SEQUENTIAL OPTIMAL DESIGN OF NEUROPHYSIOLOGY EXPERIMENTS.

A Thesis
Presented to
The Academic Faculty

by

Jeremy Lewi

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in
Bioengineering

Biomedical Engineering
Georgia Institute of Technology
May 2009

SEQUENTIAL OPTIMAL DESIGN OF NEUROPHYSIOLOGY EXPERIMENTS.

Approved by:

Professor Robert Butera,
Committee Chair
School of Electrical Engineering
Georgia Institute of Technology

Professor Liam Paninski, Advisor
Department of Statistics
Columbia University

Professor Charles Isbell
College of Computing
Georgia Institute of Technology

Professor Christopher Rozell
School of Electrical Engineering
Georgia Institute of Technology

Professor Garrett Stanley
Department of Biomedical Engineering
Georgia Institute of Technology

Professor Brani Vidakovic
Department of Biomedical Engineering
Georgia Institute of Technology

Date Approved: 1 April 2009

For my parents,

for not making me get a real job.

ACKNOWLEDGEMENTS

I would like to thank my adviser Dr. Liam Paninski whose enthusiasm, encouragement, and unfailing patience were instrumental in seeing this project through to the end. I also want to thank my adviser Dr. Robert Butera for giving me the freedom to pursue research outside his purview and for his help and advice over the years.

I want to thank Quentin Huys, Joshua Vogelstein, and Isaac Clements for their support and friendship over the years.

In addition, I would like to thank my other committee members, Drs. Christopher Rozell, Charles Isbell, Garrett Stanley, and Brani Vidakovic for their time, effort, and feedback.

I would like to acknowledge the hard work of our collaborators David Schneider and Dr. Sarah Woolley who graciously shared their data with us.

I would also like to thank Dr. Paul Sajda for introducing me to computational neuroscience research, machine learning, and Unix; Edgar Brown for having the foresight to purchase a cluster and for giving me super-user privileges; Terri Lee for all her hard work administering my fellowship; Drs. Steve DeWeerth and Robert Butera for creating the Hybrid Micro-systems Neural Program which brought me to Georgia Tech. and taught me that I am not an experimentalist.

I would also like to acknowledge the Department of Energy and the Krell Institute for supporting my research with a Computational Science Graduate Fellowship.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

For well over 200 years, scientists and doctors have been poking and prodding brains in every which way in an effort to understand how they work. The earliest pokes were quite crude, often involving permanent forms of brain damage. In the 1800's a railroad worker named Phineas P. Gage survived an accident which drove an iron rod through his head. Though he lived, Gage's behavior exhibited profound changes after the accident. Gage's trauma proved fortuitous for neuroscience because his condition provided some of the earliest evidence of how different brain regions affect behavior and personality. Though neural injury continues to be an active area of research within neuroscience, technology has given neuroscientists a number of tools for stimulating and observing the brain in much subtler ways.

Nonetheless, the basic experimental paradigm remains the same; poke the brain and see what happens. For example, neuroscientists studying the visual or auditory system can easily generate any image or sound they can imagine to see how an organism or neuron will respond. Since neuroscientists can now easily design more pokes then they could every deliver, a fundamental question is "What pokes should they actually use?" The complexity of the brain means that only a small number of the pokes scientists can deliver will produce any information about the brain. In Gage's case for example, the rod delivered just enough damage to noticeably affect his behavior without killing him. As Gage's case illustrates, one of the fundamental challenges of experimental neuroscience is finding the right stimulus parameters to produce an informative response in the system being studied. This thesis addresses this problem by developing algorithms to sequentially optimize neurophysiology experiments.

Every experiment we conduct contains information about how the brain works.

Before conducting the next experiment we should use what we have already learned to decide which experiment we should perform next. In particular, we should design an experiment which will reveal the most information about the brain. At a high level, neuroscientists already perform this type of sequential, optimal experimental design; for example crude experiments in the manner of Phineas Gage which knockout entire regions of the brain have given rise to modern experimental techniques which probe the responses of individual neurons using finely tuned stimuli. The goal of this thesis is to develop automated and rigorous methods for optimizing neurophysiology experiments efficiently and at a much finer time scale. In particular, we present methods for near instantaneous optimization of the stimulus being used to drive a neuron.

CHAPTER I

INTRODUCTION

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. To understand how neurons encode information, neurophysiologists record the responses of neurons to known inputs and then try to fit a model to the data [122, 145, 170]. Learning a neuron's response function is challenging because a neuron's response is a highly nonlinear, time-varying, noisy function of a complex, high-dimensional input. Consequently, the ease with which we can infer the input-output relationship of a neuron is highly dependent on the inputs we choose because not all inputs provide equal amounts of information; in fact most inputs provide very little information.

A primary example of the importance of selecting good inputs is the Nobel-prize winning work of Hubel and Wiesel on the activity of neurons in primary-visual cortex of cat [74]. For months, Hubel and Wiesel failed to elicit spiking in recordings from neurons in the visual cortex while projecting an image of black dots onto the cat's visual system. It was only when they serendipitously presented a moving bar by sliding a glass slide over the ophthalmoscope that the neuron they were recording from started spiking [76]. Their results led to the discovery of simple cells. In retrospect, the early failure of Hubel and Wiesel to elicit spiking is easy to understand. Simple cells are highly selective to bars oriented at specific angles. In general only stimuli meeting these criteria will depolarize a simple cell enough to drive it above its spiking threshold causing the neuron to fire [74]. Since Hubel and Wiesel could

not record sub-threshold changes in the neuron's membrane potential, stimuli which did not drive the neuron to fire provided almost no information about the neuron's response. In contrast, bars oriented at different angles are highly informative because they can be used to determine the precise tuning of a simple cell; i.e the size and orientation of bars to which the cell is sensitive. Hubel and Wiesel's experiments provide a clear illustration of how nonlinearities complicate system identification. In this case, the threshold nonlinearity in neurons means that for a simple cell only a tiny fraction of all possible images will provide information about the cell's response function. The story of Hubel and Wiesel shows that there is a clear need for rigorous methods for determining which inputs will provide the most information about a neuron's response function. This thesis addresses this problem by proposing methods for designing optimal experiments; the design in this context being a procedure for picking the inputs.

Hubel and Wiesel's success depended critically on their ability to design better, more refined experiments as they gathered information. A similar trajectory is evident in the whole of experimental sensory neurophysiology. In audition and vision early experiments used simple artificial stimuli like dots of light or single tone sounds [85, 52, 42]. As sensory neurophysiology advanced, experimentalists started using more complex stimuli such as natural images and sounds which could more effectively drive neurons to fire [158, 127, 162, 144, 168, 34]. The history of neuroscience shows that optimizing the design of neurophysiology experiments is crucial to understanding how the brain works.

1.1 Background

The methods presented in this thesis build on a rich history of experimental design research in the statistics and machine learning communities. The two communities use slightly different terminology; in statistics the problem is known as sequential

optimal experimental design [40] while in machine learning the problem is referred to as active learning [28, 155, 93]. In both cases the underlying problem is to learn a function $f(\vec{s}_t)$ which maps some input, $\vec{s}_t$, into some output, $r_t = f(\vec{s}_t)$. Early efforts in statistics focused on criterion based design and the linear model [82, 83, 57, 147]. These methods are based on optimizing an objective function related to the task the designer wishes to accomplish. For example, if the designer wants to make predictions about future responses, then he might design his experiments to minimize the expected mean squared error of the predicted responses; a criterion known as A-optimality. In contrast, if the designer is primarily interested in the shape of the function $f(\vec{s}_t)$ then he might minimize the variance of the estimator of f; a criterion known as D-optimality. A and D-optimality are two of the most popular optimality criteria and are generally referred to as the alphabet criteria. Naturally, the optimal choice of $\vec{s}_t$ depends on several factors such as the amount and variability of the noise. If for example the noise is input dependent then we might need to devote more samples to measuring the response in regions where the noise is large.

One reason the linear model has received so much attention is because in its simplest formulation the noise is constant and uniform with respect to the inputs and responses. As a result, the quality of the design depends entirely on the inputs chosen. In fact most traditional optimality criteria for the linear model can be written as a simple function of the covariance matrix of the inputs because the covariance matrix measures how the inputs are dispersed throughout input space [57, 147]. When the quality of a design depends only on the inputs we can compute the optimal design completely a-priori.

For nonlinear models, the story is more complex because it is generally impossible to pick one design which is simultaneously optimal for all possible functions [60]. Intuitively, to infer the most about the underlying input-output relationship, we want to pick inputs for which the response is highly dependent on the input and for which

their is very little noise in the responses. For arbitrary nonlinear models, identifying inputs for which the noise is low and the response is highly sensitive to the input would require knowing the very input-output relationship we are trying to learn. Nonetheless, optimal design using the traditional alphabet criteria can be extended to nonlinear models by considering locally optimal designs [60, 25, 43]. A locally optimal design is a design which is optimized for one particular function in the set of possible response functions. Hence, the classical approach to optimal design is to pick some initial guess of the unknown response function and compute the optimal design for that function. The major drawback of locally-optimal methods is that they are highly sensitive to the initial guess of the unknown function.

In contrast to the classical approach of optimal design, the Bayesian approach to optimal experimental design can handle nonlinear models in a more consistent fashion. Instead of measuring the informativeness of an input for a particular model, the Bayesian approach averages the informativeness of the input under all possible models weighted by the probability of each model. The Bayesian approach casts experimental design as a decision theory problem by formulating an objective function which measures the quality of the design [96, 13, 32, 24]. In keeping with the principles of decision theory, this utility function can depend on unknown quantities such as the parameters of the unknown response function [11]. To compute the optimal experimental design, we simply maximize the expected value of the utility function with respect to the design. The expected utility is computed using the prior on the unknown quantities. In many ways, the Bayesian approach to optimal experimental design is just an extension of classical optimality criteria which uses a prior distribution to compute an average value of classical design criteria over all possible models. In many cases, choosing a suitable prior and utility function leads to a design which is equivalent to a locally-optimal design using one of the traditional alphabet design criteria [32, 24, 35].

The Bayesian approach also provides a method for continually redesigning our experiments as data is collected. To update the design, we simply re-optimize the expected utility of the design. However, instead of computing the expected utility using our prior we use our posterior on the set of all possible response functions given the data already collected. The posterior assigns to each function a probability which measures the likelihood that a particular function provides the best model of a neuron's input-output function. Since the posterior measures what we have learned from past experiments, this sequential optimization procedure takes advantage of past experiments to design the best experiment to conduct next.

While the Bayesian approach presents a consistent mathematical framework for optimal experimental design, actually computing the Bayes optimal design is a very difficult problem. In decision theory, one of the fundamental challenges is representing the posterior distribution and computing expectations with respect to the posterior distribution [11]. Various algorithms using sampling techniques [63], genetic algorithms [70], and dynamic programming [40], have been proposed for finding the optimal design. Most of these methods, however, only work for low dimensional systems. The methods presented in this thesis rely on parametric representations of the posterior and the unknown response function to make the computations tractable. In particular, we assume the response function is a generalized linear model(GLM) (see Chapter 2) [104].

The generalized linear model is a parametric family of nonlinear, statistical models which has received a lot of attention in the statistics literature because it can be used for many applications e.g. analyzing clinical trials [87], climate patterns [171], and the spread of infectious diseases [78]. In neuroscience, the GLM is a more general version of the linear-nonlinear-Poisson (LNP) model which is widely used to model neurons in the early visual system [154, 125, 106, 26, 146, 139]. The success of the LNP model has led to the development of a framework for modeling neurons as point

processes using the GLM [17, 18, 113, 156, 116]. Thus, there is an extensive literature dealing with the robust estimation of GLMs which we exploit in this thesis [161, 92].

Despite the long history of GLMs in the statistics literature, the problem of optimal experimental design with GLMs is still largely unsolved [81]. In particular there are no general algorithms for solving the sequential optimal design problem for high-dimensional (multi-parameter) GLMs. Previous work has largely focused on specific models in the GLM family with the binary response model, which includes logistic regression, receiving by far the most attention [23, 27, 84, 103, 63, 132, 130]. One reason this model has received so much attention is that the outputs are binary which helps make optimizing the design tractable. Other distributions which have received attention are the Poisson Model [108, 27]. The vast majority of previous research has considered the simple case where the input was either a scalar or low-dimensional (e.g 2-4 dimensions) [164, 47]. The restriction to low dimensional inputs is necessary because the vast majority of the numerical techniques used to optimize the design do not scale well to high dimensions. Compared to methods using classical alphabet criteria, methods involving Bayesian optimality criteria have an even harder time scaling to high-dimensions because of the difficulty of computing the expected utility in high-dimensions [22, 142]. Trying to sequentially optimize the design as data is collected only exacerbates the computational problems. As a result, most efforts involving sequential optimal design have focused on low dimensional models using local optimality criteria [25, 105, 47]. Generally, these methods work by computing the locally optimal design using the best guess, given the data obtained so far, of the response function. Therefore an important and very open problem in statistics is how to sequentially compute Bayes optimal designs for high-dimensional GLMs.

1.2 Outline

This thesis presents methods for sequentially optimizing neurophysiology experiments using GLMs. We use a Bayesian approach because we want to take our uncertainty into account and exploit prior knowledge (see Chapter 5). We quantify the utility of a design by computing the mutual information between the data and the response function. The mutual information measures the expected reduction in our uncertainty about the neuron's response function (see Chapter 2) [99, 114]. Neurophysiology presents a particularly challenging application for sequential optimal experimental design because neurons respond to complex, high-dimensional inputs, e.g sounds and movies. We therefore cannot use existing algorithms which can typically only handle stimuli with 2-10 dimensions. Consequently, the focus of this thesis is on developing methods for sequential optimal design using high dimensional GLMs and Bayesian optimality criteria. While we primarily focus on GLMs with a Poisson likelihood, our methods are general enough that they can be extended with little modification to a much larger set of models within the GLM family [89]. As a result, we think the work presented in this thesis will be of interest to both the neurophysiology and statistics/machine learning communities. The statistics community will be interested in our methods for computing the Bayes optimal design for GLMs with hundreds to thousands of parameters. Neurophysiologists in contrast will be interested in our results showing that by using optimal designs, they could potentially reduce by a factor of 2-4 the amount of data needed to learn a neuron's response function (see Chapter 4).

Most previous attempts at optimal design of neurophysiology experiments used empirical methods. In general, these methods work by perturbing the most recent stimulus so as to increase or hold steady some easily observable quantity such as the firing rate [59, 65, 48, 111, 10] (for a more thorough discussion of previous work see Chapter 2). In contrast, our work is one of the few attempts to optimize

neurophysiology experiments by maximizing a principled objective function based on information theory [97, 98].

We begin in Chapter 2 by presenting a greedy algorithm which considers only the information to be gained from the next trial. One of the main contributions of this chapter is to show how the computations may be performed in real time (10-100ms) even for high dimensional systems by using a GLM to model the neuron and a Gaussian approximation of the posterior. We test our algorithm using simulations which show our information maximizing design can provide an order of magnitude of improvement over traditional, non-optimized designs. This chapter also presents an extended discussion of the mutual information and the GLM which is necessary for understanding later chapters.

In Chapter 3 we consider the problem of finding the optimal sequence of trials, i.e. non-greedy optimization, in the limiting case that the number of trials goes to infinity. We prove that as the number of trials goes to infinity, the optimal sequence of inputs is just as informative as a sequence of inputs generated by sampling an optimal stochastic process. This stochastic process can be found by solving a convex optimization problem. We present approximate methods for computing the optimal process when we restrict consideration to Gaussian processes. Simulations show that this algorithm leads to significant improvements over non-optimized designs. Non-greedy optimization is particularly important when a neuron integrates information over time; e.g. neurons which detect motion. In this case non-greedy optimization is essential in order to generate stimuli which can effectively probe a neuron's dependence on the temporal structure of the input. The algorithm presented in this chapter is also much easier to implement in actual experiments then our greedy algorithm.

In Chapter 4 we investigate the application of our methods to auditory neurophysiology experiments in zebra finch. We show using real data that 1) the GLM

provides good fits to real data and 2) using our methods to design optimal experiments could reduce the amount of data needed by a factor of 2-4. We think these results underestimate the actual speedup that could be achieved in real experiments because in these simulations we were necessarily restricted to picking the best stimulus from the small set of stimuli actually presented to the neuron. In an actual experiment, we could potentially optimize over the entire stimulus space instead of restricting ourselves to a very small subset which was chosen in an ad-hoc fashion. In comparison to our simulations using real data, our simulations in Chapter 2, in which we optimize over a much larger stimulus domain, show a factor of 10 speedup.

Finally in Chapter 5 we show how the methods presented in Chapter 2 and Chapter 3 can be modified to incorporate realistic prior information about a neuron's response function in a way that makes computing the mutual information tractable.

CHAPTER II

GREEDY OPTIMIZATION OF THE STIMULI DURING AN EXPERIMENT.

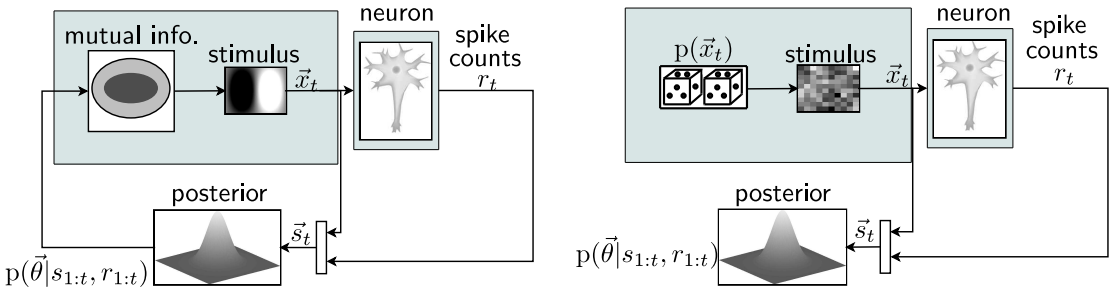
Adaptively optimizing experiments has the potential to significantly reduce the number of trials needed to build parametric statistical models of neural systems. However, application of adaptive methods to neurophysiology has been limited by severe computational challenges. Since most neurons are high 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) which 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 2-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, non-systematic drifts in a neuron's activity.

2.1 Introduction

In most neurophysiology experiments, data is collected according to a design that is finalized before the experiment begins. During the experiment, the data already collected is rarely analyzed to evaluate the quality of the design. The data already collected, however, often contains information which could be used to redesign our experiments to better test our hypotheses [57, 24, 84, 159, 130]. Adaptive experimental designs are particularly valuable in domains where data is expensive and/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 neuroscientists to investigate high-dimensional, complex neural systems. By using adaptive experimental designs, neuroscientists could potentially collect data more efficiently. In this chapter, 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



(a) (b)

Figure 1: a) Schematic of the process for designing information maximizing 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 info. max. algorithm updates the posterior after each trial. b) Schematic of the typical i.i.d. design of experiments. Stimuli are selected by drawing i.i.d. samples from a distribution which is chosen before the experiment starts. An i.i.d. design does not use the posterior distribution to choose stimuli.

past activity in addition to recent stimuli. To model the dependence on past stimuli and responses, we define the input as a vector comprised of the current and recent stimuli, $\{\vec{x}_t, \vec{x}_{t-1}, ..., \vec{x}_{t-t_k}\}$, as well as the neuron's recent activity, $\{r_{t-1}, \ldots, r_{t-t_a}\}$ [80, 156]. $\vec{x}_t$ and r_t denote the stimulus and firing rate at time t respectively. When we optimize the input for time $t+1$ we can only control $\vec{x}_{t+1}$, 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,

$$\vec{s}_t = \left[\vec{x}_t^T, \vec{s}_{f,t}^T\right]^T \tag{1}$$

$$\vec{s}_{x,t} = \vec{x}_t \tag{2}$$

$$\vec{s}_{f,t} = \left[\vec{x}_{t-1}^T, \ldots, \vec{x}_{t-t_k}^T, r_{t-1}, \ldots, r_{t-t_a}\right]^T. \tag{3}$$

$\vec{s}_t$ is the input at time t. $\vec{s}_{f,t}$ is a vector comprised of 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 dependency on past stimuli and/or responses; i.e. the values of t_k and t_a will depend on the model adopted for a particular experiment.

We can describe a model which 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 which lie in some space Θ. Each vector $\vec{\theta}$ denotes a particular model in this space. To fit a model, $p(r_t|\vec{s}_t, \vec{\theta})$, 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. Non-random 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 non-sequential approach is to fix this distribution before the experiment starts and then select the stimulus on each trial by drawing independently 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 which 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 [95, 13, 160, 31, 99, 24, 114],

$$I(\{r_t, \vec{s}_t\}; \vec{\theta}) = \int p(r_t, \vec{s}_t, \vec{\theta}) \frac{\log p(r_t, \vec{s}_t, \vec{\theta})}{\log p(r_t, \vec{s}_t) p(\vec{\theta})} dr_t d\vec{s}_t d\vec{\theta}. \tag{4}$$

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 very 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 $p(\vec{\theta}|\{r_t, r_{t-1}, \ldots, r_1\}, \{\vec{s}_t, \vec{s}_{t-1}, ..., \vec{s}_1\})$ 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, spatio-temporal 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 paper 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 which can be solved numerically. The solution to this 2-d 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 2.5.4), 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 our algorithm reduces the complexity with respect to dimensionality from exponential to on average quadratic running time.

This chapter is organized as follows. We start in Section 2.2 by presenting the generalized linear model of neural systems. In Section 2.3 we present an online method for computing a Gaussian approximation of the posterior distribution on the GLM's parameters. In Section 2.4 we show how the mutual information, $I(r_t; \vec{\theta}|\vec{s}_t)$, can be approximated by a much simpler, low-dimensional function. In Section 2.5 we present the procedure for picking optimal stimuli and show some simulation results. In Section 2.6 we generalize our basic methods to some important extensions of the GLM which are needed to handle more complicated experiments. In Section 2.7, 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 [114]. We therefore conclude that this efficient (albeit approximate) implementation produces designs which are in fact asymptotically optimal. Simulations investigating the issue of model misspecification

Table 1: Definitions of the various symbols and conventions we use throughout the chapter.

$\vec{x}_t$		The stimulus at time t.
r_t		The response at time t.
$\vec{s}_t$	$=\left[\vec{s}_{x,t}^T, \vec{s}_{f,t}^T\right]^T$	The complete input at time t.
$\vec{s}_{x,t}$		The controllable part of the input at time t
$\vec{s}_{f,t}$		The fixed part of the input at time t
$\boldsymbol{x_{1:t}}$	$\triangleq \{\vec{x}_1, \ldots, \vec{x}_t\}$	The sequence of stimuli up to time t. We use boldface to denote a matrix.
$\boldsymbol{r_{1:t}}$	$\triangleq \{r_1, \ldots, r_t\}$	The sequence of observations up to time t.
$\boldsymbol{s_{1:t}}$	$\triangleq \{\vec{s}_1, \ldots, \vec{s}_t\}$	The sequence of inputs up to time t.
$E_\omega(\omega)$	$= \int p(\omega)\omega d\omega.$	The expectation with respect to the distribution on the random variable denoted in the subscript.
$H(p(\omega\|\gamma))$	$\triangleq \int -p(\omega\|\gamma)\log p(\omega\|\gamma)d\omega.$	The entropy of the distribution $p(\omega\|\gamma)$.
$\dim(\vec{\theta})$	$= \dim(\vec{\theta})$	The dimensionality of the model.
$p(\vec{\theta}\|\vec{\mu}_t, \boldsymbol{C}_t)$		The Gaussian approximation of the posterior distribution, $p(\vec{\theta}\|\boldsymbol{s_{1:t}}, \boldsymbol{r_{1:t}})$. $(\vec{\mu}_t, \boldsymbol{C}_t)$ are the mean and covariance matrix respectively.

are presented in Section 2.8. Finally, we discuss some limitations and directions for future work in Section 2.9. To help the reader, we summarize in Table 1 the notation that we will use in the rest of the chapter.

2.2 The parametric model

For the model space, Θ, we choose the set of generalized linear models (GLM). The GLM is a tractable and flexible parametric family which has proven useful in neurophysiology [104, 139, 113, 156, 116]. GLMs are fairly natural from a physiological point of view, with close connections to biophysical models such as the integrate-and-fire cell. Consequently, GLMs have been applied in a wide variety of experimental settings [17, 18, 26, 149, 115].

A generalized linear 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, which

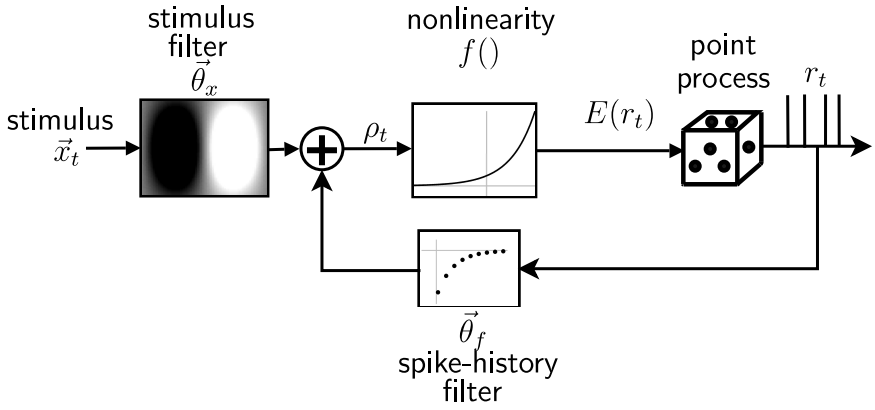


Figure 2: A diagram of a general linear model of a neuron. A GLM consists of a linear filter followed by a static nonlinearity. The output of this cascade is the estimated, instantaneous firing rate of a neuron. The unknown parameters $\vec{\theta} = \left[\vec{\theta}_x^T, \vec{\theta}_f^T\right]^T$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

is a non-linear function of the input,

$$\lambda_t = E_{r_t|\vec{s}_t,\vec{\theta}}(r_t) = f\left(\vec{\theta}^T\vec{s}_t\right) = f\left(\vec{\theta}_x^T\vec{s}_{x,t} + \vec{\theta}_f^T\vec{s}_{f,t}\right). \tag{5}$$

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 [14, 80, 113, 156]. As noted earlier, we use subscripts to distinguish the components which we can control from those which are fixed, 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 $\vec{\theta} = \left[\vec{\theta}_x^T, \vec{\theta}_f^T\right]^T$ 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 $\vec{\theta}$,

$$\log p\left(r_t|\vec{s}_t,\vec{\theta}\right) = \log\frac{e^{-\lambda_t dt}(\lambda_t dt)^{r_t}}{r_t!} \tag{6}$$

$$= r_t \log f(\vec{\theta}^T\vec{s}_t) - f(\vec{\theta}^T\vec{s}_t)dt + const. \tag{7}$$

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 chapter, we always use a Poisson distribution for the conditional likelihood, $p(r_t|\vec{s}_t, \vec{\theta})$, because it is the best distribution for modeling spiking neurons. However, by making some minor modifications to our algorithm, we can use our algorithm with other distributions in the exponential family [89].

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 $p(r_t|\vec{s}_t, \vec{\theta})$ is a Poisson distribution, a sufficient condition for concavity of the log-likelihood is that the non-linearity $f()$ is a convex and log-concave function [161, 69, 104, 113]. $f()$ can only be convex and log-concave if its contours are linear. When the contours of $f()$ 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,

$$\rho_t = \vec{\theta}^T \vec{s}_t. \tag{8}$$

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 [92, 113] and our own results, (see Section 2.8), 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.

2.3 Representing and updating the posterior

Our first computational challenge is representing and updating the posterior distribution on the parameters, $p(\vec{\theta}|\boldsymbol{r_{1:t}}, \boldsymbol{s_{1:t}})$. We use a fast, sequential procedure for constructing a Gaussian approximation of the posterior, Figure 3. This Gaussian

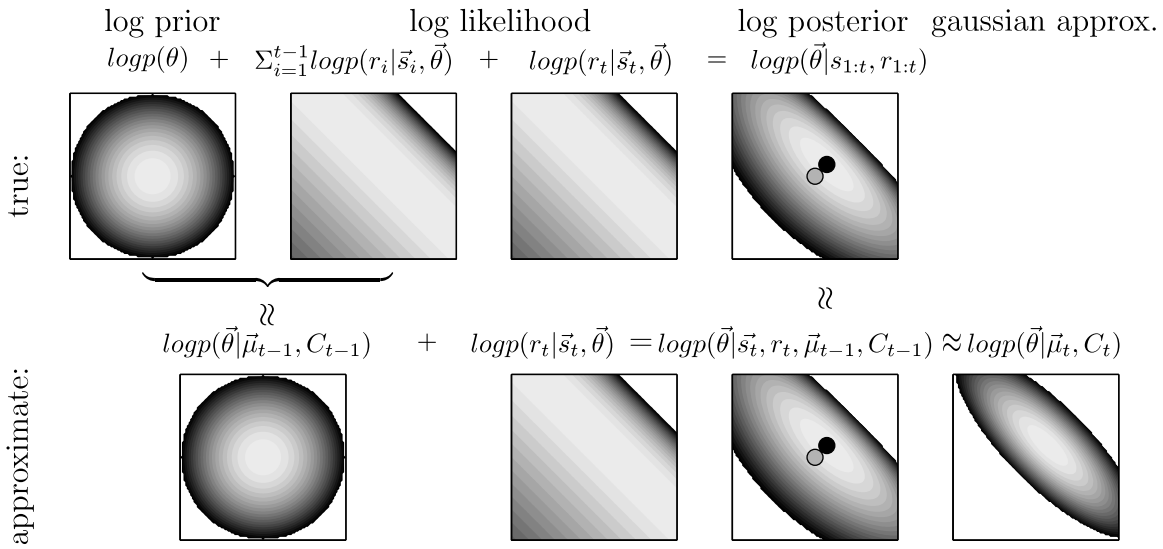


Figure 3: A schematic illustrating the procedure for recursively constructing the Gaussian approximation of the true posterior; $\dim(\vec{\theta}) = 2$. The images are contour plots of the log prior, log likelihoods, log posterior, and log of the Gaussian approximation of the posterior (see text for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

approximation leads to an update which 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 [114] 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, Figure 3. Since the Gaussian approximation of the posterior at time $t-1$, $p(\vec{\theta}|\vec{\mu}_{t-1}, \boldsymbol{C}_{t-1})$, summarizes the information in the first $t-1$ trials, we can use this distribution to approximate the log-posterior after the t^{th} trial,

$$\log p(\vec{\theta}|\boldsymbol{s_{1:t}}, \boldsymbol{r_{1:t}}) = \underbrace{\log p(\vec{\theta}) + \sum_{i=1}^{t-1} \log p(r_i|\vec{s}_i, \vec{\theta})} + \log p\left(r_t|\vec{s}_t, \vec{\theta}\right) + const. \quad (9)$$

$$\approx \log p\left(\vec{\theta}|\vec{\mu}_{t-1}, \boldsymbol{C}_{t-1}\right) + \log p\left(r_t|\vec{s}_t, \vec{\theta}\right) + const. \quad (10)$$

$$\approx \log p\left(\vec{\theta}|\vec{\mu}_t, \boldsymbol{C}_t\right) + const. \quad (11)$$

We fit the log of a Gaussian to the approximation of the log-posterior in Eqn. 10 using the Laplace Method [11, 100]. 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 $O(\dim(\vec{\theta})^2)$ whereas fitting a Gaussian distribution to the true posterior is $O(t\dim(\vec{\theta})^3)$. Since t and $\dim(\vec{\theta})$ 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, $\dim(\vec{\theta}) = O(10)$, 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 Eqn. 10. The key to rapidly updating our posterior is that we can easily compute the direction in which the peak of Eqn. 10 lies relative to $\vec{\mu}_{t-1}$. Once we know the direction in which $\vec{\mu}_t$ lies, we just need to perform a 1-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 Eqn. 10,

$$\frac{d \log p(\vec{\theta}|\boldsymbol{r_{1:t}}, \boldsymbol{s_{1:t}})}{d\vec{\theta}} \approx \frac{\partial \log p(\vec{\theta}|\vec{\mu}_{t-1}, \boldsymbol{C}_{t-1})}{\partial \vec{\theta}} + \frac{\partial \log p(r_t|\vec{s}_t, \vec{\theta})}{\partial \vec{\theta}} \tag{12}$$

$$= -(\vec{\theta} - \vec{\mu}_{t-1})^T \boldsymbol{C}_{t-1}^{-1} + \left(\frac{r_t}{f(\rho_t)} - dt\right) \frac{df}{d\rho}\bigg|_{\rho_t} \vec{s}_t^T. \tag{13}$$

At the peak of the log posterior, the gradient equals zero which means the first term in Eqn. 13 must be parallel to $\vec{s}_t$. Since $\boldsymbol{C}_{t-1}$ is non-singular, $\vec{\mu}_t - \vec{\mu}_{t-1}$ must be parallel to $\boldsymbol{C}_{t-1}\vec{s}_t$,

$$\vec{\mu}_t = \vec{\mu}_{t-1} + \Delta_t \boldsymbol{C}_{t-1}\vec{s}_t. \tag{14}$$

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

$$-\Delta_t + \left(\frac{r_t}{f(\rho_t)} - dt\right) \frac{df}{d\rho}\bigg|_{\rho_t = \vec{s}_t^T \vec{\mu}_{t-1} + \Delta_t \vec{s}_t^T \boldsymbol{C}_{t-1} \vec{s}_t} = 0. \tag{15}$$

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, Eqn. 15 is the solution to a 1-dimensional concave optimization problem. Eqn. 15 is therefore guaranteed to have a single, unique solution. Solving this 1-dimensional problem involves a single matrix-vector multiplication which requires $O(\dim(\vec{\theta})^2)$ time.

Having found $\vec{\mu}_t$, we estimate the covariance matrix $\boldsymbol{C}_t$ of the posterior by forming

the Taylor approximation of Eqn. 10 about $\vec{\mu}_t$:

$$\log p(\vec{\theta}|\boldsymbol{r_{1:t}}, \boldsymbol{s_{1:t}}) \approx -\frac{1}{2}(\vec{\theta} - \vec{\mu}_t)^T \boldsymbol{C}_t^{-1}(\vec{\theta} - \vec{\mu}_t) + const. \tag{16}$$

$$-\boldsymbol{C}_t^{-1} = \frac{\partial^2 \log p(\vec{\theta}|\vec{\mu}_t, \boldsymbol{C}_t)}{d\vec{\theta}^2} \tag{17}$$

$$= \frac{\partial^2 \log p(\vec{\theta}|\vec{\mu}_{t-1}, \boldsymbol{C}_{t-1})}{\partial\vec{\theta}^2} + \frac{\partial^2 \log p(r_t|\vec{s}_t, \vec{\theta})}{\partial\vec{\theta}^2} \tag{18}$$

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, Eqn. 18. 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 [11]. The second derivative of the log-likelihood provides an intuitive metric for the informativeness of an observation because a larger second derivative means 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 2nd derivatives, of the log-posterior we only need to sum 2 matrices: $\boldsymbol{C}_{t-1}^{-1}$ and the Hessian of $\log p\left(r_t|\vec{s}_t, \vec{\theta}\right)$. The Hessian of the log-likelihood is a rank one matrix. We can therefore efficiently invert the Hessian of the updated log posterior in $O(\dim(\vec{\theta})^2)$ time using the Woodbury matrix lemma [71, 135]. Evaluating the derivatives in Eqn. 18 and using the Woodbury

lemma yields

$$\boldsymbol{C}_t = \left(\boldsymbol{C}_{t-1}^{-1} - \frac{\partial^2 \log p(r_t|\rho_t)}{\partial \rho^2}\vec{s}_t\vec{s}_t^T\right)^{-1} \tag{19}$$

$$= \boldsymbol{C}_{t-1} - \frac{\boldsymbol{C}_{t-1}\vec{s}_t D(r_t,\rho_t)\vec{s}_t^T\boldsymbol{C}_{t-1}}{1 + D(r_t,\rho_t)\vec{s}_t^T\boldsymbol{C}_{t-1}\vec{s}_t} \tag{20}$$

$$D(r_t,\rho_t) = -\frac{\partial^2 \log p(r_t|\rho)}{\partial \rho^2}\bigg|_{\rho_t} = -\left(\frac{r_t}{f(\rho_t)} - dt\right)\frac{d^2 f}{d\rho^2}\bigg|_{\rho_t} + \frac{r_t}{(f(\rho_t))^2}\left(\frac{df}{d\rho}\bigg|_{\rho_t}\right)^2 \tag{21}$$

$$\rho_t = \vec{\theta}^T\vec{s}_t. \tag{22}$$

$D(r_t,\rho_t)$ is the 1-dimensional Fisher information; i.e. 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

$$\rho_t \approx \vec{\mu}_t^T\vec{s}_t \tag{23}$$

to compute the new covariance matrix. Since computing the covariance matrix is just a rank one update, computing the updated Gaussian approximation only requires $O(\dim(\vec{\theta})^2)$ 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 [107, 133].

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 [56, 137]. 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, Section 2.5.2, 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 [134].

2.4 Computing the mutual information

A rigorous Bayesian approach to sequential optimal experimental design is to pick the stimulus which maximizes the expected value of a utility function [13]. Common functions are the mean squared error of the model's predictions [57, 28, 132], the entropy of the responses [8], and the expected information gain [95, 13, 99, 24, 138]. 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, $I(\{\vec{s}_{t+1}, r_{t+1}\}; \vec{\theta} | \boldsymbol{s_{1:t}}, \boldsymbol{r_{1:t}})$, conditioned on the data already collected as a function of the design $p(\vec{x}_{t+1})$,

$$p_{opt}(\vec{x}_{t+1}) = \arg\max_{p(\vec{x}_{t+1})} I(\{\vec{s}_{t+1}, r_{t+1}\}; \vec{\theta} | \boldsymbol{s_{1:t}}, \boldsymbol{r_{1:t}}). \tag{24}$$

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 which 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 only varies 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 only depends on $p(\rho_{t+1}|\vec{s}_{t+1}, \vec{\mu}_t, \boldsymbol{C}_t)$ instead of our full posterior $p(\vec{\theta}|\vec{s}_{t+1}, \vec{\mu}_t, \boldsymbol{C}_t)$.

Furthermore, we only have to evaluate the mutual information for non-random 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 $I(r_{t+1}; \vec{\theta}|\vec{s}_{t+1}, \boldsymbol{s_{1:t}}, \boldsymbol{r_{1:t}})$ [99, 114]. This property means we can focus on the simpler problem of efficiently evaluating $I(r_{t+1}; \vec{\theta}|\vec{s}_{t+1}, \boldsymbol{s_{1:t}}, \boldsymbol{r_{1:t}})$ 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,

$$I(\vec{\theta}; r_{t+1}|\vec{s}_{t+1}, \boldsymbol{s_{1:t}}, \boldsymbol{r_{1:t}}) = H(p(\vec{\theta}|\boldsymbol{s_{1:t}}, \boldsymbol{r_{1:t}})) - E_{\vec{\theta}|\vec{\mu}_t, \boldsymbol{C}_t} E_{r_{t+1}|\vec{\theta}, \vec{s}_{t+1}} H(p(\vec{\theta}|\boldsymbol{s_{1:t+1}}, \boldsymbol{r_{1:t+1}})). \tag{25}$$

The first term, $H(p(\vec{\theta}|\boldsymbol{s_{1:t}}, \boldsymbol{r_{1:t}}))$, measures our uncertainty at time t. Since $H(p(\vec{\theta}|\boldsymbol{s_{1:t}}, \boldsymbol{r_{1:t}}))$ 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, $p(\vec{\theta}|\boldsymbol{s_{1:t+1}}, \boldsymbol{r_{1:t+1}})$, 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} [99, 24]. 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, $p(\vec{\theta}|\vec{\mu}_t, \boldsymbol{C}_t)$.

We compute the posterior entropy, $H(p(\vec{\theta}|\boldsymbol{s_{1:t+1}}, \boldsymbol{r_{1:t+1}}))$, as a function of r_{t+1} by first approximating $p(\vec{\theta}|r_{t+1}, \vec{s}_{t+1})$ as Gaussian. The entropy of a Gaussian is easy to compute [31]:

$$H(p(\vec{\theta}|\boldsymbol{s_{1:t+1}}, \boldsymbol{r_{1:t+1}})) \approx H(p(\vec{\theta}|\vec{\mu}_{t+1}, \boldsymbol{C}_{t+1})) \tag{26}$$

$$= \frac{1}{2}\log|\boldsymbol{C}_{t+1}| + const. \tag{27}$$

According to our update rule,

$$\boldsymbol{C}_{t+1} = \boldsymbol{C}_t - \frac{\boldsymbol{C}_t\vec{s}_{t+1}D(r_{t+1}, \rho_{t+1})\vec{s}_{t+1}^T\boldsymbol{C}_t}{1 + D(r_{t+1}, \rho_{t+1})\vec{s}_{t+1}^T\boldsymbol{C}_t\vec{s}_{t+1}} \tag{28}$$

$$\rho_{t+1} = \vec{\theta}^T\vec{s}_{t+1}. \tag{29}$$

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

$$J_{obs}(r_{t+1}, \vec{s}_{t+1}, \vec{\theta}) = -\frac{\partial^2 \log p(r_{t+1}|\rho_{t+1})}{\partial\rho^2}\vec{s}_{t+1}\vec{s}_{t+1}^T \tag{30}$$

as a random variable since it is a function of $\vec{\theta}$. We then estimate our expected uncertainty as the expectation of $H(p(\vec{\theta}|\vec{\mu}_{t+1}, \boldsymbol{C}_{t+1}))$ with respect to $\vec{\theta}$ using the posterior at time t. The mutual information, Eqn. 25, 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,

$$\rho_{t+1} \approx \vec{\mu}_{t+1}^T \vec{s}_{t+1} \tag{31}$$

where $\vec{\mu}_{t+1}$ is computed using Eqns. 14 & 15. 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 [99, 25, 29]. 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 why 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 2.6.2). 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 [95, 24]. 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 which are informative under all models with high probability as opposed to inputs which are informative only if $\vec{\theta} = \vec{\mu}_t$.

To compute the mutual information, Eqn. 25, we need to evaluate a high-dimensional expectation over the joint distribution on $(\vec{\theta}, r_t)$. Evaluating this expectation is

tractable because 1) we approximate the posterior as a Gaussian distribution and 2) 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 $|I + \vec{w}\vec{z}^T| = 1 + \vec{w}^T\vec{z}$ to evaluate the entropy at time $t+1$,

$$|\boldsymbol{C}_{t+1}| = |\boldsymbol{C}_t| \left| I - \frac{\vec{s}_{t+1} D(r_{t+1}, \rho_{t+1}) \vec{s}_{t+1}^T \boldsymbol{C}_t}{1 + D(r_{t+1}, \rho_{t+1}) \vec{s}_{t+1}^T \boldsymbol{C}_t \vec{s}_{t+1}} \right| \tag{32}$$

$$= |\boldsymbol{C}_t| \cdot \left(1 + D(r_{t+1}, \rho_{t+1}) \sigma_\rho^2\right)^{-1} \tag{33}$$

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

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

Consequently,

$$E_{\vec{\theta}|\vec{\mu}_t, \boldsymbol{C}_t} E_{r_{t+1}|\vec{s}_{t+1}, \vec{\theta}} H(p(\vec{\theta}|\vec{\mu}_{t+1}, \boldsymbol{C}_{t+1})) \tag{34}$$

$$= -\frac{1}{2} E_{\vec{\theta}|\vec{\mu}_t, \boldsymbol{C}_t} E_{r_{t+1}|\vec{s}_{t+1}, \vec{\theta}} \log\left(1 + D(r_{t+1}, \rho_{t+1}) \sigma_\rho^2\right) + const. \tag{35}$$

We can evaluate Eqn. 35 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,

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

$$= -\frac{1}{2} E_{\rho_{t+1}|\vec{\mu}_{t+1}, \boldsymbol{C}_{t+1}} E_{r_{t+1}|\rho_{t+1}} \log\left(1 + D(r_{t+1}, \rho_{t+1}) \sigma_\rho^2\right) \tag{36}$$

Since $\rho_{t+1} = \vec{\theta}^T \vec{s}_{t+1}^T$ and $p(\vec{\theta}|\vec{\mu}_t, \boldsymbol{C}_t)$ is Gaussian, ρ_{t+1} is a 1-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,

$$I(r_{t+1}; \vec{\theta}|\vec{s}_{t+1}, \boldsymbol{s_{1:t}}, \boldsymbol{r_{1:t}}) \approx E_{\rho_{t+1}|\mu_\rho, \sigma_\rho^2} E_{r_{t+1}|\rho_{t+1}} \log\left(1 + D(r_{t+1}, \rho_{t+1}) \sigma_\rho^2\right) + const \tag{37}$$

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

The right hand side of Eqn. 37 is an approximation of the mutual information because the posterior is not in fact Gaussian.

Eqn. 37 is a fairly intuitive metric for rating the informativeness of different designs. To distinguish between different 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 2nd 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 Eqn. 37, we have reduced the mutual information to a 2-dimensional integration over ρ_{t+1} and r_{t+1} which depends on $(\mu_\rho, \sigma_\rho^2)$. While 2-d 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 2-d 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 Eqn. 37, 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 Eqn. 37 to

$$E_{\rho_{t+1}|\mu_\rho,\sigma_\rho^2} E_{r_{t+1}|\rho_{t+1}} \log\left(1 + D(r_{t+1}, \rho_{t+1})\sigma_\rho^2\right) \approx E_{\rho_{t+1}|\mu_\rho,\sigma_\rho^2} E_{r_{t+1}|\rho_{t+1}} D(r_{t+1}, \rho_{t+1})\sigma_\rho^2, \tag{38}$$

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 2.7). Consequently, $\sigma_\rho^2 \to 0$ as $t \to \infty$. Therefore, if $D(r_{t+1}, \rho_{t+1})$ is bounded, then asymptotically $D(r_{t+1}, \rho_{t+1})\sigma_\rho^2 \to 0$.

2.4.1 Special case: exponential nonlinearity

When the nonlinear function $f()$ is the exponential function, we can derive an analytical approximation for the mutual information, Eqn. 37, 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 [26, 118, 131]. 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 the response 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 2nd derivative of the log-likelihood with respect to $\vec{\theta}$ is independent of r_{t+1},

$$D(r_{t+1}, \rho_{t+1}) = \exp(\rho_{t+1}). \tag{39}$$

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

By eliminating the expectation over r_{t+1} and using the linear approximation $\log(1+x) = x + o(x)$, we can simplify Eqn. 37,

$$E_{\rho_{t+1}|\mu_\rho,\sigma_\rho^2} E_{r_{t+1}|\rho_{t+1}} \log\left(1 + D(r_{t+1},\rho_{t+1})\sigma_\rho^2\right) = E_{\rho_{t+1}|\mu_\rho,\sigma_\rho^2} \log\left(1+\exp(\rho_{t+1})\sigma_\rho^2\right) + const. \tag{40}$$

$$= E_{\rho_{t+1},\mu_\rho,\sigma_\rho^2} \log\left(1+\exp(\rho_{t+1})\sigma_\rho^2\right) \tag{41}$$

$$\approx E_{\rho_{t+1}|\mu_\rho,\sigma_\rho^2} \exp(\rho)\sigma_\rho^2. \tag{42}$$

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

$$E_{\rho_{t+1}|\mu_\rho,\sigma_\rho^2} \exp(\rho_{t+1})\sigma_\rho^2 = \sigma_\rho^2 \exp\left(\mu_\rho + \frac{1}{2}\sigma_\rho^2\right). \tag{43}$$

Our objective function is increasing with μ_ρ and σ_ρ^2. In Section 2.5.2, we show that this property makes optimizing the design for an exponential nonlinearity particularly tractable.

2.4.2 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 [99, 24]. It is enlightening to re-derive 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

$$r_t = \vec{\theta}^T \vec{s}_t + \epsilon, \tag{44}$$

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,

$$\log p(r_t|\vec{s}_t,\vec{\theta},\sigma^2) = -\frac{1}{2\sigma^2}(r_t - \vec{\theta}^T\vec{s}_t)^2 + const \tag{45}$$

$$= -\frac{1}{2\sigma^2} r_t^2 + \frac{1}{\sigma^2}\rho_t r_t - \frac{1}{2\sigma^2}\rho_t^2 + const. \tag{46}$$

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 [25]. Mathematically this means the observed Fisher information $D(r_{t+1}, \rho_{t+1})$ is a constant equal to the reciprocal of the variance,

$$D(r_{t+1}, \rho_{t+1}) = \frac{1}{\sigma^2}. \tag{47}$$

Plugging $D(r_{t+1}, \rho_{t+1})$ into Eqn. 37, we obtain the simple result

$$E_{\vec{\theta}|\vec{\mu}_t, \boldsymbol{C}_t} E_{r_{t+1}|\vec{\theta}, \vec{s}_{t+1}} I(r_{t+1}; \vec{\theta} | \vec{s}_{t+1}, \boldsymbol{s_{1:t}}, \boldsymbol{r_{1:t}}) = \log\left(1 + \frac{\sigma_\rho^2}{\sigma^2}\right) + const. \tag{48}$$

Since σ^2 is a constant, we can only increase the mutual information 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

$$\boldsymbol{C}_t^{-1} = \boldsymbol{C}_0^{-1} + \sum_{i=1}^{t} \frac{1}{\sigma^2} \vec{s}_i \vec{s}_i^T. \tag{49}$$

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 [99].

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.

2.5 Choosing the optimal stimulus

The simple expression for the conditional mutual information, Eqn. 37, means we can find the optimal stimulus by solving the following simple program,

$$1) \quad (\mu_\rho, \sigma_\rho^2)^* = \text{argmax}_{(\mu_\rho,\sigma_\rho^2)\in\mathcal{R}_{t+1}} E_{\rho_{t+1}|\mu_\rho,\sigma_\rho^2} E_{r_{t+1}|\rho_{t+1}} \log\left(1 + D(r_{t+1},\rho_{t+1})\sigma_\rho^2\right) \quad (50)$$

$$\mathcal{R}_{t+1} = \left\{(\mu_\rho, \sigma_\rho^2) : \mu_\rho = \vec{\mu}_t^T \vec{s}_{t+1} \;\&\; \sigma_\rho^2 = \vec{s}_{t+1}^T \boldsymbol{C}_t \vec{s}_{t+1},\ \forall \vec{s}_{t+1} \in \mathcal{S}_{t+1}\right\} \quad (51)$$

$$\mathcal{S}_{t+1} = \left\{\vec{s}_{t+1} : \vec{s}_{t+1} = \left[\vec{x}_{t+1}^T, \vec{s}_{f,t+1}^T\right]^T, \vec{x}_{t+1} \in \mathcal{X}_{t+1}\right\} \quad (52)$$

$$2) \quad \text{find} \quad \vec{s}_{t+1} \quad \text{s.t} \quad {\mu_\rho}^* = \vec{\mu}_t^T \vec{s}_{t+1} \quad {\sigma_\rho^2}^* = \vec{s}_{t+1}^T \boldsymbol{C}_t \vec{s}_{t+1}. \quad (53)$$

$\mathcal{R}_{t+1}$ is the range of the mapping $\vec{s}_{t+1} \rightarrow (\mu_\rho, \sigma_\rho^2)$ corresponding to the stimulus domain, $\mathcal{X}_{t+1}$. Once we have computed $\mathcal{R}_{t+1}$, we need to solve a highly tractable 2-d 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 $|\hat{\mathcal{X}}_{t+1}|$. 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 $\mathcal{X}_{t+1}$ which 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; i.e. $\mathcal{X}_{t+1}$ 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.

2.5.1 Optimizing over a finite set of stimuli

Our first method simultaneously addresses two issues 1) how to deal with arbitrary stimulus domains and 2) 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 which 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 only depends on $(\mu_\rho, \sigma_\rho^2)$. 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 2-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

which tend to be dispersed throughout $\mathcal{R}_{t+1}$. We delay until sections 2.5.3 & 2.6.1 the presentation of the heuristics that we used in our simulations so that we can first introduce the specific problems and the stimulus domains for which these heuristics are suited.

2.5.2 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[2] $||\vec{x}_{t+1}||_2 \leq m$. This is an important stimulus domain because of its connection to white noise which is often used to study sensory systems [49, 30, 26, 36, 170]. 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 only applies 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 which includes many important subsets of stimuli such as random dot patterns [44].

Our main result is a simple, efficient procedure for finding the boundary of $\mathcal{R}_{t+1}$ as a function of a 1-d 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

[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.

[12],

$$\mathcal{R}_{t+1} = \Big\{(\mu_\rho, \sigma_\rho^2) : \left(-m||\vec{\mu}_{x,t}||_2 + \vec{s}_{f,t+1}^T\vec{\mu}_{f,t}\right) \leq \mu_\rho \leq \left(m||\vec{\mu}_{x,t}||_2 + \vec{s}_{f,t+1}^T\vec{\mu}_{f,t}\right), \\ \sigma^2_{\rho,\min}(\mu_\rho) \leq \sigma_\rho^2 \leq \sigma^2_{\rho,\max}(\mu_\rho)\Big\} \tag{54}$$

$$\sigma^2_{\rho,\max}(\mu_\rho) = \max_{\vec{x}_{t+1}} \sigma_\rho^2 \quad s.t \quad \mu_\rho = \vec{\mu}_t^T\vec{s}_{t+1} \quad \& \quad ||\vec{x}_{t+1}||_2 \leq m \tag{55}$$

$$\sigma^2_{\rho,\min}(\mu_\rho) = \min_{\vec{x}_{t+1}} \sigma_\rho^2 \quad s.t \quad \mu_\rho = \vec{\mu}_t^T\vec{s}_{t+1} \quad \& \quad ||\vec{x}_{t+1}||_2 \leq m. \tag{56}$$

By solving Eqns. 55 & 56, we can walk along the curves which 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 which 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 Eqns. 55 & 56. To eliminate the linear constraint, we derive an alternative quadratic expression for σ_ρ^2 in terms of $\vec{x}_{t+1}$,

$$\sigma_\rho^2 = \vec{x}_{t+1}^T A\vec{x}_{t+1} + \vec{b}(\alpha)^T\vec{x}_{t+1} + d(\alpha). \tag{57}$$

Here we only discuss the most important points regarding Eqn. 57; the derivation and definition of the terms are provided in Appendix 2.10.1. The linear term of this modified quadratic expression ensures that the value of this quadratic 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 quadratic 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 Eqn. 57 subject to the quadratic constraint $||\vec{x}_{t+1}||_2 \leq$

m we use the Karush-Kuhn-Tucker (K.K.T.) conditions. For these optimization problems, it can be proved that the K.K.T. are necessary and sufficient [61]. To compute the boundary of $\mathcal{R}_{t+1}$ as a function of μ_ρ, we need to solve the K.K.T. 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 2.10.1. This approach is faster because to compute μ_ρ as a function of the Lagrange multiplier, we only need to find the root of a 1-d quadratic expression.

