Monte Carlo methods for localization of cones given multielectrode retinal ganglion cell recordings

K. Sadeghi, J. Gauthier, G. Field, M. Greschner, M. Agne, E.J. Chichilnisky, L. Paninski

September 23, 2012

Abstract

It has recently become possible to identify cone photoreceptors in primate retina from multi-electrode recordings of ganglion cell spiking driven by visual stimuli of sufficiently high spatial resolution. In this paper we present a statistical approach to the problem of identifying the number, locations, and color types of the cones observed in this type of experiment. We develop an adaptive Markov Chain Monte Carlo (MCMC) method that explores the space of cone configurations, using a Linear-Nonlinear-Poisson (LNP) encoding model of ganglion cell spiking output, while analytically integrating out the functional weights between cones and ganglion cells. This method provides information about our posterior certainty about the inferred cone properties, and additionally leads to improvements in both the speed and quality of the inferred cone maps, compared to earlier "greedy" computational approaches.

Introduction

The retina is among the neural systems to which we have the best experimental access. On one hand, we can present stimuli at a fine enough resolution to simultaneously and individually excite all the photoreceptors in a wide area. On the other hand, we can simultaneously record from and sort the spikes of nearly all the ganglion cells of certain types within a region spanning hundreds of ganglion cells (Chichilnisky and Kalmar, 2002; Segev et al., 2004; Frechette et al., 2005; Gauthier et al., 2009). We also have straightforward phenomenological models which can predict the spiking output of these ganglion cells types in response to random white noise stimuli with reasonable accuracy (Chichilnisky, 2001; Keat et al., 2001; Pillow et al., 2005; Pillow et al., 2008). To move forward in building more accurate models of the retina and understanding its function, we are at a stage where it is useful to identify individual cells involved in transducing the signal before it reaches the ganglion cell level.

A milestone in this direction was recently described in (Field et al., 2010), who showed that it is possible to identify individual cone photoreceptors given the spiking of multiple simultaneously recorded ganglion cells. Given sufficiently high resolution random stimuli, it was observed that the spike triggered averages (STA) of ganglion cells are made up of small islands of light sensitivity which overlap across different ganglion cells and are similar in shape, size and color to cone receptive fields. By comparing the locations of these islands of sensitivity with images of the cone layer from the same experiments, it was established that the small islands making up STAs are in fact individual cone receptive fields.

A typical multi-electrode experiment will gather spikes from hundreds of ganglion cells, whose STAs, covering different areas of the field of view, can be more or less sharp and

informative about cone locations, depending on the number of spikes that were sorted and the characteristics of each ganglion cell. Thus for any experiment, the evidence pinpointing the locations of cones will be strong in some regions, while in other regions it may be hard to distinguish the underlying signal from noise in the estimated STA.

This work describes a statistical method for inferring cone locations and types, along with measures of confidence we can have in each inferred cone. We build upon the model and methods used in (Field et al., 2010), but utilize Markov Chain Monte Carlo (MCMC) computational inference methods (Robert and Casella, 2005) instead of the simpler greedy optimization approach used in (Field et al., 2010). In addition to providing measures of posterior confidence in our inferred results, we find that the MCMC approach obtains cone maps of higher likelihood (demonstrating that the greedy approach is prone to local optima in this setting); we also note that a "lazy" implementation of the approach leads to a faster solution than the greedy method.

Experimental methods summary

The same protocols as in (Field et al., 2010) were used to collect similar data. Extracellular multi-electrode recordings were obtained from ganglion cells of isolated retinas taken from macaque monkeys (Macaca fascicularis and Macaca mulatta) used in other laboratories (Chichilnisky and Baylor, 1999). Spikes from several hundred cells were segregated offline (Litke et al., 2003). Receptive-field maps were obtained using a fine grained random colored checkerboard stimulus projected onto the retina. Most importantly, pixel sizes were chosen to be smaller than cone receptive field sizes, for sufficient spatial resolution.

Linear-nonlinear Poisson hierarchical model

We model all the ganglion cells whose spikes were recorded simultaneously during an experiment jointly, with each cell indexed by i; in particular, the set of cones we infer is shared by all the ganglion cells of an experiment. However, to keep our notation simple, we start by suppressing i and focus on a single ganglion cell. As in (Field et al., 2010), we model each ganglion cell as a Linear-Nonlinear-Poisson (LNP) spike generator, with an exponential nonlinearity:

$$n_t \sim Poiss[\ e^{b+\mathbf{k}.\mathbf{s}_t}\, dt\ ]$$

.

We use the following notations:

$\mathbf{s}_t$: the stimulus at time t.

T : length of the experiment, in time bins.

n_t : spike train of the ganglion cell, discretized into spike counts per time bin.

$b \in \mathbb{R}$: an offset parameter that sets the neuron's baseline firing rate.

$\mathbf{k}$: the ganglion cell's receptive field, a linear filter acting on the stimulus $\mathbf{s}_t$.

$\mathbf{k}.\mathbf{s}_t$: the dot product of the ganglion cell's receptive field with the stimulus.

N_{spikes} : the total number of spikes recorded from the ganglion cell.

$\mathbf{STA} \;=\; \frac{1}{N_{spikes}} \sum_t n_t s_t$: the spike triggered average of the ganglion cell.

All of these quantities except for s_t and T are specific to a particular ganglion cell; they will all later be subscripted with ganglion cell number i.

The raw stimulus at time t and the raw **STA** have one temporal and two spatial dimensions. However, in these experiments ganglion cell receptive fields can be approximated as being separable in space and time (see (Chichilnisky and Kalmar, 2002) for further discussion), and their temporal profile is easily obtained from the singular value decomposition of **STA** as in (Gauthier et al., 2009) (see Appendix). Since our cone finding problem is essentially spatial in nature, to simplify calculations we start by integrating out this temporal dimension in a preprocessing step (Gauthier et al., 2009). In our notation, both the stimulus s_t at time t and the **STA** of each ganglion cell only have two spatial dimensions (as well as one color dimension): the temporal dimension has been integrated out, so that s_t represents an effective filtered stimulus that represents the recent history of the presented visual stimulus. See the appendix for full details.

Under an LNP model, the log-likelihood of seeing spike train $\{n_t\}$ given stimulus $\mathbf{s}_t$ and parameters b and $\mathbf{k}$ is

$$\begin{aligned} \log p(\ \{n_t\} \mid b\,,\ \mathbf{k}\,,\ \mathbf{s}_t\ ) &= \sum_{t=1}^{T} n_t(b + \mathbf{k}.\mathbf{s}_t) - e^{b+\mathbf{k}.\mathbf{s}_t} dt \ + const. \\ &= N_{spikes}\ (b + \mathbf{k}.\mathbf{STA}) - \sum_t e^{b+\mathbf{k}.\mathbf{s}_t} dt \ + const. \end{aligned}$$

Models of this type are well known and have been shown (Chichilnisky, 2001) to be adequate for macaque parasol ganglion cells given low-to-moderate effective contrast, as defined by the standard deviation of the dot product $\mathbf{k}.\mathbf{s}_t$. Note however that more elaborate models that include spike history couplings between ganglion cells have been shown to be more appropriate for high-contrast stimuli (Pillow et al., 2008) (see also (Keat et al., 2001; Pillow et al., 2005)). Since our stimulus has such small pixels, the receptive fields $\mathbf{k}$ cover many pixels, so that the effective contrast, which averages the stimulus variability over all of $\mathbf{k}$, is low enough here that spike history terms could be ignored. As discussed in the next section, this permitted the use of some approximations that were extremely helpful in speeding the computation.

Approximate profile log-likelihood

We begin with two key approximations of the loglikelihood:

$$\begin{aligned} \log p(\ \{n_t\} \mid b\,,\ \mathbf{k}\,,\ \mathbf{s}_t\ ) &\approx N_{spikes}\ (b + \mathbf{k}.\mathbf{STA}) \ - \ T \int e^{b+\mathbf{k}.\mathbf{s}} p(\mathbf{s}) d\mathbf{s} \ + \ const. \\ &= N_{spikes}\ (b + \mathbf{k}.\mathbf{STA}) \ - \ T \int e^{b+\mathbf{k}.\mathbf{s}} p(\mathbf{k}.\mathbf{s}) d(\mathbf{k}.\mathbf{s}) \ + \ const. \quad (1) \\ &\approx N_{spikes}\ (b + \mathbf{k}.\mathbf{STA}) \ - \ T \int e^{b+\mathbf{k}.\mathbf{s}} p_{gauss}(\mathbf{k}.\mathbf{s}) d(\mathbf{k}.\mathbf{s}) \ + \ const. \\ & \qquad (2) \\ &= N_{spikes}\ (b + \mathbf{k}.\mathbf{STA}) \ - \ T \ \exp\left(b + \frac{\sigma^2}{2}||\mathbf{k}||^2\right) \ + \ const. \quad (3) \end{aligned}$$

The first approximation replaces a sum over experimental time bins with an integral over the stimulus distribution, which becomes an equality in the limit of infinite experimental time T. See (Paninski, 2004; Park and Pillow, 2011; Ramirez and Paninski, 2012) for further discussion of this approximation. The second equation follows from the fact that the resulting integral only depends on the probability distribution of the one-dimensional projection $\mathbf{k}.\mathbf{s}$. The approximation in the third line is an application of the central limit theorem: since $\mathbf{k}.\mathbf{s}$ is a large weighted sum over many random variables (each element of the vector $\mathbf{s}$ in these experiments is independent, with mean zero and variance σ^2; the weights are given by $\mathbf{k}$), we can expect the distribution of $\mathbf{k}.\mathbf{s}$ to be approximately Gaussian (with mean zero and variance $\sigma^2||\mathbf{k}||_2^2$) for all receptive fields $\mathbf{k}$ with a sufficiently large number of nonzero pixels. (See (Diaconis and Freedman, 1984) for further discussion.) The final equality follows by analytically evaluating this Gaussian integral. (Note that this integral is not analytically available if spike history coupling terms are included in the model; see (Ramirez and Paninski, 2012) for further discussion.)

Note that this sequence of approximations entails a huge computational savings, since once the STA and number of spikes N_{spikes} are obtained, we can discard the stimuli $\mathbf{s}_t$ and spikes $\{n_t\}$ completely; in statistical language, the STA and N_{spikes} form an approximately sufficient statistic. Since the stimulus set is huge here (both in space, due to the high spatial resolution, and in T, because long experiments are required to collect enough data to adequately constrain the cone inference), this approximation makes the computation orders of magnitude more efficient.

We can simplify further if we note that it is possible to optimize for the offset parameter b in (3) analytically, as a function of $\mathbf{k}$:

$$\hat{b} = \log \frac{N_{spikes}}{T} - \frac{\sigma^2}{2}||\mathbf{k}||_2^2. \tag{4}$$

Plugging this back into (3) results in a rather dramatic simplification:

$$\frac{1}{N_{spikes}} \log p(\ \{n_t\} \mid \mathbf{k}\,,\ \mathbf{STA}\ ) \ \approx\ \ \mathbf{k}.\mathbf{STA}\ -\ \frac{\sigma^2}{2}\,||\mathbf{k}||^2\ +\ const.$$

In the statistical literature, this partially-maximized likelihood is often referred to as a "profile" likelihood; the key conclusion here is that the profile loglikelihood can be well-approximated as quadratic. The resulting approximately Gaussian likelihood will allow us to analytically integrate out the functional connectivity weights $\mathbf{a}$ between cones and ganglion cells that will be introduced in the next section.

