UNIVERSITY OF CALIFORNIA, SAN DIEGO

Characterizing neural responses to natural stimuli

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Physics (Biophysics)

by

Ryan John Rowekamp

Committee in charge:

Professor Tatyana O. Sharpee, Chair
Professor Henry Abarbanel, Co-Chair
Professor Michael Anderson
Professor Timothy Gentner
Professor Terrence Sejnowski

2014

The dissertation of Ryan John Rowekamp is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

____

____

____

____
Co-Chair

____
Chair

University of California, San Diego

2014

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

t	time
N	number of frames
N_{sp}	number of spikes
B	number of histogram bins per dimension
C	covariance matrix
C_p	prior (stimulus) covariance matrix
C_{sp}	spike-triggered covariance matrix
ΔC	$C_{sp} - C_p$
y_t	response at time t
$\bar{y}$	mean firing rate
$\hat{y}$	predicted firing rate
$\mathbf{s}_t$	full stimulus at time t
D	dimensionality of full stimulus $\mathbf{s}$
D_t	length of model's memory
K	Number of relevant stimulus dimensions
B	Number of bins per dimension used to calculate mutual information
$\hat{e}_k$	kth relevant stimulus dimension
s_k	the projection of $\mathbf{s}$ on to $\hat{e}_k$
$\mathbf{v}_k$	the estimated kth relevant dimension
$\mathbf{x}$	reduced stimulus

x_k	the kth coordinate of the reduced stimulus
$f()$	general nonlinear function
V	matrix containing the vectosrs $\mathbf{v}_1$ through $\mathbf{v}_K$
$\mathbf{z}$	location of model subunits
G	set of locations $\mathbf{z}$
$T_{\mathbf{z}}$	translation matrix, translates by $\mathbf{z}$
$\mathbf{s}_{\mathbf{z}}$	$T_{\mathbf{z}}\mathbf{s}$, the stimulus translated by $\mathbf{z}$
$\sigma()$	logistic function
a	bias term for QMNE or ILS
$\mathbf{v}$	linear filter for QMNE, QMID, and ILS
J	quadratic filter for QMNE, QMID, and ILS
C_{PCA}	covariance matrix of a set of vectors to be averavged
λ_k	eigenvalue associated with the kth eigenvector
N_{jack}	number of jackknifes
ϕ	phase
$R_{+}()$	softplus rectifier function

LIST OF ABBREVIATIONS

LN	linear-nonlinear
LNP	linear-nonlinear poisson
STA	spike-triggered average
STC	spike-triggered covariance
PPR	projection pursuit regression
MSE	mean squared error
ePPR	extended projection pursuit regression
MID	maximally informative dimensions
SMID	serial MID
IMID	invariant MID
ME	maximum entropy
MNE	maximum noise entropy
QMID	quadratic MID
QMNE	quadratic MNE
RGC	retinal ganglion cell
LGN	lateral geniculate nucleus
V1	primary visual cortex
V4	visual area V4
PCA	principle component analysis

ACKNOWLEDGEMENTS

Chapter 2 contains work that was published in Rowekamp and Sharpee (2011). The dissertation author was the primary investigator and author of this paper.

Chapter 3 contains work that was published in Eickenberg et al. (2012). The dissertation author was the co-primary investigator and author of this paper.

Chapter 4 contains work that was published in Fitzgerald et al. (2011b). The dissertation author was the primary investigator for the portion of the work presented in the dissertation and was an author of this paper.

Chapter 5 contains work that is being prepared for submission for publication of the material by Ryan Rowekamp and Tatyana O. Sharpee. The dissertation author was the primary investigator and author of this material. I would also like to acknowledge Minjoon Kouh, Jude Mitchell, and John Reynolds for allowing us to analyze their data.

VITA

2005	Bachelor of Arts in Mathematics *magna cum laude*, University of Saint Thomas, Saint Paul, Minnesota
2005	Bachelor of Science in Physics *magna cum laude*, University of Saint Thomas, Saint Paul, Minnesota
2007	Master of Science in Physics, University of California, San Diego, California
2014	Doctor of Philosophy in Physics (Biophysics), University of California, San Diego, California

PUBLICATIONS

Eickenberg, M.; Rowekamp, R.J.; Kouh, M.; Sharpee, T.O.; Characterizing Responses of Translation-Invariant Neurons to Natural Stimuli: Maximally Informative Invariant Dimensions, Neural Computation, 24(9), 2384-2421, 2012.

Fitzgerald, J.D.; Rowekamp, R.J.; Sincich, L.C.; Sharpee, T.O.; Second order dimensionality reduction using minimum and maximum mutual information models, PLoS Computational Biology, 7(10): e1002249, 2011.

Rowekamp, R.J.; Sharpee, T.O.; Analyzing multicomponent receptive fields from neural responses to natural stimuli, Network: Computation in Neural Systems, Vol. 22(1-4), 45-73, 2011.

ABSTRACT OF THE DISSERTATION

Characterizing neural responses to natural stimuli

by

Ryan John Rowekamp

Doctor of Philosophy in Physics (Biophysics)

University of California, San Diego, 2014

Professor Tatyana O. Sharpee, Chair
Professor Henry Abarbanel, Co-Chair

The sensory nervous system converts external stimuli into electrical signals that are used to process and transmit information about the stimuli. An ongoing goal of systems neuroscience is to describe the processing of stimuli as compactly as possible using a small number of features from the stimulus, which is known as dimensionality reduction. This task is especially difficult when analyzing stimuli with complex correlations between dimensions as is found in natural stimuli. This dissertation begins by presenting Maximally Informative Dimensions (MID), which selects the features that modulate the neuron's responses the stimuli. It then presents three variants of this method that seek to address specific limitations of the method. Sequential Maximally Informative Dimensions seeks to perform this analysis without calculating multidimensional probability distributions. Invariant Maximally Informative Dimensions allows simplified analysis of neurons that respond to similar but offset features. Quadratic Maximally Informative Dimensions incorporates quadratic features to allow one to find many linear features. Finally, Invariant Logistical Subunits combines the ideas of Invariant Maximally Informative Dimensions and Quadratic Maximally Informative Dimensions in a more flexible manner.

Chapter 1

Introduction to neural coding and dimensionality reduction

The fundamental task of the nervous system is to capture information about the internal and external environment, process these stimuli into a relevant form, and make behavioral and physiological responses to the state of the environment. This collective behavior is the result of the computations performed by individual nervous cells called neurons. Sensory neurons convert stimuli into changes in the electrical potential across the cell membrane. If the neuron's potential depolarizes sufficiently, the nonlinear response of specialized ion channels creates a characteristic voltage fluctuation called an action potential or a spike. The spike propagates to the neuron's synaptic connections with downstream neurons, where it releases neural transmitters that bind with receptors on the postsynaptic neuron which open ion channels and change the electrical potential. Through networked repetitions of this process, the nervous system is capable of performing its full range of observed computations.

One of the goals of systems neuroscience is to characterize the properties of these computations. One way to accomplish this is to build a model of the system that can accurately predict the responses of a neuron to novel stimuli. In the general case of the firing rate model, a neuron's response can be described using

$$y_t = f\left(\mathbf{s}_t\right), \tag{1.1}$$

where y_t is the average firing rate at time t, $\mathbf{s}_t$ is a vector of all stimulus variables that determine the firing rate at t (possibly including the neuron's recent response history), and f is a non-linear function that converts $\mathbf{s}$ into a firing rate. Unfortunately, characterizing a neuron's response as a general function of even a moderate number of stimulus dimensions is impractical with realistic amounts of data, so one must make some simplifying assumptions.

One frequent assumption is that the neuron's response is primarily determined by a subspace of the stimulus $\mathbf{x}$ with a dimensionality K that is much smaller than the dimensionality D of the full stimulus $\mathbf{s}$. Once the stimulus is in the reduced space, it is easier to characterize the the nonlinearity parametrically or by using Bayes' theorem

$$f(\mathbf{s}) = P(y|\mathbf{x}(\mathbf{s})) = P(y)\frac{P(\mathbf{x}|y)}{P(\mathbf{x})}. \tag{1.2}$$

1.1 Linear-nonlinear model

The linear-nonlinear model (LN) (de Boer and Kuyper, 1968) makes one further assumption: the reduced subspace is a linear transformation of the stimulus

$$\mathbf{x} = V^T\mathbf{s}. \tag{1.3}$$

V is a D by K matrix containing the vectors $\mathbf{v}_1$ through $\mathbf{v}_K$ that represent the dimensions of the stimulus space corresponding to the coordinates x_1 through x_K. With this assumption we can define the distribution of the reduced stimulus as

$$P(\mathbf{x}) = \int d\mathbf{x} P(\mathbf{s}) \prod_{i=1}^{K} \delta(x_i - \mathbf{v}_i^T\mathbf{s}). \tag{1.4}$$

$P(\mathbf{x}|y)$ is defined similarly using the conditional stimulus distribution:

$$P(\mathbf{x}|y) = \int d\mathbf{x} P(\mathbf{s}|y) \prod_{i=1}^{K} \delta(x_i - \mathbf{v}_i^T\mathbf{s}). \tag{1.5}$$

Note that it is the reduced subspace that matters rather than a particular choice of V. Any set of vectors that span the subspace define a functionally equivalent system of coordinates. Given a set of vectors V, we can define any other description of the subspace as $V_L = VL$, where L is a K by K non-degenerate linear transformation. Given L and $f(\mathbf{x})$, it is trivial to calculate $f_L(\mathbf{x}_L) = f(L^{-1}\mathbf{x}_L)$.

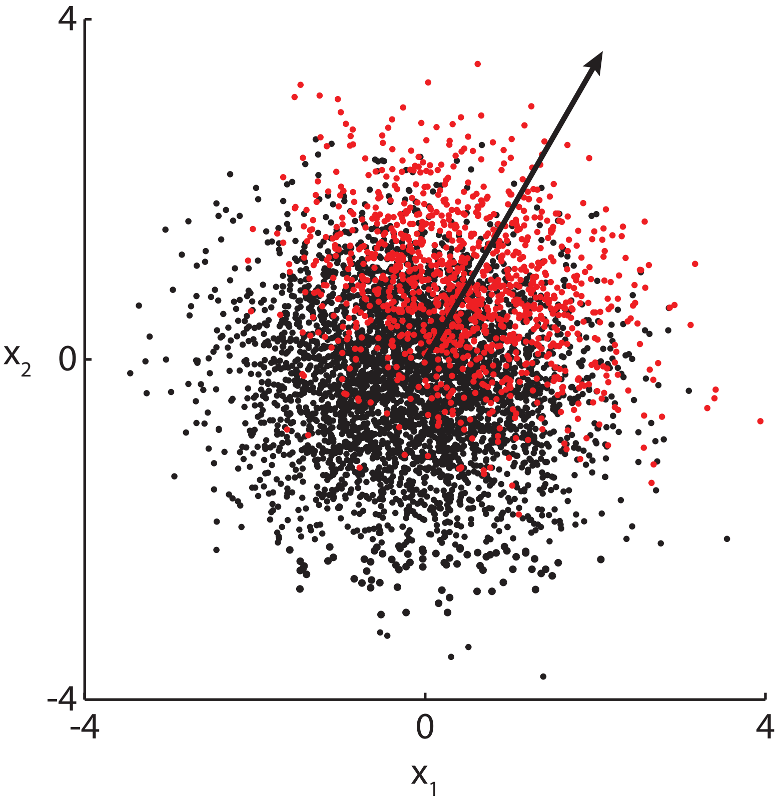

Figure 1.1: **STA demonstration.** Each point is a stimulus sampled from a two-dimensional normal distribution. The probability of a spike is determined by a sigmoid function of a single dimension. Black points indicate no spike, and red points indicate a spike. The black arrow points along the direction of the difference between the stimulus average and the spike-triggered average. The STA correctly selects the dimension used to generate the spike probability.

1.2 Spike-triggered average

Given the LN model, the question arises of how to find V. One simple and popular technique is the spike-triggered average (STA), also known as reverse correlation (de Boer and Kuyper, 1968). Given a set of N stimuli $\left\{\mathbf{s}^{(t)}\right\}$ and the corresponding responses $\left\{y^{(t)}\right\}$, the spike-triggered average is

$$\mathbf{v}_{STA} = \frac{1}{N}\sum_{t=1}^{N} \mathbf{s}_t y_t - \langle s \rangle \langle y \rangle \,. \tag{1.6}$$

Fig. 1.1 shows a demonstration of STA. The black and red points indicate a stimulus associated with no spike and a spike, respectively. The black arrow points from the stimulus mean to the mean of the stimuli associated with a spike. The STA corresponds with the dimension used to determine the probability of spiking.

The advantage of the STA is its simplicity. However, it is only able to find a single dimension and it is only consistent with radially symmetric stimulus distributions and nonlinearities with $\langle x_1 | \mathrm{y} \rangle \neq 0$ (Paninski, 2003). If the stimulus is elliptically symmetric, the correlations can be rotated out using the inverse of the covariance matrix to whiten the stimulus

$$\mathbf{v}_{RSTA} = \left(\frac{1}{N} \sum_{t=1}^{N} \mathbf{s}_t \mathbf{s}_t^T - \langle \mathbf{s} \rangle \langle \mathbf{s} \rangle^T + \lambda I \right)^{-1} \mathbf{v}_{STA}, \tag{1.7}$$

where λ is a regularization parameter dampens the effects of noise on the inverse covariance matrix.

1.3 Spike-triggered covariance

Spike-triggered covariance (STC) is an extension of STA that looks at the second-order moments of the spike-triggered distribution. Given the stimulus covariance

$$C_p = \left\langle \mathbf{s}\mathbf{s}^T \right\rangle - \langle \mathbf{s} \rangle \langle \mathbf{s} \rangle^T = \frac{1}{N-1} \sum_{t=1}^{N} \mathbf{s}_t \mathbf{s}_t^T - \langle \mathbf{s} \rangle \langle \mathbf{s} \rangle^T \tag{1.8}$$

and the spike-triggered stimulus covariance

$$C_{sp} = \left\langle \mathbf{s}\mathbf{s}^T | \text{spike} \right\rangle - \langle \mathbf{s} | \text{spike} \rangle \langle \mathbf{s} | \text{spike} \rangle^T = \frac{1}{N_{sp}} \sum_{t=1}^{N} \mathbf{s}_t \mathbf{s}_t^T y_t - \langle \mathbf{s} | \text{spike} \rangle \langle \mathbf{s} | \text{spike} \rangle^T, \tag{1.9}$$

the relevant dimensions are the eigenvectors of the difference ΔC of C_{sp} and C_p whose eigenvalues are significantly different from zero (de Ruyter van Steveninck and Bialek, 1988; Schwartz et al., 2002). Positive eigenvalues indicate that responses are associated with large values along that eigenvector. Because large magnitudes along this dimensions are associated with higher responses, these dimensions are excitatory. Negative eigenvalues indicate that responses are correlated with small magnitudes along that dimension. These dimensions are inhibitory.

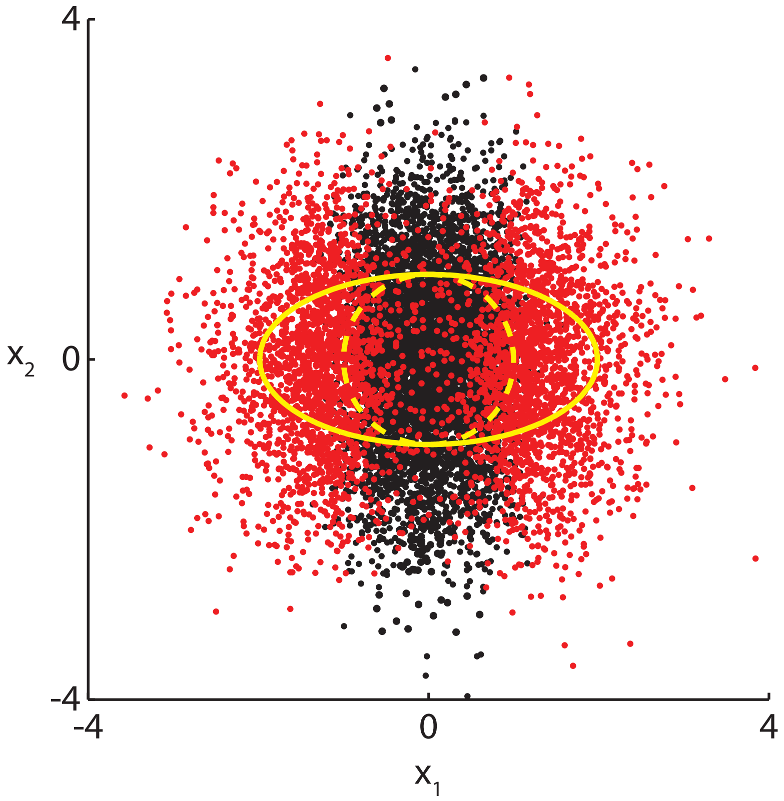

Figure 1.2: **STC demonstration.** Black points indicate no spike. Red points indicate a spike. Because of the symmetric nonlinearity, the STA is approximately zero. The dashed yellow line shows the stimulus covariance at 1 standard deviation. The solid yellow line shows the spike-triggered covariance. Along x_2, the irrelevant dimension, the covariances are equal. Along x_1, the dimension used to generate the probability of a spike, the covariances are maximally separated.

Fig. 1.2 shows STC for a simple example. Because the nonlinearity is symmetric, the stimulus and spike-triggered averages are the same, and therefore STA cannot reveal the relevant dimension. However, the separation between the stimulus covariance (dashed line) and the spike-triggered covariance (solid line) reveals that large magnitudes of x_1 are associated with spikes and x_2 is irrelevant to spiking.

If the stimulus has correlations, these correlations will show up as biases in the eigenvectors of ΔC. To compensate for second–order correlations, we can multiply the eigenvectors by the inverse of the stimulus covariance matrix C_p^{-1}.

This can completely correct the biases due to a correlated Gaussian stimuli, which has statistics completely determined by the first- and second-order correlations. For other distributions, biases due to higher-order correlations remain.

In the case of finite data, this whitening procedure can amplify the noise for poorly sampled dimensions. To reduce this effect, one can regularize the procedure by replacing C_p^{-1} with $(C_p^{-1} + \lambda I)^{-1}$. Increasing the regularization λ reduces the effect the dimensions with the smallest eigenvalues and approaches the limit of no whitening as $\lambda \to \infty$.

1.4 Statistical requirements of spike-triggered methods

Errors in the estimates of the relevant subspace can come from two sources: sampling error and the bias of the method. Sampling error occurs because random samples of a distribution can happen to have statistics that differ from those of the distribution. These errors decrease as the amount of data increases and the probable magnitude of these deviations shrink. Bias is more serious because it results in errors that cannot be reduced with additional data.

In order to produce unbiased estimates of the relevant subspace, both STA and STC have requirements on the stimulus and the system they are attempting to characterize (Paninski, 2003).

STA has three requirements:

1. The dimensionality of the relevant subspace (K) must be 1. STA only provides one dimension, so it is unable to provide any estimate of additional dimensions.

2. $P(\mathbf{s})$ (or in the case of RSTA, $P(C_p^{-1/2}\mathbf{s})$) must be radially symmetric. This includes the Gaussian distribution, but any radially symmetric is consistent with STA.

3. $\langle s_i y \rangle \neq \langle s_i \rangle \langle y \rangle$. If the the two sides of the equation are equal, the expected value of $\mathbf{v}_{STA}$ will be zero and the measured value will be determined by

sampling error. This situation can occur when the nonlinearity is symmetric, such as with an energy model where $\langle y(s_i) \rangle \propto s_i^2$.

STC has two requirements:

1. $P(\mathbf{s})$ is Gaussian. This is a more strict requirement than the required radial symmetry of STA.

2. $\langle s_i^2 \rangle - \langle s_i \rangle^2 \neq \langle s_i^2 | \text{spike} \rangle - \langle s_i | \text{spike} \rangle^2$ for every dimension in some orthogonal basis of the relevant subspace.

The limitations on the stimulus distribution are probably the most relevant. Firstly, it prevents the study of adaptation of statistics other than the mean and variance. Secondly, efficient coding predicts that neural systems will be tuned via evolution and development to the statistics of natural stimuli and may not respond robustly to non-natural Gaussian distributions (Simoncelli and Olshausen, 2001). This motivates us to find methods to characterize neural responses to arbitrary stimulus distributions.

Chapter 2

Maximally informative dimensions

STA (Sec. 1.2) and STC (Sec. 1.3) provide biased estimates of the relevant subspace when applied to stimuli that do not meet a restrictive set of requirements. If we wish to probe the responses of neurons to other types of stimuli (including the stimuli animals experience in nature), we need an algorithm that can provide accurate reconstructions of the relevant subspace for stimuli with arbitrary statistics.

We turned to entropy to solve this problem. Entropy, given by the formula

$$H\left(X\right)=\int dX P\left(X\right)\log_2\left(P\left(X\right)\right), \tag{2.1}$$

is a measure of the uncertainty about some measurement. For example, a fair coin that has a 50% chance of landing heads-up and 50% chance of landing tails-up would have an entropy of 1 bit per coin toss. If we were to modify the coin so that it was more likely to land on one side rather than the other, this would reduce the entropy because we have more knowledge about the outcome. In the extreme case where the coin always lands on one side (or both sides have the same markings), the entropy would be 0 bits.

Mutual information, defined as

$$I\left(X;Y\right)=H\left(X\right)-H\left(X|Y\right), \tag{2.2}$$

measures the change in entropy of one measurement when we have access to a second measurement. If Y is caused by X, X causes Y, or both X and Y are caused by one or more other variables, knowing the value of Y can tell us something about the distribution of values of X, which would give us a positive mutual information. Otherwise, the value of Y tells us nothing about the value of X and the mutual information is zero.

Maximally informative dimensions (MID) finds stimulus dimensions that affect the neuron's response (the relevant dimensions) by selecting the stimulus dimensions that maximize the mutual information between the projections into the reduced stimulus space and the neuron's response (Sharpee et al., 2003, 2004). The mutual information is given by

$$I_V = \sum_{y=\{0,1\}} P(y) \int d\mathbf{x} P_V(\mathbf{x}|y) \log_2 \left(\frac{P_V(\mathbf{x}|y)}{P_V(\mathbf{x})} \right) \tag{2.3}$$

where $\mathbf{v}$ is the dimension of interest and x is the projection of the stimulus onto that dimension.

This can be simplified and extended to the case of a non-binary stimulus by taking the limit of describing the spike train with sufficiently fine temporal resolution such that all time bins contain either 0 or 1 spike. In this case,

$$\lim_{\Delta T \to 0} P_V(\mathbf{x}|y=0) = P_V(\mathbf{x}) \tag{2.4}$$

because the number of bins without spikes is much greater than the number of bins with spikes and therefore removing the bins with spikes from the distribution has a negligible effect on the distribution. Because the fraction within the logarithm is now approximately 1, the non-spiking term does not contribute to the information. What remains is the information per spike

$$I_V = \int d\mathbf{x} P_V(\mathbf{x}|\text{spike}) \log_2 \left(\frac{P_V(\mathbf{x}|\text{spike})}{P_V(\mathbf{x})} \right). \tag{2.5}$$

Fig 2.1 gives a visual demonstration of how MID is able to distinguish between relevant and irrelevant dimensions. Fig. 2.1A shows the computation of a model with a two-dimensional Gaussian stimulus. Whether a stimulus sample is associated with a spike is determined by the nonlinearity

$$P(\text{spike}|\mathbf{x}) = \sigma(2 * x_1 - 1). \tag{2.6}$$

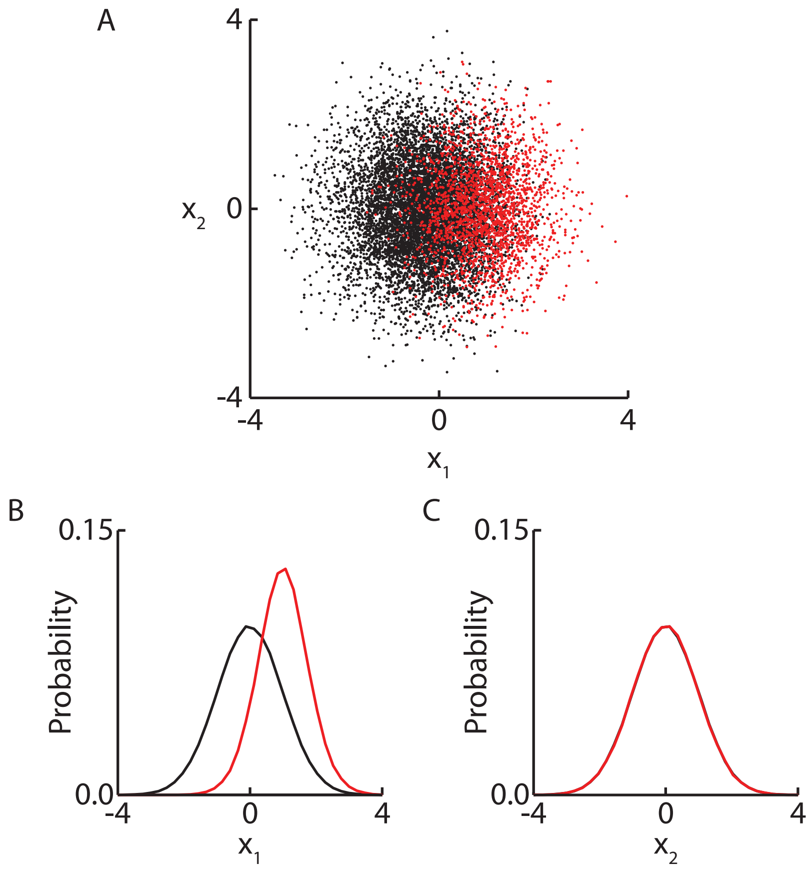

Figure 2.1: **Demonstration of MID.** (**A**) A simple model cell with a two-dimensional Gaussian stimulus. The probability of a spike depends only on the position along x_1. $10,000$ example stimulus points are shown with the color indicating whether the cell spiked (red) or was silent (black). (**B**) The distributions of stimuli and stimuli associated with a spike along the x_1. Because the value of x_1 determines the probability of a spike, the distribution of stimuli associated with a spike($P(x_1|\text{spike})$, red) diverges from the full distribution of stimuli ($P(x_1)$, black). The information explained is at a maximum for x_1 because it was used to generate the spikes and the data processing inequality (Sec. 2.1) prevents transforming from x_1 to another x from resulting in an increase in information. (**C**) The distributions along x_2. Because the response of the model cell does not depend on x_2, $P(x_2)$ and $P(x_2|\text{spike})$ are identical and the information per spike explained by x_2 is zero.

Stimuli associated with a spike are red while stimuli associated with no spike are black.

Fig. 2.1B shows $P(x_1)$ (black) and $P(x_1|\text{spike})$. Because the probability distributions are different, the logarithm in Eq. 2.5 will be non-zero and therefore x_1 explains some of the information in the spiking. Because this dimension was used to generate the spikes, it will explain more information than any other dimension and is therefore the maximally informative dimension. In contrast, Fig. 2.1C shows $P(x_2)$ and $P(x_2|\text{spike})$. Because x_2 was not used to generate the spikes and is not correlated with the dimension x_1 that was, these distributions are identical. Therefore, the information about the spikes explained by x_2 is 0.

The model above could have been successfully analyzed with methods such as STA. The power of MID is its ability to distinguish between the dimensions that are directly related to whether the neuron has spiked and the dimensions that are merely correlated with those dimensions. Fig. 2.2A shows what happens when the correlation between x_1 and x_2 is 0.8 rather than 0.0 like in Fig. 2.1. In this case, the probability distributions along the x_2 are no longer identical even thought the dimension is orthogonal to the relevant dimension x_1. However, x_2 only explains 0.51 bits of information compared to the 0.85 bits of information explained by x_1. In this case, the STA (Fig. 2.2A, arrow) is not the relevant dimension. The dot product with x_1 is 0.78. MID correctly rejects this estimate in favor of x_1 because it only explains 0.77 bits of information about the spikes. Note that because the stimulus in this example is normally distributed, STA and STC could compensate for the correlations (as described in Sec. 1.2 and Sec. 1.3). However, MID is able to overcome the correlations in the stimulus without making any assumptions about their structure.

To improve the estimate of the relevant dimensions, MID takes the gradient of (2.5) with respect to the current estimate of the dimensions V

$$\nabla_{\mathbf{v}_i} I_V = \int d\mathbf{x} P_V(\mathbf{x}|\text{spike})(\langle \mathbf{s}|\mathbf{x}\rangle_V - \langle \mathbf{s}|\mathbf{x}, \text{spike}\rangle_V)\frac{d}{dx_i}\left(\frac{P_V(\mathbf{x}|\text{spike})}{P_V(\mathbf{x})}\right). \tag{2.7}$$

A full derivation of the gradient can be found in Appendix A.

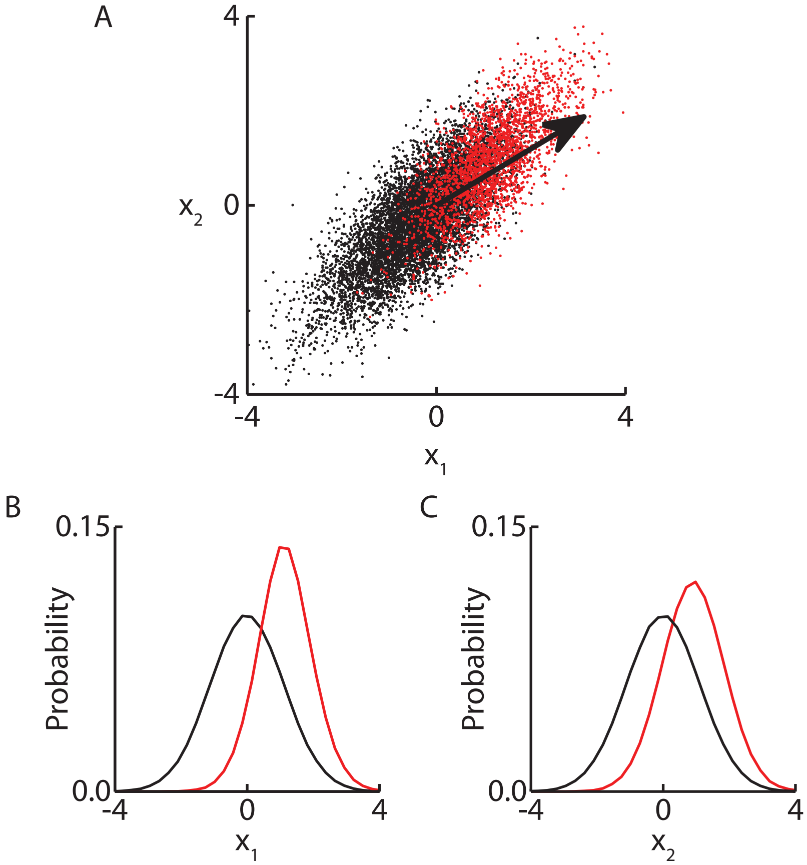

Figure 2.2: **MID demonstration with correlated stimulus.** (**A**) A model cell with a correlated Gaussian stimulus. The correlation between x_1 and x_2 is 0.80. Red circles indicate stimuli that is associated with a spike, and black circles indicate stimuli associate with silence. The arrow indicates the direction of the STA, which explained 90% bits of information. (**B**) $P(x_1)$ (black) and $P(x_1|\text{spikes})$ (red). x_1 explains 100% of the information per spike. (**C**) $P(x_2)$ (black) and $P(x_2|\text{spikes})$ (red). Because of the strong correlation with x_1, x_2 explains 60% of the information between the stimulus and the spikes.