To solve the K.K.T. conditions we need the eigendecomposition of A. Computing the eigendecomposition of A is the most expensive operation and in the worst case, requires $O(\dim(\vec{\theta})^3)$ operations. A, however, is a rank-2 perturbation of $\boldsymbol{C}_t$, Eqn. 107. 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 [68], as discussed in the next section. The key point is that we can on average compute the eigendecomposition in $O(\dim(\vec{\theta})^2)$ time.

Having computed $\mathcal{R}_{t+1}$, we can perform a 2-d 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 just find a linear combination

of these two vectors which yields σ_ρ^{2*},

$$\text{find} \quad \gamma \quad \text{s.t} \quad \sigma_\rho^{2*} = \vec{s}_{t+1}(\gamma)^T \boldsymbol{C}_t \vec{s}_{t+1}(\gamma) \tag{58}$$

$$\vec{s}_{t+1}(\gamma) = (1-\gamma)\vec{s}_{\min}({\mu_\rho}^*) + \gamma \vec{s}_{\max}({\mu_\rho}^*) \quad \gamma \in [0,1]. \tag{59}$$

All $\vec{s}_{t+1}(\gamma)$ necessarily satisfy the power constraint because the power constraint defines a convex set and $\vec{s}_{t+1}(\gamma)$ is a linear combination of two stimuli in this set. Similar reasoning guarantees $\vec{s}_{t+1}(\gamma)$ has projection ${\mu_\rho}^*$ on $\vec{\mu}_t$. While 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 only need to consider $\sigma_{\rho,max}^2(\mu_\rho)$ for each possible value of μ_ρ. Consequently, a one-dimensional search over $\sigma_{\rho,max}^2(\mu_\rho)$ for

$$\mu_\rho \in \left[-m||\vec{\mu}_{x,t}||_2 + \vec{s}_{f,t+1}^T \vec{\mu}_{f,t}, m||\vec{\mu}_{x,t}||_2 + \vec{s}_{f,t+1}^T \vec{\mu}_{f,t}\right] \tag{60}$$

is sufficient for finding the optimal input. A sufficient condition for guaranteeing 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 2.10.2). An important example satisfying this condition is $f(\rho_{t+1}) = \exp(\rho_{t+1})$, which satisfies the convexity condition because

$$\frac{\partial^2 \log(1+D(r_{t+1},\rho_{t+1})\sigma_\rho^2)}{\partial \rho_{t+1}^2} = \frac{\exp(\rho_{t+1})\sigma_\rho^2}{(1+\exp(\rho_{t+1})\sigma_\rho^2)^2} > 0. \tag{61}$$

2.5.3 Heuristics for the power constraint

Even though we can compute $\mathcal{R}_{t+1}$ when $\mathcal{X}_{t+1} = \{\vec{x}_{t+1} : ||\vec{x}_{t+1}||_2 \leq m\}$, 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

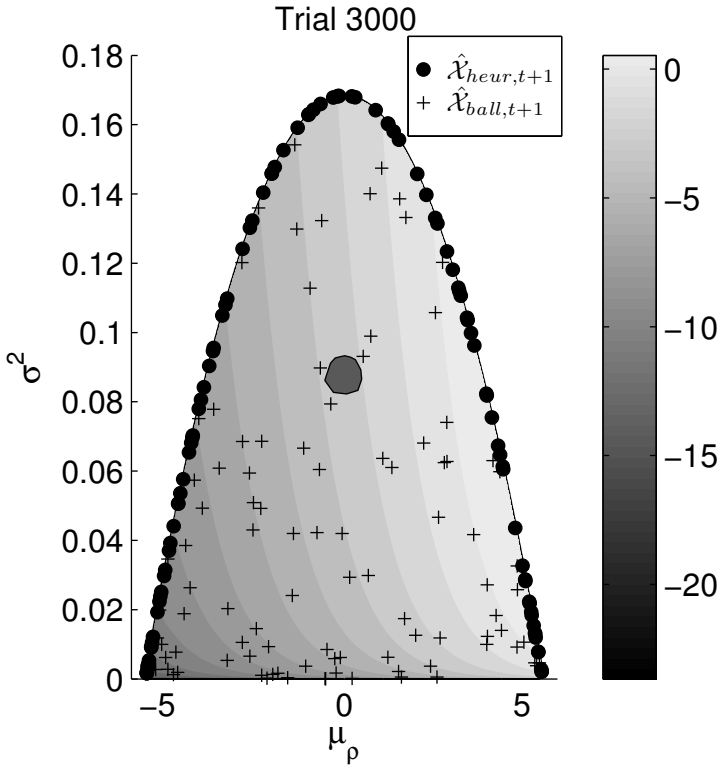


Figure 4: A plot showing $\mathcal{R}_{t+1}$, Eqn. 54. The grayscale indicates the objective function, Eqn. 37. The dots and crosses show the points corresponding to the stimuli in $\hat{\mathcal{X}}_{heur,t+1}$ and $\hat{\mathcal{X}}_{ball,t+1}$ respectively. The dark grey region centered at $\mu_\rho = 0$ 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 which are parallel and anti-parallel 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 1st row and 3rd column of Figure 5).

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 $\mathcal{X}_{t+1} = \{\vec{x}_{t+1} : ||\vec{x}_{t+1}||_2 \leq m\}$ 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 $\vec{g}_{\perp} = \frac{\vec{g}_{\max} - \frac{\vec{\mu}_{x,t}^T}{||\vec{\mu}_{x,t}||_2} \vec{g}_{\max}}{||\vec{g}_{\max} - \frac{\vec{\mu}_{x,t}^T}{||\vec{\mu}_{x,t}||_2} \vec{g}_{\max}||_2}$. $\vec{g}_{\max}$ is the maximum eigenvector of $\boldsymbol{C}_{x,t}$.

This procedure tends to produce a set of stimuli which 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 which map to the boundaries of $\mathcal{R}_{t+1}$. Nonetheless, since this heuristic produces stimuli which tend to be dispersed throughout $\mathcal{R}_{t+1}$, we can usually find a stimulus in $\hat{\mathcal{X}}_{ball,t+1}$ which 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 only pick stimuli which 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 which 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 $\frac{E_{\vec{x}_{t+1}}(||\vec{x}_{t+1}||_2^2)}{\dim(\vec{\theta})}$. As a result, the variance of μ_ρ, $||\vec{\mu}_t||_2^2 \frac{E_{\vec{x}_{t+1}}(||\vec{x}_{t+1}||_2^2)}{\dim(\vec{\theta})}$, decreases as $1/\dim(\vec{\theta})$, ensuring that for high-dimensional systems the stimuli in $\hat{\mathcal{X}}_{iid,t+1}$ have μ_ρ close to zero with high probability, Figure 4. Uniformly sampling the ball or sphere, therefore, does a poor job of selecting stimuli which are dispersed throughout $\mathcal{R}_{t+1}$. As a result, $\hat{\mathcal{X}}_{iid,t+1}$ is unlikely to contain stimuli which are close to being maximally informative.

2.5.4 Simulation results

We tested our algorithm using computer simulations which roughly emulated typical neurophysiology experiments. The main conclusion of our simulations is that using our information maximizing design, we can reduce by an order of magnitude the number of trials needed to estimate $\vec{\theta}$ [114]. 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 10m and 1sec 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 [126]. We generated synthetic responses by sampling Eqn. 7 with $\vec{\theta}$ set to a 40×40 Gabor patch. The nonlinearity was the exponential function.

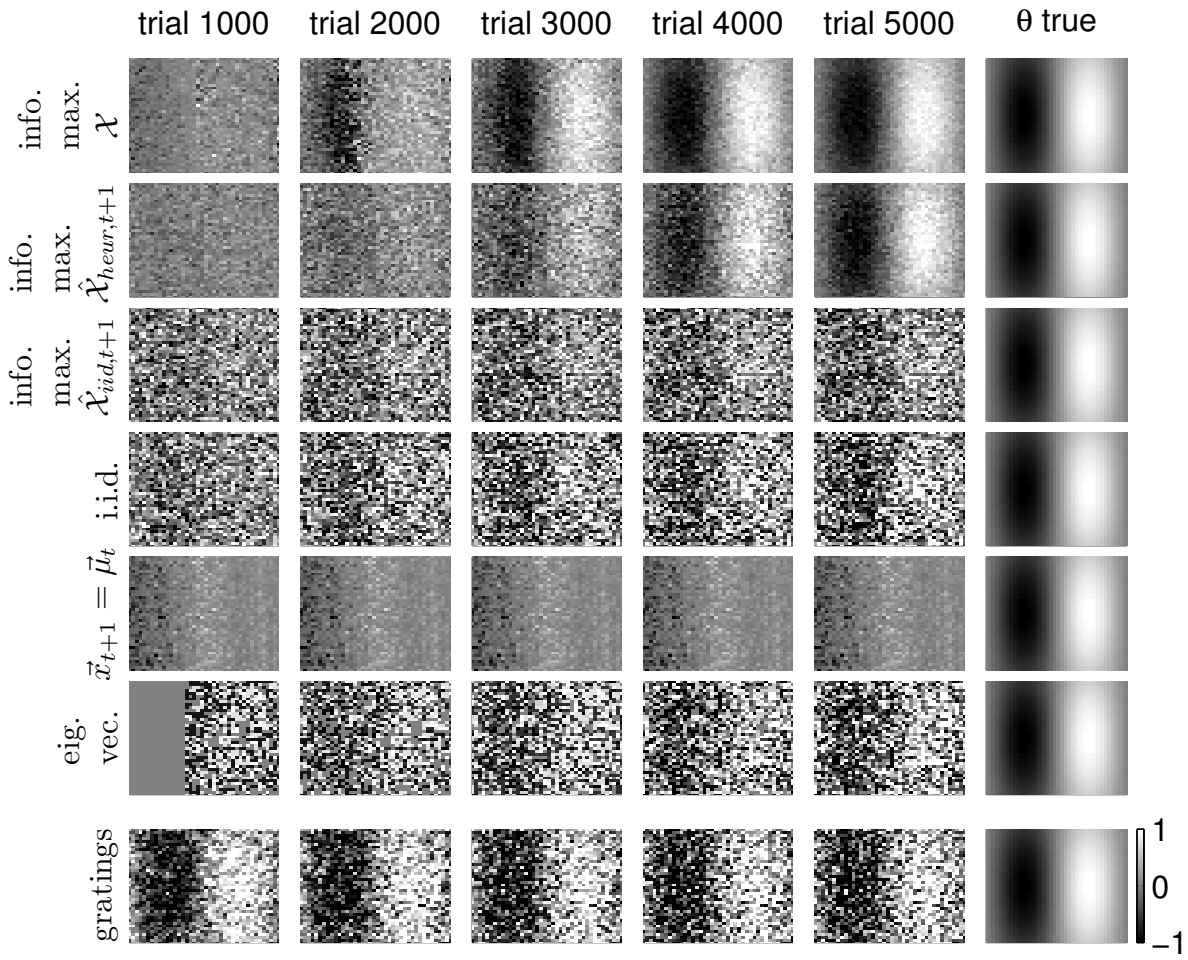


Figure 5: The receptive field, $\vec{\mu}_t$, of a simulated neuron estimated using different designs. The neuron's receptive field $\vec{\theta}$ was the 40x40 Gabor patch shown in the last column (spike history effects were set to zero for simplicity, $\vec{\theta}_f = 0$). The stimulus domain was 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 a 1000 stimuli. The final 4 rows show the results for an i.i.d. design, a design which set $\vec{x}_{t+1} = \vec{\mu}_t$, a design which set the stimulus to the maximum eigenvector of $\boldsymbol{C}_t$, and a design which 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.

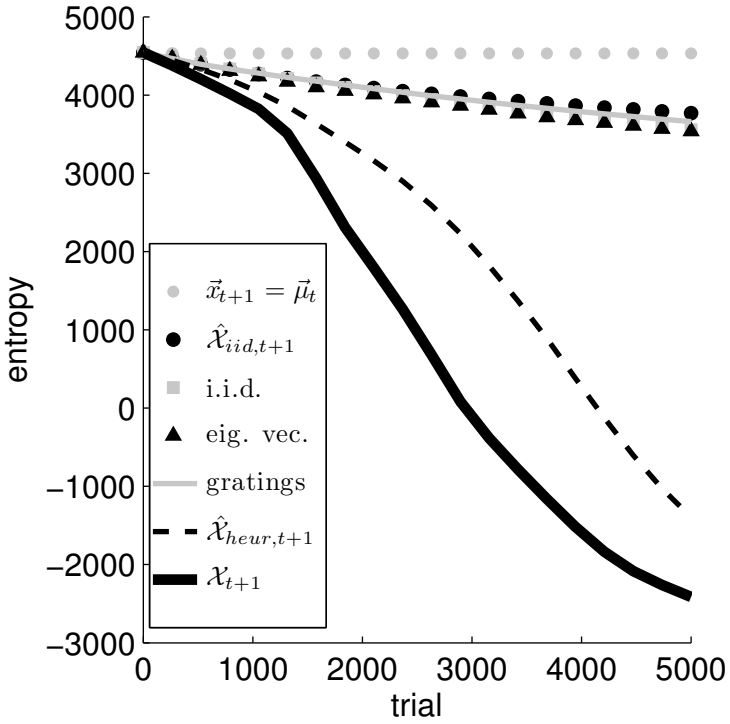


Figure 6: The posterior entropies for the simulations shown in Fig. 5. Picking the optimal input from $\mathcal{X}_{t+1}$ decreases the entropy much faster than restricting ourselves to a subset of $\mathcal{X}_{t+1}$. However if we pick a subset of stimuli using our heuristic, then we can decrease the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

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 1) all info. max. designs do better than an i.i.d. design and 2) an info. max. design which optimizes over the full domain of the input, $\mathcal{X}_{t+1} = \{\vec{x}_{t+1} : ||\vec{x}_{t+1}||_2 = m\}$, does much better then choosing the best stimulus in a subset constructed by uniformly sampling $\mathcal{X}_{t+1}$.

The results in Figure 5 and Figure 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 2.5.3. 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 which 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 re-using 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 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 info. max. designs to the limiting cases where we put all stimulus energy along the mean or maximum eigenvector, Figure 5 and Figure 6. Putting all energy along the maximum eigenvector performs nearly as well as an i.i.d. design. Our update, Eqn. 20, ensures that if the stimulus is an eigenvector of $\boldsymbol{C}_t$ then the updated covariance matrix is just the result of shrinking the eigenvalue corresponding to that eigenvector. Consequently, setting the stimulus to the max eigenvector ends up scanning through the different eigenvectors on successive trials. The resulting sequence of stimuli is statistically similar to that of an i.i.d. design

because 1) the stimuli are highly uncorrelated with each other and 2) the stimuli are highly uncorrelated 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 only varies with $\rho_{t+1} = \vec{\theta}^T \vec{s}_{t+1}$ both methods pick stimuli which are roughly equally informative. Consequently, both designs end up shrinking the posterior entropy at very 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 which used sinusoidal gratings as the stimuli. In Figure 5, this design produces an estimate of $\vec{\theta}$ which already has the basic inhibitory/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 because 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. This spatial correlation explains why the coarse inhibitory/excitatory pattern of the receptive field appears after so few trials. However, the spatial correlation of the stimuli 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 [114] used a brute force computation and optimization of the mutual information. The computation in [114] was possible only because $\vec{\theta}$ was assumed to be a Gabor function specified by just three parameters (the 2-d 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 40x40 image with power m^2. Attempting to use brute force in this high dimensional space would have been hopeless. Our results show that 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, $\vec{\theta}_x$, which 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 between 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 info. max. 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, $\vec{\theta} = \left[\vec{\theta}_x^T, \vec{\theta}_f^T\right]^T$, consists of the stimulus coefficients $\vec{\theta}_x$, which were a 1-d 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 info. max. 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 (M.S.E.) on all trials. In Figure 7 the M.S.E. increases on roughly the first 100 trials because the mean of the prior is zero. The data collected on these early trials tends 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 M.S.E..

By converging more rapidly to the stimulus coefficients, the info. max. 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

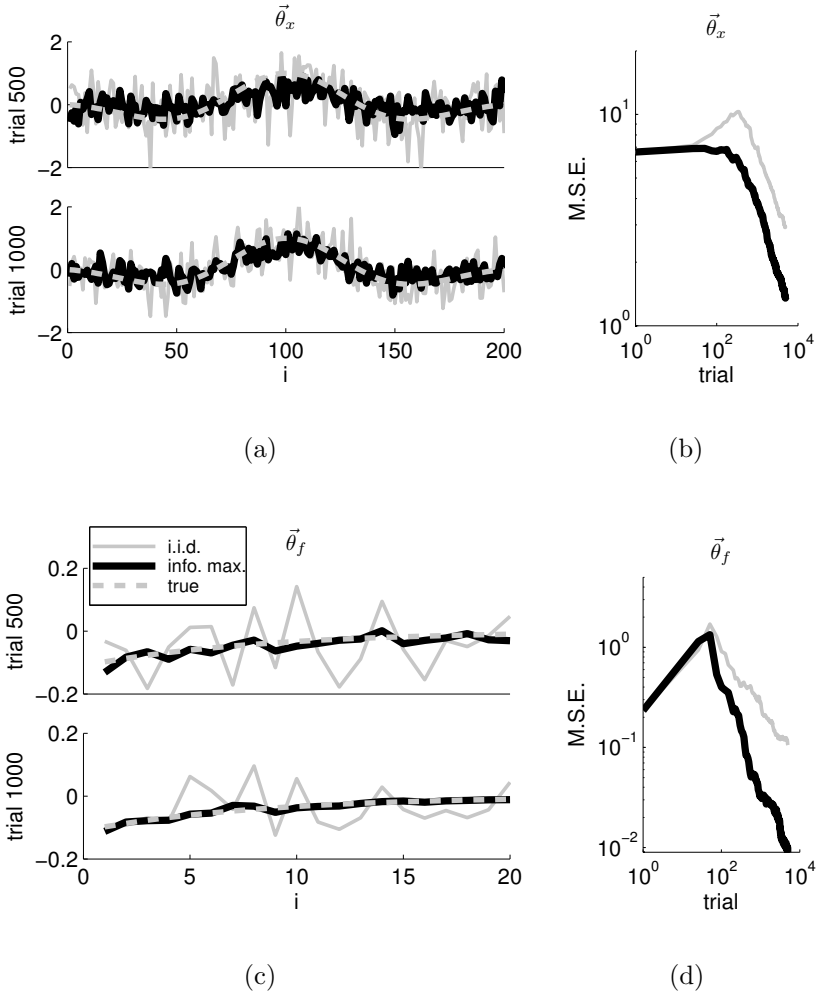


Figure 7: A comparison of parameter estimates using an info. max. design vs. an i.i.d.design for a neuron whose conditional intensity depends on both the stimulus and the spike history. a) The estimated stimulus coefficients $\vec{\theta}_x$, after 500 and 1000 trials, for the true model (dashed grey), info max design (solid black), and an i.i.d.design (solid grey). b) The mean squared error (M.S.E.) of the estimated stimulus coefficients for the info max. design (solid black line) and the i.i.d. design (solid grey line). c) The estimated spike-history coefficients, $\vec{\theta}_f$, after 500 and 1000 trials. d) The M.S.E of the estimated spike-history coefficients.

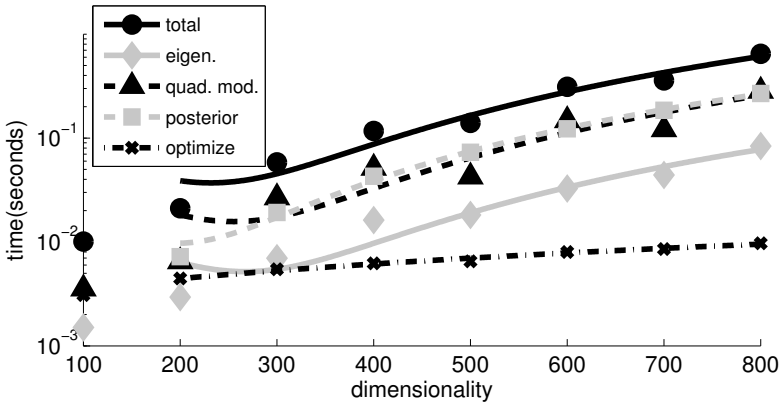
$\vec{\theta}_x$ and $\vec{\theta}_f$ which is given by $\boldsymbol{C}_{x,f}$ in Eqn. 99. Consider a simple example where the first entry of $\boldsymbol{C}_{x,f}$ is negative and the remaining entries are zero. In this example θ_{x_1} and θ_{f_1} (the first components of $\vec{\theta}_x$ and $\vec{\theta}_f$ respectively) would be anti-correlated. 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 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}.

2.5.4.1 Running time

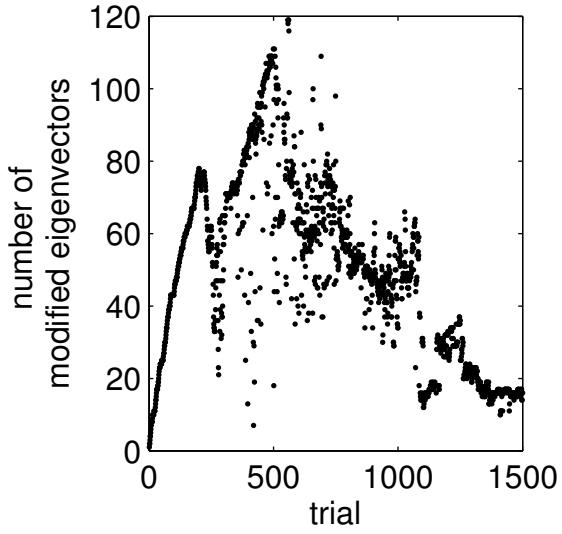
Our algorithm is suited to high-dimensional, real-time applications because it reduces the exponential complexity of choosing the optimal design to on average quadratic and at worst cubic running time. We verified this claim empirically by measuring the running time for each step of the algorithm as a function of the dimensionality of $\vec{\theta}$, Figure 8(a)[3]. These simulations used a GLM with an exponential link function. This nonlinearity leads to a special case of our algorithm because 1) we can derive an analytical approximation of our objective function, Eqn. 43, and 2) only a 1-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 $\dim(\vec{\theta})$. 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 $\dim(\vec{\theta})$ because $\mathcal{R}_{t+1}$ is always 2-d. Similarly, the increased complexity of a full 2-d search compared to a 1-dimensional search in $\mathcal{R}_{t+1}$ is independent of $\dim(\vec{\theta})$.

The main conclusion of Figure 8(a) 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

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



(a)



(b)

Figure 8: a) The 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 Eqn. 107 (quad. mod.), 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 eigen decomposition of the posterior covariance on average grows quadratically because many of our eigenvectors remain unchanged by the rank one perturbation. We verified this claim empirically for one simulation by plotting the number of modified eigenvectors as a function of the trial. The data is from a $20x10$ Gabor simulation.

time of the 4 steps that make up our 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 (that is finding the eigendecomposition of A in Eqn. 107) 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 which maps to $(\mu_\rho^*, \sigma_\rho^{2*})$. The running time of these operations scale as $O(\dim(\vec{\theta}))$ because computing $\sigma_{\rho,\max}^2$ as a function of λ requires summing $\dim(\vec{\theta})$ terms, Eqn. 113. When $\vec{\theta}$ was 100 dimensions, the total running time was about 10ms 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 $O(\dim(\vec{\theta})^3)$. Figure 8(a), however, shows that the average running time of the eigendecomposition only grows quadratically with the dimensionality. The average running time grows as $O(\dim(\vec{\theta})^2)$ 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,

$$M' = M + \vec{z}\vec{z}^T, \tag{62}$$

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

$$M'\vec{g} = M\vec{g} + \vec{z}\vec{z}^T\vec{g} = M\vec{g} = c\vec{s}, \tag{63}$$

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 $\dim(\vec{\theta})$ eigenvalues and eigenvectors, thereby achieving on average quadratic running time [68, 41, 135]. Asymptotically, we can prove that the perturbation is correlated with at most 2 eigenvectors, Section 2.7. 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 8(b) we plot, for one simulation, the number of eigenvectors which are perturbed by the rank 1 modification. On most trials fewer than $\dim(\vec{\theta})$ eigenvectors are perturbed by the update. These results rely to some extent on the fact that our prior covariance matrix was white and hence had only 1 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.

2.6 Important extensions

In this section we consider two extensions of the basic GLM which expands the range of neurophysiology experiments to which we can apply our algorithm. The two extensions are: 1) handling nonlinear transformations of the input and 2) 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.

2.6.1 Input nonlinearities

Neurophysiologists routinely record from neurons which are not primary sensory neurons. In these experiments, the input to a neuron is a nonlinear function of the

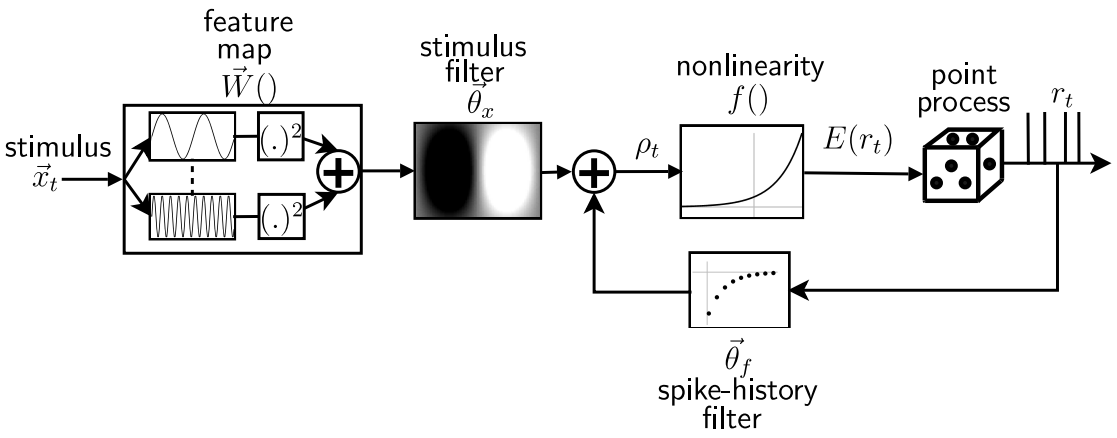


Figure 9: A GLM in which we first transform the input into some feature space defined by the nonlinear functions $W_i(\vec{x}_t)$ which in this case are squaring functions.

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 [170, 2, 116]. The only difference from the original GLM is how we define the input,

$$\vec{s}_t = [W_1(\vec{x}_t), \ldots, W_{n_w}(\vec{x}_t), r_{t-1}, \ldots, r_{t-t_a}]^T . \tag{64}$$

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 will 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 which 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 [131]. 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 [37, 117, 91, 143]. NLN models can be used to model this spectral decomposition of the auditory input, as well as the subsequent integration of information across frequency [65]. 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 [1, 36, 124]. In audition, energy models are used to explain frequency integration and phase insensitivity in auditory processing [65, 20].

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,

$$E(r_t) = f\left(\sum_i (\vec{\phi}^{i,T}\vec{x}_t)^2\right). \tag{65}$$

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 a nonlinear-linear-nonlinear cascade. We start by expressing the energy of each channel as a vector matrix multiplication by introducing the matrices Q^i,

$$(\vec{\phi}^{i,T}\vec{x}_t)^2 = \vec{x}_t^T\vec{\phi}^i\vec{\phi}^{i,T}\vec{x}_t = \vec{x}_t^T Q^i \vec{x}_t. \tag{66}$$

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 one matrix. Letting $Q = \sum_i Q^i$, we

can write the energy model as

$$E(r_t) = f(\vec{x}_t^T Q \vec{x}_t) = f(\sum_{i,j} Q_{i,j} x_{i,t} x_{j,t}) \tag{67}$$

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

where $x_{i,t}$ denotes the i^{th} 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 $[x_{1,t}x_{1,t}, ..., x_{i,t}x_{j,t}, ...]^T$. 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 2.5.2 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, $p(\vec{\theta}|\vec{\mu}_t, \boldsymbol{C}_t)$. 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 non-zero singular values should equal the subspace spanned by the true filters, $\vec{\phi}^i$.

Since we can only optimize the design with respect to a finite set of stimuli, we

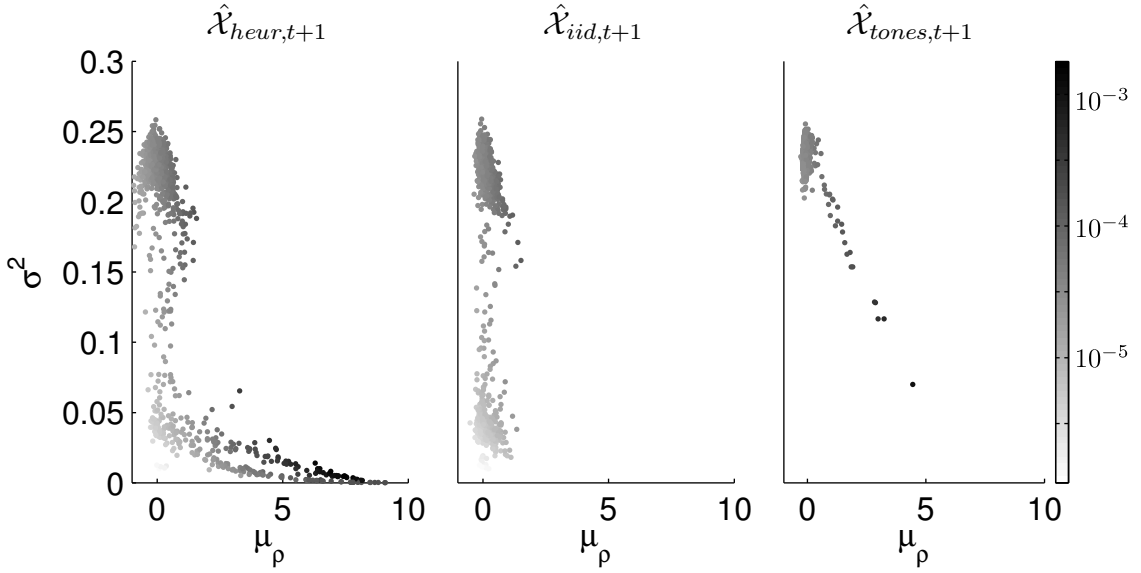


Figure 10: Plot shows the mapping of different stimulus sets into $\mathcal{R}_{t+1}$ after 500 trials. $\hat{\mathcal{X}}_{heur,t+1}$ consists of 1000 stimuli selected using the heuristic described in the text. $\hat{\mathcal{X}}_{iid,t+1}$ consists of 1000 stimuli randomly sampled from the sphere $||\vec{x}_{t+1}||_2 = m$. $\hat{\mathcal{X}}_{tones}$ 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 which picked the optimal stimulus in $\hat{\mathcal{X}}_{heur,t+1}$ on each trial. The shading of the dots is proportional to the the mutual information for each input, Eqn. 43. 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 $(\mu_\rho, \sigma_\rho^2)$ space.

devised a heuristic for making this set more dispersed throughout $\mathcal{R}_{t+1}$. For the energy model,

$$\mu_\rho = \vec{\mu}_t^T \vec{s}_{t+1} \tag{68}$$

$$= \sum_{i=1}^{n_w} \vec{\mu}_{i,t} W_i(\vec{x}_{t+1}) \tag{69}$$

$$= \vec{x}_{t+1}^T \hat{Q} \vec{x}_{t+1} \tag{70}$$

$$\hat{Q}_{i,j} = \vec{\mu}_{i+(j-1)\cdot \dim(\vec{x}),t} \tag{71}$$

where $\vec{\mu}_{i,t}$ is the i^{th} 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, $\vec{\nu}$, 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 $\vec{\nu}$ by uniformly sampling the $\dim(\vec{\theta}) - 1$ unit sphere orthogonal to $\vec{\nu}$.

4. We add the stimulus,

$$\vec{x} = \omega\vec{\nu} + \sqrt{m^2 - \omega^2}\vec{\omega} \tag{72}$$

   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 $\hat{Q}$, Eqn. 70.

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 which is more spread out on the range of μ_ρ. As a result, $\hat{\mathcal{X}}_{heur,t+1}$ contains more informative stimuli than $\hat{\mathcal{X}}_{iid,t+1}$.

2.6.1.1 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 which were identical except for a 90 degree difference in phase [37, 117]. We generated spikes by sampling a conditional Poisson process whose instantaneous, conditional firing rate was set by Eqn. 65 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 info. max. designs. The i.i.d. design uniformly sampled the stimulus from the sphere $||\vec{x}_{t+1}||^2 = m^2$. The two info. max. 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.

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 non-zero singular values of $\hat{Q}$.

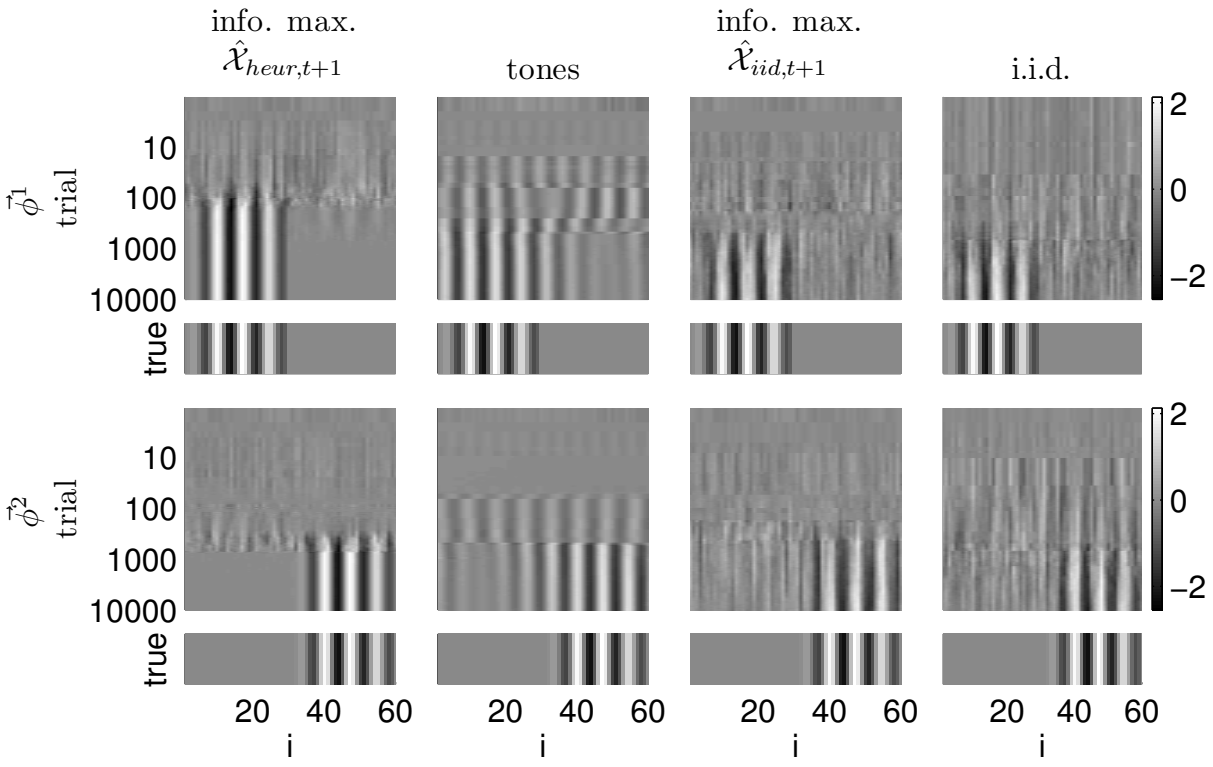


Figure 11: Simulation results for the hypothetical auditory neuron described in the text. Simulated responses were generated using Eqn. 65 with $\vec{\phi}^1$ and $\vec{\phi}^2$ being gammatone filters. These filters were identical except for the phase which differed by 90 degrees. The results compare an i.i.d. design, two info. max. designs, and a design using pure tones. The two info. max. 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 a 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 info. max. design using a heuristic does much better than an i.i.d. design. For this info. max. design, the gammatone structure of the two filters is evident starting around 100 and 500 trials respectively. By 1000 trials, the info max 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 only starting to be revealed after 1000 trials.

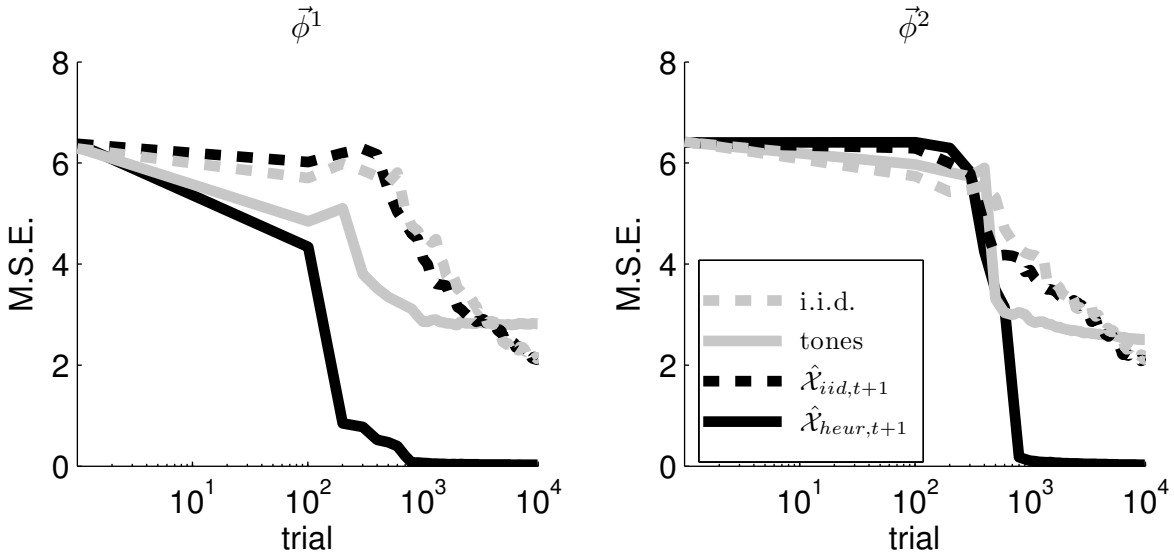


Figure 12: The mean squared error (M.S.E.) of the estimated filters shown in Figure 11. a) The M.S.E. of $\vec{\phi}^1$. b) The M.S.E. of $\vec{\phi}^2$. The solid black and dashed black lines show the results for designs which picked the optimal stimulus in $\hat{\mathcal{X}}_{heur,t+1}$ and $\hat{\mathcal{X}}_{iid,t+1}$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

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 & 12 show that by picking the optimal stimulus in $\hat{\mathcal{X}}_{heur,t+1}$, the MAP converges more rapidly to the true gammatone filters. In Figure 11, the design which 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 which 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 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 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 M.S.E. for the pure tones design does not decrease as fast as for the alternative designs.

Also evident in the info. max. results is the exploitation-exploration trade-off [79]. To increase the information about one of the expected filters, we need to pick stimuli which are correlated with this filter. Since the input filters are orthogonal and the stimulus power is constrained, we can only efficiently probe one filter at a time. The exploitation-exploration trade-off explains why on trials 100-500, 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 then exploiting our estimate of $\vec{\phi}^1$. Hence, the info. max. design picks stimuli orthogonal to the first gammatone filter, which eventually leads to us finding the second filter.

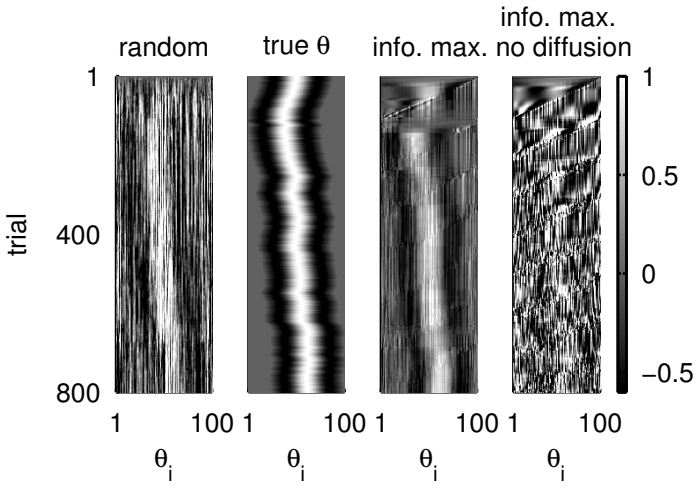
2.6.2 Time-varying $\vec{\theta}$

Neural responses often change slowly over the course of an experiment due to changes in the health, arousal, or attentive state of the preparation [88]. If we knew the underlying dynamics of $\vec{\theta}$ then we could try to model these changes. Unfortunately, incorporating arbitrary, nonlinear dynamical models of $\vec{\theta}$ into our information maximizing strategy is non-trivial 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 [54],

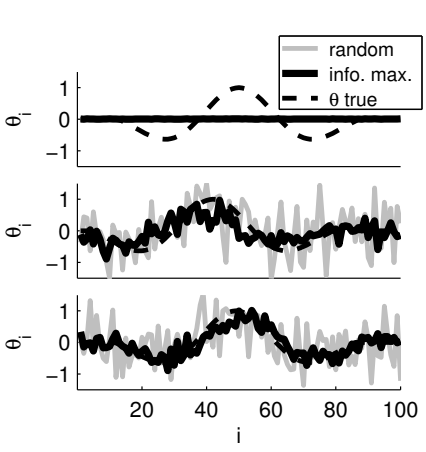
$$\vec{\theta}_{t+1} = \vec{\theta}_t + \vec{w}_t \tag{73}$$

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 our info. max. 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 info. max. algorithm would do a poor job of picking the optimal stimulus.

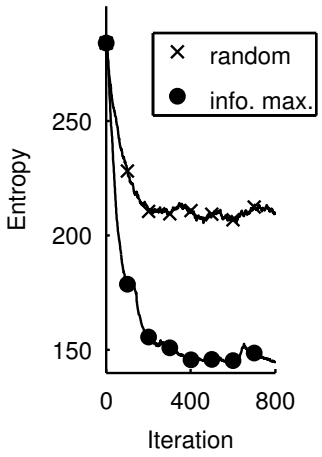
To update the posterior and choose the optimal stimulus, we use the procedures



(a)



(b)



(c)

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 in 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 info. max. 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.

described in Section 2.3 and Section 2.5. The only difference due to a time-varying $\vec{\theta}$ is that the covariance matrix of $p(\vec{\theta}_{t+1}|\boldsymbol{s_{1:t+1}}, \boldsymbol{r_{1:t+1}})$ is in general no longer just a rank-one modification of the covariance matrix of $p(\vec{\theta}_t|\boldsymbol{s_{1:t}}, \boldsymbol{r_{1:t}})$. Therefore, we cannot use the rank-one 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 i^{th} eigenvalue of $\boldsymbol{C}_t$; in this case, our methods may be applied without modification. In cases where we expect $\vec{\theta}$ 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 info. max. design to fit a GLM to a neuron whose receptive field drifts non-systematically with time. The receptive field was a 1-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). Even though only the center of $\vec{\theta}$ moved, we still modeled changes in $\vec{\theta}$ using Eqn. 73. The results demonstrate the benefits of using an information-maximization design to estimate a time varying $\vec{\theta}$. Even though we cannot reduce our uncertainty below a level determined by Π, the info. max. design can still improve our estimate of $\vec{\theta}$ compared to using random stimuli.

2.7 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 $I(r_{t+1}; \vec{\theta}|\boldsymbol{s_{1:t+1}}, \boldsymbol{r_{1:t}})$ 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.

The basis of this section is a central-limit like theorem for information maximizing designs proved in [114]. This theorem states that asymptotically the information maximizing design decreases our uncertainty at the same rate as a design which maximizes the expected Fisher information. This theorem uses the fact that the posterior of the information maximizing design is asymptotically normal, with mean and covariance

$$\vec{\mu}_t \xrightarrow{p} \vec{\theta} \tag{74}$$

$$(1/t)\boldsymbol{C}_t^{-1} \xrightarrow{p} E_{\vec{x}}(J_{exp}(\vec{\theta}, \vec{x})) \tag{75}$$

$$p_{opt}(\vec{x}) = \arg\max_{p(\vec{x})} \log |E_{\vec{x}}(J_{exp}(\vec{\theta}, \vec{x}))|. \tag{76}$$

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 which picks the stimulus by sampling the stimulus distribution which maximizes the expected Fisher information, Eqn. 76. This design is non-adaptive, i.e. independent of the data already observed, because unlike our information maximizing design, $p_{opt}(\vec{x})$ is independent of the posterior at time t. Asymptotically, the information maximizing 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, i.e. the expected information of each observation, Eqn. 76. Even though the info. max. design is asymptotically equivalent to $p_{opt}(\vec{x})$, we cannot use $p_{opt}(\vec{x})$ instead of the info. max. design in actual experiments because to compute $p_{opt}(\vec{x})$ we

need to know $\vec{\theta}$.

The limit theorem for information maximizing designs, Eqns. 74 & 75, only holds if the order of the trials does not matter as $t \to \infty$ [114]. Consequently, we can only apply Eqn. 75 to situations where r_t depends only on the current stimulus, i.e. $\vec{s}_t = \vec{x}_t$. 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 [57]. 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 semi-analytical 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 2.7.3, we present results showing that our information-maximizing 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 information maximizing design compared to an i.i.d. design. For an i.i.d. design, Eqns. 74 & 75 still hold, under appropriate conditions, provided we take the expectation in Eqn. 75 with respect to the distribution, $p_{iid}(\vec{x})$, from which stimuli are selected on each trial [157]. As a result we can use Eqn. 75 to compute and compare the asymptotic performance of our information maximizing design and $p_{iid}(\vec{x})$. In this section the stimulus distribution $p_{iid}(\vec{x})$ will refer to a uniform distribution on the

sphere $||\vec{x}||_2 = m$.

2.7.1 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 which 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

$$F(p(\vec{x})) = \log |E_{\vec{x}} J_{exp}(\vec{\theta}, \vec{x})| \tag{77}$$

$$= \log \left| E_{x_1} \left(E_{r|x_1} D(r, x_1\theta_1) E_{x_2 \ldots x_{\dim(\vec{\theta})}|x_1} \left(\vec{x}\vec{x}^T \right) \right) \right| \tag{78}$$

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 $p(x_2, \ldots, x_{\dim(\vec{\theta})}|x_1)$ which maximizes F for some $p(x_1)$ (the proof is in Appendix 2.10.4).

If we consider only spherically symmetric $p(x_2, \ldots, x_{\dim(\vec{\theta})}|x_1)$, we can easily evaluate the inner integral in Eqn. 78:

$$E_{\vec{x}_2, \ldots, x_{\dim(\vec{\theta})}|x_1} \vec{x}\vec{x}^T = \begin{bmatrix} x_1^2 & \\ & \frac{1}{\dim(\vec{\theta})-1} (E_{||\vec{x}||_2 \big|_{x_1}} ||\vec{x}||_2^2 - x_1^2) I_{\dim(\vec{x})-1} \end{bmatrix}, \tag{79}$$

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,

$$\log |E_{\vec{x}} J_{exp}(\vec{\theta}, \vec{x})| = \log E_{x_1} E_{r|x_1} D(r, x_1\theta_1) x_1^2 + (\dim(\vec{\theta}) - 1) \log E_{x_1} E_{r|x_1} D(r, x_1\theta_1) \frac{E_{||\vec{x}||_2^2 \big|_{x_1}} ||\vec{x}||_2^2 - x_1^2}{\dim(\vec{\theta}) - 1}. \tag{80}$$

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 $p_{opt}(x_2, \ldots, x_{\dim(\vec{\theta})} | x_1)$ is spherically symmetric, $p_{opt}(x_2, \ldots, x_{\dim(\vec{\theta})} | x_1)$ is just a uniform distribution on the $\dim(\vec{\theta}) - 1$-dimensional sphere of radius $\sqrt{m^2 - x_1^2}$. To find the optimal distribution on x_1 we solve

$$p_{opt}(x_1) = \arg\max_{p(x_1)} \left[\log \phi + (\dim(\vec{\theta}) - 1) \log(\frac{m^2\beta - \phi}{\dim(\vec{\theta}) - 1}) \right] \tag{81}$$

$$\phi = E_{x_1}(E_{r|x_1} D(r, x_1\theta_1) {x_1}^2) \tag{82}$$

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

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 $p_{opt}(x_1)$ which is supported on just two values of ${x_1}$[4]. Thus we have reduced our objective function to

$$\begin{aligned}
&\log \phi + (\dim(\vec{\theta}) - 1) \log \left(\frac{m^2\beta - \phi}{\dim(\vec{\theta}) - 1} \right) \\
&= \log \left(w E_{r|y_1} D(r, y_1\theta_1) y_1^2 + (1 - w) E_{r|y_2} D(r, y_2\theta_1) y_2^2 \right) \\
&+ (\dim(\vec{\theta}) - 1) \log \left(w E_{r|y_1} D(r, y_1\theta_1)(m^2 - y_1^2) + (1 - w) E_{r|y_2} D(r, y_2\theta_1)(m^2 - y_2^2) \right) \\
&+ const.,
\end{aligned} \tag{83}$$

which has just three unknown parameters: the two support points (y_1, y_2) of $p(x_1)$, where $-m \leq y_1 \leq y_2 \leq m$ 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 Eqn. 83 with respect to w to zero. As a result, solving for the best values of (y_1, y_2, w) requires a simple 2-dimensional

[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.

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 $\dim(\vec{\theta})$.

To prove that $p_{opt}(x_1)$ converges to a distribution with support on a single point as $\dim(\vec{\theta}) \to \infty$, 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 Eqn. 83 with respect to w to 0,

$$w = \frac{b \dim(\vec{\theta}) y_2^2 (am^2 - bm^2 - ay_1^2 + by_2^2) - abm^2 y_1^2 + abm^2 y_2^2}{\dim(\vec{\theta}) \left(by_2^2 - ay_1^2\right) \left(am^2 - bm^2 - ay_1^2 + by_2^2\right)} \tag{84}$$

$$a = E_{r|y_1} D(r, y_1\theta_1) \tag{85}$$

$$b = E_{r|y_2} D(r, y_2\theta_2). \tag{86}$$

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 Eqn. 83.

We can easily evaluate the limit of w as $\dim(\vec{\theta}) \to \infty$,

$$\lim_{\dim(\vec{\theta}) \to \infty} w = \frac{by_2^2}{by_2^2 - ay_1^2}. \tag{87}$$

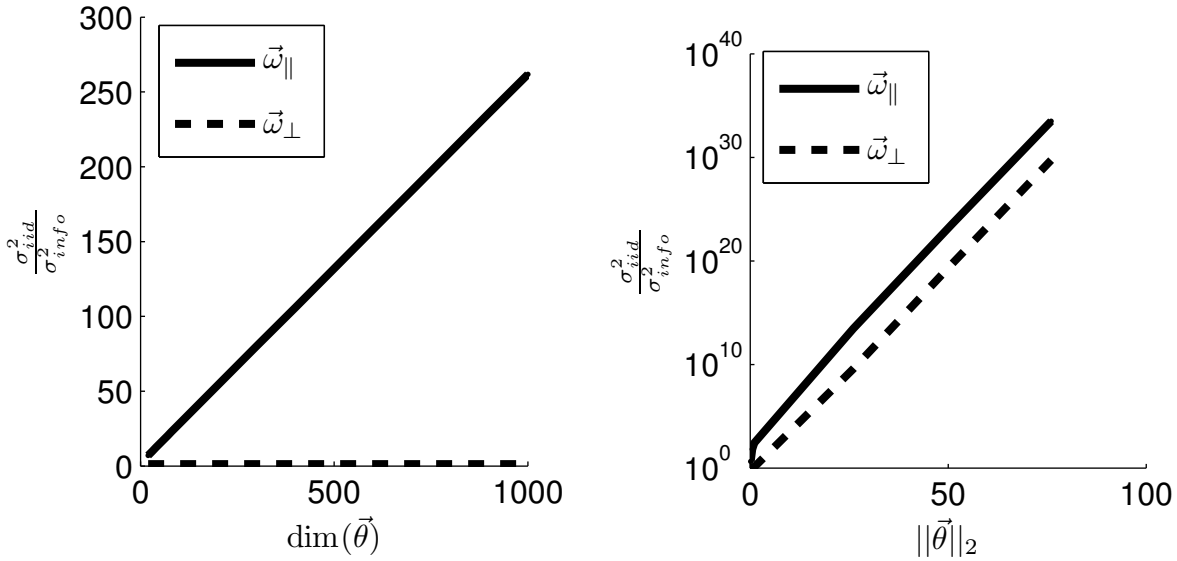
b and a are positive because the Fisher information is always positive . Furthermore, $y_2 > y_1$ by assumption. These facts ensure that

$$\lim_{\dim(\vec{\theta}) \to \infty} w \leq 0 \qquad \text{or} \qquad \lim_{\dim(\vec{\theta}) \to \infty} w \geq 1 \tag{88}$$

In either case, the optimal weight ends up being $w = 1$ or $w = 0$ so the optimal distribution only has support on a single point as $\dim(\vec{\theta}) \to \infty$.