If we do not impose any constraints, structure or regularization on the linear filter $\mathbf{k}$, then maximizing the likelihood would result in $\mathbf{k}$ being proportional to the **STA**; this becomes apparent by equating the gradient of (3) with respect to $\mathbf{k}$ to zero (Paninski et al., 2004; Park and Pillow, 2011). In this setting, we could write $\mathbf{k} = \alpha\mathbf{STA}$ and solve for the unknown α which maximizes the profile likelihood, leading to $\hat{\alpha} = 1/\sigma^2$. In the following, the linear filter $\mathbf{k}$ will be constructed from cone receptive fields, and it will therefore not be strictly proportional to the **STA**. However, the approximation $||\mathbf{STA}||/\sigma^2$ for the norm of $\mathbf{k}$ will continue to be useful below.

Weighted sums of cone receptive fields

We now assume that each ganglion cell's linear filter $\mathbf{k}$ is a weighted sum of appropriately placed cone receptive fields. The same set of cones is shared by all the ganglion cells in

a recording. We also assume that cones have typical and known receptive fields that only depend on each cone's location and color type: a cone of a given color has a circular Gaussian receptive field of a certain known width[1]. The vector of weights connecting cones to ganglion cell receptive fields will be denoted by $\mathbf{a}$; note that this vector depends implicitly on the number of cones, as well as their locations and colors.

We limit ourselves to a fixed region of interest of stimulus space comprising N_{pixel} pixels. Cones are placed with sub-pixel resolution: cone centers are not assumed to be located at pixel centers or corners. Instead, cone centers are allowed to be on a 4-by-4 square grid of locations within each pixel.

In practice, cone receptive fields appear pixelated: the true receptive fields are projected onto the N_{pixel} squares representing pixels. The pixelated receptive field is obtained by first placing the spatial component of a stereotyped cone receptive field (a Gaussian) at its location coordinates, without regard for color. This stereotyped receptive field is then integrated over the square surface of each stimulus pixel, so that we have calculated the relative integrated sensitivity of the cone to each pixel, resulting in a vector of N_{pixel} numbers representing the pixelated spatial profile of the cone receptive field. The color sensitivity of the cone, which is separable from its spatial profile, is then incorporated by taking a Kronecker product with the 3-element color sensitivity of the cone. The size $3\,N_{pixels}$ vector which we obtain represents the cone's sensitivity to all $3\,N_{pixels}$ pixels and colors. (Similarly each **STA**, which is already pixelated and in color by construction, is a vector of size $3\,N_{pixels}$ capturing a ganglion cell's sensitivity.)

We now represent any given cone configuration *cones* by a matrix $\mathbf{W}(cones)$ containing one column of size $3\,N_{pixels}$ for each cone in *cones*. Taking a weighted sum of cone receptive fields then reduces to multiplying this matrix on the right by a vector of weights:

$$\mathbf{k} \;=\; \mathbf{W}(cones)\;\mathbf{a}.$$

In practice, we know that cones are never closer to each other than a certain exclusion distance, corresponding to roughly one cone diameter, which is a parameter we assume known to the experimentalist. We can enforce such a cone exclusion with the use of a hard prior that gives zero probability to any cone configuration with cones that are too close together, as discussed further below.

Marginalizing out the weights a

Having replaced $\mathbf{k}$ with its value $\mathbf{W}\,\mathbf{a}$ in the approximate log-likelihood, we notice that the log-likelihood is a quadratic function of $\mathbf{a}$:

$$\frac{1}{N_{spikes}}\;\log p(\;\{n_t\}\mid \mathbf{W}\,,\;\mathbf{a}\,,\;\mathbf{STA}\;)\quad\approx\quad \mathbf{STA}^T\,\mathbf{W}\,\mathbf{a}\;-\;\frac{\sigma^2}{2}\,\mathbf{a}^T\mathbf{W}^T\mathbf{W}\mathbf{a}\;+\;const. \qquad (5)$$

Conveniently, this allows us to marginalize $\mathbf{a}$ out to obtain a likelihood that only depends on cone locations and colors, once we have specified a suitable prior distribution on $\mathbf{a}$. The quadratic kernel in the prior should capture the overlaps between pixelated cone receptive

[1]The cone width and minimal cone spacing (discussed further below) are two key parameters for the analysis here which are currently set by hand by the experimenter; incorrect parameter choices can be detected by visible mismatches between the spike-triggered averages and the inferred cone spacings. These parameters could in principle be selected by automatic criteria (e.g., cross-validation), but we have not yet pursued this approach.

fields: if cone receptive fields had no overlap, we would expect the weights $\mathbf{a}$ to be independent, and we would have chosen a prior where weights would have been independent of each other, so as to not introduce any spurious correlations between them. However, cone receptive fields have some overlap, due to optical blur and pixelization, implying that we expect weights to be correlated *a priori*. Thus it is natural to consider a prior inverse covariance matrix of the form $g\,\mathbf{W}^T\mathbf{W}$ for some proportionality constant g, since the matrix which captures overlaps due to pixelation is $\mathbf{W}^T\mathbf{W}$. Such a prior is also very convenient, since the inverse covariance of the weights $\mathbf{a}$ in the approximate likelihood of the data itself does not depend on the data and is of the same form, namely $\sigma^2\,\mathbf{W}^T\mathbf{W}$: this is a conjugate prior, which simplifies calculations. Since weights between cones and ganglion cells can be positive or negative, we take a zero mean prior. Thus we arrive at a prior of the form

$$prior\,(\,\mathbf{a}\mid cones\,) \;=\; \frac{1}{\sqrt{|2\pi(g\mathbf{W}^T\mathbf{W})^{-1}|}}\,\exp\Big(\,-\frac{g}{2}\mathbf{a}^T\mathbf{W}^T\mathbf{W}\mathbf{a}\Big)\,. \tag{6}$$

There is a natural way of setting the prior parameter g_i for ganglion cell i. Since $\mathbf{k}_i = \mathbf{W}\,\mathbf{a}_i$, the prior (6) can be seen as a prior on $\mathbf{k}_i$: $p(\,\mathbf{k}_i\,|\,cones\,) \;\propto\; \exp(\,-g_i\,||\mathbf{k}_i||^2/2)$, and g_i is a parameter which sets our *a priori* expectation of the norm of $\mathbf{k}_i$. However, recall that we could calculate the norm of $\mathbf{k}_i$ in advance when we calculated the offset parameter b, which gave $\mathbf{k}_i \;\approx\; \mathbf{STA}_i/\sigma^2$. We therefore choose the scalar parameter g_i as

$$\frac{1}{g_i} \;= \left|\left|\frac{\mathbf{STA}_i}{\sigma^2}\right|\right|^2,$$

so that the *a priori* expected norm of $\mathbf{k}_i$ is equal to the square norm of $\mathbf{STA}_i/\sigma^2$.

With our prior thus specified, we can proceed to marginalize out the weights $\mathbf{a}$. Considering a single ganglion cell and dropping the index i for simplicity:

$$\begin{aligned}
p\,(\,data\,|\,cones\,) \;&=\; \int d\mathbf{a}\; prior\,(\,\mathbf{a}\mid cones\,)\; p\,(\,data\,|\,cones\,,\mathbf{a}\,) \\
&\propto\; \frac{1}{\sqrt{|2\pi(g\mathbf{W}^T\mathbf{W})^{-1}|}} \int d\mathbf{a}\;\exp\left[\;N_{spikes}\left(\mathbf{STA}^T\,\mathbf{W}\,\mathbf{a}\;-\frac{\sigma^2}{2}\mathbf{a}^T\mathbf{W}^T\mathbf{W}\mathbf{a}\right)\;-\frac{g}{2}\mathbf{a}^T\mathbf{W}^T\mathbf{W}\mathbf{a}\right]
\end{aligned}$$

$$\begin{aligned}
2\;\log p\,(\,data\,|\,cones\,) \;&=\; \frac{N_{spikes}^2}{N_{spikes}\,\sigma^2+g}\;\mathbf{STA}^T\,\mathbf{W}\,\big(\,\mathbf{W}^T\mathbf{W}\big)^{-1}\,\mathbf{W}^T\,\mathbf{STA} \\
&-\;\log\;\left|\frac{2\pi}{g}(\mathbf{W}^T\mathbf{W})^{-1}\right|\;+\;\log\;\left|\frac{2\pi}{N_{spikes}\,\sigma^2+g}(\mathbf{W}^T\mathbf{W})^{-1}\right| + const. \\
&=\; \frac{N_{spikes}^2}{N_{spikes}\,\sigma^2+g}\;\mathbf{STA}^T\,\mathbf{W}\,\big(\,\mathbf{W}^T\mathbf{W}\big)^{-1}\,\mathbf{W}^T\,\mathbf{STA} \\
&+\;dim(\mathbf{a})\;\log\;\left(\frac{g}{N_{spikes}\,\sigma^2+g}\right) + const.
\end{aligned}$$

This last equation is the log-posterior of observing the spikes from a given ganglion cell given a particular cone configuration. The first term in this equation is always positive. It reflects the likelihood of placing cones at particular locations, by how much the cones (represented in $\mathbf{W}^T\mathbf{W}$) contribute to explaining the ganglion cell's **STA**. The second term, which stems

from the normalization constants in the Gaussian integral, is negative and proportional to the dimension of the Gaussian integral, namely the number of cones $dim(\mathbf{a})$ which have non-zero connections with the ganglion cell: this term is a Bayesian penalty on the complexity of the model, which in our case is controlled by the dimensionality of $\mathbf{a}$. This second term effectively sets the boundary between connection weights in $\mathbf{a}$ that stand out from the noise and connections which do not contribute enough to the posterior and thus should be set to zero.

For computational convenience, and to avoid over-fitting, we determine which cone locations are allowed to have non-zero connections to which ganglion cells in advance. The connection between a ganglion cell and a cone location is allowed to be non-zero if the data likelihood of the cone configuration consisting of a single cone at that location and color is positive. In other words, the connection is set to zero if the first term in the equation above is smaller than than $\log(N_{spikes}\,\sigma^2 + g) - \log g$ for single cone configurations. The number of cones which are allowed to be non-zero for a given ganglion cell and a given cone configuration is the $dim(\mathbf{a})$ which appears above.