2.1 Consistency with correlated stimulus

In order for MID to be consistent with a stimulus, the objective function (mutual information) must have a maximum when evaluated for the relevant subspace. This follows directly from the data processing inequality. Given two random variables X and Y, the mutual information is $I(X;Y)$. We then introduce another random variable Z that is a probabilistic function of only Y, which is to say that $P(Z|Y,X) = P(Z|Y)$. From Bayes' theorem, it follows that

$$\begin{aligned}
P(X|Y,Z)P(Y,Z) &= P(Z|X,Y)P(X,Y) \\
P(X|Y,Z)P(Y,Z) &= P(Z|Y)P(X|Y)P(Y) \\
P(X|Y,Z)P(Y,Z) &= P(Y,Z)P(X|Y) \\
P(X|Y,Z) &= P(X|Y).
\end{aligned} \tag{2.8}$$

We can then derive the data processing inequality:

$$\begin{array}{rclcl}
I(X;Z) & = & H(X) - H(X|Z) & \leq & H(X) - H(X|Y,Z) \\
I(X;Z) & \leq & H(X) - H(X|Y,Z) & = & H(X) - H(X|Y) \\
I(X;Z) & \leq & H(X) - H(X|Y) & = & I(X;Y) \\
I(X;Z) & \leq & I(X;Y). & &
\end{array} \tag{2.9}$$

Therefore, the dimensions that determined the response are a global maximum of the mutual information. This has the corollary that adding additional dimensions will not increase the information, which allows us to determine the number of relevant dimensions.

2.2 Convergence

Given a sufficiently long recording, MID should be able to in theory reconstruct any set of dimensions and nonlinearities. The practical question remains how the algorithm performs under experimentally realistic conditions.

To explore this question, we created sets of one-, two-, and three-dimensional model cells with different mean firing rates and therefore different number of spikes. The form of the model is described in Subsection 2.6.2 and shown in Fig. 2.14. The three-dimensional model is as shown while the one- and two-dimensional models

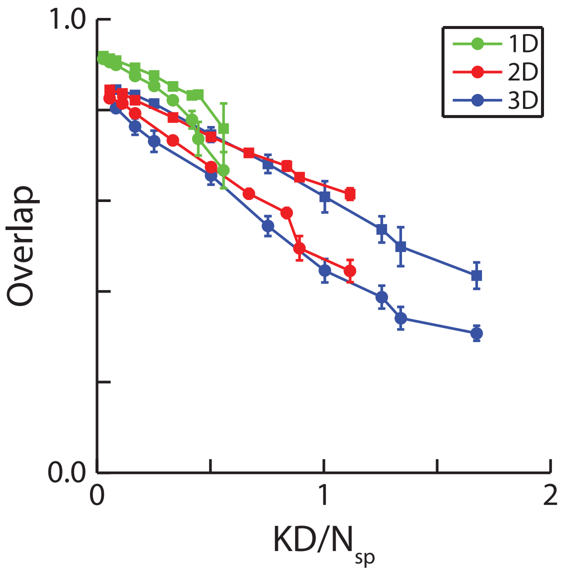

Figure 2.3: **Convergence of MID with additional spikes.** Subspace projection between the models and the reconstructions from MID (vertical axis) versus the ratio between the number of parameters defining the model dimensions to the number of spikes (horizontal axis). Small values on the horizontal axis correspond to a large number of spikes relative to the number of parameters of the model. Green, red, and blue lines correspond to one-, two-, and three-dimensional models. Averaging across jackknife estimates (squares) improved the overlap compared to the unaveraged dimensions (circles). With sufficiently large numbers of spikes, MID is able mostly recover the model dimensions (overlap > 0.8). Modest reconstructions (overlap ≈ 0.5) occur when the number of spikes is equal to the number of model parameters. Averaging across different jackknife estimates improved performance, especially in the case of limited data.

only use the first one or two dimensions respectively. We varied the parameter γ to achieved the desired number of spikes for each variation. For each model and firing rate, we generated 8 different spike trains.

Fig. 2.3 summarizes the results of the analysis. The subspace overlap of the reconstruction is approximately inversely proportional to the ratio of the stimulus dimensionality D times the number of model dimensions K to the number of spikes N_{sp}. We do see a drop between the reconstructions of the one- and two-dimensional models, but the reconstructions of the two- and three-dimensional models similar once we account for the increased number of parameters needed to describe the

Table 2.1: **Convergence of MID.** The intercept of the fit represents the predicted overlap when $KD << N_{sp}$. The slope represents the rate at which the overlap changes with changes in the ratio of stimulus dimensions per spike. For both the 1D and 2D models, the primary effect of averaging was to decrease the rate at which the reconstruction degraded with fewer spikes. For the 3D model, averaging both improved the reconstruction with many spikes and decreased the rate at which the reconstruction degraded.

Model	Overlap with infinite spikes	Slope
1D unaveraged	0.941 ± 0.011	-0.44 ± 0.04
1D averaged	0.934 ± 0.011	-0.26 ± 0.03
2D unaveraged	0.854 ± 0.008	-0.367 ± 0.013
2D averaged	0.878 ± 0.004	-0.222 ± 0.006
3D unaveraged	0.813 ± 0.013	-0.333 ± 0.013
3D averaged	0.877 ± 0.014	-0.270 ± 0.015

model dimensions.

Averaging the dimensions reconstructed from multiple jackknife estimates (App. C), improved the performance of MID. We quantified this by performing a linear regression between the overlap and the model dimensionality per spike (Table 2.1).

The extrapolated overlap does not go to 1 as KD/N_{sp} goes to 0 (N_{sp} goes to ∞). This is because there are two additional sources of error. First, the number of bins used to calculate the probability histograms $P(\mathbf{x})$ and $P(\mathbf{x}|\text{spike})$ remained constant. This puts a limit on how accurately $f(\mathbf{x})$ may be represented. Second, the number of stimuli remained fixed. This also limits the accuracy of the probability distributions $P(\mathbf{x})$ and $P(\mathbf{x}|\text{spike})$ as well as the stimulus expectations $\langle \mathbf{s}|\mathbf{x} \rangle$ and $\langle \mathbf{s}|\mathbf{x}, \text{spike} \rangle$ that are used to calculate the gradient in Eq. 2.7.

2.3 MID on cells from V1

While we have shown in Sec. 2.2 that MID is capable of reconstructing a three-dimensional model cell, but the question remains of how it performs on real neurons. We analyzed the recordings from 60 neurons from the cat primary visual cortex (App. D.1). This set included 40 simple cells and 20 complex cells. Of

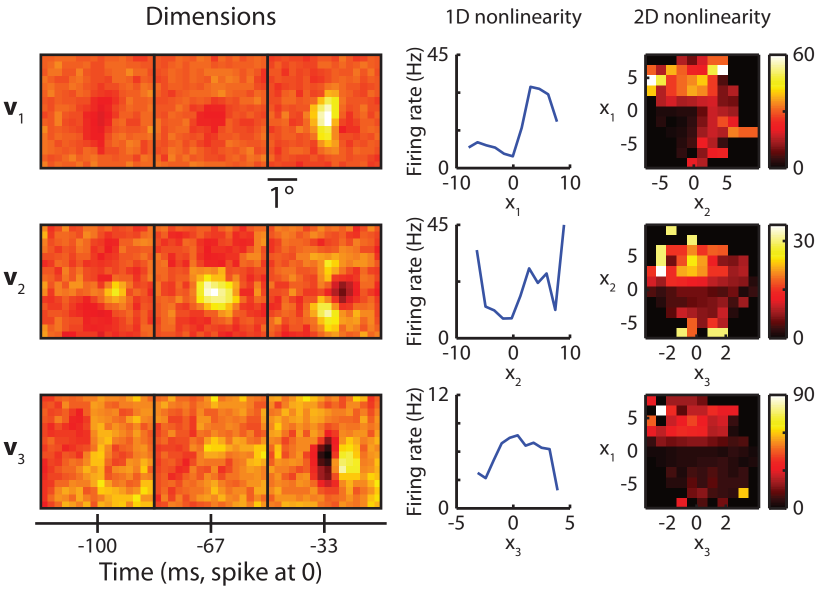

Figure 2.4: **MID on example simple V1 neuron.** This example neuron is sensitive to stimuli of a particular orientation and spatial phase. The dimensions are ordered by which contributed the most additional information. The first dimension is sensitive to bright bars of a particular orientation in the frame immediately before the spike. The second dimension is sensitive to changes in the intensity within the receptive field. The third dimension inhibits the response of the neuron to edges aligned with the center of the receptive field. Together, these dimension predict a response to gratings of only a particular spatial phase and orientation, which agrees with the classification as a simple cell based on its measured response to gratings. Cell 761–1.

the 60 sets of recordings, 47 (32 simple, 15 complex) included the responses to repeated stimuli, which allows us to estimate the information transmitted by each spike about the stimulus and therefore calculate an absolute estimate of the model performance. For this analysis, all of the stimuli were natural movies.

Fig. 2.4 shows the three-dimensional reconstruction of a simple cell, labeled 761–1. The first dimension is a vertical bar sensitive primarily to the frame immediately before the spike. The one-dimensional nonlinearity associated with this dimension resembles a threshold linear function. This combination produces the linear response to gratings associated with simple cells.

The second dimension is associated with changes in intensity in the neuron's receptive field between the first and second frame before the spike. The positive

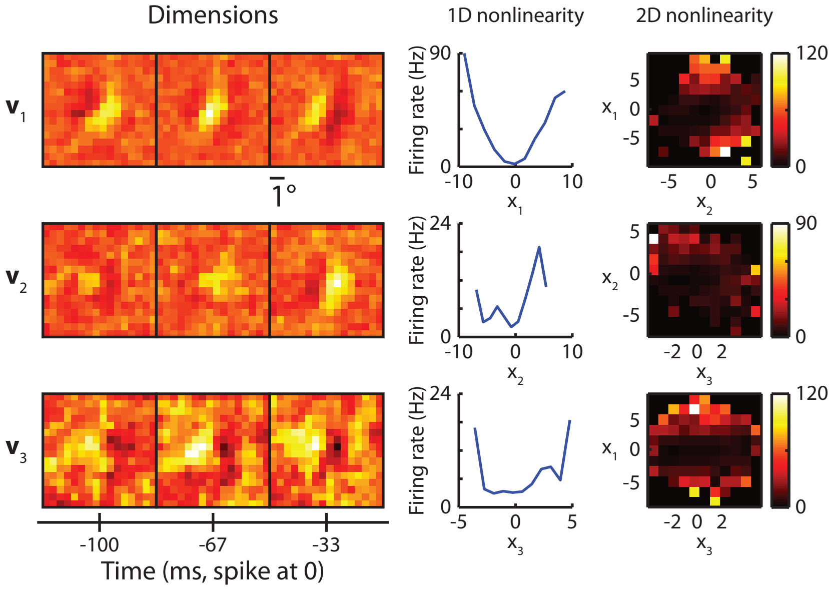

Figure 2.5: **MID on example complex V1 neuron.** This example complex cell is sensitive to motion along a particular orientation. The symmetric response to motion provided by the first dimension reproduces the characteristic constant response to moving gratings using a single spatiotemporal dimension rather than the two spatial dimensions with orthogonal phases often used to model complex cells. Cell 946–2.

blob in the middle frame and the negative blob in the frame before the spike indicate a sensitivity to decreases in the intensity, causing this neuron to also have an OFF response.

The third dimension is a Gabor filter in the frame before the spike oriented along the same direction as the first dimension. The one-dimensional nonlinearity shows that this is an inhibitory dimension. This dimension suppresses responses to stimuli out of phase with first dimension which sharpens the the selectivity of the neuron.

Fig. 2.5 shows the three-dimensional reconstruction of an example complex cell, labeled 946–2. Unlike the example simple cell in Fig. 2.4, the example complex cell is sensitive to stimuli 100 ms in the past.

The first dimension is symmetrically sensitive to motion orthogonal to its preferred orientation. This spatiotemporal dimension can replicate the constant

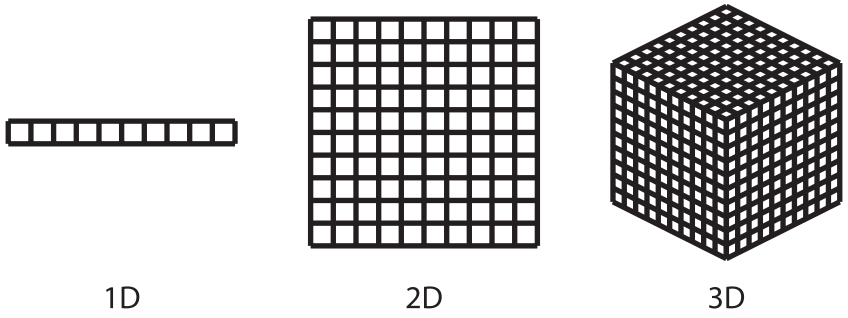

Figure 2.6: **Curse of dimensionality.** As the number of dimensions increases, the number of bins required to calculate the probability distribution increases exponentially. In this example, the number of bins increases from 10 for 1D to 100 for 2D and $1,000$ for 3D.

response to moving gratings characteristic of complex cells that is often modeled as the quadratic summation of two spatial dimensions with orthogonal spatial phases. The second and third dimensions modulate the neuron's response much more weakly and are tuned to orientations close to the first dimension's preferred orientation.

2.4 Curse of dimensionality

The primary limitation of MID is the need to calculate $P(\mathbf{x})$, $P(\mathbf{x}|\text{spike})$, $\langle \mathbf{s}|\mathbf{x}\rangle$, and $\langle \mathbf{s}|\mathbf{x}, \text{spike}\rangle$. The algorithm approximates these continuous functions using piecewise constant functions calculated using normalized histograms. Each of the K dimensions is split into B parts, so the total number of histogram bins is B^K. This exponential dependence of the size of the histograms on the number of dimensionality is the curse of dimensionality. Fig. 2.6 shows this exponential increase for 10 bins per dimension.

Even with relatively coarse binning, the number bins can quickly reach number of unique stimuli when K is only 4 or 5. Because the fractional error of the estimate of $P(\mathbf{x})$ is $1/\sqrt{n(\mathbf{x})}$, the ratio of stimuli to bins needs to be high enough that the number samples in each bin is high enough that the differences in information explained by sets of dimensions is not overwhelmed by the error due to finite sampling. This is partially mitigated by $P(\mathbf{x})$ being large near $\mu_{\mathbf{x}}$ and small for large combinations of x_i and x_j. With many samples concentrated at the center and many bins empty, the average error is less than $\sqrt{B^K/T}$. However, these empty bins do not affect the computational requirements. The size of $\langle \mathbf{s}|\mathbf{x}\rangle$ and $\langle \mathbf{s}|\mathbf{x}, \text{spike}\rangle$ scales as DB^K, which would cause problems even if we were to have infinite data.

2.5 Serial MID

The challenge of extending multiple dimensions stems from the need to calculate the probability distributions $P_V(\mathbf{x})$ and $P_V(\mathbf{x}|\text{spike})$. Using B bins per dimension, calculating the information explained by K dimensions requires B^K bins (Fig. 2.6). This is known as the curse of dimensionality (Bellman, 1961). This exponentially growing number of bins quickly makes finding additional dimensions impractical. As the data is spread across more bins, fewer data points are used to estimate the value of the probability for each bin and therefore our estimates of these values become more noisy. Furthermore, even with enough data to compensate for this, the computational requirements also grow exponentially.

One possible solution to this problem is to only calculate the information with respect to the dimension that is currently being optimized. In order to prevent the subsequent dimensions from converging to the first dimension found by the algorithm, we subtract off the component of the gradient along the elements correlated with the previously found dimensions. More precisely, the gradient is

$$\nabla_{\mathbf{v}_i,\perp} I_{\mathbf{v}_i} = \nabla_{\mathbf{v}_i} I_{\mathbf{v}_i} - \sum_{k=1}^{i-1} \nabla_{\mathbf{v}_i} I_{\mathbf{v}_i} \cdot \mathbf{v}_k. \tag{2.10}$$

We call this procedure serial MID (SMID).

2.5.1 Consistency

In order for SMID to find the relevant dimensions, the gradient must be zero for some ordering of the relevant dimensions. Otherwise, the gradient would pull the estimate out of the relevant subspace and settle on a biased estimate.

We assume a neuron that is dependent on two stimulus dimensions $\hat{e}_1$ and $\hat{e}_2$, with $\hat{e}_1$ defined as the dimension of the relevant subspace that maximizes the one-dimensional information and $\hat{e}_2$ defined as the dimension of the subspace orthogonal to the first. The projections of the stimulus on to these dimensions are defined as s_1 and s_2, respectively.

We begin with the gradient of the information:

$$\nabla_{\mathbf{v}_i} I_V = \int d\mathbf{x} P_V(\mathbf{x}|\text{spike})(\langle \mathbf{s}|\mathbf{x}\rangle_V - \langle \mathbf{s}|\mathbf{x},\text{spike}\rangle_V)\frac{d}{dx_i}\left(\frac{P_V(\mathbf{x}|\text{spike})}{P_V(\mathbf{x})}\right). \tag{2.7}$$

First, we note that

$$\langle \mathbf{s}|s_1\rangle P(s_1) = \int ds_2 P(s_1,s_2)\langle \mathbf{s}|s_1,s_2\rangle \tag{2.11}$$

and that

$$\langle \mathbf{s}|s_1,\text{spike}\rangle P(s_1|\text{spike}) = \int ds_2 P(s_1,s_2|\text{spike})\langle \mathbf{s}|s_1,s_2,\text{spike}\rangle . \tag{2.12}$$

We will also need to use

$$\langle \mathbf{s}|s_1,s_2,\text{spike}\rangle = \langle \mathbf{s}|s_1,s_2\rangle , \tag{2.13}$$

which follows from the dependence of the response to the stimulus coming solely through the intermediary stimulus dimensions s_1 and s_2. We can then rewrite Eq. 2.7 as

$$\begin{aligned}\nabla_{\hat{e}_1} I_{\hat{e}_1} &= \int ds_1 ds_2 P(s_1|\text{spike})\langle \mathbf{s}|s_1,s_2\rangle \\ &\qquad (P(s_1,s_2|\text{spike}) - P(s_1,s_2))\frac{d}{ds_1}\log_2\left(\frac{P(s_1|\text{spike})}{P(s_1)}\right) \\ &= \int ds_1 ds_2 P(s_1,s_2)\langle \mathbf{s}|s_1,s_2\rangle \\ &\qquad \left(\frac{P(\text{spike}|s_1,s_2) - P(\text{spike}|s_1)}{P(\text{spike})}\right)\frac{d}{ds_1}\log_2\left(\frac{P(s_1|\text{spike})}{P(s_1)}\right).\end{aligned} \tag{2.14}$$

The behavior of the gradient is determined by the term $\langle \mathbf{s}|s_1,s_2\rangle$.

In the case of an uncorrelated stimulus,

$$\langle \mathbf{s}|s_1,s_2\rangle = s_1\hat{e}_1 + s_2\hat{e}_2 + \mathbf{c}_\perp \tag{2.15}$$

where $\mathbf{c}_\perp$ is a constant vector that is orthogonal to $\hat{e}_1$ and $\hat{e}_2$. In this case, the gradient perpendicular to $\hat{e}_1$ and $\hat{e}_2$ is

$$\begin{aligned}\nabla_{\hat{e}_1\perp} I_{\hat{e}_1} = \frac{\mathbf{c}_\perp}{P(\text{spike})} \int ds_1 \frac{d}{ds_1} \log_2 \left(\frac{P(s_1|\text{spike})}{P(s_1)} \right) \\ \int ds_2 P(s_1, s_2)(P(\text{spike}|s_1, s_2) - P(\text{spike}|s_s)).\end{aligned} \tag{2.16}$$

The integral with respect to s_2 is 0, and therefore, $\nabla_{\hat{e}_1} I_{\hat{e}_1}$ does not have any components outside of the relevant subspace. Furthermore, the gradient will have no component along $\hat{e}_1$ because information is scale-invariant, and it will have no component along $\hat{e}_2$ because we defined $\hat{e}_1$ as the dimension that maximizes the one-dimensional information. $\nabla_{\hat{e}_2} I_{\hat{e}_2}$ may have a component along $\hat{e}_1$, but as we found $\hat{e}_1$ first, we can subtract off this component. In this case, $\mathbf{c}_1 = \hat{e}_1$ because the covariance matrix is the identity matrix. Therefore, SMID is compatible with uncorrelated stimuli. It is also compatible with correlated Gaussian stimuli if the stimuli are decorrelated during preprocessing.

For a stimulus with arbitrary correlations, such as natural stimuli, the correlations cannot be simplified any further. If the correlations are close to linear, the resulting bias may be small, but in the general case, SMID is not compatible with stimuli with higher order correlations.

2.5.2 SMID on model cells

In the last section, we predicted that SMID would be able to find multiple dimensions if the stimulus is an uncorrelated Gaussian stimulus. For natural stimuli, SMID will provide biased estimates. We now test these predictions with a model cell.

Model cell

The model cell consists of two spatiotemporal filters that respond to the onset of a spatial frequency at a particular orientation and location. The filters are identical except that they have orthogonal spatial phase. The dimensions of the filters are 16 × 16 pixels with a duration of 3 frames, so each filter has a size of 768 values. The filters are shown in Fig. 2.7A.

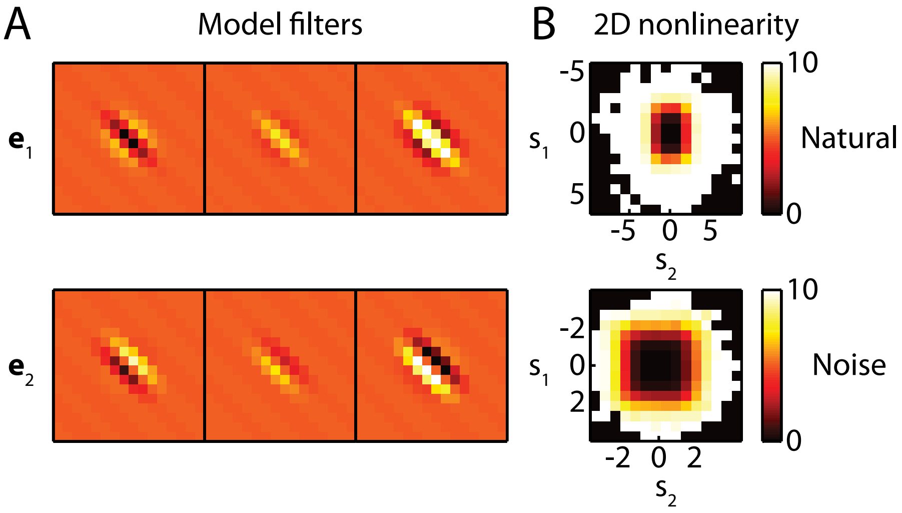

Figure 2.7: **Two-dimensional model cell.** (**A**) Two spatiotemporal filters responding to the onset or offset of an oriented grating. (**B**) The nonlinearities for the natural and noise stimuli calculated using the mode spike trains. The cell spiked if the projection plus random noise exceeded the threshold in either the positive or negative direction for either dimension.

If the magnitude of the projection along either filter plus some noise exceeded a threshold, the model neuron spiked. We repeated the stimulus ten times and summed the response. The empirical nonlinearities using the model dimensions and the generated spikes are shown in Fig. 2.7B. We used both the movie stimulus and the noise stimulus described in Section D.1.

SMID and MID on model cell

We ran SMID on the responses of the model cell to both noise and natural stimuli. The natural stimuli is described in Section D.1. The noise stimulus was a 64 × 64 pixel image with values drawn uniformly from the integers between 0 and 255, inclusive. The stimuli were downsampled to the 16 × 16 pixel resolution of the model. At this level of downsampling, the distribution of values becomes approximately normal.

As predicted, SMID was able accurately recover the model dimensions in the case of noise stimulus (Fig. 2.8). The overlap between the model and reconstruction

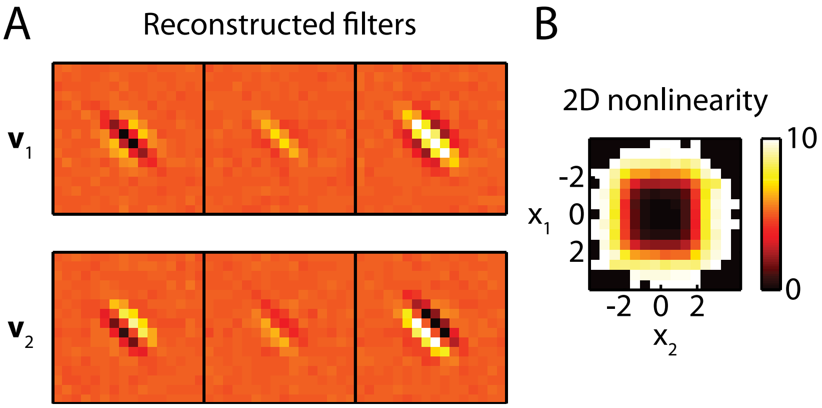

Figure 2.8: **SMID on noise stimulus.** (**A**) Reconstructed filters. SMID is able to recover the model dimensions with an overlap of 0.83±0.15 across jackknifes that increases to 0.99 when the filters are averaged. (**B**) Two-dimensional nonlinearity.

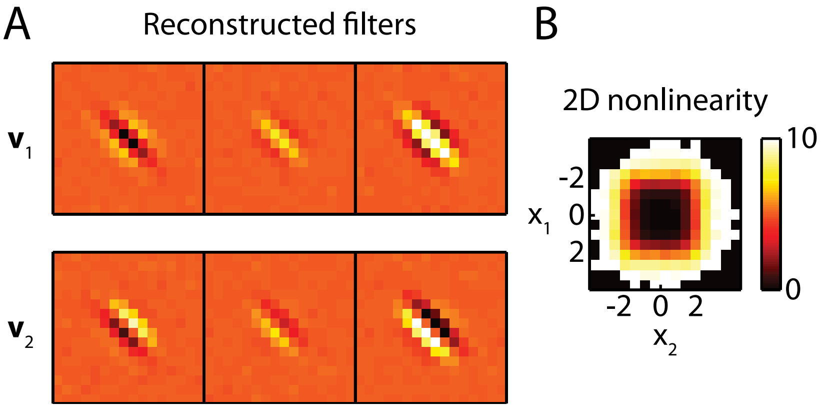

Figure 2.9: **MID on noise stimulus.** (**A**) Reconstructed filters. The overlap is 0.8 ± 0.2 across jackknifes and 0.99 for the averaged filter. (**B**) Two-dimensional nonlinearity.

was 0.83 ± 0.15 across jackknifes and 0.99 for the averaged dimensions from PCA (see Appendix C).

We also analyzed the model cell using MID. In the case of the noise stimulus (Fig. 2.9), MID performed similarly to SMID with an overlap of 0.8 ± 0.2 across jackknifes and 0.99 for the averaged filter. SMID was able to perform as well as MID because the magnitude of the correlation within the planes defined by the model or the reconstructed filters did not exceed 0.01.

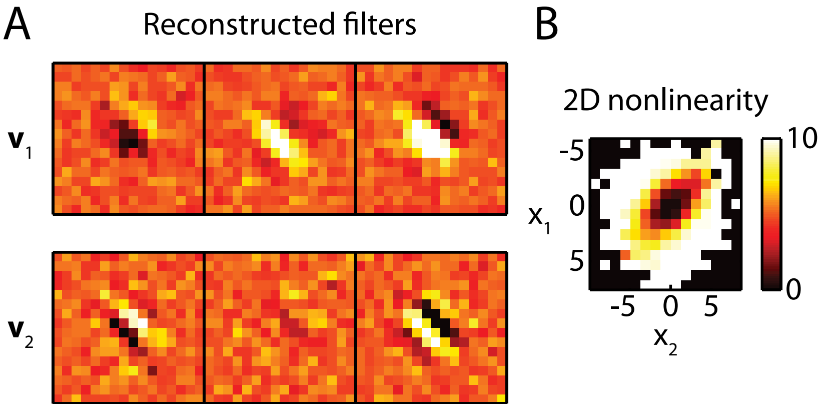

Figure 2.10: **SMID on natural stimulus.** (**A**) Reconstructed filters. The overlap is 0.60 ± 0.04 across jackknifes and 0.65 when they are averaged. This is much poorer than the performance on the noise stimulus. (**B**) Two-dimensional nonlinearity. The alignment of the non-spiking region in the center along the diagonal is a result of the correlation between x_1 and x_2.