2.7.2 Relative efficiency of the info. max. design

We can quantify the relative efficiency of the information maximizing design to the i.i.d. design by computing the ratio of the asymptotic variances; i.e. the ratio of the



(a) (b)

Figure 14: We measure the relative efficiency of the info. max. design to the i.i.d. as the ratio of the variances, Eqn. 89, 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 info. max. design decreases the variance in the direction of $\vec{\theta}$ faster than the i.i.d. design by a factor which increases linearly with $\dim(\vec{\theta})$. $\frac{\sigma^2_{iid}(\vec{\omega}_{\perp})}{\sigma^2_{info}(\vec{\omega}_{\perp})}$ has a value greater than one and is relatively flat with respect to $\dim(\vec{\theta})$. Consequently, as $\dim(\vec{\theta})$ increases the info. max. 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 info. max. 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.

dotted grey lines to the dotted black lines in Figures 15 & 16. The ratio,

$$\frac{\sigma^2_{iid}(\vec{\omega})}{\sigma^2_{info}(\vec{\omega})} \triangleq \frac{\vec{\omega}^T \boldsymbol{C}_{iid} \vec{\omega}}{\vec{\omega}^T \boldsymbol{C}_{info} \vec{\omega}}, \tag{89}$$

measures how much faster the info. max. 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 which come from Eqn. 75,

$$C_{info} = \left(E_{p_{opt}(\vec{x})} \left(J_{exp}(\vec{\theta}, \vec{x}) \right) \right)^{-1} \qquad C_{iid} = \left(E_{p_{iid}(\vec{x})} \left(J_{exp}(\vec{\theta}, \vec{x}) \right) \right)^{-1}. \tag{90}$$

We know from Section 2.7.1 that for both designs one eigenvector of $E_{\vec{x}} \left(J_{exp}(\vec{\theta}, \vec{x}) \right)$ 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 $E_{\vec{x}} \left(J_{exp}(\vec{\theta}, \vec{x}) \right)$ 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,

$$\sigma^2(\vec{\omega}_{\|}) = \left(E_{\vec{x}} E_{r|\vec{x}} D(r, \vec{x}^T \vec{\theta}) \frac{(\vec{x}^T \vec{\theta})^2}{||\vec{\theta}||_2^2} \right)^{-1} \tag{91}$$

$$\sigma^2(\vec{\omega}_{\perp}) = \left(E_{\vec{x}} E_{r|\vec{x}} D(r, \vec{x}^T \vec{\theta}) \frac{m^2 - \frac{(\vec{x}^T \vec{\theta})^2}{||\vec{\theta}||_2^2}}{\dim(\vec{\theta}) - 1} \right)^{-1}, \tag{92}$$

where $p(\vec{x})$ depends on whether we are computing $\sigma^2_{info}(\vec{\omega})$ or $\sigma^2_{iid}(\vec{\omega})$. $\vec{\omega_{\|}}$ is a unit vector parallel to $\vec{\theta}$ and $\vec{\omega_{\perp}}$ is a unit vector orthogonal to $\vec{\theta}$. Using these 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 $\dim(\vec{\theta})$ and $||\vec{\theta}||_2$.

For the exponential nonlinearity,

$$\frac{\sigma^2_{iid}(\vec{\omega}_{\|})}{\sigma^2_{info}(\vec{\omega}_{\|})} = \frac{E_{p_{opt}(\vec{x})} \exp(\vec{x}^T \vec{\theta}) \frac{(\vec{x}^T \vec{\theta})^2}{||\vec{\theta}||_2^2}}{E_{p_{iid}(\vec{x})} \exp(\vec{x}^T \vec{\theta}) \frac{(\vec{x}^T \vec{\theta})^2}{||\vec{\theta}||_2^2}} \tag{93}$$

$$\frac{\sigma^2_{iid}(\vec{\omega}_{\perp})}{\sigma^2_{info}(\vec{\omega}_{\perp})} = \frac{E_{p_{opt}(\vec{x})} \exp(\vec{x}^T \vec{\theta})(m^2 - \frac{(\vec{x}^T \vec{\theta})^2}{||\vec{\theta}||_2^2})}{E_{p_{iid}(\vec{x})} \exp(\vec{x}^T \vec{\theta})(m^2 - \frac{(\vec{x}^T \vec{\theta})^2}{||\vec{\theta}||_2^2})} \tag{94}$$

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

Since the information of any stimulus depends on ρ_t, we would expect that the info. max. design. would become more efficient as $\dim(\vec{\theta})$ increases. Intuitively, as $\dim(\vec{\theta})$ increases, the probability of an i.i.d. design picking a direction which is highly correlated with $\vec{\theta}$ decreases because the variance of $(\vec{x}^T\vec{\theta})$ decreases linearly with $\dim(\vec{\theta})$(see Section 2.5.3). In contrast, the info. max. 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 $\frac{\sigma^2_{iid}(\vec{\omega}_\parallel)}{\sigma^2_{info}(\vec{\omega}_\parallel)}$ is asymptotically linear in $\dim(\vec{\theta})$. The d^{-1} scaling of the variance of $(\vec{x}^T\vec{\theta})$ for the i.i.d. design means that $\sigma^2_{iid}(\vec{\omega}_\parallel)$ 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 $\dim(\vec{\theta})$. As a result, the variance in any direction increases linearly with $\dim(\vec{\theta})$.

In contrast, the info. max. 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 info. max. design we use the fact that as $\dim(\vec{\theta}) \to \infty$, $p_{opt}(x_1)$ converges to a distribution which has support on a single point, x_1. Furthermore, we can easily show, see Appendix 2.10.5, that as $\dim(\vec{\theta}) \to \infty$, 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 $\dim(\vec{\theta})$ while $\sigma^2_{info}(\vec{\omega_\perp})$ increases linearly with $\dim(\vec{\theta})$. Since $\sigma^2_{iid}(\vec{\omega}_\parallel)$ scales linearly with $\dim(\vec{\theta})$ and $\sigma^2_{info}(\vec{\omega}_\parallel)$ is asymptotically constant with respect to $\dim(\vec{\theta})$, the relative efficiency

[5]For the i.i.d. design, $p(\vec{x}^T\frac{\vec{\theta}}{||\vec{\theta}||_2})$ has mean zero and variance $m^2/\dim(\vec{\theta})$ (see Section 2.5.3 and [114];note that [114] mistakenly had a scaling of $\dim(\vec{\theta})^{-2}$ here, instead of the correct rate of $\dim(\vec{\theta})^{-1}$). This result ensures that $\vec{x}^T\frac{\vec{\theta}}{||\vec{\theta}||_2}$ converges to zero at the rate $\dim(\vec{\theta})^{-1/2}$. Since the power of $\vec{x}$ is constrained and the variance of $\vec{x}^T\frac{\vec{\theta}}{||\vec{\theta}||_2}$ decreases as $1/\dim(\vec{\theta})$, it follows that both $\sigma^2_{iid}(\vec{\omega}_\parallel)$ and $\sigma^2_{iid}(\vec{\omega}_\perp)$ increase linearly with $\dim(\vec{\theta})$.

of the info. max. design in direction $\vec{\omega}_{\|}$ increases linearly with $\dim(\vec{\theta})$:

$$\frac{\sigma^2_{iid}(\vec{\omega}_{\|})}{\sigma^2_{info}(\vec{\omega}_{\|})} = O(\dim(\vec{\theta})). \tag{95}$$

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

$$\frac{\sigma^2_{iid}(\vec{\omega}_{\perp})}{\sigma^2_{info}(\vec{\omega}_{\perp})} = O(1). \tag{96}$$

These results are also plotted in Figure 14(a). The important conclusion is that as $\dim(\vec{\theta})$ increases we can reduce our uncertainty about $\vec{\theta}$ by a factor of $\dim(\vec{\theta})$ by using an info. max. 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 info. max. 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 info. max. 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 $\dim(\vec{\theta}) \to \infty$. By taking the limit of x_1 as $||\vec{\theta}||_2 \to \infty$ (see Appendix 2.10.5), 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 info. max. design imply that the ratios in Eqns. 93 & 94 grow exponentially with $||\vec{\theta}||_2$.

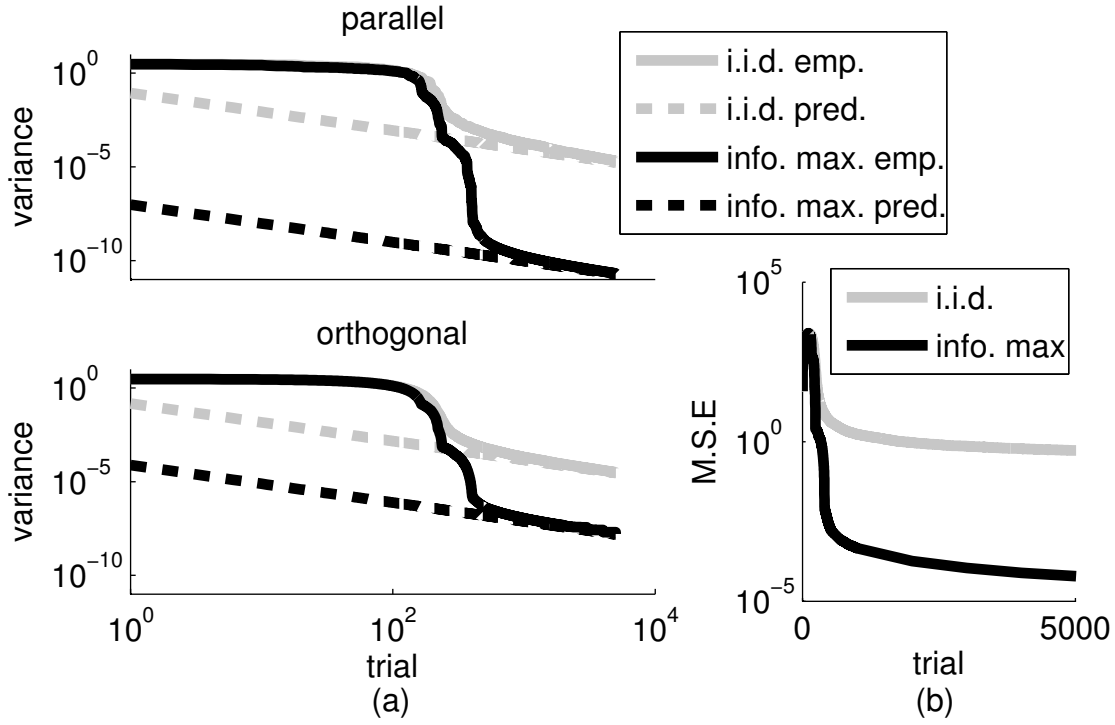


Figure 15: Comparison of the empirical posterior covariance matrix to the asymptotic variance predicted by Eqn. 75. Despite our approximations, the empirical covariance matrix under an info. max. design converged to the predicted value. a) The top axis shows the variance in the direction of the posterior 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 11x15 Gabor patch. Stimuli were selected under the power constraint using an i.i.d. or info. max. design. b) The mean squared error between the empirical variance and the asymptotic variance.

2.7.3 Convergence to the asymptotically optimal covariance matrix

We can verify whether our design converges to the asymptotic design by testing whether the covariance matrix of the posterior converges to the value predicted by Eqn. 75, Figures 15 & 16. 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

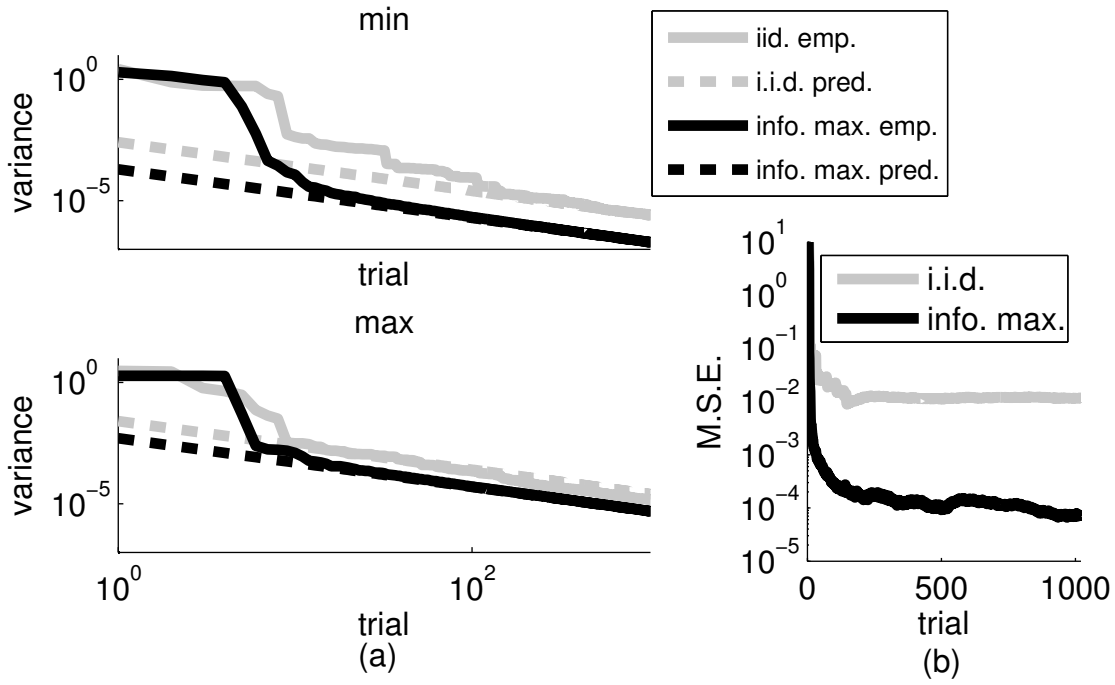


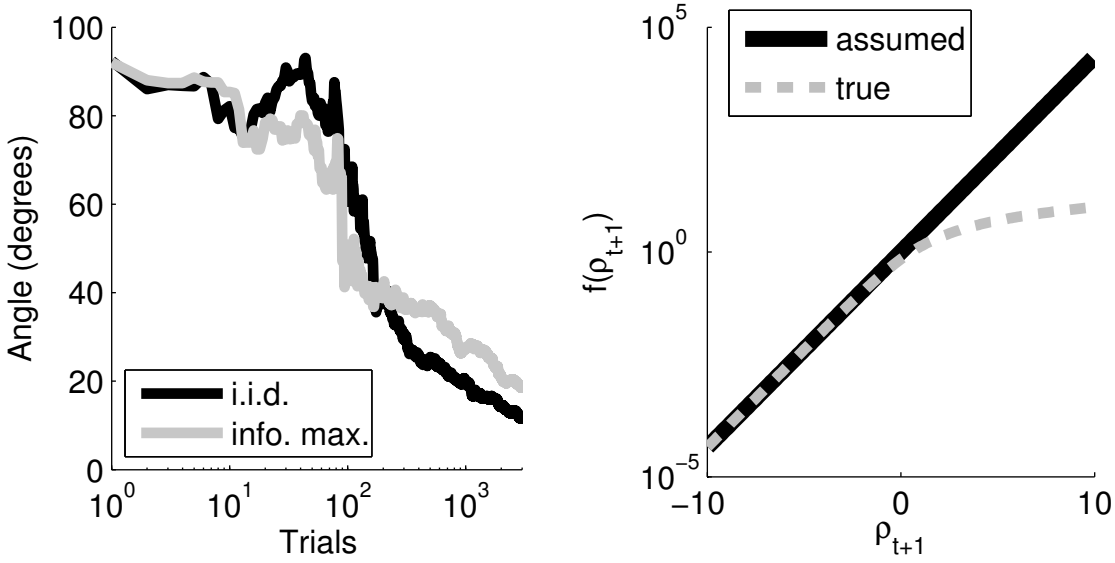
Figure 16: Comparison of the empirical variance of the posterior in our simulations to the asymptotic variance predicted based on the central limit theorem. The info. max. design picked the optimal stimulus from a small number of stimuli (see text for details). a) The axes compare the minimum eigenvalue and maximum eigenvalue of the asymptotic covariance matrix to the empirical variance in the direction of the corresponding eigenvalue. b) A plot of the mean squared error between the empirical variance and the asymptotic variance.

Gabor results presented earlier. The symmetry of $p(x_2 \dots x_{\dim(\vec{\theta})}|x_1)$ 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 info. max. design when we pick the stimulus from a finite set. We chose a low 5-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 16, we compare the maximum and minimum eigenvalues of the asymptotic covariance matrix to the empirical variance in these directions. We also plot the mean squared error 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 1-dimensionality of our GLM and the flatness of our prior. Since the 1-dimensional GLM only collects information in one direction, we need to make $\dim(\vec{\theta})$ observations in order to decrease our initial uncertainty in all directions. Until we make $\dim(\vec{\theta})$ 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 [114] that ensures the posterior entropy of an info. max. design will in general be asymptotically no greater than that of an i.i.d. design. Info. max does better whenever the limiting design $p_{opt}(\vec{x})$ depends on $\vec{\theta}$, as this ensures there is not



(a) (b)

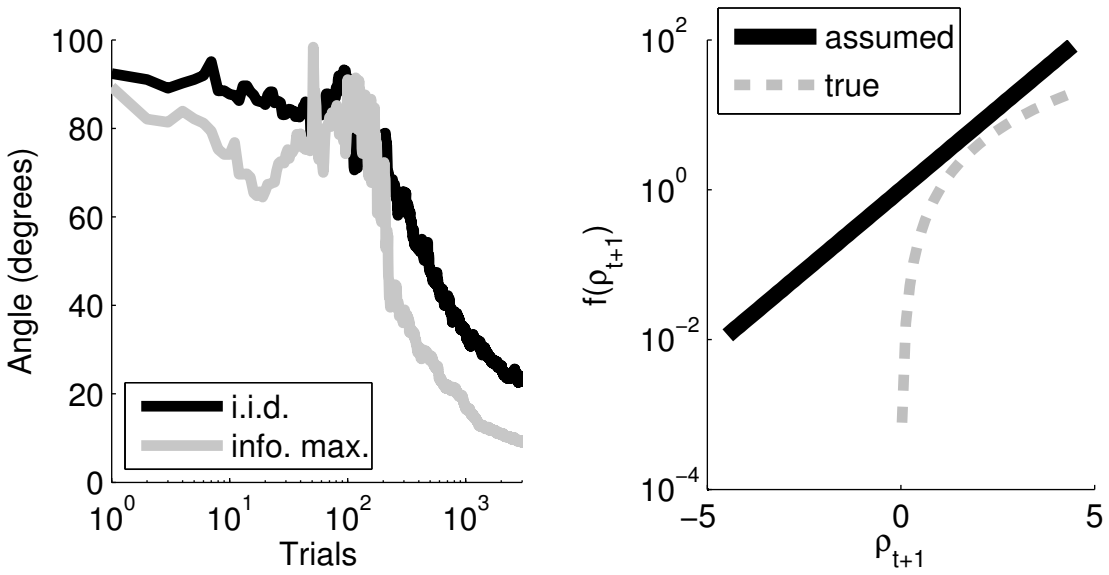
Figure 17: Effect of model misspecification. Info. max. stimuli were selected using the wrong nonlinearity. The results compare the accuracy of the estimated $\vec{\theta}$ using i.i.d. stimuli versus info. max. stimuli. Since the parameters can at best be estimated up to a scaling factor, a) shows the angle between the estimated parameters and their true value. b) A 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 $f(\rho_{t+1}) = \exp(\vec{\theta}^T\vec{s}_{t+1})$.

a single distribution which simultaneously maximizes the efficiency for all (a priori unknown) values of $\vec{\theta}$. For our GLM with a conditional Poisson, Figure 2.2, the Fisher information depends on the stimulus and $\vec{\theta}$. Therefore, the optimal design cannot be determined a-priori.

2.8 Misspecified Models

We used simulations to investigate the performance of the info. max. algorithm when the link function, $f()$, is incorrect. The two primary questions we are interested in are 1) whether the estimated $\vec{\theta}$ converges to the true value, and 2) 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



(a) (b)

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

scaling factor even if the link function is misspecified, provided the input distribution, $p(\vec{s}_{t+1})$, is elliptically symmetric [92, 113]. 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 info. max. design does not in general produce elliptically symmetric stimulus distributions because the 1-d 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 $p(\rho_{t+1} = \vec{\mu}_t^T \vec{s}_{t+1})$ which is not symmetric about zero. We would therefore expect the info. max. 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 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, e.g. by using an elliptically symmetric design, but at the expense of being less efficient than the info. max. design.

Nonetheless, our simulations showed that the estimates produced by the info. max. 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 was generated using the nonlinearity $f(\rho_{t+1}) = \log(1+\exp(\vec{\theta}^T\vec{s}_{t+1}))$. The info. max. 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 info. max. design does a poor job of picking optimal stimuli. Nonetheless using an info. max. design leads to estimates which 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 info. max. 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 info. max. design miscalculates the Fisher information, it correctly predicts that the Fisher information is increasing with ρ_{t+1}. Consequently, the info. max. 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.

2.9 Discussion

Previous work [99, 24, 114] 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 which are often possessed by models of neural systems.

Our algorithm uses several ideas which are frequently employed in experimental design. The mutual information as a design criterion has been proposed by many authors [95, 13, 57, 99, 114]. To evaluate the mutual information, we use a normal approximation of the posterior. While we rely on a theorem due to [114] which proves asymptotic normality for the mutual information criterion, similar results concerning the asymptotics of sequential designs exist in the statistics literature [169, 25, 129]. 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 [114, 130]. The logistic model in particular has received a great deal of attention because the logistic model is frequently used for classification [84, 63, 132, 130]. Compared to our algorithm, previous algorithms for sequential design with GLMs do not scale nearly as well in high-dimensions [25, 105].

Optimal experimental design is also closely related to problems in optimal control [109, 153] and reinforcement learning [79, 16]. In reinforcement learning the goal is to find the set of actions which 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.

2.9.1 Optimal design in neurophysiology

The application of sequential design to neurophysiology is not new [10]. 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 [59, 65, 48, 98, 111]. 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 which maximizes the firing rate of the neuron [110, 39, 59, 173, 111]. 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 studying encoding in sensory systems, natural objective functions are the mutual information between the stimulus and response [98] and the stimulus reconstruction error [48]. These metrics are used to find stimuli which 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 [59, 65, 111]. 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 non-stationarities, we automatically take their impact on the informativeness of different designs into account when optimizing our design.

2.9.2 Future work

In most experiments, neurophysiologists are interested in how well we can model the neuron after all the data has been collected. We can measure the utility of the dataset as the mutual information between all observations and $\vec{\theta}$, $I(\{\boldsymbol{r_{1:t}}, \vec{\theta}\}|\boldsymbol{x_{1:t}})$ where t represents the total number of trials. Unfortunately, there is no guarantee that a design based on maximizing $I(r_{t+1}|\vec{\theta}, \vec{x}_{t+1}, \boldsymbol{s_{1:t}}, \boldsymbol{r_{1:t}})$, will also maximize $I(\{\boldsymbol{r_{1:t}}, \vec{\theta}\}|\boldsymbol{x_{1:t}})$. When we pick stimuli by maximizing $I(r_{t+1}|\vec{\theta}, \vec{x}_{t+1}, \boldsymbol{s_{1:t}}, \boldsymbol{r_{1:t}})$ 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 on future trials [33]. 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 only increase the information obtained about $\vec{\theta}_f$ 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 which 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.

Non-greedy optimization is more challenging than maximizing $I(r_{t+1}|\vec{\theta}, \vec{x}_{t+1}, \boldsymbol{s_{1:t}}, \boldsymbol{r_{1:t}})$.

One of the primary challenges of non greedy 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, non-greedy 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 non-greedy 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 [133], low-rank [42, 94], 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, non-greedy optimization and more refined priors, is to permit experiments which 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 chapter. We plan to apply this methodology

to real experimental data in the near future.

2.10 Appendix

2.10.1 Computing $\mathcal{R}_{t+1}$ under the power constraint

In Section 2.5.2 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, Eqns. 55 & 56, as a function of μ_ρ. To solve these optimization problems, we use the Karush-Kuhn-Tucker(K.K.T.) conditions.

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

$$\sigma_\rho^2 = \vec{s}_{t+1}^T \boldsymbol{C}_t \vec{s}_{t+1} \tag{97}$$

$$= \vec{s}_{x,t+1}^T \boldsymbol{C}_x \vec{s}_{x,t+1} + 2\vec{s}_{f,t+1}^T \boldsymbol{C}_{fx} \vec{s}_{x,t+1} + \vec{s}_{f,t+1}^T \boldsymbol{C}_f \vec{s}_{f,t+1}, \tag{98}$$

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

$$\boldsymbol{C}_t = \left[\begin{array}{c|c} \boldsymbol{C}_x & \boldsymbol{C}_{xf} \\ \hline \boldsymbol{C}_{fx} & \boldsymbol{C}_f \end{array}\right]. \tag{99}$$

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

$$\max \sigma_\rho^2 = \max_{\vec{s}_{x,t+1}} \vec{s}_{x,t+1}^T \boldsymbol{C}_x \vec{s}_{x,t+1} + 2\vec{s}_{f,t+1}^T \boldsymbol{C}_{fx} \vec{s}_{x,t+1} + \vec{s}_{f,t+1}^T \boldsymbol{C}_f \vec{s}_{f,t+1} \tag{100}$$

$$\min \sigma_\rho^2 = \min_{\vec{s}_{x,t+1}} \vec{s}_{x,t+1}^T \boldsymbol{C}_x \vec{s}_{x,t+1} + 2\vec{s}_{f,t+1}^T \boldsymbol{C}_{fx} \vec{s}_{x,t+1} + \vec{s}_{f,t+1}^T \boldsymbol{C}_f \vec{s}_{f,t+1} \tag{101}$$

$$\text{s.t} \quad \mu_\rho = \vec{s}_{t+1}^T \vec{\mu}_t \quad ||\vec{s}_{x,t+1}||_2 \leq m. \tag{102}$$

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 1-d function of, α, the projection of $\vec{s}_{x,t+1}$ along the

mean,

$$\mu_\rho = \alpha||\vec{\mu}_{x,t}||_2 + \vec{\mu}_{f,t}^T\vec{s}_{f,t+1}. \tag{103}$$

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 $\vec{\mu}_{x,t}$,

$$\vec{s}'_{x,t+1} \triangleq \vec{s}_{x,t+1} - \frac{\vec{\mu}_{x,t}^T\vec{s}_{x,t+1}}{||\vec{\mu}_{x,t}||_2^2}\vec{\mu}_{x,t} + \frac{\alpha}{||\vec{\mu}_{x,t}||_2}\vec{\mu}_{x,t}. \tag{104}$$

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 Eqn. 104,

$$\sigma_\rho^2 = \vec{s}'^{\,T}_{x,t+1}\boldsymbol{C}_t\vec{s}'_{x,t+1} \tag{105}$$

$$= \vec{s}_{x,t+1}^T A\vec{s}_{x,t+1} + \vec{b}(\alpha)^T\vec{s}_{x,t+1} + d(\alpha) \tag{106}$$

$$A = \boldsymbol{C}_x - \frac{1}{2}\vec{v}\vec{v}^T + \frac{1}{2}\vec{u}\vec{u} \tag{107}$$

$$\vec{v} = \frac{-\vec{\mu}_{x,t}^T\boldsymbol{C}_x\vec{\mu}_{x,t} + 2||\vec{\mu}_x||_2^2}{2||\vec{\mu}_{x,t}||_2^3}\vec{\mu}_{x,t} + \boldsymbol{C}_x\frac{\vec{\mu}_{x,t}}{||\vec{\mu}_{x,t}||_2} \tag{108}$$

$$\vec{u} = \frac{-\vec{\mu}_{x,t}^T\boldsymbol{C}_x\vec{\mu}_{x,t} - 2||\vec{\mu}_x||_2^2}{2||\vec{\mu}_{x,t}||_2^3}\vec{\mu}_{x,t} + \boldsymbol{C}_x\frac{\vec{\mu}_{x,t}}{||\vec{\mu}_{x,t}||_2} \tag{109}$$

$$\vec{b}(\alpha) = 2\alpha\boldsymbol{C}_x\frac{\vec{\mu}_{x,t}}{||\vec{\mu}_{x,t}||_2} - 2\alpha(\vec{\mu}_{x,t}^T\boldsymbol{C}_x\vec{\mu}_{x,t})\frac{\vec{\mu}_{x,t}}{||\vec{\mu}_{x,t}||_2^3} + 2(\boldsymbol{C}_{xf}\vec{s}_{f,t+1}) - 2(\vec{\mu}_{x,t}^T\boldsymbol{C}_{xf}\vec{s}_{f,t+1})\frac{\vec{\mu}_{x,t}}{||\vec{\mu}_{x,t}||_2^2} \tag{110}$$

$$d(\alpha) = \alpha^2\frac{(\vec{\mu}_{x,t}^T\boldsymbol{C}_x\vec{\mu}_{x,t})}{||\vec{\mu}_{x,t}||_2^2} + 2\alpha\frac{\vec{\mu}_{x,t}^T}{||\vec{\mu}_{x,t}||_2}\boldsymbol{C}_{xf}\vec{s}_{f,t+1} + \vec{s}_{f,t+1}^T\boldsymbol{C}_f\vec{s}_{f,t+1}. \tag{111}$$

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, Eqn. 106 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 Eqn. 106 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 Eqn. 106 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) [61, 12]. For the TRS the K.K.T. conditions are both necessary and sufficient [61]. Therefore, we can find all local minima and maxima by solving the K.K.T. conditions.

Before we compute the K.K.T. conditions, we transform our coordinates using the eigenbasis of A,

$$A = \boldsymbol{G}_t \Lambda_t \boldsymbol{G}_t^T \qquad \vec{y}_{t+1} = \boldsymbol{G}_t^T \vec{s}_{x,t+1} \qquad \vec{w}_t(\alpha) = \boldsymbol{G}_t^T \vec{b}(\alpha). \tag{112}$$

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 $\vec{y}_{t+1}$,

$$\max_{\vec{y}_{t+1}} \sigma_\rho^2 = \max_{\vec{y}_{t+1}} \sum_i c_{i,t} \left(y_{i,t+1} + \frac{w_{i,t}(\alpha)}{2c_{i,t}} \right)^2 - \frac{w_{i,t}^2(\alpha)}{4c_{i,t}} \tag{113}$$

$$\min_{\vec{y}_{t+1}} \sigma_\rho^2 = \min_{\vec{y}_{t+1}} \sum_i c_{i,t} \left(y_{i,t+1} + \frac{w_{i,t}(\alpha)}{2c_{i,t}} \right)^2 - \frac{w_{i,t}^2(\alpha)}{4c_{i,t}} \tag{114}$$

$$\text{s.t } ||\vec{y}_{t+1}||_2^2 \leq m^2 - \alpha^2, \tag{115}$$

where c_i denotes the i^{th} eigenvalue of A. To enforce the power constraint we introduce a Lagrange multiplier,

$$\sum_i c_{i,t} \left(y_{i,t+1} + \frac{w_{i,t}(\alpha)}{2c_{i,t}} \right)^2 - \frac{w_{i,t}^2(\alpha)}{4c_{i,t}} - \lambda y_{i,t+1}^2. \tag{116}$$

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 K.K.T. conditions, result in a system of $\dim(\vec{\theta})$ equations for the gradient of σ_ρ^2 with respect to $\vec{y}_{t+1}$:

$$2y_{i,t+1}(c_{i,t} - \lambda) = -w_i(\alpha) \quad \forall i. \tag{117}$$

When $\lambda \neq c_{i,t}$ we can solve the first order K.K.T. for $y_{i,t+1}$,

$$y_{i,t+1} = \frac{-w_{i,t}(\alpha)}{2(c_{i,t} - \lambda)} \tag{118}$$

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 K.K.T. 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

$$c_{i,t} - \lambda \leq 0 \quad \forall i. \tag{119}$$

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

$$c_{i,t} - \lambda \geq 0 \quad \forall i, \tag{120}$$

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

By solving the K.K.T. 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 will assume the eigenvalues $c_{i,t}$ of $\boldsymbol{G}_t$, Eqn. 112, 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 will also use $c_{\max,t}$ to denote the maximum eigenvalue. We will refer to the set of $(\mu_\rho, \sigma_\rho^2)$ which solve the K.K.T. conditions as $\mathcal{B}$. We will 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 K.K.T. conditions for $\sigma^2_{\rho,\max}$ are only satisfied 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 Eqn. 118.
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 [61].

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 non-empty only if $w_{\dim(\vec{\theta}),t}(\alpha) = 0$. 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, Eqn. 118. 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: i) there are a finite number of α such that $w_{\dim(\vec{\theta}),t}(\alpha) = 0$, and ii) $w_{\dim(\vec{\theta}),t}(\alpha) = 0 \quad \forall\, \alpha$.

The first case is easy. Since we set $\lambda = c_{\max,t}$, we can find α by solving $w_{\dim(\vec{\theta}),t}(\alpha) = 0$. We can then compute all components of the stimulus except $y_{\dim(\vec{\theta}),t+1}$ by plugging α and λ into Eqn. 118. Since $\sigma_{\rho,max}^2$ is increasing with the stimulus power, we set $y_{\dim(\vec{\theta}),t+1}$ so that the power constraint is satisfied with equality,

$$y_{\dim(\vec{\theta}),t+1}^2 = m^2 - \alpha^2 - \sum_{i=1}^{d-1} y_{i,t+1}(c_{max}, \alpha)^2. \tag{121}$$

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

The second case, $w_{\dim(\vec{\theta}),t}(\alpha) = 0 \quad \forall\, \alpha$ 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 $y_{\dim(\vec{\theta}),t+1}^2$ on the interval $[0, m^2]$ and for each value evaluate steps 2-4.

2. Use $\lambda = c_{\max,t}$ and Eqn. 118 to compute $y_{i,t+1}$ for $1 \leq i < d$ in terms of α.

3. Compute α by solving Eqn. 121 using the results from steps 1 & 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 in the eigenspace of the maximum eigenvector 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 K.K.T. 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 K.K.T. conditions for $\sigma^2_{\rho,\min}$ are only satisfied for $\lambda \leq 0$, 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,

$$\Phi = \{\alpha : \sum_i y_{i,t+1}(\alpha)^2 = \sum_i \frac{w_{i,t}^2}{4c_{i,t}^2} \leq m^2 - \alpha^2 \ \& \ \alpha \in [-m, m]\} \tag{122}$$

2. For each $\alpha \in \Phi$ compute $\vec{y}_{t+1}(\alpha)$ by plugging $\lambda = 0$ and α into Eqn. 118.

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

Clearly, Eqn. 114 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 Eqn. 118.

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 K.K.T. 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 2.5.4.1 the average running time of computing the eigendecomposition of A scales as $O(\dim(\vec{\theta})^2)$ in practice.

2.10.2 Proof of convexity condition

We now prove the lemma used in Section 2.5.2 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,

$$y = \frac{x - \mu}{\sigma}, \tag{123}$$

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

$$E_x g(x,\sigma^2) = \int_{-\infty}^{\infty} g(x,\sigma^2)\frac{1}{\sigma\sqrt{2\pi}}\exp(-\frac{(x-\mu)^2}{2\sigma^2})dx \tag{124}$$

$$= \int_{-\infty}^{\infty} g(y\sigma+\mu,\sigma^2)\frac{1}{\sqrt{2\pi}}\exp(-\frac{y^2}{2})dy \tag{125}$$

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

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

$$+ \frac{y}{2\sigma}\big(g_x(y\sigma+\mu,\sigma^2) - g_x(-y\sigma+\mu,\sigma^2)\big)] > 0 \tag{126}$$

$$g_x(x,\sigma^2) = \frac{\partial g(x,\sigma^2)}{\partial x} \quad g_{\sigma^2}(x,\sigma^2) = \frac{\partial g(x,\sigma^2)}{\partial \sigma^2} \tag{127}$$

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, $\frac{dE_x g(x,\sigma^2)}{d\sigma^2}$ is positive which guarantees $E_x g(x,\sigma^2)$ is monotonically increasing in σ^2.

We can easily modify our solution for optimizing the stimulus under the power constraint, Section 2.5.2, so that we can choose the stimulus from an ellipsoid with arbitrary center and radii. In this case the stimulus domain is defined as,

$$\vec{s}_{t+1} = \vec{s}_{c,t+1} + \vec{s}_{r,t+1} \quad \vec{s}_{r,t+1}^T M \vec{s}_{r,t+1} \leq m^2, \tag{128}$$

where M is a symmetric, positive semi-definite matrix which 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

$$\sigma_\rho^2 = \vec{s}_{r,t+1}^T A \vec{s}_{r,t+1} + (2\vec{s}_{c,t+1}^T A + \vec{b})^T \vec{s}_{r,t+1} + \vec{s}_{c,t+1}^T A \vec{s}_{c,t+1} + \vec{b}^T \vec{s}_{c,t+1} + d. \tag{129}$$

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,

$$M = \boldsymbol{G}_M \Lambda_M \boldsymbol{G}_M^T \quad \vec{y}_{r,t+1} = \Lambda^{1/2} \boldsymbol{G}_M^T \vec{s}_{r,t+1}. \tag{130}$$

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.

2.10.3 Minimizing the M.S.E.of $\vec{\theta}$

The mean squared error (M.S.E.) of the parameters provides an alternative metric for our uncertainty about $\vec{\theta}$. The M.S.E. is advantageous if we care about some components of $\vec{\theta}$ more than others. In this case we can use the weighted M.S.E. 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 mean squared error is,

$$E_{\vec{\theta}|\{\boldsymbol{x_{1:t}}, \boldsymbol{r_{1:t}}\}}(||\vec{\theta} - \vec{\theta}_o||_2^2) = E_{\vec{\theta}|\{\boldsymbol{x_{1:t}}, \boldsymbol{r_{1:t}}\}}(\vec{\theta}^T\vec{\theta}) - 2\vec{\theta}_o^T E_{\vec{\theta}|\{\boldsymbol{x_{1:t}}, \boldsymbol{r_{1:t}}\}}(\vec{\theta}) + \vec{\theta}_o^T\vec{\theta}_o \tag{131}$$

$$= E_{\vec{\theta}|\{\boldsymbol{x_{1:t}}, \boldsymbol{r_{1:t}}\}}(\vec{\theta}^T\vec{\theta}) - 2\vec{\theta}_o^T\vec{\mu}_t + const, \tag{132}$$

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 M.S.E. by taking the expectation with respect to our current posterior.

$$E_{\vec{\theta}|\{\boldsymbol{x_{1:t}}, \boldsymbol{r_{1:t}}\}}(||\vec{\theta} - \vec{\theta}_o||_2^2) \approx E_{\vec{\theta}|\vec{\mu}_t, \boldsymbol{C}_t}\left(\vec{\theta}^T\vec{\theta} - \vec{\theta}^T\vec{\mu}_t\right) + const \tag{133}$$

$$= Tr(\boldsymbol{C}_t) + const, \tag{134}$$

where Tr is the trace.

To optimize the accuracy of the predicted responses, we pick the stimulus which will minimize the M.S.E. 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 M.S.E. if we pick $\vec{s}_{t+1}$ is,

$$E_{\vec{\theta}}E_{r_{t+1}|\vec{s}_{t+1},\vec{\theta}}Tr(\boldsymbol{C}_{t+1}) = E_{\rho_{t+1}}E_{r_{t+1}|\rho_{t+1}}(Tr\boldsymbol{C}_{t+1}) \tag{135}$$

$$= E_{\rho_{t+1}}E_{r_{t+1}|\rho_{t+1}}Tr(\boldsymbol{C}_t - \frac{\boldsymbol{C}_t\vec{s}_{t+1}D(r_{t+1},\rho_{t+1})\vec{s}_{t+1}^T\boldsymbol{C}_t}{1+D(r_{t+1},\rho_{t+1})\vec{s}_{t+1}^T\boldsymbol{C}_t\vec{s}_{t+1}}) \tag{136}$$

$$= Tr(\boldsymbol{C}_t) - E_{\rho_{t+1}}E_{r_{t+1}|\rho_{t+1}}\frac{D(r_{t+1},\rho_{t+1})\vec{s}_{t+1}^T\boldsymbol{C}_t\boldsymbol{C}_t\vec{s}_{t+1}}{1+D(r_{t+1},\rho_{t+1})\vec{s}_{t+1}^T\boldsymbol{C}_t\vec{s}_{t+1}} + const. \tag{137}$$

The expected M.S.E. 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 the expected M.S.E. 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 2.5.1.

2.10.4 Spherical symmetry of $p_{opt}(x_2,\ldots,x_{\dim(\vec{\theta})}|x_1)$

To derive the optimal asymptotic design in Section 2.7.1, we used the fact that there always exists an optimal $p(x_2,\ldots,x_{\dim(\vec{\theta})}|x_1)$ which is spherically symmetric. Here we prove this claim using a proof by contradiction: let us assume that some distribution $\hat{p}(\vec{x}) = \hat{p}(x_1)\hat{p}(x_2,\ldots,x_{\dim(\vec{\theta})}|x_1)$ with non-symmetric $\hat{p}(x_2,\ldots,x_{\dim(\vec{\theta})}|x_1)$ maximizes our objective function $F()$. We will show that we can construct a spherically symmetric $p^*(x_2,\ldots,x_{\dim(\vec{\theta})}|x_1)$ 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 $\hat{p}(x_2,\ldots,x_{\dim(\vec{\theta})}|x_1)$ over all possible rotations Ψ_R. We define these rotations as

$$\Psi_R p(\vec{x}) = p(R\vec{x}) \tag{138}$$

$$R = \begin{bmatrix} 1 & 0 \\ 0 & R_{\dim(\vec{\theta})-1} \end{bmatrix}, \tag{139}$$

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

$$F\left(\Psi_R p(\vec{x})\right) = \log\left|\int D(r, x_1\theta_1)\vec{x}\vec{x}^T p(R\vec{x})d\vec{x}\right| \tag{140}$$

$$= \log\left|\int D(r, x_1\theta_1)R^T\vec{x}'\vec{x}'^T Rp(\vec{x}')d\vec{x}'\right| \tag{141}$$

$$= 2\log|R| + F(p(\vec{x})) \tag{142}$$

$$= F\left(p(\vec{x})\right). \tag{143}$$

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,

$$p^*(\vec{x}) = E_{\Psi_R}(\Psi_R(\hat{p}(\vec{x})). \tag{144}$$

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

$$F(p^*(\vec{x})) = F(E_{\Psi_R}\Psi_R\hat{p}(\vec{x})) \geq E_{\Psi_R}F(\Psi_R\hat{p}(\vec{x})) = F(\hat{p}(\vec{x})). \tag{145}$$

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

2.10.5 Support of $p_{opt}(\vec{x})$

In Section 2.7.2 we derived some analytical results regarding the relative efficiency of the info. max. 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 Eqn. 83 by setting the derivative of Eqn. 83 equal to zero. The derivative of Eqn. 83 with respect to x_1 is the cubic polynomial

$$h(x_1) = -\dim(\vec{\theta})||\vec{\theta}||_2 x_1^3 - 2\dim(\vec{\theta})x_1^2 + \dim(\vec{\theta})m^2||\vec{\theta}||_2 x_1 + 2m^2. \tag{146}$$

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 $h(x_1)$,

$$\frac{d^2h(x_1)}{dx_1^2} = -6d||\vec{\theta}||_2 x_1 - 4\dim(\vec{\theta}). \tag{147}$$

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 $h(m) = -2(\dim(\vec{\theta}) - 1)m^2$ which is negative for all $\dim(\vec{\theta}) > 1$. 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:

$$\log \left| E_{\vec{x}} \exp(x_1||\vec{\theta}||_2)\vec{x}\vec{x}^T \right| = \dim(\vec{\theta})x_1||\vec{\theta}||_2 + \log x_1^2 + (\dim(\vec{\theta}) - 1)\log(m^2 - x_1^2) \tag{148}$$

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 $\dim(\vec{\theta}) \to \infty$. To compute the limit, we divide both sides of the equation $h(x_1) = 0$ by $\dim(\vec{\theta})$ and take the limit:

$$\lim_{\dim(\vec{\theta})\to\infty} \frac{h(x_1)}{\dim(\vec{\theta})} = x_1(-||\vec{\theta}||_2 x_1^2 - 2x_1 + m^2||\vec{\theta}||_2). \tag{149}$$

The roots of this polynomial are

$$x_1 = 0 \qquad \& \qquad x_1 = \frac{-1 \pm \sqrt{1 + ||\vec{\theta}||_2^2 m^2}}{||\vec{\theta}||_2} \tag{150}$$

We showed earlier that the optimal value of x_1 must be greater than zero. So as $\dim(\vec{\theta}) \to \infty$, 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,

$$\lim_{||\vec{\theta}||_2 \to \infty} \frac{h(x_1)}{||\vec{\theta}||_2} = -\dim(\vec{\theta})x_1^3 + \dim(\vec{\theta})m^2 x_1, \tag{151}$$

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.

CHAPTER III

NON-GREEDY OPTIMIZATION FOR LEARNING TEMPORAL FEATURES.

In this chapter we consider the problem of non-greedy optimization of the conditional mutual information between $\vec{\theta}$ and a sequence of observations, $\boldsymbol{r_{t+1:t+b}}$, with respect to a sequence of inputs, $\boldsymbol{s_{t+1:t+b}}$. The goal is to select the sequence of inputs which will provide the most information about $\vec{\theta}$. We derive two important results for solving this problem in the case of the infinite horizon, $b \to \infty$. First we show that as $b \to \infty$, maximizing the mutual information is equivalent to maximizing the average Fisher information per trial. The Fisher information is independent of the prior but depends on the unknown parameters. Therefore, we compute the expected Fisher information with respect to $\vec{\theta}$ using our posterior on $\vec{\theta}$. Second we show that for any infinitely long sequence, there exists a stochastic process such that the average information per trial for the original sequence equals the average information per trial for any sequence sampled from the stochastic process. Consequently, in the infinite horizon we can find an optimal stochastic process by solving a convex optimization problem and then sampling this process to generate an optimal sequence of inputs. We use these results to find an approximately optimal design when we restrict $p(\vec{s})$ to be a Gaussian distribution. Finally we present some simulation results showing that using the optimized Gaussian design leads to faster convergence to the best model of a neuron. The results in this chapter are a natural extensions of previous work in [114] which considered the batch optimization problem when the input at time t did not depend on past stimuli or responses.

3.1 Introduction

Neurons have an amazing ability to store and integrate information over time. One of the most obvious examples is auditory processing. Since sound is a temporal signal, auditory neurons need to integrate information over time in order to detect acoustical features [72, 62]. Neurons in the visual system also exhibit tuning to temporal signals. For example, neurons in the MT region of visual cortex are selective for the direction of motion in the scene [3, 131]. Integrating information over time is also important for higher level processing. LIP neurons integrate evidence over time for the purpose of making decisions [77]. These examples show that we need to understand how neurons integrate information over time if we want to unravel the neural code. Towards this end, we would like to optimize neurophysiology experiments to identify the dependence of a neuron's response on the temporal features in the input. To accomplish this task, we need to create stimuli with complex temporal features; e.g. movies and sounds. This chapter considers the problem of designing optimal stimuli for neurons which integrate information over time. The goal of this chapter is to address some of the limitations of the greedy methods presented in Chapter 2.

Consider the simple example of a neuron in MT which is selective to the direction of motion [131]. To determine the direction tuning of an MT neuron we need to create movies with objects moving in different directions. We can think of these movies as sequences of images which are highly-correlated over time. Each stimulus, $\vec{x}_t$, is a still-image and the input at time t, $\vec{s}_t = \{\vec{x}_{t-t_k}, \ldots, \vec{x}_t\}$, is a short movie constructed by presenting these still images in rapid succession. Using the methods presented in the previous chapter there are two ways to optimize the inputs. In Chapter 2 we presented a method which at time t finds the optimal value of $\vec{x}_{t+1}$. The other components of the input, $\{\vec{x}_{t-t_k+1}, \ldots, \vec{x}_t\}$, were determined by past trials and could not be changed. Clearly, this approach can lead to very poor designs. Since this approach only looks one step ahead, i.e. it is greedy, we never plan far enough into

the future to create stimuli with the temporal features that might drive the neuron to fire; e.g a correlated sequence of still-images of a moving object.

The second way to optimize the stimulus using the methods presented in Chapter 2 is to look $t_k + 1$ steps into the future at time t and optimize $\vec{s}_{t+1+t_k} = \{\vec{x}_{t+1}, \ldots, \vec{x}_{t+1+t_k}\}$. In this case, we are planning far enough into the future that we can create stimuli with complex temporal features. However, this approach would consider only the information in the data point $(\vec{s}_{t+1+t_k}, r_{t+1+t_k})$ when picking $\vec{s}_{t+1+t_k}$. This method therefore ignores the information in the trials in between time t and time $t + 1 + t_k$. To be optimal, we need to take the informativeness of these trials into account when optimizing $\vec{s}_{t+1+t_k}$. In Chapter 4 we modify our greedy algorithm to address this problem. In this chapter, however, we take a different approach.

In this chapter, we consider the problem of computing the optimal sequence of future stimuli $\{\vec{x}_{t+1}, \vec{x}_{t+2}, \ldots, \vec{x}_{t+b}\}$ in the limit $b \to \infty$, i.e. the infinite horizon. The asymptotic properties of the optimal design lead to a simpler optimization problem in the infinite horizon [83, 114]. In particular, instead of finding the optimal sequence, we can optimize the design with respect to some sufficient statistic. This problem is easier to solve in the infinite horizon because in the infinite horizon the set of sufficient statistics for all possible designs is convex.

Consider the simple case of a neuron with a purely spatial receptive field such as a simple cell in V1. A simple cell responds to bars oriented at different angles [75]. Naturally to determine the cell's orientation tuning we should present stimuli containing bars oriented at different angles. Suppose we conduct b trials using a fixed experimental design. Clearly the information collected from these trials depends entirely on the number of times each bar oriented at a specific angle is presented. The exact ordering of the stimuli does not matter because whether we present a vertical bar before or after a horizontal bar will not change the informativeness of the data collected. Hence, to compute the informativeness of a particular design we only

need to specify the fraction of trials on which each input gets picked. In this case we can specify the design as a probability distribution, $p(\vec{x})$.

Finding the optimal $p(\vec{x})$ is difficult in the finite horizon because the set of all valid $p(\vec{x})$ is non-convex. If we will conduct b trials where b is finite then a design is valid only if

$$p(\vec{x}) = \frac{n}{b} \quad n = 0, 1, \ldots, b \quad \forall \vec{x} \tag{152}$$

$$\int p(\vec{x}) = 1. \tag{153}$$

The first constraint simply ensures that $p(\vec{x}) \times b$ is an integer which specifies the number of trials on which we present $\vec{x}$. Unfortunately this constraint makes finding the optimal $p(\vec{x})$ difficult because the set of $p(\vec{x})$ satisfying this constraint is non-convex. However, as $b \to \infty$ the set of valid designs converges to a set which is convex. In the limit $b \to \infty$, $p(\vec{x})$ can be any valid probability distribution on the stimulus; i.e. the only constraints are that $p(\vec{x})$ is positive and sums to one. In general, optimizing a function over a convex set is much easier then optimizing over a non-convex domain [83]. We can, for example, use gradient methods because we can make a small perturbation to $p(\vec{x})$ which does not violate the constraints. In the finite horizon, the only way to perturb $p(\vec{x})$ is to increase the frequency count of one stimulus by one and decrease the count of another stimulus by one; this is not a sufficiently small perturbation to allow gradient methods to work.

The above reasoning clearly depends on the critical assumption that the order of the stimuli does not matter. We used this assumption to conclude that only the relative frequency of each stimulus mattered. If the receptive field of a neuron is purely spatial this assumption is valid. However, we want to optimize neurophysiology experiments to probe the temporal structure of a neuron's receptive field. In this case we obviously cannot reshuffle the stimuli without changing the informativeness of the data. For example, consider a movie of a moving object. Clearly, reshuffling the

frames in this movie would drastically change the amount of information this movie provides about the direction tuning of neurons in MT. Hence, it is not obvious that the principles which make the infinite horizon an easier problem still apply when a neuron has a temporal receptive field.

For neurons with temporal receptive fields the sufficient statistic is still the relative frequency of stimuli with different features. The only difference is that the features now have a temporal component to them; i.e. the features are sequences of the instantaneous stimuli, $\{\vec{x}_t, \vec{x}_{t+1}, \ldots\}$. To illustrate this idea consider a simple auditory neuron which responds to the amplitude envelope of the sound being played [58]. Suppose the amplitude of the sound is simply a train of square pulses of varying duration and that we know this neuron is tuned to pulses of a particular duration. Clearly, to compute the informativeness of the design we only need to know the number of times we present pulses of particular durations. The key difference from the spatial case is that in this case the features of the stimulus are coupled to the relative frequency with which each feature is presented. If we increase the duration of one of the pulses in our stimulus set then it necessarily takes more time to collect a single data point using that longer duration pulse. Thus, if the duration of our experiment is fixed, to increase the duration of one of the pulses we must either present that stimulus fewer times, the other stimuli fewer times, or decrease the duration of the other stimuli. We can continue to specify the design as a probability distribution on the stimuli but in the temporal case this is necessarily a distribution on sequences of stimuli, $p(\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_b)$. The design is therefore a stochastic process which captures the interdependence of the temporal features and the frequency of their occurrence.

We begin in Section 3.2 by deriving our objective function in the infinite horizon. We show that as $b \to \infty$ maximizing the mutual information is equivalent to maximizing the average information per trial. In this section we prove that under

suitable conditions, finding the optimal sequence as $b \to \infty$ is equivalent to finding an optimal stochastic process. The optimal process provides a convenient representation for the relative frequency with which inputs containing different spatio-temporal features should be selected. The relative frequency of these features is a sufficient statistic for computing the informativeness of the design. Hence, we can show that all infinitely long sequences with the same sufficient statistics are equally informative. These results generalize some conjectures made in [114] which only considered the case where a neuron's response is independent of past responses and stimuli. In Section 3.3.2 we show how the optimal stochastic process may be computed in the case of the canonical Poisson. In Section 3.4 we present some simulation results illustrating the benefit of designing experiments using the methods presented in this chapter.

A major advantage of the methods presented in this chapter is that they are in principle much easier to implement in actual experiments then the methods presented in Chapter 2. If we pick an optimal sequence of length b, where b is finite, then the speed with which we can compute the optimal sequence is a huge bottleneck. This optimization must be performed in the time it takes to present b stimuli, otherwise the experimenter must wait while the algorithm computes the next optimal sequence. In contrast, the methods presented in this chapter compute an optimal distribution on the stimuli. Thus while re-optimizing the design, i.e computing a new distribution, we can continue to draw stimuli from the most recent design.

3.2 Maximizing the average information per trial is optimal as $b \to \infty$

The main point of this section is that as $b \to \infty$ to compute the informativeness of a sequence we do not need to know the exact sequence but only certain sufficient statistics of the sequence. Rather than optimizing the mutual information with respect to sequences, we can optimize the mutual information with respect to the sufficient

statistics of the optimal sequence. Since the sufficient statistics define a stochastic process we can generate an optimal sequence just by sampling the optimal stochastic process. The mapping from sequences to sufficient statistics is many to one. Therefore, optimizing over the sufficient statistics is easier because in some sense the space of sufficient statistics is smaller than the space of all possible sequences.

