Published chapter in *Closed Loop Neuroscience*. ed. A. El Hady, Elsevier (2016).

Adaptive Bayesian methods for closed-loop neurophysiology

Jonathan W. Pillow
Princeton Neuroscience Institute
Dept. of Psychology
Princeton University
pillow@princeton.edu

Mijung Park
Gatsby Computational Neuroscience Unit
University College London
mijung@gatsby.ucl.ac.uk

Abstract

An important goal in the design of neurophysiology experiments is to select stimuli that rapidly probe a neuron's tuning or response properties. This is especially important in settings where the neural parameter space is multi-dimensional and the experiment is limited in time. Bayesian active learning methods provide a formal solution to this problem using a statistical model of the neural response and a utility function that quantifies what we want to learn. In contrast to staircase and other ad hoc stimulus selection methods, Bayesian active learning methods use the entire set of past stimuli and responses to make inferences about functional properties and select the next stimulus. Here we discuss recent advances in Bayesian active learning methods for closed-loop neurophysiology experiments. We review the general ingredients for Bayesian active learning and then discuss two specific applications in detail: (1) low-dimensional nonlinear response surfaces (also known as "tuning curves" or "firing rate maps"); and (2) high-dimensional linear receptive fields. Recent work has shown that these methods can achieve higher accuracy in less time, allowing for experiments that are infeasible with non-adaptive methods. We conclude with a discussion of open problems and exciting directions for future research.

1 Introduction

A primary goal in systems neuroscience is to characterize the statistical relationship between environmental stimuli and neural responses. This is commonly known as the *neural coding problem*, which is the problem of characterizing what aspects of neural activity convey information about the external world [1–5]. This problem is challenging because the relevant stimulus space is often high-dimensional and neural responses are stochastic, meaning that repeated presentations of a single stimulus elicit variable responses. Moreover, neural datasets are limited by the finite length of neurophysiological recordings. In many cases, experimenters wish to rapidly characterize neural tuning as a precursor to other experiments, or to track dynamic changes in tuning properties over time. Adaptive Bayesian methods for stimulus selection, which seek to choose stimuli according to an optimality criterion, provide a natural framework for addressing these challenges.

In classic "fixed design" experiments, stimuli are selected prior to the start of the experiment, or are

selected randomly from a fixed distribution, without regard to the observed responses. By contrast, adaptive or closed-loop designs take account of the responses as they are observed during the experiment in order to select future stimuli. Thus, if one observes that a neuron is insensitive to one dimension or region of stimulus space, one can spend more time exploring how the response varies across others. A variety of studies have developed adaptive methods for closed-loop experiments, with applications to linear receptive field estimation [6–9], color processing in macaque V1 [10–12], sound processing in auditory neurons [13–16], and nonlinear stimulus integration [17–19]. See [20, 21] for recent reviews.

Here we focus on Bayesian methods for adaptive experimental design, known in machine learning as *Bayesian active learning*. The basic idea is to define a statistical model of the neural response, then design stimuli or experiments to estimate the model parameters as efficiently as possible. The learning goal is specified by a utility function that determines the "most useful" stimulus given posterior uncertainty [22–27]. In the following, we introduce the basics of Bayesian active learning and describe two specific applications to closed-loop neurophysiology.

2 Bayesian active learning

A Bayesian active learning method or Bayesian adaptive design has three basic ingredients:

1. An *encoding model* $p(r|\mathbf{x}, \theta)$, which describes the conditional probability of a neural response r given a stimulus $\mathbf{x}$ and model parameters θ.
2. A *prior distribution* $p(\theta)$ over the model parameters.
3. A *utility function* $u(\theta, r, \mathbf{x}|\mathcal{D}_t)$, which quantifies the usefulness of stimulus-response pair $(\mathbf{x}, r)$ for learning about θ, given the data recorded so far in the experiment, $\mathcal{D}_t = \{(\mathbf{x}_1, r_1), \ldots, (\mathbf{x}_t, r_t)\}$.

Here we will consider the stimulus $\mathbf{x}$ to be a vector (e.g., the position, orientation, and spatial frequency of a sinusoidal grating, or the vector formed from a binary white noise image). We will consider the elicited response r to be a scalar (e.g., the spike count in some time window), although extending this framework to multivariate responses represents an important avenue for future research. Taken together, these ingredients fully specify both the uncertainty about the parameters given the observed data and the optimal stimulus at any point in the experiment.

2.1 Posterior and predictive distributions

The encoding model $p(r|\mathbf{x}, \theta)$ captures our assumptions about the encoding relationship between stimulus and response, i.e., the noisy process that takes stimuli and transforms them into spike responses. When considered as a function of the parameters θ, the encoding model provides the likelihood function. The prior distribution $p(\theta)$, in turn, characterizes our uncertainty about the parameters before the beginning of the experiment. These two ingredients combine to specify the *posterior distribution* over

the parameters given data according to Bayes' rule:

$$p(\theta|\mathcal{D}_t) \propto p(\theta) \prod_{i=1}^{t} p(r_i|\mathbf{x}_i, \theta), \tag{1}$$

where the model provides a likelihood term $p(r_i|\mathbf{x}_i, \theta)$ for each stimulus-response pair. The product of likelihood terms arises from the assumption that responses are conditionally independent given the stimulus on each time step. (Later, we will discuss relaxing this assumption).

Another important distribution that we can obtain from these ingredients is the *predictive distribution* of the response, $p(r|\mathbf{x}, \mathcal{D}_t)$ which is the conditional distribution of the response given the stimulus and all previously observed data $\mathcal{D}_t$, with uncertainty about the parameters θ integrated out. This is given by

$$p(r|\mathbf{x}, \mathcal{D}_t) = \int p(r|\mathbf{x}, \theta)\, p(\theta|\mathcal{D}_t)\, d\theta. \tag{2}$$

2.2 Utility functions and optimal stimulus selection

In sequential optimal designs, the experimenter selects a stimulus on each trial according to an optimality criterion known as the *expected utility*. This quantity is the average of the utility function over $p(r, \theta|x, \mathcal{D})$, the joint distribution of r and θ given a stimulus and the observed data so far in the experiment:

$$U(\mathbf{x}|\mathcal{D}_t) \triangleq \mathbb{E}_{r,\theta}\left[u(\theta, r, \mathbf{x}|\mathcal{D}_t)\right] = \iint u(\theta, r, \mathbf{x}|\mathcal{D}_t)\, p(r, \theta|\mathbf{x}, \mathcal{D}_t)\, d\theta\, dr, \tag{3}$$

or if responses are integer spike counts,

$$= \int \sum_{r=0}^{\infty} u(\theta, r, \mathbf{x}|\mathcal{D}_t)\, p(r, \theta|\mathbf{x}, \mathcal{D}_t)\, d\theta. \tag{4}$$

The experimenter selects the stimulus with maximal expected utility for the next trial:

$$\mathbf{x}_{t+1} = \arg\max_{\mathbf{x}^* \in \Omega} U(\mathbf{x}^*|\mathcal{D}_t), \tag{5}$$

where Ω is some set of candidate stimuli. Fig. 1 shows a schematic of the three iterative steps in Bayesian active learning: (1) present stimulus and observe response; (2) update posterior; (3) maximize expected utility to select a new stimulus.

The choice of utility function determines the notion of optimality for a Bayesian active learning paradigm. Intuitively, the utility function can be understood as providing a precise specification of what kinds of posterior distributions are considered "good". We will now review several choices for utility.

Maximum mutual information (infomax)

The most popular Bayesian active learning method seeks to maximize the gain in information about model parameters, an approach commonly known as *infomax learning* [7, 22, 27–31]. This approach

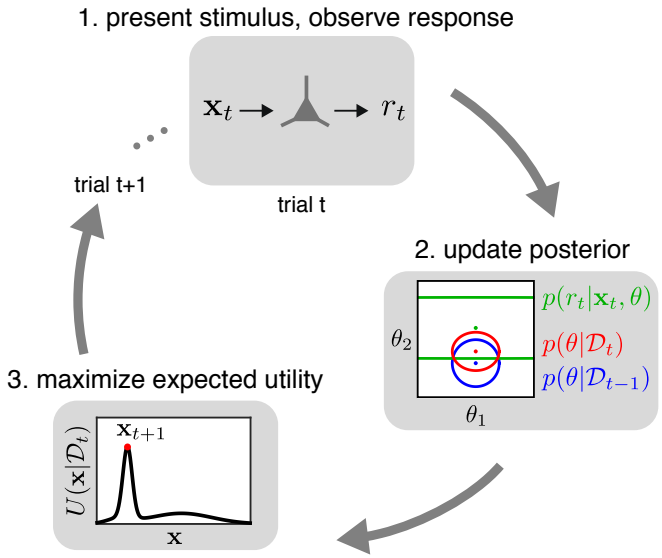

Figure 1: Schematic of Bayesian active learning for closed-loop neurophysiology experiments. At time step t of the experiment, we present stimulus $\mathbf{x}_t$ and record neural response r_t. Then, we then update the posterior $p(\theta|\mathcal{D}_t)$ by combining the likelihood $p(r_t|\mathbf{x}_t, \theta)$ with the prior $p(\theta|\mathcal{D}_{t-1})$, which is the posterior from the previous time step. Finally, we search for the stimulus $\mathbf{x}_{t+1}$ that maximizes the expected utility $U(\mathbf{x}|\mathcal{D}_t)$, which quantifies the learning objective in terms of a utility function integrated over the joint distribution of r and θ given the data. These steps are repeated until some stopping criterion.

selects stimuli that maximize the mutual information between response r and the parameters θ, which is equivalent to minimizing the expected entropy of the posterior distribution.

Formally, infomax learning arises from a utility function given by the log ratio of the posterior to the prior,

$$u(\theta, r, \mathbf{x}|\mathcal{D}_t) = \log \frac{p(\theta|r, \mathbf{x}, \mathcal{D}_t)}{p(\theta|\mathcal{D}_t)}, \tag{6}$$

where the numerator is the updated posterior after observing a new stimulus-response pair $(\mathbf{x}, r)$, and the denominator is the prior, given by the posterior at trial t. The expected utility is therefore the mutual information between r and θ:

$$U_{\text{infomax}}(\mathbf{x}|\mathcal{D}_t) = \mathbb{E}_{r,\theta}\Big[\log \frac{p(\theta|r, \mathbf{x}, \mathcal{D}_t)}{p(\theta|\mathcal{D}_t)}\Big] = H(\theta|\mathcal{D}_t) - H(\theta; r\big|\mathbf{x}, \mathcal{D}_t) \triangleq I(\theta, r|\mathbf{x}, \mathcal{D}_t), \tag{7}$$

where we use $H(\theta; r\big|\mathbf{x}, \mathcal{D}_t)$ to denote the conditional entropy of θ given r for fixed $\mathbf{x}$ and $\mathcal{D}_t$, and $H(\theta|\mathcal{D}_t)$ is the entropy of the posterior after the previous trial. Note that we can perform infomax learning by selecting the stimulus that minimizes $H(\theta; r\big|\mathbf{x}, \mathcal{D}_t)$, since $H(\theta|\mathcal{D}_t)$ is independent of the stimulus and response on the current trial. The mutual information utility function is also commonly referred to as the *expected information gain*, since it is the expected change in the posterior entropy from a single stimulus-response pair [7, 27].

