SI Appendix

A. Computing instantaneous firing rates for Eq. 2 and 3 (Kass and Ventura method)

B. Comparing behavior and decoder performance (Wichmann and Hill method)

C. Decoding using a factorized joint conditional response distribution

D. Calculating ganglion cell number and decoding assuming different distributions of ganglion cell classes

E. Measuring the effects of data limitation on code performance

F. Measuring stimulus intensity, chromatic aberration

G. Visualizing performance growth as a function of cell number in finer detail

H. Information carried by spike count code for all cells in the data set, assessing sampling bias

I. Decoding with the spike count code using responses shorter than the complete stimulus period (i.e., 50 ms and 100 ms, rather than the complete 300 ms.)

A. Computing instantaneous firing rates for Eq. 2 and 3 (Kass and Ventura method)

Here we describe how we compute the firing rates in Eqs. 2 and 3 of the main text. We also describe the terms in the equations in more detail. (For convenience, we use the same numbering (i.e., Eq. 2, Eq. 3) as we used in the main text, since the equations are exactly the same).

We start with Eq. 2. For the spike timing code (also referred to as an instantaneous rate code), the probability of a response given a stimulus follows that of an inhomogeneous Poisson process (1). Letting r be a list of spike times at resolution dt, denoted t_{ijl} where t_{ijl} is the jth spike on the lth look of cell i, this probability is given by

$$p_{\text{spike timing code}}(r \mid s) = \prod_l \left[\prod_{ij} \nu_i(t_{ijl} \mid s)dt \right] \exp\left[-\sum_i \int_0^T dt \nu_i(t \mid s) \right] \tag{2}$$

where $\nu_i(t|s)$ is the firing rate of cell i at time t, given that stimulus s was presented, and the upper and lower limits (0 and T) correspond to the start and end of each trial. Equation 2 is obtained in the standard way, as follows: Time is discretized into bins of size dt with dt small enough to ensure that that there is at most one spike per bin. One then proceeds though the bins and writes down the probability of observing a spike or "no-spike" in each bin. If there is a spike, the probability is $\nu_i(t|s)dt$; if there is no spike

(i.e., the bin is empty), the probability is $(1-v_i(t|s)dt)$. Multiplying the terms together for all bins and taking the small *dt* limit gives us Eq. 2. (In the small *dt* limit, the product of all the terms with $(1-v_i(t|s)dt)$ results in the exponential term in Eq. 2.)

For the temporal correlation code, again *r* is a list of spike times at resolution *dt*, denoted t_{ijl}, but this time the firing rate has an additional dependence on the time of the previous spike on the same spike train,

$$p_{\text{temporal correlation code}}(r \mid s) = \prod_l \left[\prod_{ij} v_i (t_{ijl}, \tau(t_{ijl}) \mid s) \right] \exp\left[-\sum_i \int_0^T dt v_i (t, \tau(t) \mid s) \right] \qquad (3)$$

where $\tau(t)$ is the time interval between *t* and the spike that preceded *t* on the same neuron (e.g., $\tau(t_{ijl}) = t_{ijl} - t_{i,j-1,l}$).

We now turn to the procedure for finding $v(t|s)$ and $v(t,\tau(t)|s)$ from spike train data. For $v(t|s)$, we followed the method of Kass and Ventura (2-4). Briefly, we presented the stimulus repeatedly (the main text shows results with different numbers of repeats, Fig. 3d), binned the responses at 1 ms, and on each trial determined whether or not there was a spike in each of the bins. Based on this trial-by-trial data, we parameterized $v(t|s)$ with cubic splines, and used maximum likelihood to estimate the spline parameters. We used cubic splines because they provide a way to accurately capture firing rate changes, including sharp transitions, but require a small number of parameters (3). The ratio of spikes to parameters was large: the mean number of knots (and thus the mean number of parameters) averaged over our data set was 4.6, with a maximum of 12 and a minimum of 3. Given that we had, on average, 210 spikes/trial, this means we had ~46 spikes per parameter.