To motivate the results in this section we start by considering the simple case where a neuron's response, r_t, depends only on its instantaneous input, $\vec{x}_t$ [114]. In this case the mutual information of any set of input-output pairs $\{(\vec{x}_1, r_1), \ldots, (\vec{x}_t, r_t)\}$ is independent of the order in which these data points are collected. We can easily show this by writing down the mutual information

$$I(\vec{\theta}; \{(\vec{x}_1, r_1), \ldots, (\vec{x}_t, r_t)\}) = H(p(\vec{\theta})) - H(p(\vec{\theta}|\{(\vec{x}_1, r_1), \ldots, (\vec{x}_t, r_t)\})) \tag{154}$$

$$= H(p(\vec{\theta})) - E_{\vec{\theta}} E_{\boldsymbol{r_{1:t}}|\boldsymbol{s_{1:t}},\vec{\theta}} \log \left| \boldsymbol{C}_0^{-1} + \sum_{i=1}^{t} J_{obs}(r_i, \vec{x}_i) \right|. \tag{155}$$

We use H to denote the entropy of its argument and E to denote the expectation over the random variable denoted in its subscript. J_{obs} is the observed Fisher information. To derive this expression for the mutual information, we use our Gaussian approximation of the posterior to approximate its entropy. Clearly we can rearrange the trials in any order without affecting the value of this expression. Thus, a set of sufficient statistics for describing the sequence is the number of times each stimulus is picked. We can thus represent a sequence as a distribution $p(\vec{x})$ which specifies the fraction of trials on which each stimulus gets picked. It seems natural to conjecture that in some sense optimizing over the set of sequences should be equal to optimizing over the set of distributions $p(\vec{x})$ [114].

In this section, we generalize this result to the case where the neuron's response depends on past stimuli. Since the neuron depends on past stimuli, we can no longer shuffle the order of the trials without changing the informativeness of the data. Thus,

it is not immediately obvious that we can compute the informativeness of a sequence without knowing the actual sequence. However, in this section we show that if the impulse response of a neuron is finite, then we can in fact find a set of statistics which are sufficient for computing the informativeness of any sequence. A finite impulse response means that the input at time t only affects a finite number of future responses. Probabilistically this means the conditional response is independent of all but the most recent $t_k + 1$ stimuli,

$$p(r_t|\vec{x}_{-\infty}, \ldots, \vec{x}_t) = p(r_t|\vec{x}_{t-t_k+1}, \ldots, \vec{x}_t) = p(r_t|\vec{s}_t) \tag{156}$$

$$\vec{s}_t \triangleq \{\vec{x}_{t-t_k}, \ldots, \vec{x}_t\}. \tag{157}$$

When $t_k = 0$, r_t depends only on $\vec{x}_t$.

The main result of this section is Theorem 1. This theorem states that instead of optimizing the sequence, we can optimize a stochastic process with respect to some sufficient statistics, and then generate a sequence by sampling this process,

Theorem 1. *Suppose* $p(r_t|\{\vec{x}_{-\infty}, \ldots, \vec{x}_t\}, \vec{\theta}) = p(r_t|\vec{x}_{t-t_k}, \ldots, \vec{x}_t, \vec{\theta})$*, then*

$$\lim_{b\to\infty} \max_{\vec{x}_1,\ldots,\vec{x}_b} I(\vec{\theta}; \{(\vec{x}_1, r_1), \ldots, (\vec{x}_b, r_b)\}) - \dim(\vec{\theta}) \log b$$

$$= \lim_{b\to\infty} I(\vec{\theta}; \{(\vec{x}'_1, r_1), \ldots, (\vec{x}'_b, r_b)\}) - \dim(\vec{\theta}) \log b \tag{158}$$

$$\{\vec{x}'_1, \ldots, \vec{x}'_b\} \sim p_{opt}(\vec{s}) \tag{159}$$

$$p_{opt}(\vec{s}) = \arg \max_{p(\vec{s})\in\mathcal{P}_{\vec{f}}} E_{\vec{\theta}} \log \left| E_{p(\vec{s})} J_{exp}(\vec{s}) \right|. \tag{160}$$

where $\mathcal{P}_{\vec{f}}$ *is the set of all possible marginal distributions corresponding to a* $t_k + 1$ *stationary,* φ*-irreducible Markov process as defined in Definition 1.*

Theorem 1 has two parts. The first part says that a sequence which maximizes the mutual information is equivalent to a sequence which maximizes the average information per trial; i.e the rate at which information is acquired. The second part of the theorem states that as $b \to \infty$, the informativeness of a sequence, as measured

by the information rate, depends only on the marginal distribution of the inputs, $p(\vec{s})$. We can thus generate an optimal sequence by maximizing the information rate with respect to $p(\vec{s})$, Eqn. 160, and then sampling the stochastic process defined by $p(\vec{s})$. As we explain later, this procedure turns out to be much easier than finding a sequence which maximizes the mutual information.

The proof of Theorem 1 proceeds in three stages. We start in Section 3.2.1 by showing that as $b \to \infty$, a sequence which maximizes the mutual information is just as informative as a sequence which maximizes the average information per trial. We also show in this section that the marginal distribution $p(\vec{s})$ on subsequences of length $t_k + 1$ stimuli is a sufficient statistic for computing the average information per trial.

The second stage of the proof, Section 3.2.2, establishes an equivalency between sequences and processes. In this section we show that for any infinitely long sequence, we can find an equivalent $t_k + 1$ order stationary, φ-irreducible process such that any infinitely long sequence sampled from this process is as informative as the original sequence.

In the final stage of the proof, Section 3.2.3, we combine the results in Section 3.2.1 and Section 3.2.2 to prove Theorem 1. To simplify the proofs in Section 3.2.2, we will assume $\vec{x}_t$ takes on discrete values. In practice this entails no loss of generality because we can choose the number of different values for $\vec{x}_t$ to be arbitrarily large. Furthermore in actual experiments the values of $\vec{x}_t$ would necessarily be quantized due to the inherent limitations of the physical devices used to create $\vec{x}_t$.

3.2.1 Maximizing the mutual information is equivalent to maximizing the average information per trial.

We begin by showing that as $b \to \infty$ maximizing the mutual information is equivalent to maximizing the average information per trial. If we think of the mutual

information as measuring the total information acquired from b trials then the average information per trial (or the information rate) is just the mutual information normalized by the number of trials. Therefore, it is not surprising that maximizing the mutual information is equivalent as $b \to \infty$ to maximizing the information rate.

If we approximate the posterior distribution as Gaussian then our uncertainty after b trials is inversely proportional to the sum of the information in our prior and the Fisher information of the observations,

$$H\left(p(\vec{\theta}|\boldsymbol{s_{1:b}}, \boldsymbol{r_{1:b}})\right) = E_{\vec{\theta}} E_{\boldsymbol{r_{1:b}}|\boldsymbol{s_{1:b}},\vec{\theta}} \log |\boldsymbol{C}_0^{-1} + \sum_{i=1}^{b} J_{obs}(r_i, \vec{s}_i)^T| + const. \tag{161}$$

The observed Fisher Information depends on the unknown $\vec{\theta}$ and responses. Since these quantities are unknown, we take the expectation with respect to the distribution $p(\vec{\theta}, \boldsymbol{r_{1:b}}|\boldsymbol{s_{1:b}})$. We compute the joint distribution $p(\vec{\theta}, \boldsymbol{r_{1:b}}|\boldsymbol{s_{1:b}})$ using our prior on $\vec{\theta}$, $p(\vec{\theta})$, and the conditional likelihood $p(\boldsymbol{r_{1:b}}|\boldsymbol{s_{1:b}}, \vec{\theta})$. Here we define the Fisher information as some function of the observation and the input rather than using the special structure of the Fisher information for the GLM. We do this because we wish to make the results of this section as general as possible.

The objective function above measures the total information. Consequently, our objective function is not well defined in the limit $b \to \infty$ because the sum of the J_{obs} terms will keep increasing [1]. One way to derive a suitably bounded objective function as $b \to \infty$ is to consider the average information per trial,

$$\lim_{b\to\infty} E_{\vec{\theta}} E_{\boldsymbol{r_{1:b}}|\boldsymbol{s_{1:b}},\vec{\theta}} \log \left| \boldsymbol{C}_0^{-1} + \sum_{i=1}^{b} J_{obs}(r_i, \vec{s}_i) \right| =$$

$$\lim_{b\to\infty} E_{\vec{\theta}} E_{\boldsymbol{r_{1:b}}|\boldsymbol{s_{1:b}},\vec{\theta}} \log |\frac{\boldsymbol{C}_0^{-1}}{b} + \frac{1}{b}\sum_{i=1}^{b} J_{obs}(r_i, \vec{s}_i)| + \dim(\vec{\theta}) \log b. \tag{162}$$

Since $\dim(\vec{\theta}) \log b$ is constant with respect to the inputs, we can just ignore it when

[1] We assume it is possible to pick $\vec{s}_i$ such that the sum of J_{obs} is full rank and leave for future consideration cases where this condition may not be satisfied. If $\sum J_{obs}(r, \vec{s})$ is not full rank then we can apply a transformation to $\vec{\theta}$ such that in the transformed, lower-dimensional coordinates J_{obs} is full rank [114]. In this case most of our methods should continue to work.

optimizing the inputs. Now since the contribution of the prior becomes negligible as $b \to \infty$,

$$\lim_{b\to\infty} \max_{\vec{x}_1,\ldots,\vec{x}_b} E_{\vec{\theta}} E_{\boldsymbol{r_{1:b}}|\boldsymbol{s_{1:b}},\vec{\theta}} \log |\frac{\boldsymbol{C}_0^{-1}}{b} + \frac{1}{b}\sum_{i=1}^{b} J_{obs}(r_i, \vec{s}_i)|$$

$$= \lim_{b\to\infty} \max_{\vec{x}_1,\ldots,\vec{x}_b} E_{\vec{\theta}} E_{\boldsymbol{r_{1:b}}|\boldsymbol{s_{1:b}},\vec{\theta}} \log |\frac{1}{b}\sum_{i=1}^{b} J_{obs}(r_i, \vec{s}_i)|. \tag{163}$$

The result is that we end up maximizing the average information per trial instead of the total information. To derive this well defined objective function we simply normalize the mutual information by subtracting the logarithm of the number of trials.

In the introduction we motivated this chapter by considering some simple examples where it was clear the informativeness of a design depended only on the relative frequency with which stimuli containing different features were presented. Using the objective function derived above we can formalize this idea. The relevant features in this case are the different sequences of stimuli of length $t_k + 1$. Thus, the sufficient statistic is the fraction of trials on which we present each sequence of length $t_k + 1$,

$$\hat{p}_b(\vec{s} = a) = \frac{1}{b}\sum_{i=1}^{b} \delta(\vec{s}_i = a) \tag{164}$$

$$\delta(\vec{s}_i = a) = \begin{cases} 1 & \text{if} \quad \vec{s}_i = a \\ 0 & \text{otherwise} \end{cases}. \tag{165}$$

To prove $\hat{p}_b(\vec{s})$ is sufficient we need to define the empirical distributions

$$\hat{p}_b(r = b|\vec{s} = a) = \frac{\frac{1}{b}\sum_{i=1}^{b} \delta(\vec{s} = a, r = b)}{\hat{p}_b(\vec{s} = a)} \tag{166}$$

$$\delta(\vec{s}_t = a, r_t = b) = \begin{cases} 1 & \text{if} \quad \vec{s}_t = a \quad \& \quad r_t = b \\ 0 & \text{otherwise} \end{cases} \tag{167}$$

These empirical distributions simply count how many times each input and response occurs in the dataset. Using these empirical distributions we can rewrite our objective

function as

$$\lim_{b\to\infty} E_{\vec{\theta}} E_{\boldsymbol{r_{1:b}}|\boldsymbol{s_{1:b}},\vec{\theta}} \log |E_{\hat{p}_b(\vec{s})} E_{\hat{p}_b(r|\vec{s})} J_{obs}(r,\vec{s})|. \tag{168}$$

The expectation only depends on the frequency of pairs $(\vec{s}, r)$, i.e $\hat{p}_b(\vec{s}, r)$. However, the overlap in the inputs necessarily imposes certain constraints on $\hat{p}_b(\vec{s}, r)$. Since the objective function depends only on $\hat{p}_b(\vec{s}, r)$ and not the actual order of the trials, we may replace the outer expectation over $p(\boldsymbol{r_{1:b}}|\boldsymbol{s_{1:b}}, \vec{\theta})$ with an expectation over $p(\hat{p}_b(r|\vec{s})|\hat{p}_b(\vec{s}), \vec{\theta})$. To compute $p(\hat{p}_b(r|\vec{s})|\hat{p}_b(\vec{s}), \vec{\theta})$ we simply sum $p(\boldsymbol{r_{1:b}}|\boldsymbol{s_{1:b}}, \vec{\theta})$ over all $(\boldsymbol{s_{1:b}}, \boldsymbol{r_{1:b}})$ for which the corresponding empirical distributions $\hat{p}_b(\vec{s})$ and $\hat{p}_b(r|\vec{s})$ have the appropriate value,

$$p(\hat{p}_b(r|\vec{s}) = P_1|\hat{p}_b(\vec{s}) = P_2, \vec{\theta}) \propto \int_{(\boldsymbol{r_{1:b}},\boldsymbol{s_{1:b}})\in A} p(\boldsymbol{r_{1:b}}|\boldsymbol{s_{1:b}}, \vec{\theta}) \tag{169}$$

$$A = \{\boldsymbol{r_{1:b}}, \boldsymbol{s_{1:b}} : \hat{p}_b(\vec{s}) = P_2, \hat{p}_b(r|\vec{s}) = P_1\} \tag{170}$$

Using this distribution we can rewrite our objective function as

$$\lim_{b\to\infty} E_{\vec{\theta}} E_{\boldsymbol{r_{1:b}}|\boldsymbol{s_{1:b}},\vec{\theta}} \log |E_{\hat{p}_b(\vec{s})} E_{\hat{p}_b(r|\vec{s})} J_{obs}(r,\vec{s})|$$

$$= \lim_{b\to\infty} E_{\vec{\theta}} E_{p(\hat{p}_b(r|\vec{s})|\hat{p}_b(\vec{s}),\vec{\theta})} \log |E_{\hat{p}_b(\vec{s})} E_{\hat{p}_b(r|\vec{s})} J_{obs}(r,\vec{s})| \tag{171}$$

$$= \lim_{b\to\infty} E_{\vec{\theta}} E_{p(\hat{p}_b(r|\vec{s})|\hat{p}_b(\vec{s}),\vec{\theta})} F(\hat{p}_b(r|\vec{s}), \hat{p}_b(\vec{s})) \tag{172}$$

where F is defined as the log-determinant of the expected Fisher information. Computing the expectation with respect to $p(\hat{p}_b(r|\vec{s})|\hat{p}_b(\vec{s}), \vec{\theta})$ is easy because in the limit $b \to \infty$ we can assume that $p(\hat{p}_b(r|\vec{s})|\hat{p}_b(\vec{s}), \vec{\theta})$ only has support on the true conditional distribution; i.e $\hat{p}_b(r|\vec{s}) = p(r|\vec{s}, \vec{\theta})$. Suppose as $b \to \infty$ we pick $\vec{s} = a$ an infinite number of times. The corresponding observations are drawn from the true distribution $p(r|\vec{s} = a, \vec{\theta})$; hence the empirical distribution $p(r|\vec{s} = a)$ is a consistent estimator of $p(r|\vec{s} = a, \vec{\theta})$ [157]. On the other hand, suppose $\vec{s} = a$ gets chosen only a finite number of times then $\hat{p}_b(r|\vec{s} = a)$ is some random, unknown quantity, which

could be very different from $p(r|\vec{s}=a,\vec{\theta})$. However, since $\vec{s}=a$ is only chosen a finite number of times, the data on these trials makes a negligible contribution to the average information per trial as $b\rightarrow\infty$. Therefore, the value of $\hat{p}_b(r|\vec{s}=a)$ for these inputs is irrelevant. For convenience, we can therefore compute the average information assuming $\hat{p}_b(r|\vec{s})=p(r|\vec{s},\vec{\theta})$,

$$\lim_{b\rightarrow\infty} E_{\vec{\theta}}E_{p(\hat{p}_b(r|\vec{s})|\hat{p}_b(\vec{s}),\vec{\theta})}F(\hat{p}_b(r|\vec{s}),\hat{p}_b(\vec{s}))=\lim_{b\rightarrow\infty} E_{\vec{\theta}}F(p(r|\vec{s},\vec{\theta}),\hat{p}_b(\vec{s})). \tag{173}$$

We present a rigorous argument in Appendix 3.6.1 to support this result.

To summarize, in this section we have shown that as $b\rightarrow\infty$ the mutual information does not provide a well defined objective function. Therefore, instead of maximizing the mutual information, we maximize the average information per trial,

$$\lim_{b\rightarrow\infty} I(\vec{\theta};\{(\vec{x}_1,r_1),\ldots,(\vec{x}_b,r_b)\})-\dim(\vec{\theta})\log b\propto E_{\vec{\theta}}\log|\lim_{b\rightarrow\infty} E_{\hat{p}_b(\vec{s})}E_{p(r|\vec{s},\vec{\theta})}J_{obs}(\vec{s},r)| \tag{174}$$

The key implication of this result is that our objective function only depends on the marginal distribution $p(\vec{s})$ and not the actual order of the stimuli. In principle we can therefore optimize the average information with respect to $p(\vec{s})$ which might be easier than computing the sequence which maximizes the mutual information. However, we can only optimize $p(\vec{s})$ if we can restrict our attention to $p(\vec{s})$ which correspond to valid sequences, i.e there must exist a sequence $\vec{x}_1,\ldots,\vec{x}_b$ such that $\hat{p}_b(\vec{s})=p(\vec{s})$. The next part of the proof of Theorem 1 is to establish restrictions on $p(\vec{s})$ to ensure we can generate valid sequences by sampling it.

3.2.2 Equally informative stochastic processes.

Since $\hat{p}_b(\vec{s})$ is a sufficient statistic for the informativeness of a design we could in principle optimize the design with respect to $\hat{p}_b(\vec{s})$; i.e. determine what fraction of the trials we should devote to each feature. However as we explained in the introduction if b is finite the constraint that $\hat{p}_b(\vec{s})\times b$ is an integer is problematic. Furthermore,

given $\hat{p}_b(\vec{s})$ we need to create a sequence of inputs for which the sufficient statistic would be $\hat{p}_b(\vec{s})$. Both problems turn out to be much easier to solve if we consider the infinite horizon. In the infinite horizon, we can drop the constraint that $\hat{p}_b(\vec{s}) \times b$ is an integer and just optimize the sufficient statistic with respect to some suitable set of stochastic processes. Furthermore, to generate a sequence with the desired sufficient statistic we can simply sample the stochastic process. The main result of this section is the following lemma establishing an equivalency between sequences and stochastic processes in the infinite horizon,

Lemma 1. *Let $\{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_b\}$ be some sequence of stimuli with corresponding empirical distribution $\hat{p}_b(\vec{s})$, then there exists a $t_k + 1$ order stationary, φ-irreducible, Markov process with stationary distribution $p(s)$ such that*

$$E_{\vec{\theta}} \log \left| \lim_{b\to\infty} E_{\hat{p}_b(\vec{s})} E_{p(r|\vec{s},\vec{\theta})} J_{obs}(\vec{s}, r) \right| = E_{\vec{\theta}} \log \left| \lim_{b\to\infty} E_{\hat{p}'_b(\vec{s})} E_{p(r|\vec{s},\vec{\theta})} J_{obs}(\vec{s}, r) \right| \tag{175}$$

$$\lim_{b\to\infty} \hat{p}_b(\vec{s}) = p(\vec{s}) \tag{176}$$

$$\lim_{b\to\infty} \hat{p}'_b(\vec{s}) = p(\vec{s}), \tag{177}$$

where $\hat{p}'_b(\vec{s})$ is the empirical distribution for a sequence $\{\vec{x}'_1, \vec{x}'_2, \ldots, \vec{x}'_b\}$ sampled from the Markov process defined by $p(\vec{s})$.

This lemma says that for any sequence, we can find an optimal process such that for all sequences sampled from this process, the average information per trial equals the average information per trial of the original sequence. The conditions that the process be $t_k + 1$ order stationary and φ-irreducible ensure that we can generate a valid sequence by sampling the process.

We begin by defining some notation which makes it easy to express necessary and sufficient conditions for a sequence to be valid. By definition $\vec{s}_t$ is a sequence of $t_k + 1$ stimuli $\vec{s}_t = \{\vec{x}_{t-t_k}, \ldots, \vec{x}_t\}$. We use superscripts on $\vec{s}_t$ to denote the sub-vector of $\vec{s}_t$

corresponding to the stimulus, $\vec{x}_t$, at different times,

$$s_t^{-j} = \vec{x}_{t-j} \tag{178}$$

$$s_t^{-j:-i} = \{\vec{x}_{t-j}^T, \vec{x}_{t-j+1}^T, \ldots, \vec{x}_{t-i-1}, \vec{x}_{t-i}^T\}^T. \tag{179}$$

Thus a necessary condition for a valid sequence is that $s_t^{-t_k+1:0} = s_{t+1}^{-t_k:-1} \quad \forall\, t$. A distribution $p(\vec{s})$ is valid only if it assigns non-zero probability to sequences which satisfy this constraint.

A process is a $t_k + 1$ order stationary process if the likelihood of any subsequence of $t_k + 1$ or fewer stimuli $\{\vec{x}_{t-t_k}, \ldots, \vec{x}_t\}$ is independent of the time t at which this sequence occurs.

Definition 1. *The stochastic process defined by the joint distribution $p(\vec{x}_1, \vec{x}_2, \ldots)$ is a $t_k + 1$ order stationary process if*

$$p(\vec{x}_{t-t_k}, \ldots, \vec{x}_t) = p(\vec{x}_{t+\Delta-t_k}, \ldots, \vec{x}_{t+\Delta}) \quad \forall t, \Delta. \tag{180}$$

For example, consider the case of a visual neuron where each $\vec{x}_t$ is an image and $\{\vec{x}_{t-t_k}, \ldots, \vec{x}_t\}$ is a short movie constructed by playing the still images in rapid succession. Hence, the distribution of $p(\{\vec{x}_{t-t_k}, \ldots, \vec{x}_t\})$ is the probability that we play a specific movie, $\{\vec{x}_{t-t_k}, \ldots, \vec{x}_t\}$, to the neuron at time t. Eqn. 180 says that the probability of playing a particular movie at time t must equal the probability of playing that same movie at time $t + \delta$.

Clearly, not all marginal distributions $p(\vec{s}_t) = p(\vec{x}_{t-t_k}, \ldots, \vec{x}_t)$ define a valid $t_k + 1$ stationary process. We can show that for a t_k+1 order stationary process the marginal distributions on sub-sequences of length t_k+1 or shorter must be independent of time. This conclusion leads to the following necessary and sufficient conditions for a $t_k + 1$ order stationary process.

Lemma 2. *We can construct a $t_k + 1$ order stationary process with marginal distribution $p(\vec{s}) = p(\vec{x}_{t-t_k}, \ldots, \vec{x}_t)$ if and only if*

$$p(s^{-i:0}) = p\left(s^{-(1+i):-1}\right) \quad \{i : 0 \leq i < t_k, i \in \mathcal{Z}\}, \tag{181}$$

where $\mathcal{Z}$ is the set of integers.

The details of the proof are not particularly important so we leave the proof for Appendix 3.6.2.

We can think of the $t_k + 1$ order stationary process $p(\vec{x}_1, \vec{x}_2, \ldots)$ as a stationary Markov process, $p(\vec{s}_1, \vec{s}_2, \ldots)$, where the transition matrix is constructed so as to enforce the constraint $s_t^{-t_k+1:0} = s_{t+1}^{-t_k:-1} \quad \forall t$. In Appendix 3.6.2 we prove using induction that if $p(\vec{s})$ satisfies Lemma 2 then we can construct a Markov process with stationary distribution $p(\vec{s})$ for which $s_t^{-t_k+1:0} = s_{t+1}^{-t_k:-1} \quad \forall t$. Consequently we say $p(\vec{s})$ is a $t_k + 1$ order stationary process; we use "$t_k + 1$ order" to describe a stationary Markov process which satisfies the additional constraint $s_t^{-t_k+1:0} = s_{t+1}^{-t_k:-1}$.

We can prove Lemma 1 by showing that the empirical distribution $\hat{p}_b(\vec{s})$ for any sequence converges to a distribution $p(\vec{s})$ which defines a $t_k + 1$ order stationary, φ-irreducible Markov process. Clearly, for finite b, $\hat{p}_b(\vec{s})$ may not define a $t_k + 1$ order stationary process because we can easily imagine sequences for which the empirical distribution violates Lemma 2. Consider the simple case, $t_k = 1$ and the sequence $\{\vec{x}_t : \vec{x}_1 = a_1, \vec{x}_t = a_2 \forall t > 1\}$. In this case,

$$\hat{p}_b(\vec{s}) = \begin{cases} \frac{1}{b-1} \text{ if } \vec{s} = \{a_1, a_2\} \\ \frac{b-2}{b-1} \text{ if } \vec{s} = \{a_2, a_2\} \end{cases} \tag{182}$$

Clearly this does not satisfy Definition 1 because the marginal distribution $\hat{p}_b(s^{-1})$ has support on a_1 but $\hat{p}_b(s^0)$ does not. The problem is clearly the "edge effects" due to finite b. We can reasonably expect that in the limit $b \to \infty$ these effects

become negligible. This leads to Lemma 3 which says that as $b \to \infty$ the empirical distribution $\hat{p}_b(\vec{s})$ for any sequence converges to a $t_k + 1$ order stationary process.

Lemma 3. *Let $\{\hat{p}_1(\vec{s}), \hat{p}_2(\vec{s}), \ldots, \hat{p}_b(\vec{s})\}$ be a sequence of empirical distributions corresponding to some sequence $\{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_b\}$, then*

$$\lim_{b\to\infty} \hat{p}_b(\vec{s}) = p(\vec{s}) \in \mathcal{P}_{\vec{f}}, \tag{183}$$

where $\mathcal{P}_{\vec{f}}$ is the set of $t_k + 1$ order stationary, φ-irreducible processes.

Proof: The reasoning is straightforward. The sequence $\{s_1^{-i+1}, s_2^{-i+1}, \ldots\} = \{\vec{x}_{t_k+2-i}, \vec{x}_{t_k+3-i}, \ldots\}$ is nearly the same as the sequence $\{s_1^{-i}, s_2^{-i}, \ldots\} = \{\vec{x}_{t_k+1-i}, \vec{x}_{t_k+2-1}, \ldots\}$ except delayed by one time step. The only difference is a finite number of stimuli at the start and end of these sequences. Thus, if we compute the empirical distributions, $\hat{p}_b(s^{-i})$ and $\hat{p}_b(s^{-i+1})$ they should be equal in the limit $b \to \infty$. We show this more formally by starting with the definition of $\hat{p}_b(\vec{s})$. For $0 \leq i < t_k$,

$$\hat{p}_b(\overleftarrow{s}^{-i:0} = a) = \frac{1}{b - t_k} \sum_{t=t_k+1}^{b} \delta(\overleftarrow{s}_t^{-i:0} = a) = \frac{1}{b - t_k} \sum_{t=t_k+1}^{b} \delta(\{\vec{x}_{t-i}, \ldots, \vec{x}_t\} = a) \tag{184}$$

$$\hat{p}_b(\overleftarrow{s}^{-(i+i):-1} = a) = \frac{1}{b - t_k} \sum_{t=t_k+1}^{b} \delta(\overleftarrow{s}_t^{-(i+i):-1} = a) = \frac{1}{b - t_k} \sum_{t=t_k+1}^{b} \delta(\{\vec{x}_{t-i-1}, \ldots, \vec{x}_{t-1}\} = a) \tag{185}$$

$$= \frac{1}{b - t_k} \sum_{t=t_k}^{b-1} \delta(\overleftarrow{s}_t^{-i:0} = a) \tag{186}$$

$$= \frac{1}{b - t_k} \delta(\overleftarrow{s}_{t_k}^{-i:0} = a) + \frac{1}{b - t_k} \sum_{t=t_k+1}^{b} \delta(\overleftarrow{s}_t^{-i:0} = a)$$

$$- \frac{1}{b - t_k} \delta(\overleftarrow{s}_{b+1}^{-i:0} = a). \tag{187}$$

As $b \to \infty$ the first and last terms in Eqn. 187 go to zero. Therefore,

$$\lim_{b\to\infty} \hat{p}_b(\overleftarrow{s}^{-(i+1):-1} = a) = \frac{1}{b - t_k} \sum_{t=t_k+1}^{b} \delta(\overleftarrow{s}_t^{-i:0} = a) \tag{188}$$

$$= \lim_{b\to\infty} \hat{p}_b(\overleftarrow{s}^{-i:0} = a). \tag{189}$$

Thus by Lemma 2 the empirical distribution converges to a distribution which defines a $t_k + 1$ order stationary Markov process.

To complete the proof of Lemma 3 we also need to show that the limiting distribution $p(\vec{s})$ defines a φ-irreducible Markov process. By definition a Markov process is irreducible if it is possible to get from any state to any other state in a finite number of steps. For $p(\vec{s})$, the different states are the possible values for $\vec{s}$. φ-irreducible is a less restrictive notion of irreducibility in which it is possible to transition from any state to any state in φ in a finite number of steps. For our purposes, we define φ as all inputs on which $p(\vec{s})$ has positive support ,

$$\varphi = \{a : \lim_{b \to \infty} \hat{p}_b(\vec{s} = a) > 0\}. \tag{190}$$

Since we assume that $\vec{x}_t$ is quantized, there are a finite number of possible values for $\vec{s}$. In this case, φ is just the set of the sub-sequences $\{\vec{x}_{t-t_k}, \ldots, \vec{x}_t\}$ which occur infinitely often in the sequence $\{\vec{x}_1, \vec{x}_2, \ldots\}$. Since $\vec{x}_t$ is discrete, the only way to create an infinitely long sequence is if at least one of the values of $\vec{s}_t$ is repeated infinitely often; therefore φ is necessarily not empty. Since each $a \in \varphi$ appears an infinite number of times in the original sequence, it is always possible to transition from any state to any state in φ in a finite number of steps. We can therefore always find an equivalent, with respect to average information, Markov process, in which it is possible to transition from any state to any state in φ in a finite number of steps. Thus, we can without loss of generality assume that for any sequence, $\hat{p}_b(\vec{s})$ converges to a stationary, φ-irreducible Markov chain with stationary distribution $p(\vec{s})$. This completes the proof of Lemma 3.

Ensuring that $p(\vec{s})$ defines a φ-irreducible Markov process guarantees that if we sample this process, the empirical distribution, $\hat{p}_b(b)$, converges to $p(\vec{s})$. In Appendix 3.6.2 we show how we can compute a transition matrix $p(\vec{s}_t|\vec{s}_{t-1})$ from $p(\vec{s})$. We also show that sampling this distribution produces a sequence of inputs for which $s_t^{-t_k+1:0} = s_{t+1}^{-t_k:-1} \quad \forall t$ and has stationary distribution $p(\vec{s}_t) = p(\vec{s})$. Since this Markov

process is φ-irreducible the empirical distribution $\hat{p}_b(b)$ must be a consistent estimator of $p(\vec{s})$ [128].

In the previous section we showed that $p(\vec{s})$ is a sufficient statistic for the average information per trial of a sequence. In this section, we showed that for any sequence, we can construct a t_k+1 order stationary, φ-irreducible process for which the empirical distributions of the original sequence and any sequence sampled from the process converge to the same distribution $p(\vec{s})$. Consequently, it follows that the original sequence and any sequence sampled from $p(\vec{s})$ are equivalent with respect to the average information per trial as $b \to \infty$.

3.2.3 Sampling the optimal t_k+1 stationary process produces a maximally informative sequence.

The proof of Theorem 1 follows almost immediately from the results in the previous two sections. In Section 3.2.1, we showed that as $b \to \infty$ maximizing the mutual information is equivalent to maximizing the average information per trial, Eqn. 174. The results in Section 3.2.2 show that for any sequence we can find an equally informative process defined by $p(\vec{s})$. Thus, we can maximize the average information per trial with respect to $p(\vec{s})$ where $p(\vec{s})$ is a $t_k + 1$ order stationary, φ irreducible process. We can then sample this process to generate a sequence which maximizes the average information per trial. This completes the proof of Theorem 1.

Computing the optimal distribution is a well defined optimization problem because the set $\mathcal{P}_{\vec{f}}$ is a compact, convex space. As a result, the maximum

$$p_{opt}(\vec{s}) = \arg \max_{p(\vec{s}) \in \mathcal{P}_{\vec{f}}} E_{\vec{\theta}} \log |E_{p(s)} J_{exp}(\vec{s})| \tag{191}$$

is well defined because the log-determinant is a concave function. The maximizer, $p_{opt}(\vec{s})$, may not be unique. However, if $p_{opt}(\vec{s})$ is non-unique, then the set of maximizers is convex.

To summarize, we have proved that asymptotically the optimal sequence is as

informative as any stochastic process with marginal distribution $p_{opt}(\vec{s})$. Thus, instead of finding the optimal sequence we can find $p_{opt}(\vec{s})$ by solving a convex optimization problem. To produce an optimal sequence we just sample $p_{opt}(\vec{s})$. Since $p_{opt}(\vec{s})$ depends on our current posterior, we should recompute $p_{opt}(\vec{s})$ after every trial.

3.2.4 Discussion

We conclude this section by briefly discussing some of the implications of Theorem 1. We motivated Theorem 1 by considering the case in which the neuron's response depends on past stimuli. In this case, we believe the greedy algorithm will be sup-optimal because it fails to take into account the influence of $\vec{x}_{t+1}$ on future trials. Consequently, the greedy algorithm does not generate good stimuli for learning the temporal structure of the receptive field. Theorem 1 shows that in general our greedy algorithm will produce a design which is sub-optimal when $t_k > 0$. Our results show that for the optimal sequence, the stimuli are correlated with past stimuli i.e in general $p_{opt}(\vec{s}) \neq \prod_{i=tk}^{0} p_{opt}(\vec{x}_{t-i})$. In contrast, we know that asymptotically our greedy algorithm produces a sequence which is equivalent to sequences in which each $\vec{x}_t$ is independent of past stimuli. This conclusion follows from a proof similar to that of Theorem 1 that the greedy algorithm produces a sequence which is on average as informative as a sequence produced by sampling some optimal distribution, $\vec{x}_t \sim p_{opt}(\vec{x})$ [114, 90] . As a result, we can conclude that in general our greedy algorithm is sub-optimal when the response depends on past stimuli.

Our motivation for considering the infinite horizon was to hopefully simplify the optimization in the non-greedy setting. Our results, Theorem 1, show that in the infinite horizon we need to optimize a concave function over the convex set of marginal distributions, $p(\vec{s})$. In general, if b is sufficiently large computing the optimal distribution $p(\vec{s})$ should be easier than computing the optimal sequence $\{\vec{x}_1, \ldots, \vec{x}_b\}$ because finding $p_{opt}(\vec{s})$ is a convex optimization problem. In the next section we show

how $p(\vec{s})$ may be computed in special cases and these methods may be contrasted with methods for computing the optimal sequence when b is reasonably small.

As we noted in the introduction, the equivalence between the optimal sequence and the optimal stochastic process is in an average sense. Theorem 1 and its derivation show that the equivalence is in terms of the average Fisher information per trial. The Fisher information is a measure of the information provided by $(\vec{s}_t, r_t)$ about the underlying conditional distribution $p(r_t|\vec{s}_t, \vec{\theta})$ [31]. Unlike our original objective function, the mutual information, the average Fisher information is independent of our prior information. In the limit $b \to \infty$ the objective function becomes independent of the prior because our prior information is negligible compared to the information contained in the observations. This result, however, implicitly assumes that we may collect information about all $\vec{\theta}$. If for example we cannot decrease our uncertainty about certain directions in $\vec{\theta}$ space, or we can decrease our uncertainty only so much, then our uncertainty in these directions is necessarily constrained by our prior. Mathematically, to ensure we can collect information in all directions we simply have to show that there exists $p(\vec{s})$ such that for all $\vec{\theta}$ the matrix $E_{p(\vec{s})}J_{exp}(\vec{s})$ is non-singular so that its determinant is well defined. We can easily show this is true for 1-d GLMs with Poisson likelihoods. Even though the Fisher information is independent of the prior, the prior still affects our choice of $p(\vec{s})$ because we must compute the expected Fisher information with respect to $\vec{\theta}$. Consequently, we will naturally favor designs which are informative, in terms of the Fisher information, under the models on which our uncertainty is concentrated. Thus, we will tend to pick designs which reduce our prior uncertainty.

Since the optimal sequence and the optimal stochastic process are equal only in an average sense, we need to consider conditions under which the equivalence will hold in practice. Naturally, the number of trials remaining must be large; otherwise $\hat{p}_b(\vec{s})$ will be a poor estimate of $p(\vec{s})$. Furthermore, the posterior must be changing

slowly. If the posterior is changing rapidly then we would expect the optimal which maximizes Eqn. 160 would change significantly on successive trials. As a result, $\hat{p}_b(\vec{s})$ would again be a poor estimate of $p(\vec{s})$. Intuitively, it seems reasonable to conjecture that in this case we might do better just using the greedy algorithm. If we expect a single trial to drastically reduce our uncertainty then arguably it makes no sense to construct a design which looks any further into the future then the next time step because it is unreasonable to expect that we would continue using that design after the next trial. This issue is similar to the exploration-exploitation trade-off in reinforcement learning [16, 148].

Finally, we have endeavored to make the results in this section as general as possible and not unnecessarily restrict our results to generalized linear models. The only assumption we have made about the conditional likelihood is that it has a finite dependence on the past, $p(r_t|\vec{x}_{-\infty}, \ldots, \vec{x}_t, \vec{\theta}) = p(r_t|\vec{x}_{t-t_k}, \ldots, \vec{x}_t)$. Consequently, our results do not hold for auto-regressive models such as GLMs with a dependence on past spike history. However, we conjecture that in practice if a neuron has a dependence on past spike history we may find a reasonably small t'_k such that the conditional distribution may be well approximated as

$$p(r_t|\vec{x}_{t-t_k}, \ldots, \vec{x}_t, r_{t-t_a}, \ldots, r_{t-1}) \approx p(r_t|\vec{x}_{t-t'_k}, \ldots, \vec{x}_t) \tag{192}$$

$$= E_{r_{t-t'_k} \ldots r_{t-1}} p(r_t|\vec{x}_{t-t_k}, \ldots, \vec{x}_t, r_{t-t_a}, \ldots, r_{t-1}) \times \prod_{i=t-t'_k}^{t-1} p(r_i|\vec{x}_{i-t_k}, \ldots, \vec{x}_i, r_{i-t_a}, \ldots, r_{i-1}) \tag{193}$$

$$\vec{x}_t = \hat{x} \ \forall t < t - t'_k \tag{194}$$

$$r_t = \hat{r} \ \forall t < t - t'_k, \tag{195}$$

Here t_a measures the number of past responses on which r_t depends. We approximate the conditional likelihood of r_t by computing the joint conditional likelihood on all responses $\{r_{t-t'_k}, \ldots, r_t\}$. This conditional response depends on stimuli and responses

that occurred before $t - t'_k$. However, we approximate all stimuli and responses before $t - t'_k$ using the point estimates $\hat{x}$ and r_r. By plugging in constant values we can truncate the chain and approximate the conditional likelihood using a finite number of stimuli. We then marginalize the conditional likelihood over all $\{r_{t-t'_k}, \ldots, r_{t-1}\}$. For $\hat{x}$ and $\hat{r}$ we choose some suitable value such as the average stimulus and the background firing rate. This approximation should be reasonable because we would expect the effect of past spikes to eventually die out. Thus, for suitably large $t'_k > t_a$ we might expect the above approximation to work quite well.

To derive the objective function, we also implicitly assumed that the log likelihood is concave as this assumption justifies the Gaussian approximation of the posterior. The Gaussian approximation of the posterior is necessary for the analytical approximation of the mutual information which was the starting point for the derivation of the objective function in the infinite horizon.

Finally, we note that the result in Theorem 1 is closely related to the idea of D-optimality in the experimental design literature [57, 25, 24, 27, 105, 114]. The key difference is we consider the case where each input $\vec{s}_t$ is constrained by past choices. Furthermore, most previous work focused on computing locally-optimally designs; i.e maximizing the average information per trial using a point-estimate of $\vec{\theta}$. In contrast we maximize the expectation with respect to our posterior on $\vec{\theta}$.

3.3 Finding the optimal process, $p(\vec{s})$

Using Theorem 1 to design experiments is quite difficult in practice. Evaluating the average information per trial, Eqn. 191, entails computing high-dimensional expectations with respect to the stimulus distribution, $p(\vec{s})$, and the posterior on $\vec{\theta}$, $p(\vec{\theta}|\vec{\mu}_t, \boldsymbol{C}_t)$. The difficulty of evaluating these expectations is one reason previous work has often focused on locally optimal designs [25, 105]. Since maximizing the average information with respect to the design, $p(\vec{s})$, is a convex optimization problem, the

maxima is well defined and can in theory be found numerically. However, for even moderately large $\dim(\vec{s})$ or t_k numerically solving for $p_{opt}(\vec{s})$ will be too slow for online use. A natural approach to this problem is to assume $p(\vec{s})$ has some parametric form. Our method is based on restricting $p(\vec{s})$ to the set of Gaussian processes. Gaussian processes make it easier to compute expectations with respect to $p(\vec{s})$ because the marginal and conditional distributions for these processes are Gaussian. Furthermore, Gaussian processes are easy to sample; this is an important consideration because to design experiments using $p_{opt}(\vec{s})$ we must be able to sample it. In Section 3.3.1 we evaluate the average information per trial, Eqn. 160, for Gaussian processes. In Section 3.3.2 we consider the special case of a GLM with Poisson likelihood and exponential link function. We consider this special case in detail because the Poisson model is particularly useful for modeling neurons. The use of an exponential link function with the Poisson model allows us to simplify several computations and is therefore worth considering in detail.

3.3.1 Finding the optimal Gaussian Process

A Gaussian process is a stochastic process for which the joint distribution on any of the variables is Gaussian [121]. Since $p(\vec{s})$ needs to be a t_k+1 order stationary process we need only consider stationary Gaussian processes. To specify the Gaussian process we need to determine its mean, $m(t)$, and covariance functions, $v(t_i, t_j)$. Since the process must be stationary these functions can depend only on the difference in time between the samples, $|t_i - t_j|$,

$$m(t) \triangleq E_{\vec{x}_t}\vec{x}_t = u = const \tag{196}$$

$$v(t_i, t_j) \triangleq E_{\vec{x}_{t_i},\vec{x}_{t_j}}\vec{x}_{t_i}\vec{x}_{t_j} - E_{\vec{x}_{t_i}}\vec{x}_{t_i}E_{\vec{x}_{t_j}}\vec{x}_{t_j} \tag{197}$$

$$= v(t_i - t_j) = v(t_j - t_i)^T. \tag{198}$$

Since our objective function, Eqn. 160, only depends on the marginal distribution on sequences of length $t_k + 1$ we can for simplicity only consider processes with

$$v(|t_j - t_i|) = 0 \quad \text{if} \quad |t_j - t_i| > t_k. \tag{199}$$

In this case the Gaussian process is determined by u which is a vector of length $\dim(\vec{x}_t)$ and $v(t_i - t_j)$ for $|t_i - t_j| = 0, 1, \ldots, t_k$. $v(t_i - t_j)$ is a $\dim(\vec{x}_t) \times \dim(\vec{x}_t)$ matrix. Since this process is stationary, all sequences of length l have the same marginal distribution. Consequently the process is necessarily a $t_k + 1$ order stationary process.

Gaussian processes have the property that the marginal distribution on any subsequence is Gaussian,

$$p(\vec{s}) = \mathcal{N}(\vec{s}; \mu_s, C_s) \tag{200}$$

$$\mu_s = \begin{bmatrix} u \\ \vdots \\ u \end{bmatrix} \quad C_s = \begin{bmatrix} v(0) & v(1) & \ldots & v(t_k) \\ v(-1) & v(0) & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \ldots & \ldots & v(0) \end{bmatrix} \tag{201}$$

$$v(i) = v(-i)^T. \tag{202}$$

Since $p(\vec{s})$ is Gaussian, and the expected Fisher information for the 1-d GLM is 1-dimensional, we can compute the inner expectation, $E_{\vec{s}} J_{exp}(\vec{s}, \vec{\theta})$, with relative ease

(see Appendix 3.6.3). The result is we can reduce our objective function to

$$E_{\vec{\theta}} \log |E_{p(\vec{s})} J_{exp}(\vec{s}^T\vec{\theta})\vec{s}\vec{s}^T|$$

$$= \dim(C_s) E_{\vec{\theta}} \log \varpi_1 + E_{\vec{\theta}} \log |I + V^T(\varpi_1 C_s)^{-1} U| + \log |C_s| \quad (203)$$

$$U = \varpi_3 \left[(\vec{\kappa} + \frac{\varpi_2}{\varpi_3}\vec{\delta}), (\frac{\varpi_1}{\varpi_3} - (\frac{\varpi_2}{\varpi_3})^2)\vec{\delta} \right] \quad (204)$$

$$V = \left[(\vec{\kappa} + \frac{\varpi_2}{\varpi_3}\vec{\eta}), \vec{\eta} \right] \quad (205)$$

$$\varpi_1 = E_{w_1} J_{exp}(w_1 ||\vec{\theta}||_2) \quad (206)$$

$$\varpi_2 = E_{w_1} J_{exp}(w_1 ||\vec{\theta}||_2) w_1 \quad (207)$$

$$\varpi_2 = E_{w_1} J_{exp}(w_1 ||\vec{\theta}||_2) w_1^2 \quad (208)$$

$$p(w) = \mathcal{N}(\mu_{\omega_1}, \sigma^2_{\omega_1}) \quad (209)$$

$$\mu_{\omega_1} = \frac{\vec{\theta}^T}{||\vec{\theta}||_2}\mu_s \qquad \sigma^2_{\omega_1} = \frac{\vec{\theta}^T}{||\vec{\theta}||_2} C_s \frac{\vec{\theta}}{||\vec{\theta}||_2} \quad (210)$$

$$\vec{\delta} = \mu_s - \frac{\boldsymbol{\gamma}}{\sigma^2_{\omega_1}}\mu_{\omega_1} - \frac{\boldsymbol{\gamma}}{\sqrt{\sigma^2_{\omega_1}}} \quad (211)$$

$$\vec{\eta} = \mu_s - \frac{\boldsymbol{\gamma}}{\sigma^2_{\omega_1}}\mu_{\omega_1} + \frac{\boldsymbol{\gamma}}{\sqrt{\sigma^2_{\omega_1}}} \quad (212)$$

$$\boldsymbol{\gamma} = C_s \frac{\vec{\theta}}{||\vec{\theta}||_2}. \quad (213)$$

Since $I + V^T(\varpi_1 C_s)^{-1} U$ is a 2-d matrix, we can compute its determinant analytically.

In certain cases, e.g the canonical Poisson, we may further simplify this expression. More generally we can potentially evaluate the expectation with respect to $\vec{\theta}$ numerically either by direct integration or using Monte-Carlo techniques. The key fact is that Eqn. 203 depends only on $(\mu_{\omega_1}, \sigma^2_{\omega_1}, ||\vec{\theta}||_2)$; i.e our objective function depends only on three scalars which are linear and quadratic functions of $\vec{\theta}$. Thus, once we compute $p(\mu_{\omega_1}, \sigma^2_{\omega_1}, ||\vec{\theta}||_2)$, we just have to perform a 3-dimensional integration to evaluate the expectation with respect to $\vec{\theta}$. Three dimensions is small enough that we can expect Monte Carlo Techniques to provide good approximations of the expectation.

Qualitatively, we can evaluate the utility of designing experiments using Eqn. 203

by considering what features of the design the different terms quantify. The first two terms depend on the expected Fisher information of the inputs sampled from the design. Naturally we want to maximize the Fisher information because the Fisher information quantifies how well we can infer the parameters from the observations. Since we take the expectation of these quantities with respect to $\vec{\theta}$, our objective function favors designs which are informative for all models $\vec{\theta}$ on which the probability mass of our posterior is concentrated. As a result maximizing Eqn. 203 should lead to better designs then methods based on a point approximation of $\vec{\theta}$ [25, 105]. In particular, the expectation over $\vec{\theta}$ means the optimal design depends on both the mean and covariance matrix of our posterior and thus by extension our prior knowledge as well.

Simply maximizing the Fisher information, however, can lead to poor designs. For non-linear designs the Fisher information is non-uniform with respect to $\vec{\theta}$ and $\vec{s}$. For the 1-d GLM, the Fisher information depends on the projection of the input on the parameters. This means the cost of reducing our uncertainty in the subspace parallel to $\vec{\theta}$ is not the same as the cost of reducing our uncertainty on the subspace orthogonal to $\vec{\theta}$. Here the cost refers to the number of observations needed to reduce our uncertainty by some fixed amount.

If we simply maximize the Fisher information we will only explore the region of $\vec{\theta}$ space where information is cheap. Early in an experiment, it makes sense to pick inputs which produce the largest reduction in our uncertainty using the fewest observations. However, as our experiment progresses we would like to consider regions of $\vec{\theta}$ space where collecting information is more expensive. The $\log |C_s|$ term ensures that we explore the entire model space because $\log |C_s|$ blows up if any of the eigenvalues of C_s become too small. This term, which depends only on the design, acts as a restoring force which tends to whiten our design. Consequently, the optimal design always assigns a non-zero probability to inputs in any direction of inputs space. This

feature of the optimal design ensures that we explore all regions of model space and do not get stuck obsessively exploring regions where information is cheap.

Intuitively, whitening our design is necessary to ensure our design is robust. If our model is misspecified, our prior knowledge is incorrect, or the neuron is adapting over time, then we might end up with a design which obsessively gathers information about a set of models which does not include the best model [5, 89]. Whitening our design tends to make our design more robust to such misspecification, reducing the amount of bias introduced by model misspecification [92].

3.3.2 The optimal Gaussian Process for the canonical Poisson model.

In this section, we consider a special case of the GLM; the Poisson model with an exponential nonlinearity. For the Canonical Poisson the Fisher Information is the exponential function, $J_{exp}(\vec{s}^T\vec{\theta}) = \exp(\vec{s}^T\vec{\theta})$. Since the Fisher Information is independent of r_t, and the distributions on $\vec{s}$ and $\vec{\theta}$ are Gaussian, we can compute some of the expectations in our objective function, Eqn. 160, analytically. In this section, we use these properties to simplify the objective function. While we cannot derive a completely analytical expression for the average information per trial, we can derive a completely analytical lower bound. In the following sections we focus on finding the Gaussian process which maximizes this lower bound. Unfortunately, the constraint that the process be $t_k + 1$ order stationary, i.e. that C_s must be a block-Toeplitz matrix, makes optimizing the lower bound rather complicated. Therefore, in Section 3.3.2.1 we find the Gaussian process which maximizes the lower bound without enforcing the constraint that C_s is a block-Toeplitz matrix. We then show how we can modify this Gaussian process to construct a $t_k + 1$ order stationary process. In some sense this approach works by finding the best Gaussian distribution for $p(\vec{s})$ and then finding the "closest" $t_k + 1$ order stationary Gaussian process to $p(\vec{s})$. While there is no guarantee that the resulting design will be optimal, it leads to a very tractable

1-dimensional optimization and works well in simulations. In Section 3.3.2.2 we briefly discuss numerically optimizing our lower bound subject to the constraint that C_s is block-Toeplitz. Since the complexity of the numerical optimization grows with the dimensionality of $\vec{\theta}$, numerical methods are largely impractical for designing neurophysiology experiments.

For the Canonical Poisson, we can easily compute $E_{\vec{s}} \exp(\vec{s}^T\vec{\theta})\vec{s}\vec{s}$ because it is just a weighted Gaussian,

$$E_{\vec{\theta}} \log \left| E_{\vec{s}} \exp(\vec{s}^T\vec{\theta})\vec{s}\vec{s} \right| = E_{\vec{\theta}} \log \left| \exp(\vec{\theta}^T\mu_s + \frac{1}{2}\vec{\theta}^T C_s\vec{\theta}) \left((\mu_s + C_s\vec{\theta})(\mu_s + C_s\vec{\theta})^T + C_s \right) \right| \tag{214}$$

$$= E_{\vec{\theta}} \left(d_s\vec{\theta}^T\mu_s + \frac{d_s}{2}\vec{\theta}^T C_s\vec{\theta} + \log|C_s| + \log\left(1 + (\mu_s + C_s\vec{\theta})^T C_s^{-1}(\mu_s + C_s\vec{\theta})\right)\right) \tag{215}$$

$$= d_s\vec{\mu}_t^T\mu_s + \frac{d_s}{2}Tr(C_s\vec{\mu}_t\vec{\mu}_t^T + C_s\boldsymbol{C}_t) + \log|C_s|$$
$$+ E_{\vec{\theta}} \log(1 + (\mu_s + C_s\vec{\theta})^T C_s^{-1}(\mu_s + C_s\vec{\theta})) \tag{216}$$

Computing the expected value of the log term is difficult. The expected value of the log term, however, is necessarily positive because

$$(\mu_s + C_s\vec{\theta})^T C_s^{-1}(\mu_s + C_s\vec{\theta}) \geq 0 \Rightarrow \log(1 + (\mu_s + C_s\vec{\theta})^T C_s^{-1}(\mu_s + C_s\vec{\theta})) \geq 0. \tag{217}$$

As a result by dropping the log term we end up with the lower bound

$$E_{\vec{\theta}} \log|E_{\vec{s}} \exp(\vec{s}^T\vec{\theta})\vec{s}\vec{s}| \geq d_s\vec{\mu}_t^T\mu_s + \frac{d_s}{2}Tr(C_s\vec{\mu}_t\vec{\mu}_t^T + C_s\boldsymbol{C}_t) + \log|C_s|. \tag{218}$$

Qualitatively this lower bound leads to a reasonable objective function for optimizing the design. Our goal is to pick inputs which maximize the amount of new information provided by the experiment. The utility of an input is thus a function of 1) the informativeness of the experiment as measured by the Fisher information, which is independent of what we already know, and 2) our posterior which quantifies what we already know. As noted in the previous section the effect of $\log|C_s|$ is to whiten our

design. In contrast, $Tr(C_s\vec{\mu}_t\vec{\mu}_t^T + C_s\boldsymbol{C}_t)$ is directly related to our prior uncertainty and the Fisher information. For the canonical Poisson, the Fisher information is $\exp(\vec{s}_t^T\vec{\theta})$. Thus, to increase the Fisher information of the inputs we want to maximize the projection of the inputs on $\vec{\theta}$. Clearly, maximizing $Tr(C_s\vec{\mu}_t\vec{\mu}_t^T) = \vec{\mu}_t^T C_s\vec{\mu}_t$ entails placing as much stimulus power as we can in the direction of $\vec{\mu}_t$ which is our best estimate of $\vec{\theta}$ at time t. As a result, the first term quantifies the extent to which the design picks inputs with large Fisher information. In contrast the $Tr(C_s\boldsymbol{C}_t)$ tries to force us to explore areas of uncertainty. Ignoring the Toeplitz constraint on C_s, maximizing $Tr(C_s\boldsymbol{C}_t)$ subject to a constraint on $Tr(C_s)$ entails putting all stimulus power along the largest eigenvector of $\boldsymbol{C}_t$. Thus, maximizing $Tr(C_s\boldsymbol{C}_t)$ favors designs which would explore regions of $\vec{\theta}$ space where our uncertainty is high.