Sparsifying the connectivity matrix between cones and ganglion cells is more than just a way to avoid over-fitting. If all connections were allowed, every ganglion cell would be connected to every cone, including spurious cones that are in areas of the visual field where the **STA**s are purely noise, i.e. very far from any ganglion cell receptive field. In the limit where the **STA**s are taken in a region of interest of the visual field that is much larger than all the ganglion cell receptive fields, the norm of each **STA** would be dominated by noise, and most inferred cones would be spurious; both of the terms in the posterior would be dominated by noise. This would invalidate the simple requirement that our model infer qualitatively similar cone configurations regardless of the size of the region of interest considered, as long as this region includes the bulk of all ganglion cell receptive fields. Our sparsification, by allowing ganglion cells to be locally connected to only those cones which are statistically significantly part of their receptive field, makes our inferences independent of the size of the region of interest considered.

For a recording with multiple ganglion cells, we are interested in the log-posterior given the responses of all of the observed ganglion cells. Since under our LNP model, the spike trains of ganglion cells are independent when conditioned on the stimulus and model parameters, this joint likelihood is simply the sum of the likelihoods of each ganglion cell spike train. Indexing ganglion cells by i, and with $dim(\mathbf{a}_i)$ determined as described above:

$$\begin{aligned}\log p\,(\,data\,|\,cones\,,\,stimulus\,) \;=\; & \frac{1}{2}\sum_i \left[\,\frac{N^2_{spikes_i}}{N_{spikes_i}\,\sigma^2 + g_i}\,\mathbf{STA}_i^T\,\mathbf{W}\,(\,\mathbf{W}^T\mathbf{W}\,)^{-1}\,\mathbf{W}^T\,\mathbf{STA}_i\,\right] \\ & -\frac{1}{2}\sum_i dim(\mathbf{a}_i)\log\left(\frac{N_{spikes_i}\,\sigma^2 + g_i}{g_i}\right) + const. \qquad (7)\end{aligned}$$

Once again, the first term in (7) is always positive. It reflects how much the cones (represented in $\mathbf{W}^T\mathbf{W}$) contribute to explaining all **STA**s jointly. The second term is a penalty proportional to the total number of connections between cones and ganglion cells, which was used to make these connections sparse in advance. It is convenient to normalize this likelihood by forming

$$\frac{1}{(\log 2)\sum_i N_{spikes,i}}\Big[\,\log p\,(\,data\,|\,cones\,,\,stimulus\,) - \log p\,(\,data\,)\,\Big];$$

we have subtracted off the log-probability of spiking under the zero-order model that all the observed ganglion cells fire in a homogeneous Poisson manner, independent of the stimulus, and then normalized to obtain a quantity expressed in terms of bits per spike. All log-likelihoods presented in the results section will be of this form.

Now that we have marginalized out $\mathbf{a}_i$ for all ganglion cells indexed by i, we do not need to keep track of these weights during our MCMC simulations. However, we still have access to the posterior distribution over these weights, since given any cone configuration, the posterior is jointly Gaussian with a known mean and covariance given by (for ganglion cell i):

$$\begin{aligned} \mathbb{E}\,(\,\mathbf{a}_i \,|\, cones\,,\, data\,) &= \left[(N_{spikes_i}\,\sigma^2 + g_i)\mathbf{W}^T\mathbf{W}\right]^{-1}\; N_{spikes_i}\;\sigma^2\;\mathbf{W}^T\;\mathbf{STA}_i \\ \mathbf{Cov}\,(\,\mathbf{a}_i \,|\, cones\,,\, data\,) &= \left[(N_{spikes_i}\,\sigma^2 + g_i)\mathbf{W}^T\mathbf{W}\right]^{-1} \end{aligned}$$

We can use these expressions to estimate $\mathbb{E}\,(\,\mathbf{a}_i \,|\, data\,)$: we sample cone configurations *cones* from the posterior distribution $p(\,cones \,|\, data\,)$, then form the expectation of $\mathbf{a}_i$ over *cones* given *data* by averaging over our sample of cones:

$$\mathbb{E}\,(\,\mathbf{a}_i \,|\, data\,) \;=\; \mathbb{E}_{cones\,|\,data}\;\mathbb{E}\,(\,\mathbf{a}_i \,|\, cones\,,\, data\,) \;=\; \mathbb{E}_{samples}\;\mathbb{E}\,(\,\mathbf{a}_i \,|\, cones\,,\, data\,)\,.$$

Similarly, we obtain the covariance of $\mathbf{a}_i$ by using the following expression, which combines the expressions given above for the expectation and covariance of $\mathbf{a}_i$ given *cones* and *data*:

$$\mathbf{Cov}\,(\,\mathbf{a}_i \,|\, data\,) \;=\; \mathbb{E}_{samples}\;\mathbf{Cov}\,(\,\mathbf{a}_i \,|\, cones\,,\, data\,) + \mathbf{Cov}_{samples}\;\mathbb{E}\,(\,\mathbf{a}_i \,|\, cones\,,\, data\,)$$

Using such a collapsed sampler, i.e. a sampler in which the weights $\mathbf{a}_i$ have been marginalized out, is both a great computational relief and statistically more efficient (due to the Rao-Blackwell theorem (Casella and Berger, 2001)) than the naive alternative of forming $\mathbb{E}\,(\,\mathbf{a}_i \,|\, data\,)$ and $\mathbf{Cov}\,(\,\mathbf{a}_i \,|\, data\,)$ by sampling from $p(\,a\,,\, cones \,|\, data)$ and averaging over samples of both cones and weights $\mathbf{a}$.

Visualizing the evidence

For visualization purposes, we would like to see in a single plot how much evidence there is for placing cones of different colors at different locations: we would like to plot a color map indicating where cones are most likely to be, given the observed data. In order to see what color should be displayed for a particular location, we consider cone configurations with a single cone, for which $\mathbf{W}$ consists of a single column $\mathbf{w}$. For a particular cone with receptive field in column vector $\mathbf{w}$, the log-posterior (7) is a scalar $V(\mathbf{w})$ given by:

$$V(\mathbf{w}) \;=\; \frac{1}{2}\;\sum_i\;\max\left[\,0,\;\frac{N_{spikes_i}^2}{N_{spikes_i}\,\sigma^2 + g_i}\;(\mathbf{STA}_i^T\,\mathbf{w})^2 -\;\log\;\left(\frac{N_{spikes_i}\,\sigma^2 + g_i}{g_i}\right)\right]$$

The max operation carried out for each ganglion cell reflects the sparseness of connections between ganglion cells and cones; a ganglion cell only contributes to $V(\mathbf{w})$ if the contribution of $\mathbf{w}$ to explaining its $\mathbf{STA}$ is larger than a penalty term.

Consider the three cones which could possibly be placed at a particular location, represented by the three column vectors $\mathbf{w}_{red}$, $\mathbf{w}_{green}$ and $\mathbf{w}_{blue}$. To visualize the evidence for placing a cone of each of the three colors at this location, we multiply the 3×1 vector of likelihoods $[V(\mathbf{w}_{red})\;V(\mathbf{w}_{green})\;V(\mathbf{w}_{blue})]$ by the 3-by-3 matrix $COLOR^{-1}$, where $COLOR$

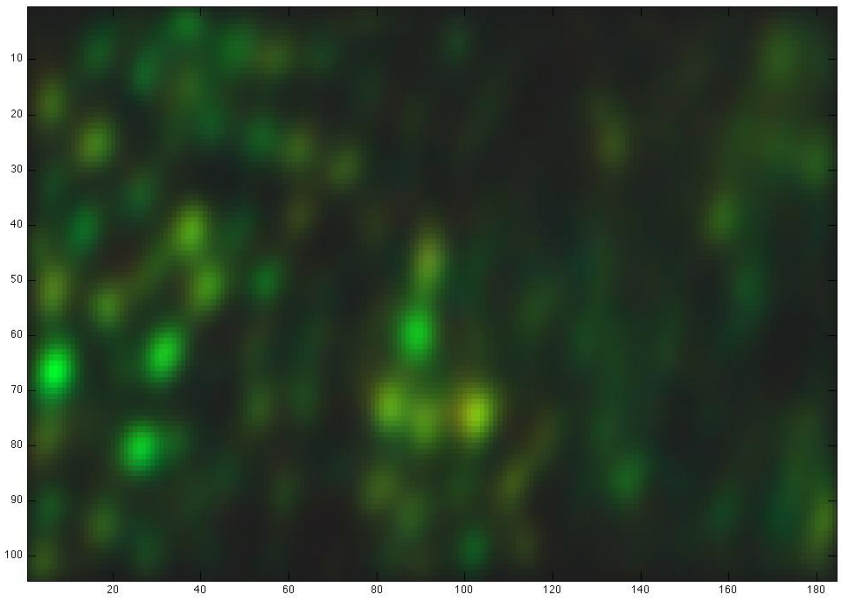

Figure 1: Visual summary of the evidence for cones across spatial locations. Dark areas show less evidence for the presence of a cone, while red, green and blue areas show evidence for cones of the corresponding types. The colors displayed, which cannot be assigned meaningful units, were obtained as explained in the main text. The axes count the number of possible cone positions considered, which is 4 times the number of pixels in the stimulus: the evidence map is supersampled by a factor of 4 compared to the original stimulus and **STA**s.

is the matrix of light sensitivities of the three cone types. This results in a RGB color vector where differences between the colors displayed for the evidence for the three types of cones are more pronounced; it is most particularly useful to exaggerate color differences because the spectral sensitivity of red and green cones are so close that the difference between evidence for these two cone types would not otherwise be distinguishable in the plot. Figure (1) shows the resulting evidence map for the first dataset we will examine below.

MCMC sampling

Using our likelihood (7), our goal is to infer likely cone configurations, consisting of a set of cone locations and colors. Because of the hard cone exclusion prior, the space of possible cone configurations is not a simple subset of a vector space, complicating inference. In this context, one method for finding likely configurations could involve sampling from the posterior distribution using Markov Chain Monte Carlo (MCMC) methods (Robert and Casella, 2005). An MCMC chain is a random walk in the space of admissible cone configurations, where care has been taken to ensure that the time spent visiting a configuration is proportional to its posterior probability, at least asymptotically. Starting from an initial configuration which may not be very likely, the configuration will walk randomly towards more likely configurations, during an initial phase called burn-in, where the chain is not yet stationary in its distribution of sampled configurations. The MCMC chain will eventually reach the region of likely configurations, where it will spend most of its time, sampling in a stationary manner

from the posterior distribution.

In order to ensure that the stationary distribution of the random walk coincides with our desired posterior $p(\, cones \,|\, data \,)$, it is sufficient that the transition probabilities $t(\, cones' \,|\, cones \,)$ governing the walk satisfy detailed balance:

$$t(\, cones' \,|\, cones \,)\; p(\, cones \,|\, data \,) \;=\; t(\, cones \,|\, cones' \,)\; p(\, cones' \,|\, data \,).$$

We ensure that our random walk $t(\, cones' \,|\, cones \,)$ satisfies detailed balance by using the Metropolis-Hastings algorithm (Robert and Casella, 2005). From configuration *cones*, a move to a candidate configuration *cones*$'$ is proposed randomly with probability $proposal(\, cones' \,|\, cones \,)$, and with a probability $proposal(\, cones \,|\, cones' \,)$ of proposing the reverse move from *cones*$'$. We then accept the proposed move with probability

$$p_{accept}(\, cones' \,|\, cones \,) \;=\; min\left(1,\; \frac{p(\, cones' \,|\, data \,)\; proposal(\, cones \,|\, cones' \,)}{p(\, cones \,|\, data \,)\; proposal(\, cones' \,|\, cones \,)}\right). \qquad (8)$$

Finding the set of likely cone configurations entails sampling from the product of the likelihood (the exponential of (7)) with the hard cone exclusion prior on cone configurations. We sample from this distribution with a random walk in the allowed cone configuration space. However, because of the hard exclusion prior, the posterior has many deep local optima, and naive MCMC samplers will get stuck in these traps, leading to unacceptably slow mixing.