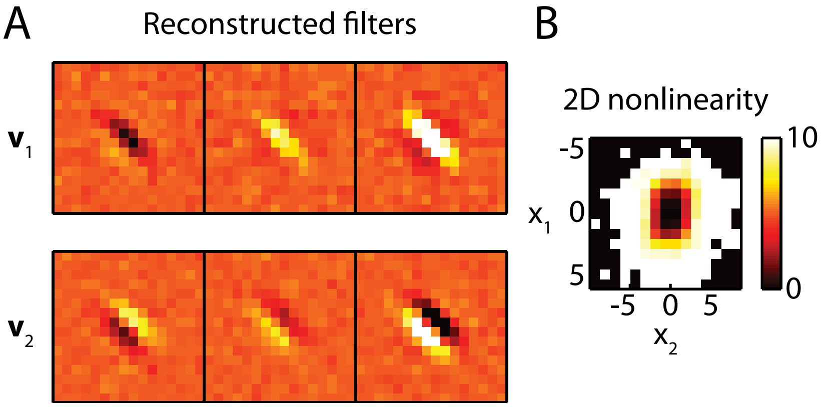

Figure 2.11: **MID on natural stimulus.** (**A**) Reconstructed filters. The overlap is 0.875 ± 0.008 across jackknifes and 0.89 for the averaged filter. (**B**) Two-dimensional nonlinearity.

In the case of natural stimuli, the reconstruction was less successful. The recovered dimensions (Fig. 2.8A) had only an overlap of 0.60±0.04 across jackknifes and 0.65 for the average filters. The difference between the reconstruction and the model is visually apparent, and the correlations between the reconstructed subspace cause the nonlinearity's non-spiking central region to have a more curved shape compared to model in Fig. 2.7B or the MID reconstruction in Fig. 2.11B.

With the natural stimulus (Fig. 2.11), MID performed much better. The overlap between the model dimensions and the dimensions recovered by MID was 0.875 ± 0.008 across jackknifes and 0.89 for the averaged filters.

Our theoretical analysis predicts that SMID will be biased because the second dimension will recover dimensions that are orthogonal but correlated with the first dimension, so we expect that the subspace of the SMID reconstruction will have stronger correlations than the subspaces of MID or the model.

We checked this prediction by rotating an orthogonal basis for each subspace and calculating the correlation of the stimulus along those dimensions. The correlation of the subspace was the maximum magnitude of the correlation along the rotated dimensions. Checking the correlation along rotated bases is essential because the correlation will be zero if the dimensions happen to be the principle axes of the subspace.

The model subspace itself had a moderate correlation of 0.39. The subspace reconstructed by SMID had a much higher correlation of 0.77. MID is able to remove some but not all of the superfluous correlations. Its reconstructed subspace had a correlation of 0.52. MID performed better than SMID because it was better able to ignore some of the uninformative correlations with previously found dimensions.

2.5.3 Convergence of SMID

In this section, we analyze the convergence of SMID with an increasing number of spikes. We followed the same procedure used in Section 2.2. Fig. 2.5.3 shows the results along with the results from MID for comparison.

As expected, SMID performed comparably to MID with the one-dimensional model cell where the algorithms are equivalent. For the two-dimensional model cell (red), the reconstructed dimensions from SMID had a much lower overlap with the model dimensions. Averaging four jackknife estimate from SMID performed only as well as a single jackknife estimate from MID.

For the three-dimensional model cell, SMID performed poorly. It only achieved an overlap of ~ 0.2 even with large numbers of spikes. This drop in

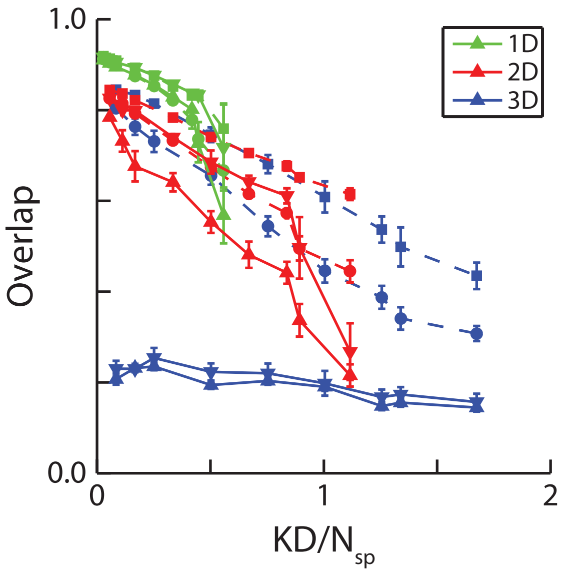

Figure 2.12: **Convergence of SMID with additional spikes.** Subspace projection between the models and reconstructions from SMID (solid line). Both values for dimensions averaged across jackknifes (▽) and unaveraged reconstructions (△) are shown. Values for MID (dashed line) are included for comparison. As SMID is equivalent to MID for one dimension, it does comparably for the one-dimensional model (green). For the two-dimensional model, SMID did worse than MID with a single jackknife estimate from MID performing as well as four SMID jackknife estimates averaged together. For the three-dimensional model, SMID performed poorly and did not improve with increasing data.

performance is likely due to the relatively weak modulation of the inhibitory third dimension compared to the first two excitatory dimensions. A dimension outside the relevant subspace could carry more information about the model's spiking via correlations with the first two dimensions than is explained by the third dimension.

2.5.4 SMID on cells from V1

In order to test whether the results with respect to model cells held up under realistic experimental conditions, we also tested SMID on cells from V1. We used recordings from the cat visual cortex described in Appendix D.1. We used SMID and MID to create three-dimensional reconstructions of these cells.

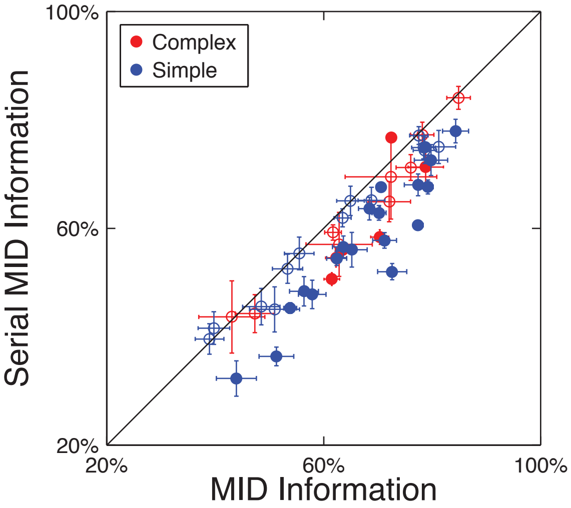

Figure 2.13: **Information in V1 responses explained by SMID and MID.** Percent information explained three spatiotemporal dimensions found by SMID and MID about novel responses of V1 neurons (n = 47) to repeated stimulus. Percentage is out of the total estimated information per spike. Red circles indicate the cell was classified as complex while blue circles indicate the cell was classified as simple. Filled circles indicate there was a significant (p < 0.05) difference between SMID and MID. MID performed significantly better than SMID (p $< 10^{-4}$, paired t-test).

We had 47 neurons with responses to repeated stimuli, which are necessary to estimate the information per spike. Of these neurons, 32 were classified as simple and 15 were classified as complex according to their responses to moving gratings (Skottun et al., 1991).

For each neuron, we calculated the information about the spiking explained by the three-dimensional model and extrapolated to infinite data (Strong et al., 1998). Using the responses to repeated stimuli, we were also able to estimate the information transmitted by each spike based on the variation of the response across stimuli and the consistency of the response to the same stimulus. This allows us to describe the information explained as a fraction of the total information carried by the neuron's response. Fig. 2.13 compares the information explained by SMID compared to that explained by MID. Across the population, MID explained more information than serial MID ($p = 3 \times 10^{-10}$, paired t-test).

2.6 Comparison between MID and extended projection pursuit regression

Rapela et al. (2010) asserted that ePPR performed better on natural stimuli. We sought to investigate their claims.

2.6.1 Projection pursuit regression

Projection pursuit regression (PPR) models the response as the sum of a series linear kernels passed through a nonlinear function:

$$\hat{y} = \bar{y} + \sum_{k=1}^{K} \beta_k f_k(\mathbf{v}_k \cdot \mathbf{s}) + \epsilon. \tag{2.17}$$

The model is subject to the conditions that the kernels are normalized,

$$||\mathbf{v}_k||_2 = 1; \tag{2.18}$$

the nonlinear functions have zero mean,

$$\frac{1}{N}\sum_{t=1}^{N} f_k(\mathbf{v}_k \cdot \mathbf{s}_t) = 0; \tag{2.19}$$

and unit variance,

$$\frac{1}{N}\sum_{t=1}^{N} f_k^2(\mathbf{v}_k \cdot \mathbf{s}_t) = 1. \tag{2.20}$$

The parameters of the model are found by minimizing the mean squared error (MSE) between the model prediction and the observations

$$MSE = \frac{1}{N}\sum_{t=1}^{N}(y_t - \hat{y}_t)^2. \tag{2.21}$$

If we define the residual as

$$r_{k,t} = y_t - \sum_{k'=1}^{k-1} \beta_{k'} f_{k'}(\mathbf{v}_{k'} \cdot \mathbf{s}_t), \tag{2.22}$$

the MSE for the k-dimensional model is

$$MSE_k = \frac{1}{N}\sum_{t=1}^{N}(r_{k,t} - \beta_k f_k(\mathbf{v}_k \cdot \mathbf{s}_t))^2. \tag{2.23}$$

Extended PPR

Rapela et al. (2010) suggested an extension of PPR to include spatiotemporal models. These models add a memory of D_t past stimulus frames. This extended PPR (ePPR) had two variants. The first incorporated time by stacking the stimulus from multiple frames so that the dimensionality of the model vectors $\mathbf{v}_k$ goes from D to $D \times D_t$. The second variant retains a vector dimensionality of D but adds additional models that respond to a delayed stimulus. The first model, called ePPR with time interactions, has the advantage of being able to capture interactions between different points in time, but the higher dimensionality makes fitting the model more difficult. Furthermore, the size of D_t has to be determined in advance. The second variant, called ePPR without time interactions, has the advantage of being easier to fit and can determine the size of D_t during optimization. The disadvantage is that it treats each time independently.

The form of the nonlinearity can approximate any arbitrary polynomial function which in turn can approximate any arbitrary function (Rapela et al., 2010). However, this statement does not put any constraints on the number of ePPR dimensions required to approximate the nonlinearity.

Systemic bias in ePPR

In Rowekamp and Sharpee (2011), we demonstrated that mismatch between the estimated nonlinearity $\hat{y}(\mathbf{s})$ and the actual nonlinearity $y(\mathbf{s})$ will cause biased estimations for PPR when the stimulus has non-Gaussian equations.

We begin with an analysis of a one-dimensional case which we can later generalize to multidimensional case. We start with Eq. 2.21

$$MSE = \frac{1}{N}\sum_{t=1}^{N}(y_t - \hat{y}_t)^2. \tag{2.21}$$

We can rewrite this sum as an integral on the probability distribution

$$MSE = \int d\mathbf{s}P(\mathbf{s})(y(\mathbf{s}) - \hat{y}(\mathbf{s}))^2. \tag{2.24}$$

Taking the gradient with respect to relevant dimension $hate_1$ gives us

$$\nabla_{\hat{e}_1} MSE = -2\int d\mathbf{s}P(\mathbf{s})(y(\mathbf{s}) - \bar{y} - \beta_1 f_1(\hat{e}_1 \cdot \mathbf{s}))\nabla_{\hat{e}_1} f(\hat{e}_1 \cdot \mathbf{s}). \tag{2.25}$$

We can rewrite this as an integral with respect to s_1, the projection of the stimulus along $\hat{e}_1$ by inserting Eq. 2.17 and integrating across all other dimensions:

$$\nabla_{\hat{e}_1} MSE = -2 \int ds_1 P(s_1) \langle \mathbf{s}|s_1 \rangle \left(\Delta y(s_1) - \beta_1 f_1(s_1)\right) \frac{d}{ds_1}(\beta_1 f_1(s_1)). \tag{2.26}$$

Δy is the deviation of the observed firing rate y from its mean $\bar{y}$, which is the first term in Eq. 2.17. From this equation, we can note that the gradient will be zero whenever $\hat{y}(\mathbf{s})$ is equal to $\Delta y(\mathbf{s})$.

If there is a mismatch between $\hat{y}(\mathbf{s})$ and $\Delta y(\mathbf{s})$, the bias depends on the properties of $\langle \mathbf{s}|s_1 \rangle$. In the case of a correlated Gaussian stimuli, which has linear correlations, this becomes

$$\langle \mathbf{s}|s_1 \rangle = \mathbf{c}_1 s_1 + \mathbf{c}_\perp. \tag{2.27}$$

2.6.2 ePPR and MID on model cell

We know that given sufficient number of dimensions and data, ePPR can approximate a neuron's nonlinear response, but this is also true of MID. Two questions remain: first, how well can the algorithms perform under realistic experimental conditions, and second, can we interpret the reconstructed model parameters to understand and characterize the neuron's computation.

We first tested ePPR and MID on a three-dimensional model cell in order to evaluate the algorithms' performance when the underlying model is known.

Description of three-dimensional model cell

We attempted to replicate the behavior of the model cell from Rapela et al. (2010) as closely as possible in order to perform the best comparison between our analysis and theirs.

The model consists of three filters of size 16 × 16 pixels in space and 3 frames in time, shown in Fig 2.14A. The first two are identical except with orthogonal spatial phases. Each contains a Gabor function in the first and second frame before the spike. The gratings are aligned along the diagonal, and they are shifted in space along this axis. The third filter contains a Gabor function in the third frame before the spike. It is oriented orthogonally to the first two filters.

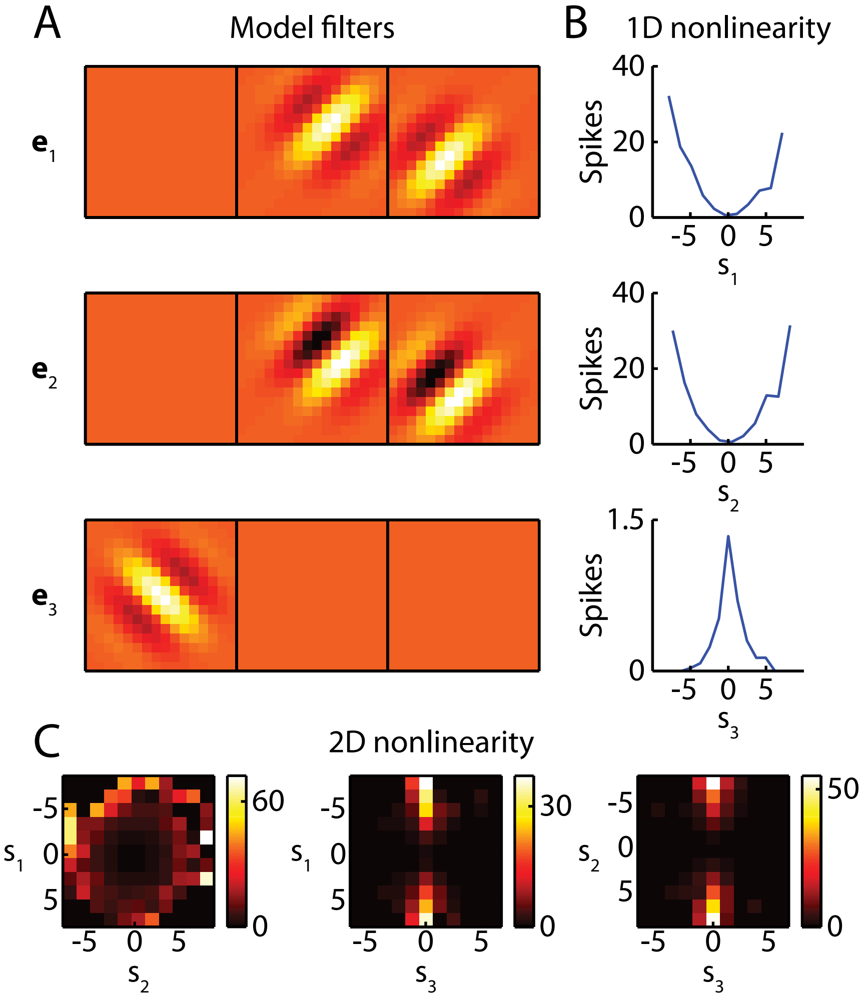

Figure 2.14: **Three-dimensional model cell.** (**A**) The three spatiotemporal filters of the model cell ($\hat{e}_1, \hat{e}_2, \hat{e}_3$). The frames are arranged with the part of the filter corresponding with the earliest frame on the left and the part of the filter corresponding with the frame closest to the spike on the right. (**B**) The one-dimensional nonlinearities for each dimension calculated using the simulated spike train for the long-stimulus condition. The response is in units of spikes per bin. (**C**) The two-dimensional nonlinearities.

The expected firing rated is determined using the equation

$$f\left(\mathbf{s}\right)=\gamma\frac{\left(\mathbf{s}\cdot\hat{e}_1\right)^2+\left(\mathbf{s}\cdot\hat{e}_2\right)^2}{1+\omega\left(\mathbf{s}\cdot\hat{e}_3\right)^2}. \tag{2.28}$$

The parameters γ and ω control the overall firing rate and the level of inhibition. The firing rate is converted into spikes using a Poisson distribution. To remain consistent with Rapela et al. (2010), we chose γ such that $\left\langle f\left(\mathbf{s}\right)\right\rangle$ was equal to 0.56 and ω such that $\left\langle 1+\omega\left(\mathbf{s}\cdot\hat{e}_3\right)^2\right\rangle$ was equal to 4.26.

This model cell captures some of the characteristics observed in complex cells in V1: orientation selectivity, phase-invariance, and cross-orientation inhibition.

The stimulus used to generate the spikes is the natural movie described in Section D.1. We analyzed two stimulus durations: a shorter 20000 frame stimulus consistent with the size of the dataset used by Rapela et al. (2010) and a longer 49152 frame stimulus that utilized the full natural movie stimulus. Fig. 2.14B and C show the nonlinearities calculated with the model filters and the simulated spike train.

Analysis of model cell

We ran both ePPR and MID on both the short- and the long-stimulus model cells. Filters were averaged using PCA, as described in Appendix C. For ease of comparison, we rotated the resulting filters so that they were aligned as much as possible with the corresponding model filters. We did this by defining the rotated reconstruction $V' = VR$, where V are the reconstructed filters and R is a rotation matrix. We then chose R as the rotation that maximized $\sum\limits_{i=j} A_{i,j}^2 - \sum\limits_{i\neq j} A_{i,j}^2$, where $A = E^T VR$. This maximizes the diagonal elements while minimizing the off-diagonal elements.

On the 20000-frame short stimulus condition, ePPR was able to partially reconstruct the model filters. The reconstructed filters are shown in Fig. 2.15A with the corresponding one- and two-dimensional nonlinearities in Fig. 2.15B and C. The overlap across jackknifes was 0.55 ± 0.04. Averaging the estimates together increased this to 0.62. The model does much better at recovering the inhibitory

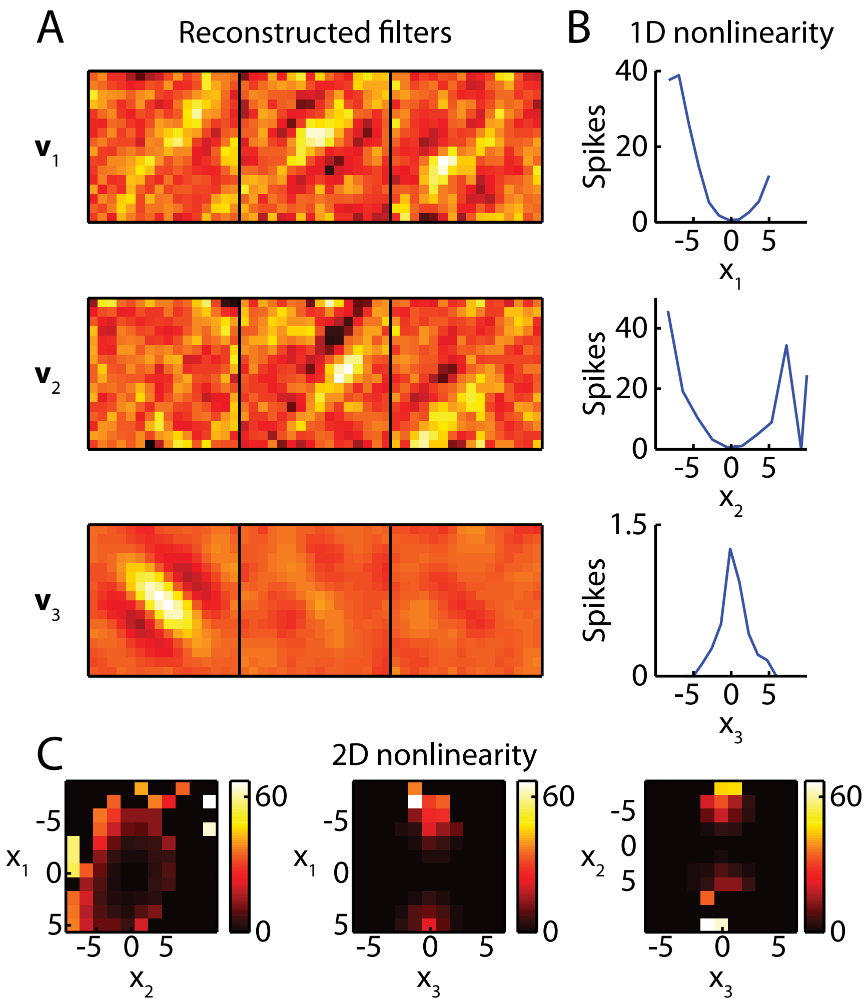

Figure 2.15: **ePPR reconstruction of short-stimulus model.** (**A**) Reconstructed filters. The overlap was 0.55 ± 0.04 for the individual jackknifes and increased to 0.62 when the jackknifes were averaged together. Filters were rotated to align with model dimensions for ease of comparison. (**B**) The one-dimensional nonlinearities calculated from reconstructed dimensions and spikes. (**C**) The two-dimensional nonlinearities.

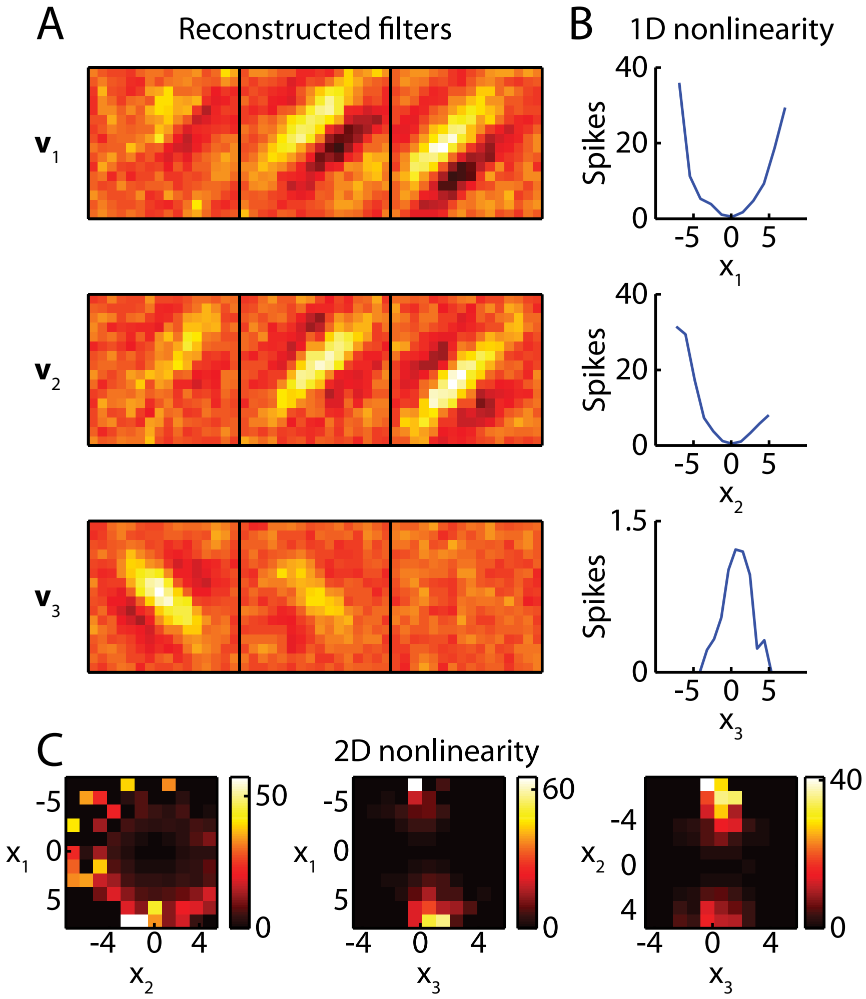

Figure 2.16: **MID reconstruction of short-stimulus model.** (**A**) Reconstructed filters. The overlap was 0.65 ± 0.08 for the individual jackknifes and increased to 0.81 when averaged. This is better than ePPR on either the short or the long stimulus. (**B**) One-dimensional nonlinearities. (**C**) Two-dimensional nonlinearities.

filter than the phase invariant excitatory filters. The overlap with the first two model dimensions was 0.44 ± 0.10 across jackknifes and 0.51 for the averaged filters. For the third model filter, the overlap was 0.89 ± 0.02 across jackknifes and 0.92 for the averaged filter.

MID did better at reconstructing the model dimensions for the short-stimulus condition. The reconstructed filters and nonlinearities are shown in Fig. 2.16. The

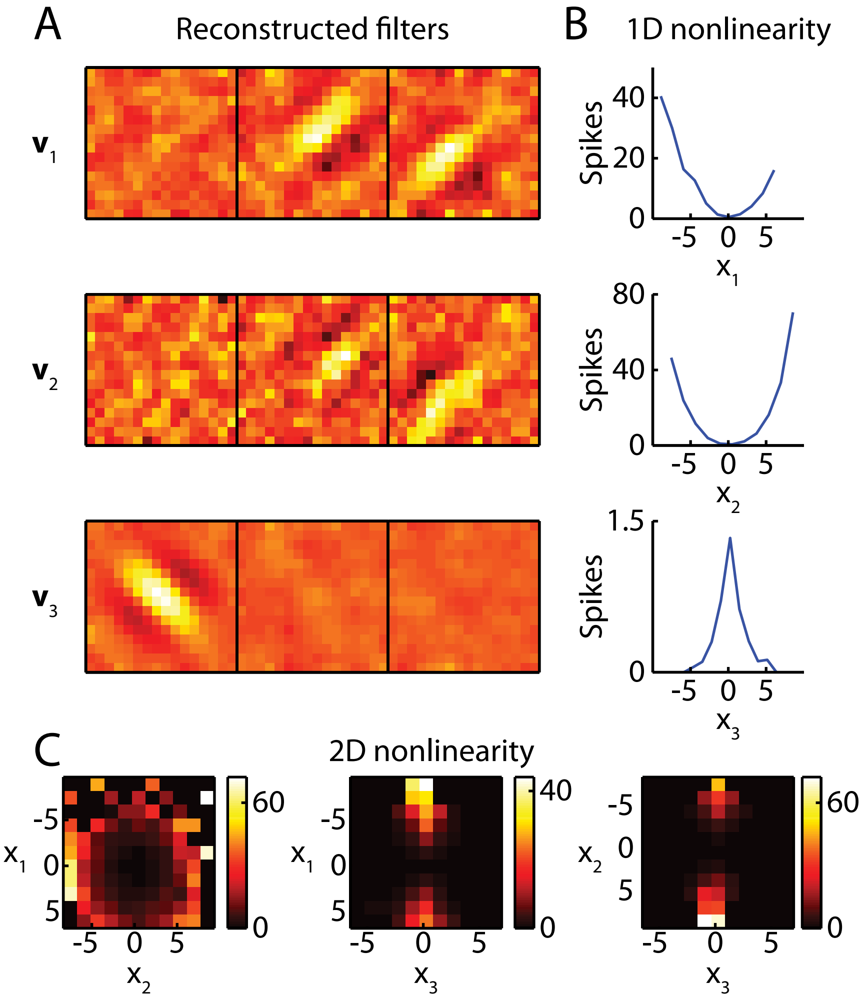

Figure 2.17: **ePPR reconstruction of long-stimulus model.** (**A**) Reconstructed filters. The overlap was 0.59 ± 0.11 across jackknifes and 0.77 for the averaged filters. This is better than ePPR on the short stimulus but below the values for MID on either the short or long stimulus. (**B**) One-dimensional nonlinearities. (**C**) Two-dimensional nonlinearities.

overlap was 0.65 ± 0.08 for the individual jackknifes and 0.81 for the averaged filters. MID did better at recovering the excitatory filters (overlap 0.83 ± 0.04 and 0.83) than ePPR but did worse on the inhibitory filter (overlap 0.49 ± 0.13 and 0.77).

Increasing the amount of data to 49152 frames for the long-stimulus condition improved the performance of both algorithms. Shown in Fig. 2.17, ePPR

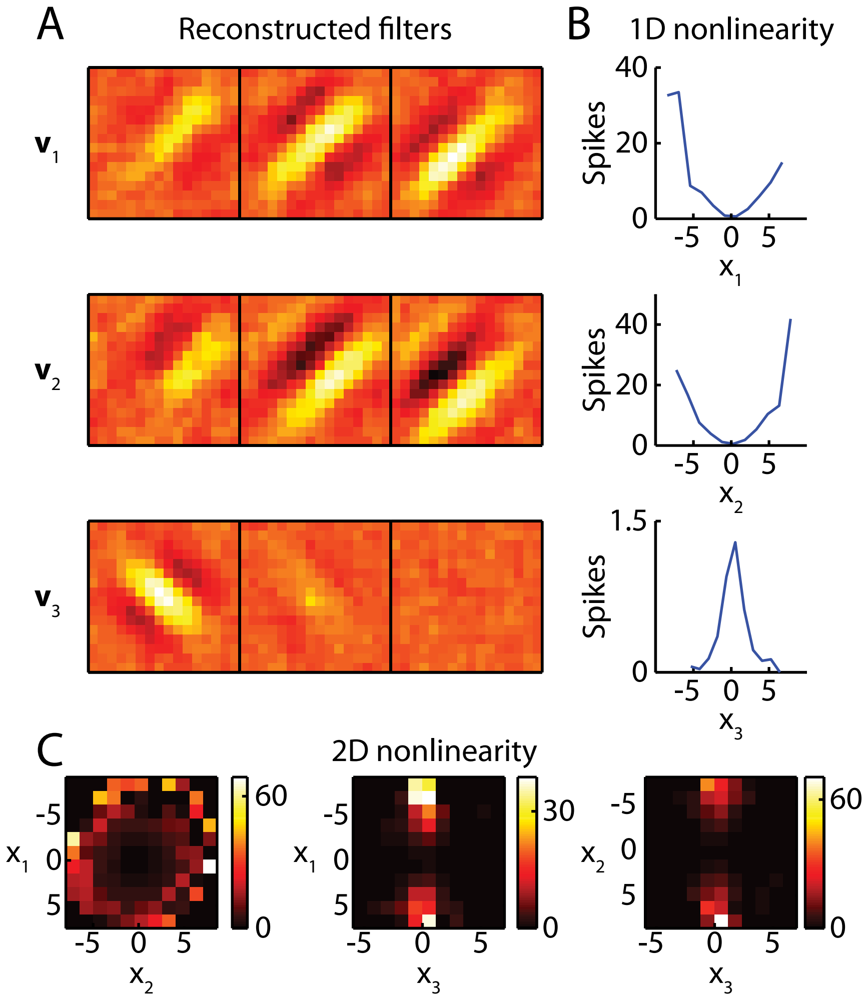

Figure 2.18: **MID reconstruction of long-stimulus model.** (**A**) Reconstructed filters. The overlap was 0.829 ± 0.005 across jackknifes and 0.86 for the averaged filters. This is the best reconstruction of the three-dimensional model. (**B**) One-dimensional nonlinearities. (**C**) Two-dimensional nonlinearities.

improved to have an overlap of 0.59 ± 0.11 across jackknifes and 0.77 for the averaged filters. While better than ePPR on the short stimulus, these values are below those found for MID on the short stimulus.