For the canonical Poisson, the informativeness of an experiment increases with the magnitude of the stimuli. Clearly by increasing the magnitude of μ_s or the variance of C_s we can make the linear term, the terms outside the log, arbitrarily large. Since the linear function grows faster than the logarithm, this means we can make our objective function arbitrarily large. Therefore, we must constrain the stimuli in order to get a well defined optimization problem. A reasonable constraint is the average power of the stimuli,

$$E_{\vec{x}}\vec{x}^T\vec{x} \leq m. \tag{219}$$

For the Gaussian process

$$E_{\vec{x}}x^T x = Tr(E_{\vec{x}}xx^T) \tag{220}$$

$$= Tr(C_{xx} + \mu_x\mu_x^T) \tag{221}$$

$$= Tr(v(0) + uu^T). \tag{222}$$

Clearly the set of u and $\{v(0), \ldots, v(t_k)\}$ satisfying these constraints is convex. Unfortunately these constraints are nonlinear which in general complicates the optimization problem.

3.3.2.1 The optimized Gaussian Process under relaxed stationarity constraints

In this section we show that if we drop the constraint that C_s is block-Toeplitz, we can maximize the lower bound for Eqn. 216 using a simple 1-dimensional search. Since $p(\vec{s})$ does not define a $t_k + 1$ order stationary process we cannot generate a sequence of stimuli with stationary marginal distribution $p(\vec{s})$. However, we can modify this Gaussian Process to compute a Gaussian Process which is $t_k + 1$ order stationary. We can think of this modification as projecting $p(\vec{s})$ into the space $\mathcal{P}_{\vec{f}}$. While the solution is no longer guaranteed to be optimal it gives a result which can easily be implemented and therefore of practical use. Furthermore, simulation results indicate this design can still outperform an i.i.d. design, Section 3.4.

If we drop the constraint that C_s is block-Toeplitz then we just need to find

$$\arg\max_{\mu_s, C_s} E_{\vec{\theta}} \left(d_s \vec{\theta}^T \mu_s + \frac{d_s}{2} \vec{\theta}^T C_s \vec{\theta} + \log |C_s| \right) = \arg\max_{\mu_s, C_s} d_s \vec{\mu}_t^T \mu_s + \frac{d_s}{2} Tr(C_s R) + \log |C_s| \tag{223}$$

$$R = \vec{\mu}_t \vec{\mu}_t^T + \boldsymbol{C}_t \tag{224}$$

over all (μ_s, C_s) subject to the constraints

$$\vec{s}^T C_s \vec{s} > 0 \quad \forall \vec{s} \neq 0 \tag{225}$$

$$Tr(C_s) < m - ||\mu_s||^2 \tag{226}$$

Clearly the optimal μ_s will be parallel to $\vec{\mu}_t$. Therefore, only the magnitude of the optimal μ_s is unknown. We can therefore rewrite the objective function as

$$\arg\max_{||\mu_s||} \left(d_s ||\vec{\mu}_t|| ||\mu_s|| + \arg\max_{C_s} \frac{d_s}{2} Tr(C_s R) + \log |C_s| \right) \tag{227}$$

The inner problem depends on $||\mu_s||$ because of the power constraint. The inner

problem is

$$\arg\max_{C_s} \frac{d_s}{2} Tr(C_s R) + \log|C_s| \tag{228}$$

$$s.t \quad \vec{s}^T C_s \vec{s} > 0 \quad \forall \vec{s} \neq 0 \tag{229}$$

$$Tr(C_s) < m - ||\mu_s||^2 \tag{230}$$

Since the determinant of C_s is independent of its eigenvectors, the eigenvectors of C_s are completely determined by $Tr(C_s R)$. We can easily show, see Appendix 3.6.4, that $Tr(C_s R)$, is optimized when the eigenvectors of C_s equal the eigenvectors of R. As a result, finding the optimal C_s is a constrained eigenvalue optimization problem which we can solve by introducing a Lagrange multiplier λ. The result is,

$$C_s = \frac{2}{d_s} O \begin{bmatrix} \frac{1}{\lambda - r_1} & 0 & 0 \\ 0 & \ddots & \\ 0 & & \frac{1}{\lambda - r_{d_s}} \end{bmatrix} O' \tag{231}$$

where O are the eigenvectors of R, r_i are the eigenvalues of R, $d_s = \dim(\vec{s})$, and we find λ by numerically solving the 1-d equation

$$\frac{2}{d_s} \sum_i \frac{1}{\lambda - r_i} = m - ||\mu_s||^2. \tag{232}$$

The details of our solution are in Appendix 3.6.4.

In principle to find the optimal $(||\mu_s||, C_s)$, we need to do a search over $||\mu_s||$ and compute the optimal C_s for each $||\mu_s||_2$ by solving for λ given $||\mu_s||$. However, we can significantly reduce the amount of computation required by doing a search over λ for $\lambda > \max r_i$ as opposed to a search over $||\mu_s||$. For each value of λ we can compute the optimal $(||\mu_s||, C_s)$. Thus, a single 1-d search over λ is guaranteed to find the optimal (μ_s, C_s).

Since the $p_{opt}(\vec{s})$ which maximizes the lower bound does not satisfy the conditions for a t_k+1 order process we cannot sample it using the procedure presented in Section 3.2.2. However, we can generate a valid sequence of stimuli by drawing samples

of sequences of length $t_k + 1$ by sampling $\vec{s} = \{\vec{x}_{t-t_k}, \ldots, \vec{x}_t\}$ from the marginal distribution $p_{opt}(\vec{s})$; that is after every $t_k + 1$ trials, we pick a batch of $t_k + 1$ stimuli to be presented on the next $t_k + 1$ trials. The empirical distribution for this sequence, $\hat{p}_b(\vec{s})$, does not converge to $p_{opt}(\vec{s})$. Hence, we cannot use $p_{opt}(\vec{s})$ to compute the informativeness of the sequence generated by sampling $p_{opt}(\vec{s})$.

The results in Section 3.2.2, however, still apply. Thus, there exists a t_k+1-order stationary process which is equivalent with respect to the average information to the sequence produced by sampling $p_{opt}(\vec{s})$ as described above. To find this equivalent process we just need to find the limiting distribution of the empirical marginal distribution, $\hat{p}_b(\vec{s})$. Using the sampling procedure defined above each batch of $t_k + 1$ stimuli is an i.i.d. sample from $p_{opt}(\vec{s})$. Hence, we can easily compute the marginal distribution on any subsequence of length $t_k + 1$,

$$p(\vec{x}_{t-t_k}, \ldots, \vec{x}_t) = \int_{\vec{x}_i, \vec{x}_{i+1}, \ldots, \vec{x}_{t-t_k}} \int_{\vec{x}_{t+1}, \vec{x}_{t+2}, \ldots, \vec{x}_{j+t_k}} p(\vec{x}_i, \ldots, \vec{x}_{i+t_k}) p(\vec{x}_j, \ldots, \vec{x}_{j+t_k}) \tag{233}$$

$$p(\vec{x}_i, \ldots, \vec{x}_{i+t_k}) = p_{opt}(\vec{s}) \tag{234}$$

$$p(\vec{x}_j, \ldots, \vec{x}_{j+t_k}) = p_{opt}(\vec{s}) \tag{235}$$

$$i = \lfloor \frac{t-1}{t_k+1} \rfloor + 1 \tag{236}$$

$$j = i + t_k + 1. \tag{237}$$

Here $\lfloor \rfloor$ denotes the floor function and $p_{opt}(\vec{s})$ is the Gaussian distribution computed by optimizing the lower bound for Eqn. 160. Since $\{\vec{x}_i, \ldots, \vec{x}_{i+t_k}\}$ and $\{\vec{x}_j, \ldots, \vec{x}_{j+t_k}\}$ are i.i.d. samples drawn from $p_{\vec{s}}(\vec{s})$, the empirical distribution converges to a distribution which is a uniform mixture of the marginals $p(\vec{x}_{t-t_k+\Delta}, \ldots, \vec{x}_{t+\Delta})$ for $\Delta =$

$0, 1, \ldots, t_k$,

$$\lim_{b\to\infty} \hat{p}_b(b) = p'(\vec{s}) \tag{238}$$

$$= \frac{1}{t_k+1} \sum_{\Delta=0}^{t_k} p(\vec{x}_{t-t_k+\Delta}, \ldots, \vec{x}_{t+\Delta}) \tag{239}$$

where $p(\vec{x}_{t-t_k+\Delta}, \ldots, \vec{x}_{t+\Delta})$ is computed using Eqn. 233. $p'(\vec{s})$ is necessarily a t_k+1 order stationary process. This result holds in general for all $p(\vec{s})$ and not just $p(\vec{s})$ which are Gaussian; that is for any $p(\vec{s})$ we can always construct a t_k+1 order stationary process, $p'(\vec{s})$, by using a mixture of $p(\vec{s})$.

3.3.2.2 Optimal t_k+1 stationary Gaussian Process for the canonical Poisson

In this section, we return to the problem of finding the optimal Gaussian process subject to the constraints that $p(\vec{s})$ is a t_k+1 order stationary process. Consequently, we want to maximize the same objective function as in the previous section except with the added constraint that C_s is a block Toeplitz matrix,

$$(\mu_s, C_s) = \arg \max_{u,v(0),\ldots,v(t_k)} d_s \vec{\mu}_t^T \mu_s + \frac{d_s}{2} Tr(C_s \vec{\mu}_t \vec{\mu}_t^T + C_s \boldsymbol{C}_t) + \log|C_s| \tag{240}$$

$$+ E_{\vec{\theta}} \log(1 + (\mu_s + C_s\vec{\theta})^T C_s^{-1} (\mu_s + C_s\vec{\theta})) \tag{241}$$

$$\mu_s = \begin{bmatrix} u \\ \vdots \\ u \end{bmatrix} \quad C_s = \begin{bmatrix} v(0) & v(1) & \ldots & v(t_k) \\ v(-1) & v(0) & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \ldots & \ldots & v(0) \end{bmatrix} \tag{242}$$

$$\sum_i diag(v(0))_i + u^T u \leq m \tag{243}$$

$$\vec{s}^T C_s \vec{s} > 0 \quad \forall \vec{s} \neq 0. \tag{244}$$

We could attempt to solve this problem directly using numerical methods; i.e we can compute the objective function and its derivatives for any μ_s and C_s and we can approximate the expectation over $\vec{\theta}$ and its derivatives using Monte-Carlo techniques.

However, since $t_k \sim O(10)$ and $\dim(\vec{x}(t)) \sim O(10 - 100)$, the dimensionality will simply be too large to easily optimize the design let alone to continually re-optimize the design during an actual experiment.

As in the previous section we can avoid numerical integration by ignoring the expectation of the log-term and just maximizing the lower bound. Furthermore, we can still divide our objective function into an inner and outer problem,

$$\arg\max_{||\mu_s||} \left(d_s \vec{\mu}_t^T \mu_s + \arg\max_{C_s} \frac{d_s}{2} Tr(C_s R) + \log |C_s| \right). \tag{245}$$

Since μ_s depends only on u,

$$\vec{\mu}_t^T \mu_s = u^T M(\vec{\mu}_t)\vec{1} \tag{246}$$

$$M(\vec{\mu}_t) = [\vec{\mu}_t^{1:\dim(\vec{x}_t)}\ \vec{\mu}_t^{\dim(\vec{x}_t)+1:2\dim(\vec{x}_t)}, \ldots, \vec{\mu}_t^{t_k \dim(\vec{x}_t)+1:(t_k+1)\dim(\vec{x}(t))}]. \tag{247}$$

$M(\vec{\mu}_t)$ is just a matrix whose columns correspond to $\dim(\vec{x}_t)$ consecutive elements of $\vec{\mu}_t$. $\vec{1}$ is just a vector of ones. Clearly to maximize our lower bound u should be parallel to $M(\vec{\mu}_t)\vec{1}$. Therefore we just need to find the optimal value of $||u||$. Unfortunately we cannot use the same approach as the previous section to solve the inner problem. Our solution in the previous section used the fact that there was no constraint on the eigenvectors of C_s. Consequently, since $\log |C_s|$ is independent of its eigenvectors, we could just pick the eigenvectors of C_s to equal those of R. In this case, the Toeplitz constraint on C_s restricts the eigenvectors of C_s. In principle, we can numerically optimize the inner problem with respect to the coefficients of $\vec{v}$, but in general this leaves too many degrees of freedom to be a practical solution for sequential, optimal, experimental design.

3.3.3 Bias and Spike history terms

Our analysis so far has assumed that the input, $\vec{s}$, consisted only of the stimulus. In practice, $\vec{s}$ may have fixed terms; for example a bias term or terms corresponding to spike history. We can handle this using our existing methods by considering the

information provided only about the stimulus coefficients and ignoring the information about the bias and spike history components. This approach is reasonable because the bias term can typically be estimated very well simply by observing the background firing rate in the absence of any stimuli. Ignoring the information provided about the spike history terms is reasonable because it seems unlikely that we will be able to control the neuron's response well enough to generate spike-histories to directly probe the neuron's dependence on its past responses. However, decreasing our uncertainty about the stimulus coefficients will tend to also reduce our uncertainty about the spike history coefficients because we can do a better job estimating how much of the response is due to the spike history.

Suppose we let $\vec{s}^T = \{\vec{s}_x^T, \vec{s}_f^T\}$ where $\vec{s}_x$ are the terms corresponding to $\{\vec{x}_{t-t_k}, \ldots, \vec{x}_t\}$ and $\vec{s}_f$ are the non-stimulus terms, e.g. the bias and spike history terms. Following the derivation in Section 3.2, our objective function in this case is

$$\begin{aligned}\arg\max_{p(\vec{s}_x)} E_{\vec{\theta}} \log \left| E_{p(\vec{s})} E_{\vec{s}_f|\vec{s}_x} J_{exp}(\vec{s}) \vec{s}\vec{s}^T \right| \\ = \arg\max_{p(\vec{s}_x)} E_{\vec{\theta}} \log \left| E_{p(\vec{s}_x)} E_{\vec{s}_f|\vec{s}_x} J_{exp}(\vec{s}) \begin{bmatrix} \vec{s}_x\vec{s}_x^T & \vec{s}_x\vec{s}_f^T \\ \vec{s}_f\vec{s}_x^T & \vec{s}_f\vec{s}_f^T \end{bmatrix} \right|. \end{aligned} \tag{248}$$

If we focus on just maximizing the information about the stimulus coefficients then our objective function is

$$\arg\max_{p(\vec{s}_x)} E_{\vec{\theta}} \log \left| E_{p(\vec{s}_x)} E_{\vec{s}_f|\vec{s}_x} J_{exp}(\vec{s}) \vec{s}_x\vec{s}_x^T \right|. \tag{249}$$

This result is equivalent to assuming we want to minimize the entropy of the marginal distribution of our posterior on the stimulus coefficients of $\vec{\theta}$. We can show, following a derivation like that in Section 3.2, that minimizing the entropy of the marginal distribution on the stimulus coefficients is equivalent in the limit $b \to \infty$ to Eqn. 249. The only complication in Eqn. 249 is computing the expected value of the Fisher information with respect to $p(\vec{s}_f|\vec{s}_x)$. One way to handle this is simply to use a point

estimate of $\vec{s}_f$. For the canonical Poisson if $\vec{s}_f$ contains fixed terms we can simply ignore these terms because they simply scale the Fisher information matrix,

$$J_{exp}(\vec{s}^T\vec{\theta}) = \exp(\vec{s}_x^T\vec{\theta}_x)\exp(\vec{s}_f^T\vec{\theta}_f). \tag{250}$$

The term $\exp(\vec{s}_f^T\vec{\theta}_f)$ just adds a constant, with respect to the design $p(\vec{s}_x)$, to our objective function and thus has no impact on the optimization.

3.4 Results

We tested our methods using simulated experiments which mimicked real experiments investigating the response properties of auditory neurons in song bird [168]. In these simulations, we generated synthetic responses by simulating a neuron using a GLM. The parameters of the GLM were the parameters of a GLM fitted to real data taken from experiments with song birds. The data was provided to us by David Schneider and Dr. Sarah Woolley and is described in detail in Chapter 4. $\vec{\theta}$ in this case consisted of the STRF fitted to the bird-song data, as well as three of the spike-history coefficients and the bias term. Using this synthetic neuron, we ran simulated experiments in which stimuli were chosen either by sampling an optimized Gaussian process using the methods presented in Section 3.3.2.1 or by sampling a white Gaussian process. Both processes were subject to the same average power constraint.

In Figure 19 we compare the MAP estimate of the parameters using the two designs as a function of the trial. The corresponding mean squared error of the MAPs is shown in Figure 20. The results clearly show that the MAP converges more rapidly to the true value of the STRF using the optimized design. For the spike-history and bias terms the results are more mixed. Even though we ignore the spike-history and bias terms when optimizing the design, the optimized Gaussian process still produces a small albeit somewhat transient improvement in the estimated spike-history and bias coefficients compared to the design using white noise. In Figure 21 we plot the observed firing rate as a function of time for the synthetic neuron. This plot shows

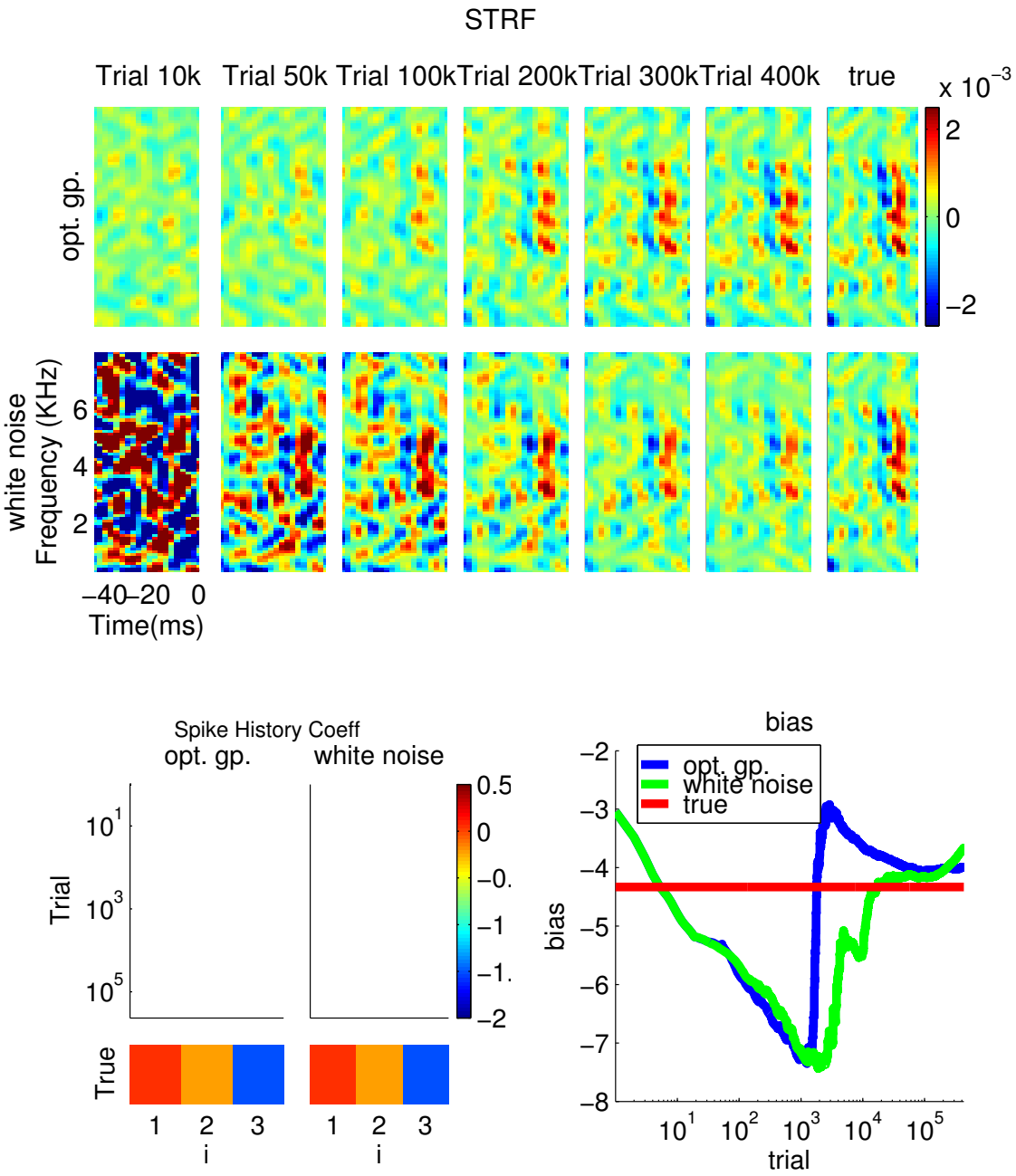


Figure 19: A) Plots of the MAP estimates of the STRF estimated on several trials using a white vs. optimized Gaussian process. B) The estimated spike history coefficients after each trial. Each row shows the spike history coefficients on a different trial. C) The estimated value of the bias term after each trial.

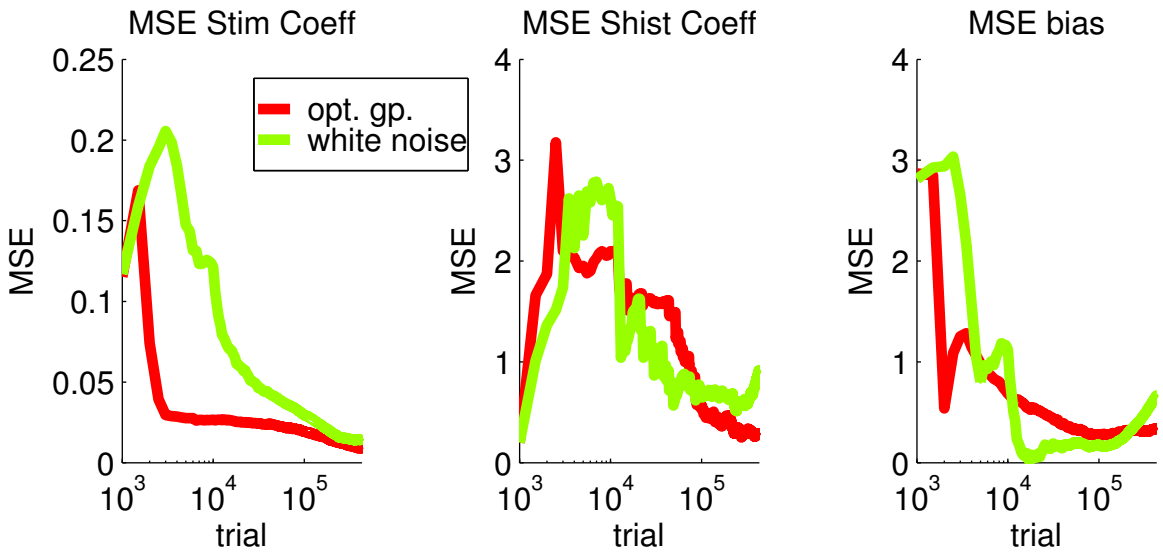


Figure 20: Plots of the mean squared error between each component of the MAP and the corresponding true value of $\vec{\theta}$. Left panel, the MSE between the estimated stimulus coefficients, the STRF, and their true value. Middle panel, the MSE for the spike-history coefficients. Right panel, the MSE for the bias term.

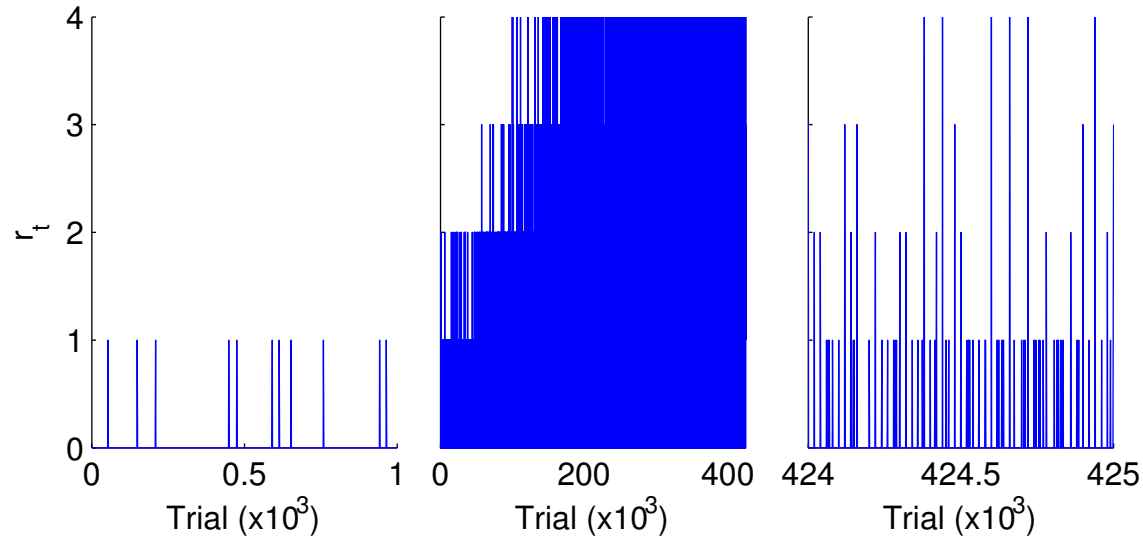


Figure 21: A plot of the number of spikes observed for the design using the optimized Gaussian process. Left panel, the spikes on the first 1000 trials. Middle panel, the number of spikes on each trial. Right plot, the number of spikes on the last 1000 trials. The plots clearly show that the optimized design ends up picking stimuli which drive the neuron to fire more often.

that the optimized design ends up picking inputs which drive the neuron to fire at a higher rate because for the canonical Poisson the Fisher information increases with the firing rate. In real experiments we might expect the neuron to adapt so that it would no longer fire if we keep picking the same input [141]. One way to handle this is by incorporating a simple model of adaptation as discussed in Chapter 2. By modeling adaptation, we can take the effects of adaptation into account when computing the expected information gain for each input.

The main conclusion of Figure 19 and Figure 20 is that even though we optimized the Gaussian process by ignoring stationarity constraints, the resulting design still decreased the error faster than a white noise design. This result is important because the methods described in Section 3.3.2.1 can be implemented in a real experiment with a minimal amount of effort.

3.5 Discussion

In this chapter we have shown how the problem of non-greedy optimization of the stimulus can be turned into a tractable problem by considering the infinite horizon. For the Canonical Poisson we have shown how we can compute a better design then white noise by relaxing the stationarity constraints, Section 3.3.2.1, or by numerically optimizing a concave objective function, Section 3.3.2.2.

Our solution in the case of the canonical Poisson used the lower bound for the objective function presented in Section 3.3.2. Naturally we would like to get some sense of how good this lower bound is. One way we can potentially address this issue is by using Jensen's inequality to establish an upper bound for the error due to

ignoring the log term,

$$E_{\vec{\theta}} \log(1 + (\mu_s + C_s\vec{\theta})^T C_s^{-1}(\mu_s + C_s\vec{\theta})) \leq \log(1 + E_{\vec{\theta}}(\mu_s + C_s\vec{\theta})^T C_s^{-1}(\mu_s + C_s\vec{\theta})) \tag{251}$$

$$= \log\left(1 + \mu_s C_s^{-1}\mu_s + 2\vec{\mu}_t^T\vec{\mu}_s + Tr(C_s(\boldsymbol{C}_t - \vec{\mu}_t\vec{\mu}_t^T))\right). \tag{252}$$

Since we can easily evaluate this bound for any (μ_s, C_s), we can easily do a numerical investigation to get some sense of how tight our lower bound is.

In Section 3.3.2.1 we showed how an optimal Gaussian process may be computed by relaxing the stationarity constraints. An obvious question is how the resulting design will compare to the optimized Gaussian process in Section 3.3.2.2 when we enforce the stationarity constraints during the optimization. In Section 3.3.2.1 we showed that while the optimal Gaussian process computed is not $t_k + 1$ order stationary it is equivalent to a mixture of Gaussians which is $t_k + 1$ order stationary. Thus, we can view the results of Section 3.3.2.1 as a heuristic for computing the optimal design when we restrict $p(\vec{s})$ to being a mixture of Gaussians. Since the mixture of Gaussians is a much more flexible model we would hope that the solution in Section 3.3.2.1 would do better than the solution in Section 3.3.2.2.

In this chapter we have focused on approximate methods which ignore stationarity constraints because the resulting, semi-analytical solution is much better suited for designing actual experiments. Real experiments involve high-dimensional stimuli which numerical optimization methods will unlikely be able to handle, particularly in a real-time setting. Consequently, an open question is whether numerical methods like those discussed in Section 3.3.2.2 are really worth pursuing. If the dimensionality is small enough for numerical methods to be feasible we might reasonably expect a non-optimized design to work nearly as well as an optimized design. Furthermore, the methods in this chapter implicitly assume that a large number of trials is required to estimate the model because if b is small the empirical marginal distribution provides

a poor estimate of $p(\vec{s})$. If the dimensionality of the model is small enough to make numerical optimization tractable then there is a good chance only a small number of observations will be needed to fit the model. In this case, the methods in this chapter will not work well because they are based on the assumption that b is large.

Despite the limitations of the methods presented in this chapter, they nonetheless address a major limitation of our greedy methods. In particular, the methods in this chapter create stimuli with complex temporal structure based on the expected temporal features of the neuron's receptive field. Consequently, we think the methods presented in this chapter will be particularly valuable for investigating neurons with complex spatio-temporal receptive fields.

3.6 Appendix

3.6.1 Why we do not need to know $\hat{p}_b(r|\vec{s})$ to compute the average information per trial as $b \to \infty$

In Section 3.2.1 we showed that the average information per trial only depends on the marginal distribution of subsequences of length $t_k + 1$. To establish this result, we made the claim that as $b \to \infty$ we can compute the average information per trial by substituting $p(r|\vec{s}, \vec{\theta})$ for $\hat{p}_b(r|\vec{s})$ because the average information per trial will be the same whether we use the true conditional distribution, $p(r|\vec{s}, \vec{\theta})$, or the empirical distribution for some sequence $\hat{p}_b(r|\vec{s})$. In this section, we provide a rigorous argument to justify this claim.

To prove this, we first show that in the limit $b \to \infty$, $p(\hat{p}_b(r|\vec{s})|\hat{p}_b(\vec{s}), \vec{\theta})$ only has support on a single distribution,

$$\lim_{b\to\infty} p(\hat{p}_b(r|\vec{s})|\hat{p}_b(\vec{s}), \vec{\theta}) = \delta(\tilde{p}(r|\vec{s})) \tag{253}$$

$$\tilde{p}(r|\vec{s}) = \begin{cases} p(r|\vec{s} = a, \vec{\theta}) & \forall a \text{ s.t } \lim_{b\to\infty} \hat{p}_b(a) > 0 \\ \hat{p}_b(r|\vec{s} = a) & \forall a \text{ s.t } \lim_{b\to\infty} \hat{p}_b(a) = 0 \end{cases} \tag{254}$$

(Note by definition $\tilde{p}$ depends on $\hat{p}_b(\vec{s})$). We can prove this result by first considering

a such that $\lim_{b\to\infty} p(\vec{s} = a) = 0$ and then a such that $\lim_{b\to\infty} p(\vec{s} = a) > 0$. If $\lim_{b\to\infty} p(\vec{s} = a) = 0$ then there exists t_a such that for all $t > t_a$ $\vec{s}_t \neq a$. Thus

$$\lim_{b\to\infty} \hat{p}_b(r = b|\vec{s} = a) = \frac{\sum_{i=1}^{t_a} \delta(\vec{s}_i = a, r_i = b)}{\sum_{i=1}^{t_a} \delta(\vec{s}_i = a)} \tag{255}$$

$$= const \tag{256}$$

Consequently $\hat{p}_b(r|\vec{s} = a)$ converges to some unknown distribution for all a s.t $\lim_{b\to\infty} \hat{p}_b(\vec{s} = a) = 0$.

For all a such that $\lim_{b\to\infty} \hat{p}_b(\vec{s} = a) > 0$, $\hat{p}_b(r|\vec{s} = a)$ is just a uniform distribution on the pairs $(\vec{s} = a, r = b)$. The pairs $(\vec{s} = a, r = b)$ are a set of i.i.d. samples drawn from the distribution $p(r|\vec{s} = a, \vec{\theta})$. Thus the empirical conditional distribution converges to the true conditional distribution for these inputs [157],

$$\lim_{b\to\infty} \hat{p}_b(r|\vec{s} = a) = p(r|\vec{s} = a, \vec{\theta}). \tag{257}$$

Using this result,

$$\lim_{b\to\infty} E_{\vec{\theta}} E_{p(\hat{p}_b(r|\vec{s})|\hat{p}_b(\vec{s}),\vec{\theta})} F\left(\hat{p}_b(r|\vec{s}), \hat{p}_b(\vec{s})\right) = \lim_{b\to\infty} E_{\vec{\theta}} E_{\delta(\tilde{p}(r|\vec{s}))} F\left(\hat{p}_b(r|\vec{s}), \hat{p}_b(\vec{s})\right) \tag{258}$$

$$= E_{\vec{\theta}} \lim_{b\to\infty} F\left(\tilde{p}(r|\vec{s}), \hat{p}_b(\vec{s})\right) \tag{259}$$

$$= E_{\vec{\theta}} F\left(\lim_{b\to\infty} \left(\tilde{p}(r|\vec{s}), \hat{p}_b(\vec{s})\right)\right) \tag{260}$$

The last equality is true because the log-determinant is a continuous function (provided its arguments are non-singular a complication which we ignore for now.). For any continuous function, the limit of a function evaluated on a sequence equals the function evaluated on the limit of the sequence [7].

We can further simplify our objective function by showing that we may substitute

$p(r|\vec{s},\vec{\theta})$ for $\tilde{p}(r|\vec{s})$ in the limit $b \to \infty$.

$$E_{\vec{\theta}} F\left(\lim_{b\to\infty}(\tilde{p}(r|\vec{s}),\hat{p}_b(\vec{s}))\right) = E_{\vec{\theta}} \log\left|\lim_{b\to\infty} E_{\hat{p}_b(\vec{s})} E_{\tilde{p}(r|\vec{s})} J_{obs}(\vec{s},r)\right| \tag{261}$$

$$= E_{\vec{\theta}} \log\Big|\lim_{b\to\infty}\int_{a\in A_1} \hat{p}_b(\vec{s}) p(r|\vec{s},\vec{\theta}) J_{obs}(\vec{s},r)$$

$$+ \lim_{b\to\infty}\int_{a\in A_2} \hat{p}_b(\vec{s})\hat{p}_b(r|\vec{s}) J_{obs}(\vec{s},r)\Big| \tag{262}$$

$$A_1 = \{a : \lim_{b\to\infty} \hat{p}_b(a) > 0\} \tag{263}$$

$$A_2 = \{a : \lim_{b\to\infty} \hat{p}_b(a) = 0\}. \tag{264}$$

Clearly it follows from the definition of A_2 that

$$\lim_{b\to\infty}\int_{a\in A_2} \hat{p}_b(\vec{s})\hat{p}_b(r|\vec{s}) J_{obs}(\vec{s},r) = 0 \tag{265}$$

for all $\hat{p}_b(r|\vec{s})$ because $\hat{p}_b(r|\vec{s})$ is bounded between zero and one and $\hat{p}_b(\vec{s}) \to 0$ for $a \in A_2$. Thus for convenience we may assume $\hat{p}_b(r|\vec{s}) = p(r|\vec{s},\vec{\theta})$ for $a \in A_2$. We can therefore rewrite our objective function as

$$E_{\vec{\theta}} \log|\lim_{b\to\infty} E_{\hat{p}_b(\vec{s})} E_{\tilde{p}(r|\vec{s},\vec{\theta})} J_{obs}(\vec{s},r)| = E_{\vec{\theta}} \log|\lim_{b\to\infty} E_{\hat{p}_b(\vec{s})} E_{p(r|\vec{s},\vec{\theta})} J_{obs}(\vec{s},r)| \tag{266}$$

3.6.2 Sufficient and necessary conditions for a $t_k + 1$ order process.

Here we present the proof that $p(\vec{s})$ defines a valid $t_k + 1$ order stationary process if and only if

$$p(s^{-i:0}) = p\left(s^{-(1+i):-1}\right) \quad \{i : 0 \le i < t_k, i \in \mathcal{Z}\}. \tag{267}$$

This proof also shows we can sample this stochastic process using the conditional distribution defined in Eqn. 272.

We start by showing that Eqn. 181 is necessary. Suppose the process with joint distribution $p(\vec{x}_1, \vec{x}_2, \ldots)$ is a $t_k + 1$ order stationary process. For $i < t_k$ we may compute the marginal distribution $p(s_t^{-i:0})$ at any t by marginalizing the distribution $p(\vec{s}_t)$ over all subsequences $\{\vec{x}_{t-t_k}, \ldots, \vec{x}_{t-j-1}\}$,

$$p(s_t^{-i:0}) = p(\vec{x}_{t-i}, \ldots, \vec{x}_t) = \int_{\vec{x}_{t-t_k},\ldots,\vec{x}_{t-i-1}} p(\vec{x}_{t-t_k}, \ldots, \vec{x}_t). \tag{268}$$

$p(s_t^{-(i+1):-1})$ is the marginal distribution on subsequences $\{\vec{x}_{t-i-1}, \ldots, \vec{x}_{t-1}\}$. We may compute this distribution by marginalizing $p(\vec{s}_{t-1}) = p(\vec{x}_{t-1-t_k}, \ldots, \vec{x}_{t-1})$ over all sequences $\{\vec{x}_{t-1-t_k}, \ldots, \vec{x}_{t-i-2}\}$,

$$p(s_t^{-(i+1):-1}) = p(s_{t-1}^{-i:0}) = \int_{\vec{x}_{t-1-t_k}, \ldots, \vec{x}_{t-i-2}} p(\vec{x}_{t-1-t_k}, \ldots, \vec{x}_{t-1}). \tag{269}$$

Since $\vec{s}_t$ is a t_k+1 stationary process it follows that by definition $p(\vec{x}_{t-1-t_k}, \ldots, \vec{x}_{t-1}) = p(\vec{x}_{t-t_k}, \ldots, \vec{x}_t)$. Thus it follows that

$$\int_{\vec{x}_{t-t_k}, \ldots, \vec{x}_{t-i-1}} p(\vec{x}_{t-t_k}, \ldots, \vec{x}_t) = \int_{\vec{x}_{t-1-t_k}, \ldots, \vec{x}_{t-i-2}} p(\vec{x}_{t-1-t_k}, \ldots, \vec{x}_{t-1}) \tag{270}$$

$$p(s_t^{-i:0}) = p(s_t^{-(i+1):-1}). \tag{271}$$

To show Eqn. 181 is sufficient, we need to show that for any $p(\vec{s})$ satisfying Eqn. 181 we can always construct a process which has a stationary marginal distribution equal to $p(\vec{s})$. We can construct such a process by first defining the time-invariant conditional distribution

$$p(\vec{x}_t = a_t^0 | \vec{x}_{t-t_k} = a_t^{-t_k}, \ldots, \vec{x}_{t-1} = a_t^{-1}) \triangleq \begin{cases} \frac{p(\vec{s}=a_t)}{p(s^{-t_k:-1}=a_t^{-t_k:-1})} & \text{if} \quad p(s^{-t_k:-1} = a_t^{-t_k:-1}) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{272}$$

We can easily show that the stochastic process defined by the joint distribution

$$p(\vec{x}_1, \ldots, \vec{x}_t) = p(\vec{x}_1, \ldots, \vec{x}_{t_k+1}) \prod_{i=t_k+2}^{t} p(\vec{x}_i | \vec{x}_{i-t_k}, \ldots, \vec{x}_{i-1}) \tag{273}$$

$$p(\vec{x}_1, \ldots, \vec{x}_{t_k+1}) = p(\vec{s}) \tag{274}$$

is a $t_k + 1$ order stationary process with a marginal joint distribution equal to $p(\vec{s})$. We can easily prove this using induction to show that $p(\vec{s}_t) = p(\vec{s})$. By definition $p(\vec{x}_1, \ldots, \vec{x}_{t_k+1}) = p(\vec{s})$. To prove $p(\vec{x}_{t-t_k}, \ldots, \vec{x}_t) = p(\vec{s})$ by induction for

all t we assume $p(\vec{x}_{t-t_k}, \ldots, \vec{x}_t) = p(\vec{s})$ and then show using this assumption that $p(\vec{x}_{t-t_k+1}, \ldots, \vec{x}_{t+1}) = p(\vec{s})$.

$$p(\vec{x}_{t-t_k+1}, \ldots, \vec{x}_{t+1}) = \int_{\vec{x}_{t-t_k}} p(\vec{x}_{t-t_k}, \ldots, \vec{x}_{t+1}) \tag{275}$$

$$= \int_{\vec{x}_{t-t_k}} p(\vec{x}_{t+1} | \vec{x}_{t-t_k+1}, \ldots, \vec{x}_t) p(\vec{x}_{t-t_k}, \ldots, \vec{x}_t), \tag{276}$$

Suppose $p(\vec{x}_{t-t_k+1}, \ldots, \vec{x}_t) \neq 0$, then using Eqn. 272

$$p(\vec{x}_{t-t_k+1}, \ldots, \vec{x}_{t+1}) = \int_{\vec{x}_{t-t_k}} \frac{p(\vec{s} = a_{t+1})}{p(s^{-t_k:-1} = a_{t+1}^{-t_k:-1})} p(\vec{x}_{t-t_k}, \ldots, \vec{x}_t) \tag{277}$$

$$= \frac{p(\vec{s} = a_{t+1})}{p(s^{-t_k:-1} = a_{t+1}^{-t_k:-1})} \int_{\vec{x}_{t-t_k}} p(\vec{x}_{t-t_k}, \ldots, \vec{x}_t) \tag{278}$$

$$= \frac{p(\vec{s} = a_{t+1})}{p(s^{-t_k:-1} = a_{t+1}^{-t_k:-1})} p(\vec{x}_{t-t_k+1}, \ldots, \vec{x}_t) \tag{279}$$

By the inductive hypothesis

$$p(\vec{x}_{t-t_k+1}, \ldots, \vec{x}_t) = p(s^{-t_k+1:0}). \tag{280}$$

Now since $p(\vec{s})$ satisfies Eqn. 181,

$$p(s^{-t_k+1:0}) = p(s^{-t_k:-1}). \tag{281}$$

Thus,

$$p(\vec{x}_{t-t_k-1}, \ldots, \vec{x}_{t+1}) = \frac{p(\vec{s} = a_{t+1})}{p(s^{-t_k:-1} = a_{t+1}^{-t_k:-1})} p(s^{-t_k:-1} = a_{t+1}^{-t_k:-1}) \qquad = p(\vec{s}) \tag{282}$$

which completes the proof for $p(\vec{x}_{t-t_k+1}, \ldots, \vec{x}_t) \neq 0$.

Now suppose $p(\vec{x}_{t-t_k+1} = a_t^{-t_k}, \ldots, \vec{x}_t = a_t^{-1}) = 0$. In this case Eqn. 272 does not specify a proper distribution because it assigns zero probability to all $\vec{x}_{t+1}$. Thus, the sequence is effectively terminated with $\vec{x}_t$. Consequently to show that Eqn. 272 defines a $t_k + 1$ order stationary process we must show that we would never generate a subsequence for which $p(\vec{x}_{t-t_k+1} = a_t^{-t_k}, \ldots, \vec{x}_t = a_t^{-1}) = 0$. So we need to prove

$$\text{If } p(s^{-t_k:-1} = a^{-t_k:-1}) = 0 \text{ then} \tag{283}$$

$$p(\vec{x}_{t-t_k+1} = a^{-t_k}, \ldots, \vec{x}_t = a^{-1}) = 0 \ \forall t. \tag{284}$$

We can prove this by induction. By definition $p(\vec{s}_1) = p(\vec{s})$. Therefore

$$p(s_1^{-t_k:-1} = a^{-t_k:-1}) = p(s^{-t_k:-1} = a^{-t_k:-1}) \tag{285}$$

$$= 0. \tag{286}$$

Now to complete our proof we need to show that if $p(s_t^{-t_k:-1} = a^{-t_k:-1}) = 0$ then $p(s_{t+1}^{-t_k:-1} = a^{-t_k:-1}) = 0$; that is we need to show that if $s_t^{-t_k+2:0} = a^{-t_k:-2}$ then we would never pick $\vec{x}_{t+1} = a^{-1}$ using 272.

The inductive hypothesis is that $p(s_{t'}^{-t_k:-1} = a^{-t_k:-1}) = 0 \; \forall \, t' \leq t$. Thus from the inductive hypothesis and Eqn. 272 it follows that

$$p(\vec{s}_{t+1}) = p(\vec{s}) \tag{287}$$

$$\Rightarrow p(s_{t+1}^{-t_k+1:0} = a^{-t_k:-1}) = p(s^{-t_k+1:0} = a^{-t_k:-1}). \tag{288}$$

Now since $p(\vec{s})$ satisfies Lemma 2

$$p(s^{-t_k+1:0} = a^{-t_k:-1}) = p(s^{-t_k:-1} = a^{-t_k:-1}). \tag{289}$$

Thus,

$$p(s_{t+1}^{-t_k+1:0} = a^{-t_k:-1}) = p(s^{-t_k:-1} = a^{-t_k:-1}) = p(s_t^{-t_k:-1} = a^{-t_k:-1}) = 0. \tag{290}$$

where the last equality follows from the inductive hypothesis.

3.6.3 Computing the average information for a Gaussian process

In this section we show how the average information per trial, Eqn. 216, can be computed when the input distribution is a Gaussian process. The structure of the GLM and the Gaussian distribution for $p(\vec{s})$ makes it relatively easy to compute $E_{\vec{s}} J_{exp}(\vec{s}, \vec{\theta})$. For the 1-d GLM the expected Fisher information matrix has a simple

1-dimensional dependence on $\vec{\theta}$,

$$J_{exp}(\vec{s},\vec{\theta}) = J_{exp}(\vec{s}^T\vec{\theta})\vec{s}\vec{s}^T \tag{291}$$

$$J_{exp}(\vec{s}^T\vec{\theta}) = -E_r\frac{\partial^2 \log p(r|\rho = \vec{s}^T\vec{\theta})}{\partial\rho^2}\vec{s}\vec{s}^T \tag{292}$$

$$= J_{exp}(\rho = \vec{s}^T\vec{\theta})\vec{s}\vec{s}^T. \tag{293}$$

This 1-dimensional structure along with the fact that $p(\vec{s})$ is Gaussian makes computing the expectations tractable. We start by defining a new coordinate system in which the first axis is aligned with $\vec{\theta}$. This coordinate system is defined by the orthonormal matrix, $\mathcal{R}_{\vec{\theta}}$. The first column of $\mathcal{R}_{\vec{\theta}}$ is $\frac{\vec{\theta}}{||\vec{\theta}||_2}$ and the remaining columns are a suitable set of orthonormal vectors. We can thus define the transformation of $\vec{s}$ and $\vec{\theta}$ into this new coordinate system,

$$\vec{\theta}' = \mathcal{R}_{\vec{\theta}}^T\vec{\theta} \tag{294}$$

$$\vec{w} = \mathcal{R}_{\vec{\theta}}^T\vec{s}. \tag{295}$$

This coordinate system has the convenient properties

$$\theta_i' = 0 \quad \forall i \neq 1 \tag{296}$$

$$\Rightarrow \vec{w}^T\vec{\theta}' = w_1\theta_1'. \tag{297}$$

We can now rewrite our objective function

$$\mathcal{F}(p(\vec{s})) = E_{\vec{\theta}}\log|E_{p(\vec{s})}J_{exp}(\rho)\vec{s}\vec{s}^T| \tag{298}$$

$$= E_{\vec{\theta}}\log|E_{p(\vec{w})}J_{exp}(w_1\theta_1')\vec{w}\vec{w}^T| \tag{299}$$

$$= E_{\vec{\theta}}\log|E_{w_1}J_{exp}(w_1\theta_1')E_{w_2,\ldots,w_{\dim(\vec{s})}|w_1}\vec{w}\vec{w}^T| \tag{300}$$

Since $p(\vec{s})$ is Gaussian and $\vec{w} = \mathcal{R}_{\vec{\theta}}^T\vec{s}$, $p(\vec{w})$ is Gaussian with mean $\vec{w}^T\mu_s$ and covariance matrix $\vec{w}^TC_s\vec{w}$. Consequently, $p(\vec{w}|w_1)$ is also Gaussian and can be computed

using the standard Gaussian conditioning formulas,

$$p(\vec{w}|w_1) = \mathcal{N}(\mathcal{R}_{\vec{\theta}}^T \mu_s + \frac{1}{\sigma_{\omega_1}^2}\mathcal{R}_{\vec{\theta}}^T \boldsymbol{\gamma}(w_1 - \mu_{\omega_1}), \mathcal{R}_{\vec{\theta}}^T C_s \mathcal{R}_{\vec{\theta}} - \frac{1}{\sigma_{\omega_1}^2}\mathcal{R}_{\vec{\theta}}^T \boldsymbol{\gamma}\boldsymbol{\gamma}^T) \tag{301}$$

$$\mu_{\omega_1} = \frac{\vec{\theta}^T}{||\vec{\theta}||_2}\mu_s \tag{302}$$

$$\sigma_{\omega_1}^2 = \frac{\vec{\theta}^T}{||\vec{\theta}||_2} C_s \frac{\vec{\theta}}{||\vec{\theta}||_2} \tag{303}$$

$$\boldsymbol{\gamma} = C_s \frac{\vec{\theta}}{||\vec{\theta}||_2}. \tag{304}$$

Using this distribution we can easily compute the conditional expectation,

$$E_{\vec{w}|w_1}\vec{w}\vec{w}^T = \mathcal{R}_{\vec{\theta}}^T \left(C_s - \frac{1}{\sigma_{\omega_1}^2}\boldsymbol{\gamma}\boldsymbol{\gamma}^T + \left(\mu_s + \frac{1}{\sigma_{\omega_1}^2}\boldsymbol{\gamma}(w_1 - \mu_{\omega_1})\right)\left(\mu_s + \frac{1}{\sigma_{\omega_1}^2}\boldsymbol{\gamma}(w_1 - \mu_{\omega_1})\right)^T \right) \mathcal{R}_{\vec{\theta}} \tag{305}$$

$$= \mathcal{R}_{\vec{\theta}}^T \left(C_s + \left(\mu_s + \frac{1}{\sigma_{\omega_1}^2}\boldsymbol{\gamma}(w_1 - \mu_{\omega_1}) - \frac{1}{\sqrt{\sigma_{\omega_1}^2}}\boldsymbol{\gamma}\right) \times \left(\mu_s + \frac{1}{\sigma_{\omega_1}^2}\boldsymbol{\gamma}(w_1 - \mu_{\omega_1}) + \frac{1}{\sqrt{\sigma_{\omega_1}^2}}\boldsymbol{\gamma}\right)^T \right) \mathcal{R}_{\vec{\theta}} \tag{306}$$

$$= \mathcal{R}_{\vec{\theta}}^T \left(C_s + \left(\vec{\kappa}w_1 + \vec{\delta}\right)\left(\vec{\kappa}w_1 + \vec{\eta}\right)^T \right) \mathcal{R}_{\vec{\theta}} \tag{307}$$

$$\vec{\kappa} = \frac{\boldsymbol{\gamma}}{\sigma_{\omega_1}^2} \tag{308}$$

$$\vec{\delta} = \mu_s - \frac{\boldsymbol{\gamma}}{\sigma_{\omega_1}^2}\mu_{\omega_1} - \frac{\boldsymbol{\gamma}}{\sqrt{\sigma_{\omega_1}^2}} \tag{309}$$

$$\vec{\eta} = \mu_s - \frac{\boldsymbol{\gamma}}{\sigma_{\omega_1}^2}\mu_{\omega_1} + \frac{\boldsymbol{\gamma}}{\sqrt{\sigma_{\omega_1}^2}} \tag{310}$$

$$\tag{311}$$

The key point is the expected value is just a rank-1 perturbation of a rotated C_s. We

can now evaluate the expectation over w_1,

$$E_{w_1} J_{exp}(w_1\theta_1^{'}) E_{\vec{w}|w_1} \vec{w}\vec{w}^T = \mathcal{R}_{\vec{\theta}}^T \left(C_s \varpi_1 + \varpi_3 \left[(\vec{\kappa} + \frac{\varpi_2}{\varpi_3}\vec{\delta})(\vec{\kappa} + \frac{\varpi_2}{\varpi_3}\vec{\eta})^T + (\frac{\varpi_1}{\varpi_3} - (\frac{\varpi_2}{\varpi_3})^2)\vec{\delta}\vec{\eta}^T \right] \right) \mathcal{R}_{\vec{\theta}} \tag{312}$$

$$\varpi_1 = E_{w_1} J_{exp}(w_1\theta_1^{'}) \tag{313}$$

$$\varpi_2 = E_{w_1} J_{exp}(w_1\theta_1^{'}) w_1 \tag{314}$$

$$\varpi_2 = E_{w_1} J_{exp}(w_1\theta_1^{'}) w_1^2 \tag{315}$$

$w_1 = \frac{\vec{\theta}^T}{||\vec{\theta}||_2}\vec{s}$, $p(w_1)$ is Gaussian with mean and variance $(\mu_{\omega_1}, \sigma_{\omega_1}^2)$. The above are just 1-dimensional expectations so for any value of $\vec{\theta}$ we could compute them numerically.