To find the firing rate conditioned on previous spikes (as well as stimulus), $v(t,\tau(t)|s)$, we again presented the stimulus repeatedly and determined, on each trial, whether or not there was a spike in each 1 ms bin. We then wrote the firing rate as the product of two terms: $v(t,\tau(t)|s)=v_1(t|s)v_2(\tau(t)|s)$ (1,2,5,6). Both terms were parameterized, based on the trial-by-trial data, using cubic splines (2, 3) and, again, maximum likelihood was used to estimate the spline parameters. We used 4 knots for the second term, and the same number (but not placement) of knots for the first. Thus, for the correlation code, we had approximately 24 (210/(4.6+4)) spikes per parameter, again, a large spike to parameter ratio.

2

B. Comparing behavior and decoder performance (Wichmann and Hill method)

As mentioned in the main text, we compared behavior and decoder performance using Psignifit (version 2.5.6), a software package that implements Wichmann and Hill's maximum-likelihood method (7,8). The procedure was to first measure animal performance. Each animal performed many trials of the task, with the spatial frequency varying from trial to trial. For each animal, we computed the fraction of trials on which it made the correct choice for each spatial frequency. We then measured the Bayesian decoder's performance. Like the animal, the decoder performed many trials of the task, and the fraction of trials on which it made the correct choice was computed for each spatial frequency. The two performance curves, the one produced by the animal and the one produced by the Bayesian decoder, were then compared using the Psignifit program: Each curve was fit with a logistic function of spatial frequency using two parameters, threshold and slope. Confidence intervals were found by the percentile bootstrap technique, which is implemented by the program, based on 1000 simulations. Once the functions were fit, they were compared using the Monte Carlo method implemented by pfcmp, part of the psignifit package (also described by Wichmann and Hill (7, 8, 9). Briefly, Monte Carlo simulations with a large number of runs (10000 or more) were done to generate 2D distributions of the differences in the threshold and slope parameters for the fits to the two datasets separately and jointly, then the observed differences were tested against the null hypothesis that the two data sets, in fact, came from the same underlying function. If the *p*-value for the null hypothesis was less than 0.05, the function produced by the animal and that produced by the decoder were designated as different, otherwise, they were designated as the same.

C. Decoding using a factorized joint conditional response distribution

As mentioned in the main text, we find $p(r|s)$ in Eq. 1-3 by factorizing it, that is, we compute it from the marginals. The justification for this comes from several different groups (10-14) that show that factorizing $p(r|s)$ has little or no effect on the estimation of $p(s|r)$ for all r that occur (measured under conditions where the true joint distribution can be obtained, i.e., using pairs of neurons). Here we addressed this issue directly using the stimuli in our task. Using pairs of cells and the spike count code, we measured

3

performance under two conditions: when the factorized conditional response distribution was used, and when the true joint conditional response distribution was used. The results show that there was no statistically significant difference in performance ($p > 0.1$, paired t-test, $n=426$ pairs of cells) (Fig. S1).

Note that when one factorizes $p(r|s)$, what one is eliminating is what is commonly referred to as "noise correlations" (10, 15-17). To provide intuition for what noise correlations are, we give an example, adapted from ref. 10: Suppose we have two ON cells with overlapping receptive fields, and we present a flash of light. The two cells will fire in a correlated way because they both see the light at the same time. These correlations are called "signal correlations", and we do not in any way eliminate these. The two cells will also show some correlation beyond this because they have circuitry in common (common photoreceptors, common amacrine cells, etc.). These additional correlations are the noise correlations (10, 15-17). For most pairs of cells in the mouse retina, noise correlations constitute <5% of the total spikes the cells produce (10).