MID on the long-stimulus condition, shown in Fig. 2.18, did the best of all. The overlap increased to 0.829 ± 0.005 across individual jackknifes and 0.86 for the averaged filters.

Analysis of ePPR and MID on a model cell with natural movie stimulus demonstrated that MID was better able to reconstruct the dimensions used by the model, which agrees with our theoretical predictions. This increased performance persisted even when MID had only 40% as much data to analyze.

2.6.3 ePPR and MID on V1 neurons

While it is promising that the analysis of simulated neurons supports our theoretical predictions, the key test is whether it produces better reconstructions of real neurons under experimental conditions. To test ePPR and MID, we again analyzed the recordings of the responses of neurons in V1 to natural movie stimuli. These experiments are described in Appendix D.1.

We ran ePPR and MID on 47 neurons for which we had responses to repeated stimuli. These responses are necessary to estimate the total information in the neuronal response and how much of that information is captured by the reconstructed model. Of the 47 neurons, 32 were simple cells and 15 were complex cells as defined by their responses to moving gratings (see Appendix D.1 for details).

To compare the quality of the reconstructions given by the two algorithms, we calculated fraction of the information between a novel repeated stimulus and the corresponding neuronal responses. The repetition allowed us to estimate the mean information transmitted by a spike based on the variability of the response (see Appendix E). Across the population, the MID dimensions explained significantly more information than the ePPR dimensions (p $< 10^{-4}$, paired t-test). The distribution of performances are shown in Fig. 2.19.

2.7 Discussion

Finding dimensions of the stimulus that have the most mutual information with the corresponding responses of the cell creates a maximum likelihood LN model for the dataset. As a maximum likelihood estimator, it produces an estimate with the minimum variance for unbiased estimators with the same form. MID is vulnerable to the curse of dimensionality as the number of dimensions increases. Increasing the number of dimensions causes the observations to be spread across an

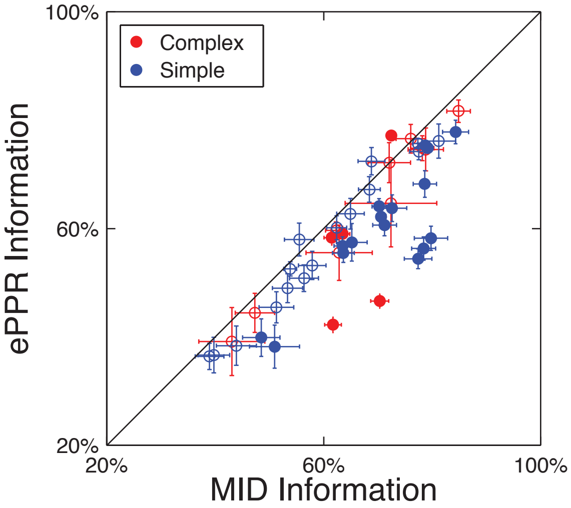

Figure 2.19: **Information in V1 responses explained by ePPR and MID.** Percent of the information between a repeated stimulus and recorded responses explained by the dimensions reconstructed using ePPR and MID. The MID dimensions where significantly better at capturing the information than the ePPR dimensions ($p < 10^{-4}$, paired t-test). Red circles indicate complex cells while blue circles indicate simple. Open circles indicate that the difference between ePPR and MID was not significant ($p > 0.05$) while filled circles indicate that the difference was significant ($p < 0.05$).

increasing number of bins, which increases the uncertainty in our estimates of the corresponding probabilities. We found that this is not a problem for up to three dimensions for model cells with datasets whose sizes were comparable to the sizes of datasets in our recordings from V1. For our model cells, increasing the size of the dataset required to obtain comparable performance appears to increase linear with the number of dimensions rather than exponentially as could be expected based on the exponentially increasing number of bins. This may be due to the non-uniform distribution of stimuli where stimuli are concentrated around the mean and observations of extreme values along more than one dimension are rare, which leaves many bins empty. Furthermore, the calculation of the gradient is weighted by the distribution of observations which discounts observations from the poorly sampled regions of the stimulus.

The sequential optimization of the one-dimensional information allows SMID to overcome the curse of dimensionality. Adding additional dimensions only requires additional searches with the same complexity as the search for the initial dimension. The trade-off for this simplicity is a loss of the ability to ignore stimulus correlations. We have shown both analytically and through the use of model cells that correlations in the stimulus lead to biased estimates of the relevant subspace. Furthermore, analysis of the responses of neurons from V1 revealed that MID was better able to model the responses of neurons than SMID under experimental conditions.

Given SMID's trade-off between being able to find many dimensions in exchange for the ability to accommodate stimulus correlations, it does not provide a viable alternative to the existing methods of STC and MID. When the stimulus is Gaussian, STC is capable of finding multiple dimensions in a computationally more efficient manner than SMID. When the stimulus has higher-order correlations, MID is capable of finding an unbiased estimate of the relevant subspace of the stimulus.

Contrary to (Rapela et al., 2010), we found that the MID performed better than ePPR. Part of this discrepancy may be due to the differences in analysis. (Rapela et al., 2010) used principal angles, which privileges recovering the original coordinate system even if the computation contains symmetries that make it invariant to certain transformations, such as rotation of the first two dimensions of their model cell. Instead, we used the subspace overlap (App. B) which compares subspaces independent of the choice of coordinate systems. To compare the performance on the recordings from V1, we used information explained by the reconstructed dimensions rather than the correlation between predicted and observed response because ePPR has a parameterized nonlinearity and MID has an arbitrary nonlinearity. To make predictions for MID, (Rapela et al., 2010) fit a multidimensional polynomial model. We made this choice because we are primarily concerned about the feature selectivity of the neurons.

While both ePPR and MID are able in theory to model an arbitrary nonlinearity, they both require an infinite number of dimensions to be assured to achieve this. The question remains which algorithm is more effective under experimental conditions where there is a finite amount of data with which to train the models.

We found that MID was superior to ePPR when applied to both simulated cells and recordings from V1. MID's superior performance may be the result of its ability to directly measure the effect of correlations between stimulus dimensions via the multidimensional information which eliminates the need for additional features to compensate for biases resulting from a constrained lower dimensional model.

Ch. 2 draws upon the work published in Rowekamp and Sharpee (2011). The dissertation author was the primary author of this paper.

Chapter 3

Invariant MID

An alternative way to extend MID to higher-dimensional computations is to take advantage of underlying symmetries to simplify high-dimensional system into one with a more manageable number of dimensions. One promising area where this assumption may prove fruitful is with translation-invariant cells. These neuron do not alter their response to a stimulus when the stimulus is shifted in space. This behavior is found is higher-order visual areas, and it is believed to aid in object recognition.

Neurons may not have unlimited spatial invariance. Invariance results from pooling of the responses of earlier neurons, so invariance is limited both by the area covered by the lower-level neurons and by the density of these neurons. A neuron cannot be invariant to a stimulus that is outside of the receptive fields of its input neurons, and it may not be able to interpolate as the stimulus moves between the locations of the input neuron receptive fields.

Our model of translationally invariant neurons begins with the assumption that neurons receive input from several lower-level neurons that are identical except for the location of their receptive fields in space. Each lower-level neuron responds to a set of K features $V = \{\mathbf{v}_i\}$. The projections of the stimulus on these features is

$$\mathbf{x}_\mathbf{z} = \mathbf{s}_\mathbf{z} \cdot V. \tag{3.1}$$

$\mathbf{z}$ is the position of a particular subunit. The set of all vectors $\mathbf{z}$ is G. $\mathbf{s}_\mathbf{z}$ is the translated stimulus $T_{\mathbf{z}}\mathbf{s}$, where $T_{\mathbf{z}}$ is a translation operator that shifts the stimulus by $\mathbf{z}$. As with the regular LN-model, the response of each subunit is an arbitrary nonlinear function of the linear subspace $f(\mathbf{x}_{\mathbf{z}})$ The set of responses are then fed into a position invariant function that depends does not depend on which $\mathbf{z}$ produced which response.

Two biologically plausible invariant functions are the OR function

$$\hat{y}_{\text{or}}(\mathbf{s}) = 1 - \prod_{\mathbf{z}\in G} \left(1 - \hat{y}(\mathbf{s}_{\mathbf{z}} \cdot \mathbf{v})\right), \tag{3.2}$$

which fires whenever one of the subunits fires, and the MAX function

$$\hat{y}_{\max}(\mathbf{s}) = \max_{\mathbf{z}\in G} \hat{y}(\mathbf{s}_{\mathbf{z}} \cdot \mathbf{v}), \tag{3.3}$$

which responds with the maximum of the subunit responses. In the case where f is the probability of a binary response, two functions are similar in their output except in the case $\hat{y}(\mathbf{x}_j)$ is significantly greater than 0 for more than one subunit without being close to 1.

3.1 Maximum projection

An advantage of the MAX function is that if $\hat{y}(x)$ is one-dimensional and monotonic, the maximum response is associated with the maximum projection of the stimulus on to the filter. Under this condition, we can rewrite Eq. 3.3 as

$$\hat{y}_{\max}(\mathbf{s}) = \hat{y}(x_{\max}), \tag{3.4}$$

where

$$x_{\max} = \max_{\mathbf{z}\in G} \mathbf{s}_{\mathbf{z}} \cdot \mathbf{v} \tag{3.5}$$

This form of the nonlinearity makes the problem amenable to existing forms of MID with slight modification. Once again, we can derive $\hat{y}$ from $P(x_{max})$ and $P(x_{max}|\text{spike})$. $\langle \mathbf{s}|x\rangle$ and $\langle \mathbf{s}|x, \text{spike}\rangle$ become $\langle \mathbf{s}_{\mathbf{z}_{\max}}|x_{\max}\rangle$ and $\langle \mathbf{s}_{\mathbf{z}_{\max}}|x_{\max}, \text{spike}\rangle$ where

$$\mathbf{z}_{\max} = \underset{\mathbf{z}\in G}{\text{argmax}}\ \mathbf{s}_{\mathbf{z}} \cdot \mathbf{v}. \tag{3.6}$$

The advantage of this method is in its relative simplicity. Once the location of the maximum projection is found, it is straightforward to run MID using the projections and stimuli associated with these locations. The disadvantages come from the assumptions. While monotonic nonlinearities do exist, neuronal responses are often non-monotonic, especially in deeper sensory areas.

For more than one dimensional, we continued to use the maximum projection along the first dimension to choose the location associate with a spike. Specifically,

$$\mathbf{z}_{\max} = \operatorname*{argmax}_{\mathbf{z} \in G} \mathbf{s}_{\mathbf{z}} \cdot \mathbf{v}_1. \tag{3.7}$$

Additional dimensions automatically imply that the maximum projection of the first dimension is not necessarily associated with the maximum response unless the other dimensions have no effect. If the other dimensions are relevant, the function is no longer a MAX function but is instead some other function that while still spatially invariant is no longer the product of independent subunits. We named this algorithm invariant MID (IMID).

3.2 IMID on model cell

We tested IMID and MID on a model cell to demonstrate the differences in performance when a cell fits the assumptions of IMID. The model consists a group of subunits with a single curved Gabor filter, shown in Fig. 3.1A. The subunits are located in a square grid pattern. The locations of the 3 × 3 grid are shown by the black x's on the figure. For the 5 × 5 and 17 × 17 grids, the subunits are spread over the same spatial area but packed more densely, which provides a finer spatial resolution to the model's invariance. For the 5 × 5 grid, the additional x's are located halfway between the x's of the 3 × 3 grid while the 17 × 17 grid has x's at every pixel inside the area of invariance. For each subunit, the stimulus was dotted with the filter, and the subunit spiked if the projection plus noise exceeded a threshold. This results in an error function nonlinearity. The model cell spikes if any of the subunits spiked. In this binary case, this is equivalent to either an OR or a MAX function. However, with respect the the subunit nonlinearity, it is

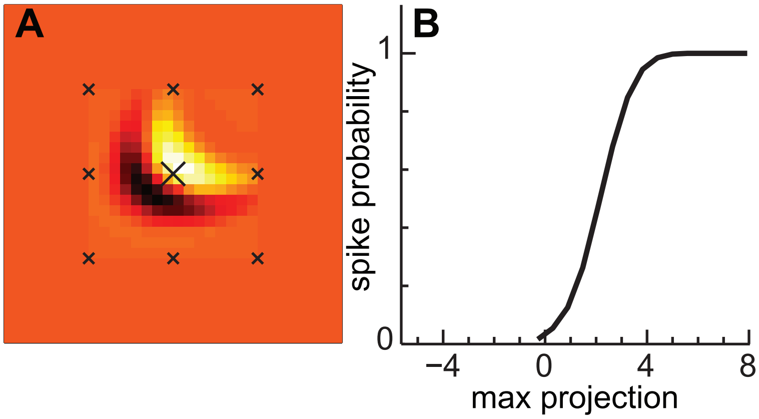

Figure 3.1: **Invariant model cell.** (**A**) The model filter is a curved Gabor function. The x's indicate the locations of the centers of the translated subunits for the 3 × 3 grid. The 5 × 5 grid and the 17 × 17 grid have the spatial extent but a finer spacing of the subunits. (**B**) The subunit nonlinearity for a threshold θ of 2.5 and a noise level σ of 1.0 in units of standard deviation and a mean of zero. The combination of a threshold and Gaussian noise causes the nonlinearity to be a Gauss error function.

an OR function.

We tested the algorithm on multiple variations of this model. Varying the spike threshold changed the mean firing rate and varying the noise level changed the reliability of the spikes. We also tested 3 × 3, 5 × 5, and 17 × 17 translation grids to test how the number of subunits affected the quality of the reconstruction and whether the grid can be determined from the data. We also varied the number of repetitions of the stimulus that we analyzed to explore the effect of increasing amounts of data on the reconstruction of the model.

First, we analyzed the invariant model using MID to demonstrate the need for a novel algorithm. Fig. 3.2 shows the result of running MID on a mode cell with a 3×3 translation grid. The reconstructed filter is a combination of the translated templates, which obscures the underlying structure. Because of the mismatch in the reconstructed dimension, MID also fails to recover the model's nonlinearity (Fig. 3.2B, solid line). Even if we incorporate the model filter into a non-invariant model with, the nonlinearity still differs from the model.

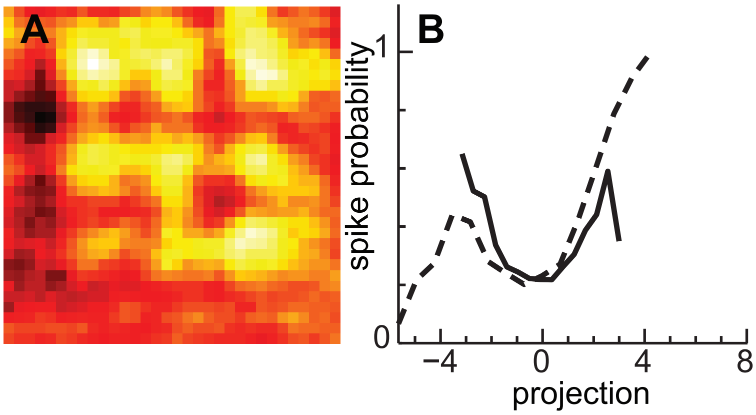

Figure 3.2: **MID reconstruction of** 3×3 **invariant model.** (**A**) 32×32 pixel reconstruction of the model filter. The recovered dimension is a combination of the translated model templates. The model had a 3×3 translation grid, a spike threshold θ of 2.5, a noise level σ of 1.0, and 20 stimulus repetitions. (**B**) The nonlinearity associated with the estimated template (solid line) and the model template without invariance (dashed line). MID is also unable to recover the model nonlinearity. Even if it were to recover the model template, the non-invariant nonlinearity differs from the model nonlinearity.

While MID is insufficient for characterizing this type of translation-invariant cells, the question remains of whether IMID can reconstruct the model. Fig. 3.3A shows the reconstruction of the 3×3 grid model. The overlap was 0.898 ± 0.011 and the invariant reconstruction explained $96.9 \pm 0.8\%$ of the information. Increasing the number of subunits to a 5×5 model (Fig. 3.3B) decreased the overlap to 0.78 ± 0.02 and the information explained to $82.6 \pm 0.3\%$. IMID was also able to recover the subunit nonlinearity (Fig. 3.3C) and copy the model's response to the stimulus (Fig. 3.3D).

3.3 IMID on complex V1 neurons

To compare the performance of IMID with MID, we ran both algorithms on 55 complex cells from the experiment described in App. D.1. We chose to analyze only complex cells because their characteristic phase invariance could arise from

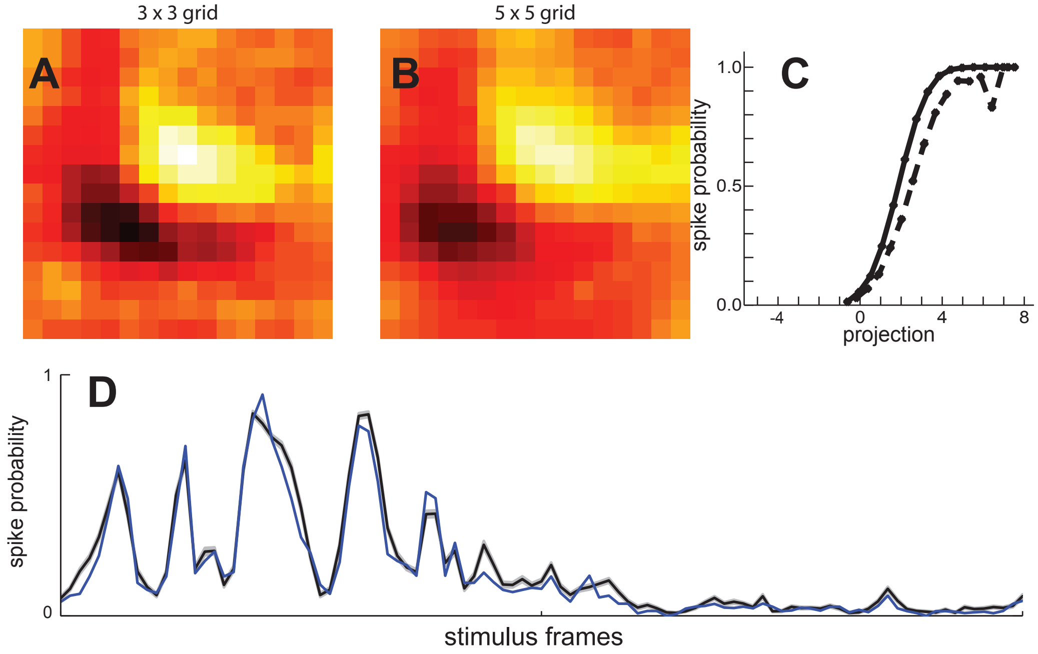

Figure 3.3: **IMID on invariant model cell.** (**A**) Reconstruction of 3 × 3 grid model had an overlap of 0.898±0.011 and explained 96.9±0.8% of the information. (**B**) Reconstruction of 5 × 5 grid model had an overlap of 0.78 ± 0.02 and explained 82.6 ± 0.3% of the information. Increasing the number of subunits decreased the performance of the algorithm but did not prevent invariant MID from recovering the model template. (**C**) Model (solid) and reconstructed (dashed) nonlinearities for the 3 × 3 grid model. Invariant MID was able to successfully recover the model nonlinearity in addition to the template. Projection is scaled to have zero mean and a standard deviation of 1. (**D**) Comparison between the model spike probability (black line with gray areas indicating standard error) and the predictions of the reconstructed cell (blue) for a novel set of frames not used in estimating the model.

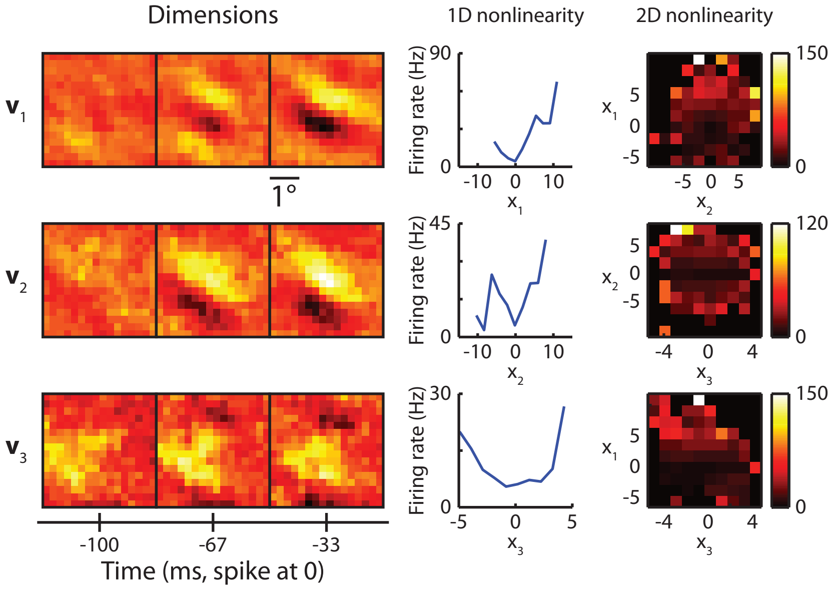

Figure 3.4: **IMID on example complex cell.** The projection along the first dimension determines which subunit determines the neuron's response. The associated one-dimensional nonlinearity is approximately monotonic, which is expected given the structure of IMID. The second and third dimension are sensitive to edges along the same orientation with larger spatial frequency and more weakly modulate the neuron's response. Cell 883–2.

combining the responses of spatially offset neurons.

Figs. 3.4 and 3.5 show the results of running IMID and MID on an example V1 complex cell which was better fit by IMID (as measured by the mutual information between the model subspace and a novel set of corresponding responses).

Fig. 3.6 shows a comparison between the performance of MID v. IMID. For the one-dimensional models (A), IMID performs better on the population of complex cells (Wilcoxon signed rank test, $p = 4 \times 10^{-4}$). This continues to be true for the comparison between the two-dimensional models (B, Wilcoxon, $p = 1 \times 10^{-4}$). For the three-dimensional models (C), neither IMID nor MID are significantly more likely to perform better than the other (Wilcoxon, $p = 0.2$).

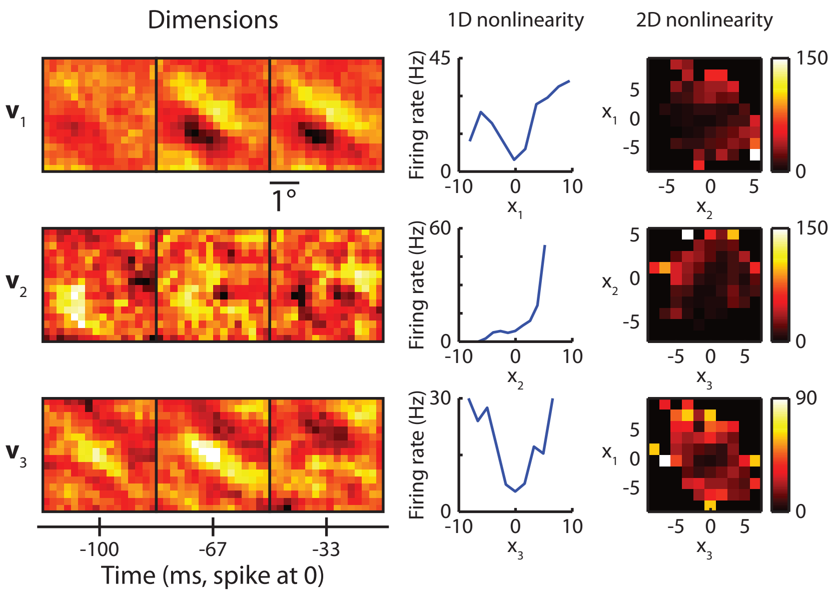

Figure 3.5: **MID on example complex cell.** The first and third dimension form a pair of Gabor wavelet filters with similar spatial frequency and orthogonal spatial phase. They are combined using approximately an energy model. The second dimension is sensitive to motion along a particular direction. Cell 883–2.

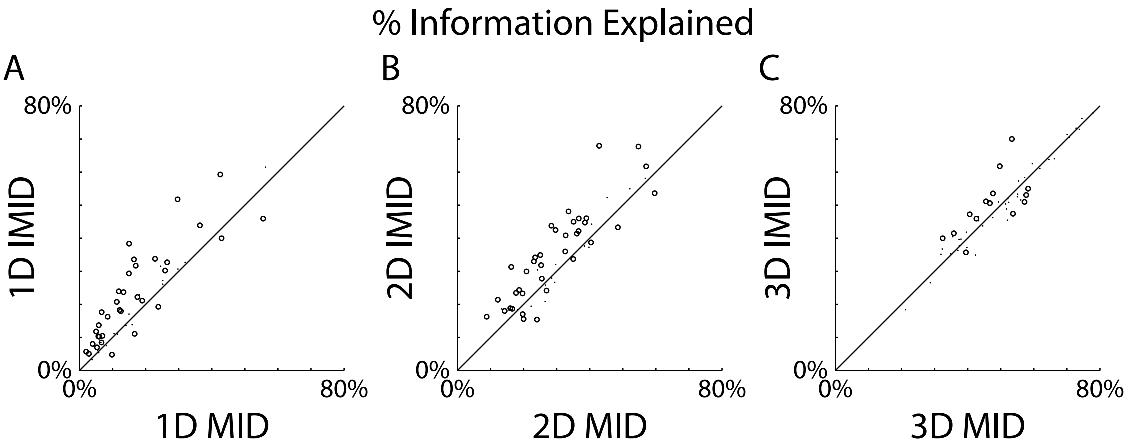

Figure 3.6: **Performance of IMID and MID.** (**A**) For a one-dimensional model, IMID performs better across the population than MID. (**B**) For a two-dimensional model, IMID continues to perform better across the population. (**C**) MID and IMID are not different across the population for the three-dimensional model.

3.4 Discussion

In this chapter, we extended MID to include a form of translation invariance. This algorithm approximated a maximum operation across a set of identical spatially offset monotonic subunits. We demonstrated that IMID was able to recover the original feature of a model cell when MID recovered a combination of offset features that obscured the underlying computation.

We applied this algorithm to complex cells from V1 and found that a one-dimensional IMID model was better able to separate spiking and non-spiking stimuli than a one-dimensional MID model. A two-dimensional IMID model was also better than a two-dimensional MID model, but the three-dimensional IMID model was not significantly more likely to better than the MID model. This may be an effect of our approximation of the maximum function. Selecting the maximum projection along the first dimension selects the location with the largest response for a monotonically increasing nonlinearity, but the monotonic condition is not defined for multiple dimensions. The more a secondary dimension modulates the subunit firing rate (and therefore the more informative it is), the less likely the subunit with the maximum projection along the first dimension is also the subunit with the maximum firing rate. This may limit the usefulness of additional dimensions. However, IMID is still likely to explain more information with a one-dimensional model.

Ch. 3 draws upon work published in Eickenberg et al. (2012). The dissertation was co-primary author of that publication.

Chapter 4

Quadratic MID

Another potential way around the curse of dimensionality is to extend MID to work with quadratic projections. This procedure has similarities to STC. Instead of just a linear filter $\mathbf{v}$, quadratic MID (QMID) also adds a quadratic filter J so that the projections are given by

$$x = \mathbf{v} \cdot \mathbf{s} + \mathbf{s}^T J \mathbf{s}. \tag{4.1}$$

As with STC, we take the eigenvectors of the matrix J and select the dimensions with significantly non-zero eigenvalues. The linear filter $\mathbf{v}$ is also selected if its magnitude is the same as the significant eigenvalues.

QMID has the advantage that it only requires the calculation of a one-dimensional information while still being able to find multiple relevant dimensions. The disadvantage is that it requires the optimization of $D(D+3)/2$ filter parameters. (This is less than $D^2 + D$ because we can constrain J to be a symmetric matrix.) Furthermore, while the algorithm can find multiple linear dimensions, the nonlinear function f is a one-dimensional in the quadratic space, which limits its ability to represent an arbitrary non-linearity. As with STC, this should not be a problem. The primary goal is to elicit the stimulus dimensions that modulate the neural response. Once the high-dimensional stimulus is reduced to a low-dimensional relevant subspace, more flexible methods can be used to characterize the actual nonlinearity. By first finding the reduced subspace, we reduce the complexity of the characterization of the nonlinearity

4.1 Quadratic Maximum Noise Entropy

A similar approach fits a maximum entropy distribution to the data while constraining the measured moments between the stimulus and response. Fitzgerald et al. (2011b) dubbed this approach Maximum Noise Entropy (MNE). The mutual information information between the stimulus and responses is the difference between the entropy of the responses (the response entropy) and the entropy of the responses conditional on the stimulus (the noise entropy). Unlike previous maximum entropy studies that analyze the entropy of the responses of many neurons, MNE takes the response entropy of a single neuron as a given and maximizes the noise entropy given the measurements, hence the name. Quadratic MNE (QMNE) matches the measured zeroth-, first-, and second-order moments. In this case, the nonlinearity takes the form

$$\hat{y}(\mathbf{s}) = \sigma(a + \mathbf{v} \cdot \mathbf{s} + \mathbf{s}^T J \mathbf{s}), \tag{4.2}$$

where σ is a logistic function

$$\sigma(x) = \frac{1}{1 + e^{-x}}. \tag{4.3}$$

The parameters of the equation are chosen such that

$$\begin{array}{rcl} \langle y \rangle & = & \langle \hat{y} \rangle \\ \langle y\mathbf{s} \rangle & = & \langle \hat{y}\mathbf{s} \rangle \\ \left\langle y\mathbf{s}\mathbf{s}^T \right\rangle & = & \left\langle \hat{y}\mathbf{s}\mathbf{s}^T \right\rangle \end{array} . \tag{4.4}$$

The form of Eq. 4.2 ensures that all other moments are not constrained beyond what is required to satisfy the constraints of Eq. 4.4. The relevant features are found in a manner identical to QMID: the eigenvectors of J with significantly large eigenvalues as well as $\mathbf{v}$ if its magnitude is comparably large. This algorithm is an extension of the one proposed by Fitzgerald et al. (2011b), which characterized the nonlinearity in the reduced subspace rather than finding the subspace itself.