Eqn. 312 is a rank 2 update of $\boldsymbol{\alpha}$. Therefore we can use the matrix determinant lemma to compute $|E_{w_1} E_{\vec{w}|w_1} J_{exp} \vec{w}\vec{w}^T|$,

$$\log |E_{w_1} E_{\vec{w}|w_1} J_{exp}(w_1\theta_1^{'}) \vec{w}\vec{w}^T|$$

$$= \dim(C_s) \log \varpi_1 + \log |I + V^T(\varpi_1 C_s)^{-1} U| + \log |C_s| \tag{316}$$

$$U = \varpi_3 \left[(\vec{\kappa} + \frac{\varpi_2}{\varpi_3}\vec{\delta}), (\frac{\varpi_1}{\varpi_3} - (\frac{\varpi_2}{\varpi_3})^2)\vec{\delta} \right] \tag{317}$$

$$V = \left[(\vec{\kappa} + \frac{\varpi_2}{\varpi_3}\vec{\eta}), \vec{\eta} \right] \tag{318}$$

Since $I + V^T(\varpi_1 C_s)^{-1} U$ is a 2-d matrix, we can compute its determinant analytically. Taking the expectation with respect to $\vec{\theta}$ yields,

$$E_{\vec{\theta}} \log |E_{w_1} E_{\vec{w}|w_1} J_{exp}(w_1\theta_1^{'}) \vec{w}\vec{w}^T|$$

$$= \dim(C_s) E_{\vec{\theta}} \log \varpi_1 + E_{\vec{\theta}} \log |I + V^T(\varpi_1 C_s)^{-1} U| + \log |C_s|. \tag{319}$$

3.6.4 Finding the optimal C_s given $||\mu_s||$.

In Section 3.3.2.1 we presented a method for optimizing a lower bound for the utility of a design by dropping the Toeplitz constraints on C_s. Our solution breaks up the optimization into an outer problem, which finds the optimal value of $||\mu_s||$, and an inner problem which finds the best C_s given $||\mu_s||$. In this appendix we present the details of our solution for computing the optimal C_s by solving the inner problem.

The inner problem is

$$\arg\max_{C_s} \frac{d_s}{2} Tr(C_s R) + \log|C_s| \tag{320}$$

$$s.t \quad \vec{s}^T C_s \vec{s} > 0 \quad \forall \vec{s} \neq 0 \tag{321}$$

$$Tr(C_s) < m - ||\mu_s||^2 \tag{322}$$

where $d_s = \dim(\vec{s})$ and m is the maximum average power we want the stimuli to have. We can maximize this subject to the power constraint by introducing a Lagrange multiplier,

$$L = \frac{d_s}{2} Tr(C_s R) + \log|C_s| - \lambda Tr(C_s) \tag{323}$$

$$= \frac{d_s}{2} Tr(C_s(R - \lambda I)) + \log|C_s| \tag{324}$$

$$= \frac{d_s}{2} Tr(C_s A) + \log|C_s| \tag{325}$$

$$A = R - \lambda I \tag{326}$$

$$= O diag(\vec{a}) O^T \tag{327}$$

Here O is defined as the eigenvectors of A and $\vec{a}$ are the eigenvalues of A. We can use these eigenvectors to define

$$B = O' C_s O \tag{328}$$

Since $\log|C_s|$ depends only on its eigenvalues, its eigenvectors are completely determined by $Tr(C_s R)$. Thus

$$L = \frac{d_s}{2} Tr(C_s A) + \log|C_s| \tag{329}$$

$$= \frac{d_s}{2} Tr(OBO' O diag(\vec{a}) O') + \log|OBO'| \tag{330}$$

$$= \frac{d_s}{2} Tr(OB diag(\vec{a}) O') + \log|OBO'| \tag{331}$$

$$= \frac{d_s}{2} Tr(B diag(\vec{a})) + \log|B| \tag{332}$$

Now for C_s to be positive definite B must be positive definite. Using this fact we can show that the maximum must occur at a value of λ which makes A negative

definite. Clearly, the power constraint must be active because $Tr(C_sR) + \log|C_s|$ is increasing with respect to the eigenvalues of C_s because R is positive definite. For A to be negative definite λ must be greater than the largest eigenvalue of R^2.

Now we can define Q as $-B{\cdot}diag(\vec{a})$ and its eigendecomposition as $Q = Wdiag(\vec{q})W'$. Therefore,

$$L = -\frac{d_s}{2}Tr(Q) + \log|-Qdiag(\vec{a})^{-1}| \tag{334}$$

$$L = \sum_i -\frac{d_s}{2}q_i + \log q_i + \log|-diag(\vec{a})^{-1}| \tag{335}$$

To maximize L with respect to q_i we can just take the derivative of L and set it equal to zero.

$$\frac{\partial L}{\partial q_i} = -\frac{d_s}{2} + \frac{1}{q_i} = 0 \tag{336}$$

$$\Rightarrow q_i = \frac{2}{d_s}. \tag{337}$$

Since our objective function with respect to Q is independent of the eigenvectors of Q we may choose any eigenvectors we like. As a result, we can set $W = I$ as this choice makes solving for C_s easy. Therefore Q is just proportional to the identity matrix and we can easily solve for B.

$$Q = \frac{2}{d_s}I = -Bdiag(\vec{a}) \tag{338}$$

$$B = -\frac{2}{d_s}diag(\vec{a})^{-1} \tag{339}$$

$$= \frac{2}{d_s}(\lambda I - diag(\vec{r}))^{-1} \tag{340}$$

[2] R is positive definite so its eigenvalues r_i are positive. Now suppose $\lambda < r_i$ for some i. In this case we can let B be a diagonal matrix with $B_{jj} = 1 \quad \text{for } j \neq i$. In this case our objective function is

$$L = \frac{d_s}{2}(\sum_{j\neq i} a_j + B_{ii}a_i) + \log B_{ii} \tag{333}$$

Since $a_i > 0$, we can make L arbitrarily large by increasing B_{ii}. Since $C_s = OBO'$ the power constraint would not be satisfied. Thus we can conclude that $\lambda > \max r_i$ as this ensures $a_i < 0 \quad \forall\, i$

We can now solve for λ by plugging B into our power constraint.

$$Tr(C_s) = \frac{2}{d_s} \sum_i \frac{1}{\lambda - r_i} = m - ||\mu_s||^2 \tag{341}$$

We can easily solve this equation numerically to compute the optimal value of λ as a function of $||\mu_s||$. We can then do a search over all $||\mu_s||$ to find the optimal value (μ_s, C_s).

CHAPTER IV

OPTIMAL LEARNING OF SONG BIRD AUDITORY RECEPTIVE FIELDS USING GENERALIZED LINEAR MODELS.

In this chapter we discuss the application of our methods to designing optimal experiments for learning the receptive fields of auditory neurons in zebra finch. We show using real data that 1) the generalized linear model (GLM) can be used to estimate the receptive field of auditory neurons in zebra finch and 2) by optimizing experiments using our methods we can reduce the number of trials needed to estimate the receptive field. Using data obtained from actual experiments, we simulated an information maximizing design and show a factor of 3 reduction in the number of trials required to fit a GLM compared to a non-optimized design. We show how over-fitting of the STRF can be avoided by using a prior which acts like a low-pass filter. Furthermore, we consider the problem of computing an optimal sequence of inputs and show how this problem may be solved using our existing methods.

4.1 Introduction

In the previous chapters we have presented a rigorous approach to the problem of optimizing neurophysiology experiments using GLMs. Using simulations, we have shown that our methods can improve the amount of information gathered during an experiment by an order of magnitude. A major limitation of these simulations was that the data was generated from a GLM. As a result these simulations failed to address how well our methods would work in actual neurophysiology experiments. In particular, our methods assume that the GLM provides an adequate enough model of

the neuron that we may use it to predict the informativeness of different stimuli. In this chapter we address the issue of model misspecification by considering a particular application; learning how auditory neurons in adult zebra finch encode information. The goal of this chapter is to show using offline analysis of real data that 1) the GLM predicts the responses of auditory neurons with sufficient accuracy for the purposes of designing better experiments and 2) using our methods we could potentially increase the amount of data gathered during experiments.

The study of songbirds has a long history in the neuroscience community [163]. The auditory system of songbirds has received a great deal of attention because song plays a crucial role in behavior. Male birds use songs as a defense mechanism and to attract mates [21] while females use songs to facilitate cooperation and pair interaction [102]. Similarities between bird-songs and human speech create the possibility that understanding auditory processing in birds will lead to a better understanding of vocal communication in humans. Human speech and bird song are both complex auditory sounds and exhibit similar spectral and temporal features [140]. Furthermore in both humans and song birds auditory processing occurs in a hierarchy of brain areas and involves interactions between auditory and motor centers [101, 45]. Songbirds are also an ideal model for studying reinforcement learning because learning to sing requires interaction between motor and auditory processing [62].

The crucial role that song plays in bird behavior necessitates an auditory system capable of recognizing and discriminating the songs of different birds. Yet, how the auditory system of songbirds performs these functions is relatively unknown. Experiments have repeatedly shown that neurons in the auditory system of songbirds respond preferentially to natural sounds, such as the vocalizations of other birds [122, 151, 167, 168]. Previous work has also shown that some of this auditory processing occurs at the level of single cells. In zebra finch, auditory neurons have been shown to selectively respond to con-specific sounds [150, 67]. Similarly in starlings, single

neurons have been shown to selectively respond to specific acoustical features called motifs which could play a crucial in song recognition [62].

One of the primary advantages of using sensory systems to study neural encoding and decoding is that experimentalists have at least some sense of which stimuli are behaviorally relevant. This has lead to a crude form of stimulus optimization based in part on the efficient coding hypothesis [6]. Early auditory neurophysiology experiments used artificial stimuli like white noise, tone pips, and ripple stimuli [52, 50, 51, 49, 42]. One reason for using these stimuli was that the reverse correlation methods used to estimate the receptive field required uncorrelated stimulus ensembles. Recent work has extended reverse correlation to take into account stimulus correlations so that the receptive field can be estimated in response to arbitrary stimulus ensembles [151, 149, 152]. These extensions of reverse correlation have allowed experimentalists to use more natural and behaviorally relevant sounds such as the vocalizations of other birds [62, 167, 168]. This progression from simple, artificial sounds to complex, natural sounds represents a crude form of stimulus optimization.

Previous work has also attempted to optimize the design of auditory neurophysiology experiments using methods similar to our own. Early work tried to adapt stimuli as data was collected to drive auditory neurons to fire at higher rates [110, 39]. A similar idea is to adapt stimuli to trace out iso-response curves which are curves in stimulus space along which the neuron's response is constant [66]. Recently, investigators have tried to adapt stimuli to maximize the mutual information between a neuron's response and the sensory input [97, 98]. This approach differs from our methods because we maximize the mutual information between the responses and the unknown parameters. Consequently our approach tries to find stimuli which will decrease our uncertainty about the unknown receptive field. In contrast, maximizing the mutual information between the input and output leads to stimuli which can be reconstructed from the responses with the least amount of error. Our methods,

however, will only work if the GLM provides a reasonable model of auditory neurons.

We expect the GLM to provide a useful model for stimulus optimization because it is more general then the linear model which has been used extensively in auditory neurophysiology. To date, most previous research has focused on using reverse correlation to fit a linear model to the responses of single neurons in the auditory system of songbirds [52, 151, 42, 167, 168]. The linear model has been shown to be a good first order model which is capable of predicting, for some neurons at least, the peaks and troughs in the peristimulus time histogram (PSTH) with high fidelity [136]. For other neurons, the linear model provided poor fits which led the authors to conclude that the neurons had a strong nonlinear component which the linear model failed to account for [136]. Since the GLM is a family of nonlinear models which includes the linear model as a special case, we can reasonably expect the GLM to do at least as well as and hopefully better than the linear model.

Single cell auditory neurophysiology in songbirds therefore provides a good application for testing our methods for sequential optimal experimental design because 1) single neurons are known to play an important but poorly understood role in complex auditory processing and 2) choosing appropriate stimuli has proven crucial to understanding auditory processing. In this chapter we investigate the application of our methods to single cell auditory neurophysiology experiments. In Section 4.2.1 we briefly describe the actual setup of experiments and the data collected. In Section 4.2.2 we discuss how the GLM may be fitted to the bird song data and present results illustrating the quality of the estimated receptive fields. In Section 4.3 we show how our methods for optimal experimental design can be tested offline with the bird song data. The results show that our methods reduce the amount of data needed to fit the GLM by on average a factor of 3. We expect these results to underestimate the actual improvement that could be achieved because our offline analysis was restricted to picking stimuli from the small set of stimuli actually presented.

4.2 Fitting a GLM to auditory neurons in MLd

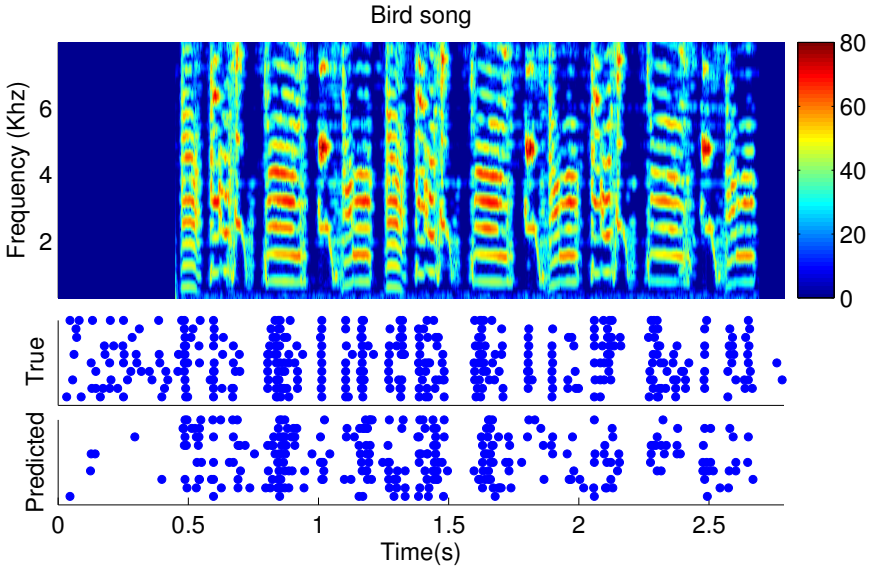
A typical experiment in sensory neurophysiology entails recording the response of a neuron to some known stimuli. After the data is collected, reverse correlation, least-squares regression, or maximum likelihood can be used to estimate the receptive field of the neuron. [52, 123, 36, 170]. In this section, we describe our efforts to learn the receptive field of auditory neurons in zebra finch using the GLM. We begin this section by briefly summarizing the key points of the experiments. We then discuss our efforts to fit the GLM and finally present the receptive fields learned with our methods. The main point of this section is that fitting a GLM to auditory data using maximum likelihood leads to receptive fields that are very similar to those produced using liner models [149, 168].

4.2.1 Experimental setup

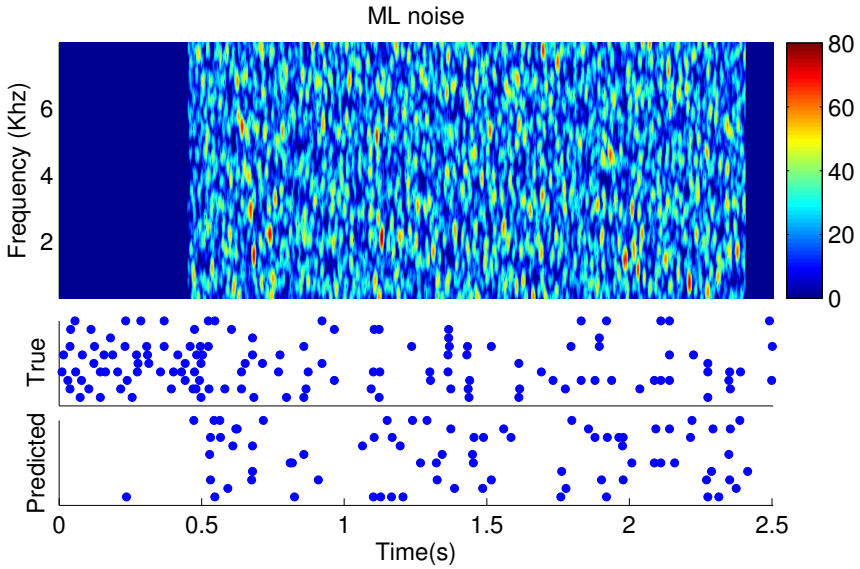
The experiments in this case were performed by our collaborators David Schneider and Dr. Sarah Woolley. During the experiments, Schneider and Woolley played wave-files to a bird and recorded the responses of neurons in the Mesencephalicus lateralis pars dorsalis (MLd) of an adult male zebra finch using extracellular electrodes (for details see [168]). MLd is a midbrain region which could be responsible for conspecific tuning [165, 166].

The set of stimuli consisted of 20 different segments taken from bird-songs and 10 modulation-limited noise stimuli (ml-noise), which is described below. Each wave file had a duration of approximately 2 seconds. In general, each wave file was repeated 10 different times to the bird in a random order. Before and after each wave file was played there was a period of silence lasting roughly .5 seconds. This period of silence allowed the neuron to return to its resting state before the next wave file was played thereby minimizing the effects of adaptation [172].

Examples of each type of stimulus and an accompanying raster plot for one neuron



(a)



(b)

Figure 22: a) The top plot shows the spectrogram of one of the bird songs used during the experiments. The spectrogram includes the periods of silence before and after the actual stimulus. The middle plot shows the raster plot of the recorded neuron's spiking in response to this stimulus. The bottom plot shows the predicted raster plot computed using a GLM fitted to the training set. Each row of the raster plots shows the firing of the neuron on independent presentations of the input. The training set did not include this wave-file or the wave-file shown in (b). b) The same as A except the stimulus is ml-noise. When fitting a GLM, the stimulus, $\vec{x}_t$, corresponds to one column of the spectrogram.

are shown in Figure 22. In response to the vocalizations of other birds this neuron tends to respond more strongly and in a more stereotyped fashion then in response to ml-noise. These results are consistent with results by other investigators showing that auditory neurons respond more strongly to natural sounds [122, 55, 73]. However our collaborators report that across the population, not all neurons respond as strongly or as reliably to vocalizations as the neuron exhibited in Figure 22 (personal correspondence with David Schneider). For comparison, ml-noise was also presented to the birds. ML-noise is a form of broadband noise which is similar to white noise. The power spectrum of ml-noise is flat but band limited to 250-8000Hz, which is the frequency range occurring in the songs of zebra finch [112, 172]. The ml-noise was also designed to have the same power and maximum spectral and temporal modulations that occur in the songs of adult zebra finch [73, 168]. Thus, ml-noise can be used to contrast the responses to con specific vocalizations compared to noise stimuli with similar spectral properties.

4.2.2 Fitting a GLM

In this section we describe our efforts to estimate the auditory receptive of neurons in the MLd region of zebra finch by fitting a GLM to their spike trains. Since the GLM assigns a likelihood to the observed responses given the inputs, we can fit the GLM by maximizing the likelihood as described in Chapter 2.

The first step in fitting a GLM is to choose 1) the distribution in the exponential family to use and 2) the nonlinearity. We used the canonical Poisson model because the canonical Poisson has some desirable computational properties. For the canonical Poisson the expected Fisher information is exponential and independent of the responses. This property makes the computations required in the next section to optimize the design much more tractable. In the discussion we consider other GLMs that we might have used and their potential impact on our results.

The canonical Poisson is a firing rate model which assigns a probability to the number of spikes we expect to observe in some window of time. The GLM provides a mechanism for computing the expected firing rate as a function of the stimulus, past responses, and background firing rate Chapter 2. The log-likelihood of the response at time t is thus

$$\log p(r_t|\vec{s}_t, \vec{\theta}) = -\log r_t! + r_t \exp(\vec{s}_t^T \vec{\theta}) - \exp(\vec{s}_t^T \vec{\theta}) \tag{342}$$

$$\vec{s}_t^T = \{\vec{x}_{t-t_k}^T, \ldots, \vec{x}_t^T, r_{t-t_a}^T, \ldots, r_t^T, 1^T\}. \tag{343}$$

The response, r_t, is the number of spikes observed in some small time window. The input, $\vec{s}_t$, consists of the most recent $t_k + 1$ stimuli, the most recent t_a responses of the neuron, and a constant term 1. The constant term allows us to include a bias which can be used to set the background firing rate of the neuron.

For auditory neurons, the receptive field of the neuron is typically represented in the spectral temporal domain because the early auditory system is known to perform a frequency decomposition. Furthermore, transforming the input into the spectral domain is a nonlinear transformation which generally improves the accuracy of the linear model for auditory data [64]. The spectro-temporal receptive field (STRF) of the neuron, $\vec{\theta}_x(\tau, \omega)$, is a 2-d filter which relates the firing rate at time t to the amount of energy at frequency ω and time $t - \tau$ in the stimulus. The subscript on $\vec{\theta}$ is used to distinguish the elements of $\vec{\theta}$ which measure the dependence of the response on the stimulus, spike-history, and bias terms respectively.

The stimuli and responses were computed from the experimental data by dividing the recordings into time bins of 2.5ms. The time bin was small enough that more than one spike was almost never observed in any bins. To compute the corresponding stimulus, $\vec{x}_t$, we computed the power spectrum over a small interval of time centered on t [64]. The power was computed for frequencies in the range 300 to 8000 Hz in

intervals of approximately $100Hz$ [1]. Previous work has suggested that using frequency spacing around 125Hz is a good choice for computing the STRF [140, 64].

We initially fitted a GLM with an STRF that had a duration of 50ms, 20 time bins, and had 8 spike history terms as well as a bias term, for a total of 1589 unknown parameters[2]. The duration of the STRF and spike history dependence was chosen based on prior knowledge that these durations were long enough to capture most of the salient features of the STRF and spike history dependence [168]. Examples of the estimated STRF, spike-history, and bias terms are shown in Figure 23 and Figure 24. The STRFs are very noisy. Nonetheless the STRFs have similar temporal and frequency tuning to the STRFs trained on ml-noise using reverse-correlation methods presented in previous work [168]. Also plotted is the estimated spike history filter. The largest coefficients are negative and occur for delays close to zero. Thus the effect of the spike-history terms is to inhibit spiking immediately after the neuron fires. The spike history terms therefore help enforce a refractory period in the model. The bias terms were also very negative which corresponds to low background firing rates of roughly $3 - 5$Hz.

The high-frequency noise in the estimated STRFs is an indication of over-fitting. The 30 wave-files which are roughly 2s in duration translates into 20,000 distinct inputs when using a 50ms STRF. Furthermore, most of these inputs are highly correlated due to the structure of bird-song and the fact that we generate the inputs by sliding a window over the input's spectrogram. As a result, the fact that we are over-fitting the STRFs is not surprising. One way to deal with this noise is by incorporating a low pass filter into the STRF estimation procedure [151, 149, 152]. To low-pass filter the STRF, we represent the STRF in the frequency domain. We can then use the prior on the amplitudes of the frequency coefficients to bias the STRF

[1] The interval was not a round number because when the sounds were presented to the bird they were played at a sampling rate of 48828Hz.

[2] 79x20 coefficients of the STRF + 8 spike history coefficients +1 bias term=1589 unknowns.

towards smoother features when data is limited.

4.2.3 Using a frequency representation to smooth the STRF

To represent the STRF in the Fourier domain, we applied the Fourier transform separately to the spectral and temporal dimensions of the STRF because we wanted to represent the STRF as a linear combination of matrices which were separable in the spectral and temporal dimensions. Previous work has shown that low-rank approximations of the the STRF can be used to produce accurate approximations of the receptive field while significantly reducing the number of unknown parameters [42, 136, 120, 94, 2]. A low rank assumption is a more general version of the space-time separable assumption that is often used when studying visual receptive fields [38].

Applying the separable Fourier transform to the STRF is just a linear transformation. This transformation maps the STRF into a coordinate system in which the basis functions are rank one matrices. Each of these matrices is the product of 1-dimensional sine-waves in the spectral and temporal directions of the STRF. Using these basis functions we can write the STRF such that each row and column of the STRF is a linear combination of 1-d sine-waves,

$$\boldsymbol{\Theta}(i,j) = \sum_{\alpha=1}^{m_f}\sum_{\beta=1}^{m_t} \gamma^1_{\alpha,\beta} \sin(2\pi \cdot f_{o,f} \cdot \alpha \cdot i) \sin(2\pi \cdot f_{o,t} \cdot \beta \cdot j) \tag{344}$$

$$+ \sum_{\alpha=1}^{m_f}\sum_{\beta=0}^{m_t} \gamma^2_{\alpha,\beta} \sin(2\pi \cdot f_{o,f} \cdot \alpha \cdot i) \cos(2\pi \cdot f_{o,t} \cdot \beta \cdot j) \tag{345}$$

$$+ \sum_{\alpha=0}^{m_f}\sum_{\beta=1}^{m_t} \gamma^3_{\alpha,\beta} \cos(2\pi \cdot f_{o,f} \cdot \alpha \cdot i) \sin(2\pi \cdot f_{o,t} \cdot \beta \cdot j) \tag{346}$$

$$+ \sum_{\alpha=0}^{m_f}\sum_{\beta=0}^{m_t} \gamma^4_{\alpha,\beta} \cos(2\pi \cdot f_{o,f} \cdot \alpha \cdot i) \cos(2\pi \cdot f_{o,t} \cdot \beta \cdot j). \tag{347}$$

The functions $\sin(2\pi \cdot f_{o,f} \cdot \alpha \cdot i)$ and $\cos(2\pi \cdot f_{o,f} \cdot \alpha \cdot i)$ determine how each basis function varies across the spectral dimension of the STRF while the functions $\sin(2\pi \cdot f_{o,t} \cdot \beta \cdot j)$ and $\cos(2\pi \cdot f_{o,t} \cdot \beta \cdot j)$ determine how the basis functions vary across time in the STRF. Each pair of sine-waves measures the amount of energy at particular frequencies in

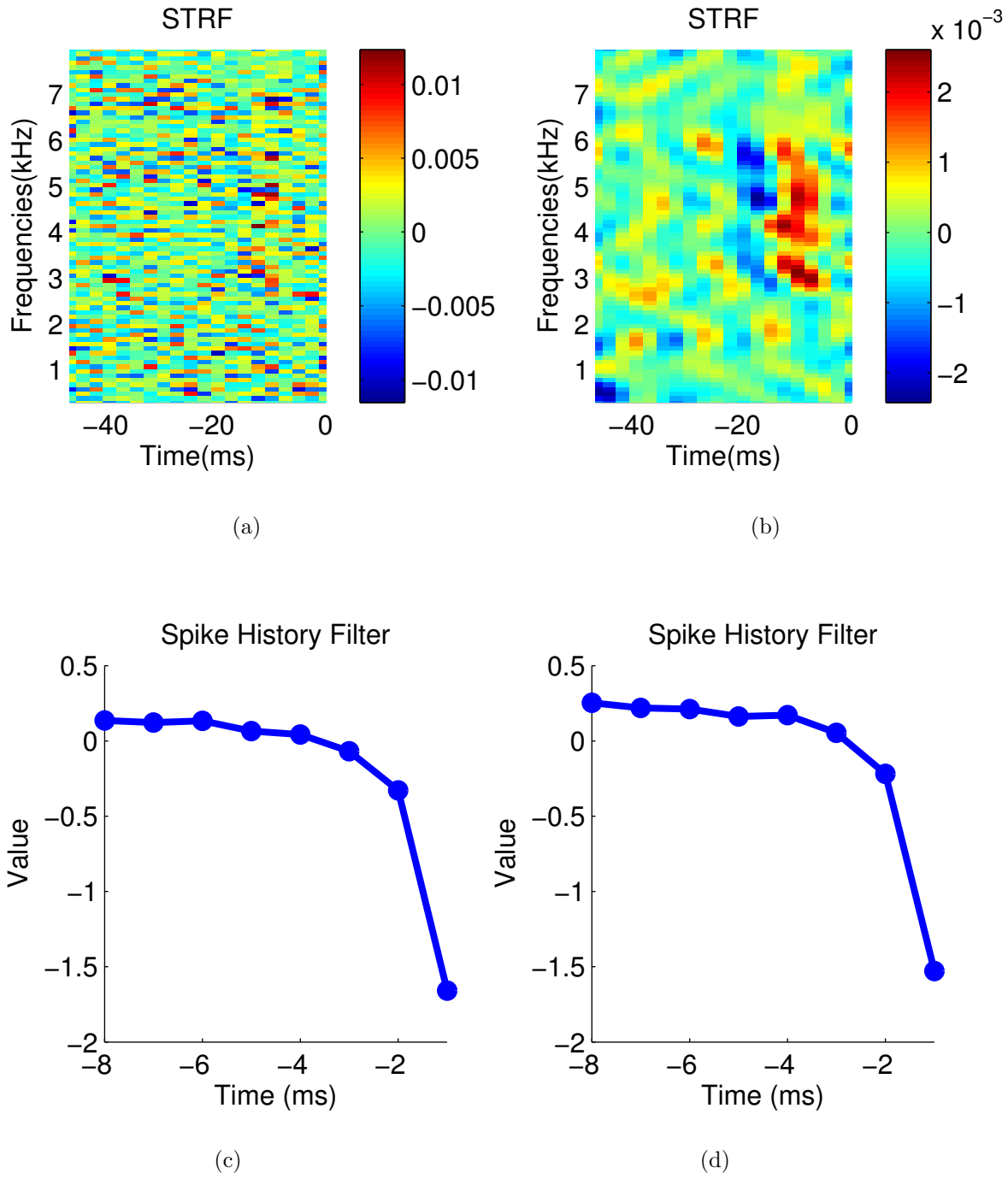


Figure 23: a) The STRF estimated without low-pass filtering. b) The STRF estimated with cutoff frequencies $n_{fc} = 10$ and $n_{tc} = 4$. c) The spike history for the model estimated in (a) (the curve shows the values of the filter coefficients at different delays). The bias in this case was -4.20. d) The spike history for the model estimated in (b). The bias in this case was -4.33.

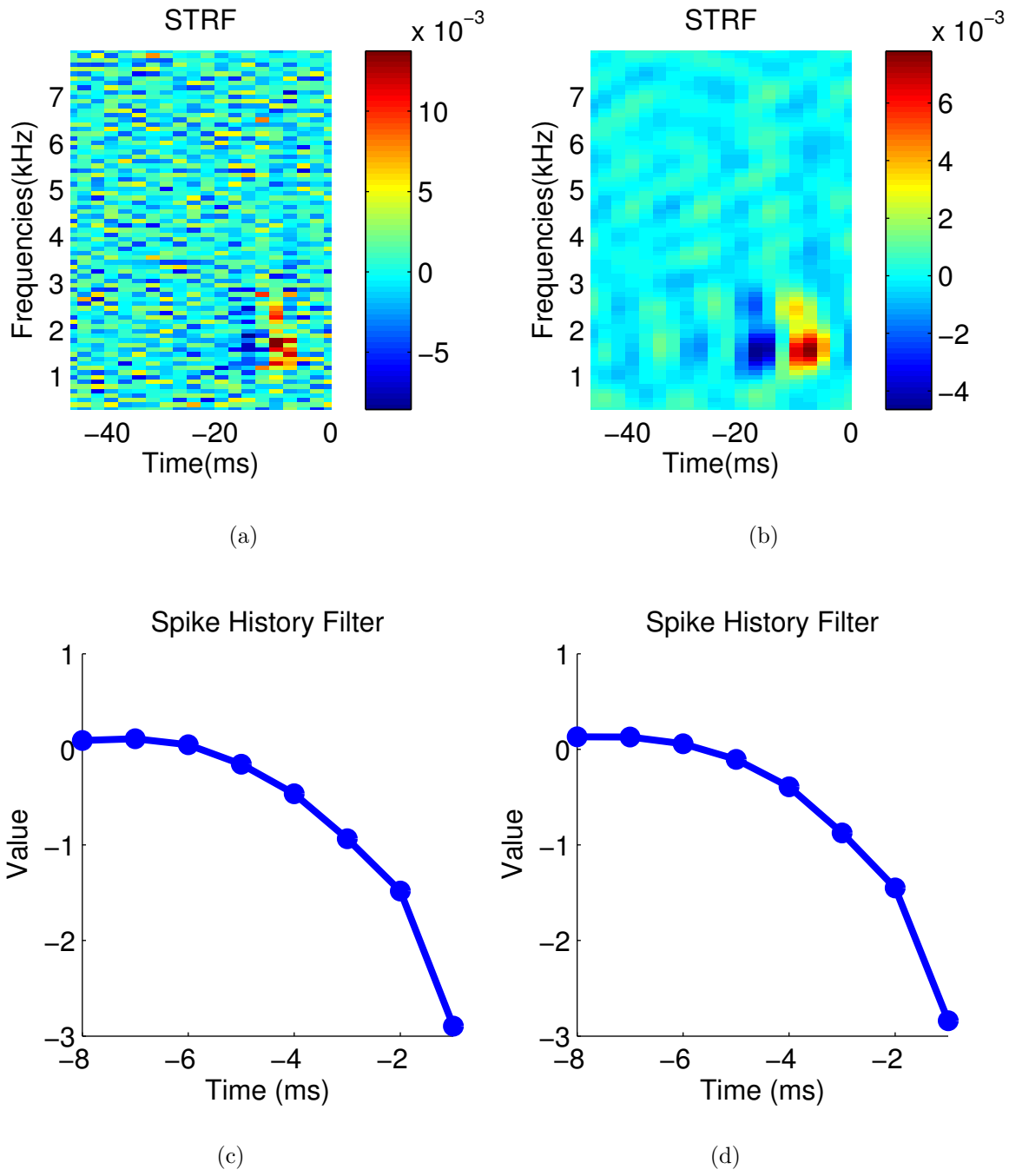


Figure 24: The same as Figure 23 except the data is from a different neuron. a) The STRF estimated without low-pass filtering. b) The STRF estimated with cutoff frequencies $n_{fc} = 10$ and $n_{tc} = 4$. c) The spike history for the model estimated in (a) (the curve shows the values of the filter coefficients at different delays). The bias in this case was -4.76. d) The spike history for the model estimated in (b). The bias in this case was -4.59.

the spectral and temporal dimensions. The amplitude of each frequency is determined by the coefficients $\gamma^i_{\alpha,\beta}$. To form an orthogonal basis for the STRF we need to project the STRF onto sinusoids with frequencies

$$\{0, f_{o,f}, 2f_{o,f}, \ldots m_f f_{o,f}\} \quad \{0, f_{o,t}, 2f_{o,t}, \ldots, m_t f_{o,t}\} \tag{348}$$

$$f_{o,f} = \frac{1}{n_f} \quad f_{o,t} = \frac{1}{n_t} \tag{349}$$

$$m_f = \lceil \frac{1}{2f_{o,f}} - 1 \rceil \quad m_t = \lceil \frac{1}{2f_{o,t}} - 1 \rceil. \tag{350}$$

$f_{o,f}$ and $f_{o,t}$ are the fundamental frequencies and are set so that 1 period corresponds to the dimensions of the STRF (n_t and n_f denote the dimensions of the STRF in the time and frequency dimensions respectively). m_f and m_t are the largest integers such that $m_f f_{o,f}$ and $m_t f_{o,t}$ are less than the Nyquist frequency. We subtract 1 and take the ceiling to make sure the frequencies of our basis functions are less than the Nyquist frequency. The unknown parameters in this new coordinate system are the amplitudes, $\vec{\gamma} = \{\gamma^1_{\alpha,\beta}, \gamma^2_{\alpha,\beta}, \gamma^3_{\alpha,\beta}, \gamma^4_{\alpha,\beta}\}$. For simplicity, we will continue to refer to the unknown parameters as $\vec{\theta}$ realizing that the STRF is represented using this new basis. Since this transformation is linear we can continue to apply our methods for fitting the GLM and optimizing the stimuli.

To low pass filter the STRF we can simply force the coefficients of $\vec{\theta}$ corresponding to high frequencies to zero; i.e we pick cutoffs n_{tc} and n_{fc} for the time and spectral directions respectively and set

$$\gamma^i_{\alpha,\beta} = 0 \quad \text{if } \alpha > n_{fc} \text{ or } \beta > n_{tc}. \tag{351}$$

Decreasing the cutoff frequencies not only makes the estimated STRFs smoother, it also reduces the dimensionality of the model. Reducing the dimensionality makes it easier to fit the GLM and optimize the stimuli but the risk is that the lower-dimensional model may be too simple to adequately model auditory neurons. We can mitigate this risk by using a soft-cutoff. Rather than force all high-frequencies to

zero, we can adjust our prior to reflect our strong belief that high-frequencies should have little energy; we simply set the prior mean of these coefficients to zero and decrease their prior variance. If we now estimate the STRF using the maximum of the posterior then the amplitudes of high-frequencies will be biased by our prior towards zero. However, given sufficient evidence the MAP will yield non-zero estimates for the amplitudes of high-frequencies.

We chose to impose a hard-cutoff because we wanted to reduce the dimensionality to make online estimation of the model and online optimization of the stimuli more tractable. To pick the cutoff frequencies, we picked a single neuron and estimated the STRF using maximum-likelihood for a variety of cutoff frequencies. We evaluated the quality of each model by computing the log-likelihood of the bird's responses to inputs in a test set. The test set consisted of one bird song and one ml-noise stimulus which were not used to train the models. Table 2 lists the log-likelihoods on the test set for the different models. The results clearly show that setting the cutoff frequencies too high led to over-fitting. Setting the cutoff frequencies too low also decreased the predictive accuracy of the model. Based on these results we chose the cutoff frequencies to be $n_{fc} = 10$ and $n_{tc} = 4$ because these values provided good predictive performance for both the bird song and ml-noise while keeping the number of unknown parameters tractable (in this case the STRF has 189 unknown parameters).

Table 2 also shows that the log-likelihood for ml-noise is much higher than for the bird song. One explanation for this is the fact that the neuron spikes much less in response to ml-noise, Figure 22. In some sense predicting silence is much easier than predicting spikes. Even if the GLM accurately predicates elevated firing rates during the intervals in which the neuron seems to be responding to the bird song, the model will still likely get the actual timing of the spikes wrong which will decrease the likelihood of the data.

Table 2: As described in the text, we used cross-validation to determine the best values for the cutoff frequencies in our model. This table lists the log-likelihood, up to an additive constant, computed on the test set for models with different cutoff frequencies. a) The stimulus is bird song. b) The stimulus is ml-noise.

n_{fc} \ n_{tc}	1	4	7	9
2	-0.754	-0.688	-0.693	-0.692
4	-0.706	-0.686	-0.692	-0.692
10	-0.734	**-0.729**	-0.735	-0.74
20	-0.742	-0.738	-0.744	-0.751
39	-0.737	-0.792	-0.814	-0.833

(a)

n_{fc} \ n_{tc}	1	4	7	9
2	-0.538	-0.454	-0.458	-0.458
4	-0.458	-0.424	-0.432	-0.433
10	-0.477	**-0.431**	-0.438	-0.442
20	-0.5	-0.451	-0.463	-0.464
39	-0.502	-0.47	-0.473	-0.484

(b)

Table 3: To compare how well the smoothed and unsmoothed STRFs in Figure 23 and Figure 24 fitted the neuron we computed the expected log-likelihood of the responses in a test set. The test set consisted of one bird song and one ml noise stimulus. a) The log-likelihood for the models shown in Figure 23. In this case the smoothed STRF leads to better fits on both stimuli in the test set. b) The log-likelihood for the models shown in Figure 24. In this case the smoothed model does better on the bird song stimulus but slightly worse on ML noise.

	Bird Song	ML Noise
unsmoothed	-1.41	-.72
smoothed	**-1.36**	**-.70**

(a)

	Bird Song	ML Noise
unsmoothed	-2.04	**-2.53**
smoothed	**-1.98**	-2.56

(b)

In Figure 23 and Figure 24 we compare the STRFs and spike history components estimated with $n_{fc} = 10$ and $n_{tc} = 4$ to the estimated parameters without any smoothing for two different neurons. The STRFs estimated using the cutoff frequencies were much less noisy but had some oscillatory artifacts. To measure how well each model fit the data we computed the log-likelihood of the responses in a test set to the fitted model; the results are in Table 3. The results show that for the first neuron, the smoothed STRF did better than the un-smoothed STRF. For the second neuron, the smoothed STRF did better on the bird song stimulus but worse on ML noise then the un-smoothed STRF. We also evaluated the model by comparing predicted raster plots to actual raster plots as in Figure 22. In general, the model did a good job of predicting the peaks and valleys in the neuron's firing; in particular we could easily classify the stimulus as being bird song or ml-noise just by looking at the predicted raster plots.

We can compare these STRFs to previously published results (e.g Figure 7 in [168]) which estimated the receptive field using reverse correlation methods. A key difference in our methods is that the authors of [168] trained separate STRFs on the ml-noise and bird song stimuli. In contrast, we trained a single model on both types of inputs. Our STRFs are very similar in structure to the STRFs fitted to ml-noise in [168]; both STRFs have narrow frequency tuning and similar temporal structures. This result is also consistent with Table 2 which shows the model predicts a higher likelihood for responses to ml-noise then responses to bird song. When the authors of [168] trained the STRF on bird song alone, the resulting STRF had much broader frequency tuning. Since ml-noise shows much less correlation among the power at different frequencies than bird song, it is not surprising that including the ml-noise would lead to much narrower frequency tuning then training on bird song alone. We trained the GLM on both sets of input because we wanted the stimuli to span the input space as much possible. This is important in the next section when we re-sample

the data in an effort to compare an optimized experimental design to a non-optimized experimental design.

Our results show that fitting the canonical Poisson model to auditory data leads to reasonable estimates of the STRF of auditory neurons which are consistent with previously published results [167, 168]. Based on these results, we can justify the use of a GLM to optimize the stimuli during experiments. Regardless of whether the GLM is the best model, the results in this section support the expectation that a GLM will fit the data well enough that our methods will be able to design more informative experiments than the status quo.

4.3 Simulating sequential optimal experimental design using real data.

Having shown that the canonical Poisson model fits auditory data well enough to use the GLM to optimize data collection, we can now turn our attention to using this data to evaluate our methods for stimulus optimization. In previous chapters we tested our methods using simulations in which we generated synthetic responses using a GLM to model real neurons. Since real neurons are not GLMs, the simulations in the previous chapter provide little insight into how our methods will perform with actual neurons. We could try to improve our previous simulations by using a more biophysical model of a neuron. However, this approach is necessarily limited by the quality of the model used to generate the synthetic responses. A better approach is to use actual data from a real neuron to evaluate our methods offline.

In this section we describe a set of simulations in which the response to a stimulus is the actual response of a neuron to that stimulus. In the previous section we discussed how the GLM may be fitted to real data using maximum likelihood by dividing the recordings into stimulus-response pairs $(\vec{s}_t, r_t)$. The set of all input-response pairs gives us a set of inputs $\mathcal{S}$ for which we observed the actual responses of a neuron. Thus, if we run simulated experiments in which we restrict the inputs to inputs in $\mathcal{S}$

then for the responses to these inputs we may use the neuron's actual responses as opposed to generating synthetic responses using some model. We can therefore repeat our previous simulations in which we compare an information maximizing design to a random, non-optimized design, only now we use the actual responses of a neuron instead of generating synthetic responses.

When optimizing the stimuli for auditory experiments, we do not want to use the greedy method presented in Chapter 2 because it will be suboptimal. The input corresponding to r_t is a sound played over the previous t_k units of time. Thus each stimulus-response pair $(\vec{s}_t, r_t)$ actually requires t_k units of time to obtain. While this stimulus is played to the neuron, the response of the neuron is continuously recorded which yields observations $\{r_{t-t_k}, r_{t-t_k+1}, \ldots, r_{t-1}\}$. For each of these responses the corresponding input $\{\vec{s}_{t-t_k}, \ldots, \vec{s}_{t-1}\}$ is known; these inputs are just a combination of the sound played before time $t - t_k$ and the sound presented from time $t - t_k$ to t. We do not want to throw these observations out because they contain valuable information. Consequently, we care about the total information contained in the sequence $\{(\vec{s}_{t-t_k}, r_{t-t_k}) \ldots (\vec{s}_t, r_t)\}$ and not just the information contained in the last observation $(\vec{s}_t, r_t)$. This is particularly important when comparing two designs because the design which produces the largest information on the last trial may not be the design for which the sequence of trials, $\{(\vec{s}_{t-t_k}, r_{t-t_k}) \ldots (\vec{s}_t, r_t)\}$, is most informative.

Consequently when optimizing the design we do not want to use the greedy algorithm developed in Chapter 2 because this algorithm would ignore the information in $\{(\vec{s}_{t-t_k}, r_{t-t_k}) \ldots (\vec{s}_{t-1}, r_{t-1})\}$ and just maximize the information in the final trial $(\vec{s}_t, r_t)$ in the batch. Therefore, we consider the problem of picking an optimal sequence of stimuli $\{\vec{x}_{t+1}, \ldots, \vec{x}_{t+b}\}$ to be presented on the next b trials. If the stimulus is a temporal signal, e.g. a sound, then these inputs are obviously not independent.

4.3.1 Finding an optimal sequence of stimuli.

In this section we consider the problem of finding the optimal sequence of stimuli $\{\vec{x}_{t+1}, \ldots, \vec{x}_{t+b}\}$ at time t. As described in the previous section each $\vec{x}$ is a column of the spectrogram of the input. Thus the sequence $\{\vec{x}_{t+1}, \ldots, \vec{x}_{t+b}\}$ gives a spectrogram which can be inverted to compute the actual sound that should be played to the bird [3]. We show that maximizing the mutual information between the responses on the next b trials and $\vec{\theta}$ leads to a very similar objective function to the objective function presented in Chapter 2. We then show how we can easily derive a lower bound for the mutual information which we can optimize using the methods presented in Chapter 2.

To derive our objective function we follow a nearly identical derivation to that presented in Chapter 2. As before, we can write the mutual information

$$I(\vec{\theta}; \boldsymbol{r_{t+1:t+b}} | \boldsymbol{s_{1:t+b}}, \boldsymbol{r_{1:t}})$$

as the difference between the entropy of our posterior at time t and the entropy of our posterior at time $t+b$.

$$I(\vec{\theta}; \boldsymbol{r_{t+1:t+b}} | \boldsymbol{s_{1:t+b}}, \boldsymbol{r_{1:t}}) = H(p(\vec{\theta} | \boldsymbol{s_{1:t}}, \boldsymbol{r_{1:t}})) - E_{\boldsymbol{r}_{t+1:t+b} | \boldsymbol{s}_{t+1:t+b}, \vec{\theta}} H(p(\vec{\theta} | \boldsymbol{s_{1:t+b}}, \boldsymbol{r_{1:t+b}})). \tag{352}$$

Using our Gaussian approximation of the posterior, we can easily compute the mutual information,

$$\begin{aligned} &I(\vec{\theta}; \boldsymbol{r_{t+1:t+b}} | \boldsymbol{s_{1:t+b}}, \boldsymbol{r_{1:t}}) \\ &\quad \propto E_{\vec{\theta}} E_{\boldsymbol{r}_{t+1:t+b} | \boldsymbol{s}_{t+1:t+b}, \vec{\theta}} \log \left| \boldsymbol{C}_t^{-1} + \sum_{i=1}^{b} J_{obs}(r_{t+i}, \vec{s}_{t+i}^T \vec{\theta}) \vec{s}_{t+i} \vec{s}_{t+i}^T \right| + const. \end{aligned} \tag{353}$$

The mutual information is just the sum of what we already know at time t (i.e $\boldsymbol{C}_t^{-1}$) and the information in the sequence of observations.

[3] To ensure the spectrogram is invertible certain restrictions must be enforced on how the spectrogram is computed (see e.g [151]). For example, since we do not specify the phase of each frequency band there must be sufficient overlap in the frequency bands to allow the phase to be recovered.

Unfortunately we cannot use the methods presented in Chapter 2 to optimize Eqn. 353. In Chapter 2 we used the fact that the Fisher information only depends on $\rho_t = \vec{s}_{t+1}^T\vec{\theta}$ to show that the mutual information lived on the 2-d space ($\mu_\rho = \vec{s}^T\vec{\mu}_t, \sigma_\rho^2 = \vec{s}^T\boldsymbol{C}_t\vec{s}$). More generally, for a sequence of length b we can use the 1-d property of the Fisher information to show that the mutual information is a function of

$$\mu_\rho = \vec{\mu}_t^T[\vec{s}_{t+1}, \ldots, \vec{s}_{t+b}] \tag{354}$$

$$\sigma_\rho^2 = [\vec{s}_{t+1}, \ldots, \vec{s}_{t+b}]^T\boldsymbol{C}_t[\vec{s}_{t+1}, \ldots, \vec{s}_{t+b}]. \tag{355}$$

Here σ_ρ^2 is a $b \times b$ matrix and μ_ρ is a vector of length b. If $b < \dim(\vec{\theta})$ then computing the mutual information as a function of $(\mu_\rho, \sigma_\rho^2)$ significantly reduces the dimensionality of the problem. Unfortunately, unless b is small the dimensionality will still be too large to easily maximize the mutual information as a function of $(\mu_\rho, \sigma_\rho^2)$.

The inclusion of spike-history dependence in the model makes evaluating Eqn. 353 even harder because we no longer have full control over future inputs $\vec{s}_{t+i}$. In this case $\vec{s}_{t+i}$ depends on the unknown responses on the trials preceding $t+i$. One way to handle this complication is by focusing on the mutual information between the sequence of observations and just the stimulus coefficients. This turns out to be equivalent to minimizing the posterior entropy of just the stimulus coefficients. Our objective function in this case is

$$I(\vec{\theta}_x; \boldsymbol{r_{t+1:t+b}} | \boldsymbol{s_{1:t+b}}, \boldsymbol{r_{1:t}})$$
$$\propto E_{\vec{\theta}} E_{\boldsymbol{r_{t+1:t+b}}|\boldsymbol{s_{t+1:b}},\vec{\theta}} \log \left| \boldsymbol{C}_{x,t}^{-1} + \sum_{i=1}^{b} J_{obs}(r_{t+i}, \vec{s}_{t+i}^T\vec{\theta})\vec{s}_{x,t+i}\vec{s}_{x,t+i}^T \right| + const. \tag{356}$$

Here the subscript x means we are taking the elements of $\vec{\theta}$ and $\vec{s}$ which correspond to the stimulus coefficients. The Fisher information, however, still depends on the responses preceding $t+i$. One way to handle this is to simply use a point estimate, e.g. the background firing rate, for the components of $\vec{s}_{t+i}$ corresponding to the unknown

responses on trials preceding $t+i$. For the rest of this chapter we assume that when the neuron depends on past responses we only maximize the information about the stimulus coefficients and we compute the Fisher information $J_{obs}(r_{t+i}, \vec{s}_{t+i})$ by using a point approximation of all responses prior to $t+i$.

To derive a tractable optimization problem we focus on maximizing a lower bound of Eqn. 353 which we obtain by using Jensen's inequality

$$I(\vec{\theta}_x; \boldsymbol{r_{t+1:t+b}} | \boldsymbol{s_{1:t+b}}, \boldsymbol{r_{1:t}})$$

$$\propto E_{\vec{\theta}} E_{\boldsymbol{r_{t+1:t+b}}|\boldsymbol{s_{t+1:t+b}},\vec{\theta}} \log |\sum_{i=1}^{b} \frac{1}{b} \boldsymbol{C}_{x,t}^{-1} + \frac{b}{b} J_{obs}(r_{t+i}, \vec{s}_{t+i}^T \vec{\theta}) \vec{s}_{x,t+i} \vec{s}_{x,t+i}^T| \tag{357}$$

$$\geq E_{\vec{\theta}} E_{\boldsymbol{r_{t+1:t+b}}|\boldsymbol{s_{t+1:t+b}},\vec{\theta}} \sum_{i=1}^{b} \frac{1}{b} \log |\boldsymbol{C}_{x,t}^{-1} + b J_{obs}(r_{t+i}, \vec{s}_{t+i}^T \vec{\theta}) \vec{s}_{x,t+i} \vec{s}_{x,t+i}^T| \tag{358}$$

$$= E_{\vec{\theta}} E_{\boldsymbol{r_{t+1:t+b}}|\boldsymbol{s_{t+1:t+b}},\vec{\theta}} \sum_{i=1}^{b} \frac{1}{b} \left(\log |\boldsymbol{C}_{x,t}^{-1}| + \log(1 + b J_{obs}(r_{t+i}, \vec{s}_{t+i}^T \vec{\theta}) \vec{s}_{t+i}^T \boldsymbol{C}_t \vec{s}_{t+i}) \right) \tag{359}$$

$$= \log |\boldsymbol{C}_{x,t}^{-1}| + \sum_{i=1}^{b} \frac{1}{b} E_{\vec{\theta}} E_{\boldsymbol{r_{t+1:t+b}}|\boldsymbol{s_{t+1:t+b}},\vec{\theta}} \log \left(1 + b J_{obs}(r_{t+i}, \vec{s}_{t+i}^T \vec{\theta}) \vec{s}_{x,t+i}^T \boldsymbol{C}_{x,t} \vec{s}_{x,t+i}) \right). \tag{360}$$

As noted before, in general we cannot factor the joint distribution $p(\boldsymbol{r_{t+1:t+b}}|\boldsymbol{s_{t+1:t+b}}, \vec{\theta})$ because spike history means $\vec{s}_{t+i}$ depends on r_{t+j} for $j < i$. However as discussed earlier we can use a point approximation for past responses to compute $\vec{s}_{t+i}$. Using this approximation we can approximate the conditional distribution by factoring it across responses,

$$p(\boldsymbol{r_{t+1:t+b}}) \approx \prod_{i=1}^{b} p(r_{t+i}|\vec{s}_{t+i}, \vec{\theta}). \tag{361}$$

Plugging this approximation into our lower bound yields

$$I(\vec{\theta}_x;\boldsymbol{r_{t+1:t+b}}|\boldsymbol{s_{1:t+b}},\boldsymbol{r_{1:t}}) \geq \log|\boldsymbol{C}_{x,t}^{-1}| + \sum_{i=1}^{b}\frac{1}{b}E_{\vec{\theta}}E_{r_{t+i}|\vec{s}_{t+i},\vec{\theta}}\log\left(1+bJ_{obs}(r_{t+i},\vec{s}_{t+i}^{T}\vec{\theta})\vec{s}_{x,t+i}^{T}\boldsymbol{C}_{x,t}\vec{s}_{x,t+i})\right). \tag{362}$$

Each term in the summation over i depends on a single stimulus-response pair.