It is worth noting that the mutual information can also be written as

$$I(\theta, r\big|\mathbf{x}, \mathcal{D}_t) = H(r\big|\mathcal{D}_t) - H(r; \theta\big|\mathbf{x}, \mathcal{D}_t), \tag{8}$$

which is the difference between the marginal entropy of r (under the predictive distribution) and the conditional entropy of r given θ. This expression is sometimes easier to compute than the expected utility as given in (eq. 7), as we will show in our first application below.

Minimum mean-squared-error (MMSE)

Another possible approach is to select stimuli that will allow for optimal least-squares estimation of θ, a paradigm commonly known as *minimum mean squared error (MMSE)* learning [12, 32, 33]. We can formalize MMSE learning with a utility function given by the negative mean squared error (since we wish to maximize utility):

$$u = -||\theta - \hat{\theta}_{(r,\mathbf{x},\mathcal{D}_t)})||^2, \tag{9}$$

where $\hat{\theta}_{(r,\mathbf{x},\mathcal{D}_t)} = \mathbb{E}_\theta[\theta|r,\mathbf{x},\mathcal{D}_t]$ is the mean of the posterior $p(\theta|r,\mathbf{x},\mathcal{D}_t)$, also known as the *Bayes' least squares estimate*. The expected utility is therefore given by

$$U_{\text{mmse}}(\mathbf{x}|\mathcal{D}_t) = -\mathbb{E}_{r,\theta}\Big[(\theta - \hat{\theta}_{(r,\mathbf{x},\mathcal{D}_t)})^\top(\theta - \hat{\theta}_{(r,\mathbf{x},\mathcal{D}_t)})\Big] = -\mathbb{E}_r\Big[\mathrm{Tr}\big[\mathrm{cov}\big(\theta|r,\mathbf{x},\mathcal{D}\big)\big]\Big], \tag{10}$$

which is equal to the negative trace of the posterior covariance given the response, averaged over $p(r|\mathbf{x},\mathcal{D}_t)$. Because trace and expectation can be exchanged, this can also be written as the sum of the expected marginal posterior variances:

$$U_{\text{mmse}}(\mathbf{x}|\mathcal{D}_t) = -\sum_i \mathbb{E}_r[\mathrm{var}(\theta_i|r,\mathbf{x},\mathcal{D}_t)]. \tag{11}$$

One might therefore refer to MMSE learning as *minimum-variance* learning since it seeks to minimize the sum of posterior variances for each parameter [12].

Other utility functions

A variety of other utility functions have been proposed in the literature, including prediction error on test data [24, 34]; misclassification error [35]; and the mutual information between function values at tested and untested locations [36]. Other "non-greedy" active learning methods do not attempt to maximize expected utility in each time step, but define optimality in terms of the stimulus that will best allow one to achieve some learning or prediction goal in 2 or n trials after the current trial [37].

For the remainder of this chapter we will focus on infomax learning, due to its popularity and relative computational tractability. But readers should be aware that the choice of utility function can have significant effects on learning, due to the fact that different utility functions imply different notions of "goodness" of a posterior distribution. For example, if we consider an uncorrelated Gaussian posterior, entropy depends on the product of variances, whereas the MSE depends on the sum of variances. Thus, infomax learning would strongly prefer a posterior with variances $(\sigma_1^2 = 1, \sigma_2^2 = 100)$ to one with variances $(\sigma_1^2 = 20, \sigma_2^2 = 20)$ since $(1 \cdot 100) < (20 \cdot 20)$. MMSE learning, however, would have the opposite preference because $(1 + 100) > (20 + 20)$. These differences may have practical effects on learning by determining which stimuli are selected. (See [12] for an detailed exploration in the context of tuning curve estimation).

Uncertainty sampling

Before moving on to applications, it is worth mentioning methods that are not fully Bayesian but rely on some of the basic ingredients of Bayesian active learning. A prominent example is *uncertainty sampling*, which was introduced for training probabilistic classifiers [38]. The original idea was to select stimuli for which the current classifier (given the labelled data available so far) is maximally uncertain. This employs the entirely reasonable heuristic that a classifier should be able to learn more from examples with uncertain labels. However, uncertainty sampling is not necessarily a Bayesian active learning method because the selection rule does not necessarily maximize an expected utility function.

Uncertainty sampling can take at least two different forms in a neurophysiology setting. The first, which we will call *response uncertainty sampling* and involves picking the stimulus for which the response has maximal entropy,

$$\mathbf{x}_{t+1} = \arg\max_{\mathbf{x}^* \in \Omega} H(r|\mathbf{x}^*, \mathcal{D}_t) \tag{12}$$

which involves maximizing entropy of the predictive distribution (eq. 2). For a binary neuron, this would correspond to selecting a stimulus for which the spike probability is closest to 0.5.

An alternative approach is to select stimuli for which we have maximal uncertainty about the function underlying the response, which we will refer to as *parameter uncertainty sampling*. This method can be used when the parameters are in a one-to-one correspondence with the stimulus space. If we consider a tuning curve $f(\mathbf{x})$ that specifies the mean response to stimulus $\mathbf{x}$, parameter uncertainty sampling would correspond to selection rule:

$$\mathbf{x}_{t+1} = \arg\max_{\mathbf{x}^* \in \Omega} H(f(\mathbf{x}^*)|\mathcal{D}_t), \tag{13}$$

where $p(f(\mathbf{x})|\mathcal{D}_t)$is the posterior distribution over the value of the tuning curve at $\mathbf{x}$. This differs from infomax learning because it fails to take into account the *amount* of information that the response r is likely to provide about f. For example, higher-mean regions of the tuning curve will in general provide less information than lower-mean regions under Poisson noise because noise variance grows with the mean. In the next two sections, we will explore active learning methods for two specific applications of interest to neurophysiologists.

3 Application: tuning curve estimation

First, we consider the problem of estimating a neuron's tuning curve in a parametric stimulus space. The object of interest is a nonlinear function f that describes how a neuron's firing rate changes as a function of some stimulus parameters (e.g., orientation, spatial frequency, position). Canonical examples would include orientation tuning curves, spatial frequency tuning curves, speed tuning curves, hippocampal place fields, entorhinal grid cell fields, and absorption spectra in photoreceptors. We might equally call such functions "firing rate maps" or "response surfaces". Our goal is to select stimuli in order to characterize these functions using the smallest number of measurements. We will separately discuss methods for parametric and non-parametric tuning curves under Poisson noise.

3.1 Poisson encoding model

We will model a neuron's average response to a stimulus $\mathbf{x}$ by a nonlinear function $f(\mathbf{x})$, known as the tuning curve, and assume that the response is corrupted by Poisson noise. The resulting encoding model is given by:

$$\lambda = f(\mathbf{x}) \tag{14}$$

$$p(r|\mathbf{x}) = \tfrac{1}{r!}\lambda^r \mathsf{e}^{-\lambda}. \tag{15}$$

Let $\mathcal{D}_t = \{(\mathbf{x}_i, r_i)\}_{i=1}^t$ denote the data collected up to time t in an experiment. Then we have log-likelihood function

$$\mathcal{L}(\boldsymbol{\lambda}_t|\mathcal{D}_t) = \log p(R_t|\boldsymbol{\lambda}_t) = {R_t}^\top \log \boldsymbol{\lambda}_t - \mathbf{1}^\top \boldsymbol{\lambda}_t, \tag{16}$$

where $R_t = (r_1, \ldots, r_t)^\top$ is the vector of responses, $\boldsymbol{\lambda}_t = (f(\mathbf{x}_1), \ldots, f(\mathbf{x}_t))^\top$ is the vector of spike rates for the stimuli presented, and $\mathbf{1}$ is a vector of ones. We have ignored a constant that does not depend on f.

3.2 Parametric tuning curves

In many settings, the experimenter has a particular parametric tuning curve in mind (e.g., a von-Mises function for a V1 orientation tuning curve, or a Naka-Rushton function for a contrast-response function). This approach confers advantages in terms of speed: by making strong assumptions about the form of the tuning curve, active learning algorithms can rule out many functions *a priori* and more quickly identify regions of stimulus space that are informative about the parameters. For example, if the desired tuning curve is a Gaussian bump, Bayesian active learning will not waste time trying to determine if there are other bumps in unexplored regions of parameter space once a single bump has been identified. The potential disadvantage of this approach is that it may fail when tuning curves violate the assumptions of the parametric model. If a neuron has a bimodal tuning curve, for example, an active learning algorithm designed for unimodal function may never discover the second mode.

Here we describe a simple approach for infomax learning of a parametric tuning curve $f(\mathbf{x}; \theta)$, which describes the mean response to a stimulus $\mathbf{x}$ and is described by parameters θ. In general, the log-likelihood (eq. 16) is not convex as a function of θ, and gradient ascent methods may therefore not find the global maximum of the likelihood. We therefore use Markov Chain Monte Carlo (MCMC) sampling to obtain samples from the posterior distribution over θ, an approach used previously for Bayesian tuning curve inference in a fixed design setting [39].

We can use a standard MCMC sampling method (e.g., Metropolis-Hastings or slice sampling) to obtain a set of m samples $\{\theta^{(i)}\} \sim p(\theta|\mathcal{D}_t)$ from the posterior given the data so far in the experiment (eq. 1). We can then evaluate the expected information gain for any candidate stimulus $\mathbf{x}^*$ using a grid over spike counts $r \in \{0, 1, \ldots r_{max}\}$ to compute the marginal and conditional response entropies. We set r_{max} to some suitably high value (e.g., 200 spikes) based on the current posterior over spike rates. Mutual information is given by the difference of these entropies:

$$I(\theta, r\big|\mathbf{x}^*, \mathcal{D}_t) \;\approx\; -\sum_{r=0}^{r_{max}} p(r|\mathbf{x}^*) \log p(r|\mathbf{x}^*) \;+\; \frac{1}{m}\sum_{i=1}^{m}\sum_{r=0}^{r_{max}} p(r|\mathbf{x}^*, \theta^{(i)}) \log p(r|\mathbf{x}^*, \theta^{(i)}), \tag{17}$$