Given that the degree of noise correlation in the mouse retina is small (10), and that several studies, as well as the data in Fig. S1 show that factorizing to construct $p(r|s)$ has very little impact on decoding (the studies include both mouse retina and several other brain areas (10-14)), the factorized distribution seems a reasonable approximation; the probability that our conclusions about code performance would be overturned is small.
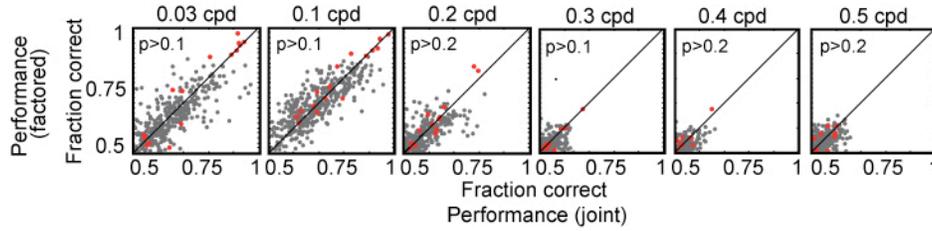
Fig. S1. Decoding using the factorized versus the true joint conditional response distribution. Performance was measured, as in all figures, as the fraction correct on the standard grating task. The analysis was performed on the retinas with the highest degree of correlation in the dataset: the red dots indicate the most correlated pairs (those with >5% correlation). The highest correlation was 33%, consistent with (10). No statistically significant difference in performance was observed (p>0.1, paired t-test, n=426 pairs). *Methods*: Using 60 (out of 120) trials, we constructed $p(r_1,r_2|s)$, $p(r_1|s)$ and $p(r_2|s)$ where r_1 and r_2 are spike counts from neurons 1 and 2, and s is the stimulus (either a grating or a gray screen). Using the remaining 60 trials, we ran our task: we presented two stimuli, one of which was a gray screen and the other a grating, and used either $p(r_1,r_2|s)$, or $p(r_1|s)p(r_2|s)$ to estimate which was which. What is plotted in the figure is the fraction correct using these distributions. Note that in some cases, the fraction correct for the factorized distribution exceeded that of the true. This is consistent with the expected scatter that occurs with cross validation. Using binomial statistics, we estimate the scatter as follows: The standard deviation in fraction correct is $[np(1-p)]^{1/2}/n = [p(1-p)/n]^{1/2}$ where p is the fraction correct and n is the number of trials. Using p=0.75 (halfway between chance and 100% correct), the standard deviation is $[(0.75)(0.25)/(60)]^{1/2} = 0.06$, consistent with the size of the scatter in the figure. Note that with 426 pairs of cells, the SEM is $0.06/426^{1/2} = 0.2\%$. Thus, the fact that we found no significant difference between the fraction correct for the factorized and true distributions implies that the two gave the same fraction correct to within <1%.

D. Calculating ganglion cell number and decoding assuming different distributions of ganglion cell classes

Calculating ganglion cell number : Two electron microscopic studies of ganglion cell axon number report 44,860 (18) and 54,600 (19) respectively. As reported in ref. 21, retina area varies as a function of age, such that area in mm^2=12.5+3.3log (age in days). The same reference indicates a 6% shrinkage due to fixation. Animals in our dataset were ~8 months=240 days old, giving an area of [12.5+3.3log(240)]/0.94=21.6 mm^2. The number of ganglion cells viewing a stimulus of size 0.144 mm^2 then ranges from (44,860/21.6)×0.144=299 cells to (54,600/21.6)×0.144=364 cells. With rounding, this gives 300 and 360 cells, respectively. The total dataset collected was 423 cells. To measure performance for 300 and 360 cells, the dataset was sampled 20 times, then averaged. Likewise, in Fig. 3a, main text, where performance was measured for 1, 2, 4, 8, 16, 32, 64, 128, 256, 300, and 360 cells, the dataset was sampled 20 times, then averaged.

<u>Decoding assuming different distributions of ganglion cell classes:</u> Mouse retinal ganglion cells divide into 5 classes (5 statistically significantly distinct clusters (18). As shown in Fig. S2, they first divide into short and long latency groups. The short latency cells divide into ON, OFF and ON-OFF cells, and the ON cells divide into ON sustained and ON transient cells. These classes are consistent with other reports on mouse retina both *in vitro* (22, 23) and *in vivo* (24, 25) using the same or very similar stimuli.

While the characterization of the classes is consistent across studies, the distribution is less so. Physiological reports suggest that it is skewed toward cells with ON responses, with the degree of skewness varying among studies (21, 22, 24, 25). Anatomical reports, though, suggest a more even representation, specifically, a more even representation of ON and OFF-type cells (26). The latter is based on the relative density of the projections into the ON and OFF sublamina in the plexiform layer, which reflects the relative proportions of cells with ON and OFF responses (26).