To evaluate this lower bound we just need to evaluate

$$E_{\vec{\theta}}E_{r_{t+i}|\vec{s}_{t+i},\vec{\theta}}\log\left(1+bJ_{obs}(r_{t+i},\vec{s}_{t+i}^{T}\vec{\theta})\vec{s}_{x,t+i}^{T}\boldsymbol{C}_{x,t}\vec{s}_{x,t+i}\right)$$

for each stimulus in our batch. This expression is equivalent to our objective function in the greedy case, except J_{obs} is scaled by b. As before, we can replace the expectation over θ with an expectation over the scalar $\rho_i = \vec{s}_{t+i}^{T}\vec{\theta}$. For the canonical Poisson, we can drop the expectation over r_{t+i} because the Fisher information is independent of the responses. Thus for the canonical Poisson we need to maximize

$$I(\vec{\theta}_x;\boldsymbol{r_{t+1:t+b}}|\boldsymbol{s_{1:t+b}}) \propto \log|\boldsymbol{C}_{x,t}^{-1}| + \sum_{i=1}^{b}\frac{1}{b}E_{\rho}\log\left(1+bJ_{obs}(\vec{s}_{t+i}^{T}\vec{\theta})\vec{s}_{x,t+i}^{T}\boldsymbol{C}_{t}\vec{s}_{x,t+i})\right). \tag{363}$$

In Chapter 2 we used the approximation $\log(1+x) \approx x$ to further simplify this approximation. In this chapter we avoid this approximation because we found that it performed quite poorly when $b > 1$. Thus in this chapter we compute the expectations with respect to ρ numerically.

Using this lower bound we can reshuffle the bird song data in a more realistic manner. First we pick a size, b, for the batch. Then we slide a window of duration b over the responses. For each response in this window, r_t, we extract the corresponding stimulus, $\vec{s}_t$. This yields a batch of inputs, $\boldsymbol{s_{t:t+b-1}} = \{\vec{s}_t,\ldots,\vec{s}_{t+b-1}\}$. We compute the lower bound of the informativeness of all such batches and then select the batch with the largest value. We then update our posterior using all responses $\{r_t,\ldots,r_{t+b-1}\}$ to the inputs in the batch.

Since we pick the inputs without replacement, a subset of the responses gets thrown out. For example, suppose b corresponds to a 100ms time window and that on trials $1 : b$ we use the responses to the first 100ms of a wave-file and on trials $b+1 : 2b$ we use the same wave-file but beginning at 150ms after the start of the wave file. This leaves a 50ms gap between the portions of the wave file used for the first and second batch. Since this gap is less than the 100ms needed per batch, this gap will never be selected because doing so would require reusing responses which had been included as part of the first or second batches of trials.

4.4 Results: simulated experiments using real data

Using the methods presented in the previous section, we ran simulations using actual responses of zebra finch to compare the use of an info. max. design to a random non-optimized design. The data consisted of the recordings of 11 neurons obtained by our collaborators David Schneider and Dr. Sarah Woolley. Every b trials, we selected the inputs corresponding to b sequential observations in the data obtained from one of the neurons. The info. max. design picked this batch of stimuli by maximizing the lower bound for the mutual information described in the previous section. In contrast the random design randomly picked one of the batches containing stimuli which had not been picked yet; hence we call this a shuffled design. After the input was selected, the responses were obtained simply by selecting the actual responses of the neuron to this batch of stimuli. In between successive batches of b stimuli there is a window of duration t_k in which the stimuli is known but the responses are not. This gap corresponds to the time required to play the segment of the wave-file which serves as the input for the first response, r_{t+1}, in the batch.

We compare the designs by evaluating how well the GLM fits the data as a function of the number of trials used to train the GLM. To quantify how well the trained model fits the data we used the model to compute the expected log-likelihood of

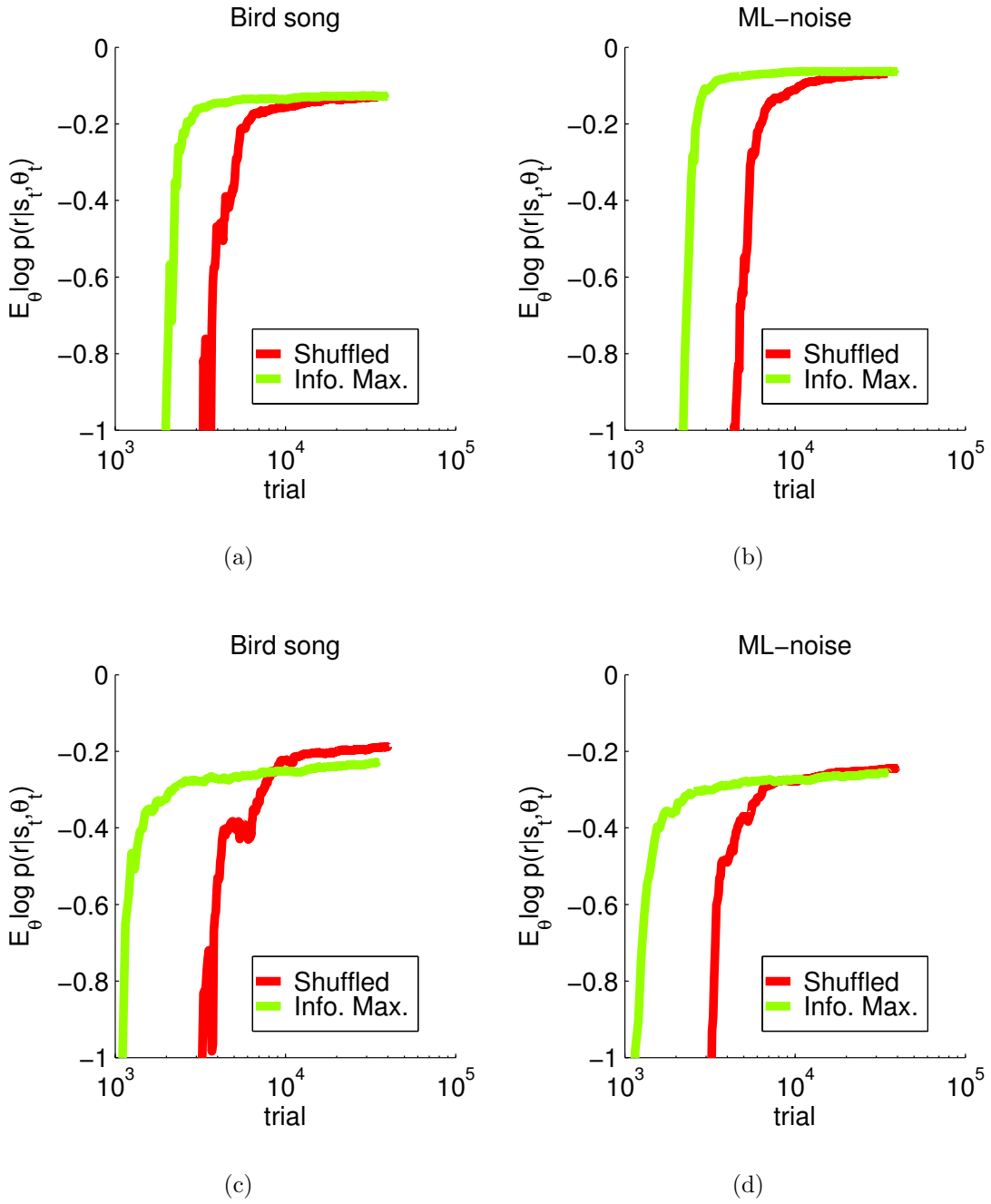


Figure 25: Each row shows the expected log-likelihood, up to a normalization constant, computed on the test sets for a different neuron. The test set for each neuron consisted of one bird song and one ml-noise stimulus. The expected log-likelihood is plotted as a function of the number of trials used to train a model using inputs chosen by either an info. max. or shuffled design as described in the text. The results clearly show that the info. max. design achieves a higher level of prediction accuracy using fewer trials. We quantify the improvement as the speedup, see Figure 27.

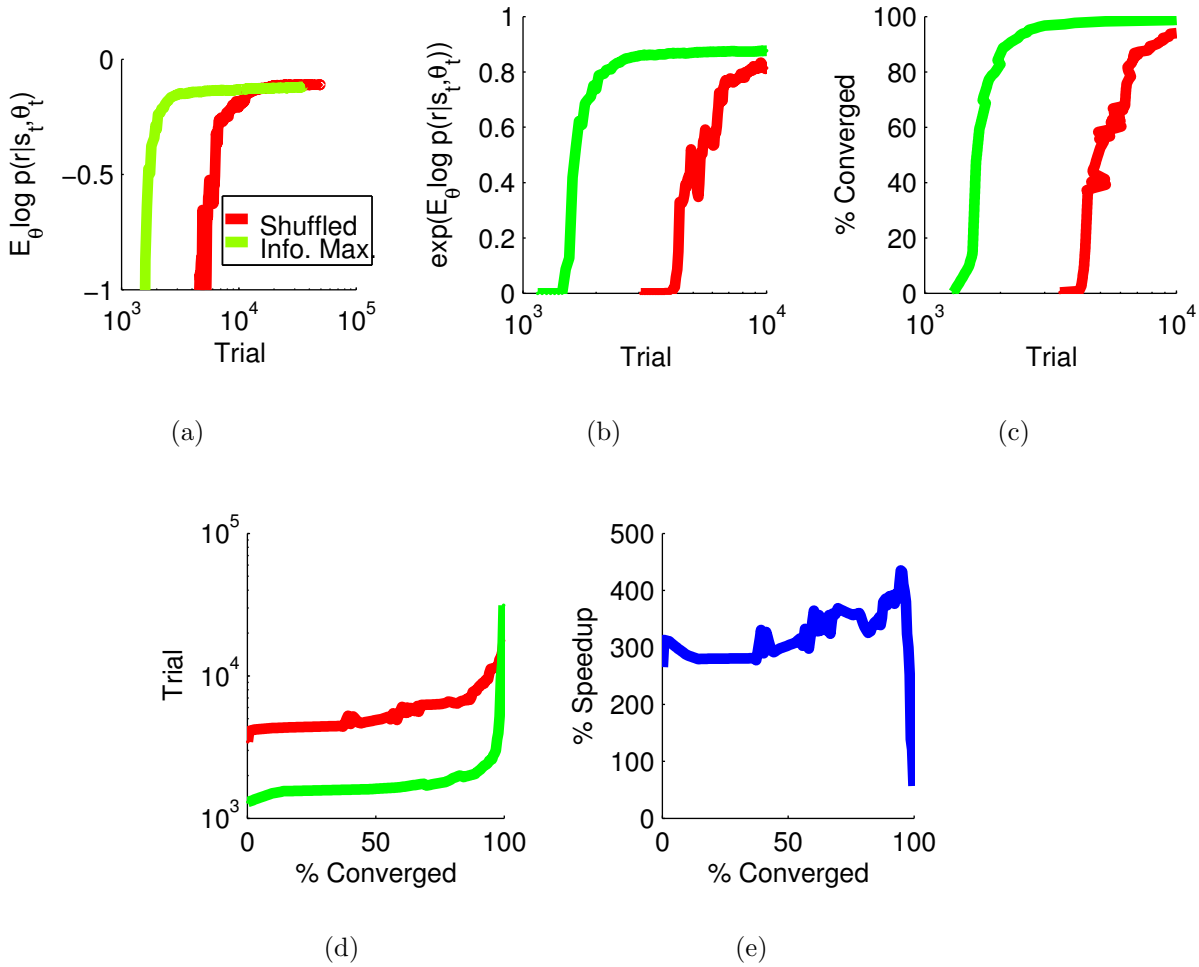


Figure 26: A sequence of plots illustrating how we compute the speedup. a) We start with a plot of the expected log-likelihood as in Figure 25. b) We convert the y-axis from the log domain into the linear domain. c) We rescale the y-axis so that it varies between 0 and 100%. This gives a plot of how close the model is to the best model as a function of the trial. d) We flip the x and y axes. This gives a plot of the number of trials needed as a function of the model's quality as measured by % Converged. For any value of % Converged the distance between the two curves measures how many more trials are needed by the shuffled design. e) We compute the Speedup as a function of % Converged by computing the ratio of the two curves in (d) for each value of % Converged.

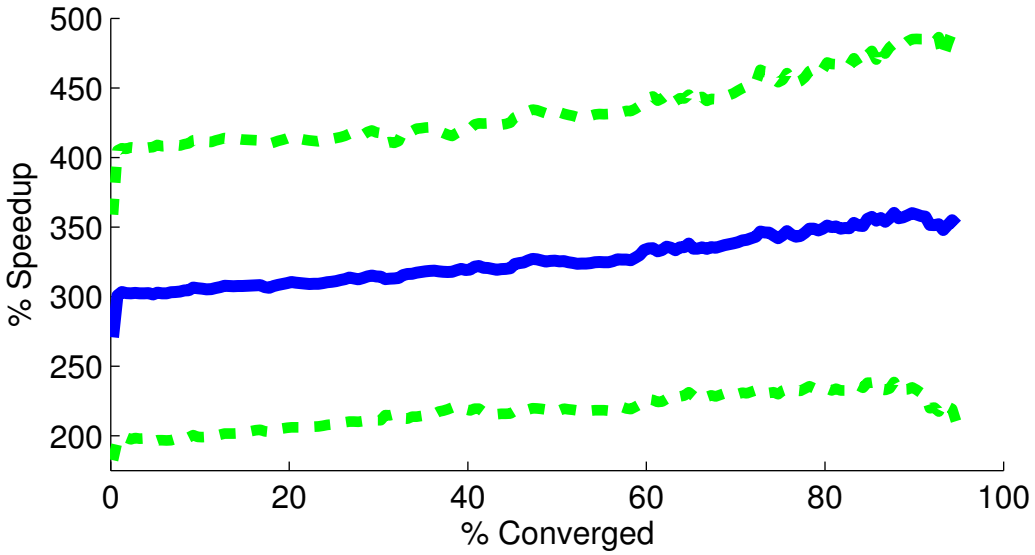


Figure 27: A plot of the speedup achieved by using the info. max. design instead of a shuffled design. The speedup is plotted as a function of % Converged as described in the text. The solid blue line shows the average speedup across all 11 neurons and the dashed green lines show plus and minus one standard deviation. The results show that using a shuffled design would require roughly 3 times as many trials to achieve the same level of prediction accuracy as the info. max. design. The speedup is not computed for values of % Converged $> 95\%$ because the Speedup cannot be accurately computed for these values. To compute the Speedup we need to compute the trial on which the curves in Figure 25 have some particular value for the y-coordinate. Values of % Converged $> 95\%$ correspond to y-values close to the flat part of the curves. Thus, for % Converged $> 95\%$ we cannot accurately measure the trial on which a particular value of % Converged was reached.

responses to wave files which were not included in the training set; i.e. these stimulus-response pairs could not be picked by either the info. max. or shuffled designs. The expected log-likelihood provides a metric for measuring how well the model predicts the responses to novel stimuli. We compute the expected log-likelihood with respect to our Gaussian posterior on $\vec{\theta}$. We compute the expected value rather than simply using a point estimate of $\vec{\theta}$ to compute the log-likelihood because we want to take into account our uncertainty about $\vec{\theta}$. The info. max. design minimizes the entropy of our posterior; thus if we use a point estimate we are ignoring the quantity which our method is trying to optimize. The expected log-likelihood for a file in the test set is thus

$$Q(t) = \frac{1}{T} \sum_{i=1}^{T} E_{\vec{\theta}|\vec{\mu}_t, \boldsymbol{C}_t} \log p(r_i|\vec{s}_i). \tag{364}$$

Each term in the summation over i corresponds to a different stimulus-response pair obtained from the test set. We sum the log-likelihoods of each stimulus-response pair because we want to compute the log-likelihood of the joint distribution on all such pairs. We normalize by T, the number of stimulus-response pairs so we can compare the expected log-likelihood for wave files of different lengths.

The test set consisted of the responses to one bird song and one ml-noise stimulus. The results are shown in Figure 25 for two different neurons. The results clearly show that using an info. max. design reduces the number of trials needed to train the model. Furthermore, in all cases the expected log-likelihood appears to be converging as a function of the trials to a value which is nearly the same for both designs. Thus, given enough trials both designs produce GLMs which fit the data equally well. This is not surprising because the training sets for both designs are necessarily the same if we train on all the data.

For both designs the curves tend to level off as the number of trials increases for two reasons. First, the average information per trial, as measured by the Fisher information, is constant. However, the amount of information in our posterior is

increasing with each trial. Thus, the amount of new information in each additional trial as a percent of our total information tends to decrease; i.e. the return on each additional trial is diminishing [9]. Second, as noted earlier there are only approximately 20000 distinct stimuli each of which is repeated 10 times. Naturally, the amount of information from a given stimulus should decrease with each repetition of that stimulus.

In Figure 25 the log-likelihoods for the test sets for the first neuron (top row) are generally higher than for the second neuron (bottom row). Naturally, we would expect the GLM to fit some neurons better than others and this should be reflected in the log-likelihood. Unfortunately, this makes it difficult to evaluate the average performance of the info. max. design. In particular, to compare the info. max. design to the shuffled design, we would like to measure how many more trials a shuffled design needs to produce a model which is as good as the model estimated by the info. max. design. For this purpose, we cannot simply use the value of the expected log-likelihood to measure the quality of the fit because this will vary across neurons independent of the design.

To facilitate comparisons across neurons, we define a quantity which we call % Converged which quantifies the quality of a GLM fitted to a particular neuron relative to the best possible GLM for that neuron. % Converged is nothing more than a nonlinear rescaling of the $y - axis$ in Figure 25 such that after the rescaling all curves are bounded between 0 and 100, see Figure 26. At any time t during the simulation, we define %Converged as

$$\%\text{Converged} = \frac{\exp(Q(t))}{\exp(Q(\infty))} \times 100 \tag{365}$$

$$= \omega(t). \tag{366}$$

Here $Q(\infty)$ is the expected log-likelihood evaluated on the test set for the fully converged model; i.e it is the value to which the traces in Figure 25 converge and is

independent of the design. $Q(\infty)$ therefore measures how well the best fit GLM, one which was trained using all the data, would predict the responses in the test set. % Converged therefore gives a metric for measuring how close a model trained using t trials is to the best model for one particular neuron. % Converged is just a rescaling of the y-axis in Figure 25 in which we first raise the y-coordinates to *exp* and then rescale the resulting values so that all values are between 0 and 100, see Figure 26. % Converged therefore provides a mechanism for evaluating model fit in a way that controls for the fact that no model can predict with 100% accuracy the neuron's responses. In this sense, % Converged is similar to other metrics such as "potentially explainable variance", used to evaluate the quality of neural models [19].

Using % Converged, we can compare the info. max. designs and shuffled designs, by measuring how many more trials the shuffled design requires to produce an equally well fit model as measured by % Converged. We define speedup as

$$\text{Speedup} = \frac{t_{\text{shuffled}}(\omega_1)}{t_{\text{info. max.}}(\omega_1)} \tag{367}$$

Here $t_{\text{shuffled}}(\omega_1)$ and $t_{\text{info. max.}}(\omega_1)$ measure the number of trials required for a shuffled design and an info. max. design respectively to produce an estimated model with the desired value for ω; i.e t s.t $\omega(t) = \omega_1$. The speedup measures the distance between the two traces along the x-axis in Figure 25 at a particular value of the y-axis. Speedup depends only on the performance of the info. max. vs. shuffled designs and not how well the GLM fits a particularly neuron. Figure 26 presents a series of figures illustrating how speedup is computed.

We computed the speedup for each neuron as a function of % Converged. In Figure 27 we plot the average and standard deviation of the Speedup as a function of % Converged over all neurons. The results show that on average the shuffled design required three times as many trials to produce a model that fit the data as well as a model trained using the information maximizing design. This is a large enough improvement to potentially warrant the effort required to implement stimulus

optimization in an actual experiment. Furthermore, we expect the results in Figure 27 to underestimate the potential improvement in actual experiments because in our simulations the info. max. design could only pick inputs which were actually presented to the birds, which was necessarily the same set of inputs as used in the shuffled design. We know from Chapter 2 that if we select the input from a finite set, choosing a bad set of stimuli can severely limit the ability of an info. max. design to outperform a random design. Therefore, an info. max. design which could pick any input could potentially do much better.

4.5 Discussion

In this chapter we have shown using real data, that using the canonical Poisson model we could potentially reduce the amount of data needed to estimate the STRF of auditory neurons in zebra finch by a factor of 3. While our efforts in this chapter focused on using the canonical Poisson model, this may not be the best GLM for modeling auditory neurons. The Poisson model is a firing rate model since it provides a distribution on the number of spikes expected in some appropriate time window. In our case, that time window is usually small enough ~ 2.5ms, that the neuron almost never fires more than once in each observation window. Given the neuron's refractory period the neuron would have to fire at an uncharacteristically high rate ($> 400Hz$), in order to spike more than once in each observation window. As a result, we might reasonably expect that the Poisson distribution with exponential nonlinearity provides a poor model of the neuron's response. In particular, the exponential model fails to model the fact that the response saturates due to the refractory period. We cannot simply substitute a saturating nonlinearity like a sigmoid function for the exponential function because the log-likelihood would no longer be concave and our methods are highly dependent on concavity of the log-likelihood for efficiently optimizing the experimental design.

We could, however, potentially construct a better spike-time model by using a Bernoulli distribution with a sigmoid nonlinearity; i.e the logistic model. This produces a spike-time model because the conditional probability gives the probability that the neuron spikes in some small time window. As a result, for this model the maximum predicted firing rate is determined by the size of the observation window. The log-likelihood for the logistic model is concave so most of our methods can be applied with minimal modifications [89].

A key difference between the logistic and canonical Poisson model is that maximizing the Fisher information leads to very different experimental designs. For the canonical Poisson, the Fisher information increases with the expected firing rate. As a result, to increase the informativeness of the experiments we want to drive the neuron to fire as much as possible. In contrast, for the logistic model the Fisher information saturates for both low and high firing rates because the probability of a spike saturates at either zero or one. To increase the information about the logistic model, we want to pick an input for which the probability of the spike is close to 50%. Consequently, we would expect that using the logistic-model would lead to a very different information maximizing design then the one obtained using the canonical Poisson model.

Despite the fact that the Poisson model does not impose a limit on the firing rate, we were still able to achieve a 300% speedup compared to the shuffled design. We hypothesis that this is because the Poisson model does in an approximate sense model the threshold nonlinearity of a neuron. Since we can only detect spikes and not sub-threshold changes in membrane voltage, very little information is obtained about the receptive field when the neuron does not fire; i.e. all we know is that the neuron did not fire, we do not know whether the neuron was close to firing or even if it was depolarized or hyper-polarized. The exponential function provides a very rough approximation for a threshold nonlinearity because for $\rho < 0$, the expected firing rate, $\exp(\rho)$, is relatively flat and close to zero. Thus, the Poisson model leads us to pick

inputs which will drive the neuron to fire which is consistent with our intuitive notion of the optimal inputs.

To simulate experiments using the bird song data, we had to consider the problem of non-greedy optimization in Section 4.3.1. We can compare the methods presented here to the methods presented in Chapter 3. In Chapter 3 we solved the non-greedy optimization problem by considering the limit of maximizing the mutual information as $b \rightarrow \infty$. This led to a convex optimization problem for the optimal stimulus distribution which we solved by assuming the distribution was a Gaussian process. In actual experiments this approach might work and might be easier to implement then the approach described in Section 4.3.1. Unfortunately, we could not use this approach to simulate experiments using the bird song data. The approach in Chapter 3 would not work because that approach generates stimuli by sampling the optimal distribution. If we simply sample a Gaussian process it is unlikely that we will pick one of the inputs which was actually presented to the bird and thus one for which the response is known. In contrast, the approach developed in this section considered a finite b and then established a lower bound which we could optimize with relative easy. The main difference in the resulting objective functions is that the objective function in Eqn. 353 explicitly depends on the prior; i.e it is the information in our prior plus the information in the observations. In comparison, the objective function in Chapter 3 was the average information per trial; in that case the prior only mattered because we used the posterior on $\vec{\theta}$ to compute the expected information of each trial.

In this chapter we have shown that our methods are robust enough to work with real data. We have shown that fitting the GLM to the responses of zebra finch leads to estimated STRFs which are very similar to those estimated using reverse correlation techniques. Furthermore, we have shown using real data that using an information maximizing design could offer a factor of 2-4 speedup over a typical, non-optimized

experimental design. Thus, even though our methods are approximate and do not necessarily use the best model of auditory neurons, they can still be used to collect more informative data. These results are strong enough to warrant further efforts to actually implement our methods in experiments.

CHAPTER V

USING PRIOR INFORMATION TO DESIGN OPTIMAL NEUROPHYSIOLOGY EXPERIMENTS.

Early in an experiment when little data is available, trials should be optimized using all available prior knowledge. The methods in the previous chapters, however, have incorporated only weak prior information about the underlying neural system due to the difficulty of computing the mutual information using plausible prior beliefs. Here we present methods for incorporating strong prior information about the receptive field. For example, if we believe that the receptive field is well-approximated by a Gabor function then our method constructs stimuli that optimally constrain the Gabor parameters (orientation, spatial frequency, etc.) using as few experimental trials as possible. More generally, we assume our prior knowledge specifies a sub-manifold of model space in which we expect the GLM's parameters to lie. This sub-manifold defines the expected structure of the receptive field; e.g. that the receptive is sensitive to dynamic stimuli. In light of our prior knowledge, we want to design experiments to reduce our uncertainty on the sub-manifold as rapidly as possible. To make the computations tractable we use the tangent space to approximate the sub-manifold. Applications to simulated and real data indicate that these methods may improve the efficiency of data collection in real experiments.

5.1 Introduction

When neurophysiologists begin experimenting with a new animal, brain region, or class of neuron, they often use knowledge gained from investigations in other animals or with other brain regions to guide their initial experiments. When Hubel and

Wiesel, for example, first began recording from neurons in V1 they used dots of light to stimulate the cat's visual system because earlier work had shown that neurons in the early visual system fire in response to simple dots [76]. An obvious question is "What should we do with this type of prior knowledge when trying to optimize our experiments?" Intuitively, the theory of Bayesian experimental design tells us that we should always use our prior knowledge because even if our prior knowledge is only approximately correct it can lead to huge gains in efficiency. On the other hand, if our prior knowledge is wrong, our design will in some sense still be nearly as good as an optimal design which ignored our prior knowledge. This conclusion follows from the simple fact that for nonlinear models no design is simultaneously optimal for all possible models.

Consider the following hypothetical example. Suppose armed with our knowledge of Hubel and Wiesel's results we set out to record from another region of visual cortex. If we make no assumptions about the neuron's receptive field then any stimulus is equally likely to drive the neuron to fire. Furthermore, since neurons have a strong threshold nonlinearity, only a small subset of all possible visual images will drive a particular neuron to fire. Thus, any input we might pick will generally be rather uninformative with regard to most of the possible models. On the other hand suppose based on Hubel and Wiesel's results we expect that the receptive field of the neurons we are recording from will resemble simple cells. In this case, the optimal design will pick stimuli containing bars oriented at different angles. Since these stimuli are highly informative for simple cells, a small number of trials will be sufficient to collect enough data to support or reject the belief that we are dealing with simple cells. If the cell turns out not to be a simple cell then we are no worse off then had we not used our prior beliefs to optimize these experiments; in particular, whether we ignore our prior information or not it is unlikely we will pick an image which will cause the neuron to fire. However, if we get lucky and the cell is a simple cell then we

will converge much more rapidly to the true model then had we ignored our prior beliefs. Even if our prior knowledge is only approximately correct, we can often design much more informative experiments. For example, suppose the neuron we are recording from is a complex cell so that it responds to bars but is invariant to the bar's position [75]. In this case, our prior belief that it is a simple will lead us to pick bar stimuli which will drive the neuron to fire quite efficiently even though our prior beliefs are incorrect. Furthermore, these stimuli will clearly reveal that the response is invariant to the bar's position which is a key property of complex cells. This example shows that there is very little downside to using our prior knowledge to guide initial experiments provided we are willing to reject our prior beliefs given sufficient evidence. In Hubel and Wiesel's case this meant accepting the fact that neurons in V1 do not respond to simple dots and switching to other stimuli. Since the brain exhibits an amazing amount of structure, we can often form prior beliefs about a neuron's response function which turn out to be approximately correct. As our example illustrates, incorporating these beliefs can lead to superior experimental designs.

In the introduction to this thesis, we argued that one of the key benefits of Bayesian optimal experimental design over traditional design criteria was that the Bayesian approach made it easy to incorporate prior knowledge. In fact, the methods presented in the previous chapters all incorporated an explicit prior on the parameters, $\vec{\theta}$, of the neuron's response function. In previous chapters, we used rather uninformative priors; i.e Gaussians distributions which were relatively flat so that all models were nearly equally likely. These flat priors do not encode the typical prior beliefs of neurophysiologists. For the generalized linear model, the model parameters correspond to the receptive field of the neuron. We can think of the receptive field as defining the features that a neuron detects; e.g simple cells detect bars and this is evident in $\vec{\theta}$. Since the brain is highly structured, e.g cortex can be divided into functional areas,

we can often infer a great deal about a neuron's receptive field simply by knowing a neuron's location. For example, we would expect a neuron in the MT region of visual cortex is more likely to respond to images containing dynamic features as opposed to static features [3, 131]. In principle, we can just represent this prior knowledge as a probability distribution on $\vec{\theta}$. Unfortunately, this straightforward approach is not feasible because it generally leads to complicated priors which make computing the expected utility of a design intractable.

In this chapter, we show how the methods presented in the previous chapters can be modified to use prior knowledge to design better experiments. We assume that our prior knowledge defines a sub-manifold in parameter space which we expect a-priori to contain the best parameters. After each trial we compute the optimal design by maximizing the mutual information which is a function of our posterior. Before we compute the mutual information, we regularize our posterior using our prior knowledge about the sub-manifold in which the parameters should lie. By regularizing the posterior before computing the mutual information, we are essentially finding the optimal design with respect to a smaller class of models. Since no design is simultaneously optimal for all models, it makes sense to initially focus on the models which we think are more likely a-priori. To make the computations tractable we regularize the posterior by using a linear approximation of the manifold. The resulting distribution is a Gaussian distribution. Consequently, we can compute the optimal design using the methods presented in Chapter 2 and Chapter 3.

5.2 Optimizing experiments using strong prior information about the sub-manifold containing the parameters.

Neurophysiologists often expect a-priori that a neuron will respond to certain features of the input. We can think of this prior knowledge as defining a low-dimensional subspace of all possible receptive fields. One way to specify this manifold is by assuming the model parameters, $\vec{\theta}$, have some low-dimensional parametric structure;

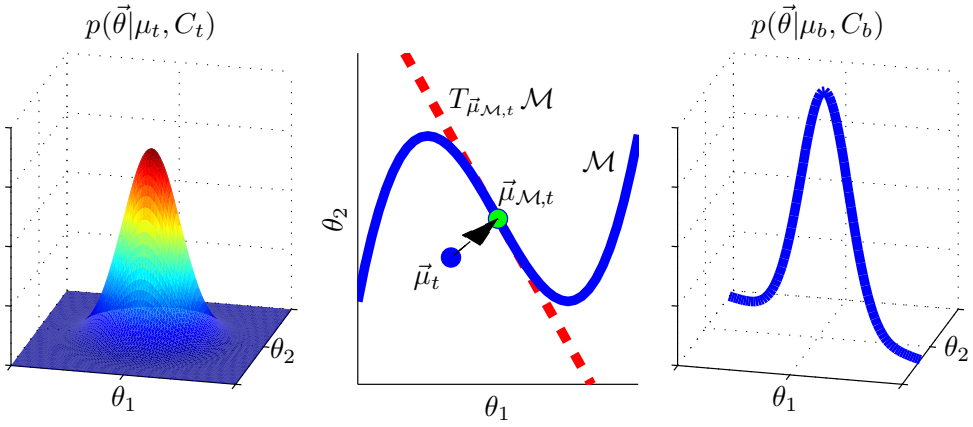


Figure 28: A schematic illustrating how we use the manifold to improve stimulus design. Our method begins with a Gaussian approximation of the posterior on the full model space after t trials, $p(\vec{\theta}|\vec{\mu}_t, \boldsymbol{C}_t)$. The left panel shows an example of this Gaussian distribution when $dim(\vec{\theta}) = 2$. The next step involves constructing the tangent space approximation of the manifold $\mathcal{M}$ on which $\vec{\theta}$ is believed to lie, as illustrated in the middle plot; $\mathcal{M}$ is indicated in blue. The MAP estimate (blue dot) is projected onto the manifold to obtain $\vec{\mu}_{\mathcal{M},t}$ (green dot). We then compute the tangent space (dashed red line) by taking the derivative of the manifold at $\vec{\mu}_{\mathcal{M},t}$. The tangent space is the space spanned by vectors in the direction parallel to $\mathcal{M}$ at $\vec{\mu}_{\mathcal{M},t}$. By definition, in the neighborhood of $\vec{\mu}_{\mathcal{M},t}$, moving along the manifold is roughly equivalent to moving along the tangent space. Thus, the tangent space provides a good local approximation of $\mathcal{M}$. In the right panel we compute $p(\vec{\theta}|\vec{\mu}_{b,t}, C_{b,t})$ by evaluating $p(\vec{\theta}|\vec{\mu}_t, \boldsymbol{C}_t)$ on the tangent space. The resulting distribution concentrates its mass on models which are probable under $p(\vec{\theta}|\vec{\mu}_t, \boldsymbol{C}_t)$ and close to the manifold.

e.g the receptive field might be well-approximated by a Gabor function [126, 120], or by a difference of Gaussians [53], or by a low rank spatiotemporal matrix [94, 120]. We can think of this structure as defining a sub-manifold, $\mathcal{M}$, of the full model space, Θ,

$$\mathcal{M} = \{\vec{\theta} : \vec{\theta} = \Psi(\vec{\phi}), \forall \vec{\phi}\}. \tag{368}$$

The vector, $\vec{\phi}$, essentially enumerates the points on the manifold and $\Psi()$ is a function which maps these points into Θ space. A natural example is the case where we wish to enforce the constraint that $\vec{\theta}$ has some parametric form.

The availability of prior knowledge obviously has strong implications for the optimal design. If a-priori we believe a neuron has a Gabor receptive field then our goal is to identify the optimal parameters of the Gabor function. Intuitively, to measure how well different stimuli will allow us to discriminate between different Gabor functions, we need to integrate over all possible models given our prior knowledge. To compute the mutual information for a particular design, we sum the amount of evidence we expect to collect for each model, weighted by the posterior probability on each model. We integrate over model space because the informativeness of an experiment clearly depends on what we already know (i.e. the likelihood we assign to each model given the data and our prior knowledge). Furthermore, the informativeness of an experiment will depend on the outcome. Hence, we use what we know about the neuron to make predictions about the experimental outcome. Unfortunately, since $\mathcal{M}$ can in general have some arbitrary nonlinear shape we cannot easily compute integrals over the manifold. Furthermore, we do not want to continue to restrict ourselves to models on the manifold if the data indicates our prior knowledge is wrong.

Our approach is therefore based on maintaining a Gaussian approximation of the posterior on the full model space, Θ. This posterior ignores our knowledge of $\mathcal{M}$. We know from Chapter 2 that the MAP of this full posterior, $\vec{\mu}_t$, is a consistent estimator of the true parameters (provided the design satisfies certain properties; an issue we

return to later). Thus, by computing the full posterior on Θ we can guarantee that $\vec{\mu}_t$ will converge to the true parameters even if our prior knowledge is wrong.

When optimizing the design, however, we do not simply want to use the full posterior on $\vec{\theta}$ because this posterior ignores our prior knowledge about $\mathcal{M}$. Therefore, we want to regularize our posterior so as to reduce the likelihood of models not on $\mathcal{M}$ and increase the likelihood of models on or close to $\mathcal{M}$. As noted earlier, it is critical that the regularized posterior have a structure which makes computing the mutual information tractable. Our solution involves a linear approximation of $\mathcal{M}$ using the tangent space of the manifold as illustrated in Figure 28 [86]. The tangent space is a linear space which provides a local approximation of the manifold. Since the tangent space is a linear subspace of Θ, integrating over the tangent space is much easier than integrating over all $\vec{\theta}$ on the manifold; in fact, the methods introduced in Chapter 2 may be applied directly to this case. The tangent space is a local linear approximation evaluated at a particular point, $\vec{\mu}_{\mathcal{M},t}$, on the manifold. For $\vec{\mu}_{\mathcal{M},t}$ we use the projection of $\vec{\mu}_t$ onto the manifold (i.e., $\vec{\mu}_{\mathcal{M},t}$ is the closest point in $\mathcal{M}$ to $\vec{\mu}_t$). Depending on the manifold, computing $\vec{\mu}_{\mathcal{M},t}$ can be nontrivial; the examples considered in this paper, however, all have tractable numerical solutions to this problem.

Our methods have a very intuitive explanation. Since we cannot simultaneously optimize the design for all models, it makes sense to try to optimize the design for a smaller set of models which are highly likely given our prior knowledge, $\mathcal{M}$, and the data already collected. Therefore, we want to consider models which are 1) close to $\vec{\mu}_t$ and 2) close to the manifold. Since the full posterior is a unimodal distribution, the most likely models given the data collected are models close to $\vec{\mu}_t$. The projection of the MAP onto the manifold, $\vec{\mu}_{\mathcal{M},t}$, is by definition the point on the manifold closest to $\vec{\mu}_t$. A natural approach is to focus on computing the mutual information with respect to models on the manifold and close to $\vec{\mu}_{\mathcal{M},t}$. The challenge is representing the set of models close to $\vec{\mu}_{\mathcal{M},t}$ in a way that makes integrating over the

models tractable. To find models on the manifold close to $\vec{\mu}_{\mathcal{M},t}$ we want to perturb the parameters $\vec{\phi}$ about the values corresponding to $\vec{\mu}_{\mathcal{M},t}$. Since Ψ is in general nonlinear there is no simple expression for the combination of all such perturbations. However, we can easily approximate the set of $\vec{\theta}$ resulting from these perturbations by taking linear combinations of the partial derivatives of Ψ with respect to $\vec{\phi}$. The partial derivative is the direction in Θ in which $\vec{\theta}$ moves if we perturb one of the manifold's parameters. Thus, the subspace formed by linear combinations of the partial derivatives approximates the set of models on the manifold close to $\vec{\mu}_{\mathcal{M},t}$. This subspace is the tangent space,

$$T_{\vec{\mu}_{\mathcal{M},t}}\mathcal{M} = \{\vec{\theta} : \vec{\theta} = \vec{\mu}_{\mathcal{M},t} + \boldsymbol{B}\vec{b},\ \forall \vec{b} \in \mathcal{R}^{\dim(\mathcal{M})}\} \qquad \boldsymbol{B} = orth\left(\left[\frac{\partial \Psi}{\partial \phi_1} \dots \frac{\partial \Psi}{\partial \phi_d}\right]\right), \tag{369}$$

where *orth* is an orthonormal basis for the column space of its argument. Here $T_x\mathcal{M}$ denotes the tangent space at the point x. The columns of $\boldsymbol{B}$ denote the direction in which $\vec{\theta}$ changes if we perturb one of the manifold's parameters. (In general, the directions corresponding to changes in different parameters are not independent; to avoid this redundancy we compute a set of basis vectors for the space spanned by the partial derivatives.)

Consider the simple example where $\vec{\theta}$ is a 1-d receptive field. Suppose we know a-priori that the components of $\vec{\theta}$ follow a Gabor function so that the i^{th} element of $\vec{\theta}$ is

$$\theta_i = A\cos\Big((i-c)\,\omega\Big)\exp\left(-\frac{1}{2\sigma^2}(i-c)^2\right), \tag{370}$$

where A is the amplitude and c is the center. In this case the partial derivatives with

respect to A and c are

$$\frac{\partial \theta_i}{\partial A} = \cos\Big((i-c)\omega\Big)\exp\left(-\frac{1}{2\sigma^2}(i-c)^2\right) \tag{371}$$

$$\frac{\partial \theta_i}{\partial c} = \left(\cos\Big((i-c)\omega\Big)\frac{1}{\sigma^2}(i-c) + \omega\sin\Big((i-c)\omega\Big)\right) A\exp\left(-\frac{1}{2\sigma^2}(i-c)^2\right). \tag{372}$$

The partial derivatives $\frac{\partial \vec{\theta}}{\partial A}$ and $\frac{\partial \vec{\theta}}{\partial c}$ define vectors in Θ. The subspace formed by linear combinations of these vectors is the tangent space.

We can use our Gaussian posterior on the full parameter space to easily compute the posterior likelihood of the models in the tangent space. Since the tangent space is a subspace of Θ, restricting our Gaussian approximation, $p(\vec{\theta}|\vec{\mu}_t, \boldsymbol{C}_t)$, to the tangent space means we are taking a slice through our Gaussian approximation of the posterior. Mathematically, we are conditioning on $\vec{\theta} \in T_{\vec{\mu}_{\mathcal{M},t}}\mathcal{M}$. The result is a Gaussian distribution on the tangent space whose parameters may be obtained using the standard Gaussian conditioning formula:

$$p_{tan}(\vec{\theta}|\vec{\mu}_{b,t}, C_{b,t}) = \begin{cases} \mathcal{N}(\vec{b}; \vec{\mu}_{b,t}, C_{b,t}) & if \quad \exists\ \vec{b}\ s.t\, \vec{\theta} = \vec{\mu}_{\mathcal{M},t} + \boldsymbol{B}\vec{b} \\ 0 & if \qquad \vec{\theta} \notin T_{\vec{\mu}_{\mathcal{M},t}} \end{cases} \tag{373}$$

$$\vec{\mu}_{b,t} = -C_{b,t}\boldsymbol{B}^T\boldsymbol{C}_t^{-1}(\vec{\mu}_{\mathcal{M},t} - \vec{\mu}_t) \qquad C_{b,t} = (\boldsymbol{B}^T\boldsymbol{C}_t^{-1}\boldsymbol{B})^{-1} \tag{374}$$

where $\mathcal{N}$ denotes a normal distribution with the specified parameters. Now, rather than optimizing the stimulus by trying to squeeze the uncertainty $p(\vec{\theta}|\boldsymbol{r_{1:t}}, \boldsymbol{s_{1:t}}, \mathcal{M})$ on the nonlinear manifold $\mathcal{M}$ down as much as possible (a very difficult task in general), we pick the stimulus which best reduces the uncertainty $p_{tan}(\vec{\theta}|\vec{\mu}_{b,t}, C_{b,t})$ on the vector space $T_{\vec{\mu}_{\mathcal{M},t}}$.

Following the methods presented in Chapter 2 we can quantify the informativeness of any stimulus, $\vec{x}_{t+1}$, using the mutual information,

$$I(r_{t+1}; \vec{\theta}|\vec{s}_{t+1}, \boldsymbol{s_{1:t}}, \boldsymbol{r_{1:t}})$$
$$\approx E_{\vec{\theta}|\boldsymbol{s_{1:t+1}}, \boldsymbol{r_{1:t}}, \mathcal{M}} E_{r_{t+1}|\rho_{t+1}} \log\left(1 - \frac{\partial^2 \log p(r_{t+1}|\rho_{t+!})}{\partial \rho_{t+1}^2}|_{\rho_{t+1}}\sigma_\rho^2\right) + const. \tag{375}$$

Instead of using the posterior on the manifold, $p(\vec{\theta}|\boldsymbol{r_{1:t}}, \boldsymbol{s_{1:t}}, \mathcal{M})$, to compute the mutual information we use the Gaussian approximation on the tangent space,

$$I(r_{t+1}; \vec{\theta}|\vec{s}_{t+1}, \boldsymbol{s_{1:t}}, \boldsymbol{r_{1:t}}) \approx E_{\vec{\theta}|\boldsymbol{s_{1:t+1}}, \boldsymbol{r_{1:t}}, p_{tan}} E_{r_{t+1}|\rho_{t+1}} \log\left(1 - \frac{\partial^2 \log p(r_{t+1}|\rho_{t+!})}{\partial \rho_{t+1}^2}|_{\rho_{t+1}} \sigma_\rho^2\right) + const. \tag{376}$$

$$\rho_{t+1} = \vec{s}_{t+1}^T \vec{\theta}. \tag{377}$$

Since the posterior on $\vec{b}$ is normal with mean and covariance $(\vec{\mu}_{b,}, C_{b,})$, the distribution on ρ_{t+1} is also normal with mean and covariance

$$\mu_\rho = \vec{s}_{t+1}^T \vec{\mu}_{\mathcal{M},t} + \vec{s}_{t+1}^T \boldsymbol{B} \vec{\mu}_{b,} \tag{378}$$

$$\sigma_\rho^2 = \vec{s}_{t+1}^T (\boldsymbol{B}^T C_{b,} \boldsymbol{B})^{-1} \vec{s}_{t+1}. \tag{379}$$

This result means that the mutual information is a function of just two scalars $(\mu_\rho, \sigma_\rho^2)$ which are linear and quadratic functions of the input respectively. As a result, we can continue to use the methods presented in Chapter 2 to choose the optimal stimulus. Finally, to handle the possibility that $\vec{\theta} \notin \mathcal{M}$, every so often we optimize the stimulus using the full posterior $p(\vec{\theta}|\vec{\mu}_t, \boldsymbol{C}_t)$. This simple modification ensures that asymptotically we do not ignore directions orthogonal to the manifold; i.e., that we do not get stuck obsessively sampling along the incorrect manifold. As a result, μ_t will always converge asymptotically to the true parameters, even when $\theta \notin \mathcal{M}$.

To summarize, our method proceeds as follows:

0. Initial conditions: start with a log-concave (approximately Gaussian) posterior given t previous trials, summarized by the posterior mean, $\vec{\mu}_t$ and covariance, $\boldsymbol{C}_t$.

1. Compute $\vec{\mu}_{\mathcal{M},t}$, the projection of $\vec{\mu}_t$ on the manifold. (The procedure for computing $\vec{\mu}_{\mathcal{M},t}$ depends on the manifold.)

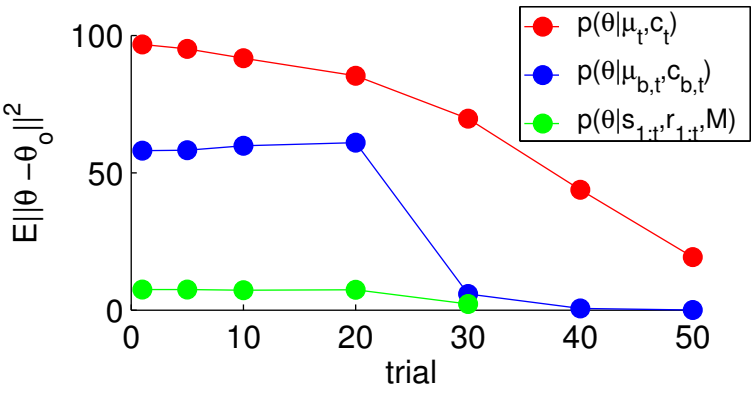


Figure 29: The mean squared error computed using the true posterior and our Gaussian approximations for our Gabor simulation. The results show that the error under $p(\vec{\theta}|\vec{\mu}_{b,t}, \boldsymbol{C}_t)$ quickly converges to the true posterior on the manifold and is much less than the error under the posterior on the full space.

2. Compute the tangent space of $\mathcal{M}$ at $\vec{\mu}_{\mathcal{M},t}$ using Eqn. 369.

3. Compute the posterior restricted to the tangent space, $p_{tan}(\vec{\theta}|\vec{\mu}_{b,t}, C_{b,t})$, using the standard Gaussian conditioning formula (Eqn. 374).

4. Apply the methods in Chapter 2 to find the optimal $t+1$ stimulus, and observe the response r_{t+1}.

5. Recursively update the posterior mean and covariance matrix: $\vec{\mu}_t \rightarrow \vec{\mu}_{t+1}$ and $\boldsymbol{C}_t \rightarrow \boldsymbol{C}_{t+1}$ (again, as in Chapter 2), and return to step 1.

5.3 Results

We tested our methods using real data and simulations designed to mimic real experiments.

5.3.1 1-d example

Our first simulation involves an overly simple model. The purpose of this example is to present a contrived example in which we can compute the true posterior on $\mathcal{M}$ numerically so that we can evaluate the quality of the tangent space approximation. In this simple example, $\vec{\theta}$ is a 1-d receptive field. The components of $\vec{\theta}$ follow a Gabor

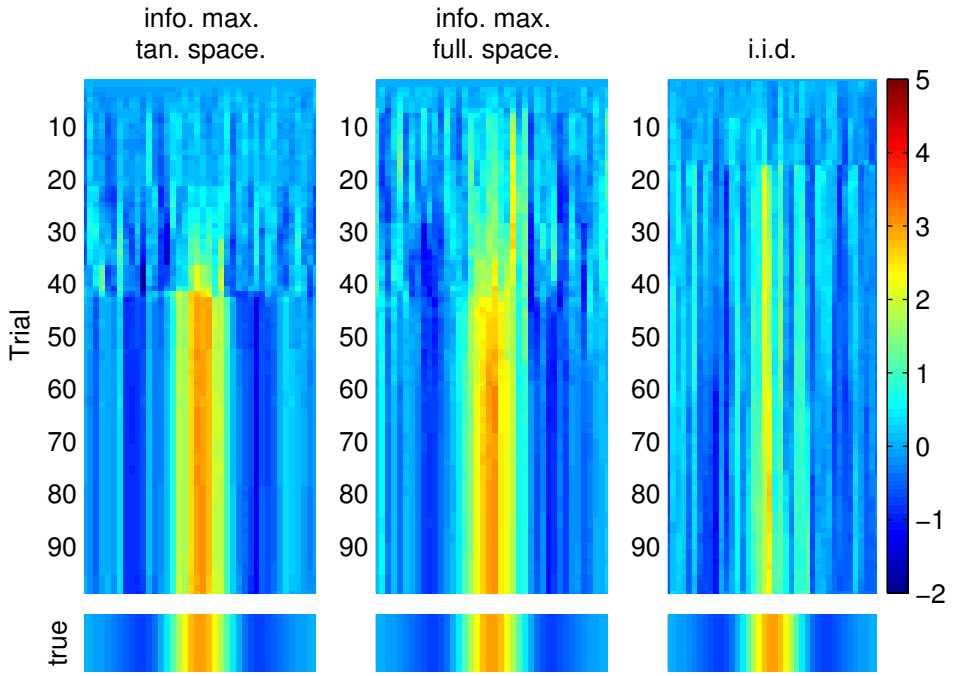


Figure 30: We compare the effectiveness of the different designs in the case where $\vec{\theta}$ is a 1-d Gabor function by plotting the MAP of the full posterior, $\vec{\mu}_t$. Each row in the images shows the MAP on a different trial for one of the designs. Below each image we plot the true parameters. Both info. max. designs converge more rapidly than the i.i.d. design to the true parameters. The design which exploits the tangent space does slightly better than the info. max. design which uses the full posterior.

function, Eqn. 370, and only the amplitude, A, and the center, c, are unknown. For this example, we set $\dim(\vec{\theta}) = 40$.

We can evaluate the effectiveness of the tangent space approximation by computing the mean squared error, $E||\vec{\theta} - \vec{\theta}_o||^2$, using the true posterior, $p(\vec{\theta}|\boldsymbol{r_{1:t}}, \boldsymbol{s_{1:t}}, \mathcal{M})$, and our Gaussian approximations, $p(\vec{\theta}|\vec{\mu}_t, \boldsymbol{C}_t)$ and $p_{tan}(\vec{\theta}|\vec{\mu}_{b,t}, C_{b,t})$. $\vec{\theta}_o$ denotes the true parameters. The results are shown in Figure 29. The results clearly show that the mean squared error using the posterior on the tangent space decreases much faster than if we ignore our prior information, although not nearly as fast as using the true posterior on the manifold. Thus, by using the prior knowledge that the true θ lives close to a 2-d sub-manifold of the whole parameter space, we are able to choose a much more informative sequence of stimuli. In Figure 5.3.1 we plot the MAPs on each trial for each design.

5.3.2 Low rank models

To test our methods in a realistic, high-dimensional setting, we simulated a typical auditory neurophysiology [151, 94, 168] experiment. Here, the objective is to identify the spectro-temporal receptive field (STRF) of the neuron. The input and receptive field of the neuron are usually represented in the spectral-domain because nonlinearly transforming the input by mapping it into the spectro-temporal domain generally leads to better fits of the data using simple, e.g. linear, models [64]. The STRF, $\theta(\tau, \omega)$, is a 2-d filter which relates the firing rate at time t to the amount of energy at frequency ω and time $t - \tau$ in the stimulus. To incorporate this spectro-temporal model into the GLM setting we simply vectorize the matrix $\theta(\tau, \omega)$.

Estimating the STRF can be quite difficult due to its high dimensionality. Several researchers, however, have shown that low-rank assumptions can be used to produce accurate approximations of the receptive field while significantly reducing the number of unknown parameters [42, 120, 94, 2]. A low rank assumption is a more general

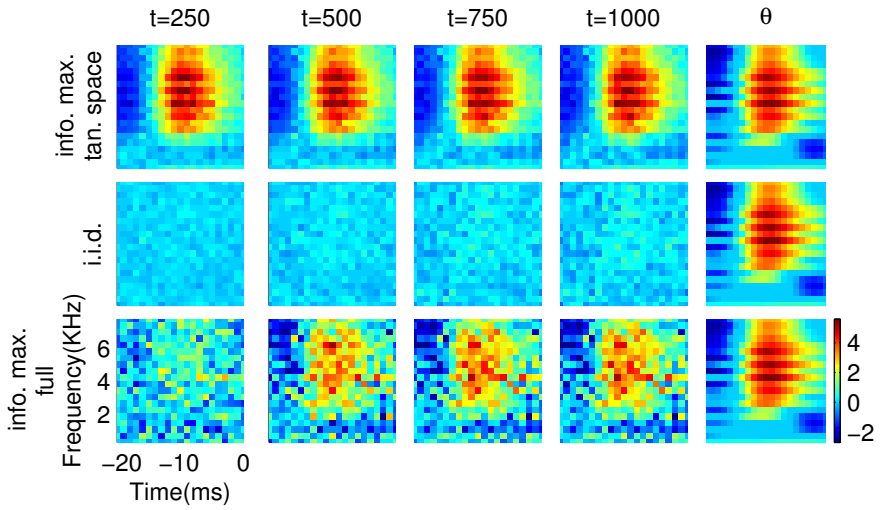


Figure 31: MAP estimates of a STRF obtained using three designs: the new info. max. tangent space design described in the text; an i.i.d. design; and an info. max. design which did not use the assumption that $\vec{\theta}$ corresponds to a low rank STRF. In each case, stimuli were chosen under the spherical power constraint, $||\vec{s}_t||_2 = c$. The true STRF (fit to real zebra finch auditory responses and then used to simulate the observed data) is shown in the last column. (For convenience we rescaled the coefficients to be between -4 and 4). We see that using the tangent space to optimize the design leads to much faster convergence to the true parameters; in addition, both info. max. designs significantly outperform the i.i.d. design here. In this case the true STRF did not in fact lie on the manifold $\mathcal{M}$ (chosen to be the set of rank-2 matrices here); thus, these results also show that our knowledge of $\mathcal{M}$ does not need to be exact in order to improve the experimental design.

version of the space-time separable assumption that is often used when studying visual receptive fields [38]. Mathematically, a low-rank assumption means that the matrix corresponding to the STRF can be written as a sum of rank one matrices,

$$\mathbf{\Theta} = Mat\,\vec{\theta} = UV^T \tag{380}$$

where Mat indicates the matrix formed by reshaping the vector $\vec{\theta}$ to form the STRF. U and V are low-rank matrices with orthonormal columns. The columns of U and V are the principal components of the column and row spaces of $\mathbf{\Theta}$ respectively, and encode the spectral and temporal properties of the STRF, respectively.