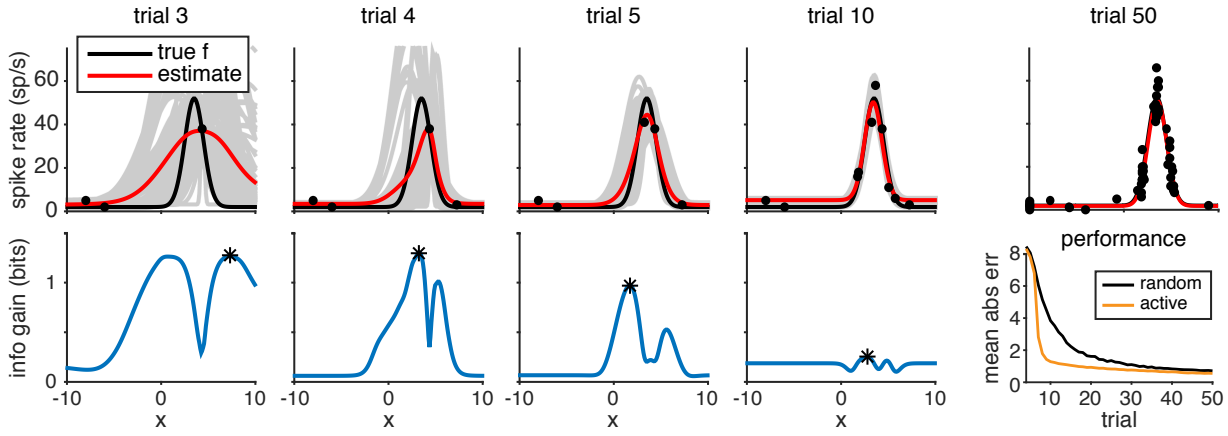

Figure 2: Active learning of 1D parametric tuning curve using MCMC sampling. The true tuning curve (black) has preferred stimulus $\mu = 3.4$, tuning width $\sigma = 1$, amplitude $A = 50$, and baseline firing rate $b = 2$. We placed a uniform prior on each of these parameters: $\mu \in [-10, 10]$, $\sigma \in [0.1, 20]$, $A \in [1, 200]$, and $b \in [0.1, 50]$. **Top row**: True tuning curve (black) and Bayes' least-squares (BLS) estimate (red), shown along with 50 samples from the posterior (gray traces) after 3, 4, 5, 10, and 50 trials. (50-trial figure at right generated from an independent run). **Bottom row**: Expected information gain for each candidate stimulus given the data so far in the experiment. Black asterisk indicates location of the selected stimulus, which is presented on the next trial. **Bottom right**: Comparison of mean absolute error between true tuning curve and BLS estimate under random (black) and Bayesian active learning stimulus election (yellow), averaged over 250 runs of each method. The active method achieves a maximal speedup factor of 2.5, with an error after 10 trials approximately equal to the random sampling error after 25 trials.

where the marginal response distribution is given by the mean over MCMC samples:

$$p(r|\mathbf{x}^*) = \frac{1}{m}\sum_{i=1}^{m} p(r|\mathbf{x}^*, \theta^{(i)}) \tag{18}$$

for each response value r. The resulting algorithm is remarkably simple, and may be implemented with fewer than 150 lines of code in Matlab using a 2D grid over stimulus locations and spike counts r. (Code available from `http://pillowlab.princeton.edu/code_activelearningTCs.html`).

Figure 2 shows an illustration of this algorithm in a simulated experiment for estimating a 1D Gaussian tuning curve with baseline, parametrized as:

$$f(x;\theta) = b + A\exp\Big(-\tfrac{1}{2\sigma^2}(x-\mu)^2\Big) \tag{19}$$

where the parameters θ include a preferred stimulus μ, tuning width σ, amplitude A, and baseline firing rate b. We obtained samples from the posterior with a standard implementation of slice sampling.

The first column of Fig. 2 shows a snapshot of the algorithm after 3 trials of the experiment. The top plot shows the three observed stimulus-response pairs (black dots), the true tuning curve (black trace), and m=100 posterior samples $f^{(i)}$ (gray traces), given by $f(x|\theta^{(i)})$ for each sample $\theta^{(i)}$ from the posterior (eq. 1). The posterior mean $\hat{f}$ (red trace) is the mean of the samples $\frac{1}{m}\sum_i f^{(i)}$. The bottom plot shows the expected information gain, computed using (eq. 17), for each stimulus x on a grid over the stimulus range $[-10, 10]$. Intuitively, the expected information gain for each stimulus is related to the spread of the sample tuning functions at that location (gray traces in the top plots); the more the

sample tuning curves disagree, the higher the information gain from that stimulus. A black asterisk marks the maximum of the expected utility function, which determines the stimulus selected for the next trial. Subsequent columns show analogous snapshots after 4, 5, and 10 trials. The top-right plot shows the estimate after 50 trials, along with the stimuli-response pairs obtained (black dots). Note that the stimuli are far from uniformly distributed, with most dots clustered at the peak and sides of the tuning curve. This indicates that these stimulus locations tend to provide maximal information about the tuning curve under this parametrization.

The bottom right plot (Fig. 2) shows a comparison of the average error in the tuning curve estimate $|f(x) - \hat{f}(x)|$ under infomax learning and non-adaptive learning (uniform *iid* stimulus sampling), averaged over 250 runs of each algorithm. On average, the non-adaptive sampling method requires 25 trials to achieve the same error as infomax learning after only 10 trials. Longer runs reveal that, even asymptotically, the non-adaptive method requires 50% more trials to achieve the same error as infomax learning for this 1D example. Substantially greater improvements can be obtained in higher dimensions.

3.3 Nonparametric tuning curves with Gaussian process priors

In some settings, the experimenter may not wish to make a strong parametric assumption about the form of the tuning curve, and instead prefer an approach that will converge for any possible f. This motivates a non-parametric approach, which allows the number of degrees of freedom in the tuning curve to grow flexibly with the amount of data. Here we discuss an approach based on transformed Gaussian process priors, described previously in [11, 12].

Gaussian processes

Gaussian processes (GPs) provide a flexible and computationally tractable family of prior distributions over smooth functions. They have been used for non-parametric tuning curve estimation [40], and for a variety of other neuroscience applications including spike rate estimation [41], factor analysis [42], and estimation of cortical maps [43]. A Gaussian process can be understood as an extension of a multivariate Gaussian distribution to the continuum, so that each value of the function has a Gaussian distribution. Formally, a GP is characterized by a *mean function* $\mu(\cdot)$ and *covariance function* $K(\cdot,\cdot)$ which specify the mean and covariance of the Gaussian distribution over function values at any collection of locations. For a d-dimensional function $\phi : \mathbb{R}^d \to \mathbb{R}$ distributed according to a GP, we write $\phi \sim \mathcal{GP}(\mu, K)$. This means that for any pair of locations $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^d$, the values of the function at these locations have a bivariate Gaussian distribution:

$$\begin{bmatrix}\phi(\mathbf{x}_1)\\ \phi(\mathbf{x}_2)\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}\mu(\mathbf{x}_1)\\ \mu(\mathbf{x}_2)\end{bmatrix}, \begin{bmatrix}K(\mathbf{x}_1,\mathbf{x}_1) & K(\mathbf{x}_1,\mathbf{x}_2)\\ K(\mathbf{x}_1,\mathbf{x}_2) & K(\mathbf{x}_2,\mathbf{x}_2)\end{bmatrix}\right). \tag{20}$$

Similarly, for any set of N input locations $\mathbf{x}_{1:N} = \{\mathbf{x}_i \in \mathbb{R}^d\}_{i=1}^N$, the vector of function values $\boldsymbol{\phi}_{1:N} = \{\phi(\mathbf{x}_i) \in \mathbb{R}\}_{i=1}^N$ has a multivariate Gaussian distribution with mean vector whose i'th element is $\mu(\mathbf{x}_i)$ and covariance matrix whose i,j'th element is $K(\mathbf{x}_i, \mathbf{x}_j)$. A common approach is to fix the mean function to a constant μ and select a covariance function that has desired degree of smoothness or

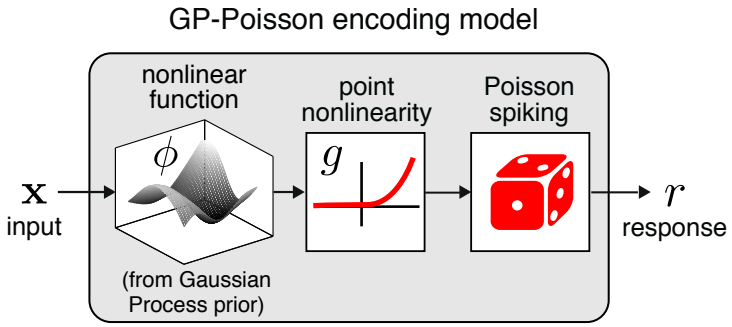

Figure 3: Schematic of GP-Poisson encoding model for tuning curves. A function ϕ takes vector stimulus $\mathbf{x}$ as input and produces scalar output, which is transformed by a nonlinear function g into a positive spike rate. The response r is a Poisson random variable with mean $g(\phi(\mathbf{x}))$. We place a Gaussian process (GP) prior over ϕ and assume g is fixed. The neuron's tuning curve or firing rate map is given by $f(\mathbf{x}) = g(\phi(\mathbf{x}))$.

differentiability (see [44]). Here we will use the popular Gaussian or "squared exponential" covariance function (inaptly named because it takes the form of an exponentiated square, not a squared exponential), which is given by:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \rho \exp\left(-||\mathbf{x}_i - \mathbf{x}_j||^2/(2\tau^2)\right). \tag{21}$$

This covariance function has a pair of hyperparameters: a marginal variance ρ, which determines how far function values deviate from its mean; and length-scale τ, which determines smoothness. We can consider the mean μ as a third hyperparameter, so that the GP is determined by the hyperparameter vector $\psi = (\rho, \tau, \mu)$.

If we wished to model a neuron's response as having fixed variance Gaussian noise, then we could place a GP prior directly over the tuning curve f, and use the standard formulas to compute the posterior over f after each trial [44]. In this case, there's no need for adaptive stimulus selection: it turns out that the posterior covariance over f and the expected utility depend only on the stimuli $\{\mathbf{x}_i\}$, and are independent of the observed responses $\{r_i\}$. This means that we can plan out a maximally informative set of stimuli before the experiment begins. (Note however that this is true only if we consider the GP hyperparameters ψ and Gaussian noise variance σ^2 to be fixed; if they are to be estimated or integrated over during the experiment, then active learning becomes worthwhile even in this setting.)

Transformed Gaussian processes

If we wish to model neurons with Poisson noise, tuning curves must be non-negative. This rules out the use of standard Gaussian process priors because they place probability mass on negative as well as positive function values. A straightforward solution is to transform a Gaussian process by a nonlinear function g with non-negative range. This suggests we parametrize the tuning curve as

$$f(\mathbf{x}) = g(\phi(\mathbf{x})), \tag{22}$$

where $g(\cdot)$ is an invertible nonlinear function with non-negative output, and ϕ is a real-valued function governed by a GP prior. We refer to the resulting model as the GP-Poisson model (Fig. 3).

This model lends itself to infomax learning because information gain about f is the same as the information gain about ϕ. An invertible nonlinearity g can neither create nor destroy information. Moreover, if g is convex and log-concave (meaning g grows at least linearly and at most exponentially), then the posterior over ϕ under a GP prior will be strictly log-concave (by an argument similar to that given in [45]). This ensures that the posterior over ϕ, written $p(\phi|\mathcal{D}_t)$, has a single mode and can be reasonably well approximated by a Gaussian process [12].