In the main text, we addressed the uncertainty about cell class distribution by measuring code performance using both estimates. For the physiological representation, we took the raw distribution as we recorded it, which, consistent with previous reports, was skewed toward cells with ON responses. The ratio of cell classes was as follows: ON-OFF, ON Transient, ON sustained, OFF, Long Latency: 19:17:11:5:1. For the anatomical estimate, we assumed a uniform representation. While the choice of distribution shifted the performances slightly, the conclusions remained the same (Fig. 3a, main text): the spike count and spike timing codes performed worse than the animal, while the temporal correlation code did not (spike count code, $p \ll 0.0001$; spike timing code, $p < 0.02$; temporal correlation code, $p > 0.3$).

In this section, we took the analysis a step further. We performed a set of extreme experiments, in which we considered distributions that were so skewed that they contained only one cell class at a time. This way, if any cell class in the population were able to perform the task better than the others, its performance would be exaggerated. We then asked whether, given these single class distributions, the spike count and spike timing codes would still perform worse than the animal. The results, along with the methods, are shown in Fig. S2b. For all distributions, both the spike count code (Fig. S2b, top) and the spike timing code (Fig. S2b, bottom) performed worse than the animal ($p < 0.02$ for all performance curves). Note that the performances of the cell classes clearly

differed from each other. The classes that contain ON responses – the ON sustained, ON transient, and ON-OFF cells – were clearly the better performers. (The differences in performance are not evident at the level of the spike count code, but emerge at the level of the spike timing code; see Fig. S2b, bottom). We thus pushed the analysis further and added three new experiments to increase the sample size of the higher performing populations. The results remained the same (Fig. S2c), suggesting the starting sample size was sufficient.

In sum, these data provide strong evidence that the conclusions of this paper – that the spike count and spike timing (instantaneous rate) codes perform worse than the animal – are robust to errors in the estimation of the ganglion cell distribution. Even when performance is pushed by drawing only from the highest performing cell classes, the spike count and spike timing codes still fall short of the animal. The only conditions under which they were able to reach animal performance was with the spike timing code, on 10% of the draws, using a ganglion cell population made up exclusively of ON sustained cells, which is an extreme, and biologically unrealistic, scenario (see Fig. S3 for the individual draws).