The sampling problem behaves differently in regions of visual space where the ganglion cell recordings carry strong evidence about cone locations, and regions with a dearth of evidence. A region with strong evidence, for example at the location of an **STA** from a ganglion cell with many spikes, will constrain cone placement more than a region outside of the receptive field center of all recorded ganglion cells. In Figure 1, the former show as bright regions, while the latter are dark.

For regions where there is a lot of evidence for cones, the typical differences in likelihood between different configurations are very large: if we were to sample from the distribution of cone configurations in such regions, one particular configuration could prevail as being astronomically more likely than all other configurations. For these regions, all that is important is to find the single most likely configuration. On the other hand, in regions where there is little evidence constraining cone configurations, the small differences in likelihoods of various configurations warrant sampling from the ensemble of possible configurations.

In order for MCMC to sample the set of likely cone configurations without getting stuck in deep wells of probability arising from regions with strong evidence and the cone exclusion prior, we combine two strategies: we take care to only propose Monte Carlo moves which respect the cone exclusion prior, and we use an adaptive simulated tempering technique (Salakhutdinov, 2010) which effectively flattens the likelihood landscape, keeping the sampler from getting stuck in local maxima of likelihood.

Adaptive simulated tempering

In order to avoid getting stuck in local maxima, we use the CAST (Coupled Adaptive Simulated Tempering) sampler proposed in (Salakhutdinov, 2010). This sampler is a hybrid (Atchadé and Liu, 2010) of the Wang-Landau algorithm (Wang and Landau, 2001) and of the sampling scheme known as simulated tempering (Marinari and Parisi, 1992), closely related to parallel tempering, aka replica exchange Monte Carlo, all of which flatten the likelihood

landscape using an additional parameter γ which plays the role of an inverse temperature in physics; see (Earl and Deem, 2005) for a review. The common idea in all these schemes is to replace the data likelihood $p(\,data\,|\,cones\,)$ with a family of functions $p_\gamma(\,data\,|\,cones\,)$ where $p_{\gamma=1}(\,data\,|\,cones\,)$ is equal to $p(\,data\,|\,cones\,)$, and which become flatter (with less pronounced local maxima) as γ moves away from 1. We are only interested in samples from the true distribution, i.e. $p_{\gamma=1}$, but we also let the system reach other values of γ at which it can escape from local maxima. See the next section for our particular definition of p_γ.

The CAST sampler consists of two MCMC chains that run in parallel, a so-called "fast" chain and a "slow" chain. The slow chain has fixed $\gamma = 1$, so it samples the desired distribution of cone configurations using the Monte Carlo moves detailed above. Because it has $\gamma = 1$, the slow chain often gets stuck in local maxima on its own. The fast chain has a γ which is allowed to fluctuate among a predefined ordered set of values after every regular Monte Carlo move. The set of γ ranges from 1 to values far enough from 1 to ensure that local maxima are shallow and do not trap the chain. The fluctuations of γ are controlled by a Wang-Landau scheme (Wang and Landau, 2001; Atchadé and Liu, 2010), which by design ensures that γ samples the whole range of available values roughly uniformly. In particular, the γ variable is guaranteed to regularly visit both values far from 1, where local maxima are not a problem, and the value $\gamma = 1$ we are interested in. We used 20 values of γ, ranging from 1 to values which flatten the likelihood enough to ensure that the fast chain does not get stuck in local minima (see section below on flattening the likelihood). The 20 steps between the hottest temperatures and temperature 1 allow the fast chain to cool down to configurations that are more typical of the slow chain as it reaches temperature 1. Since the Wang-Landau scheme ensures that the fast chain samples all temperatures, the exact progression of temperatures used is less important than for other schemes such as parallel tempering, for which the temperature schedule must be chosen very carefully (Earl and Deem, 2005).

Whenever the fast chain returns to $\gamma = 1$, we allow for cones to be swapped between the slow and fast chains' cone configurations. However, we do not swap the whole cone configurations between the two chains, as the simulated tempering algorithm usually entails; instead, we choose the smallest possible subsets of cones to be swapped, so as to maximize the probability of accepting swap moves, ensuring that information is transferred as fluidly as possible between the two chains (see the end of this section for how these subsets of cones are chosen). This exchange of cones between the slow and fast configurations will often knock the slow chain's cone configuration out of a local maximum it was stuck in. Swapping a particular subset of cones in the fast chain with a subset of cones in the slow chain is accepted or rejected according to how the swap affects the joint log-likelihood $\log p_{fast}(\,data\,|\,cones_{fast}\,) + \log p(\,data\,|\,cones_{slow}\,)$ using the Metropolis-Hastings acceptance probability (8) as for regular MCMC moves, where p_{fast} is the distribution sampled from by the fast chain when $\gamma = 1$.

The groups of cones proposed for swapping are chosen to be the smallest groups that can be swapped with each other without violating the cone exclusion distance. These groups of cones can be found by simple agglutination. If $cone_1$ in the fast chain overlaps $cone_2$ in the slow chain, then these two cones must be swapped together. If in turn cone $cone_2$ in the slow chain overlaps cone $cone_3$ in the fast chain, all three cones must be swapped together. This agglutination is continued until the cones in the group overlap no other cones than themselves. If cones are left that have not been agglutinated into a group yet, a new group is started with one of them chosen at random. This agglutination could in principle "percolate," leading to minimal groups of cones which comprise most or all of the cones across the two chain configurations. However, in practice, the sizes of groups obtained via this procedure are

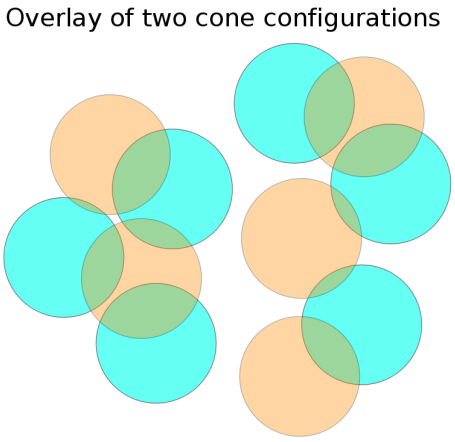

Figure 2: Overlay of the slow cone configuration (brown) and the fast cone configuration (cyan) in a toy example making the overlap relation apparent by partial transparency. Color here denotes belonging to one of the two chains, and is not to be confused with cone type, which is not denoted in this figure. In this toy example, the transitive closure of the overlap relation defines two equivalence classes of connected cones: one with five cones on the left, and one with six cones on the right. The equivalence class (or connected component) on the left has two brown and three cyan cones, while the one on the right has three brown and three cyan cones.

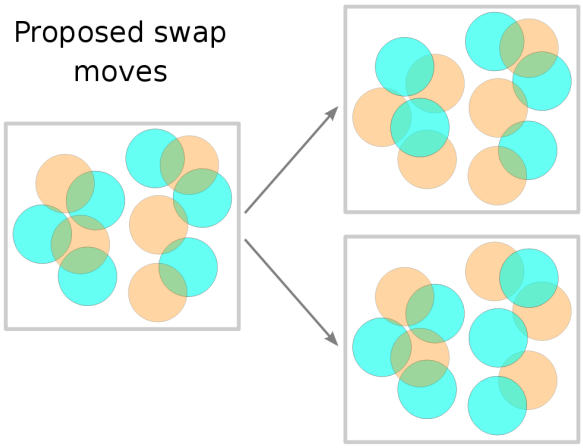

Figure 3: For the same pair of configurations as in the previous figure (left), two swap moves would be considered, one for each equivalence class (right). Each of the two admissible swap moves consists in swapping the brown cones in one equivalence class with the cyan cones in the same equivalence class. Note that only the memberships in one of the two chain configurations are changed; cone type (not denoted in this figure) and spatial location are unchanged.

small and localized. This is very convenient, since the smaller the groups of cones proposed for swapping, the greater the chances of accepting that MCMC move.

More formally, the agglutination procedure described above is a simple algorithm for

computing the transitive closure (Lidl and Pilz, 1997) of the overlap relation $\mathcal{R}$ between cones. For $cone_1$ in the fast chain and $cone_2$ in the slow chain, $\mathcal{R}(cone_1, cone_2)$ is true if and only if swapping $cone_1$ out of the fast chain would require swapping $cone_2$ out of the slow chain in order to preserve cone exclusion. In other words, $\mathcal{R}(cone_1, cone_2)$ is true when the two cones overlap in space, even though they currently belong to different chains. This relation $\mathcal{R}$ is reflexive (every cone overlaps itself) and symmetric (if $cone_1$ overlaps $cone_2$, then $cone_2$ overlaps $cone_1$). The 'must be swapped together' relation $\mathcal{B}$ which we calculate is the transitive closure of $\mathcal{R}$: it is the smallest binary relation which is implied by $\mathcal{R}$ and which is also transitive, i.e. for which $\mathcal{B}(cone_1, cone_2)$ and $\mathcal{B}(cone_2, cone_3)$ implies $\mathcal{B}(cone_1, cone_3)$. Calculating the transitive closure of $\mathcal{R}$ is computationally straightforward: one proceeds by agglutination similarly to the way described above; we use a particularly simple and fast algorithm called the Floyd-Warshall algorithm (Roy, 1959; Warshall, 1962). Once it is obtained, this transitive closure defines a number of equivalence classes, i.e. a number of groups of cones which must be swapped together. Figure 2 shows an example in which two groups of cones must be swapped together, while Figure 3 shows the two corresponding swap moves.

Our agglutination process for finding groups of cones to swap together is designed to find the smallest groups of cones that can possibly be swapped together without violating the cone exclusion distance. One notable property of the groups of cones we obtain is that if any of the groups of cones that form an equivalence class are swapped, the groups that would be found by agglutination for the new configurations are the same as the groups found before the swap. We use this property to calculate groups once, and then run several swap moves without needing to re-agglutinate groups. Any time the fast chain returns to $\gamma = 1$, we choose at most 50 such groups at random by Floyd-Warshall agglutination, and let Metropolis-Hastings accept or reject swaps using a group chosen among these at random 50 times. The likelihood used to accept or reject swap moves is the independent joint log-likelihood $\log p_{fast}(\, data \,|\, cones_{fast}\,) + \log p(\, data \,|\, cones_{slow}\,)$, which as argued above ensures that the stationary distribution sampled by $cones_{slow}$ is the desired distribution $p(\, cones_{slow} \,|\, data\,)$.