Posterior Updating

We can approximate the posterior over ϕ using the Laplace approximation, a Gaussian approximation that results from finding the posterior mode and using the Hessian (second derivative matrix) at the mode to approximate the covariance [46, 47]. Given data $\mathcal{D}_t$, we find the *maximum a posteriori* (MAP) estimate of ϕ at the set of stimulus locations presented so far in the experiment:

$$\hat{\boldsymbol{\phi}}_{map} \triangleq \arg\max_{\boldsymbol{\phi}_t} \quad \mathcal{L}(\boldsymbol{\phi}_t|\mathcal{D}_t) \quad + \quad \log p(\boldsymbol{\phi}_t|\mu\mathbf{1}, \mathbf{K}) \tag{23}$$

$$= \arg\max_{\boldsymbol{\phi}_t} \quad {R_t}^\top \log(g(\boldsymbol{\phi}_t)) - \mathbf{1}^\top g(\boldsymbol{\phi}_t) \quad - \quad \tfrac{1}{2}(\boldsymbol{\phi}_t - \mu\mathbf{1})^\top \mathbf{K}^{-1}(\boldsymbol{\phi}_t - \mu\mathbf{1}), \tag{24}$$

where $\mathcal{L}(\boldsymbol{\phi}_t|\mathcal{D}_t)$ denotes the Poisson log-likelihood (eq. 16) and $\log p(\boldsymbol{\phi}_t|\mu, \mathbf{K})$ is the log probability of the vector of function values $\boldsymbol{\phi}_t = (\phi(\mathbf{x}_1), \ldots \phi(\mathbf{x}_t))$ under the GP prior. This latter term is given by a multivariate normal density with $t \times t$ covariance matrix $\mathbf{K}_{i,j} = K(\mathbf{x}_i, \mathbf{x}_j)$, and $\mathbf{1}$ is a length-t vector of ones.

The Laplace approximation to the posterior at the stimulus locations $X_t = (\mathbf{x}_1, \ldots, \mathbf{x}_t)$ is a multivariate Gaussian with mean $\hat{\boldsymbol{\phi}}_{map}$ and covariance equal to the negative inverse Hessian of the log-posterior:

$$p(\boldsymbol{\phi}_t|\mathcal{D}_t) \approx \mathcal{N}(\hat{\boldsymbol{\phi}}_{map}, \Sigma), \qquad \Sigma = (L + \mathbf{K}^{-1})^{-1} \tag{25}$$

where L is the Hessian of the negative log-likelihood, which is a diagonal matrix with diagonal elements given by

$$L_{ii} = -\frac{\partial^2}{\partial \phi_i^2} \mathcal{L}(\boldsymbol{\phi}_t|\mathcal{D}_t) = r_i \frac{g(\phi_i)g''(\phi_i) - g'(\phi_i)^2}{g(\phi_i)^2} + g''(\phi_i) \tag{26}$$

evaluated at $\boldsymbol{\phi}_t = \hat{\boldsymbol{\phi}}_{map}$. When $g(\cdot)$ is exponential, the first term vanishes and we have simply $L = \mathrm{diag}(\exp(\boldsymbol{\phi}_t))$. The full posterior GP is then determined by the Gaussian approximation at the points in X_t: the likelihood enters only at these points, and its effects on the posterior at all other points are mediated entirely by the smoothing properties of the GP prior.

For any set of possible next stimuli $\mathbf{x}^* = \{\mathbf{x}_i^*\}_{i=1}^N$, let $\boldsymbol{\phi}_t^* = \phi(\mathbf{x}^*)$ denote the vector of associated function values. The posterior distribution over these function values under the Laplace approximation is given by:

$$p(\boldsymbol{\phi}_t^*|\mathcal{D}_t) \approx \mathcal{N}(\boldsymbol{\mu}_t, \Lambda_t), \tag{27}$$

with mean and covariance given by

$$\boldsymbol{\mu}_t = \mu\mathbf{1} + \mathbf{K}^*\mathbf{K}^{-1}(\hat{\boldsymbol{\phi}}_{map} - \mu\mathbf{1}) \tag{28}$$

$$\Lambda_t = \mathbf{K}^{**} - \mathbf{K}^*(L^{-1} + \mathbf{K})^{-1}\mathbf{K}^{*\top}, \tag{29}$$

where $\mathbf{K}^*_{ij} = K(\mathbf{x}^*_i, \mathbf{x}_j)$, and $\mathbf{K}^{**}_{ij} = K(\mathbf{x}^*_i, \mathbf{x}^*_j)$ (see [44], Sec. 3.4.2). To perform uncertainty sampling (eq. 13), we could simply select the stimulus for which the corresponding diagonal element of Λ_t is largest. However, this will not take account of the fact that the expected information gain under Poisson noise depends on the posterior mean as well as the variance, as we will see next.

Infomax learning

Because mutual information is not altered by invertible transformations, the information gain about the tuning curve f is the same as about ϕ. We can therefore perform infomax learning by maximizing mutual information between r and $\boldsymbol{\phi}_t$ or, equivalently, minimize the conditional entropy of $\boldsymbol{\phi}_t$ given r (eq. 7). Because $\boldsymbol{\phi}_t$ is a function instead of a vector, this entropy is not formally well defined, but for practical purposes we can consider the vector of function values $\boldsymbol{\phi}^*_t$ defined on some grid of points $\mathbf{x}^*$, which has a tractable, approximately Gaussian distribution (eq. 27).

The expected utility of a stimulus $\mathbf{x}^*$ can therefore be computed using the formula for negative entropy of a Gaussian:

$$U_{\textit{infomax}}(\mathbf{x}^*|\mathcal{D}_t) = -\tfrac{1}{2}\mathbb{E}_{r*|\mathbf{x}}\Big[\log\left|\Lambda_t^{-1}\Lambda^*_t\right|\Big], \tag{30}$$

where Λ_t is the posterior covariance at time t (eq. 29) and Λ^*_t is the posterior covariance after updating with observation $(\mathbf{x}^*, r^*)$. The expectation is taken with respect to $p(r^*|\mathbf{x}^*, \mathcal{D}_t)$, the predictive distribution at $\mathbf{x}^*$ (eq. 2). If we use exponential g, then the Λ^*_t lacks explicit dependence on r^* because the Hessian depends only on the value of ϕ at $\mathbf{x}^*$, that is:

$$\Lambda^*_t = \left(\Lambda_t^{-1} + \delta_{(\mathbf{x}^*,\mathbf{x}^*)}\, e^{\hat{\boldsymbol{\phi}}_{map}(\mathbf{x}^*)}\right)^{-1}, \tag{31}$$

where $\delta_{(\mathbf{x}^*,\mathbf{x}^*)}$ is a matrix with a single 1 in the diagonal position corresponding to stimulus $\mathbf{x}^*$ and zeros elsewhere, and $\hat{\boldsymbol{\phi}}_{map}(\mathbf{x}^*)$ is the MAP estimate of ϕ after observing r^*. Using the matrix-determinant lemma to compute the determinant of Λ^*_t, we can simplify the expected utility to:

$$U_{\textit{infomax}}(\mathbf{x}^*|\mathcal{D}_t) = \tfrac{1}{2}\mathbb{E}_{r^*|\mathbf{x}^*}\left[\log\left(1 + \sigma^2_t(\mathbf{x}^*)e^{\hat{\boldsymbol{\phi}}_{map}(\mathbf{x}^*)}\right)\right] \approx \tfrac{1}{2}\mathbb{E}_{r^*|\mathbf{x}^*}\left[\sigma^2_t(\mathbf{x}^*)e^{\hat{\boldsymbol{\phi}}_{map}(\mathbf{x}^*)}\right], \tag{32}$$

where $\sigma^2_t(\mathbf{x}^*) = \Lambda_t(\mathbf{x}^*, \mathbf{x}^*)$ is the marginal variance of the posterior over the value of ϕ at $\mathbf{x}^*$ (eq. 29), and the approximation on the right results from a first-order Taylor expansion of $\log(x)$. The information gain therefore depends on r^* only via its influence on the MAP estimate of $\phi(\mathbf{x}^*)$. Because there is no analytical form for this expectation, it is reasonable to use the expectation of $e^{\hat{\boldsymbol{\phi}}_{map}(\mathbf{x}^*)}$ over the current posterior, which follows from the formula for the mean of a log-normal distribution. This allows us to evaluate the information gain for each candidate stimulus $\mathbf{x}^*$:

$$U_{\textit{infomax}}(\mathbf{x}^*|\mathcal{D}_t) \approx \tfrac{1}{2}\sigma^2_t(\mathbf{x}^*)e^{\mu_t(\mathbf{x}^*)+\frac{1}{2}\sigma^2_t(\mathbf{x}^*)}. \tag{33}$$

We take the next stimulus $\mathbf{x}_{t+1}$ to be the maximizer of this function, which depends on both the mean and variance of the posterior over ϕ. (Note that this differs from uncertainty sampling, which uses only the posterior variance to select stimuli).

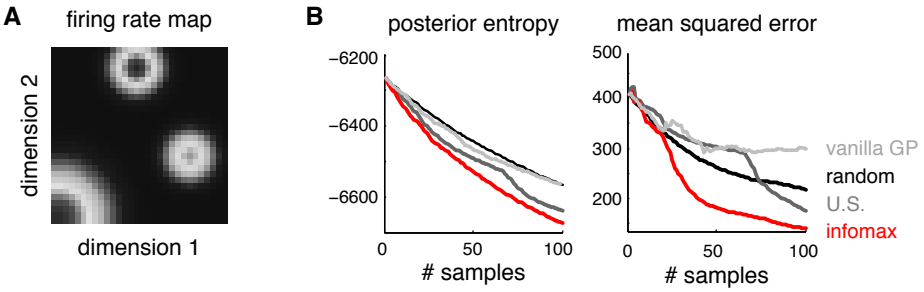

Figure 4: Comparison of stimulus selection methods for 2D tuning curve. **A**: True tuning curve in a 2D input space, with maximum of 90 sp/s and minimum of 1 sp/s. **B** The posterior entropy (left) and mean squared error (right), as a function of the number of experimental trials, for each of four methods: (1) Stimulus selection under a standard or "vanilla" Gaussian process model with Gaussian noise (where stimulus selection does not depend on the observed responses); (2) random *iid* stimulus selection; (3) uncertainty sampling; and (4) infomax learning. Infomax learning exhibits the best performance in terms of both information and MSE.

The form of the mean dependence observed in (eq. 33) depends critically on the choice of the nonlinear transformation g. For exponential g, as assumed here, the information gain is an increasing function of the mean, meaning that the algorithm will explore high-firing rate regions of the tuning curve first. For a soft-rectifying g, however, information gain turns out to be a *decreasing* function of the posterior mean [12], meaning the algorithm prefers to explore moderate-to-low firing-rate regions of the tuning curve first. See [12] for a more thorough discussion of this issue, and along with other choices for expected utility.

Simulations

Fig. 4 shows a comparison of several stimulus selection methods using data simulated from a Poisson neuron with a two-dimensional tuning curve. We compared infomax stimulus selection to random sampling and uncertainty sampling, which selects the stimulus for which the posterior variance of firing rate map is maximal. We also tested the performance of a method based on a model with Gaussian (instead of Poisson) noise, for which information gain can be computed analytically. For this model, the utility does not depend on the observations, meaning that the entire sequence of stimuli can be planned out before the experiment. Fig. 4B shows the performance of each method in terms of posterior entropy and mean squared error (average over 100 independent repetitions). Uncertainty sampling, in this case, focuses on picking stimuli only around the high peak areas of the true map, which results in slower decrease in MSE than random sampling.