For further insight into the performances of the different cell classes, see Fig. S4, where we show the performance curves for all cells in the dataset, sorted by cell class, and Fig. S5, where we provide a set of rasters that show how the performance curves arise from the raw data.
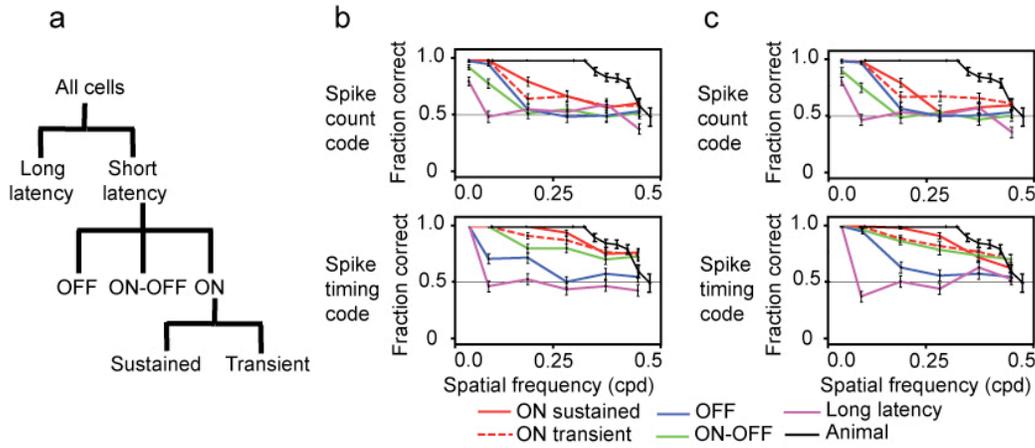
Fig. S2. Code performance assuming different distributions of ganglion cell classes. **a.** Ganglion cell classes in the mouse retina. The cells fall naturally into 5 statistically significantly distinct clusters, see (21). They first divide into short and long latency classes. The short latency cells then divide into ON, OFF and ON-OFF groups, and the ON cells divide into transient and sustained classes. **b.** Code performance, measured for each cell class separately. ***Top***, spike count code. ***Bottom***, spike timing code. Code performance (colored traces) was worse than animal performance (black trace) for all cell classes (*p<0.05* for all curves). The sample size for each class was as follows: ON sustained, 42 cells; ON transient, 134 cells; OFF, 90 cells; ON-OFF, 149 cells; Long latency, 8 cells. To match the number of cells the animal uses to perform the task (i.e., 300) we "cloned" the cells; that is, we made use of the fact that each cell sees each stimulus 40 times at 3 different phases. Thus, each cell produces 120 responses to each stimulus. This means that there are several hundred to several thousand responses to draw from for each cell class. The range is from 960 (120 × 8 cells) for the least represented cell class (the Long latency cells) to 17,880 (120 x 149 cells) for the most represented cell class (the ON-OFF cells). **c.** Performance on the task, measured for each cell class separately, using a larger dataset. Three new experiments were added to increase the representation of the cells with ON responses, as these appear to be the higher performing cell classes. The ON sustained population increased to 55, the ON transient to 136, and the ON-OFF to 207.
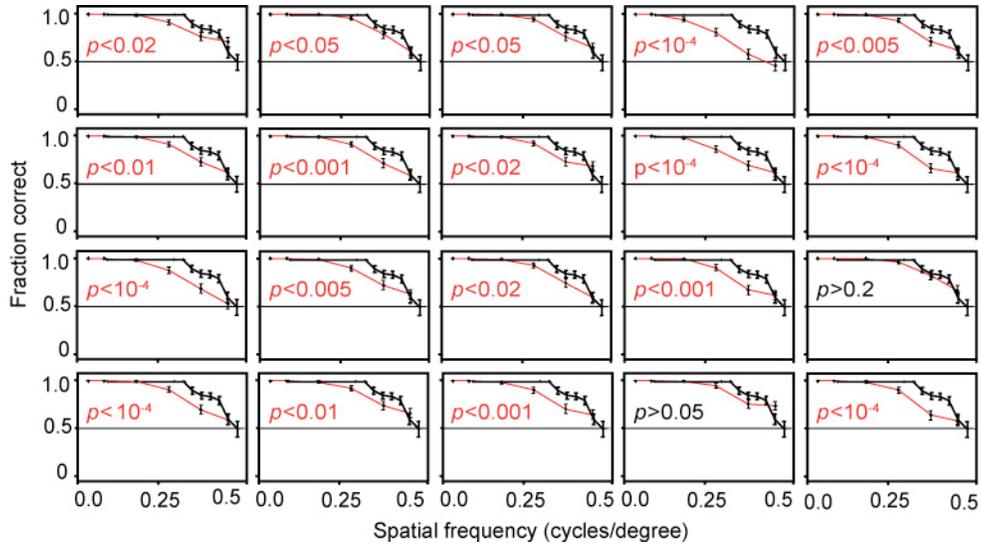
Fig. S3. Code performance for each draw from the highest performing cell class. The highest performing cells were the ON sustained cells, as shown in Fig. S2b and c, bottom panels (i.e., performance with the spike timing code). Each trace in Fig. S2 is the average of 20 draws. The panels here show each draw individually. On only two occasions was performance close to that of the animal. Thus, the only condition under which spike timing code performance reached animal performance was on 10% of the draws, using a ganglion cell population made up exclusively of ON sustained cells, an unrealistic condition.
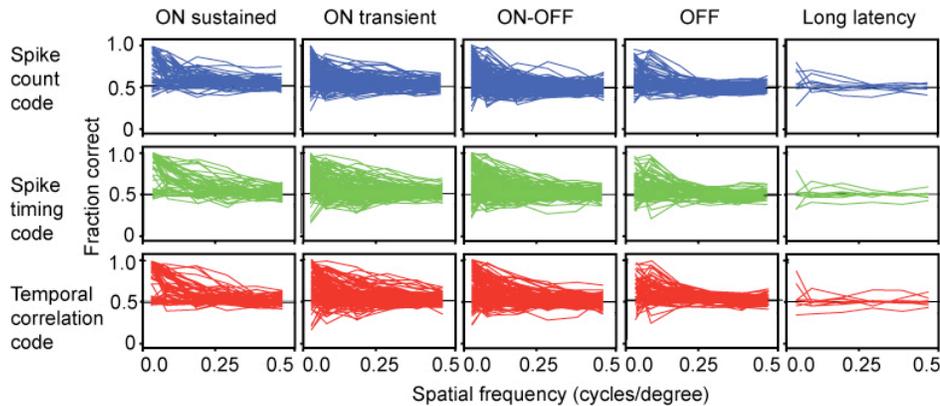

