similarly, we can prove that as $||\vec{\theta}||_2$ increases, x_1 converges to m. As $||\vec{\theta}||_2$ goes to infinity,

$$\lim_{||\vec{\theta}||_2\to\infty} \frac{h(x_1)}{||\vec{\theta}||_2} = -d x_1^3 + d m^2 x_1, \tag{F.6}$$

which has roots $x_1 = 0$ and $x_1 \pm m$. We can rule out the roots $x_1 = -m$ and $x_1 = 0$ because we know that for any finite $||\vec{\theta}||_2$, $h(x_1)$ has two negative roots and one root on the interval $(0, m)$. Therefore, as $||\vec{\theta}||_2$ increases, the two negative roots of $h(x_1)$ must approach $x_1 = 0$ and $x_1 = -m$, respectively, while the positive root converges to $x_1 = m$. Since we showed earlier that the positive root is always optimal, x_1 must approach m as $||\vec{\theta}||_2$ increases.

Acknowledgments

J.L. is supported by the Computational Science Graduate Fellowship Program administered by the DOE under contract DE-FG02-97ER25308 and by the NSF IGERT Program in Hybrid Neural Microsystems at Georgia Tech by grant DGE-0333411. L.P. is supported by grant EY018003 from the NEI and a Gatsby Foundation Pilot Grant. We thank P. Latham and M. Lewicki for helpful conversations. We also thank the reviewers for their helpful comments. Preliminary accounts of parts of this work have appeared in conference proceedings (NIPS06 and AISTATS07).

References

Adelson, E., & Bergen, J. (1985). Spatiotemporal energy models for the perception of motion. *Journal of the Optical Society of America A—Optics Image Science and Vision*, *2*(2), 284–299.

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

Bates, R. A., Buck, R. J., Riccomagno, E., & Wynn, H. P. (1996). Experimental design and observation for large systems. *Journal of the Royal Statistical Society Series B—Methodological*, *58*(1), 77–94.

Benda, J., Gollisch, T., Machens, C. K., & Herz, A. V. (2007). From response to stimulus: Adaptive sampling in sensory physiology. *Current Opinion in Neurobiology*, *17*(4), 430–436.

Berger, J. O. (1985). *Statistical decision theory and Bayesian analysis*. Berlin: Springer.

Berkes, P., & Wiskott, L. (2005). Slow feature analysis yields a rich repertoire of complex cell properties. *Journal of Vision*, *5*, 579–602.

Bernardo, J. M. (1979). Expected information as expected utility. *Annals of Statistics*, *7*(3), 686–690.

Berry, M., & Meister, M. (1998). Refractoriness and neural precision. *Journal of Neuroscience*, *18*, 2200–2211.

Boutilier, C., Dean, T., & Hanks, S. (1999). Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research*, *11*, 1–94.

Brillinger, D. (1988). Maximum likelihood analysis of spike trains of interacting nerve cells. *Biological Cyberkinetics*, *59*, 189–200.

Brillinger, D. (1992). Nerve cell spike train data analysis: A progression of technique. *Journal of the American Statistical Association*, *87*, 260–271.

Carlyon, R. P., & Shamma, S. (2003). An account of monaural phase sensitivity. *J. Acoust. Soc. Am.*, *114*(1), 333–348.

Chaloner, K., & Verdinelli, I. (1995). Bayesian experimental design: A review. *Statistical Science*, *10*(3), 273–304.

Chaudhuri, P., & Mykland, P. (1993). Nonlinear experiments: Optimal design and inference based on likelihood. *Journal of the American Statistical Association*, *88*(422), 538–546.

Chichilnisky, E. J. (2001). A simple white noise analysis of neuronal light responses. *Network—Computation in Neural Systems*, *12*(2), 199–213.

Cohn, D. A. (1994). Neural network exploration using optimal experiment design. In J. D. Cowan, G. Tesauro, & J. Alspector (Eds.), *Advances in neural information processing systems*, *6* (pp. 679–686). San Francisco: Morgan Kaufmann.

Cohn, D. A., Ghahramani, Z., & Jordan, M. I. (1996). Active learning with statistical models. *Journal of Artificial Intelligence Research*, *4*, 129–145.

Cottaris, N. P., & De Valois, R. L. (1998). Temporal dynamics of chromatic tuning in macaque primary visual cortex. *Nature*, *395*(6705), 896–900.

Cover, T. M., & Thomas, J. A. (1991). *Elements of information theory*. Hoboken, NJ: Wiley.

Dasgupta, S. (2005). Analysis of a greedy active learning strategy. In L. K. Saul, Y. Weiss, & L. Bottou (Eds.), *Advances in neural information processing systems*, *17* (pp. 337–344). Cambridge, MA: MIT Press.

Dayan, P., & Abbot, L. (2001). *Theoretical neuroscience*. Cambridge, MA: MIT Press.

de Boer, E., & de Jongh, H. R. (1978). On cochlear encoding: Potentialities and limitations of the reverse-correlation technique. *Journal of the Acoustical Society of America*, *63*(1), 115–135.

deCharms, R. C., Blake, D. T., & Merzenich, M. M. (1998). Optimizing sound features for cortical neurons. *Science*, *280*(5368), 1439–1443.