4 Application: linear receptive field estimation

Another important class of problems in electrophysiology experiments is linear receptive field (RF) estimation. This differs from tuning curve estimation in that the estimation problem is high dimensional, since RF dimensionality is equal to the number spatio-temporal elements or "pixels" in the relevant stimulus driving the neuron. This high-dimensional characterization problem is simplified, however,

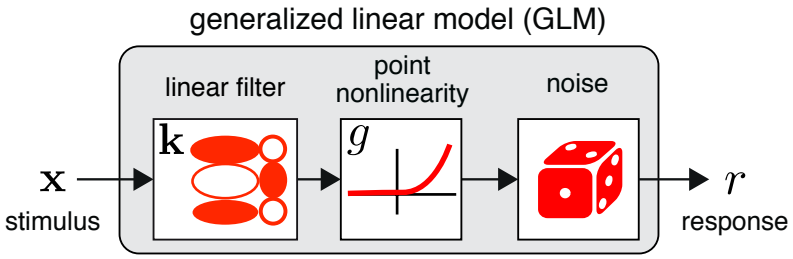

Figure 5: Schematic of generalized linear encoding model (GLM) for RF characterization. The model contains a weight vector $\mathbf{k}$ that linearly filters the stimulus, a nonlinear function g, and noise from an exponential family distribution.

by the fact that the assumed function is linear. This assumption is justified by the fact that many sensory neurons have approximately linear response properties in a suitably defined input space (which may involve a nonlinear transformation of the raw stimulus [48]). The active learning problem is to select stimuli in this high-dimensional space that are maximally informative about the neuron's weighting function. This poses unique challenges because, unlike the tuning curve problem, we cannot grid up the input space and compute the expected utility for each stimulus.

4.1 Generalized linear model

A popular model for linear receptive fields (RFs) arises from the generalized linear model (GLM) [49–51]. This model (Fig. 5) consists of a linear filter $\mathbf{k}$, which describes how the neuron integrates the stimulus over space and time, followed by a point nonlinearity $g(\cdot)$, which transforms filter output into the neuron's response range, and exponential-family noise that captures stochasticity in the response. Typically, one regards the nonlinearity in a GLM as fixed, which simplifies estimation and stimulus selection problems [45].

The Poisson GLM, also known as the linear-nonlinear-Poisson (LNP) model, is given by

$$\lambda = g(\mathbf{k}^\top \mathbf{x}), \qquad r|\mathbf{x} \sim \mathrm{Poiss}(\Delta\lambda), \tag{34}$$

where $\mathbf{k}^\top \mathbf{x}$ is the dot product between the filter $\mathbf{k}$ and the stimulus $\mathbf{x}$, the nonlinearity g ensures the spike rate λ is non-negative, and Δ is a time bin size. The model can be extended to incorporate linear dependencies on spike-history and other covariates like the responses from other neurons [50–53], but we will focus here on the simple case where the stimulus $\mathbf{x}$ is the only input.

4.2 Infomax stimulus selection for Poisson GLM

Lewi *et al* [7] developed an infomax learning method for RF characterization under a Poisson GLM, which we will refer to as "Lewi-09". This method assumes an isotropic Gaussian prior over $\mathbf{k}$, which leads to a posterior formed by the product of a Gaussian prior and Poisson likelihood, just as in the GP-Poisson tuning curve model considered in the previous section. We will omit details of the derivation, but the method developed in [7] is closely related to the infomax learning for the GP-Poisson model (and in

fact, was a direct source of inspiration for our work). In brief, the Lewi-09 method performs approximate MAP inference for $\mathbf{k}$ after each response and computes a Gaussian "Laplace" approximation to the posterior. When the nonlinearity g is exponential, this leads to a concise formula for the expected information gain for a candidate stimulus $\mathbf{x}^*$ (which closely resembles eq. 33):

$$I(r, \mathbf{k}|\mathbf{x}^*, \mathcal{D}_t) \approx \tfrac{1}{2}\sigma_\rho^2 e^{\mu_\rho + \frac{1}{2}\sigma_\rho^2}, \tag{35}$$

where μ_ρ and σ_ρ^2 are projections of $\mathbf{x}^*$ onto the posterior mean and covariance of $\mathbf{k}$:

$$\mu_\rho = \boldsymbol{\mu}_t{}^\top \mathbf{x}^*, \quad \sigma_\rho^2 = \mathbf{x}^{*\top} \Lambda_t \mathbf{x}^*. \tag{36}$$

(See eqs. 4.14 and 4.20 in [7] for details). An interesting consequence of this formula is that, although the stimulus space is high-dimensional, the informativeness of a candidate stimulus depends only its linear projection onto the current posterior mean and covariance matrix.

The key contributions of [7] include fast methods for updating the posterior mean and covariance, and efficient methods for selecting maximally informative stimuli subject to a "power" constraint (that is, $||\mathbf{x}^*||^2 < const$). The Lewi-09 algorithm yields substantial improvements relative to randomized *iid* (i.e., "white noise") stimulus selection. Intriguingly, the authors show that in high-dimensional settings, two methods that might have been expected to work well in fact do not: (1) the maximal eigenvector of the posterior covariance matrix; and (2) the most informative stimulus from a limited set of *iid*-sampled stimuli. Both of these methods turn out to perform approximately as poorly as random stimulus selection.

4.3 Infomax for hierarchical RF models

One shortcoming of the Lewi-09 method is it does not exploit prior information about the structure of neural RFs. It uses an isotropic Gaussian prior, also known as the *ridge* prior,

$$\mathbf{k} \sim \mathcal{N}(0, \rho I), \tag{37}$$

where ρ is the common prior variance of all RF coefficients. This assumes the RF elements are *a priori* independent, with prior probability mass spread out equally in all directions (see Fig. 6). By contrast, we know that RFs tend to be structured, e.g., smooth and sparse in time and space. An active learning method that incorporates such structure can concentrate prior probability mass closer to the manifold of likely RF shapes and spend less time exploring regions of stimulus space that are unlikely to be informative about the RF.

To put this in Bayesian terms, the goal of infomax learning is to obtain a posterior distribution with minimal entropy. This can be achieved by either: (1) selecting informative stimuli, i.e., stimuli that lead to maximal narrowing via new likelihood terms; or (2) using a prior with minimal entropy, so that less learning is required in the first place. Although this second point might seem trivial or uninteresting, we will show that it is not, as priors that encourage the forms of structure found in biological systems are not straightforward to construct, and are more difficult to use for active learning than Gaussian priors. Here we will discuss a method that uses this second strategy. Our method uses a hierarchical prior, formulated as a covariance mixture of Gaussians, that flexibly encodes statistical regularities in RF

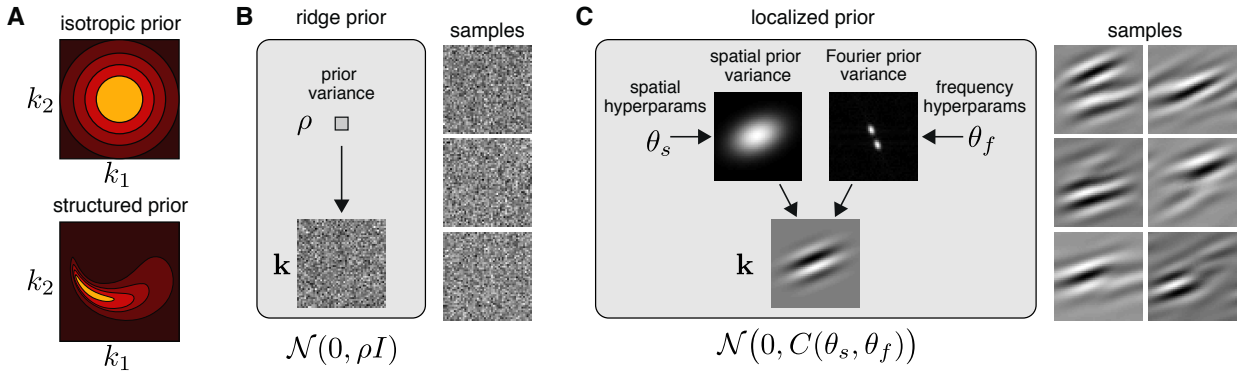

Figure 6: Effects of the prior distributions on active learning of RFs. **(A)** An isotropic Gaussian prior (top) has comparatively large prior entropy, and will thus require a lot of data to achieve a concentrated posterior. A "structured" prior (below), however, concentrates prior probability mass closer to the region of likely RF shapes, so less data is required to concentrate the posterior if the true RF lies close to this region. **(B)** Graphical model for ridge regression prior, which models all RF coefficients as *iid* Gaussian. RF samples from this prior (right) are simply Gaussian white noise. **(C)** Graphical model for localized prior from [54]. The prior simultaneously encourages localized support in space (left) and in the Fourier domain (right), with support controlled by hyperparameters θ_s and θ_f for each domain, respectively. The support depicted in this diagram assigns high prior probability to a Gabor filter (bottom), and samples from the prior conditioned on these hyperparameters exhibit similar spatial location, frequency content, and orientation (right). The full hierarchical prior consists of a mixture of these conditional distributions for hyperparameters covering all locations, frequencies, and orientations, and includes the ridge prior as a special case.

structure in a way that speeds up learning when such structure is present (e.g., smoothness, sparsity, locality), but defaults to an uninformative prior when it is not.

For simplicity, we use a linear-Gaussian encoding model, which can be viewed as a GLM with "identity" nonlinearity and Gaussian noise:

$$r = \mathbf{k}^\top \mathbf{x} + \epsilon, \qquad \epsilon \sim \mathcal{N}(0, \sigma^2), \tag{38}$$

where ϵ is zero-mean Gaussian noise with variance σ^2. While this provides a less accurate statistical model of spike responses than the Poisson GLM used in [7], it simplifies the problem of computing and optimizing the posterior expectations needed for active learning.

We consider a "structured", hierarchical, conditionally Gaussian prior of the form:

$$\mathbf{k} \mid \theta \sim \mathcal{N}(0, C_\theta), \qquad \theta \sim p_\theta, \tag{39}$$

where C_θ is a prior covariance matrix that depends on hyperparameters θ, and p_θ is a hyper-prior over θ. The effective prior over $\mathbf{k}$ is a mixture-of-Gaussians, also known as a *covariance mixture of Gaussians* because the component distributions are zero-mean Gaussians with difference covariances:

$$p(\mathbf{k}) = \int p(\mathbf{k}|\theta) p(\theta) d\theta = \int \mathcal{N}(0, C_\theta)\, p_\theta(\theta) d\theta. \tag{40}$$

Posterior distribution

For this model, the posterior distribution is also a mixture of Gaussians:

$$p(\mathbf{k}|\mathcal{D}_t) = \int p(\mathbf{k}|\mathcal{D}_t, \theta\, p(\theta|\mathcal{D}_t) d\theta = \int \mathcal{N}(\mu_\theta, \Lambda_\theta)\, p(\theta|X, Y) d\theta. \tag{41}$$

with conditional mean and covariance

$$\mu_\theta = \tfrac{1}{\sigma^2}\Lambda_\theta X^\top Y, \quad \Lambda_\theta = (\tfrac{1}{\sigma^2}X^\top X + C_\theta^{-1})^{-1}, \tag{42}$$

and mixing distribution given by the marginal posterior:

$$p(\theta|\mathcal{D}_t) \propto p(R_t|X_t, \theta)\, p_\theta(\theta). \tag{43}$$

The marginal posterior is proportional to the product of the conditional marginal likelihood or "evidence" $p(R_t|X_t, \theta)$ and the hyper-prior $p_\theta(\theta)$. For the linear-Gaussian encoding model, the conditional evidence has an analytic form:

$$p(R_t|X_t, \theta) = (\sqrt{2\pi}\sigma)^{-t}\left|\Lambda_\theta\right|^{\frac{1}{2}}\left|C_\theta\right|^{-\frac{1}{2}} \exp\left[\tfrac{1}{2}\left(\mu_\theta^\top \Lambda_\theta^{-1}\mu_\theta - m^\top L^{-1} m\right)\right], \tag{44}$$
$$L = \sigma^2({X_t}^\top X_t)^{-1}, \quad m = \frac{1}{\sigma^2} L {X_t}^\top R_t,$$

where $X = [\mathbf{x}_1, \ldots, \mathbf{x}_t]^\top$ is the $t \times d$ design matrix and $R = (r_1, \ldots, r_t)^\top$is the vector of observed responses.