Flattening the data likelihood

The most common method for "flattening" an energy landscape for simulated or parallel tempering is to simply scale the log-likelihood by a proportionality factor β, which is interpreted as an inverse temperature. In practice for our problem, this can help unfreeze cones that are in regions with very little evidence, while cones remain stuck in regions with mildly more evidence. We can try to unfreeze cones more uniformly across space by scaling the log-likelihood in a way that evens out the effective evidence. We do this by raising the $\mathbf{STA}_i^T \mathbf{W} \left(\mathbf{W}^T\mathbf{W}\right)^{-1} \mathbf{W}^T \mathbf{STA}_i$ term in (7) to a power δ smaller than 1. In this way, we carry out simulated tempering with a chain of data likelihoods of the form:

$$\begin{aligned} \log p\,(\, data \,|\, cones\,,\, \beta, \delta\,) \;=\; & \sum_i \; \frac{N_{spikes_i}^2}{2\beta(N_{spikes_i}\,\sigma^2 + g_i)} \left(\mathbf{STA}_i^T \,\mathbf{W}\,\left(\,\mathbf{W}^T\mathbf{W}\right)^{-1}\mathbf{W}^T\,\mathbf{STA}_i\right)^{\delta} \\ & + \sum_i \; \frac{dim(\mathbf{a}_i)}{2\beta} \log\left(\frac{g_i}{N_{spikes}\,\sigma^2 + g_i}\right) \end{aligned} \tag{9}$$

In practice, we allow the temperature to take 20 values $\gamma = (\beta, \delta)$ ranging from $(1, 1)$ to $(0.2, 0.1)$.

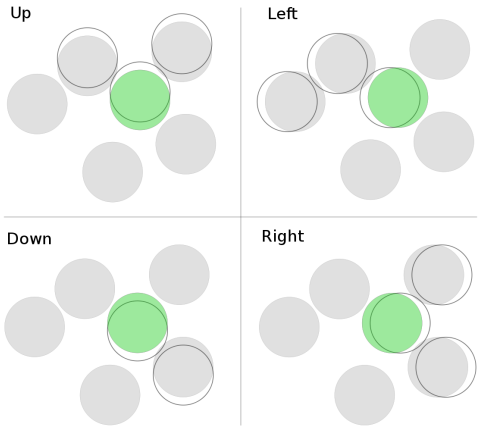

Figure 4: Shift moves in the four cardinal directions. The modified cone locations in each proposed move are denoted by empty circles. Shifting the green cone in each direction induces shifts of cones that are in the way.

Monte Carlo moves

The data likelihood (9) depends only on the set of cone locations and colors, since the weights from cones to ganglion cells have been marginalized out. The posterior we wish to sample from consists of the product of the data likelihood with the hard cone exclusion prior. Sampling proceeds by only proposing random changes in the current cone configuration that respect the hard cone exclusion prior, accepting or rejecting each change in accordance to the data likelihood.

The proposed Monte Carlo moves we use are generated as follows: first, a possible cone location is chosen. With 50% probability, a location which is currently unoccupied by a cone and which is far enough from existing cones to avoid cone exclusion is chosen; one of the three cone types is chosen to be added to that location. With 50% probability, a location which is already occupied by a cone is chosen; for the cone already present at that a location, the possible moves are to change its color, remove it, or shift it to a neighboring pixel. For the chosen move, forward and backward proposal probabilities are calculated, and the move is then accepted or rejected using the Metropolis-Hastings acceptance probability (8).

During moves which consist in shifting an existing cone to a neighboring location, shifts are propagated to any cones that are in the way: for example, the naive move consisting in shifting a single cone to the left is replaced by a shift to the left of all cones which must be moved together in order to respect the cone exclusion (Fig. 4). In other words, if there are one or more cones in contact with the left of the cone being shifted, those cones are bumped over to the left with it. Any cones which are bumped off of the region of possible cone locations considered are deleted.

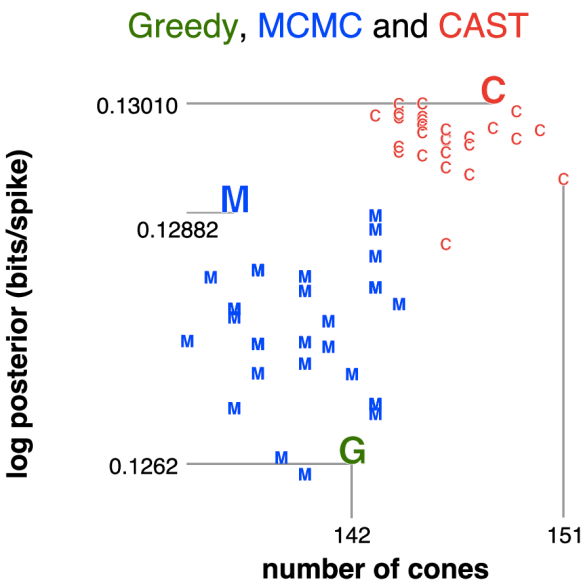

Figure 5: First dataset: posterior log-likelihood plotted against number of cones for (G) the greedy solution, (M) the best configurations over 30 simple MCMC runs with 10^6 iterations, and (C) the best configurations over 30 CAST runs with 10^6 iterations. MCMC configurations were reinitialized to the lazy greedy cone configuration (see the MCMC and CAST results section) whenever the likelihood did not improve for 400 consecutive iterations. In each category (G, M and C), the marker for the cone configuration with highest likelihood is magnified, and its log-posterior likelihood given. All log-likelihoods are in *bits per spike*.

Results

Datasets

We apply our cone finding algorithms to two datasets, one collected from 21 ganglion cells, and one collected from 324 ganglion cells. Our model only requires knowing these cells' **STA**s, the number of recorded spikes for each cell, and the mean and variance of the stimulus at each pixel. For the first dataset, we limit ourselves to a region of interest in cone location space which spans 26 by 46 stimulus pixels (see (Field et al., 2010) for full details, scale bars, etc.); for the second dataset, we limit ourselves to an area of 160 by 160 pixels. These regions were chosen to include most of the power from all the ganglion cell **STA**s. As mentioned earlier, permissible cone locations were chosen to be a square lattice 4 times finer than the pixel lattice, for a total of 16 possible cone locations per pixel.

Apart from their difference in size, the main difference between the two datasets is that the first one has positive evidence in almost all locations, whereas the second larger dataset has many areas where the evidence is not strong enough to warrant placing cones with any confidence.

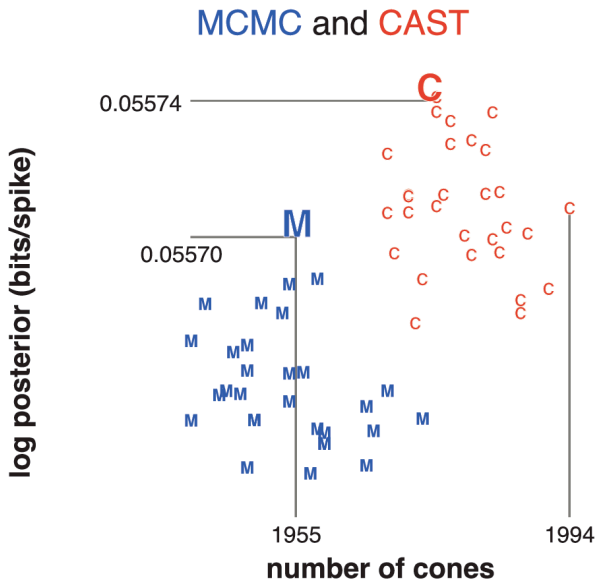

Figure 6: Second dataset: posterior log-likelihood plotted against number of cones. Conventions as in Fig. 5, except MCMC configurations were reinitialized whenever the likelihood did not improve for 8,500 consecutive iterations. For comparison, the log-likelihood per spike for the greedy solution was significantly lower, with 0.0549 *bits per spike* for 1879 cones.

The greedy algorithm

We compare our MCMC results to the configuration obtained by a straightforward greedy procedure: using the same posterior log-likelihood (7) as the MCMC algorithm, cones are added one by one with the location and type that maximizes the posterior given all the cones accumulated so far. This greedy accumulation of cones starts without cones, and stops when adding a cone can only reduce the posterior. See Figs. 5-8.

The greedy algorithm, while quick to find configurations with many cones and reasonably high likelihoods, has several drawbacks. One drawback is that by committing to each cone's placement greedily, we are missing opportunities to increase the likelihood by shifting groups of cones which are in contact together. This can be seen by seeding an MCMC run with the result of the greedy algorithm: the likelihood increases quickly from the first few MCMC iterations as groups of nearby cones shift by small distances, relaxing into more likely positions.

Another less obvious drawback of the greedy algorithm is that it is difficult to estimate the number of cones whose inference we can be confident about. This problem can be seen by tracking the increase in likelihood as a function of the number of cones (Fig. 9). With the greedy algorithm, the likelihood plateaus many iterations before the algorithm terminates (that is, when the addition of any further cones would decrease the posterior): the last cones added only contribute very little to the likelihood. This is most notable on our second dataset, which has both areas with high evidence and areas with zero to little evidence; for this dataset, the log-likelihood increases from 0.0548 *bits/spike* for 1689 cones to 0.0549 *bits/spike* for 1879 cones. The greedy algorithm places as many cones as can fit in the areas with non-zero

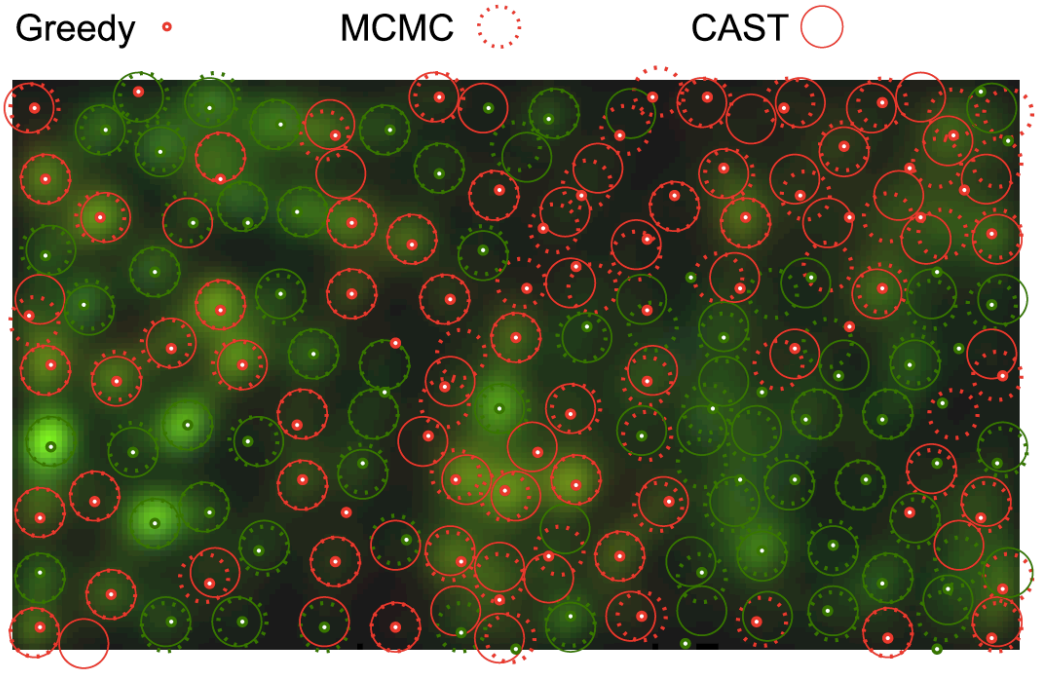

Figure 7: First dataset: superposition of the greedy configuration, the most likely configuration out of 3×10^6 iterations worth of MCMC runs, and the most likely out of as many iterations of CAST runs. The background color-scale image is a depiction of the evidence for the presence of cones of different colors across visual space (see main text).

evidence, however small this evidence, without giving any indication as to how confident we can be in each cone's position or color. This makes it very difficult to say with confidence how many cones should be kept, and which cones should be considered to be spurious or poorly constrained by the observed data.