Fig. S4. Performance curves for each cell in the dataset, sorted by cell class. The columns indicate the cell classes; the rows indicate the code. Each trace is the curve for a single cell. Note that for the cells with ON responses (the ON sustained, ON transient and ON-OFF cells), there is an increase in performance (a shift to the right) with progression from simple to complex codes (from blue to green to red). This is not the case for the OFF and Long latency cells.
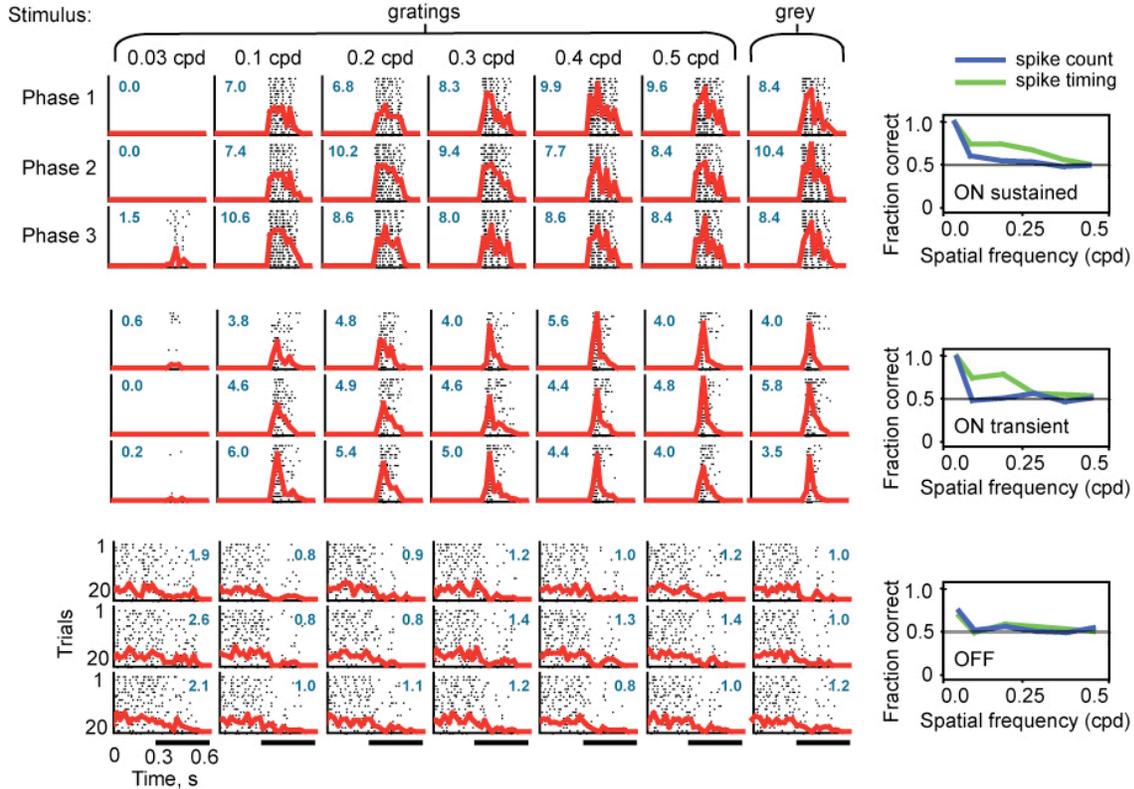
9