As mentioned earlier, active learning confers no benefits over non-adaptive stimulus selection when the response is governed by a linear-Gaussian model and a Gaussian prior, due to the fact that the posterior covariance is response-independent (eq. 42). However, this response-independence does not hold for models with mixture-of-Gaussian priors, as the response R_t affects the posterior via the conditional evidence (eq. 43). Intuitively, as the responses come in, the conditional evidence tells us what hyperparameter settings (i.e., which prior distributions) are best. As we will show, active learning under a structured hierarchical prior can confer substantial advantages over methods based on an isotropic prior.

Localized RF prior

We illustrate this approach using a flexible prior designed to capture *localized* structure in neural receptive fields [9, 54]. The prior, introduced as part of an empirical Bayesian RF estimation method called *automatic locality determination (ALD)*, seeks to exploit the observation that neural RFs tend to be localized in both space-time and spatio-temporal frequency. Locality in space-time means that neurons typically integrate sensory input over a limited region of space and time. Locality in frequency, on the other hand, means that neurons tend to respond to a restricted range of frequencies and orientations, or equivalently, that the Fourier transform of a neural RF tends to be band-limited.

The ALD prior captures localized structure using a covariance matrix $C(\theta_s, \theta_f)$ controlled by two sets of hyperparameters: (1) spatial hyperparameters θ_s, which define an elliptical region of space-time

where the RF has non-zero prior variance; and (2) frequency hyperparameters θ_f, which define a pair of elliptical regions in Fourier space where the Fourier transform of the RF has non-zero prior variance. The full covariance matrix C is given by a product of diagonal covariance matrices with a Fourier change-of-basis operator sandwiched between them [54]. Figure 6C shows a graphical illustration of this prior for one setting of the hyperparameters. The spatial support (left) covers a large region in the middle of a two-dimensional stimulus image, while the frequency support (right) includes two small regions with a particular orientation in the 2D Fourier plane, giving rise to a small range of oriented, smooth, band-pass RFs. Six samples from the conditional prior $p(\mathbf{k}|\theta_s, \theta_f)$ all exhibit this common tendency (Fig. 6C, right).

The full prior distribution $p(\mathbf{k})$ is a continuous mixture of Gaussians (eq. 40), where each mixing component is a zero-mean Gaussian with covariance $C(\theta_s, \theta_f)$, and the mixing distribution $p(\theta_s, \theta_f)$ assigns prior probability to a broad range of possible hyperparameter values, including settings of θ_s and θ_f for which there is no localized structure in either space-time or frequency. The prior does not rule out any RFs *a priori* because it includes the ridge prior as one of its components (Fig. 6B). However, many of the mixing components *do* have restricted spatial or Fourier support, meaning that probability mass is more concentrated on the manifold of likely RFs (as depicted schematically in Fig. 6A). The key virtue of this prior in the context of active learning is that, for neurons with localized, smooth, oriented, band-pass, or other relevant structure, the support of the marginal posterior $p(\theta|\mathcal{D}_t)$ shrinks down relatively quickly to a restricted region of hyperparameter space, ruling out vast swaths of the underlying parameter space. This allows the posterior entropy over $\mathbf{k}$ to shrink far more quickly than could be achieved under an isotropic prior.

Active learning with localized priors

To develop a practical method for active learning under the localized prior, we still need two ingredients: an efficient way to update the posterior distribution $p(\mathbf{k}|\mathcal{D}_t)$ after each trial, and a tractable method for computing and maximizing the expected information gain $I(r, \mathbf{k}|\mathbf{x}^*, \mathcal{D}_t)$. The posterior distribution contains an intractable integral over the hyperparameters (eq. 41) and lacks the log-concavity property that motivated the Gaussian approximation-based methods developed in [7]. However, we can instead exploit the conditionally Gaussian structure of the posterior to develop a fully Bayesian approach using Markov Chain Monte Carlo (MCMC) sampling. We approximate the posterior using a set of hyperparameter samples $\{\theta^{(i)}\}$, $i \in \{1, \ldots, m\}$, drawn from the marginal posterior $p(\theta|\mathcal{D}_t)$ (eq. 43) via a standard MCMC sampling technique. Here each sample represents a full set of space-time and frequency hyperparameters for the localized prior, $\theta^{(i)} = \{\theta_s^{(i)}, \theta_f^{(i)}\}$. The posterior can then be approximated as:

$$p(\mathbf{k}|\mathcal{D}_t) \;\approx\; \frac{1}{m}\sum_i p(\mathbf{k}|\mathcal{D}_t, \theta^{(i)}) \;=\; \frac{1}{m}\sum_i \mathcal{N}(\mu^{(i)}, \Lambda^{(i)}) \tag{45}$$

with mean and covariance of each Gaussian as given in (eq. 42). To update the posterior rapidly after each trial, we use a version of the *resample-move particle filter*, which involves resampling a full set of "particles" $\{\theta^{(i)}\}$ using the new data from each trial and then performing a small number of additional MCMC steps [55]. The main computational bottleneck is the cost of updating the conditional posterior mean $\mu^{(i)}$ and covariance $\Lambda^{(i)}$ for each particle $\theta^{(i)}$, which requires inverting of a $d \times d$ matrix. However,

this cost is independent of the amount of data, and particle updates can be performed efficiently in parallel, since the particles do not interact except for the evaluation of mutual information. Full details are provided in [9].

We can select the most informative stimulus for the next trial using a moment-based approximation to the posterior. Although the entropy of a mixture-of-Gaussians has no tractable analytic form, the marginal mean and covariance of the (sample-based) posterior is given by:

$$\tilde{\mu}_t = \frac{1}{m}\sum \mu_t^{(i)}, \qquad \tilde{\Lambda}_t = \frac{1}{m}\sum_{i=1}^{m}\left(\Lambda_t^{(i)} + \mu_t^{(i)}\mu_t^{(i)\top}\right) - \tilde{\mu}_t\tilde{\mu}_t{}^{\top}. \tag{46}$$

The entropy of the posterior is upper-bounded by $\frac{1}{2}|2\pi e\tilde{\Lambda}_t|$ due to the fact that Gaussians are maximum-entropy distributions for given covariance. We select the next stimulus to be proportional to the maximum-variance eigenvector of $\tilde{\Lambda}_t$, which is the most informative stimulus for a linear Gaussian model and a power constraint [7]. This selection criterion is not guaranteed to maximize mutual information under the true posterior, but it has the heuristic justification that it selects stimulus directions for which the posterior has maximal variance, making it a form of parameter uncertainty sampling.

It is worth noting from (eq. 46) that directions of large posterior variance can arise in at least two different ways: (1) they can be directions of large variance for all conditional covariances $\Lambda^{(i)}$, meaning that all hyperparameter samples assign high posterior uncertainty over the component of $\mathbf{k}$ in this direction of stimulus space, or (2) they can be directions in which the conditional means $\mu^{(i)}$ are highly dispersed, meaning the conditional posteriors from different hyperparameter samples disagree about the mean of $\mathbf{k}$ along this direction. In either scenario, it seems intuitively reasonable that presenting a stimulus along this direction will reduce variance in the marginal posterior, but a more careful theoretical treatment is certainly warranted. In practice, we find that the method performs well both for simulated data from a Poisson GLM and for neural data from primate V1.

4.4 Comparison of methods for RF estimation

Simulated Data

We compared our method and the Lewi-09 method using data simulated from a Poisson-GLM with exponential nonlinearity and Gabor-filter RF (20×20 pixels, shown in Fig. 7). This is the encoding model assumed by the Lewi-09 method, whereas our method (which assumes linear-Gaussian encoding) exhibits model mismatch. For the Lewi-09 method, we used a ridge prior over $\mathbf{k}$, with prior variance set by maximizing marginal likelihood for a small dataset. We tested implementations with two different numbers of particles (10 and 100) to examine the tradeoff between computational complexity and accuracy, and used the angular difference (in degrees) between the true and estimated RF as a performance measure.

Fig 7A shows estimation error as a function of the number of trials. The localized-prior estimate exhibits faster reduction in error. Moreover, the localized algorithm performed better with 100 than with 10 particles (ALD_{100} vs. ALD_{10}), suggesting that accurately preserving uncertainty over the hyperparameters aided performance. Fig 7B shows the estimates obtained by each method after a different number of

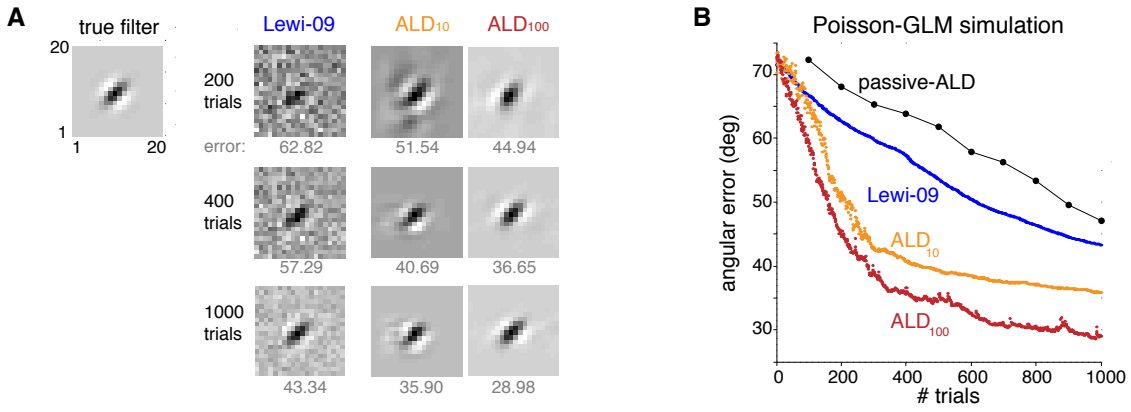

Figure 7: Simulated RF estimation experiment using a Poisson neuron with a Gabor filter RF and exponential nonlinearity. **(A)** Left: true RF is a 20×20 pixel Gabor filter, yielding a 400-dimensional stimulus space. Right: RF estimates obtained by Lewi-09 method (blue) and hierarchical model with localized ("ALD") prior using 10 (orange) or 100 (red) samples to represent the posterior, with posterior mean estimate shown after 200, 400, and 1000 trials of active learning. Grey numbers below indicate angular error in estimate (deg). **(B)** Angular error vs. number of stimuli for three active learning methods, along with ALD inference using "passive" random *iid* stimulus selection (black). Traces show average error over 20 repetitions.

trials. Notice that the estimate with 100 hyperparameter particles after 200 trials is almost identical to the true filter, where the number of trials (200) is substantially lower than the dimensionality of the filter itself ($d = 400$).