Note that all of the results shown here and below are evaluated on the full data set, with no splitting between "training" and "test" data, because our focus is on how well the algorithms are able to explore and optimize the full log-posterior, rather than the predictive error of the resulting estimators.

MCMC and CAST results

We ran 30 MCMC and CAST chains for both datasets for 10^6 iterations each. Each chain was initialized with the same configuration obtained by a faster version of the greedy algorithm, which we will call the "lazy" greedy algorithm, and which we now briefly describe. (An advantage of the methods described here is that they are highly parallelizable. All analyses performed here were performed on a parallel cluster computer; similar analyses performed in serial on a single processor would be feasible, but slower by an order of magnitude.)

Implementing the greedy algorithm requires keeping a map of how much the likelihood would increase if a cone were to be added at each possible available location and color. This map allows us to select the single cone addition which will increase the likelihood the most. Every time a new cone is added, this map of likelihood increases must be updated within the

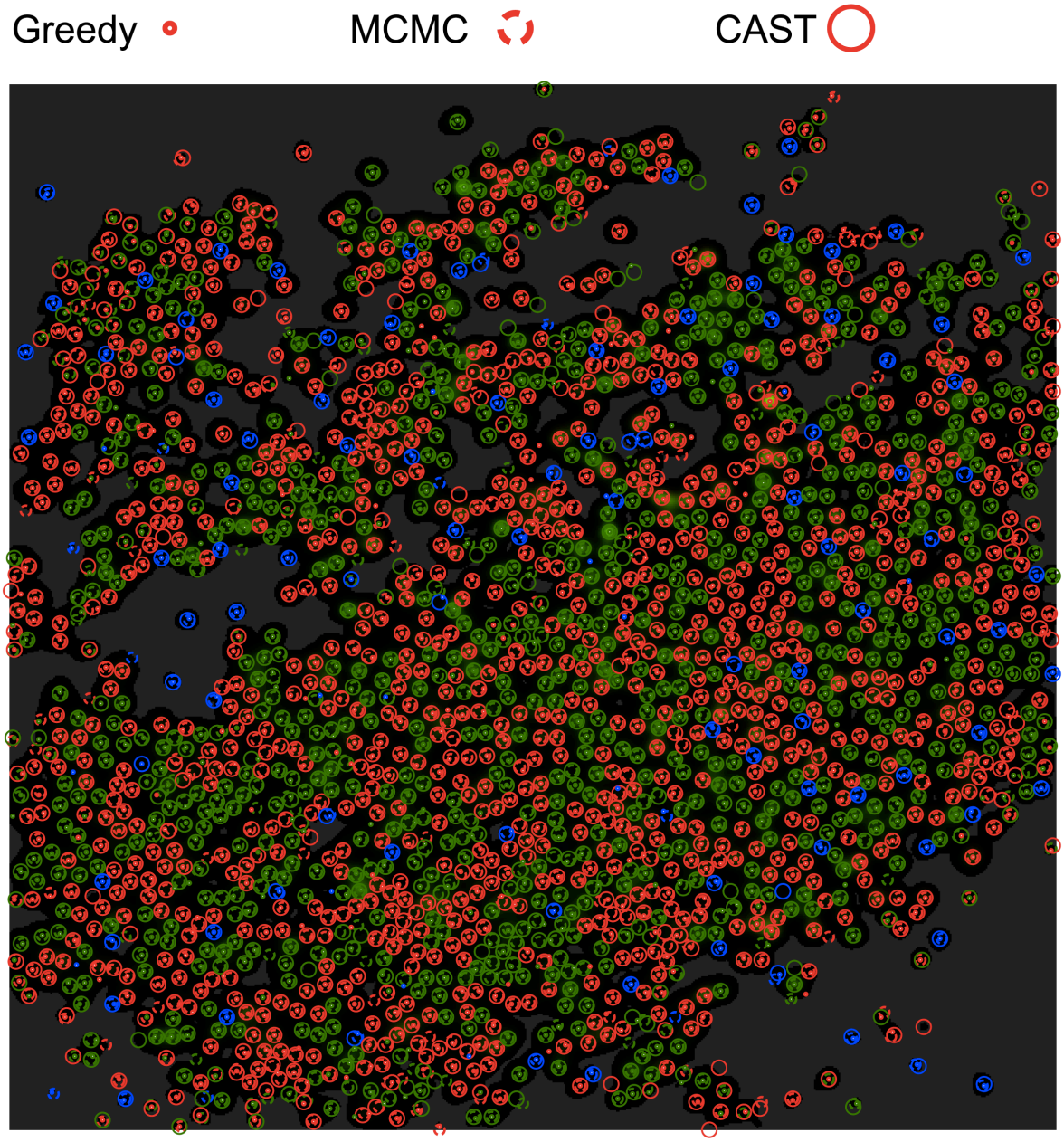

Figure 8: Second dataset: superposition of the greedy configuration with the best CAST and MCMC configurations, as in Fig. 7. Cones concentrate mostly in bright regions, i.e. in regions with a lot of evidence for cones of a particular color. Regions that are darker than a certain threshold are devoid of cones, because the evidence for cones in these regions is smaller than the penalty incurred for adding a new cone, which is induced by the prior on weights (see text explaining eq. (7)); such regions have been uniformly colored a lighter shade of grey.

vicinity of the new cone. It is easy to devise a much faster but less precise algorithm which does not update the likelihood map after each cone addition. This leads to configurations

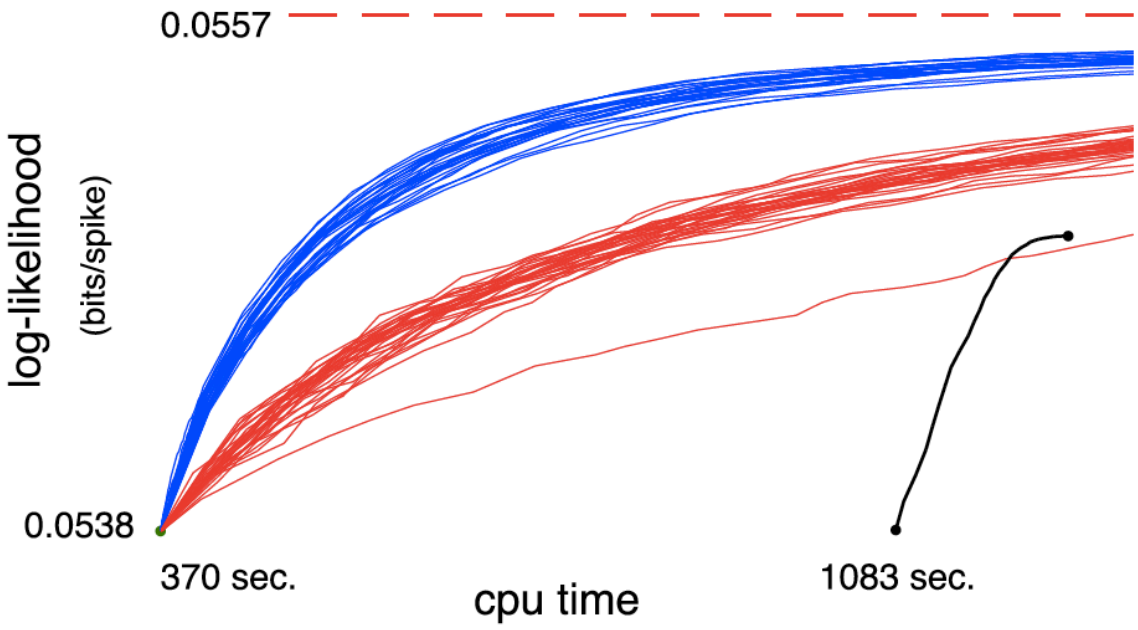

Figure 9: The log-likelihoods attained by the greedy (black), MCMC (blue), and CAST (red) methods are plotted against CPU run time, for the second, larger dataset. All times and likelihoods smaller than those obtained from the lazy greedy algorithm (green dot) are not shown; in particular, the first 1083 seconds of the greedy algorithm, during which the greedy configuration had a smaller likelihood than the lazy greedy configuration, are not shown. All the MCMC and CAST runs were initialized with the lazy greedy configuration: each run starts at the same green dot. The thick dotted red line indicates the best likelihood eventually obtained from the best of 30 CAST runs, after 36 hours.

which are of relatively poor quality, but are very fast to obtain, and which are reasonable enough to be used as the initial configuration for all of our MCMC and CAST chains.

Running naive MCMC leads to configurations with posterior likelihoods that stop increasing after a few thousand iterations: each MCMC run gets stuck in a small region of configuration space. Whenever an MCMC run likelihood did not increase for more than 400 (resp. 8500) consecutive iterations for the first (resp. second) dataset, it was reinitialized to the lazy greedy configuration. Even though naive MCMC runs get stuck in local optima, most of the resulting configurations have higher posterior likelihoods than the greedy configuration (Figs. 5, 6 and 9). This is not surprising, since in some sense, MCMC is similar to a greedy algorithm with some amount of backtracking.