The advantage of maximum entropy is the ease of fitting the parameters. The likelihood function is convex, so a simple gradient ascent algorithm is sufficient. The disadvantage compared to QMID is the more constrained nonlinearity. However, like with QMID, the hope is that mismatch of the nonlinearity will not affect the relevant dimensions selected by the algorithm.

4.2 Analysis of simulated cells

We compared QMID and QMNE on two simulated cells. The first was a two-dimensional cell to test their performance in a regime compatible with existing methods. The second was a six-dimensional cell to test the ability of QMID and QMNE to find several relevant dimensions. We also ran MID and STC to provide a baseline for comparison.

Both simulations consisted of 16 × 16 pixel spatial dimensions. The model cells were stimulated with 20000 randomly selected from the van Hateren image database (van Hateren, 1997). This image sequence was repeated 100 times during the simulated experiment.

As with other sections, we will use the subspace overlap (Appendix B) to evaluate the performance of the various algorithms on the simulated experiments.

4.2.1 Two-dimensional model cell

The first model cell had two relevant dimensions consisting of Gabor wavelets with identical location, orientation, and spatial frequency but with orthogonal spatial phases. The model's firing rate was proportional to the sum of the square so the projections:

$$f(\mathbf{x}) \propto x_1^2 + x_2^2. \tag{4.5}$$

This is an energy model. The nonlinearity erases information about the spatial phase of the stimulus, so the model is phase-invariant like the quintessential complex cell in the primary visual cortex.

In this case, MID does quite well. It achieving an overlap of 0.98, which indicates it almost completely reconstructed the model filters. STC does worse with an overlap of 0.77. QMNE was in between with an overlap of 0.90. These reconstructions are shown in Fig. 4.1.

Overall, QMID did poorly for the two-dimensional model cell. A version without a linear component had an overlap of 0.68 (not shown in Fig. 4.1). Adding the linear component improved the overlap to 0.70. Even when we used the results of QMNE as a starting estimate, the overlap of the final estimate was only 0.87,

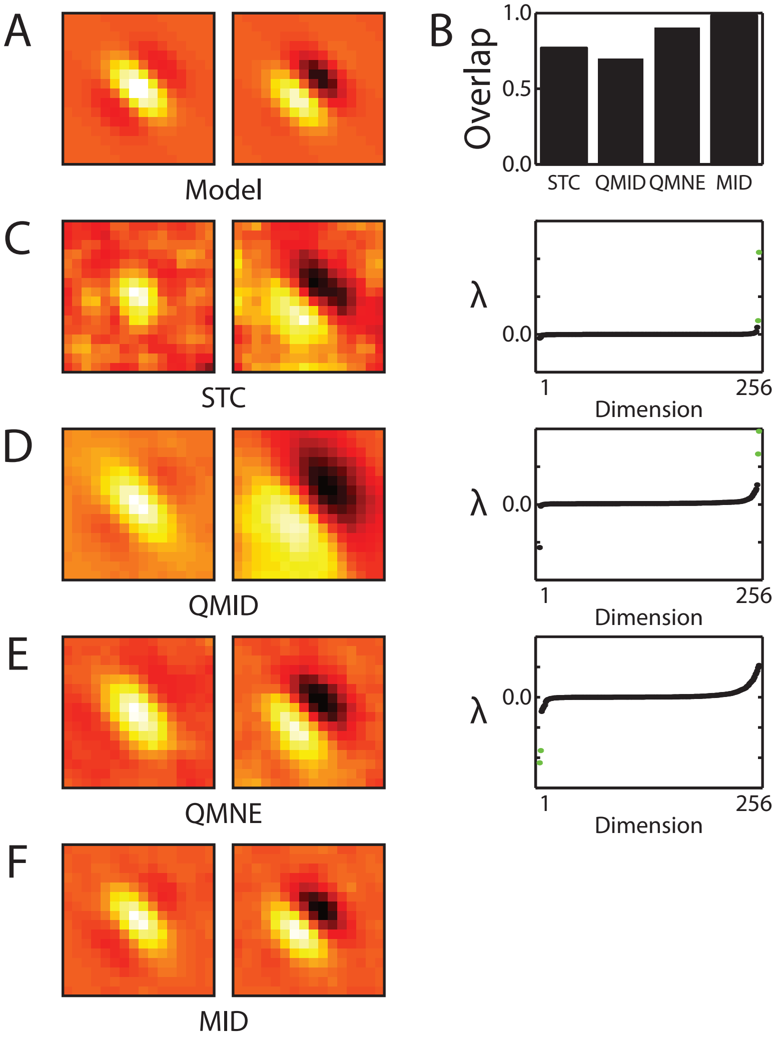

Figure 4.1: **Quadratic methods on two-dimensional model.** (**A**) Model dimensions. The dimensions are Gabor functions with identical parameters except that the spatial phases are orthogonal. The projections of the stimulus on to these dimensions were squared and summed to give the mean firing rate. (**B**) Overlap of reconstructed dimensions with model for the four methods. (**C**) STC performed moderately well with an overlap of 0.77. The reconstructed dimensions appear on the left ordered so that they are in the same column as the closest match with the model dimensions. The eigenvalues are on the right. The two largest eigenvalues were both excitatory and are marked in green. (**D**) QMID performed the worst of the four methods with an overlap of 0.70. (**E**) QMNE performed the best of the three quadratic methods with an overlap of 0.90. (**F**) MID almost perfectly reconstructed the model subspace with an overlap of 0.98. The dimensions were rotated to align with the model to aid visual comparison.

lower than that of the starting point.

In this context, MID is clearly the best method. Using information allows it to bypass stimulus correlations, linear filters have a relatively small number of parameters to fit, and for two dimensions, calculating the probability distributions and conditional stimulus expectation values is still feasible. QMID is hampered by the large number of parameters as well as the non-convex nature of the mutual information. STC covariance both is biased by the stimulus correlations and has a relatively large number of parameters. Finally, QMNE is able to do relatively well considering its large number of parameters, a feat that is partially due to having a convex objective function.

4.2.2 Six-dimensional model cell

The second model cell had three sets of paired dimensions with orthogonal spatial phases. The first pair, associated with x_1 and x_2, are excitatory features similar to the two-dimensional model cell in Section 4.2.1. The second pair, associated with x_3 and x_4, are identical to the first pair except orthogonal in spatial orientation. This pair is used to create cross-inhibition. The final pair, associated with x_5 and x_6, has the same orientation as the first pair, but the spatial location is outside the center rather than inside. This pair is used to create surround inhibition. The model dimensions are shown in Fig. 4.2A. The probability of a spike is

$$f(\mathbf{x}) \propto \frac{x_1^2 + x_2^2}{1 + x_3^2 + x_4^2 + x_5^2 + x_6^2}. \tag{4.6}$$

MID is unable to find 6 dimensions. Using 10 bins per dimension, the probability distributions will have 1,000,000 bins, which much larger than the number of stimuli that are likely to be used for any realistic experiment. Reducing the noise due to counting error would require even more data.

For this more complicated model, STC (Fig. 4.2C) does poorly with an overlap of 0.29. Meanwhile, QMNE (Fig. 4.2E) does well with an overlap of 0.85, which is only slightly worse than its performance on the two-dimensional model.

Without a linear filter (not shown), QMID had an overlap of 0.64. Including a linear filter (Fig. 4.2D), improves the reconstruction slightly with an overlap of

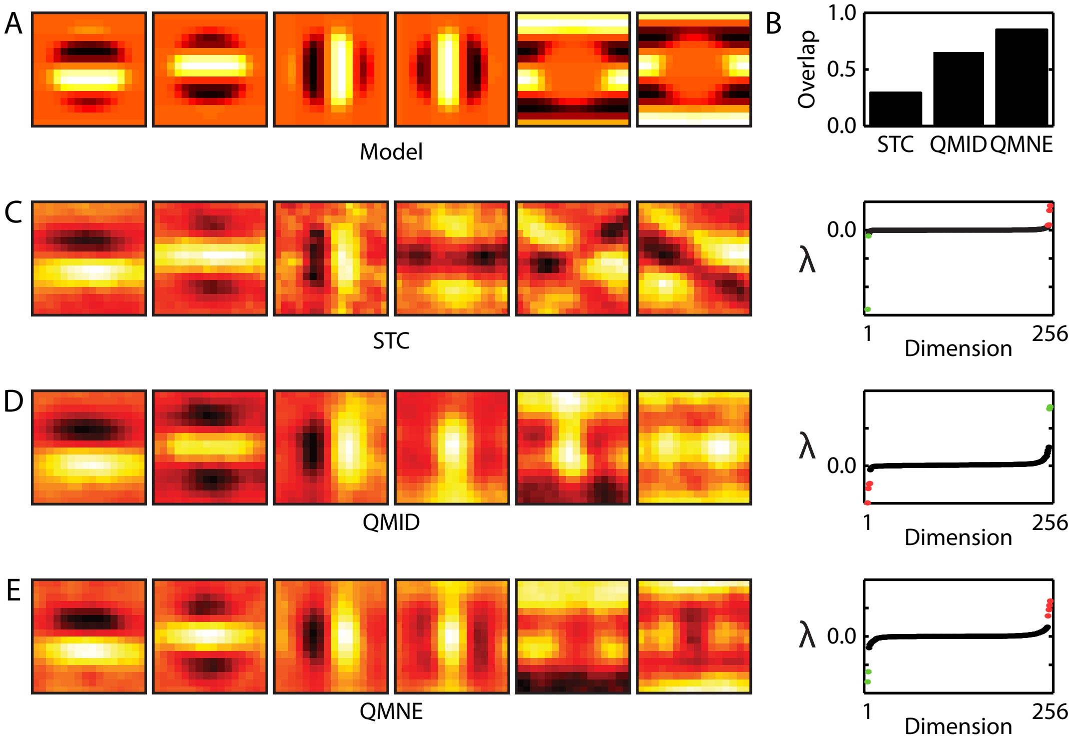

Figure 4.2: **Quadratic methods on six-dimensional model.** (**A**) Six model dimensions. The left two were excitatory and the right four were inhibitory. (**B**) Overlap of the reconstructed dimensions with the model subspace for the quadratic methods. MID is not included because calculating a six-dimensional information is beyond the limits of realistic amounts of data. (**C**) STC performed poorly with an overlap of 0.29. Reconstructed dimensions are on the left and are ordered to correspond with the closest model dimension. The eigenvalues are on the right with the six largest eigenvalues colored red for excitatory and green for inhibitory dimensions. (**D**) QMID performed better than STC with an overlap of 0.65. (**E**) QMNE performed the best with an overlap of 0.85. The match with the model dimensions is visually apparent.

0.65. Again, using QMNE as a starting estimate lead to a lower overlap (0.84) than the starting estimate.

Again, QMNE performs the best of the quadratic methods. Additionally, the performance of QMID and QMNE decreased only slightly with the addition of four additional dimensions. This may be because finding that the significant dimensions also requires determining that the other dimensions are insignificant.

4.3 Discussion

In this chapter, we examined the performance of two novel methods for finding the subspace of the stimulus that modulates a neuron's response. Both methods were extensions of STC in that they looked at pairwise interactions between stimulus dimensions. The first method, QMNE, builds a maximum entropy model of the responses given the stimulus constrained by the 0^{th}, 1^{st}, and 2^{nd}-order correlations of the stimulus with the neuron's response. The second method, QMID, searched for a quadratic and linear filter with an output that was most informative about the neuron's response.

We tested these methods on two model cells along with STC and MID for comparison. For the two-dimensional model cell, we found that MID was best able to recover the stimulus subspace that we used to generate the model responses. We believe that this was due to only having to fit two linear filters of size D rather than a matrix of size D^2. The additional number of parameters related to the two-dimensional nonlinearity is much smaller than that difference. Of the quadratic methods, QMNE performed the best, followed by STC and QMID.

For the six-dimensional model cell, QMNE performed the best. QMID came in second, while STC performed relatively poorly. Six dimensions is beyond the ability of MID because realistic amounts of data are spread too thinly across the six-dimensional nonlinearity for a meaningful representation.

QMID performed more poorly than QMNE despite QMID having an arbitrary nonlinearity. QMNE does have the advantage of having a convex error function, which simplifies optimization.

Rajan and Bialek (2012) also considered quadratic extensions of MID. They expanded the method to include mutliple quadratic filters as well as using low-rank filters and matrix basis functions to simplify the optimization.

Ch. 4 draws on work published in (Fitzgerald et al., 2011a). The dissertation author was the secondary author of the paper. This chapter includes the portions of the work relevant to the author's contribution.

Chapter 5

Invariant Logistic Subunits

Invariant MID (Chapter 3) allows us to model neurons that respond to the same features at multiple locations, but it makes a few assumptions that limit its effectiveness. First, it treats each location equally. While this creates invariant responses, a more limited but biologically plausible conception of invariance occurs when the ranking of responses to stimuli are preserved even if the absolute level of the response is modulated (Pasupathy and Connor, 1999). Second, determining the extent of invariance requires running the algorithm multiple times, which is computationally intensive, and it may be difficult to distinguish between the multitude of models. Third, in order to simplify the calculation of the subunit nonlinearities, we made the assumption that it was monotonic and that they were combined with a MAX function. This limits the types of nonlinearities that can be modeled, and it limits the effectiveness of multidimensional models, where monotonic is not defined.

In this chapter, we propose an alternative model that predicts the response as a linear combination of responses of translated linear or quadratic logistic subunits. These subunits take the form of those in Section 4.1. The weighted response is linearly rectified to allow for excitatory and inhibitory locations. Because this model incorporates logistic subunits that are invariant across shifts in the stimulus, we call this methods Invariant Logistic Subunits (ILS).

The parameters of the model are **b**, the weights corresponding to each subunit; c, the rectifier bias term; d, the rectifier scale factor; a, the bias term of

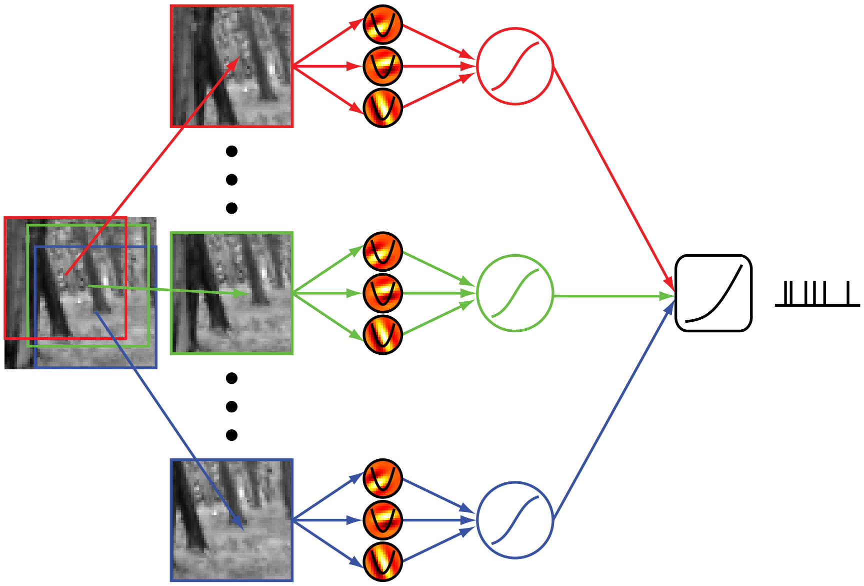

Figure 5.1: **Schematic of ILS.** ILS begins by taking subsets of the full stimulus. In this example, the subsets are offset in space relative to one another, but they could also be offset in time as well as other transformations such as different rotations or scalings of the stimulus. Then, each of these subsets is passed through the same linear and possibly quadratic filters. This example shows different linear filters that are used to create the quadratic filter J. These values are then passed through a logistic function to create a response for each subset of the stimulus. These individual responses are weighted and summed to create a single value, which is rectified using a softplus rectifier to create a predicted firing rate.

the logistic function; $\mathbf{v}$, the linear filter; and possibly J, the quadratic filter. The predicted response is given by

$$\hat{y}(\mathbf{s}) = dR_+\left(c + \sum_{\mathbf{z}\in G} b_{\mathbf{z}}\sigma\left(a + \mathbf{s}_{\mathbf{z}} \cdot \mathbf{v}\ \left(+\mathbf{s}_{\mathbf{z}}^T J \mathbf{s}_{\mathbf{z}}\right)\right)\right). \tag{5.1}$$

σ is the logistic function (Eq. 4.3), and R_+ is the softplus rectifier function

$$R_+(x) = \log\left(1 + e^x\right). \tag{5.2}$$

The softplus rectifier prevents the predicted firing rate from being negative. Like the linear rectifier

$$\max\left(0, x\right), \tag{5.3}$$

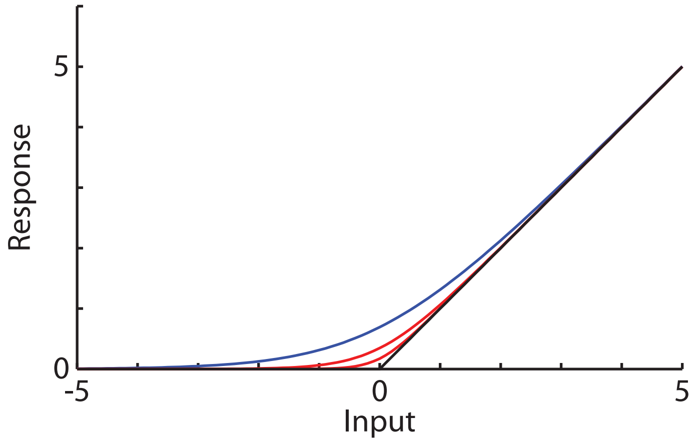

Figure 5.2: **Comparison between linear and softplus rectifiers.** The linear rectifier (black) has a discontinuity at 0 while the softplus rectifier (blue) is continuous while having the same asymptotic behavior. In the form $d \times R_{+}(x/d)$, the softplus rectifier converges to the linear rectifier. The figure shows this function for values of d equal to 0.25 and 0.5 (red).

it goes to zero as x goes to ∞ and is approximately x for large x. Unlike the linear rectifier, the soft plus rectifier has a continuous derivative. Fig. 5.1 is a visual representation of the algorithm.

Fig. 5.2 shows both the linear rectifier (black) and the softplus rectifier (blue). The softplus rectifier converges to the linear rectifier but does not have the discontinuity at 0. The scale factor d controls how close the softplus function is to a linear rectifier. As d goes to 0, the softplus rectifier converges on to the linear rectifier. Fig. 5.2 shows the effect of two intermediate values of d (red).

Vintch et al. (2012) independently created a subunit model with a different structure.

5.1 Single-subunit model cell

Before testing the algorithm's performance on models that incorporate identical subunits, we should test whether the model is able to ignore additional, irrelevant locations. The model consists of a linear ILS model applied to a single stimulus location. The linear filter is a Gabor wavelet (shown in Fig. 5.3).

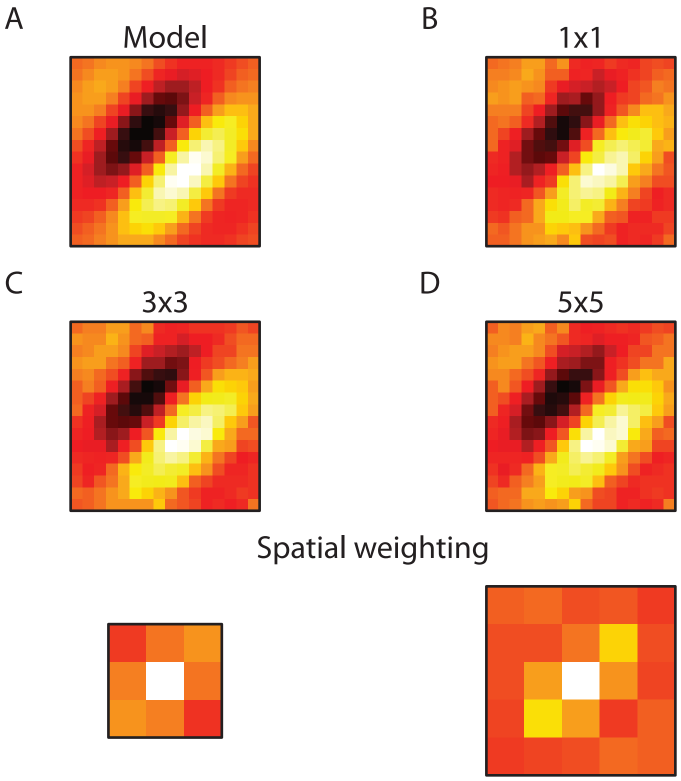

Figure 5.3: **ILS on model with single subunit.** (**A**) Model consisted of a single linear subunit with this filter. (**B**) Reconstructed linear filter using the correct 1×1 grid. (**C**) Reconstructed linear filter (top) and spatial weighting (bottom) using a 3×3 grid. The reconstruction of the linear filter is as good as the $1times1$ grid. The spatial weighting is concentrated at the position of the model subunit, but the other locations have non-zero weights. The largest error is along the axis of the Gabor wavelet. (**D**) Reconstructed linear filter (top) and spatial weighting (bottom) using a 5×5 grid. The reconstruction of the linear filter continues to be good while the reconstruction of the spatial weighting continues to degrade. The errors in the spatial weighting are again largest along the axis of the Gabor wavelet.

Table 5.1: **ILS on model with single subunit.** ILS almost perfectly reconstructed the linear filter even when analyzing stimulus including irrelevant locations. There was a decrease in the quality of the reconstruction of the spatial weights from the 3 × 3 grid to the 5 × 5 grid. Jackknife refers values for individual jackknife estimates (mean ± sem). Averaged refers to values for filter/spatial weighting averaged using PCA (App. C).

	Filter overlap		Mask correlation	
Grid	Jackknife	Averaged	Jackknife	Averaged
1×1	0.9863 ± 0.013	0.9898	—	—
3×3	0.988 ± 0.002	0.991	0.91 ± 0.06	0.94
5×5	0.986 ± 0.003	0.989	0.816 ± 0.017	0.822

We reconstructed this model using 1 × 1, 3 × 3, and 5 × 5 grids centered on the location of the model subunit. All three conditions were able to accurately reconstruct the linear filter (Fig. 5.3 and Table 5.1), and the quality of the reconstructions were not significantly different.

The reconstruction of the distribution of spatial weights was good but imperfect. Both the reconstruction using the 3 × 3 grid and the one using the 5 × 5 grid had the majority of the weight at the location of the model subunit, but the other locations had non-zero weights. This error was largest along the axis of the linear filter's Gabor wavelet.

5.2 Model of sparsifying inhibition

One potential use of ILS is characterizing neurons that have a delayed inhibition trained on the same feature as its excitatory input. This circuit can implement limited adaptation and make the response sparser by preventing the neuron from responding to its preferred stimulus feature unless is present in an intensity greater than its recent history. Fig. 5.4 shows a simple example of a circuit that could implement this function. The blue neuron responds linearly to a particular feature. It provides excitatory input to both the red and black neuron. In turn, the red neuron provides persistent inhibitory input to the black neuron. This arrangement will suppress the response of the black neuron to a steady signal

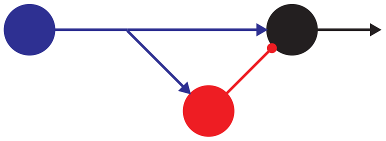

Figure 5.4: **Model sparsifying circuit.** The blue neuron responds linearly to a particular feature and provides excitatory input to the red and black neurons. The red neuron repeats the input from the blue neuron and provides inhibitory input to the black neuron with a delay. In addition, the synapse may provide a sustained inhibition to the black neuron. If excitation and inhibition are balanced, the black neuron will not respond to a steady input from the blue neuron but will respond to increases in the firing rate of the blue neuron.

from the blue neuron. The black neuron will only respond if there is an increase in the output of the blue neuron relative to the recent past.

Fig. 5.5A shows the model spatial and temporal kernels. The model is strongly driven by the frame at time 0 and has decaying inhibition associated with the frames at times −9 through −1. The balance between excitation and inhibition was such that the output to constant input is 0.58 spikes per frame times the subunit response. The inhibition reduces the mean firing rate from 3.1 to 1.0 spikes per frame. While using a definition of sparsity adapted from (Rolls and Tovee, 1995)

$$S = 1 - \frac{\langle r \rangle^2}{\langle r^2 \rangle}, \tag{5.4}$$

inhibition increased the sparsity of the model's response from 0.47 to 0.68. The sparsity ranges from 0 for a constant response to $1 - 1/N$ for a single non-zero response.

We analyzed the model using MID with a single spatiotemporal filter and singular value decomposition (SVD) to separate the filter into spatial and temporal components (Fig. 5.5B). The reconstruction of the spatial filter was very good (overlap 0.979 ± 0.002 across jackknifes, 0.981 for the averaged filter). However, the reconstruction of the temporal kernel was less successful (correlation 0.862 ± 0.005 across jackknifes, 0.862 for the averaged filter). The excitation from time 0 bleeds

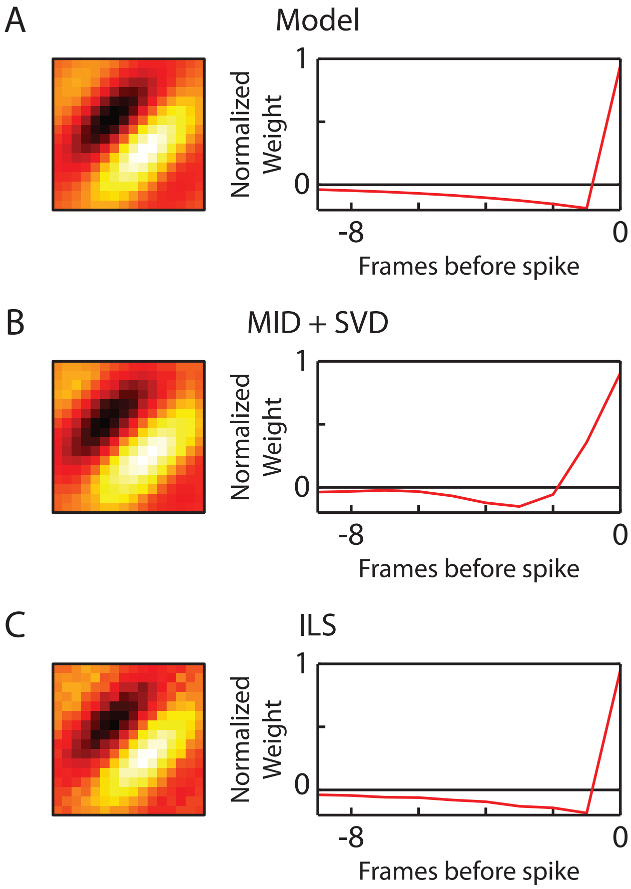

Figure 5.5: **ILS on sparsifying model.** (**A**) The model cell is tuned to an oriented edge. The feature is excitatory if it appears at the current time but is inhibitory if it appears between $1-9$ frames before the present. (**B**) Spatiotemporal filter from MID decomposed into a spatial and a temporal filter using SVD. The reconstruction of the spatial filter has an overlap of 0.979 ± 0.002. The reconstruction of the temporal profile is less successful. Instead of a sharp transition between excitatory and inhibitory contributions between the first and second frame before the present, the second frame is excitatory and the inhibition does not peak until the fourth frame. The correlation with the model temporal kernel is 0.862 ± 0.005. (**C**) Linear spatial filter and temporal kernel from ILS. ILS fits the spatial filter slightly better than MID with an overlap of 0.9906 ± 0.0004. The fit of the temporal kernel is much better, with a correlation with the model of 0.99980 ± 0.00006.

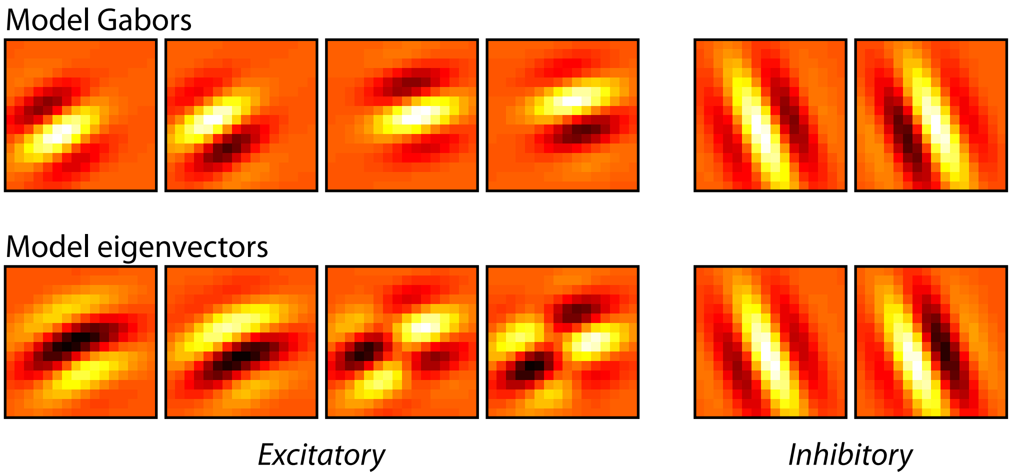

Figure 5.6: **Quadratic model cell.** (Top) Six features used to create the J matrix. Two excitatory pairs of quadrature Gabor pairs positioned on different positions of a curved contour and one inhibitory pair aligned orthogonal to the curve. (Bottom) Eigenvectors of the model J matrix. Because the excitatory pairs overlap with each other, the excitatory eigenvectors are pairs of the sums and differences of the original features.

into time -1 and the inhibition peaks at time -3. This error occurs from using a single dimension to model a fundamentally ten-dimensional function.