Demmel, J. W. (1997). *Applied numerical linear algebra*. Philadelphia: SIAM.

Depireux, D. A., Simon, J. Z., Klein, D. J., & Shamma, S. A. (2001). Spectro-temporal response field characterization with dynamic ripples in ferret primary auditory cortex. *Journal of Neurophysiology*, *85*, 1220–1234.

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. *Journal of Neuroscience*, *18*(7), 2626–2645.

Edin, F., Machens, C., Schutze, H., & Herz, A. (2004). Searching for optimal sensory signals: Iterative stimulus reconstruction in closed-loop experiments. *Journal of Computational Neuroscience*, *17*(1), 47–56.

Eggermont, J. J. (1993). Wiener and Volterra analyses applied to the auditory system. *Hearing Research*, *66*(2), 177–201.

Ergun, A., Barbieri, R., Eden, U. T., Wilson, M. A., & Brown, E. N. (2007). Construction of point process adaptive filter algorithms for neural systems using sequential Monte Carlo methods. *IEEE Transactions on Biomedical Engineering*, *54*(3), 419–428.

Fabian, V. (1978). On asymptotically efficient recursive estimation. *Annals of Statistics*, *6*, 854–866.

Fedorov, V. V. (1972). *Theory of optimal experiments*. Orlando, FL: Academic Press.

Foldiak, P. (2001). Stimulus optimisation in primary visual cortex. *Neurocomputing*, *38–40*, 1217–1222.

Fortin, C. (2000). *A survey of the trust region subproblem within a semidefinite framework*. Unpublished doctoral dissertation, University of Waterloo.

Gilad-Bachrach, R., Navot, A., & Tishby, N. (2005). Query by committee made real. In Y. Weiss, B. Schölkopf, & J. Platt (Eds.), *Advances in neural information processing systems*, *18* (pp. 443–450). Cambridge, MA: MIT Press.

Gollisch, T., Schutze, H., Benda, J., & Herz, A. V. M. (2002). Energy integration describes sound-intensity coding in an insect auditory system. *Journal of Neuroscience*, *22*(23), 10434–10448.

Gu, M., & Eisenstat, S. C. (1994). A stable and efficient algorithm for the rank-one modification of the symmetrical eigenproblem. *SIAM Journal on Matrix Analysis and Applications*, *15*(4), 1266–1276.

Haberman, S. (1977). Maximum likelihood estimation in exponential response models. *Annals of Statistics*, *5*, 815–841.

Henderson, H., & Searle, S. R. (1981). On deriving the inverse of a sum of matrices. *SIAM Review*, *23*, 53–60.

Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, *4*, 237–285.

Keat, J., Reinagel, P., Reid, R. C., & Meister, M. (2001). Predicting every spike: A model for the responses of visual neurons. *Neuron*, *30*, 803–817.

Kontsevich, L., & Tyler, C. (1999). Bayesian adaptive estimation of psychometric slope and threshold. *Vision Research*, *39*, 2729–2737.

Lesica, N. A., & Stanley, G. B. (2005). Improved tracking of time-varying encoding properties of visual neurons by extended recursive least-squares. *IEEE Trans. on Neural Systems and Rehabilitation Engineering*, *13*(2), 194–200.

Lewi, J., Butera, R., & Paninski, L. (2007). Efficient active learning with generalized linear models. In M. Meila & X. Shen (Eds.), *Proceedings of the Eleventh International Workshop on Artificial Intelligence and Statistics*. Available online at http://www.stat.umn.edu/∼aistat/proceedings/start.htm.

Lewicki, M. S. (2002). Efficient coding of natural sounds. *Nature Neuroscience*, *5*(4), 356–363.

Li, K. C., & Duan, N. (1989). Regression-analysis under link violation. *Annals of Statistics*, *17*(3), 1009–1052.

Linden, J. F., Liu, R. C., Sahani, M., Schreiner, C. E., & Merzenich, M. M. (2003). Spectrotemporal structure of receptive fields in areas AI and AAF of mouse auditory cortex. *Journal of Neurophysiology*, *90*, 2660–2675.

Lindley, D. V. (1956). On a measure of the information provided by an experiment. *Annals of Mathematical Statistics*, *27*(4), 986–1005.

Machens, C., Gollisch, T., Kolesnikova, O., & Herz, A. (2005). Testing the efficiency of sensory coding with optimal stimulus ensembles. *Neuron*, *47*(3), 447–456.

MacKay, D. J. C. (1992). Information-based objective functions for active data selection. *Neural Computation*, *4*(4), 590–604.

MacKay, D. (2003). *Information theory, inference, and learning algorithms*. Cambridge: Cambridge University Press.

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

McLeish, D. L. (1999). Designing the future: A simple algorithm for sequential design of a generalized linear model. *Journal of Statistical Planning and Inference*, *78*(1–2), 205–218.

Minka, T. P. (2001). Expectation propagation for approximate Bayesian inference. In *Proc. 17th Conf. in Uncertainty in Artificial Intelligence* (pp. 362–369). San Francisco: Morgan Kaufmann.