Running 30 instances of CAST for 10^6 iterations each, we see that CAST consistently explores regions of cone configuration space with more cones and higher likelihoods than either the greedy or naive MCMC methods (Figs. 5 and 6). CAST runs did not need to be reinitialized, as the likelihoods involved typically increase throughout the 10^6 iterations. However, it is hard to say whether or not these chains are ergodic. All chains do not attain the same range of log-likelihoods by the end of their run; this could either be due to the chains not being ergodic, or to 10^6 not being a long enough "burn-in" period for the chain to reach

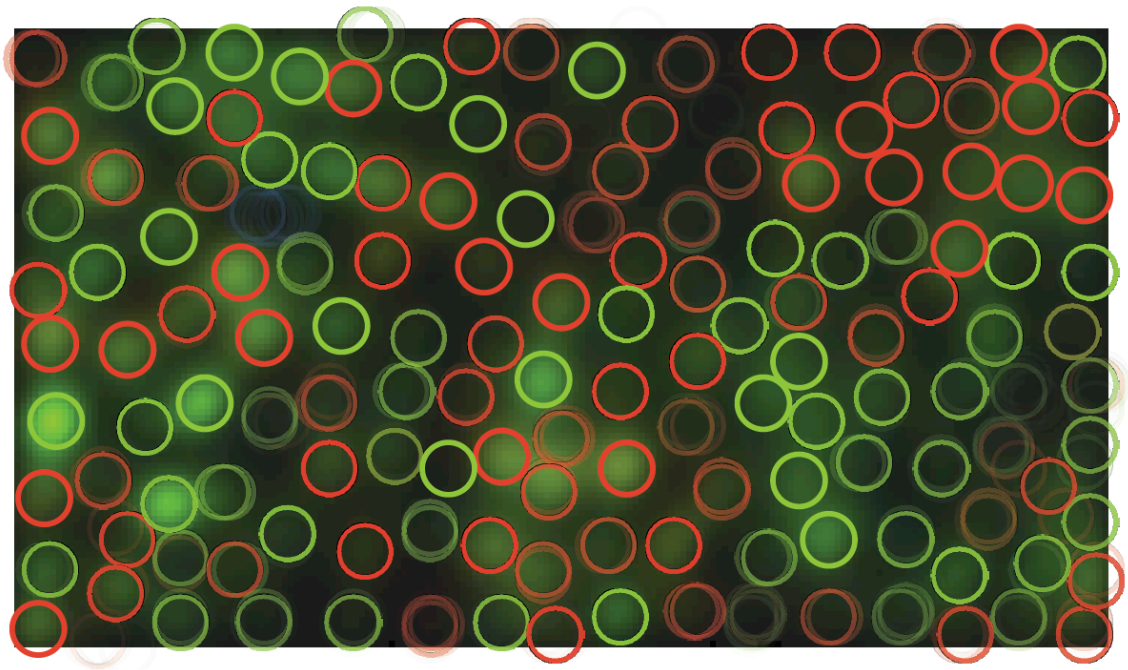

Figure 10: Ensemble of cone configurations as sampled by the best CAST chain out of 30 for the first dataset. Each depicted cone has a transparency reflecting the proportion of sample configurations where a cone was present at its location, and a color which is an average of the colors of all the cones that were placed in this location. This plot allows us to visualize the degree of confidence we can have in each cone inference.

the regime of typical configurations.

CAST runs offer a good indication of how confident we can be in each cone inference, since they are sampling from a distribution of likely cone configurations while being less prone to local optima. We can for example plot cone configurations averaged over many CAST iterations (Figs. 10 and 11). In these plots, cones have a color which is an average of the colors of cones found at a same location across samples, and an opacity which is proportional to the number of samples for which a cone was present at that location. Cones whose inferences are uncertain are either faded out by partial transparency, have a color which is a mix of green and red, or a position which is uncertain, depending on the type of uncertainty. The proportion of samples for which a cone is present at a given location gives an easily interpretable indication of how certain we are of the cone's inference.

Appliction: denoising receptive field estimates

As a simple application of the methods discussed above, we consider a basic problem: how do we construct efficient estimators of the receptive fields $\mathbf{k}_i$? In particular, can we exploit the fact that we are observing multiple ganglion cells simultaneously, to obtain better estimates than would be possible in the "classical" setting, where only one ganglion cell is observed per experiment? In the latter case, the standard approach (as discussed above) is to use the STA as an estimator for $\mathbf{k}$ (Chichilnisky, 2001; Paninski, 2004). However, in these experiments

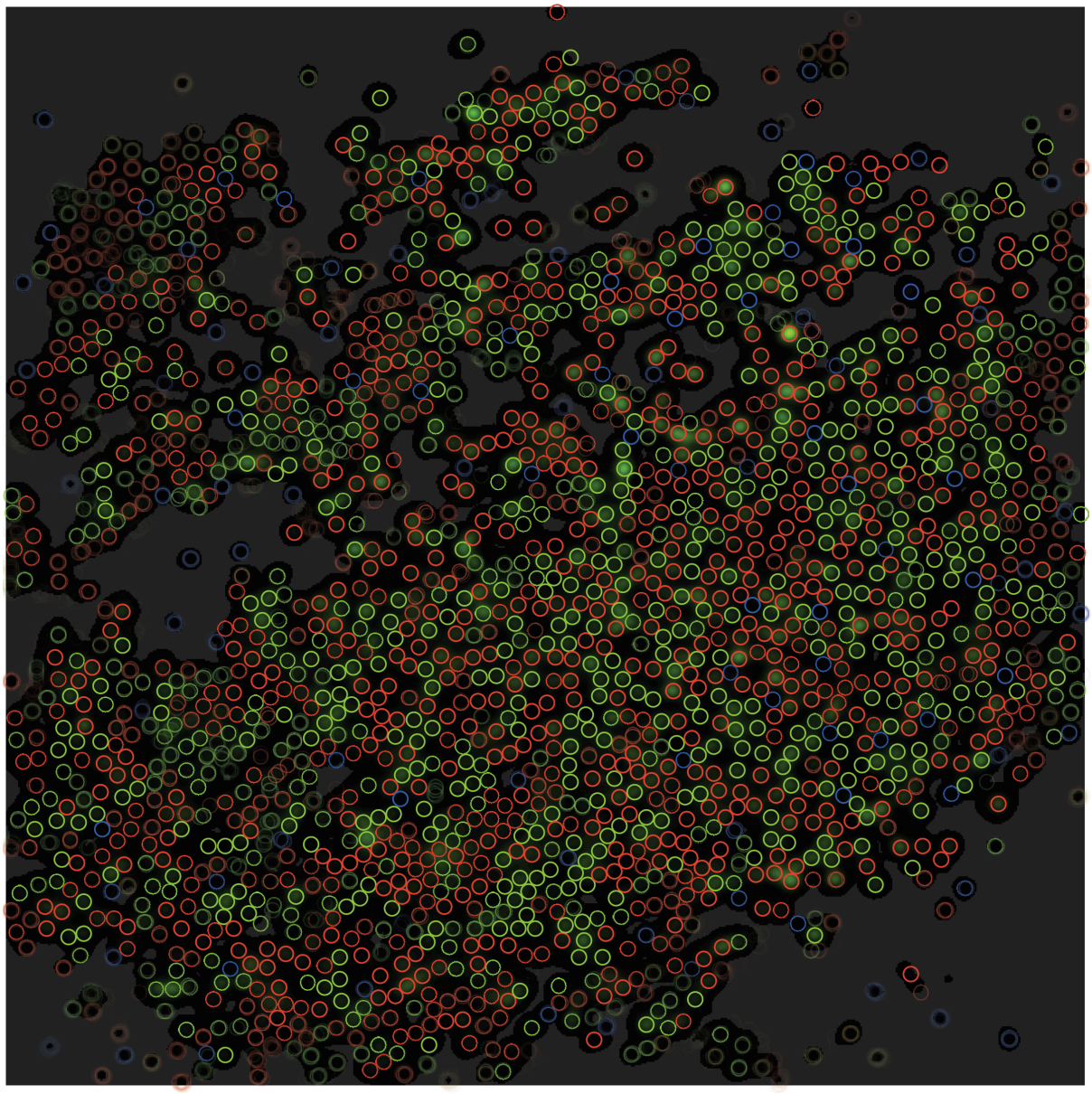

Figure 11: Ensemble of cone configurations as sampled by the best CAST chain out of 30 for the second dataset. Conventions as in Fig. 10.

STAs are highly noisy, due to the low effective contrast and the fine pixelization of the stimulus; see the left panels of Fig. 12 for some examples, and (Paninski, 2003) for further discussion of the variability of STA-based estimates.

Of course, a natural alternate approach here is to exploit the fact that all of the observed ganglion cells share a common set of cones whose configuration we have estimated. We have already discussed above how to compute the conditional expectation $\mathbb{E}\,(\,\mathbf{a}_i \,|\, data\,)$ of the receptive field weights, given the output of our MCMC samplers; now we need simply multiply these weight vectors by the corresponding cone-to-pixel matrix $\mathbf{W}$ to obtain estimates of the

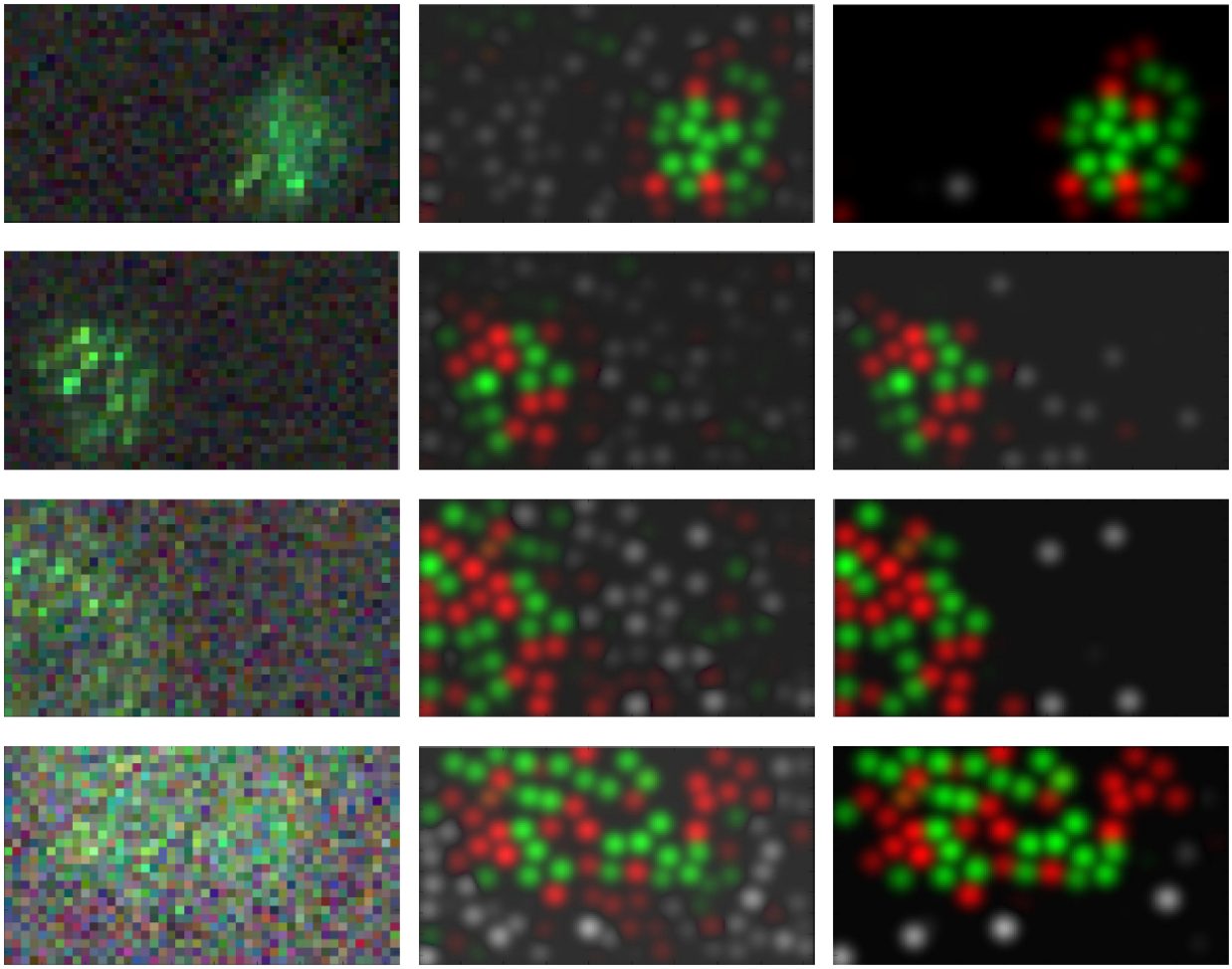

Figure 12: Denoised estimates of individual ganglion cell receptive fields. Left panels: raw STAs of four simultaneously recorded genglion cells, taken from dataset 1. Middle: posterior mean receptive fields $\mathbb{E}\left(\mathbf{W}\mathbf{a}_i|data\right)$, for each of the four cells. Colors correspond to positively-weighted cones (with higher intensity corresponding to larger inferred weights $\mathbf{a}$); grayscale corresponds to magnitude of negatively-weighted cones (i.e., the inhibitory surround). Right: thresholded estimate of the receptive fields (see text for details). Top two rows are midget cells; bottom two rows are parasol cells.

receptive fields. See the middle panels of Fig. 12 for an illustration.