ILS recovered the model almost perfectly (Fig. 5.5C). The overlap with the model of the spatial filter was 0.98399 ± 0.00010 across jackknifes and 0.98741 for the averaged filter. For the temporal kernel, the correlation with the model was 0.99974 ± 0.00011 across jackknifes and 0.99992 for the averaged filter.

5.3 Quadratic models

Having demonstrated the effectiveness of the linear algorithm, we tested the quadratic ILS algorithm.

Our model cell had the form of Eq. 5.1. J was made of three pairs of Gabors. The two excitatory pairs have identical size and spatial tuning but are offset in space and have slightly different orientations. Together they overlap to form a bent line. The inhibitory pair is centered at the intersection of the excitatory pairs and

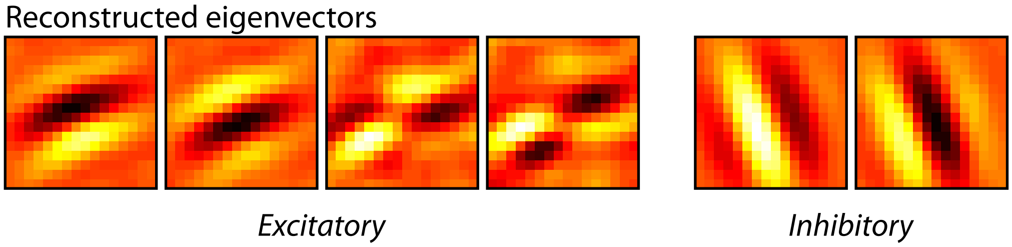

Figure 5.7: **ILS on quadratic model.** Eigenvectors of J matrix reconstructed by ILS. The overlap with the model dimensions is 0.96

has an orientation orthogonal to the average orientation of the excitatory pairs. We chose this model because each pair can be thought of as an energy model complex cell. The excitatory components create sensitivity to a curved feature, and the inhibitory pair provides cross-inhibition. Fig. 5.6 shows the Gabor features that we used to make the J matrix as well as the eigenvectors of the matrix.

We ran ILS on the model cell for four jackknifes. Fig. 5.7 shows the eigenvectors of the reconstructed J matrix, which was averaged using PCA. The overlap between the space spanned by these eigenvectors and the space of the model was 0.96.

5.4 ILS on V4 neurons

We applied ILS to recordings of 161 neurons from macaque visual area V4. The recordings had been collected for an earlier analysis (Sharpee et al., 2013) and are described in Sec. D.2. We chose to apply our method to V4 because previous studies have shown that selectivity begins to shift from retinotopic position to a relative, object-based coordinates in V4 (Gallant et al., 1996), which may be represented using invariant subunits, and sensitivity to shapes such as contours and curved gratings (Gallant et al., 1993, 1996), which can be represented by the unconstrained number of dimensions of the J matrix.

Because the size of the stimulus was fixed, we ran the analysis for two sizes: the full stimulus and a patch with the same center but half the width. Both stimuli were downsampled to 20×20 pixels. We chose the stimulus size for

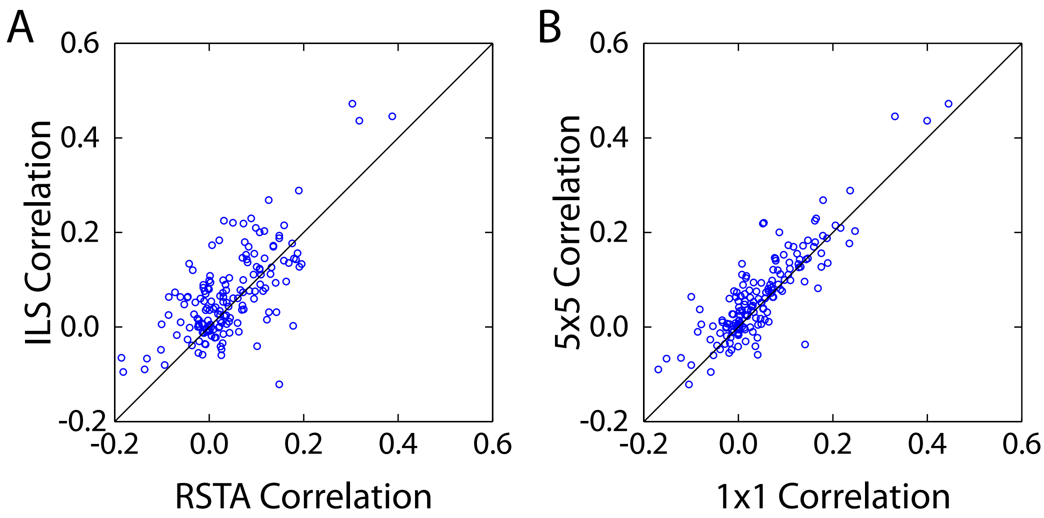

Figure 5.8: **Comparison with RSTA and non-invariant ILS.** (**A**) Correlation of predicted response from RSTA and ILS with novel set of responses. ILS outperformed the RSTA model for 103 out of the 161 V4 neurons. (**B**) Comparison between ILS model with a 5 × 5 spatial grid and a model with only a single subunit. The invariant model performed better than the non-invariant model for 109 out of 161 neurons.

later analysis based on which performed the best on average across jackknifes for likelihood and correlation with respect to the respective training set, the respective cross-validation set, and the combined training and cross-validation set.

From the 20 × 20-pixel stimulus, we extracted 25 16 × 16-pixel patches arranged in a 5 × 5 grid offset by 1 pixel.

5.4.1 Performance of ILS on V4 neurons

To evaluate how well the ILS algorithm was performing on the V4 dataset, we used the correlation between a novel set of spikes and the model's predictions for the associated stimuli. For comparison, we used RSTA (Eq. 1.7). We calculated the nonlinearity using Eq. 1.2 with 10 bins. We reserved 25% of the unrepeated data to use for cross-validation and chose the λ associated with the highest correlation with the cross-validation responses. Fig. 5.8A shows the relative performance. ILS performed better than RSTA for 103 of the 161 neurons.

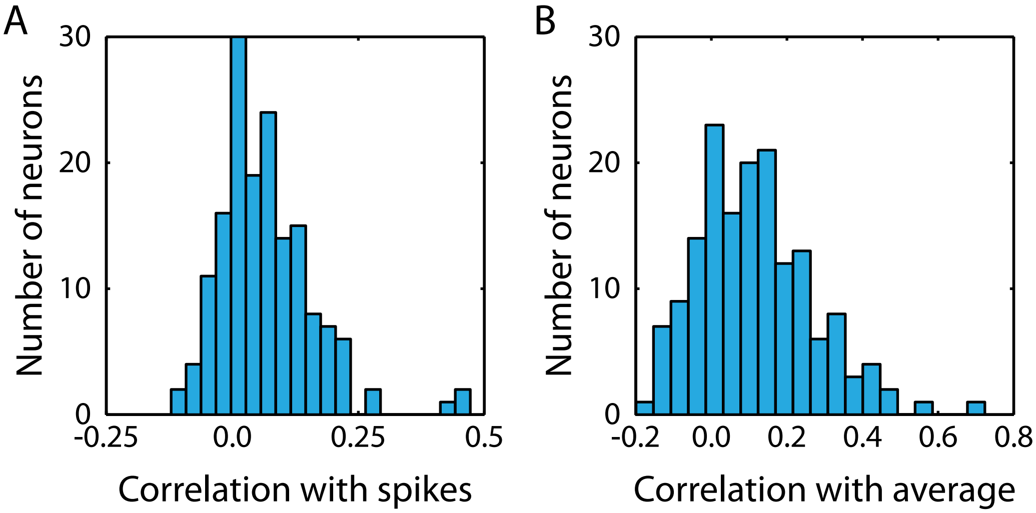

Figure 5.9: **Performance of ILS on V4 neurons.** (**A**) Distributions of correlations between ILS prediction and the spikes from a novel dataset. The median was 0.05. (**B**) Correlations between ILS prediction and the average response to the corresponding frame across repetitions. This reduces the noise associated with variability in the response to the same stimulus The median correlation was 0.10.

To determine the importance of of invariance, we compared the model with a 5×5 grid of 16×16-pixel patches with a model with a single subunit and a 20×20-pixel patch. Fig. 5.8B shows the performance on a novel dataset. The invariant model with a 5×5 grid outperformed the non-invariant model with a 1×1 grid for 109 neurons.

Fig. 5.9A shows the distribution of correlations between the ILS predictions and spikes. They ranged from -0.12 to 0.47 with a median of 0.05. Fig. 5.9B shows the correlation between the ILS prediction and a prediction based on the mean response to a particular frame across repetitions. This partially removes the effects of spike variability and is a better measure of how well the model is predicting the mean response of the neuron. The performance ranged from -0.20 to 0.72 with a median of 0.10.

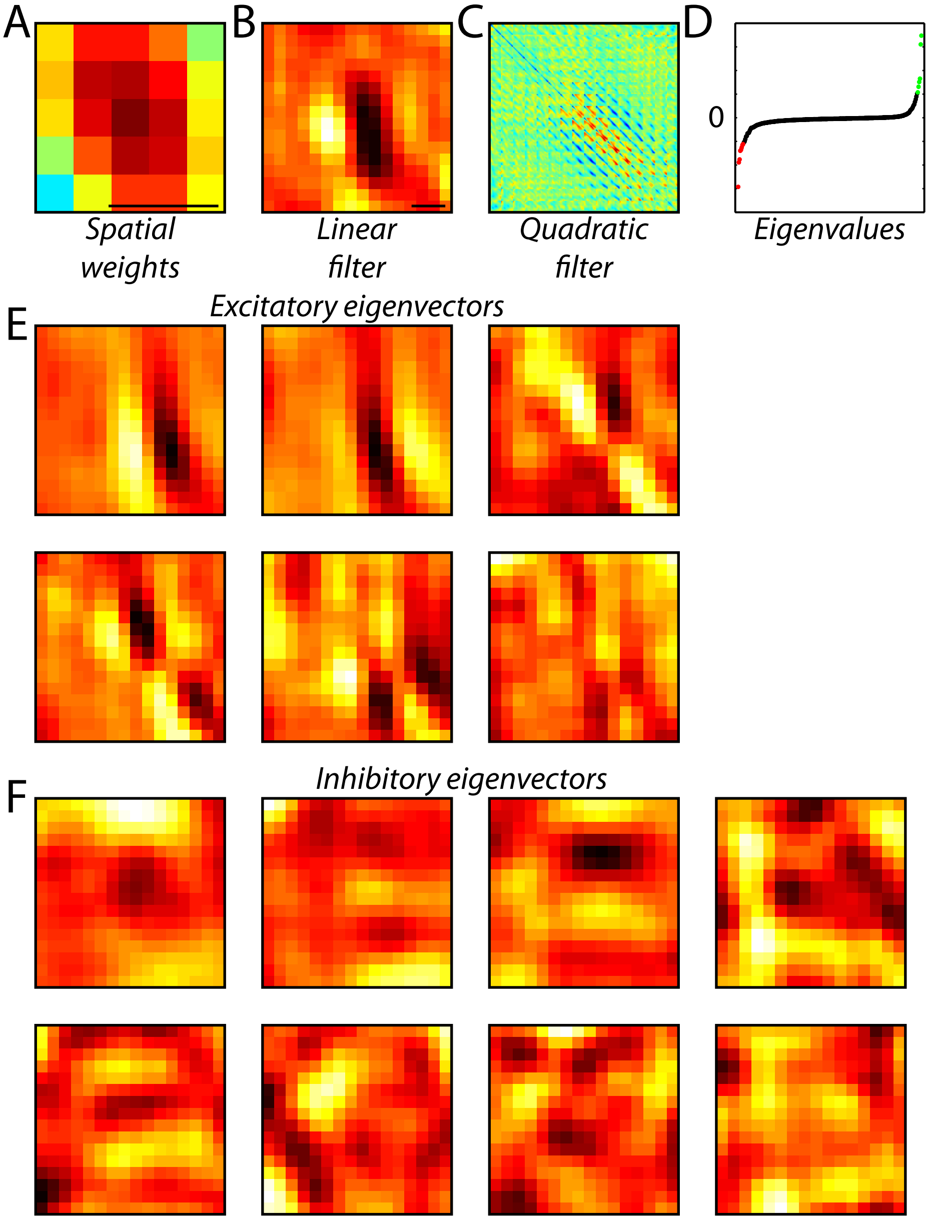

Figure 5.10: **Example cell J47A2 parameters.** (**A**) Spatial weights **b**. Invariance along vertical axis with inhibition in lower left corner. The black bar is 1° wide. (**B**) Linear filter **v**. A vertically aligned Gabor. (**C**) Quadratic filter J. The black bar is 1° wide. The eigenvectors in E and F have the same scale. (**D**) Eigenvalues of J. Six excitatory (green) and eight inhibitory significant eigenvalues. (**E**) Significant excitatory eigenvectors. (**F**) Significant inhibitory eigenvectors.

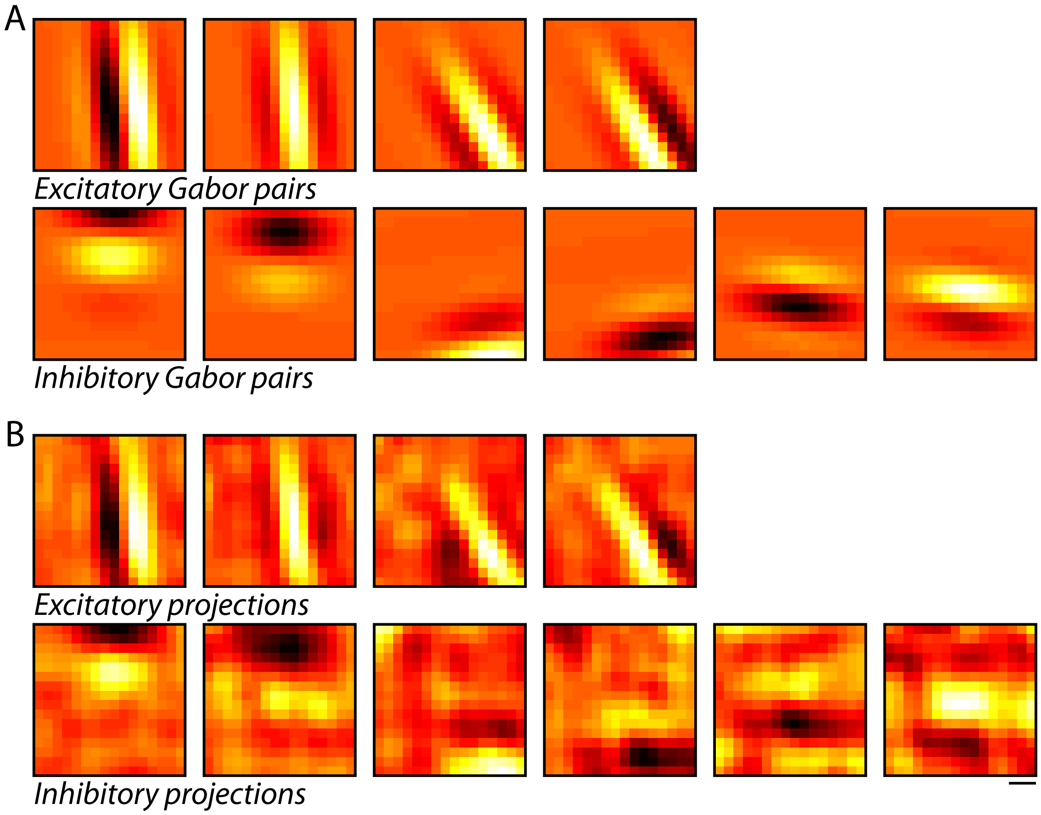

Figure 5.11: **Gabor fit for J47A2.** (**A**) Excitatory and inhibitory Gabor pairs that combine to fit J and are well represented in the subspace of the significant eigenvectors. (**B**) Gabor pairs projected into the space of the significant eigenvectors to show that they approximately exist in that space. The black bar is 1° wide.

5.4.2 Feature selectivity in V4

To analyze the feature selectivity of the V4 neurons, we used the following procedure: First, we excluded all neurons for which the correlations between the jackknife predictions to novel stimuli and the recorded responses were not significantly greater than zero. Second, we averaged the jackknife parameters using PCA (App. C). Third, we determined the significant positive and negative eigenvalues of the J matrix using the procedure described in App. F. Finally, we fit the J matrix with Gabor quadrature pairs (App. G) and kept those pairs which had a length in the space of the significant eigenvectors that was greater than 0.7. We

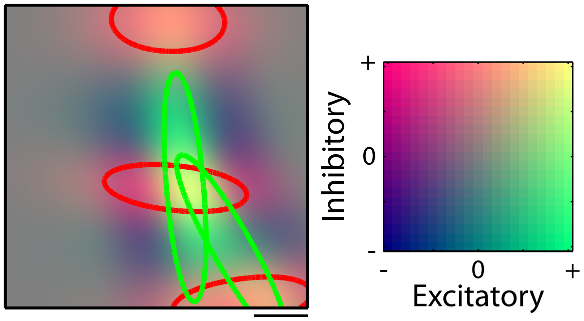

Figure 5.12: **Gabor contours for J47A2.** Contours drawn at $e^{-0.5}$ relative to maximum for zero-phase excitatory (green) and inhibitory (red) Gabors. The two excitatory Gabors form a curved contour while the three inhibitory Gabors are spaced along the contour with orthogonal orientations. The contours are superimposed on the sums of the excitatory (green channel) and inhibitory (red channel) zero-phase Gabors normalized so that the maximum value of the sum corresponds to a color value of 1 and zero corresponds to a color value of 0.5. The blue channel was set to 0.5. The box shows how the color changes as the excitatory and inhibitory sums pass from the negative maximum through zero to the maximum value. The black bar is 1° wide.

used the remaining Gabor features to interpret the neurons' selectivity.

Fig. 5.10 shows the averaged parameters of an example neuron (J47A2). Fig. 5.10A shows the spatial weighting **b**. The weights drop more quickly along the horizontal direction than the vertical revealing an invariance along a direction slightly offset from vertical. The linear filter **v** is a vertical Gabor in the center of the stimulus (Fig. 5.10B). The first two excitatory eigenvectors are a pair of curved Gabors while the remaining eigenvectors are similar to combinations of vertical Gabors (Fig. 5.10E). The inhibitory eigenvectors have many horizontal lines (Fig. 5.10F).

Fig. 5.11A shows the excitatory and inhibitory Gabor pairs that fit the J matrix. The excitatory pairs consist of a central vertical Gabor pair and another overlapping pair offset in position and orientation. Together, they make the curved Gabors seen in the eigenvectors. The inhibitory pairs are all horizontal and positioned along the contour formed by the excitatory pairs. Fig. 5.11B shows the projection of the Gabors into the space of the significant eigenvectors to demonstrate how much of the Gabors are present in this space.

Fig. 5.12 shows the relative positions of the Gabor pairs. For each Gabor pair, we created a single Gabor with a phase ϕ of 0. We drew contours at $e^{-0.5}$ relative to the maximum value of the individual Gabor. Excitatory features are green, and inhibitory features are red. We also summed the excitatory and inhibitory Gabor features together and scaled their values so that the maximum was 1 and 0 was 0.5. We used the inhibitory sum for the red channel, the excitatory sum for the green channel, and set the blue channel to 0.5. The box on the right shows how changing values of the excitatory and inhibitory sums affect the color.

The combined contours more clearly shows the interaction between excitatory and inhibitory features. The excitatory features overlap to form a bent line while the inhibitory features are spaced along the line with orientations orthogonal to it.

Fig. 5.13 shows the parameters for another example neuron (J30A1). The spatial weights for this neuron are more uniform (Fig. 5.13A). The linear filter (Fig. 5.13B) inhibits responses to bright stimuli in the center of the stimulus. Fig. 5.14 and Fig. 5.15 aid the interpretation of the excitatory features. The Gabor fit found three excitatory pairs of Gabors which are all located in the center of the stimulus and have similar spatial frequencies but tile the orientation space. We did not find Gabors that well fit the inhibitory eigenvectors.

Fig. 5.16 shows the Gabor contours and spatial weightings of eight additional examples. J33A1 has three excitatory pairs forming a curve. The two inhibitory Gabor pairs are orthogonal to the curve. The spatial weightings is relatively localized in space. J06A1 has a similar combination of excitatory Gabor pairs without any inhibitory Gabor pairs. The spatial weighting is most localized of the examples. J06A3 has two excitatory Gabor pairs on a curve. The inhibitory Gabor pair is orthogonal to the excitatory curve but is offset in space along the axis orthogonal to the center of the curve. The spatial weighting is relatively uniform. M28A1 has two excitatory pairs forming a curve. Each has an inhibitory Gabor with orthogonal orientation overlapping it. There is also a third pair of

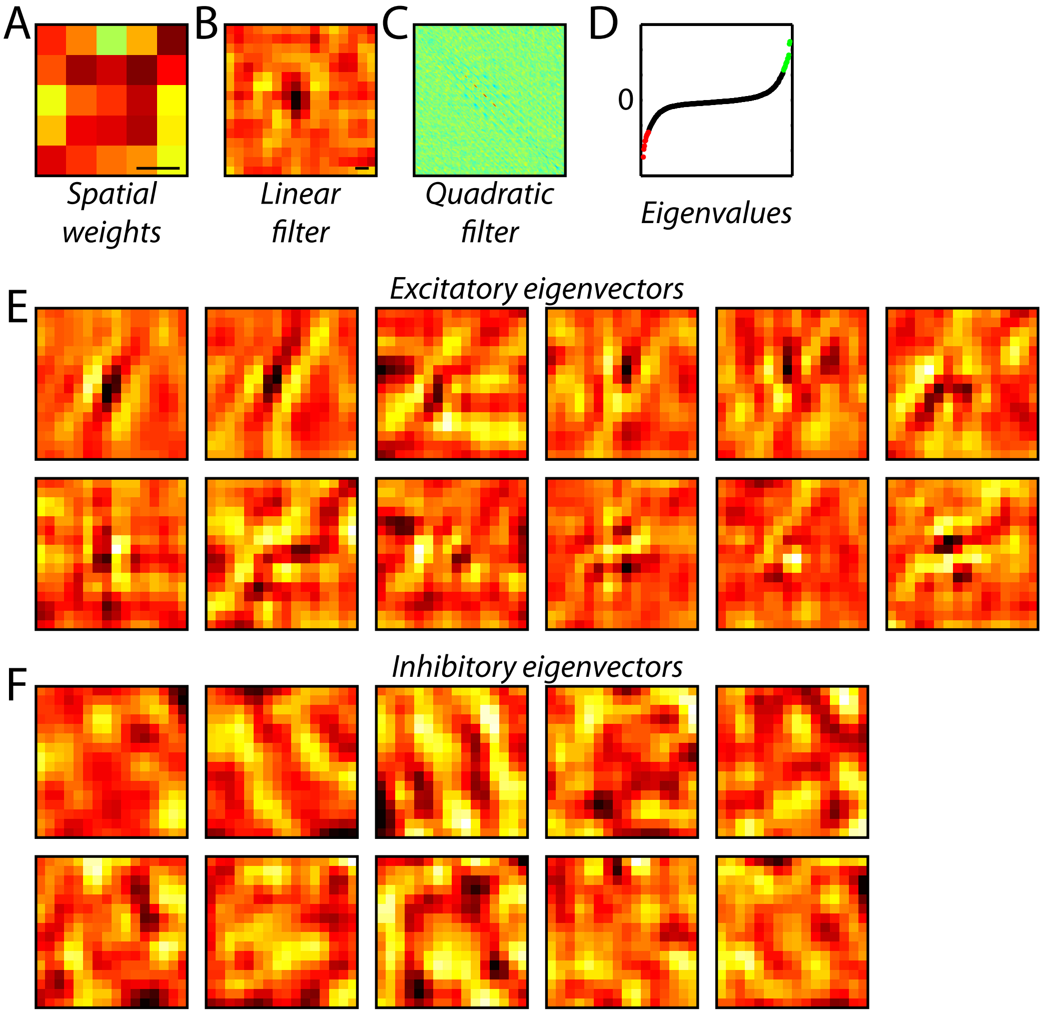

Figure 5.13: **Example cell J30A1 parameters.** (**A**) Spatial weights **b**. Broad selectivity across all locations. The black bar is 1° wide. (**B**) Linear filter **v**. Inhibits response to center of the stimulus. The black bar is 1° wide. All of the eigenvectors in E and F have the same scale. (**C**) Quadratic filter J. (**D**) Eigenvalues of J. Twelve excitatory (green) and ten inhibitory significant eigenvalues. (**E**) Significant excitatory eigenvectors. (**F**) Significant inhibitory eigenvectors.

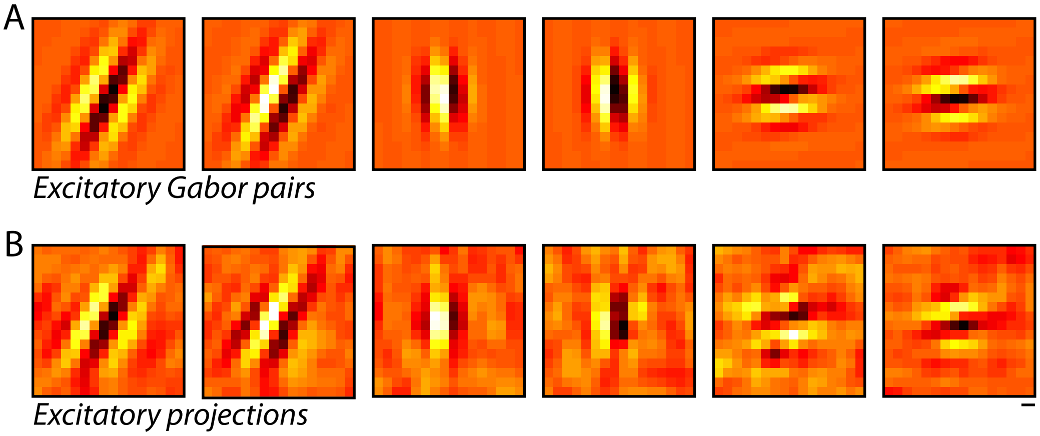

Figure 5.14: **Gabor fit for J30A1.** (**A**) Three excitatory pairs of Gabors positioned in the center of the stimulus with similar spatial frequencies but with orientations along the vertical, horizontal, and diagonal. (**B**) Projection of the Gabor features onto the subspace of significant eigenvectors. The black bar is 1° wide.

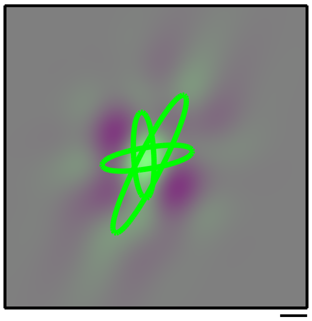

Figure 5.15: **Gabor contours for J30A1.** Contours of zero-phase Gabor for each pair superimposed on the sum of these Gabors. The Gabors are located at the same spatial location but have different orientations, which allows for a rotation invariant response. Colors of background the same as Fig. 5.12. The black bar is 1° wide.

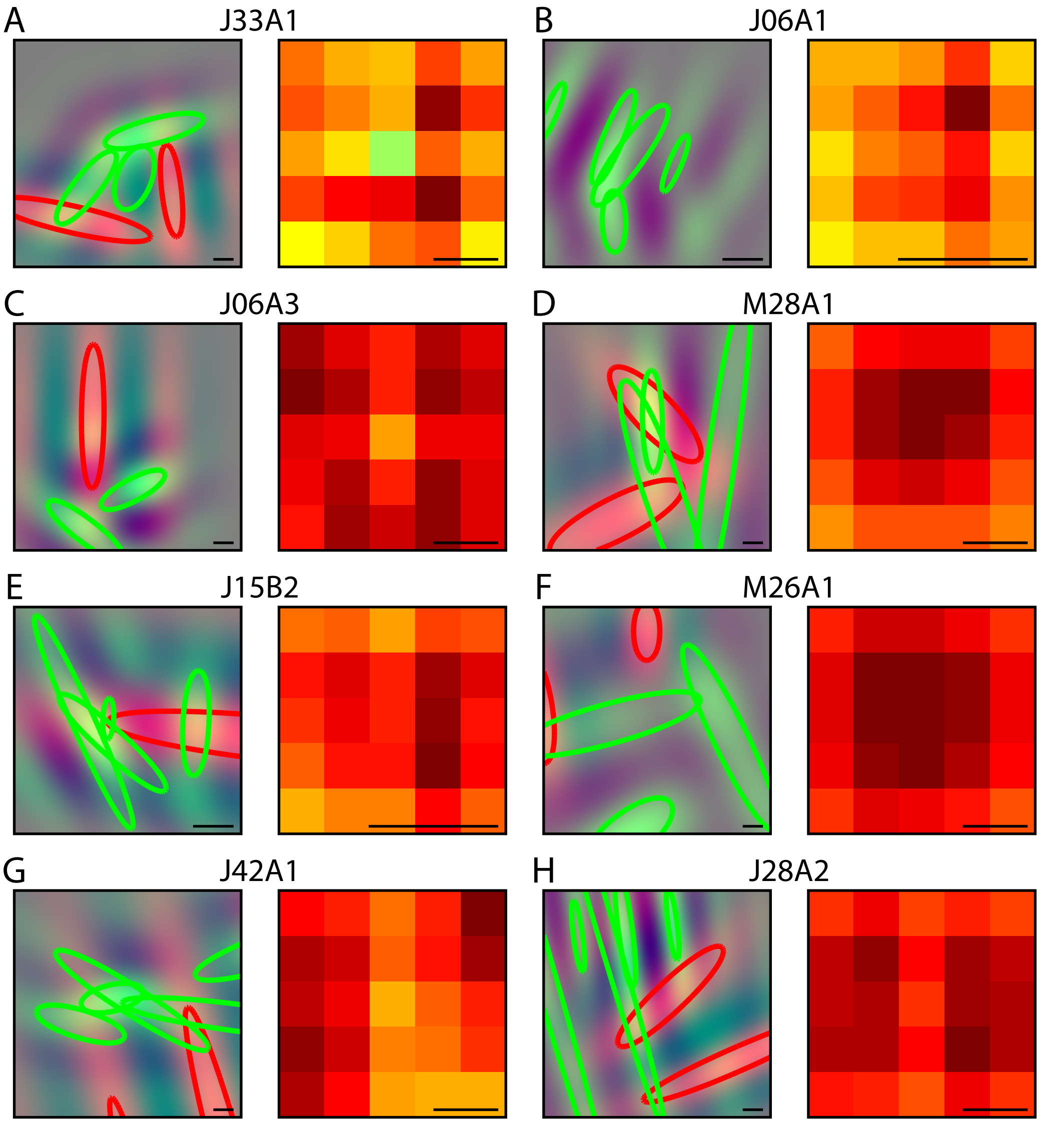

Figure 5.16: **Additional V4 contours.** Additional examples of Gabor features selectivity (left) and the corresponding spatial weighting (right). With the exception of J28A2 (H), these examples show two or more excitatory Gabors combining to form a curve. The inhibitory Gabors are oriented orthogonal to the excitatory Gabors with the exception of J06A1 (B) which did not have any inhibitory Gabor features. The spatial weightings range from relatively localized (J33A1 (A) and J06A1 (B)) to approximately uniform (J06A3 (C) and J28A2 (H)). The black bars are 1° wide.