Application to Neural Data

We also compared methods using an off-line analysis of real neural data ([56]) from a simple cell in primate V1. The stimuli were 1D white noise "flickering bars", with 16 spatial bars aligned with the cell's preferred orientation. We used 16 time bins to model the temporal RF, resulting in an a 256-dimensional stimulus space. We performed the off-line active learning by extracting the stimuli from the entire 46 minutes of recording of the experimental data. We computed the expected information gain on each trial from presenting each of the remaining stimuli, without access to the neuron's actual response. We tested our ALD-based active learning with 10 hyperparameter samples, and selected the most informative stimuli from $\approx$ 276,000 possible stimuli on each trial.

Fig 8B shows the average angular difference between maximum likelihood estimate (computed with the entire dataset) and the estimate obtained by each active learning method, as a function of trial number. The ALD-based method decreased the angular difference by 45 degrees with only 160 stimuli, while the Lewi-09 method required four times more data to achieve the same accuracy.

The most computationally expensive step in the algorithm is the eigendecomposition of the 256×256 posterior covariance matrix $\tilde{\Lambda}$, which took 30ms on a circa 2012 quad-core Mac Pro. In total, it took less than 60 ms to compute the optimal stimulus in each trial using a simple implementation of our algorithm, which we expect to be fast enough for use in real-time neurophysiology experiments.

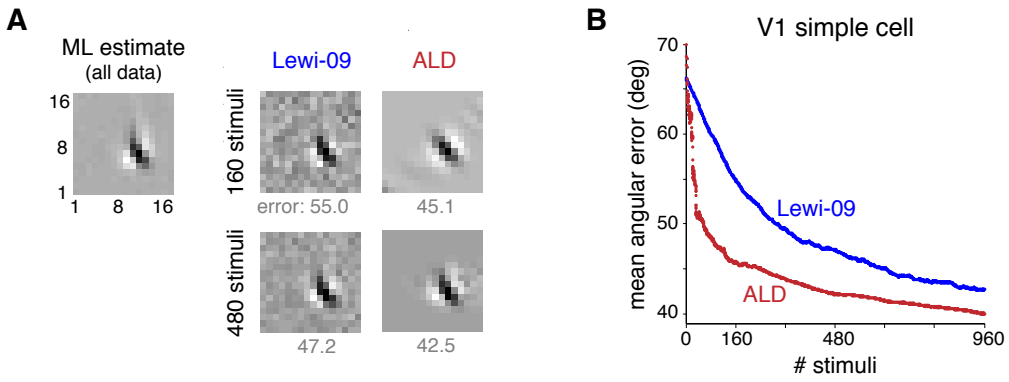

Figure 8: Active learning of a linear RF using data from a primate V1 simple cell. Original data were recorded in response to a 1D white noise "flickering bars" stimulus aligned with the neuron's preferred orientation (see [56]). We simulated active learning experiments via an offline analysis of a fixed dataset, where active learning methods had access to the set of stimuli but not the responses, and could re-order stimulus presentation according to expected utility. **(A)** Left: maximum likelihood estimate computed from entire 46-minute dataset (166K stimuli sampled at 100Hz). Right: RF estimates after 10 and 30 seconds of data selected using the Lewi-09 and localized (ALD) active learning methods. **(B)**: Mean angular error between active learning estimate and all-data maximum likelihood estimate as a function of the number of stimuli.

5 Discussion

We have discussed methods for stimulus selection in closed-loop neurophysiology experiments based on Bayesian active learning, also known as Bayesian adaptive experimental design. The key ingredients of any such method are: (1) a response model; (2) a prior distribution over model parameters; and (3) a utility function. The first two ingredients define the posterior distribution over parameters and the predictive distribution over future responses. The expectation of the loss function over the joint distribution over parameters and future responses defines the expected utility of a stimulus; the methods we have considered operate by greedily selecting the stimulus with maximal expected utility on each trial, given the data collected so far in the experiment. Finally, we discussed the details for two prominent application domains: tuning curves (or 'firing rate maps'), and linear receptive fields. For tuning curves, we discussed methods for both parametric and non-parametric [12] models; for linear receptive fields, we examined methods based on Poisson models with simple priors [7] and Gaussian models with structured hierarchical priors [9]. While these methods hold great promise for improving the efficiency of neurophysiology experiments, there remain a variety of challenges to overcome. We review several of these challenges below.

Adaptation

The methods we have described all assumed that the response on each trial was conditionally independent of responses on previous trials (eqs. 15 & 34). Clearly, this assumption is violated by the fact that neurons adapt: the response after a series of large-response trials may differ from the response to the same stimulus after a series of weak-response trials. An active learning method that models responses as conditionally independent may misattribute the effect of adaptation to a reduced response to certain

stimuli.

Fortunately, it is straightforward to incorporate a simple form of response adaptation under the conditionally Poisson models discussed in this chapter [50]. We can augment the model with a set of weights $\mathbf{h}$ that capture dependence of spike rate λ on the recent response history: $\lambda = g(\mathbf{k}^\top \mathbf{x} + \mathbf{h}^\top \mathbf{r}_{hist})$ for the receptive field model [7], or $\lambda = g(\phi(\mathbf{x}) + \mathbf{h}^\top \mathbf{r}_{hist})$ for the tuning curve model [54], where $\mathbf{r}_{hist}$ is some vector representation of response history at the current time.

This approach may not be sufficient to capture *stimulus specific* adaptation effects (e.g., differential adaptation to inputs that would ordinarily elicit similar spike rates), which suggests one key avenue for future research. However, it is worth noting that infomax learning and related Bayesian active learning methods typically interleave stimuli that elicit a wide range of spike rates (see Fig. 2), which stands in contrast to staircase methods, which tend move slowly across stimulus space, or methods that seek a maximal response [6, 13, 14] or to find a particular iso-reponse contour [10, 19]. Thus, Bayesian methods may in general be less susceptible to systematic biases induced by adaptation than other adaptive selection methods.

Greediness

A second possible problem with the methods we have discussed is that they are "greedy": they select the stimulus that maximizes expected utility *on each trial*. This may be sub-optimal compared to strategies that select stimuli with an eye to maximizing expected utility some finite number of trials in the future [37]. The computational difficulty of computing and maximizing expected utility multiple trials into the future makes this a challenging problem to undertake. Moreover, a technical result in [27] shows that greedy infomax methods are still provably better than standard methods under certain consistency conditions.

A different and more unavoidable greediness problem arises in the setting of linear receptive field estimation (Section 4). Specifically, if the filter $\mathbf{k}$ contains a temporal component, such as one might characterize using reverse correlation [57], then the methods described in Section 4 will neglect a large fraction of the available information. In this setting, the response depends on a temporal convolution of the filter with a stimulus movie. The effective stimulus at each time is a shifted copy of the previous stimulus plus a single new stimulus frame, so one must consider the response at multiple lags in order to accurately quantify the information each stimulus provides about the RF. Lewi and colleagues addressed this problem by extending the Lewi-09 method to the selection of maximally informative stimulus *sequences*, taking into account the mutual information between the model RF and an extended sequence of responses [8]. A comparable extension of our method based on structured priors [9] has not yet been undertaken, and represents an opportunity for future work.

Model specification

A third important concern for Bayesian active learning methods is the specification or selection of the neural response model. Although it is possible for methods based on a misspecified model to outperform well-specified models (e.g., as we showed in in Section 4, using a method based on Gaussian

noise for a Poisson neuron), there are few theoretical guarantees, and it is possible to find cases of model mismatch where adaptive methods are inferior to random *iid* methods [27]. It is therefore important to consider the accuracy and robustness of neural response models used for active learning.

One simple model selection problem involves the setting of hyperparameters governing the prior or response model, such the Gaussian process covariance function for tuning curve estimation (Sec. 3.3). In [12], we set hyperparameters by maximizing marginal likelihood after each trial using the data collected so far in the experiment, an approach that is common in Bayesian optimization and active learning. This is not strictly justified in a Bayesian framework, but empirically it performs better than fixing hyperparameters governing smoothness and marginal variance *a priori*. A more principled approach would be to place a prior distribution over hyperparameters and use sampling-based inference so that the expected utility incorporates uncertainty about the model [34]. Lewi *et al* explored several specific forms of model mismatch for RF estimation [7], and DiMattina & Zhang proposed methods aimed at the dual objective of parameter estimation and model comparison [17, 21].

Future directions

One natural direction for future research will be to combine the Poisson-GLM response model from [7] with the localized RF priors from [9] to build improved methods for receptive field estimation. This synthesis would combine the benefits of a more accurate neural model with a more informative and lower-entropy prior, provided that computational challenges can be overcome. A related idea was proposed in [58], which altered the utility function to maximize information along particular manifolds defined by parametric RF shapes (e.g., the manifold of Gabor filters within the larger space of all possible RFs). Both methods offer a desirable tradeoff between efficiency and robustness: they are set up to learn quickly when the true RF exhibits some form of low-dimensional structure, but still yield consistent estimates for arbitrary RFs. A promising opportunity for future work will be to design flexible priors to incorporate other forms of RF structure, such as space-time separability, [59], sparsity [60], or structured sparsity [61].

Another direction for improved active learning is the design of more flexible and accurate neural response models. Preliminary work in this direction has focused on models with nonlinear input transformations [7], and "over-dispersed" response noise [62, 63]. Other recent work has focused on hierarchical models for population responses, in which the responses from all previously recorded neurons are used to help determine the most informative stimuli for characterizing each subsequent neuron [64]. In future work, we hope to extend these methods to simultaneous multi-neuron recordings so that stimuli provide maximal information about an entire population of neurons, including their correlations. Taken together, we believe these methods will greatly improve the speed and accuracy of neural characterization and allow for ambitious, high-throughput neurophysiology experiments that are not possible with standard, non-adaptive methods. We feel these methods will be especially useful in higher cortical areas, where neurons exhibit nonlinear "mixed" selectivity in high-dimensional stimulus spaces where tuning is poorly understood.

Acknowledgments

This work was supported by the Sloan Foundation (JP), McKnight Foundation (JP), Simons Global Brain Award (JP), NSF CAREER Award IIS-1150186 (JP), the Gatsby Charitable Trust (MP) and a grant from the NIH (NIMH grant MH099611 JP). We thank N. Rust and T. Movshon for V1 data, and N. Roy and C. DiMattina for helpful comments on this manuscript.

References

[1] W. Bialek, F. Rieke, R. R. de Ruyter van Steveninck, R., and D. Warland. Reading a neural code. *Science*, 252:1854–1857, 1991.