In fact, we can go further: since we also have access to the posterior uncertainty about the weights $\mathbf{a}$ (through $\mathbf{Cov}\left(\mathbf{a}_i \mid data\right)$), we can threshold any elements of the weight vector about which we have insufficient posterior confidence, to further denoise the estimate. In the right panels of Fig. 12, we discarded any element of $\mathbf{a}$ for which the conditional expectation was less than two standard deviations away from zero. (Of course other approaches are possible here.) These results therefore illustrate the power both of simultaneous recording and also hierarchical modeling: by "sharing information" between simultaneously recorded ganglion cells (via our estimate of the cone layer), we obtain significantly sharper estimates of the ganglion cell parameters than would have been possible otherwise.

Conclusion: using the greedy, MCMC, and CAST algorithms

Our analyses consistently demonstrate that MCMC (and particularly CAST) methods achieve higher likelihood scores than does the greedy algorithm. Of course, in practice, it may be useful to use a suboptimal algorithm if it is significantly faster, and in particular if it leads to reasonable cone configurations fast enough for online use during an experiment.

Thus it is useful to conclude by taking another look at the typical running times of the various methods on current hardware for our larger dataset, which has a size typical of most experiments (Fig. 9). The "lazy" greedy configuration used to initialize the MCMC and CAST runs gives them a significant head start over the greedy method. The MCMC likelihoods quickly exceed the final greedy configuration's likelihood. In the process, the MCMC method quickly corrects mistakes made by the lazy greedy algorithm's crude approximation, adding cones which were missed by the lazy greedy method. This makes initializing our naive MCMC runs with the lazy greedy configuration the method of choice for fast evaluation of the cone configuration during an experiment.

By comparison, the CAST method takes much longer to increase likelihoods. However, it does not get stuck in local optima as the naive MCMC method does, and CAST configurations end up with significantly higher likelihoods than all other methods (Figs. 5 and 6). Thus a reasonable approach is to initialize with the fast lazy greedy method, then relax into a local optimal solution with MCMC, and finally to continue with CAST for as many iterations as desired to evaluate the true posterior over cone configurations for a given experiment.

Acknowledgments

This work was supported by NEI grant EY018003, an NSF CAREER award, and a McKnight Scholar award. Large-scale computations were run on the Hotfoot computational cluster at Columbia University.

Appendix / Supplemental

STA space-time separation using SVD

In a preprocessing step, the **STA** of each ganglion cell was approximated as being separable in space and time. (Small deviations from separability — specifically, center-surround latency differences — are present in these cells' receptive fields; see (Chichilnisky and Kalmar, 2002) for further details. However, the stimulus refresh in these experiments is sufficiently slow that these small inseparabilities are negligible here.) The temporal and spatial components were obtained using a singular value decomposition (SVD) approach, as in (Gauthier et al., 2009). First, the **STA** was put in a matrix $\mathbf{M}$ in which each row was the time course of a single pixel. The SVD consisted of decomposing $\mathbf{M} = \mathbf{U}\mathbf{D}\mathbf{V}^T$, where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices and $\mathbf{D}$ is diagonal with positive decreasing elements along the diagonal. The **STA** matrix $\mathbf{M}$ was approximated as being separable in space and time by setting all but the first element in $\mathbf{D}$ to zero, revealing the first column of $\mathbf{U}$ as the **STA** spatial component, and the first row of $\mathbf{V}$ as the **STA** temporal component. Finally, the spatial stimulus $\mathbf{s}_t$ discussed in the main text is defined as the projection of the full spatiotemporal stimulus onto the SVD-derived temporal component.

Some implementation details

In the actual implementation, $\mathbf{W}$ is never represented explicitly. This is because many of the calculations required for the MCMC simulation can be done in advance: the matrix $\mathbf{W}$ always appears in calculations either multiplied on the left by **STA**, or on the left by its own transpose. Since **STA**s are limited in space, their dot product with the cone receptive fields in $\mathbf{W}$ is sparse. The sparsity structure of these dot products is calculated in advance for all possible cone locations. The actual dot products are then only calculated for combinations of cone locations and **STA**s which are known to have a significant dot product (see main text for details).

We also exploit the sparsity of $\mathbf{W}^T\mathbf{W}$: this matrix contains dot products between pixelized cone receptive fields, which are non-zero only for overlapping cones. In addition, these dot products only depend on the relative differences in the positions of the two cones. We take advantage of this by only considering a finite number of possible cone locations that are regularly spaced with a frequency which is a multiple of the stimulus pixel width. With such regular spacing, the relative distances between overlapping cones can only take a few values, so the values populating $\mathbf{W}^T\mathbf{W}$ can be calculated in advance. Currently, there are 4 cone coordinate locations for each pixel width, for a total of 16 possible cone locations per pixel. This allows us to calculate in advance the 16 pixelated convolutions of the typical cone receptive field spatial component with itself, where the pixelation consists of integrating over the square surface of each pixel as explained in the main text.

Another implementation detail which is important for speed is to keep track of which cones are touching which others in the four cardinal directions. Indeed, whenever a cone shift move is proposed, we need to know which neighboring cones to shift over with it in order to respect cone exclusion: it is more efficient to calculate the four neighborhood relations only once when an MCMC move is accepted rather than calculating them each time a new move is proposed, and these relations can be kept track of incrementally.

The code is available upon request.

References

Atchadé, Y. and Liu, J. (2010). The wang-landau algorithm in general state spaces: applications and convergence analysis. *Statistica Sinica*, 20(1):209.

Casella, G. and Berger, R. (2001). *Statistical Inference*. Duxbury Press.

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

Chichilnisky, E. and Baylor, D. (1999). Receptive-field microstructure of blue-yellow ganglion cells in primate retina. *Nature Neuroscience*, 2:889–893.

Chichilnisky, E. and Kalmar, R. (2002). Functional asymmetries in on and off ganglion cells of primate retina. *J Neurosci*, 22(7):2737–47.

Diaconis, P. and Freedman, D. (1984). Asymptotics of graphical projection pursuit. *Annals of Statistics*, 12:793–815.

Earl, D. and Deem, M. (2005). Parallel tempering: Theory, applications, and new perspectives. *Physical Chemistry Chemical Physics*, 7(23):3910–3916.

Field, G., Gauthier, J., Sher, A., Greschner, M., Machado, T., Jepson, L., Shlens, J., Gunning, D., Mathieson, K., Dabrowski, W., Paninski, L., Litke, A., and Chichilnisky, E. (2010). Functional connectivity in the retina at the resolution of photoreceptors. *Nature*, 467(7316):673–677. 10.1038/nature09424.

Frechette, E., Sher, A., Grivich, M., Petrusca, D., Litke, A., and Chichilnisky, E. (2005). Fidelity of the ensemble code for visual motion in the primate retina. *J Neurophysiol*, 94(1):119–135.

Gauthier, J. L., Field, G. D., Sher, A., Greschner, M., Shlens, J., Litke, A. M., and Chichilnisky, E. J. (2009). Receptive fields in primate retina are coordinated to sample visual space more uniformly. *PLoS Biol*, 7(4):e1000063.

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

Lidl, R. and Pilz, G. (1997). *Applied Abstract Algebra*. Springer.

Litke, A., Bezayiff, N., Chichilnisky, E., Cunningham, W., Dabrowski, W., Grillo, A., Grivich, M., Grybos, P., Hottowy, P., Kachiguine, S., Kalmar, R., Mathieson, K., Petrusca, D., Rahman, M., and Sher, A. (2003). What does the eye tell the brain?: Development of a system for the large scale recording of retinal output activity. *IEEE Transactions on Nuclear Science*, 51(4):1434 – 1440.

Marinari, E. and Parisi, G. (1992). Simulated tempering: a new monte carlo scheme. *EPL (Europhysics Letters)*, 19:451.

Paninski, L. (2003). Convergence properties of some spike-triggered analysis techniques. *Network: Computation in Neural Systems*, 14:437–464.

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

Paninski, L., Pillow, J., and Simoncelli, E. (2004). Maximum likelihood estimation of a stochastic integrate-and-fire neural model. *Neural Computation*, 16:2533–2561.

Park, I. and Pillow, J. (2011). Bayesian spike-triggered covariance analysis. *Advances in Neural Information Processing Systems*, 24.

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

Pillow, J. W., Shlens, J., Paninski, L., Sher, A., Litke, A. M., Chichilnisky, E. J., and Simoncelli, E. P. (2008). Spatio-temporal correlations and visual signalling in a complete neuronal population. *Nature*, 454(7207):995–999.

Ramirez, A. and Paninski, L. (2012). Fast inference in generalized linear models via expected log-likelihoods. *In preparation*.

Robert, C. and Casella, G. (2005). *Monte Carlo Statistical Methods*. Springer.

Roy, B. (1959). Transitivité et connexité. *C. R. Acad. Sci. Paris*, 249:216–218.

Salakhutdinov, R. (2010). Learning deep Boltzmann machines using adaptive MCMC. In *Proceedings of the International Conference on Machine Learning*, volume 27.

Segev, R., Goodhouse, J., Puchalla, J., and Berry, M. (2004). Recording spikes from a large fraction of the ganglion cells in a retinal patch. *Nature Neuroscience*, 7:1154–1161.

Wang, F. and Landau, D. (2001). Efficient, multiple-range random walk algorithm to calculate the density of states. *Physical Review Letters*, 86(10):2050–2053.

Warshall, S. (1962). A Theorem on Boolean Matrices. *J. ACM*, 9:11–12.