Fig. S5 A complete set of raster plots for three cells; the purpose is to show that the differences in the performances of the codes can be seen at the raw spike train level. Each set of 21 rasters corresponds to 1 cell. On each raster is a number and a red trace. The number is the mean number of spikes per trial for the raster and gives a measure of spike count, and the red trace is the peri-stimulus time histogram, and gives a measure of spike pattern. To obtain a feel for how a given performance curve arises (see blue and green performance curves in the plot at the right), one can compare the mean spike counts for the gray stimulus with the mean spike counts for each of the spatial frequencies, and then compare the mean spike patterns for the gray stimulus with the mean spike patterns for each of the spatial frequencies. For example, for the first cell, the mean spike counts for the three rasters at the lowest spatial frequency, 0, 0, and 1.5, are very different from those for the gray stimulus, 8, 10, and 8. As the spatial frequency gets higher, though, the mean spike counts for the gratings become less and less different from those for the gray. In contrast, the spike patterns for the gratings remain visibly different out to much higher spatial frequencies. Thus, where spike count starts to fail as a useful parameter for distinguishing among stimuli (blue trace at the right), spike pattern continues to perform (green trace at the right). _Methods_: To remind the reader, the stimuli are flashed. Between flashes the screen is dark; thus, each stimulus produces a step increase in mean illumination. Note that the phases are a small fixed distance apart (50 microns) to mimic normal displacements due to eye movements (rather than equally spaced phases).

E. Measuring the effects of data limitation on code performance

As mentioned in the text, it is difficult to determine a priori whether increasing the amount of data would lead to an increase or a decrease in code performance. Intuitively, one would expect that increasing the amount of data would lead only to an increase in performance. The intuition is that little data produces poor estimates for the stimulus-conditioned response distributions. As the amount of data increases, the estimates improve, and, therefore, the ability to assign new responses to their correct distributions improves. Performance on the task, thus, goes up.

There are, though, circumstances that lead to a departure from this idealized situation, the most obvious one being the inability to completely separate the "training" and "testing" data sets. The training set is the set of trials used to build the stimulus-conditioned response distribution, and the testing set is the set of trials used to test code performance. If the two are at all correlated (e.g., due to experimental drift or just to chance with finite data sets), then increasing the amount of data can produce the opposite effect – it can cause a decrease in performance on the task. This can occur because correlations in the training and testing sets generate an upward bias in performance related to the upward bias observed when measuring information with small data samples (see ref. 27). This bias is reduced when the sample size increases. Because of the difficulty of completely eliminating correlations in the training and testing sets, we chose to address the issue of data-limitation empirically, as shown in Fig. 3d in main text.

F. Measuring stimulus intensity, evaluating chromatic aberration

Calculating stimulus intensity in vitro versus in vivo:

Suppose we have a light source that is a distance d from a lens. The lens, which has a collecting area denoted A_{lens}, focuses the light from the source onto a retina, forming an image of size A_{retina}. If the source has size A and luminance L (in cd/area = lumens/area/steradian), then the flux at the retina, in lumens/area, denoted F_{retina}, is given by $F_{retina} = L \times A / d^2 \times A_{lens} \times \tau / A_{retina}$, where τ is the transmissivity (the fraction of light that makes it from the lens to the retina). This equation can be understood as follows: $L \times A$ is the total luminance of the object, in lumens/steradian. Dividing that by d^2 gives us the flux at the lens in lumens/area. The factor of A_{lens} tells us how much light passes

through the lens, the additional factor of τ tells us how much light gets to the retina, and, finally, the factor of $1/A_{\text{retina}}$ tells us the flux of light at the retina, in lumens/area. This equation simplifies somewhat if we note that A/d^2 is the solid angle of the image made by an observer at the lens. Using Ω to denote the solid angle, $\Omega = A/d^2$, we have $F_{\text{retina}} = L \times \Omega \times A_{\text{lens}} \times \tau / A_{\text{retina}}$.