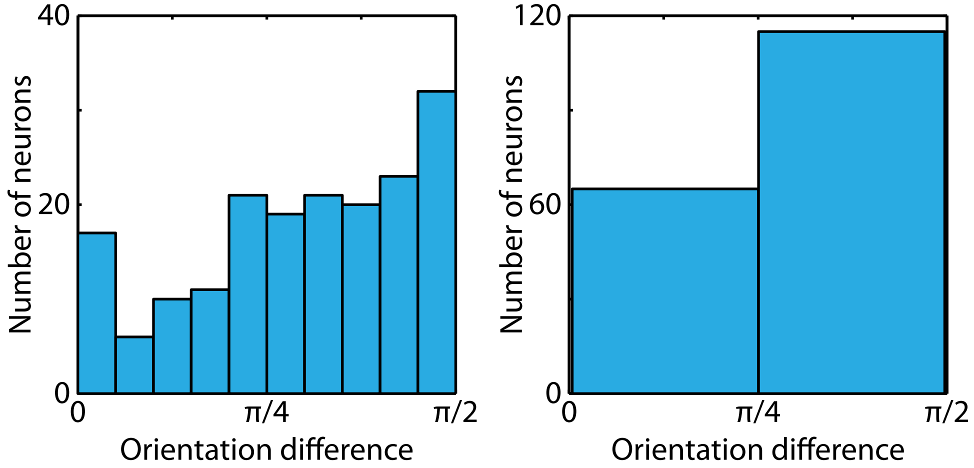

Figure 5.17: **Relative orientations of excitatory and inhibitory features.** Distribution of the difference in orientation for all pairs of excitatory and inhibitory Gabor pairs for V4 neurons. The histogram on the right uses 2 bins to show the bias for being closer to orthogonal than parallel. The histogram on the left uses 10 bins to show the distribution in more detail.

excitatory Gabors with similar orientation but located offset in space along the axis perpendicular to the curve. The spatial weighting is localized in space and falls off as the position moves away from the peak. J15B2 also has two excitatory Gabor pairs forming a curve along with an offset parallel excitatory Gabor pair. The inhibitory Gabor pair is orthogonal to the third excitatotry Gabor pair, which it overlaps. The spatial weighting falls off more quickly along the direction perpendicular to the excitatory Gabors than along the parallel direction. M26A1 has two excitatory Gabors forming a corner. The third Gabor has the same orientation as one of the other excitatory Gabors and is located parallel to it. Two inhbitory Gabor pairs are othorgonal to the parallel excitatory Gabors and are located on the other side of the curve. The spatial weighting is similar to M28A1, but it falls off more slowly as it moves from the peak. J42A1 has four excitatory Gabor pairs that form a line that crosses the frame. The inhibitory Gabor is orthogonal to this line. The spatial weighting has two peaks on opposite sides of the grid. J28A2 has two parallel excitatory Gabor pairs with two inhibitory pairs that are oriented orthogonal to the excitatory pairs. The spatial weighitng is relatively flat.

To test whether inhibitory features were preferentially orthogonal to the excitatory features, we took all Gabor features that met our criteria for being in the relevant subspace of the neuron and calculated the difference in orientation for all pairs of excitatory and inhibitory neurons. Fig. 5.17 shows the distribution of these differences. The left histogram shows the details of the distribution while the right histogram has only two bins in order elucidate the difference in frequency of being closer to parallel and being closer to orthogonal. The excitatory and inhibitory Gabor features were more likely to be orthogonal to each other than parallel.

5.5 Discussion

In this chapter, we introduced a novel method for analyzing neural responses. The goal of this method was to exploit the existence of invariances in the underlying computation to reduce the complexity of our analysis. It also uses quadratic features to find an arbitrary number of features. We showed that the algorithm is capable of reducing to a non-invariant model if the computation is localized It is also capable of able to recover invariant models in the presence of spatiotemporal correlations.

We applied the algorithm to V4 data and found that the neurons were selective for Gabor features, with the inhibition tending to be orthogonal to the excitatory features. We found examples of the Gabor features combining to form curves as well as an example of rotation invariance.

Appendix A

Derivation of information gradient

This appendix contains the full derivation of the gradient of the information per spike. The mutual information between the spikes and the projections of the stimulus onto the dimensions V is given by Eq. (2.5)

$$I_V = \int d\mathbf{x} P_V(\mathbf{x}|\text{spike}) \log_2 \left(\frac{P_V(\mathbf{x}|\text{spike})}{P_V(\mathbf{x})} \right) \tag{A.1}$$

where $P_V(\mathbf{x})$ is defined as

$$P_V(\mathbf{x}) = \int d\mathbf{s} P(\mathbf{s}) \prod_{i=1}^{K} \delta(x_i - \mathbf{v}_i \cdot \mathbf{s}). \tag{A.2}$$

$P_V(\mathbf{x}|\text{spike})$ is defined similarly by replacing $P(\mathbf{s})$ with $P(\mathbf{s}|\text{spike})$. Taking the gradient of the information gives

$$\begin{aligned} \nabla_{\mathbf{v}_i} I_V = & \int d\mathbf{x} \left(\nabla_{\mathbf{v}_i} P_V(\mathbf{x}|\text{spike})\right) \log_2 \left(\frac{P_V(\mathbf{x}|\text{spike})}{P_V(\mathbf{x})} \right) \\ & + \frac{1}{\log(2)} \int d\mathbf{x} P_V(\mathbf{x}|\text{spike}) \left(\frac{\nabla_{\mathbf{v}_i} P_V(\mathbf{x}|\text{spike})}{P_V(\mathbf{x}|\text{spike})} - \frac{\nabla_{\mathbf{v}_i} P_V(\mathbf{x})}{P_V(\mathbf{x})} \right). \end{aligned} \tag{A.3}$$

The term $P_V(\mathbf{x})$ is given by

$$\nabla_{\mathbf{v}_i} P_V(\mathbf{x}) = \int d\mathbf{s} P(\mathbf{s}) \nabla_{\mathbf{v}_i} \prod_{k=1}^{K} \delta(x_k - \mathbf{v}_i \cdot \mathbf{s}). \tag{A.4}$$

The equation for the derivative of a delta function is

$$\delta'(f(x)) = -f'(0). \tag{A.5}$$

From Eq. (A.5), it follows that

$$\nabla_{\mathbf{v}_i}\delta(x_i - \mathbf{v}_i \cdot \mathbf{s}) = \mathbf{s} \tag{A.6}$$

and

$$\frac{d}{dx_i}\delta(x_i - \mathbf{v}_i \cdot \mathbf{s}) = -1. \tag{A.7}$$

Combining these equations, we find

$$\nabla_{\mathbf{v}_i}\delta(x_i - \mathbf{v}_i \cdot \mathbf{s}) = \mathbf{s} = -\mathbf{s}\frac{d}{dx_i}\delta(x_i - \mathbf{v}_i \cdot \mathbf{s}), \tag{A.8}$$

and therefore

$$\nabla_{\mathbf{v}_i}P_V(\mathbf{x}) = -\frac{d}{dx_i}\int d\mathbf{s}P(\mathbf{s})\mathbf{s}\prod_{k=1}^{K}\delta(x_k - \mathbf{v}_k \cdot \mathbf{s}). \tag{A.9}$$

Given the definition

$$\langle\mathbf{s}|\mathbf{x}\rangle_V = \frac{\int d\mathbf{s}P(\mathbf{s})\mathbf{s}\prod_{k=1}^{K}\delta(x_k - \mathbf{v}_k \cdot \mathbf{s})}{P_V(\mathbf{x})}, \tag{A.10}$$

it follows that

$$\nabla_{\mathbf{v}_i}P_V(\mathbf{x}) = -\frac{d}{dx_i}\left(P_V(\mathbf{x})\langle\mathbf{s}|\mathbf{x}\rangle_V\right). \tag{A.11}$$

Plugging this into Eq. (A.3) gives

$$\begin{aligned}\nabla_{\mathbf{v}_i}I_V = &-\int d\mathbf{x}\frac{d}{dx_i}\left(P_V(\mathbf{x}|\text{spike})\langle\mathbf{s}|\mathbf{x},\text{spike}\rangle\right)\log_2\left(\frac{P_V(\mathbf{x}|\text{spike})}{P_V(\mathbf{x})}\right)\\ &-\frac{1}{\log(2)}\int d\mathbf{x}\frac{d}{dx_i}\left(P_V(\mathbf{x}|\text{spike})\langle\mathbf{s}|\mathbf{x},\text{spike}\rangle_V\right)\\ &+\frac{1}{\log(2)}\int d\mathbf{x}\frac{P_V(\mathbf{x}|\text{spike})}{P_V(\mathbf{x})}\frac{d}{dx_i}\left(P_V(\mathbf{x})\langle\mathbf{s}|\mathbf{x}\rangle_V\right).\end{aligned} \tag{A.12}$$

Using integration by parts and the fundamental theorem of calculus gives

$$\begin{aligned}\nabla_{\mathbf{v}_i} I_V = &- \int d\mathbf{x}_{k\neq i} P_V(\mathbf{x}|\text{spike}) \left\langle \mathbf{s}|\mathbf{x},\text{spike}\right\rangle_V \log_2\left(\frac{P_V(\mathbf{x}|\text{spike})}{P_V(\mathbf{x})}\right)\Bigg|_{x_i=-\infty}^{\infty} \\ &+ \int d\mathbf{x} P(\mathbf{x}|\text{spike}) \left\langle \mathbf{s}|\mathbf{x},\text{spike}\right\rangle_V \frac{d}{dx_i}\log_2\left(\frac{P_V(\mathbf{x}|\text{spike})}{P_V(\mathbf{x})}\right) \\ &- \frac{1}{\log(2)}\int d\mathbf{x}_{k\neq i} P_V(\mathbf{x}|\text{spike}) \left\langle \mathbf{s}|\mathbf{x},\text{spike}\right\rangle_V \Bigg|_{x_i=-\infty}^{\infty} \\ &+ \frac{1}{\log(2)}\int d\mathbf{x}_{k\neq i} P_V(\mathbf{x}|\text{spike}) \left\langle \mathbf{s}|\mathbf{x}\right\rangle_V \Bigg|_{x_i=-\infty}^{\infty} \\ &- \frac{1}{\log(2)}\int d\mathbf{x} P_V(\mathbf{x}|\text{spike}) \left\langle \mathbf{s}|\mathbf{x}\right\rangle_V \frac{P_V(\mathbf{x})}{P_V(\mathbf{x}|\text{spike})}\frac{d}{dx_i}\left(\frac{P_V(\mathbf{x}|\text{spike})}{P_V(\mathbf{x})}\right).\end{aligned} \tag{A.13}$$

As probability distributions, $P_V(\mathbf{x})$ and $P_V(\mathbf{x}|\text{spike})$ go to 0 at $\pm\infty$. Using this along with $x\frac{d}{dx}x = \frac{d}{dx}\log(x)$, the gradient becomes

$$\begin{aligned}\nabla_{\mathbf{v}_i} I_V = &\int d\mathbf{x} P(\mathbf{x}|\text{spike}) \left\langle \mathbf{s}|\mathbf{x},\text{spike}\right\rangle_V \frac{d}{dx_i}\log_2\left(\frac{P_V(\mathbf{x}|\text{spike})}{P_V(\mathbf{x})}\right) \\ &- \int d\mathbf{x} P_V(\mathbf{x}|\text{spike}) \left\langle \mathbf{s}|\mathbf{x}\right\rangle_V \frac{d}{dx_i}\log_2\left(\frac{P_V(\mathbf{x}|\text{spike})}{P_V(\mathbf{x})}\right),\end{aligned} \tag{A.14}$$

which simplifies to

$$\nabla_{\mathbf{v}_i} I_V = \int d\mathbf{x} P_V(\mathbf{x}|\text{spike})(\left\langle \mathbf{s}|\mathbf{x},\text{spike}\right\rangle_V - \left\langle \mathbf{s}|\mathbf{x}\right\rangle_V)\frac{d}{dx_i}\log_2\left(\frac{P_V(\mathbf{x}|\text{spike})}{P_V(\mathbf{x})}\right). \tag{A.15}$$

Appendix B

Subspace overlap

Given a set of model dimensions used to generate responses and a set of dimensions reconstructed from the stimulus and responses, the question arises of how well the algorithm did at reproducing the original model. A Bayesian nonlinearity depends only on the relevant subspace itself rather than the features actually used in the generation of responses. Any set of vectors that spans the same subspace provides an equivalent description of the system but with a different system of coordinates. We can easily convert one description into another using a linear transformation. Therefore, we will want a metric that compares the subspaces spanned by sets of vectors rather than the vectors themselves. This requires the metric to be invariant to non-degenerate linear transformation of the vectors.

In this dissertation, we use linear subspace projection to compare the reconstructed dimensions with the dimensions used to generate the responses from the stimulus. Given a set of K vectors $E = \{\mathbf{e}_i\}$ used to generate responses and a set of K reconstructed dimensions $V = \{\mathbf{v}_i\}$, the projection matrix P is defined such that

$$P_{i,j} = \mathbf{e}_i \cdot \mathbf{v}_j. \tag{B.1}$$

P is also the Jacobian matrix of the transformation from E to V, and $\det(P)$ is the change of volume associated with the transformation. The Gram matrices are defined as

$$G_{E,i,j} = \mathbf{e}_i \cdot \mathbf{e}_j \tag{B.2}$$

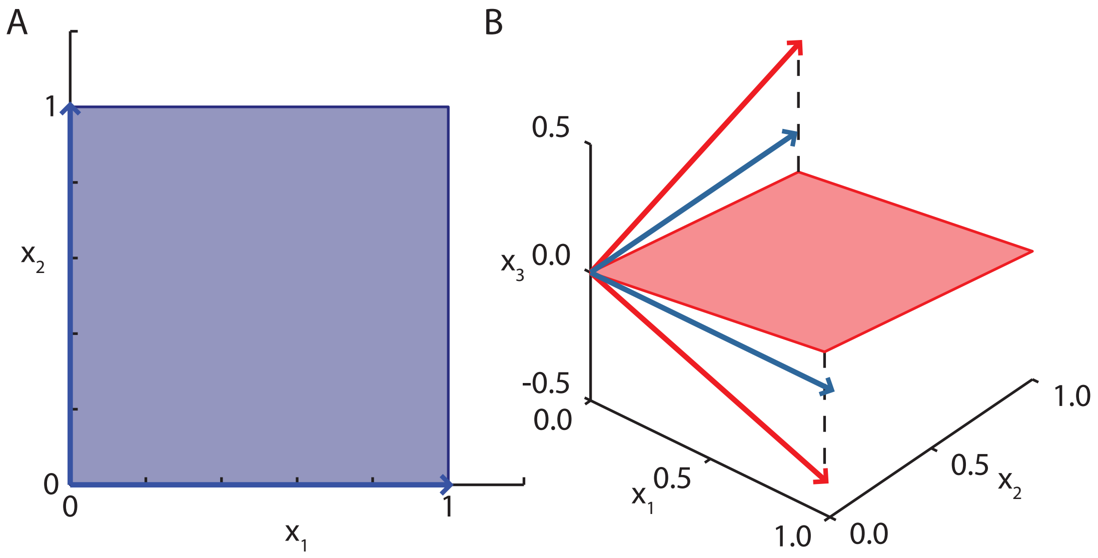

Figure B.1: **Visual demonstration of subspace overlap.** (**A**) Two orthonormal vectors define a unit square with area 1.0. Any pair of orthonormal vectors from this subspace will form a similar square and will have an overlap of 1.0. (**B**) A set of two orthonormal vectors (red) that are not in the subspace defined by the blue vectors. When projected down to the $x_1 - x_2$ plane, the vectors form a parallelogram with an area of 0.71. Taking the square root to convert this area to a linear measure gives an overlap of 0.84.

with G_V defined similarly. In this case, $\det(G)$ is the square of the volume of the parallelotope defined by the associated vectors.

The linear subspace projection is defined as

$$O = \frac{|\det(P)|^{1/K}}{|\det(G_V)|^{1/2K} |\det(G_E)|^{1/2K}}. \tag{B.3}$$

The numerator is the volume of V projected into E (or the reverse) while the denominator contains the original volumes of E and V, which normalizes the volume if one or both of E and V are not orthonormal. Taking K^{th} root converts the volume into a linear measure. This prevents the projection from becoming increasingly small as the number of dimensions increases, which aids the comparison of results with different numbers of dimensions. The resulting metric ranges from 1 when the subspaces are identical and 0 when the rank of P is less than K.

Fig. B.1 provides a visual demonstration of the geometric intuition behind the subspace overlap. Two orthonormal vectors define a unit square with an area of 1.0. Rotating the vectors or reflecting a vector across the other will not change the area of this square. To check whether a set of orthonormal vectors (red) is from the same subspace of another set (blue), we first project the first set of vectors into the second set. If the vectors are not from the same subspace, part of the projected vectors will be lost (dashed line) and vectors will define a parallelogram with an area less than 1.0.

B.1 Invariance to linear transformation

We begin with two sets of vectors: E and V. We can create two new sets of vectors E' and V' by applying the linear transformations L_E and L_V, respectively:

$$\begin{aligned} E' &= L_E E \\ V' &= L_V V. \end{aligned} \tag{B.4}$$

The linear subspace projection of these two subspaces is

$$O' = \frac{|\det(P')|^{1/K}}{|\det(G_{V'})|^{1/2K}|\det(G_{E'})|^{1/2K}}. \tag{B.5}$$

Noting the definitions of the new projection and Gram matrices

$$\begin{aligned} P' &= L_E^T P L_V \\ G_{V'} &= L_V^T G_V L_V \\ G_{E'} &= L_E^T G_E L_E, \end{aligned} \tag{B.6}$$

we can take advantage of the properties of the determinant that $\det(AB) = \det(A)\det(B)$ if A and B are square matrices and $\det(A^T) = \det(A)$ to rewrite Eq. B.5 as

$$O' = \frac{|\det(L_E)|^{1/K}|\det(P)|^{1/K}|\det(L_V)|^{1/K}}{|\det(L_V)|^{2/2K}|\det(G_V)|^{1/2K}|\det(L_E)|^{2/2K}|\det(G_E)|^{1/2K}}. \tag{B.7}$$

As long as L_E and L_V are not degenerate, their determinants are non-zero and the factors in the numerator and denominator cancel out, which leaves us with the original value of the linear subspace projection given in Eq. B.3.

B.2 Extension to differing dimensions

This metric can also be extended to the situation where the dimensionality of E and V differ. We begin by performing the substitution $|\det(P)| = |\det(P)^2|^{1/2} = |\det(P^T P)|^{1/2}$. Since P no longer needs to be a square matrix, V and E can have different numbers of dimensions, which we call K_V and K_E. In order for the determinant not to be 0, K_E must be greater than or equal to K_V. Furthermore, because we can no longer pull L_E out of the determinant, E must be orthonormal. The equation for the generalized linear subspace projection is

$$O = \left(\frac{|\det(P^T P)|}{|\det(G_V)|} \right)^{1/2K_V} . \tag{B.8}$$

The advantage of this form is that we can evaluate models with fewer or greater dimensions than the model used to generate responses compared to the maximum possible performance. The cost of this is that we no longer are able to perform the trick used in the previous section to make the linear subspace projection invariant to linear transformations of the larger subspace E.

B.3 Comparison to principal angles

The motivation for measure came from the limitations of principle angles used by Rapela et al. (2010). Principal angles compares sets of vectors using the angles between the vectors. To calculate the principal angles, one begins by calculating the angles between all of the vectors of one set and all the vectors of the other set. If the angle is greater than $\pi/2$, it is replaced by $\pi - \theta$, which is equivalent to replacing one vector with its negative. This ensures that the sign of the vectors does not matter. One takes the smallest angle as the first principal angle, removes this pair from the sets of vectors, and repeats this process until all of the vectors have been paired. The values of the principal angles range from 0 when the pair of vectors are identical to $\pi/2$ when the vectors are orthogonal. Each succeeding principal angle is greater than those that preceded it.

The disadvantage of principal angles is that it compares the vectors that define subspaces rather than the subspaces themselves. The same subspace can be described by an infinite number of sets of vectors, each of which define a different but equivalent system of coordinates. This is especially problematic when there is not even a system of coordinates that gives a relatively simple description of the response.

Consider a simple model of a complex cell where relevant dimensions are two Gabor wavelets with identical parameters except that they have orthogonal spatial phases. The response is determined by the summed square of the projections

$$f(\mathbf{x}) = x_1^2 + x_2^2. \tag{B.9}$$

In polar coordinates this becomes

$$\begin{aligned} f(\mathbf{x}) &= (r\cos(\theta+\phi_0))^2 + (r\sin(\theta+\phi_0))^2 \\ &= r^2(\cos(\theta+\phi_0)^2 + \sin(\theta+\phi_0)^2) \qquad \text{(B.10)} \\ &= r^2. \end{aligned}$$

Regardless of our choice of coordinates (represented by the angle ϕ_0 relative to the dimensions actually used to generate the response), the description is equivalently simple. However, our choice of ϕ_0 does affect the principal angles. If ϕ_0 is 0, The principal angles will by 0 and 0. As ϕ_0 increases, the principal angles will increase with it until reaching a maximum of $\pi/4$ and $\pi/4$ when $\phi_0 = \pi/4$. The subspace overlap will be 1 regardless of our choice of ϕ_0 because rotation is a non-degenerate linear transformation.

Appendix C

Averaging dimensions using PCA

Just as we wanted a metric that compared subspaces directly rather than the particular sets of dimensions used to describe the subspaces in Appendix B, we also need a method that averages subspaces directly rather than the sets of dimensions used to describe them.

The problem with averaging the individual dimensions is illustrated by the following pathological but plausible example. Consider the reconstructed dimensions using two different jackknifes of the data as the cross-validation set. Let's further assume that the dimensions recovered by each analysis are identical except that the labels of which dimension is the first and which dimension is the second are switched between jackknifes. Averaging the vectors will result in two copies of an average the first and second dimension and reduce the size of the subspace from two dimensions to one. Clever reordering of the dimensions could avoid this, but this becomes tricky when averaging across more than two jackknifes.

As an alternative, we use Principle Components Analysis (PCA) to find the average subspace. With $\mathbf{v}_{k,j}$ as the kth vector from the j jackknife, we first calculate the covariance of the dimensions:

$$C_{PCA} = \sum_{j,k} \mathbf{v}_{k,j} \mathbf{v}_{k,j}^T. \tag{C.1}$$

The averaged subspace is described by the K eigenvectors of C_{PCA} with the largest eigenvalues. Note that we do not subtract the mean of the vectors. This gives us the principal components of variation relative to the origin.

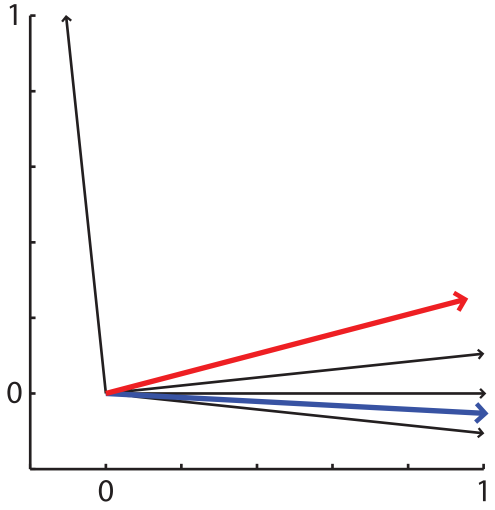

Figure C.1: **Averaging four two-dimensional vectors using PCA.** The set contains three similar vectors and one outlier (black). The red line shows the result of averaging the four vectors. The overlap with $(1, 0)$ is 0.95. The blue line shows the PCA vector of the four vectors. The overlap with $(1, 0)$ is 0.999. The PCA average is much less susceptible to outliers.

This has the advantage of reducing the effect of outliers. Fig. C.1 shows an example with four vectors (black). Three vectors are centered around $(1, 0$ while the fourth is an outlier almost perpendicular to $(1, 0)$. The red line is the dimension that results from averaging the four vectors together. The overlap with $(1, 0)$ is 0.95. The blue vector is the PCA average of the four vectors. Its overlap with $(1, 0)$ is 0.999. In this way, PCA averaging can act as a principled way to handle outliers.

To evaluate how similar the subspaces are to each other, we can look at how much of C_{PCA}'s energy is captured by the selected eigenvectors. If $\{\lambda_k\}$ are eigenvalues of the selected eigenvectors, the fraction of the energy captured is

$$\frac{\sum_{k=1}^{K} \lambda_k}{\text{Tr } C_{PCA}}, \tag{C.2}$$

where Tr is the trace. It ranges from 1 when the subspaces of the different jackknifes are contained in the averaged subspace to $1/N_{jack}$ when there is no correspondence across jackknifes.

C.1 Recovering MID basis

While PCA is useful for finding an average subspace, it can erase some relevant information. In MID, the numbering of the dimensions comes from the order in which the algorithms finds them, which is related to how much information the dimensions explain. The PCA average dimensions are ordered based on how well represented each vector is in the subspaces to be averaged.

To find the maximally informative basis of the subspace, we can resort to a brute force approach. Unlike with MID which must search a D-dimensional space, the reduced subspace is only K-dimensional. By generating a large number of random directions uniformly distributed on the unit K-sphere, we can find the MID basis by first calculating the information along the random dimensions, selecting the most informative dimension, and repeating this procedure by calculating the information between the selected dimensions and each additional random dimension.

Appendix D

Experimental details

This section contains detailed descriptions of physiological experiments analyzed in this dissertation. I have put these details in a separate chapter in the appendix to allow the analysis of the experiments to only need to discuss the details relevant to the conclusions and to avoid repetition of details of experiments analyzed with multiple methods.

D.1 Recordings of V1 neurons

The recordings of the responses of neurons in the primary visual cortex (V1) were collected as part of a previous study (Sharpee et al., 2006). The recordings are from four anesthetized cats stimulated with grating, white noise, natural movie, and repeated neural movie stimuli. These recordings were analyzed using MID, SMID, and ePPR in Chapter 2 and IMID in Chapter 5.

All recordings were conducted under protocols approved by the University of California, San Francisco Committee on Animal Research.

The response of the cells to grating stimuli determined whether we classified the cells as simple or complex (Skottun et al., 1991). The criterion for whether a cell is simple or complex is whether the ratio of the amplitude of the response at the grating frequency ($F1$) to the average response ($F0$) was greater than or less than 1, respectively. The intuition behind this metric is that simple, which Hubel and Wiesel defined as cells performing linear computations (Hubel and Wiesel,

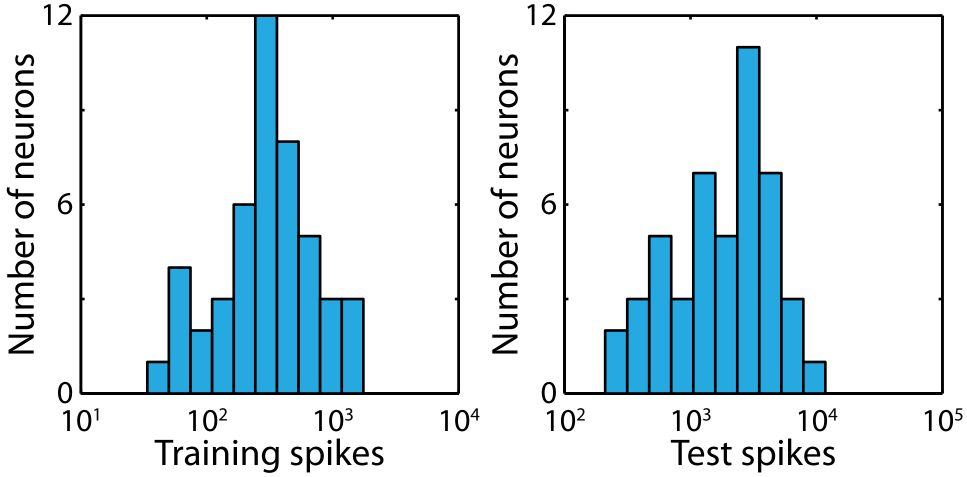

Figure D.1: **Number of spikes for V1 neurons with repeated stimuli.** Set of 47 neurons from V1 that includes 32 simple and 15 complex cells. For the training and cross-validation data (left), the number of spikes ranged from 337 to 17514 with a median of 3210. For the repeated test data (right), the number of spikes ranged from 210 to 11766 with a median of 2203.