We simulated an auditory experiment using an STRF fitted to the actual responses of a neuron in the Mesencephalicus lateralis pars dorsalis (MLd) of an adult male zebra finch [168]. To reduce the dimensionality we sub-sampled the STRF in the frequency domain and shortened it in the time domain to yield a 20×21 STRF. We generated synthetic data by sampling a Poisson process whose instantaneous firing rate was set to the output of a GLM with exponential nonlinearity and $\vec{\theta}$ proportional to the true measured zebra finch STRF [1].

For the manifold we used the set of $\vec{\theta}$ corresponding to rank-2 matrices. For the STRF we used, the rank-2 assumption turns out to be rather accurate. We also considered manifolds of rank-1 and rank-5 matrices (data not shown), but rank-2 did slightly better. The manifold of rank r matrices is convenient because we can easily project any $\vec{\theta}$ onto $\mathcal{M}$ by reshaping $\vec{\theta}$ as a matrix and then computing its singular-value-decomposition (SVD). $\vec{\mu}_{\mathcal{M},t}$ is the matrix formed by the first r singular vectors of $\vec{\mu}_t$. To compute the tangent space, Eqn. 369, we compute the derivative of $\vec{\theta}$ with respect to each component of the matrices U and V. Using these derivatives we can linearly approximate the effect on $\mathbf{\Theta}$ of perturbing the parameters of its principal components.

[1] We obtained the STRF from our collaborators David Schneider and Dr. Sarah Woolley who performed the actual experiments.

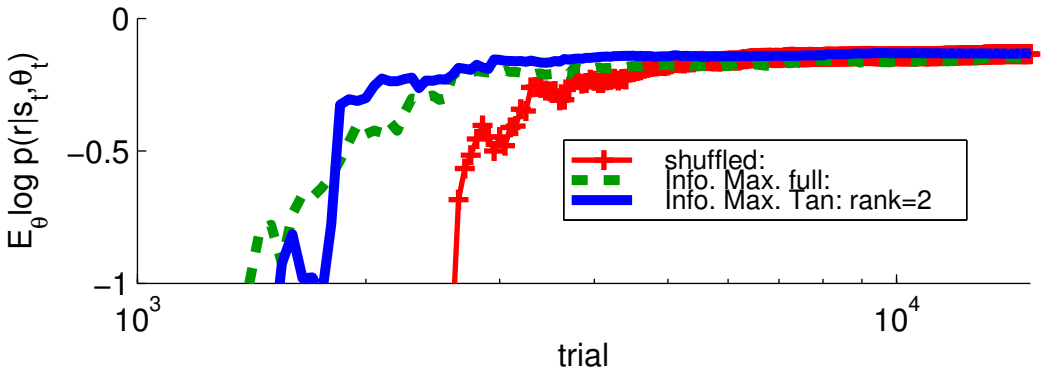


Figure 32: Plots comparing the performance of an info. max. design, an info. max. design which uses the tangent space, and a shuffled design. The manifold was the set of rank 2 matrices. The plot shows the expected log-likelihood (prediction accuracy) of the spike trains in response to a birdsong in the test set. Using a rank 2 manifold to constrain the model produces slightly better fits of the data.

In Figure 31 we compare the effectiveness of different experimental designs by plotting the MAP estimate $\vec{\mu}_t$ on several trials. The results clearly show that using the tangent space to design the experiments leads to much faster convergence to the true parameters. Furthermore, using the assumption that the STRF is rank-2 is beneficial even though the true STRF here is not in fact rank-2.

5.3.3 Real birdsong data

We also tested our method by using it to reshuffle the data collected during an actual experiment to find an ordering which provided a faster decrease in the error of the fitted model. During the experiments, the responses of MLd neurons were recorded while the songs of other birds and ripple noise were presented to the bird. The data was collected by our collaborators David Schneider and Sarah Woolley and is described in more detail in Chapter 4. We compared a design which randomly shuffled the trials to a design which used our info. max. algorithm to select the order in which the trials are processed. We then evaluated the fitted model by computing the expected log-likelihood of the spike trains, $\sum_\tau E_{\vec{\theta}|\vec{\mu}_t,\boldsymbol{C}_t} \log p(r_\tau|\vec{s}_\tau,\vec{\theta})$; the expectation is computed with respect to our posterior on $\vec{\theta}$. τ denotes all the observations made

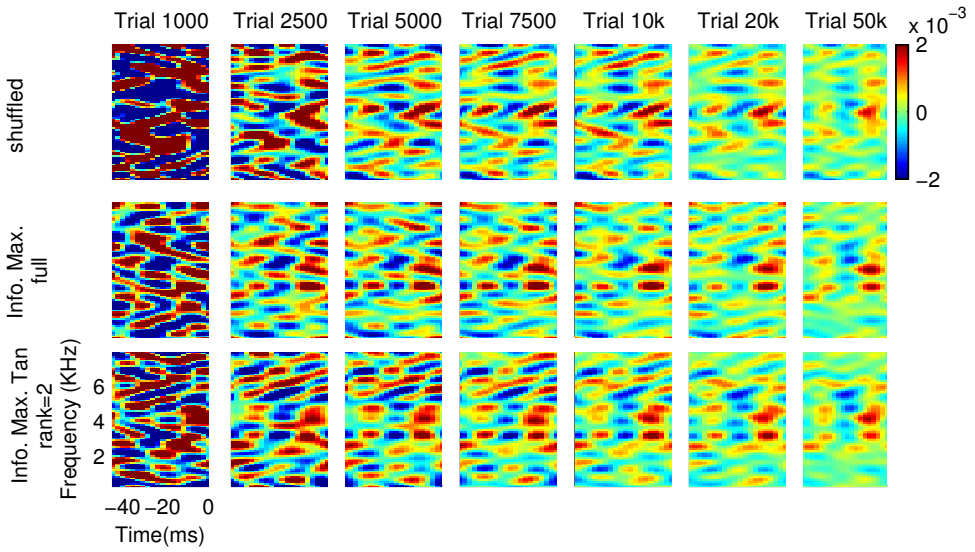


Figure 33: The STRFs estimated using the bird song data. We plot $\vec{\mu}_t$ for trials in the interval over which the expected log-likelihood of the different designs differed the most in Fig. 32. The info. max. designs converge slightly faster than the shuffled design. In these results, we smoothed the STRF by only using frequencies less than or equal to $10f_{o,f}$ and $2f_{o,t}$.

when inputs in a test set are played to the bird.

To constrain the models we assume the STRF is low-rank and that its principal components are smooth. The smoothing prior means that if we take the Fourier transform of the principal components, the Fourier coefficients of high frequencies should be zero with high probability. In other words, each principal component (the columns of U and V) should be a linear combination of sinusoidal functions with low frequencies. In this case we can write the STRF as

$$\boldsymbol{\Theta} = \boldsymbol{\mathcal{F}}\nu\boldsymbol{\omega}\eta^T\boldsymbol{\mathcal{T}}^T. \tag{381}$$

Each column of $\boldsymbol{\mathcal{F}}$ and $\boldsymbol{\mathcal{T}}$ is a sine or cosine function representing one of the basis functions of the principal spectral (columns of $\boldsymbol{\mathcal{F}}$) or temporal (columns of $\boldsymbol{\mathcal{T}}$) components of the STRF. Each column of ν and η determines how we form one of the principal components by combining sine and cosine functions. $\boldsymbol{\omega}$ is a diagonal

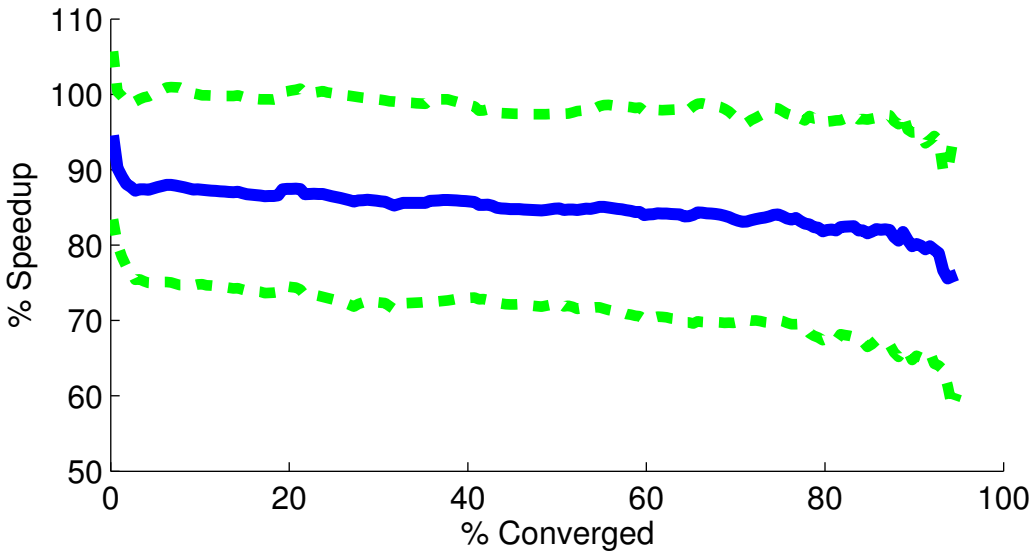


Figure 34: A plot of the speedup achieved by using the info. max. design with the tangent space compared to an info. max. design which ignores the prior information. The speedup is plotted as a function of % Converged as described in Chapter 4. The solid blue line shows the average speedup across all 11 neurons and the dashed green lines show plus and minus one standard deviation. The average is slightly less than 100% indicating that on average using the tangent space *decreased* performance; in particular using the tangent space required on average 10% more trials than the info. max. design which ignored prior information to train an equally well fit model. The speedup is not computed for values of % Converged $> 95\%$ because the Speedup cannot be accurately computed for these values (see Chapter 4).

matrix which specifies the projection of $\boldsymbol{\Theta}$ onto each principal component. The unknown parameters in this case are the matrices ν, η, and $\boldsymbol{\omega}$. The sinusoidal functions corresponding to the columns of $\boldsymbol{\mathcal{F}}$ and $\boldsymbol{\mathcal{T}}$ should have frequencies $\{0, \ldots, f_{o,f} m_f\}$ and $\{0, \ldots, f_{o,t} m_t\}$ respectively. $f_{o,f}$ and $f_{o,t}$ are the fundamental frequencies and are set so that 1 period corresponds to the dimensions of the STRF. m_f and m_t are the largest integers such that $f_{o,f} m_f$ and $f_{o,t} m_t$ are less than the Nyquist frequency. Now to enforce a smoothing prior we can simply restrict the columns of $\boldsymbol{\mathcal{F}}$ and $\boldsymbol{\mathcal{T}}$ to sinusoids with low frequencies. To project $\boldsymbol{\Theta}$ onto the manifold we simply need to compute $\nu, \boldsymbol{\omega}$ and η by evaluating the SVD of $\boldsymbol{\mathcal{F}}^T \boldsymbol{\Theta} \boldsymbol{\mathcal{T}}$.

We ran our simulations using the data from 11 neurons (these are the same neurons we analyzed in Chapter 4). In Figure 32 we compare the expected log-likelihood

for all three designs for one neuron. The estimated STRFs are shown in Figure 33. Both info. max. designs outperform the randomly shuffled design. However, incorporating the low-rank assumption using the tangent space only provides a small, transient improvement compared to the full info. max. design.

To compare the average performance of the two info. max. designs across all neurons we computed for each neuron the Speedup of the design using the tangent space compared to an info. max. design using the full space. We compute the Speedup as a function of % Converged which measures the quality of the fitted GLM relative to the best fit GLM (for a discussion of Speedup and % Converged see Chapter 4 Page 177). In Figure 34 we plot the mean and standard deviation of the Speedup of the info. max. design using the tangent space compared to the full info. max. design (a plot of the Speedup of the info. max. design compared to the shuffled design is in Chapter 4). The mean is roughly 90% which indicates that on average using the tangent space actually produced a less efficient design; i.e. the design using the tangent space required on average 10% more trials than the info. max. design using the full posterior.

We think the results in Figure 34 provide a poor indication of how the two info. max. designs would compare in actual experiments. A major limitation of our offline analysis is that we are restricted to picking stimuli which were actually presented. Thus, both info. max. designs are restricted to choosing the best input from the same set of roughly 20×10^3 distinct stimuli. We do not think this stimulus set is large enough to allow the low-rank assumption to really be exploited; i.e both info. max. designs end up picking very similar stimuli. In principle, during actual experiments the designs would be free to pick any input. As a result, we might expect that the info. max. design would be able to exploit the low rank assumption to find a much more informative stimulus when searching this much larger stimulus space. The main conclusion is that the offline analysis is really insufficient to conclude whether

the low-rank assumption can be used to further improve the info. max. design.

Furthermore, we think the expected log-likelihood might be a slightly "biased" metric. We compute the expected log-likelihood using the posterior on the full model space. The info. max. design using the low rank assumption, however, does not really attempt to decrease the variance in directions orthogonal to the manifold; i.e this design implicitly assumes that all models not on the manifold have zero probability. Therefore the full posterior for the info. max. design using the tangent space over-estimates our uncertainty. Over estimating the uncertainty would decrease the expected log-likelihood and could explain why the Speedup is less than 100%.

5.4 Discussion

In this chapter we have shown how our methods may be modified to use detailed prior information to design maximally informative experiments. Although the results presented in the previous section used the greedy algorithm presented in Chapter 2, we could just as easily have used the methods in Chapter 3 to compute an optimal distribution on the inputs. To apply the methods in Chapter 3 we simply compute the posterior on the tangent space and then use this distribution in place of the full posterior to compute the optimal distribution on the stimuli.

In general, our methods will not work equally well for all manifolds, or even all points on a manifold. Our methods will work well when the tangent space evaluated at $\vec{\mu}_{\mathcal{M},t}$ provides a good approximation of the manifold in the neighborhood of $\vec{\mu}_{\mathcal{M},t}$. Clearly, the manifold should be smooth so that the partial derivatives with respect to the manifold's parameters are well defined everywhere. Furthermore, the linear approximation provided by the tangent space will perform poorly at locations where the manifold is sharply curved; i.e locations where a small perturbation in $\vec{\phi}$ produces a large change in $\vec{\theta}$. Finally, we would expect our methods to have problems when the projection of $\vec{\theta}$ onto the manifold is not distance preserving; i.e if nearby $\vec{\theta}$ end

up being mapped to very different values of $\vec{\phi}$. The fact that we maintain the full posterior on the full $\vec{\theta}$ space makes our design fairly robust to these issues because $\vec{\mu}_t$ is a consistent estimator of $\vec{\theta}$.

To prevent incorrect prior information from leading us to erroneous conclusions we need to be willing to reject our prior information in the face of sufficient evidence. In presenting our methods, we used fairly simple methods for ensuring robustness. In particular, every so often we pick stimuli using the full posterior. Using the full posterior every so often ensures that the MAP of the full posterior, $\vec{\mu}_t$, is a consistent estimator of the parameters. In our simulations this proved sufficient, however, we could easily use a more rigorous approach. For example, we can easily compute the distance between $\vec{\mu}_t$ and $\vec{\mu}_{\mathcal{M},t}$. If after some number of trials this distance is large then we might conclude that our prior information is wrong and we should use the full posterior more often or exclusively. Alternatively, we can augment the tangent space, increasing its dimensionality by one, so that it includes the error vector $\vec{\mu}_t - \vec{\mu}_{\mathcal{M},t}$. If we augment the tangent space with the error vector then $\vec{\mu}_t$ always lies within the augmented tangent space. Consequently, we will be including $\vec{\mu}_t$ in the set of models used to optimize the design.

One of the main benefits of the Bayesian mindset is that we may use prior information to regularize high-dimensional models and design better experiments when data is scarce. Unfortunately, incorporating realistic prior information is quite difficult because we must be able to compute expectations with respect to the prior distribution. In this chapter we have presented approximate methods which allow us to use realistic prior information to optimize neurophysiology experiments.

CHAPTER VI

CONCLUSION

With over 10 billion neurons and even more synapses, reverse engineering the human brain is one of the most daunting engineering tasks ever undertaken [46]. Since natural stimuli easily have 100-1000 dimensions, understanding neural processing necessitates a more principled approach then simply measuring the responses to a set of uniformity sampled inputs drawn from stimulus space. A rigorous search of all possible functions that exist on such a large domain is simply impossible. The only feasible approach, the one long employed by neuroscientists and engineers, is an iterative process. We begin by using the simplest model we can imagine and seeing how well it captures the behavior of neurons. Once we understand the limitations of this simple model, we can attempt to develop more complex methods which address the deficiencies of the simple model while preserving its useful features. Towards this end we want to design experiments which 1) efficiently collect the data needed to fit these simple models and 2) search for the weaknesses of these simple models. In light of these objectives, we have presented methods for sequential optimal experimental (SOE) design. We think the incorporation of our methods into neurophysiology experiments will help neuroscientists continue their steady progress towards understanding neural computation.

Evaluating the methods presented in this thesis raises an obvious question, "Why is the speedup achievable using optimized designs worth the effort?" In particular, the results in Chapter 4 showed that existing non-optimized methods can estimate the STRFs just as well as our methods; they just take slightly longer. Furthermore, our methods are heavily tailored to fitting 1-d GLMs. We justified this restriction by

noting that GLMs have proven to be adequate models for several types of neurons [17, 18, 26, 149, 115]. A critic might argue that the success of the GLM shows that existing methods are sufficient and therefore conclude that optimizing neurophysiology experiments to fit GLMs is not worth the effort. This criticism misses the point of sequential optimal experimental design. The goal of SOE is not to shorten the duration of neurophysiology experiments but to increase the complexity of the models and hypotheses that can be investigated in some amount of time. If we can learn the standard 1-dimensional receptive field of a neuron in half the time then we can devote 50% of our experiment to investigating more complicated models. Furthermore, we can use the estimate of the 1-dimensional receptive field to try to design optimal experiments for fitting more complicated models.

One of the most intuitive and useful frameworks for understanding neural computation is thinking of neurons as feature detectors [4, 15, 119]. Hubel and Wiesel's work, for example, showed that we can think of simple cells as detectors for bars of different orientations [74]. Most models of neurons either explicitly or implicitly adopt this view and focus on estimating the input features to which neurons respond. The 1-d GLM for example assumes the features a neuron detects have a simple geometric representation; i.e. the features are the projection of the input onto a 1-dimensional subspace of the input. In general, neuroscientists refer to the features a neuron responds to as its subspace. Typically, the receptive field is a defined as a linear subspace of the input. Furthermore, most methods for estimating the receptive field, e.g reverse correlation or spike triggered averaging, assume the receptive field is 1-dimensional [123]. The restriction to one dimensional receptive fields is largely necessitated by the inability of existing methods to estimate higher-order receptive fields given existing data. Recently, new methods have been proposed for estimating 2-dimensional receptive fields recursively. These methods first estimate the best 1-dimensional receptive field and then increase the dimensionality of the receptive field

to account for effects not captured by the 1-d receptive field [4]. Since an estimate of the 1-d receptive field is needed to estimate higher order effects, methods which can reduce the amount of trials needed to robustly estimate the 1-d receptive field are quite valuable. In particular, the speedup attainable using our methods means we could devote fewer trials during an experiment to estimating the 1-d receptive field. Consequently, more trials could be used to determine how to best modify the 1-d receptive field to account for higher order properties of the response.

More generally when considering whether GLMs are overly restrictive, one needs to keep in mind the universal rule:"there is no free lunch." In the context of optimal experimental design, this adage means that for nonlinear models no design is simultaneously optimal for all models. Hence, to optimize our experiments with respect to a larger class of models, we must to some extent decrease the utility of the design with respect to any particular model within that class. By increasing the size of the class of models considered, we necessarily reduce how much speedup we can deliver using an optimized design. Consequently, considering a larger, more flexible, class of models than the 1-d GLM would in some sense hamper our ability to speedup our experiments. Simple, parametric models like the GLM are also essential to making the required computations tractable. Since we want to employ our methods in actual experiments, we place a premium on being able to efficiently perform the required computations.

In principle SOE can never do worse than a randomized design. However, the improvement due to a sequential, optimal experimental design depends on the underlying response function. In particular, the speedup depends on how much information we gain about the neuron's response function from each observation. The only way we can reduce the number of trials needed to estimate the response function is if we can predict how a neuron will respond to untested stimuli using the data already collected. For example, consider a visual neuron that is sharply tuned to some particular

image (e.g. the face of a parent). In this case, every image we present provides very little information about the neuron's preferred stimulus; i.e. if all we observe is that the neuron did not fire, we cannot infer from this observation what stimulus is likely to drive the neuron to fire. The key point is that a sequential optimal experimental design can only speedup an experiment by exploiting what we know; i.e what we know a-priori and what he have learned from the data already collected. In situations where we have little prior information and the data is uninformative SOE cannot do much better than non-optimized designs. Fortunately in many neurophysiology experiments there is a great deal of information for SOE to utilize.

Our hope is that the methods presented in this thesis will permit new experiments which will reveal previously unknown properties of neural computation.

REFERENCES

[1] Adelson, E. and Bergen, J., "Spatiotemporal energy models for the perception of motion," *Journal of the Optical Society of America A- Optics Image Science and Vision*, vol. 2, no. 2, pp. 284–299, 1985.

[2] Ahrens, M. B., Paninski, L., and Sahani, M., "Inferring input nonlinearities in neural encoding models.," *Network*, vol. 19, pp. 35–67, Mar 2008.

[3] Albright, T. D., "Direction and orientation selectivity of neurons in visual area mt of the macaque.," *J Neurophysiol*, vol. 52, pp. 1106–1130, Dec 1984.

[4] Arcas, B. A. Y., Fairhall, A. L., and Bialek, W., "Computation in a single neuron: Hodgkin and huxley revisited," *Neural Computation*, vol. 15, pp. 1715–1749, Aug. 2003.

[5] Bach, F., "Active learning for misspecified generalized linear models," in *Advances in Neural Information Processing Systems 19* (Schölkopf, B., Platt, J., and Hoffman, T., eds.), Cambridge, MA: MIT Press, 2007.

[6] Barlow, H., *Sensory Communication*, ch. Possible principles underlying the transform of sensory messages., pp. 213–234. MIT Press, 1961.

[7] Bartle, R. G., *The Elements of Real Analysis*. John Wiley & Sons Inc., 1976.

[8] Bates, R. A., Buck, R. J., Riccomagno, E., and Wynn, H. P., "Experimental design and observation for large systems," *Journal of the Royal Statistical Society Series B-Methodological*, vol. 58, no. 1, pp. 77–94, 1996.

[9] Ben-Gal, I. and Caramanis, M., "Sequential doe via dynamic programming," *IIE TRANSACTIONS*, vol. 34, pp. 1087–1100, Dec. 2002.

[10] Benda, J., Gollisch, T., Machens, C. K., and Herz, A. V., "From response to stimulus: adaptive sampling in sensory physiology," *Current Opinion in Neurobiology*, vol. 17, no. 4, pp. 430–436, 2007.

[11] Berger, J. O., *Statistical Decision Theory and Bayesian Analysis*. Springer, 1985.

[12] Berkes, P. and Wiskott, L., "Slow feature analysis yields a rich repertoire of complex cell properties," *Journal of Vision*, vol. 5, pp. 579–602, 2005.

[13] Bernardo, J. M., "Expected information as expected utility," *Annals of Statistics*, vol. 7, no. 3, pp. 686–690, 1979.

[14] Berry, M. and Meister, M., "Refractoriness and neural precision," *Journal of Neuroscience*, vol. 18, pp. 2200–2211, 1998.

[15] Bialek, W. and de Ruyter van Steveninck, R. R., "Features and dimensions: Motion estimation in fly vision," 2005.

[16] Boutilier, C., Dean, T., and Hanks, S., "Decision-theoretic planning: Structural assumptions and computational leverage," *Journal of Artificial Intelligence Research*, vol. 11, pp. 1–94, 1999.

[17] Brillinger, D., "Maximum likelihood analysis of spike trains of interacting nerve cells," *Biological Cyberkinetics*, vol. 59, pp. 189–200, 1988.

[18] Brillinger, D., "Nerve cell spike train data analysis: a progression of technique," *Journal of the American Statistical Association*, vol. 87, pp. 260–271, 1992.

[19] Carandini, M., Demb, J. B., Mante, V., Tolhurst, D. J., Dan, Y., Olshausen, B. A., Gallant, J. L., and Rust, N. C., "Do we know what the early visual system does?," *J Neurosci*, vol. 25, pp. 10577–10597, Nov 2005.

[20] Carlyon, R. P. and Shamma, S., "An account of monaural phase sensitivity.," *J Acoust Soc Am*, vol. 114, pp. 333–348, Jul 2003.

[21] Catchpole, C. and Slater, P., *Bird Song. Biological Themes and Variations*. Cambridge University Press, 1995.

[22] Chaloner, K., "A note on optimal bayesian design for nonlinear problems applied to logistic regression experiments.," *Journal of Statistical Planning and Inference*, vol. 37, pp. 229–235, 1993.

[23] Chaloner, K. and Larntz, K., "Optimal bayesian design applied to logistic-regression experiments," *Journal of Statistical Planning and Inference*, vol. 21, pp. 191–208, Feb. 1989.

[24] Chaloner, K. and Verdinelli, I., "Bayesian experimental design: A review," *Statistical Science*, vol. 10, pp. 273–304, Aug. 1995.

[25] Chaudhuri, P. and Mykland, P., "Nonlinear experiments: Optimal design and inference based on likelihood," *Journal of the American Statistical Association*, vol. 88, pp. 538–546, June 1993.

[26] Chichilnisky, E. J., "A simple white noise analysis of neuronal light responses," *Network-Computation in Neural Systems*, vol. 12, pp. 199–213, May 2001.

[27] Chipman, H. and Welch, W. J., "D-optimal design for generalized linear models," 1996.

[28] Cohn, D. A., Ghahramani, Z., and Jordan, M. I., "Active learning with statistical models," *Journal of Artificial Intelligence Research*, vol. 4, pp. 129–145, 1996.

[29] Cohn, D. A., "Neural network exploration using optimal experiment design," in *Advances in Neural Information Processing Systems* (Cowan, J. D., Tesauro, G., and Alspector, J., eds.), vol. 6, pp. 679–686, Morgan Kaufmann Publishers, Inc., 1994.

[30] Cottaris, N. P. and De Valois, R. L., "Temporal dynamics of chromatic tuning in macaque primary visual cortex," *Nature*, vol. 395, pp. 896–900, Oct. 1998.

[31] Cover, T. M. and Thomas, J. A., *Elements of Information Theory*. Wiley, 1991.

[32] DasGupta, A., "Review of optimal bayes designs," tech. rep., Purdue University, 1995.

[33] Dasgupta, S., "Analysis of a greedy active learning strategy," in *Advances in Neural Information Processing Systems 17* (Saul, L. K., Weiss, Y., and Bottou, L., eds.), (Cambridge, MA), pp. 337–344, MIT Press, 2005.

[34] David, S. V., Mesgarani, N., and Shamma, S. A., "Estimating sparse spectro-temporal receptive fields with natural stimuli.," *Network*, vol. 18, pp. 191–212, Sep 2007.

[35] Dawid, A. P. and Sebastiani, P., "Coherent dispersion criteria for optimal experimental design," *The Annals of Statistics*, vol. 27, pp. 65–81, 1999.

[36] Dayan, P. and Abbot, L., *Theoretical Neuroscience*. MIT Press, 2001. GLM models in computational neuroscience.

[37] de Boer, E. and de Jongh, H. R., "On cochlear encoding: Potentialities and limitations of the reverse-correlation technique.," *Journal of the Acoustical Scoiety of America*, vol. 63, no. 1, pp. 115–135, 1978.

[38] DeAngelis, G. C., Ohzawa, I., and Freeman, R. D., "Spatiotemporal organization of simple-cell receptive fields in the cat's striate cortex. i. general characteristics and postnatal development.," *J Neurophysiol*, vol. 69, pp. 1091–1117, Apr 1993.

[39] deCharms, R. C., Blake, D. T., and Merzenich, M. M., "Optimizing sound features for cortical neurons," *Science*, vol. 280, pp. 1439–1443, May 1998.

[40] Degroot, M., "Uncertatainty, information, and sequential experiments," *Annals of Mathematical Statistics*, vol. 33, no. 2, pp. 404–&, 1962.

[41] Demmel, J. W., *Applied Numerical Linear Algebra*. Siam, 1997.

[42] Depireux, D. A., Simon, J. Z., Klein, D. J., and Shamma, S. A., "Spectro-temporal response field characterization with dynamic ripples in ferret primary auditory cortex," *Journal of Neurophysiology*, vol. 85, pp. 1220–1234, 2001.

[43] Dette, H. and O'Brien, T. E., "Optimality criteria for regression models based on predicted variance.," *Biometrika*, vol. 86, pp. 93–106, 1999.

[44] DiCarlo, J. J., Johnson, K. O., and Hsiao, S. S., "Structure of receptive fields in area 3b of primary somatosensory cortex in the alert monkey," *Journal of Neuroscience*, vol. 18, pp. 2626–2645, Apr. 1998.

[45] Doupe, A. J. and Kuhl, P. K., "Birdsong and human speech: common themes and mechanisms.," *Annu Rev Neurosci*, vol. 22, pp. 567–631, 1999.

[46] Drachman, D. A., "Do we have brain to spare?," *Neurology*, vol. 64, pp. 2004–2005, Jun 2005.

[47] Dror, H. A. and Steinberg, D. M., "Sequential experimental designs for generalized linear models," *Journal of the American Statistical Association*, vol. 103, 2008.

[48] Edin, F., Machens, C., Schutze, H., and Herz, A., "Searching for optimal sensory signals: Iterative stimulus reconstruction in closed-loop experiments," *Journal of Computational Neuroscience*, vol. 17, no. 1, pp. 47–56, 2004.

[49] Eggermont, J. J., "Wiener and volterra analyses applied to the auditory system," *Hearing Research*, vol. 66, pp. 177–201, Apr. 1993.

[50] Eggermont, J. J., Aertsen, A. M., and Johannesma, P. I., "Quantitative characterisation procedure for auditory neurons based on the spectro-temporal receptive field.," *Hear Res*, vol. 10, pp. 167–190, May 1983.

[51] Eggermont, J. J., Epping, W. J., and Aertsen, A. M., "Stimulus dependent neural correlations in the auditory midbrain of the grassfrog (rana temporaria l.).," *Biol Cybern*, vol. 47, no. 2, pp. 103–117, 1983.

[52] Eggermont, J. J., Johannesma, P. M., and Aertsen, A. M., "Reverse-correlation methods in auditory research.," *Q Rev Biophys*, vol. 16, pp. 341–414, Aug 1983.

[53] Enroth-Cugell, C. and Robson, J. G., "The contrast sensitivity of retinal ganglion cells of the cat.," *Journal of Physiology*, vol. 187, pp. 517–552, 1966.

[54] Ergun, A., Barbieri, R., Eden, U. T., Wilson, M. A., and Brown, E. N., "Construction of point process adaptive filter algorithms for neural systems using sequential monte carlo methods," *IEEE Transactions on Biomedical Engineering*, vol. 54, no. 3, 2007.

[55] Escab, M. A., Miller, L. M., Read, H. L., and Schreiner, C. E., "Naturalistic auditory contrast improves spectrotemporal coding in the cat inferior colliculus.," *J Neurosci*, vol. 23, pp. 11489–11504, Dec 2003.

[56] Fabian, V., "On asymptotically efficient recursive estimation," *The Annals of Statistics*, vol. 6, pp. 854–866, 1978.

[57] Fedorov, V. V., *Theory of Optimal Experiments*. Academic Press, 1972.

[58] Fishbach, A., Nelken, I., and Yeshurun, Y., "Auditory edge detection: a neural model for physiological and psychoacoustical responses to amplitude transients.," *J Neurophysiol*, vol. 85, pp. 2303–2323, Jun 2001.

[59] Foldiak, P., "Stimulus optimisation in primary visual cortex," *Neurocomputing*, vol. 38–40, pp. 1217–1222, 2001.

[60] Ford, I., Kitsos, C., and Titterington, D. M., "Recent adavances in nonlinear experimental designs," *Technometrics*, vol. 31, pp. 49–60, 1989.

[61] Fortin, C., *A Survey of the Trust Region Subproblem within a Semidefinite Framework*. PhD thesis, University of Waterloo, 2000.

[62] Gentner, T. Q. and Margoliash, D., "Neuronal populations and single cells representing learned auditory objects.," *Nature*, vol. 424, pp. 669–674, Aug 2003.

[63] Gilad-Bachrach, R., Navot, A., and Tishby, N., "Query by committe made real," in *Advances in Neural Information Processing Systems 18*, pp. 443–450, MIT Press, 2005.

[64] Gill, P., Zhang, J., Woolley, S. M. N., Fremouw, T., and Theunissen, F. E., "Sound representation methods for spectro-temporal receptive field estimation.," *J Comput Neurosci*, vol. 21, pp. 5–20, Aug 2006.

[65] Gollisch, T., Schutze, H., Benda, J., and Herz, A. V. M., "Energy integration describes sound-intensity coding in an insect auditory system," *Journal of Neuroscience*, vol. 22, pp. 10434–10448, Dec. 2002.

[66] Gollisch, T. and Herz, A. V. M., "Disentangling sub-millisecond processes within an auditory transduction chain," *PLoS Biology*, vol. 3, 2005.

[67] Grace, J. A., Amin, N., Singh, N. C., and Theunissen, F. E., "Selectivity for conspecific song in the zebra finch auditory forebrain.," *J Neurophysiol*, vol. 89, pp. 472–487, Jan 2003.

[68] Gu, M. and Eisenstat, S. C., "A stable and efficient algorithm for the rank-one modification of the symmetrical eigenproblem," *SIAM Journal on Matrix Analysis and Applications*, vol. 15, pp. 1266–1276, Oct. 1994.

[69] Haberman, S., "Maximum likelihood estimation in exponential response models," *Annals of Statistics*, vol. 5, pp. 815–841, 1977.

[70] Hamada, M., Martz, H. F., Reese, C. S., and Wilson, A. G., "Finding near-optimal bayesian experimental designs via genetic algorithms," *AMERICAN STATISTICIAN*, vol. 55, pp. 175–181, Aug. 2001.

[71] Henderson, H. and Searle, S. R., "On deriving the inverse of a sum of matrices," *SIAM Review*, vol. 23, pp. 53–60, 1981.

[72] Hopfield, J. and Brody., C. D., "What is a moment? "cortical" sensory integration over a brief interval.," *PNAS*, vol. 97, pp. 13919–13924, December 2000.

[73] Hsu, A., Woolley, S. M. N., Fremouw, T. E., and Theunissen, F. E., "Modulation power and phase spectrum of natural sounds enhance neural encoding performed by single auditory neurons.," *J Neurosci*, vol. 24, pp. 9201–9211, Oct 2004.

[74] Hubel, D. H. and Wiesel, T. N., "Early exploration of the visual cortex.," *Neuron*, vol. 20, pp. 401–412, Mar 1998.

[75] Hubel, D. H. and Wiesel, T., "Receptive fields, binocular interaction and functional architecture in cats visual cortex," *Journal of Physiology-London*, vol. 160, no. 1, pp. 106–&, 1962.

[76] Hubel, D. H., "Evolution of ideas on the primary visual cortex, 1955-1978: A biased historical account." Nobel lecture, 8 December, 1981.

[77] Huk, A. C. and Shadlen, M. N., "Neural activity in macaque parietal cortex reflects temporal integration of visual motion signals during perceptual decision making.," *J Neurosci*, vol. 25, pp. 10420–10436, Nov 2005.

[78] Jewell, N. P. and Shiboski, S. C., "Statistical analysis of hiv infectivity based on partner studies.," *Biometrics*, vol. 46, pp. 1133–1150, Dec 1990.

[79] Kaelbling, L. P., Littman, M. L., and Moore, A. W., "Reinforcement learning: A survey," *Journal of Artificial Inteligence Research*, vol. 4, pp. 237–285, 1996.

[80] Keat, J., Reinagel, P., Reid, R. C., and Meister, M., "Predicting every spike: a model for the responses of visual neurons.," *Neuron*, vol. 30, pp. 803–817, Jun 2001.

[81] Khuri, A. I., Mukherjee, B., Sinha, B. K., and Ghosh, M., "Design issues for generalized linear models: A review," *Statistical Science*, vol. 21, pp. 376–399, 2006.

[82] Kiefer, J., "Optimum experimental designs," *Journal of the Royal Statistical Society. Series B.*, vol. 21, pp. 272–319, 1959.

[83] Kiefer, J. and Wolfowitz, J., "Optimum designs in regression problems," *Annals of Mathematical Statistics*, vol. 30, no. 2, pp. 271–294, 1959.

[84] Kontsevich, L. and Tyler, C., "Bayesian adaptive estimation of psychometric slope and threshold," *Vision Research*, vol. 39, pp. 2729–2737, 1999.

[85] Kuffler, S. W., "Neurons in the retina; organization, inhibition and excitation problems.," *Cold Spring Harb Symp Quant Biol*, vol. 17, pp. 281–292, 1952.

[86] Lee, J. M., *Introduction to Smooth Manifolds*. Springer, 2000.

[87] Lee, Y. and Nelder, J. A., "Analysis of ulcer data using hierarchical generalized linear models.," *Stat Med*, vol. 21, pp. 191–202, Jan 2002.

[88] Lesica, N. A. and Stanley, G. B., "Improved tracking of time-varying encoding properties of visual neurons by extended recursive least-squares," *IEEE Trans. On Neural Systems And Rehabilitation Engineering*, vol. 13, pp. 194–200, June 2005.

[89] Lewi, J., Butera, R., and Paninski, L., "Efficient active learning with generalized linear models," in *Proceedings of the Eleventh International Workshop on Artificial intelligence and Statistics* (Meila, M. and Shen, X., eds.), (San Juan), The Society for Artificial Intelligence and Statistics, March 21-24 2007.

[90] Lewi, J., Butera, R., and Paninski, L., "Designing neurophysiology experiments to optimally constrain parametric receptive field models.," in *submitted to Computational and Systems Neuroscience (COSYNE)*, 2008.

[91] Lewicki, M. S., "Efficient coding of natural sounds," *Nature Neuroscience*, vol. 5, pp. 356–363, Apr. 2002.

[92] Li, K. C. and Duan, N., "Regression-analysis under link violation," *Annals of Statistics*, vol. 17, pp. 1009–1052, Sept. 1989.

[93] Li, M. K. and Sethi, I. K., "Confidence-based active learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 1251–1261, Aug. 2006.

[94] Linden, J. F., Liu, R. C., Sahani, M., Schreiner, C. E., and Merzenich, M. M., "Spectrotemporal structure of receptive fields in areas ai and aaf of mouse auditory cortex," *Journal of Neurophysiology*, vol. 90, pp. 2660–2675, 2003.

[95] Lindley, D. V., "On a measure of the information provided by an experiment," *Annals of Mathematical Statistics*, vol. 27, no. 4, pp. 986–1005, 1956.

[96] Lindley, D. V., *Bayesian Statistics - A Review*. SIAM, 1972.

[97] Machens, C., "Adaptive sampling by information maximization," *Physical Review Letters*, vol. 88, pp. 228104–228107, 2002.

[98] Machens, C., Gollisch, T., Kolesnikova, O., and Herz, A., "Testing the efficiency of sensory coding with optimal stimulus ensembles," *Neuron*, vol. 47, no. 3, pp. 447–456, 2005.

[99] Mackay, D. J. C., "Information-based objective functions for active data selection," *Neural Computation*, vol. 4, pp. 590–604, July 1992.

[100] Mackay, D., *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

[101] Marler, P., "Birdsong and speech development: could there be parallels?," *American Scientist*, vol. 58, no. 6, pp. 669–673, 1970.

[102] Marshall-Bal, L. and Slater, P. J. B., "Duet singing and repertoire use in threat signalling of individuals and pairs.," *Proc Biol Sci*, vol. 271 Suppl 6, pp. S440–S443, Dec 2004.

[103] Mathew, T. and Sinha, B., "Optimal designs for binary data under logistic regression.," *Journal of Statistical PLanning and Inference.*, vol. 93, pp. 295–307, 2001.

[104] McCullagh, P. and Nelder, J., *Generalized linear models*. London: Chapman and Hall, 1989.

[105] McLeish, D. L., "Designing the future: A simple algorithm for sequential design of a generalized linear model," *Journal of Statistical Planning and Inference*, vol. 78, pp. 205–218, May 1999.

[106] Meister, M. and Berry, M., "The neural code of the retina," *NEURON*, vol. 22, no. 3, pp. 435–450, 1999.

[107] Minka, T. P., "Expectation propagation for approximate bayesian inference.," in *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*, pp. 362–369, Morgan Kaufmann, 2001.

[108] Minkin, S., "Experimental design for clonogenic assays in chemotherapy," *Journal of the American Statistical Association*, vol. 88, no. 422, pp. 410–420, 1993.

[109] Movellan, J. R., "Infomax control as a model of real time behavior: Theory and application to the detection of social contingency," Tech. Rep. 2005-1, University of California San Diego & ATR, Kyoto, 2005. Uses dynamic programming to solve the optimization problem.

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

[111] O'Connor, K. N., Petkov, C. I., and Sutter, M. L., "Adaptive stimulus optimization for auditory cortical neurons," *Journal of Neurophysiology*, vol. 94, pp. 4051–4067, Dec. 2005.

[112] Okanoya, K. and Dooling, R. J., "Hearing in passerine and psittacine birds: a comparative study of absolute and masked auditory thresholds.," *J Comp Psychol*, vol. 101, pp. 7–15, Mar 1987.

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

[114] Paninski, L., "Asymptotic theory of information-theoretic experimental design," *Neural Computation*, vol. 17, no. 7, pp. 1480–1507, 2005.

[115] Paninski, L., Shoham, S., Fellows, M. R., Hatsopoulos, N. G., and Donoghue, J. P., "Superlinear population encoding of dynamic hand trajectory in primary motor cortex," *Journal of Neuroscience*, vol. 24, pp. 8551–8561, Sept. 2004.

[116] Paninski, L., Pillow, J., and Lewi, J., *Computational Neuroscience: Theoretical Insights into Brain Function*, ch. Statistical models for neural encoding, decoding, and optimal stimulus design. Elsevier, 2007.

[117] Patterson, R., Robinson, K., Holdsworth, J., McKeown, D., Zhang, C., and Allerhand, M., "Complex sounds and auditory images," in *Auditory physiology and perception, Proceedings 9th International Symposium on Hearing*, pp. 429–446, Elsevier, 1992.

[118] Pillow, J. W., Paninski, L., Uzzell, V. J., Simoncelli, E. P., and Chichilnisky, E. J., "Prediction and decoding of retinal ganglion cell responses with a probabilistic spiking model," *J. Neurosci.*, vol. 25, no. 47, pp. 11003–11013, 2005.

[119] Pillow, J. W. and Simoncelli, E. P., "Dimensionality reduction in neural models: an information-theoretic generalization of spike-triggered average and covariance analysis.," *J Vis*, vol. 6, no. 4, pp. 414–428, 2006.

[120] Qiu, A., Schreiner, C. E., and Escab, M. A., "Gabor analysis of auditory midbrain receptive fields: spectro-temporal and binaural composition.," *J Neurophysiol*, vol. 90, pp. 456–476, Jul 2003.

[121] Rasmussen, C. E. and Williams, C. K. I., *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[122] Rieke, F., Bodnar, D. A., and Bialek, W., "Naturalistic stimuli increase the rate and efficiency of information transmission by primary auditory afferents.," *Proc Biol Sci*, vol. 262, pp. 259–265, Dec 1995.

[123] Rieke, F., Warland, D., de Ruyter van Steveninck, R., and Bialek, W., *Spikes: Exploring the neural code*. Cambridge: MIT Press, 1997.

[124] Ringach, D. L., "Mapping receptive fields in primary visual cortex," *Journal of Physiology-London*, vol. 558, pp. 717–728, Aug. 2004.

[125] Ringach, D. L., Sapiro, G., and Shapley, R., "A subspace reverse-correlation technique for the study of visual neurons.," *Vision Res*, vol. 37, pp. 2455–2464, Sep 1997.

[126] Ringach, D. L., "Spatial structure and symmetry of simple-cell receptive fields in macaque primary visual cortex.," *J Neurophysiol*, vol. 88, pp. 455–463, Jul 2002.

[127] Ringach, D. L., Hawken, M. J., and Shapley, R., "Receptive field structure of neurons in monkey primary visual cortex revealed by stimulation with natural image sequences.," *J Vis*, vol. 2, no. 1, pp. 12–24, 2002.

[128] Robert, C. P. and Casella, G., *Monte Carlo Statistical Methods*. Springer, 2004.

[129] Rosenberger, W. F. and Hu, M. X., "On the use of generalized linear models following a sequential design," *Statistics & Probability Letters*, vol. 56, pp. 155–161, Jan. 2002.

[130] Roy, A., Ghosal, S., and Rosenberger, W. F., "Convergence properties of sequential bayesian d-optimal designs with applications to phase i clinical trials," *Journal of Statistical Planning and Inference, Accepted*, 2007.

[131] Rust, N. C., Mante, V., Simoncelli, E. P., and Movshon, J. A., "How mt cells analyze the motion of visual patterns," *Nature Neuroscience*, vol. 9, pp. 1421–1431, Nov. 2006.

[132] Schein, A., *Active Learning For Logistic Regression*. PhD thesis, University of Pennsylvania, 2005.

[133] Seeger, M., Gerwinn, S., and Bethge, M., "Bayesian inference for sparse generalized linear models," in *Machine Learning: ECML 2007*, SpringerLink, 2007.

[134] Seeger, M., Steinke, F., and Tsuda, K., "Bayesian inference and optimal design in the sparse linear model," in *Proceedings of the Eleventh International Workshop on Artificial intelligence and Statistics*, The Society for Artificial Intelligence and Statistics, 2007.

[135] Seeger, M., "Low rank updates for the cholesky decomposition," tech. rep., Berkeley, 2007.

[136] Sen, K., Theunissen, F. E., and Doupe, A. J., "Feature analysis of natural sounds in the songbird auditory forebrain," *Journal of Neurophysiology*, vol. 86, pp. 1445–1458, Sept. 2001.

[137] Sharia, T., "Recursive parameter estimation: Asymptotic expansion," tech. rep., arXiv:0705.1783, 2007.

[138] Sharpee, T., Rust, N. C., and Bialek, W., "Analyzing neural responses to natural signals: Maximally informative dimensions," *Neural Computation*, vol. 16, pp. 223–250, Feb. 2004.

[139] Simoncelli, E., Paninski, L., Pillow, J., and Schwartz, O., "Characterization of neural responses with stochastic stimuli," in *The Cognitive Neurosciences* (Gazzaniga, M., ed.), MIT Press, 2004.

[140] Singh, N. C. and Theunissen, F. E., "Modulation spectra of natural sounds and ethological theories of auditory processing," *The Journal of the Acoustical Society of America*, vol. 114, no. 6, pp. 3394–3411, 2003.

[141] Smirnakis, S. M., Berry, M. J., Warland, D. K., Bialek, W., and Meister, M., "Adaptation of retinal processing to image contrast and spatial scale.," *Nature*, vol. 386, pp. 69–73, Mar 1997.

[142] Smith, D. M. and Ridout, M., "Algorithms for finding locally and bayesian optimal designs for binary dose-response models with control mortality.," *Journal of Statistical Planning and Inference*, vol. 133, pp. 463–478, 2005.

[143] Smith, E. C. and Lewicki, M. S., "Efficient auditory coding," *Nature*, vol. 439, pp. 978–982, Feb. 2006.

[144] Smyth, D., Willmore, B., Baker, G. E., Thompson, I. D., and Tolhurst, D. J., "The receptive-field organization of simple cells in primary visual cortex of ferrets under natural scene stimulation.," *J Neurosci*, vol. 23, pp. 4746–4759, Jun 2003.

[145] Stanley, G. B., *Neural Engineering*, ch. Neural System Identification. Springer, 2005.

[146] Stanley, G. B. and Webber, R. M., "A point process analysis of sensory encoding.," *J Comput Neurosci*, vol. 15, no. 3, pp. 321–333, 2003.

[147] Steinberg, D. M. and Hunter, W. G., "Experimental design: Review and comment," *Technometrics*, vol. 26, 1984.

[148] Takadama, K. and Shimohara, K., "Exploration and exploitation trade-off in multiagent learning," in *ICCIMA '01: Proceedings of the Fourth International Conference on Computational Intelligence and Multimedia Applications*, (Washington, DC, USA), p. 133, IEEE Computer Society, 2001.

[149] Theunissen, F. E., David, S. V., Singh, N. C., Hsu, A., Vinje, W. E., and Gallant, J. L., "Estimating spatio-temporal receptive fields of auditory and visual neurons from their responses to natural stimuli," *Network-Computation in Neural Systems*, vol. 12, pp. 289–316, Aug. 2001.

[150] Theunissen, F. E. and Doupe, A. J., "Temporal and spectral sensitivity of complex auditory neurons in the nucleus hvc of male zebra finches.," *J Neurosci*, vol. 18, pp. 3786–3802, May 1998.

[151] Theunissen, F. E., Sen, K., and Doupe, A. J., "Spectral-temporal receptive fields of nonlinear auditory neurons obtained using natural sounds," *Journal of Neuroscience*, vol. 20, pp. 2315–2331, Mar. 2000.

[152] Theunissen, F. E., Woolley, S. M. N., Hsu, A., and Fremouw, T., "Methods for the analysis of auditory processing in the brain.," *Ann N Y Acad Sci*, vol. 1016, pp. 187–207, Jun 2004.

[153] Todorov, E., *Bayesian Brain*, ch. Optimal Control Theory. MIT Press, 2006.

[154] Tolhurst, D. J. and Dean, A. F., "The effects of contrast on the linearity of spatial summation of simple cells in the cat's striate cortex.," *Exp Brain Res*, vol. 79, no. 3, pp. 582–588, 1990.

[155] Tong, S., *Active Learning Theory and Apllications*. PhD thesis, Stanford University, 2001.

[156] Truccolo, W., Eden, U. T., Fellows, M. R., Donoghue, J. P., and Brown, E. N., "A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects," *Journal of Neurophysiology*, vol. 93, pp. 1074–1089, Feb. 2005.

[157] van der Vaart, A., *Asymptotic statistics*. Cambridge: Cambridge University Press, 1998.

[158] Vickers, N. J., Christensen, T. A., Baker, T. C., and Hildebrand, J. G., "Odour-plume dynamics influence the brain's olfactory code.," *Nature*, vol. 410, pp. 466–470, Mar 2001.

[159] Warmuth, M. K., Liao, J., Ratsch, G., Mathieson, M., Putta, S., and Lemmen, C., "Active learning with support vector machines in the drug discovery process," *Journal of Chemical Information and Computer Sciences*, vol. 43, no. 2, pp. 667–673, 2003.

[160] Watson, A. and Pelli, D., "Quest: a bayesian adaptive psychophysical method," *Perception and Psychophysics*, vol. 33, pp. 113–120, 1983.

[161] Wedderburn, R., "On the existence and uniqueness of the maximum likelihood estimator for certain generalized linear models," *Biometrika*, vol. 63, pp. 27–32, 1976.

[162] Weliky, M., Fiser, J., Hunt, R. H., and Wagner, D. N., "Coding of natural scenes in primary visual cortex.," *Neuron*, vol. 37, pp. 703–718, Feb 2003.

[163] Williams, H., "Birdsong and singing behavior.," *Ann N Y Acad Sci*, vol. 1016, pp. 1–30, Jun 2004.

[164] Woods, D. C., Lewis, S. M., Eccleston, J. A., and Russell, K. G., "Designs for generalized linear models with several variables and model uncertainty," *Tehnometrics*, vol. 48, pp. 284–292, 2006.

[165] Woolley, S. M. N. and Casseday, J. H., "Response properties of single neurons in the zebra finch auditory midbrain: response patterns, frequency coding, intensity coding, and spike latencies.," *J Neurophysiol*, vol. 91, pp. 136–151, Jan 2004.

[166] Woolley, S. M. N. and Casseday, J. H., "Processing of modulated sounds in the zebra finch auditory midbrain: responses to noise, frequency sweeps, and sinusoidal amplitude modulations.," *J Neurophysiol*, vol. 94, pp. 1143–1157, Aug 2005.

[167] Woolley, S. M. N., Fremouw, T. E., Hsu, A., and Theunissen, F. E., "Tuning for spectro-temporal modulations as a mechanism for auditory discrimination of natural sounds.," *Nat Neurosci*, vol. 8, pp. 1371–1379, Oct 2005.

[168] Woolley, S. M., Gill, P. R., and Theunissen, F. E., "Stimulus-dependent auditory tuning results in synchronous population coding of vocalizations in the songbird midbrain," *The Journal of Neuroscience*, vol. 26, pp. 2499–2512, 2006.

[169] Wu, C. F. J., "Asymptotic inference from sequential design in a nonlinear situation," *Biometrika*, vol. 72, no. 3, pp. 553–558, 1985.

[170] Wu, M. C. K., David, S. V., and Gallant, J. L., "Complete functional characterization of sensory neurons by system identification," *Annual Review of Neuroscience*, vol. 29, pp. 477–505, 2006.

[171] Yan, Z. W., Bate, S., Chandler, R. E., ISham, V., and Wheater, H., "An analysis of daily maximum wind speed in northwestern europ using generalized linear models.," *Journal of Climate*, vol. 15, pp. 2073–2088, 2002.

[172] Zevin, J. D., Seidenberg, M. S., and Bottjer, S. W., "Limits on reacquisition of song in adult zebra finches exposed to white noise.," *J Neurosci*, vol. 24, pp. 5849–5862, Jun 2004.

[173] Zhang, K., Anderson, M., and Young, E., "Saddle-point property of nonlinear sensory response," in *Proceedings*, Computational and Systems Neuroscience Meeting, 2004.