[2] Fred Rieke, David Warland, Rob de Ruyter van Steveninck, and William Bialek. *Spikes: Exploring the Neural Code*. MIT Press, Cambridge, MA, USA, 1996.

[3] B. Aguera y Arcas and A. L. Fairhall. What causes a neuron to spike? *Neural Computation*, 15(8):1789–1807, 2003.

[4] O. Schwartz, J. W. Pillow, N. C. Rust, and E. P. Simoncelli. Spike-triggered neural characterization. *Journal of Vision*, 6(4):484–507, 7 2006.

[5] Jonathan D. Victor and Sheila Nirenberg. Indices for testing neural codes. *Neural Comput*, 20(12):2895–2936, Dec 2008.

[6] Peter Földiák. Stimulus optimisation in primary visual cortex. *Neurocomputing*, 38:1217–1222, 2001.

[7] J. Lewi, R. Butera, and L. Paninski. Sequential optimal design of neurophysiology experiments. *Neural Computation*, 21(3):619–687, 2009.

[8] Jeremy Lewi, David M. Schneider, Sarah M. N. Woolley, and Liam Paninski. Automating the design of informative sequences of sensory stimuli. *J Comput Neurosci*, 30(1):181–200, Feb 2011.

[9] Mijung Park and Jonathan W. Pillow. Bayesian active learning with localized priors for fast receptive field characterization. In P. Bartlett, F.C.N. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 2357–2365, 2012.

[10] Gregory D Horwitz and Charles A Hass. Nonlinear analysis of macaque v1 color tuning reveals cardinal directions for cortical color processing. *Nat Neurosci*, 15(6):913–919, 06 2012.

[11] Mijung Park, Greg Horwitz, and Jonathan W. Pillow. Active learning of neural response functions with gaussian processes. In J. Shawe-Taylor, R.S. Zemel, P. Bartlett, F.C.N. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24, pages 2043–2051, 2011.

[12] Mijung Park, J Patrick Weller, Gregory D Horwitz, and Jonathan W Pillow. Bayesian active learning of neural firing rate maps with transformed gaussian process priors. *Neural Computation*, 26(8):1519–1541, 2014.

[13] I. Nelken, Y. Prut, E. Vaadia, and M. Abeles. In search of the best stimulus: an optimization procedure for finding efficient stimuli in the cat auditory cortex. *Hearing Research*, 72:237–253, 1994.

[14] R. C. deCharms, D. T. Blake, and M. M. Merzenich. Optimizing sound features for cortical neurons. *Science*, 280(5368):1439–1443, May 1998.

[15] Christian K. Machens, Tim Gollisch, Olga Kolesnikova, and Andreas V. M. Herz. Testing the Efficiency of Sensory Coding with Optimal Stimulus Ensembles. *Neuron*, 47(3):447–456, August 2005.

[16] Kevin N O'Connor, Christopher I Petkov, and Mitchell L Sutter. Adaptive stimulus optimization for auditory cortical neurons. *Journal of Neurophysiology*, 94(6):4051–4067, 2005.

[17] Christopher DiMattina and Kechen Zhang. Active data collection for efficient estimation and comparison of nonlinear neural models. *Neural computation*, 23(9):2242–2288, 2011.

[18] Daniel Bölinger and Tim Gollisch. Closed-loop measurements of iso-response stimuli reveal dynamic nonlinear stimulus integration in the retina. *Neuron*, 73(2):333–346, 2012.

[19] Tim Gollisch and Andreas VM Herz. The iso-response method: measuring neuronal stimulus integration with closed-loop experiments. *Frontiers in neural circuits*, 6, 2012.

[20] J Benda, T Gollisch, C. K Machens, and A. V Herz. From response to stimulus: adaptive sampling in sensory physiology. *Curr. Opin. Neurobiol.*, 17(4):430–436, 2007.

[21] Christopher DiMattina and Kechen Zhang. Adaptive stimulus optimization for sensory systems neuroscience. *Frontiers in neural circuits*, 7, 2013.

[22] D. Mackay. Information-based objective functions for active data selection. *Neural Computation*, 4:589–603, 1992.

[23] Kathryn Chaloner and Isabella Verdinelli. Bayesian experimental design: A review. *Statistical Science*, 10(3):273–304, 1995.

[24] David A. Cohn, Zoubin Ghahramani, and Michael I. Jordan. Active learning with statistical models. *J. Artif. Intell. Res. (JAIR)*, 4:129–145, 1996.

[25] A. Watson and D. Pelli. QUEST: a Bayesian adaptive psychophysical method. *Perception and Psychophysics*, 33:113–120, 1983.

[26] Liam Paninski. Design of experiments via information theory. In S. Thrun, L. K. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, pages 1319–1326. MIT Press, 2004.

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

[28] D.V. Lindley. On a measure of the information provided an experiment. *Ann. Math. Statist.*, 27:986–1005, 1956.

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

[30] S. P. Luttrell. The use of transinformation in the design of data sampling scheme for inverse problems. *Inverse Prob.*, 1:199–218, 1985.

[31] Leonid L Kontsevich and Christopher W Tyler. Bayesian adaptive estimation of psychometric slope and threshold. *Vision research*, 39(16):2729–2737, 1999.

[32] Hendrik Kuck, Nando de Freitas, and Arnaud Doucet. SMC Samplers for Bayesian Optimal Nonlinear Design. In *Proc. IEEE Nonlinear Statistical Signal Processing Workshop*, pages 99–102. IEEE, 2006.

[33] Peter Müller and Giovanni Parmigiani. *Bayesian Analysis in Statistics and Econometrics: Essays in Honor of Arnold Zellner*. Wiley, 1996.

[34] Nicholas Roy and Andrew Mccallum. Toward Optimal Active Learning through Sampling Estimation of Error Reduction. In *Proc. 18th International Conf. on Machine Learning*, pages 441–448. Morgan Kaufmann, San Francisco, CA, 2001.

[35] E. Kapoor, A. Horvitz and S. Basu. Selective supervision: guiding supervised learning with decision-theoretic active learning. In *International Joint Conference on Artificial Intelligence*, 2007.

[36] Andreas Krause, Ajit Singh, and Carlos Guestrin. Near-Optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *J. Mach. Learn. Res.*, 9:235–284, June 2008.

[37] P Ewen King-Smith, Scott S Grigsby, Algis J Vingrys, Susan C Benes, and Aaron Supowit. Efficient and unbiased modifications of the quest threshold method: theory, simulations, experimental evaluation and practical implementation. *Vision research*, 34(7):885–912, 1994.

[38] David D. Lewis and William A. Gale. A sequential algorithm for training text classifiers. In *Proceedings of the ACM SIGIR conference on Research and Development in Information Retrieval*, pages 3–12. Springer-Verlag, 1994.

[39] Beau Cronin, Ian H. Stevenson, Mriganka Sur, and Konrad P. Körding. Hierarchical bayesian modeling and markov chain monte carlo sampling for tuning-curve analysis. *J Neurophysiol*, 103(1):591–602, Jan 2010.

[40] K. R. Rad and L. Paninski. Efficient, adaptive estimation of two-dimensional firing rate surfaces via gaussian process methods. *Network: Computation in Neural Systems*, 21(3-4):142–168, 2010.

[41] J. P. Cunningham, B. M. Yu, K. V. Shenoy, and M. Sahani. Inferring neural firing rates from spike trains using gaussian processes. *Advances in neural information processing systems*, 20:329–336, 2008.

[42] B. M. Yu, J. P. Cunningham, G. Santhanam, S. I. Ryu, K. V. Shenoy, and M. Sahani. Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. *Journal of Neurophysiology*, 102(1):614, 2009.

[43] Jakob H. Macke, Sebastian Gerwinn, Leonard E. White, Matthias Kaschube, and Matthias Bethge. Gaussian process methods for estimating cortical maps. *Neuroimage*, 56(2):570–581, May 2011.

[44] Carl Rasmussen and Chris Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

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

[46] R. Kass and A. Raftery. Bayes factors. *Journal of the American Statistical Association*, 90:773–795, 1995.

[47] J. W. Pillow, Y. Ahmadian, and L. Paninski. Model-based decoding, information estimation, and change-point detection techniques for multineuron spike trains. *Neural Comput*, 23(1):1–45, Jan 2011.

[48] S. V. David and J. L. Gallant. Predicting neuronal responses during natural vision. *Network: Computation in Neural Systems*, 16(2):239–260, 2005.

[49] John A Nelder and Robert WM Wedderburn. Generalized linear models. *Journal of the Royal Statistical Society. Series A (General)*, 135(3):370–384, 1972.

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

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

[52] K. Koepsell and F. T. Sommer. Information transmission in oscillatory neural activity. *Biological Cybernetics*, 99(4):403–416, 2008.

[53] Ryan C Kelly, Matthew A Smith, Robert E Kass, and Tai Sing Lee. Local field potentials indicate network state and account for neuronal response variability. *Journal of computational neuroscience*, 29(3):567–579, 2010.

[54] Mijung Park and Jonathan W. Pillow. Receptive field inference with localized priors. *PLoS Comput Biol*, 7(10):e1002219, 10 2011.

[55] W. R. Gilks and C. Berzuini. Following a moving target – monte carlo inference for dynamic bayesian models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(1):127–146, 2001.

[56] Nicole C Rust, Odelia Schwartz, J. Anthony Movshon, and Eero P Simoncelli. Spatiotemporal elements of macaque V1 receptive fields. *Neuron*, 46(6):945–956, Jun 2005.

[57] E. J. Chichilnisky. A simple white noise analysis of neuronal light responses. *Network: Computation in Neural Systems*, 12:199–213, 2001.

[58] Jeremy Lewi, Robert Butera, David M. Schneider, Sarah Woolley, and Liam Paninski. Designing neurophysiology experiments to optimally constrain receptive field models along parametric submanifolds. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 945–952. Curran Associates, Inc., 2009.

[59] Mijung Park and Jonathan W. Pillow. Bayesian inference for low rank spatiotemporal neural receptive fields. In *Advances in Neural Information Processing Systems 26*, pages 2688–2696. Curran Associates, Inc., 2013.

[60] Matthias Seeger, Sebastian Gerwinn, and Matthias Bethge. Bayesian inference for sparse generalized linear models. In *In Machine Learning: ECML*. Springer, 2007.

[61] Anqi Wu, Mijung Park, Oluwasanmi O Koyejo, and Jonathan W Pillow. Sparse bayesian structure learning with dependent relevance determination priors. In Z. Ghahramani, M. Welling, C. Cortes, N.D. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 1628–1636. Curran Associates, Inc., 2014.

[62] M. Park, J. P. Weller, G. D. Horwitz, and J. W. Pillow. Adaptive estimation of firing rate maps under super-poisson variability. In *Computational and Systems Neuroscience (CoSyNe) Annual Meeting*, 2013.

[63] Robbe L T. Goris, J Anthony Movshon, and Eero P. Simoncelli. Partitioning neuronal variability. *Nat Neurosci*, 17(6):858–865, Jun 2014.

[64] Woojae Kim, Mark A. Pitt, Zhong-Lin Lu, Mark Steyvers, and Jay I. Myung. A hierarchical adaptive approach to optimal experimental design. *Neural Computation*, 26(11):2465–2492, 2015/02/09 2014.