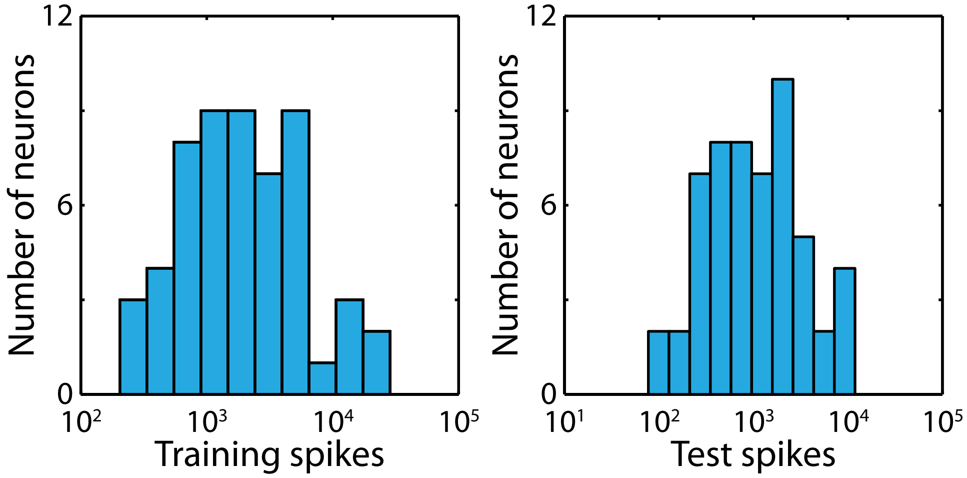

Figure D.2: **Number of spikes for complex V1 neurons.** Set of 53 complex cells, including the 15 neurons from Fig. D.1 analyzed by IMID. For the training and cross-validation data (left), the number of spikes ranged from 204 to 28574 with a median of 1953. For the repeated test data (right), the number of spikes ranged from 77 to 11885 with a median of 1015.

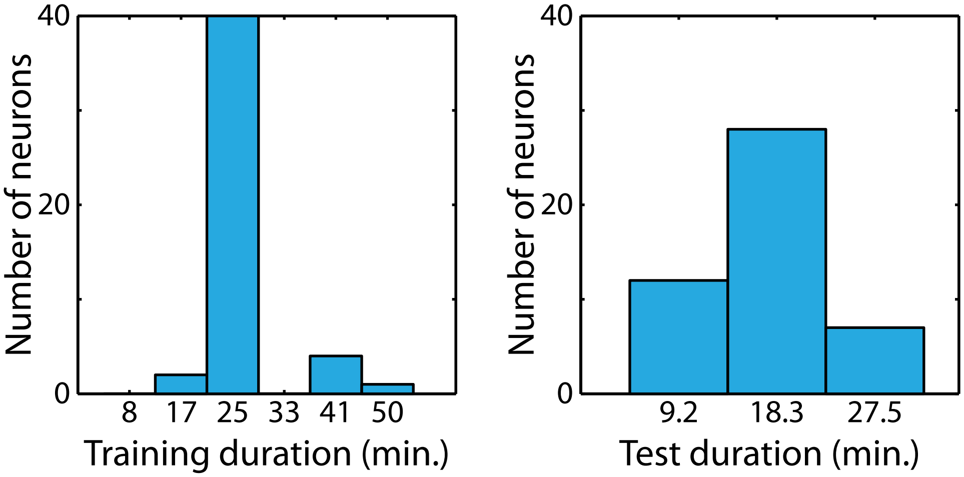

Figure D.3: **Distribution of stimulus durations for V1 neurons.** The range of stimulus durations of the training and cross-validation set (left) was 16.5 to 49.6 minutes with a median of 24.8. This is 2 to 6 presentations movies from the set of three natural movies. The durations for the test set (right) ranged from 9.2 to 27.5 minutes with a median of 18.3 minutes.

1962), will respond with the phase of the grating and will therefore have a large $F1$ component. All other cells are complex.

Spike trains were recorded using tetrode electrodes and sorted offline. Some recordings included multiple neurons, but we did not use this for any of the analysis presented here.

We used two sets of neurons for our analyses. The first was a set of 47 neurons which included 32 simple and 15 complex cells. These neurons were chosen for some reason along as well as having recorded responses to repeated stimuli which we could use to estimate the average information transmitted per spike for each neuron. The distribution of spikes and stimulus durations are shown in Fig. D.1 and Fig. D.3. The second set included all 53 complex cells from the experiment that had recordings using a repeated stimulus. The distribution of spikes and stimulus durations are shown in Fig. D.2 and Fig. D.4.

The natural movie stimuli were three recordings of a walk in a wooded area using a hand-held video camera. Each movie was 16, 384 frames with a frame rate

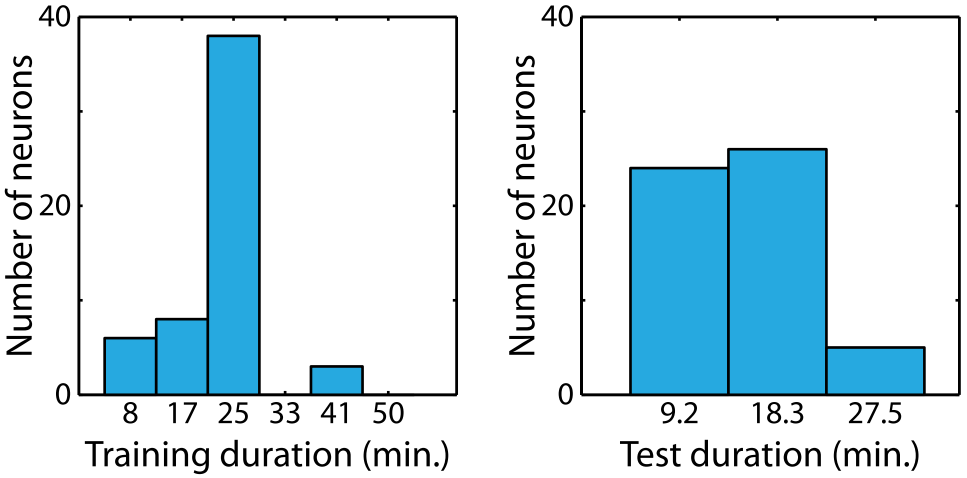

Figure D.4: **Distribution of stimulus durations of complex cell dataset.** The stimulus durations of the training and cross-validation data of the set of complex cells (left) ranged from 8.3 to 41.4 minutes with a median of 24.8 minutes. For the test data (right), the durations ranged from 9.2 minutes to 27.5 minutes.

of 33 Hz. The duration of each move was slightly over 8 minutes and 16 seconds. Some cells do not have recordings for all three movies because their fixation was lost. Other cells had recordings for more than one presentation of one or more of the movies because they were held for long enough.

The repeated movie stimulus consisted of 55 repetitions of 330 frames (10 seconds) from one of the natural movies. The purpose of this stimulus is to allow us to estimate the empirical information per spikes, as described in Appendix E. As with the natural movies, some cells have recordings for more than one presentation of this stimulus.

We did not use the white noise stimulus for any of the analysis presented here.

All stimuli were presented at a 128 × 128 pixel resolutions with an angular resolution of 0.12° per pixel.

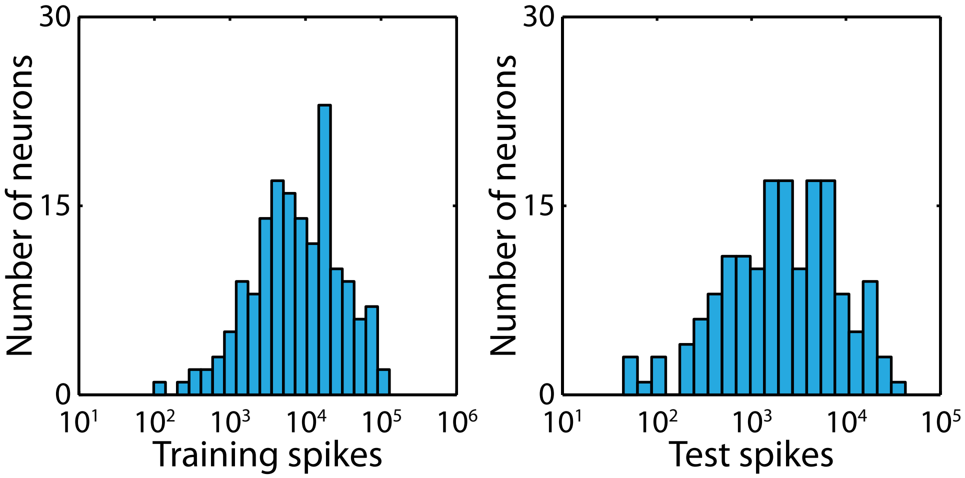

Figure D.5: **Number of spikes for V4 data.** Total number of spikes recorded from 161 V4 neurons in response to an unrepeated training stimulus (left) and repeated test stimulus (right). The median number of spikes was 7491 for the training data and 2318 for the test data.

D.2 Recordings of macaque V4 neurons

The recordings of neurons from macaque visual area V4 stimulated by natural movies were collected as part of a previous study (Sharpee et al., 2013). The data was collected from two awake, fixating monkeys that were rewarded with juice. During the experiment, the monkey fixated on a target while a 14° × 14° movie clip played over the previously estimated center of the neuron's receptive field. Neural responses were recorded extracellularly using a tungsten microelectrode. On trials where the monkey broke fixation, responses recorded after that point were not used for our analysis.

The movie stimuli consisted of 574 111-frame clips extracted from the movies from Sec. D.1. The frames were presented at 30 Hz, so duration of each clip was 3.7 s.

The movie clips were divided into two groups: One group was played a small number of times in order to maximize the diversity of stimuli and was used to train and cross-validate our models. The second group consisted of three movie clips

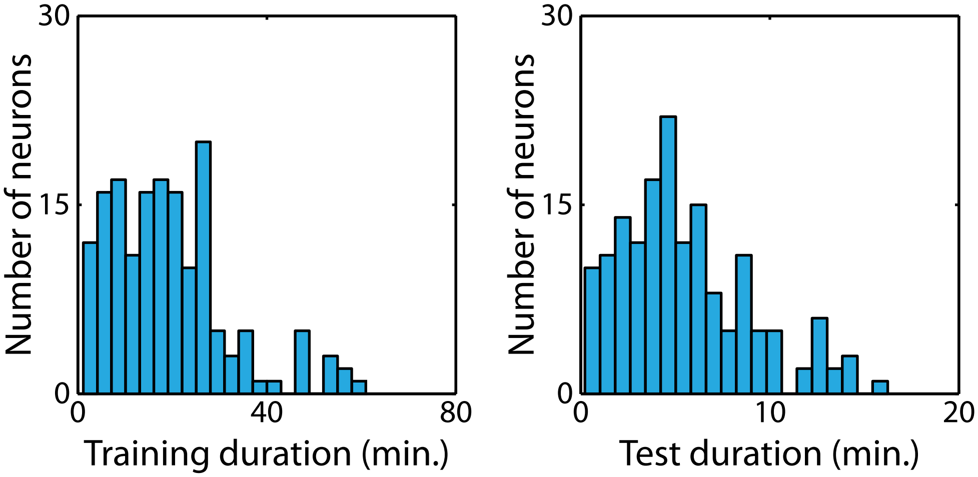

Figure D.6: **Duration of stimuli for V4 data.** Total duration of unrepeated data used for training and cross-validation (left) and repeated data used for testing (right) for 161 V4 neurons used for our analysis in Chap. 5. The median duration was 17.5 min. for training and 4.6 min. for testing.

that were repeated many times to allow us to measure the response variability and was used to test the performance of our models. Fig. D.2 shows the distribution of the amount of data available. The unrepeated data used for training and cross-validation ranged from 1.1 to 60.9 min. with a median of 17.5 min. The repeated data used for testing ranged from 0.2 to 16.2 min. with a median of 4.6 min.

Appendix E

Information extrapolation

In the presence of finite data, the mutual information (Eq. 2.3) will have a positive bias. Because a neuron's response is stochastic and we can only measure a finite number of samples, there will noise in the number of observations in each bin of the histogram probability distribution. If $n(\vec{x})$ is the expected number of observations in the bin at $\vec{x}$, the noise due to counting only a finite number of samples is $\sqrt{n(\vec{x})}$, or $1/\sqrt{n(\vec{x})}$ as a fraction of the observed count. Increasing the number of observations will reduce but not eliminate the relative size of this error.

These errors will introduce errors in the calculation of the information. If the errors shift the observed firing across the stimuli toward the mean firing rate, this will reduce the estimate of the information explained. On the other hand if the errors shift the stimulus-dependent firing rate away from the mean, the estimate of the information explained will increase. This latter case is especially worrisome for the case where the reduced dimensions do not carry any true information about the response. In this case, the expected firing rate will be the mean firing rate for all stimuli, and any errors in the estimates of stimulus or response probabilities will necessarily shift the estimated response away from the mean. This results in non-informative models describing a non-zero amount of information.

To account for this error we can take advantage of the dependence of the size of the counting errors on the number of observations. Strong et al. (1998) found that the error in the information measurement is proportional to $1/N$, where N is the number of stimuli. By calculating the information for subsets of the stimulus of different sizes, we can fit the information and the inverse of the subset size using linear regression to extrapolate to infinite data.

Appendix F

Eigenvector significance

Quadratic methods such as STC (Sec. 1.3), QMID (Chap. 4), QMNE (Sec. 4.1), and ILS (Chap. 5) determine the relevant stimulus subspace by selecting the eigenvectors whose eigenvalues are significantly larger than one would expect from noise. An unsettled question is how to determine what is significant.

In our analysis, we followed the lead of Schwartz et al. (2006) by performing a nested analysis. First, we create a set of random J matrices by shuffling the diagonal and off-diagonal elements. We chose to shuffle rather than select values from a Gaussian distribution with the observed mean and variance in order to preserve any non-Gaussian structure in the distribution of elements. For each shuffled matrix, we recorded the largest positive and negative eigenvalue. We estimated the probability that the largest eigenvalue would appear by chance on how many shuffled matrices had larger eigenvalues of the corresponding sign. If the probability was less than 0.05, we counted the eigenvector as significant. We then removed the eigenvector from J and repeated the process with the reduced matrix. When considering whether the next eigenvalue was significant, we combined its probability with the previous probability using an OR function:

$$p_{i+1} = 1 - (1 - p)\,(1 - p_i)\,, \tag{F.1}$$

where p_i is the probability that the largest i eigenvectors are significant and p is the probability that an eigenvalue as large as the current one would come from a shuffled matrix.

Appendix G

Finding Gabor wavelet representations

Quadratic methods — including as STC (Sec. 1.3), QMID (Chap. 4), QMNE (Sec. 4.1), and ILS (Chap. 5) — have the ability to identify stimulus subspaces of arbitrary and unspecified dimensionality limited only by the statistical requirements of the method and the amount of data available. One major limitation of these methods is that the relevant subspace is described by a set of significant eigenvectors that are orthogonal as a result of the symmetric nature covariance matrix or quadratic filter J. While any basis that spans the same space is equivalent, some bases may be easier to interpret.

Fig. G shows an example of this problem. The model involves two excitatory quadrature pairs of Gabor wavelets offset in space and orientation that combine to form a curve along with an inhibitory pair orthogonal to the center of the curve. When these features are combined into a quadratic filter J, the eigenvectors of J span the same space of the stimulus, but the excitatory pairs have been replaced by pairs of their sums and differences. The underlying Gabor structure is obscured.

To recover the original structure, we can search for a set of Gabors that when combined into a J matrix match the observed J matrix. This is justified in this case because the model was created using a set of Gabors, but this is also plausible for neurons in the visual system because Gabor wavelets are a rough description of simple cells in V1 while quadrature pairs of Gabors describe complex

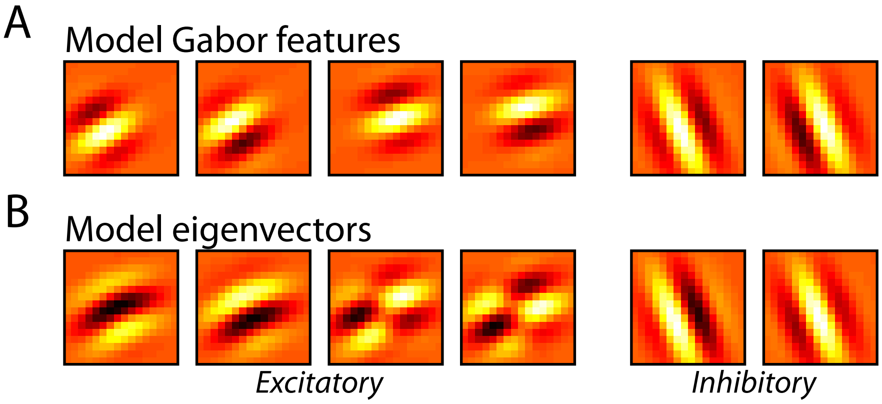

Figure G.1: **Orthogonal representation of Gabor wavelets.** (**A**) Three quadrature pairs of Gabor wavelets of the model cell used in Chap. 5. (**B**) Eigenvectors of the matrix formed by the outer products of the Gabors. While the eigenvectors span the same space as the features in **A**, the underlying structure of pairs of Gabors is obscured.

cells.

To find a Gabor basis, we want to minimize

$$\sum_{i=1}^{D}\sum_{j=1}^{D}\left(J_{i,j}-JG_{i,j}\right)^2 \tag{G.1}$$

where JG is the outer products of Gabor wavelets defined by

$$G(\vec{x}|A,\vec{x}_0,\theta,\sigma,\gamma,\lambda,\phi)=Ae^{-\frac{(x_1'^2+\gamma^2x_2'^2)}{2\sigma^2}}\cos\left(\frac{2\pi}{\lambda}x_1'+\phi\right) \tag{G.2}$$

where

$$\begin{pmatrix} x_1' \\ x_2' \end{pmatrix}=(\vec{x}-\vec{x}_0)\begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}. \tag{G.3}$$

We considered both sets of individual Gabors and sets of quadrature pairs of Gabors. In the case of Gabor pairs, all parameters are the same except for the phase, which is defined by $\phi_2=\phi_1+\frac{\pi}{2}$. We broke J into J_+ and J_-, which are composed of the significant positive and negative eigenvectors and eigenvalues, and fit the Gabors to each matrix separately.

Because Eq. G.1 is non-convex, we ran the optimization multiple times

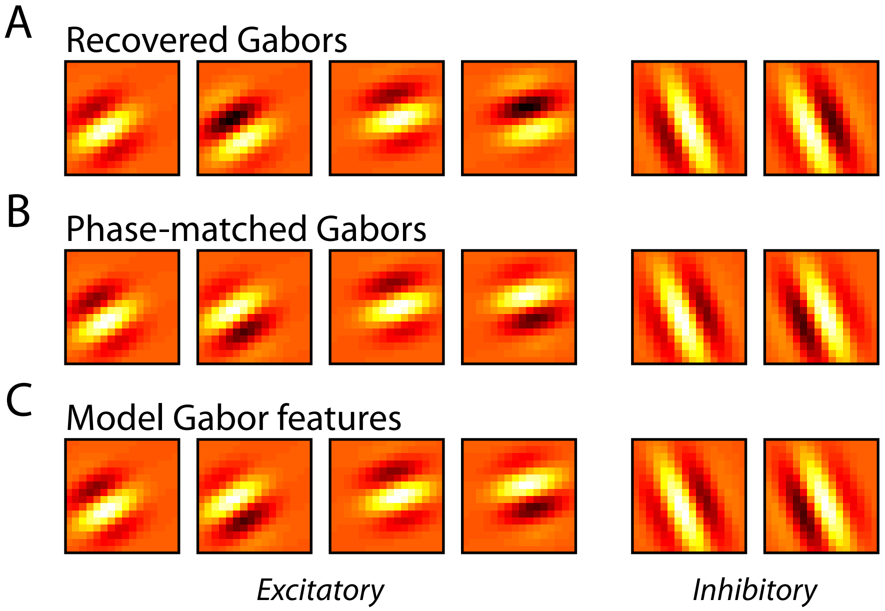

Figure G.2: **Recovering original Gabors.** (**A**) Pairs of Gabors that best match J projected onto the eigenvectors shown in Fig. GB. (**B**) Same as A except that the phases have been chosen to match those of the model. (**C**) The original Gabor features for comparison. The mean dot product between the corresponding vectors in B and C is 0.999988 ± 0.000003.

starting with the parameters roughly fit to a random linear combination of the orthogonal dimensions.

G.1 Gabor fitting example

To demonstrate the effectiveness of this technique, we applied it to the example shown in Fig. G. We fit J_+ with two pairs of Gabors and J_- with one pair. Fig. G.1A shows the best fit projected onto the orthogonal dimensions of Fig. GB. The fit captures the position, orientation, spatial frequency, and spatial extent of the model Gabor features, but the phases are different. This is because the creation of J squares and sums the features, which eliminates information about the phase from J. Because the phase is arbitrary, we can choose the phase

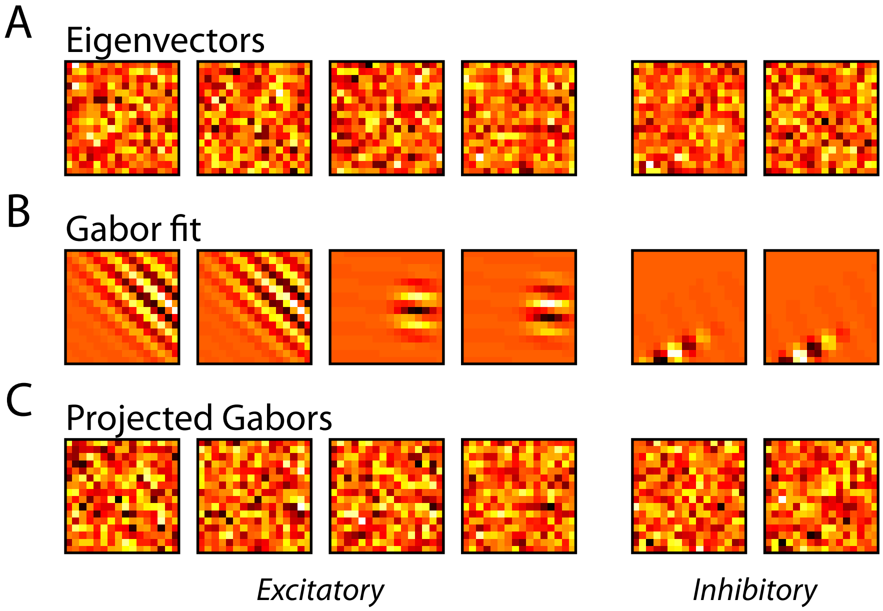

Figure G.3: **Fitting Gabor pairs to random matrix.** (**A**) The largest four excitatory and two inhibitory eigenvectors of a random symmetric matrix. (**B**) The set of Gabor pairs that best fits the J matrix. (**C**) The projection of the Gabors onto the six eigenvectors from A.

that best fits the model in order to better compare the two. Fig. G.1B and C show the phase-matched Gabor fit and the original model. They are almost identical with a dot product of 0.999988 ± 0.000003 (mean ± sem).

G.2 Fitting noise

One might wonder whether the ability of this method to find Gabor-like linear transformations regardless of whether they exist in the creation of the J matrix. This will not happen because only have a limited number of dimensions to fit a much higher dimensional space.

Fig. G.2 shows the results of trying to fit a randomly generated J matrix. We generated J by creating a matrix with normally distributed values (mean 0 and variance 1) and setting the mean of that matrix and its transpose as J. We

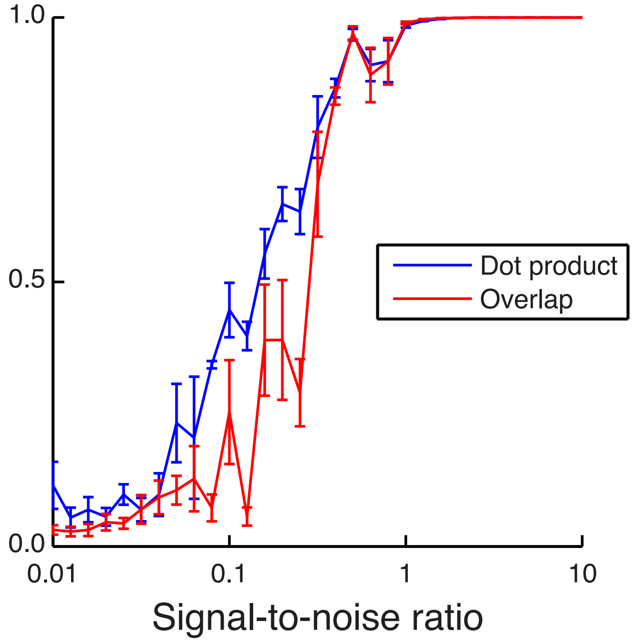

Figure G.4: **Quality of fit with noise.** Mean dot product between phase-matched Gabor fit and model (red) and overlap between Gabor fit and model (blue). The algorithm perfectly recovers the underlying Gabor pairs as long as the magnitude of the signal is at least as large as the magnitude of the noise. Performance deteriorates as the signal–to–noise ratio decreases below that level.

tried to find two excitatory pairs and one inhibitory pair like in Sec. G.1. Our algorithm finds a set of Gabor wavelets, but when they are projected onto the four largest excitatory and two largest inhibitory eigenvectors of J, there is no underlying Gabor structure.

G.3 Performance in the presence of noise

We next investigated how the algorithm performs in the presence of noise. We used the J matrix from Sec. G.1 and generated a random matrix J_R like in Sec. G.2. We then fit Gabors to to the weighted sum with the signal–to–noise ratio $||J||/||J_R||$ ranging from 0.01 to 10.

Fig. G.3 shows two measures of performance. The first is the overlap (App. B)between the Gabor fit and the model features. For the second, we followed the matching procedure from Sec. G.1 and took the mean of the normalized dot products between the model and fit features. The algorithm does very well (≥ 0.98 for both measures) as long as the signal–to–noise ratio is at least 1 and continues to capture part of the model even when the strength of the noise is greater than the signal.

Bibliography

R. Bellman. *Adaptive processes - a guided tour.* Princeton University Press, Princeton, NJ, 1961.

E. de Boer and P. Kuyper. Triggered correlation. *IEEE Trans. Biomed. Eng.*, BME-15:169–179, 1968.

R. R. de Ruyter van Steveninck and W. Bialek. Real-time performance of a movement-sensitive neuron in the blowfly visual system: coding and information transfer in short spike sequences. *Proc. R. Soc. Lond. B*, 265:259–265, 1988.

M. Eickenberg, R. J. Rowekamp, M. Kouh, and T. O. Sharpee. Characterizing responses of translation-invariant neurons to natural stimuli: Maximally informative invariant dimensions. *Neural Comp.*, 24:2384–2421, 2012.

J. D. Fitzgerald, R. J. Rowekamp, L. C. Sincich, and T. O. Sharpee. Second order dimensionality reduction using minimum and maximum mutual information mdoels. *PLoS Comput. Biol.*, 7(10):e1002249, 2011a.

J. D. Fitzgerald, L. C. Sincich, and T. O. Sharpee. Minimal models of multidimensional computations. *PLoS Comput. Biol.*, 7:e1001111, 2011b.

J. L. Gallant, J. Braun, and D. C. Van Essen. Selectivity for polar, hyperbolic, and cartesian gratings in macaque visual cortex. *Science*, 259:100–103, 1993.

J. L. Gallant, C. E. Connor, S. Rakshit, J. W. Lewis, and D. C. Van Essen. Neural responses to polar, hyperbolic, and cartesian gratings in area v4 of the macaque monkey. *J. Neurophysiol.*, 76:2718–2739, 1996.

D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J. Neurphysiol., London*, 160: 106–154, 1962.

L. Paninski. Convergence properties of three spike-triggered average techniques. *Network: Comput. Neural Syst.*, 14:437–464, 2003.

A. Pasupathy and C. E. Connor. Responses to contour features in macaque area v4. *J Neurophysiol*, 82:2490–2502, 1999.

K. Rajan and W. Bialek. Maximally informative "stimulus energies" in the analysis of neural responses to natural signals. arXiv:qbio.NC/1201.0321, 2012.

J. Rapela, J. T. Felsen, J. M. Mendel, and N. M. Grzywacz. ePPR: A new strategy for the characterization of sensory cells from input/output data. *Network: Computation in Neural Systems*, 21:35–90, 2010.

E. T. Rolls and M. J. Tovee. Sparseness of the neuronal representation of stimuli in the primate temporal visual cortex. *J. Neurophysiol.*, 73:713–726, 1995.

R. J. Rowekamp and T. O. Sharpee. Analyzing multicomponent receptive fields from neural responses to natural stimuli. *Network: Computations in Neural Systems*, 22 (1-4):45–73, 2011.

O. Schwartz, E. J. Chichilnisky, and E. Simoncelli. Characterizing neural gain control using spike-triggered covariance. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing*, volume 14, 2002.

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

T. Sharpee, N.C. Rust, and W. Bialek. Maximally informative dimensions: Analyzing neural responses to natural signals. In S.Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing*, volume 15, 2003.

T. Sharpee, N.C. Rust, and W. Bialek. Analyzing neural responses to natural signals: Maximally informatiove dimensions. *Neural Computation*, 16:223–250, 2004. See also physics/0212110, and a preliminary account in *Advances in Neural Information Processing 15* edited by S. Becker, S. Thrun, and K. Obermayer, pp. 261-268 (MIT Press, Cambridge, 2003).

T. O. Sharpee, H. Sugihara, A. V. Kurgansky, S. P. Rebrik, M. P. Stryker, and K. D. Miller. Adaptive filtering enhances information transmission in visual cortex. *Nature*, 439:936–942, 2006.

T. O. Sharpee, M. Kouh, and J. H. Reynolds. Trade-off between curvature tuning and position invariance in visual area v4. *PNAS*, 110:11618–11623, 2013.

E. P. Simoncelli and B. A. Olshausen. Natural image statistics and neural representation. *Annu. Rev. Neurosci.*, 24:1193–1216, 2001.

B.C. Skottun, R.L. De Valois, D.H. Grosof, J.A. Movshon, D.G. Albrecht, and A.B. Bonds. Classifying simple and complex cells on the basis of response modulation. *Vision Res.*, 31:1079–1086, 1991.

S. P. Strong, R. Koberle, R. R. de Ruyter van Steveninck, and W. Bialek. Entropy and information in neural spike trains. *Phys. Rev. Lett.*, 80:197–200, 1998.

J. H. van Hateren. Processing of natural time series of intensities by the visual system of the blowfly. *Vision Res*, 37:3407–3416, 1997.

B. Vintch, A. D. Zaharia, J. A. Movshon, and E. P. Simoncelli. Efficient and direct estimation of a neural subunit model for sensoriy coding. In P. Bartlett, editor, *Advances in Neural Information Processing*, volume 25, 2012.