Movellan, J. R. (2005). *Infomax control as a model of real time behavior: Theory and application to the detection of social contingency* (Tech. Rep. 2005-1). San Diego, CA: University of California, San Diego, and Kyoto: ATR.

Nelken, I., Prut, Y., Vaadia, 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.

O'Connor, K. N., Petkov, C. I., & Sutter, M. L. (2005). Adaptive stimulus optimization for auditory cortical neurons. *Journal of Neurophysiology*, *94*(6), 4051–4067.

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

Paninski, L. (2005). Asymptotic theory of information-theoretic experimental design. *Neural Computation*, *17*(7), 1480–1507.

Paninski, L., Pillow, J., & Lewi, J. (2007). *Computational neuroscience: Theoretical insights into brain function*. Amsterdam: Elsevier.

Paninski, L., Shoham, S., Fellows, M. R., Hatsopoulos, N. G., & Donoghue, J. P. (2004). Superlinear population encoding of dynamic hand trajectory in primary motor cortex. *Journal of Neuroscience*, *24*(39), 8551–8561.

Patterson, R., Robinson, K., Holdsworth, J., McKeown, D., Zhang, C., & Allerhand, M. (1992). Complex sounds and auditory images. In *Auditory Physiology and Perception, Proceedings 9th International Symposium on Hearing* (pp. 429–446). Amsterdam: Elsevier.

Pillow, J. W., Paninski, L., Uzzell, V. J., Simoncelli, E. P., & Chichilnisky, E. J. (2005). Prediction and decoding of retinal ganglion cell responses with a probabilistic spiking model. *J. Neurosci.*, *25*(47), 11003–11013.

Ringach, D. L. (2002). Spatial structure and symmetry of simple-cell receptive fields in macaque primary visual cortex. *J. Neurophysiol*, *88*(1), 455–463.

Rosenberger, W. F., & Hu, M. X. (2002). On the use of generalized linear models following a sequential design. *Statistics and Probability Letters*, *56*(2), 155–161.

Roy, A., Ghosal, S., & Rosenberger, W. F. (in press). Convergence properties of sequential Bayesian D-optimal designs with applications to phase I clinical trials. *Journal of Statistical Planning and Inference*.

Rust, N. C., Mante, V., Simoncelli, E. P., & Movshon, J. A. (2006). How MT cells analyze the motion of visual patterns. *Nature Neuroscience*, *9*(11), 1421–1431.

Schein, A. (2005). *Active learning for logistic regression*. Unpublished doctoral dissertation, University of Pennsylvania.

Seeger, M. (2007). Low rank updates for the Cholesky decomposition (Tech. Rep.). Berkeley: University of California.

Seeger, M., Gerwinn, S., & Bethge, M. (2007). Bayesian inference for sparse generalized linear models. In *Machine Learning: ECML 2007*. Berlin: Springer.

Seeger, M., Steinke, F., & Tsuda, K. (2007). Bayesian inference and optimal design in the sparse linear model. In *Proc. 11th International Workshop on Artificial Intelligence and Statistics*. N.P.: Society for Artificial Intelligence and Statistics.

Sharia, T. (2007). *Recursive parameter estimation: Asymptotic expansion* (Tech. Rep.) London: University of London.

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

Smith, E. C., & Lewicki, M. S. (2006). Efficient auditory coding. *Nature*, *439*(7079), 978–982.

Theunissen, F. E., David, S. V., Singh, N. C., Hsu, A., Vinje, W. E., & Gallant, J. L. (2001). Estimating spatio-temporal receptive fields of auditory and visual neurons from their responses to natural stimuli. *Network—Computation in Neural Systems*, *12*(3), 289–316.

Todorov, E. (2006). *Bayesian brain*. Cambridge, MA: MIT Press.

Truccolo, W., Eden, U. T., Fellows, M. R., Donoghue, J. P., & Brown, E. N. (2005). A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects. *Journal of Neurophysiology*, *93*(2), 1074–1089.

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

Warmuth, M. K., Liao, J., Ratsch, G., Mathieson, M., Putta, S., & Lemmen, C. (2003). Active learning with support vector machines in the drug discovery process. *Journal of Chemical Information and Computer Sciences*, *43*(2), 667–673.

Watson, A., & Pelli, D. (1983). Quest: A Bayesian adaptive psychophysical method. *Perception and Psychophysics*, *33*, 113–120.

Wedderburn, R. (1976). On the existence and uniqueness of the maximum likelihood estimator for certain generalized linear models. *Biometrika*, *63*, 27–32.

Wu, C. F. J. (1985). Asymptotic inference from sequential design in a nonlinear situation. *Biometrika*, *72*(3), 553–558.

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

Zhang, K., &erson, M., & Young, E. (2004). Saddle-point property of nonlinear sensory response. In *Proceedings, Computational and Systems Neuroscience Meeting*. Available online at http://cosyne.org/c/5/5a/COSYNE04_Abstracts.pdf.

Received August 20, 2007; accepted February 25, 2008.