Of interest to us is the ratio of the flux at the retina *in vitro* to that *in vivo*. Let us use a prime to denote *in vitro* quantities and a double prime to denote *in vivo* quantities (e.g. L' is the luminance *in vitro* and L'' is the luminance *in vivo*). Then, the ratio of the flux at the retina *in vitro* to that *in vivo*, denoted ρ, is given by $\rho = (L' \times \Omega' \times A_{\text{lens}}' \times \tau'/A_{\text{retina}}')/(L'' \times \Omega'' \times A_{\text{lens}}'' \times \tau''/ A_{\text{retina}}'')$.

For the *in vitro* condition, the following were the values: L', the intensity at the monitor, was 8.1 cd/m^2. The image at the monitor, which had an area, A', of 1728 mm^2 was projected a distance, d', of 1100 mm to a lens. The solid angle the image subtended was, thus, $A'/d'^2 = \Omega' = 0.0014$ steradians at the lens. The collecting area of the lens, A_{lens}', was 87 mm^2, and its transmissivity, τ', was assumed to be 1, since intensity loss in modern objectives is negligible (25). The image on the retina, A_{retina}', was 0.144 mm^2.

For the *in vivo* condition, the values were as follows: L'', the intensity at the monitor, was 21 cd/m^2. The image at the monitor, which had an area, A'', of 9669 mm^2 was projected a distance, d'', of 460 mm to the animal's eye. The solid angle subtended was 0.046 steradians at the eye. The collecting area of the eye, i.e., the pupil, A_{lens}'', was measured as 0.8 mm^2, which is in close agreement with ref. 29 and slightly larger than that reported in ref. 30 (see footnote[1]), and the transmissivity, τ'' was estimated as 0.7, following directly from (31, 32). The image on the retina, A_{retina}'', was 0.144 mm^2.

Inserting these numbers into the expression for ρ, gives a value of 1.82 (6.85 lumens/mm^2 for the *in vitro* condition and 3.76 lumens/mm^2 for the *in vivo*).

Note that the intensity for the *in vivo* condition requires two estimations – one for pupil size and one for transmissivity – this, however, has little impact on the conclusions of the paper. This is because behavioral performance on acuity tasks is stable over a large

---

[1] Ref. 30 uses a ganzfeld. To compare our measurement to theirs, we computed the luminance of our source if it were spread out over a ganzfeld. This is just a geometric factor: the luminance is reduced by a factor of Ω''/π where π is the solid angle of the ganzfeld (29,31). Thus, the luminance of an effective ganzfeld, denoted L_g, is $L_g = \Omega''/\pi \times L''$.

range of intensities. While behavioral performance is known to change as one moves from scotopic to photopic light levels, it is very stable within the photopic regime (for review, see refs. 33 and 34), which is the regime we are working in. Specifically, for mice (33), behavioral performance on acuity tasks does not change over an intensity range of several orders of magnitude ($10^{-2.7}$cd/m^2 to $10^{+1.8}$ cd/m^2), a range that spans the intensities we used and substantially absorbs the small difference in the intensities in the two conditions, which is less than a factor of 2.

Photoreceptor-equivalent photons:

To determine the number of photoreceptor-equivalent photons, we need to convert from lumens at the retina to photon flux. As discussed above, the former is given by $L'' \times \Omega'' \times A''_{lens} \times \tau''/ A''_{retina} = 21 \times 0.046 \times 0.8 \times 0.7/0.144 = 3.76$ lumens/mm^2. To convert this to photon flux, we need the absorption spectra of the cone pigments in the mouse retina. There are two cone pigments in this species, the M and the UV, with absorption spectra that peak at 508 and 360 nm, respectively (31). The UV pigment is not significantly stimulated by the monitors in this paper, which have a short wavelength cutoff of 400 nm; thus, the relevant cone is the M-cone. The mouse M-cone has an absorption spectrum very close to that of the mouse rod (M-cone peak absorption is 508 nm versus 507 nm for the rod (31), we can thus estimate M-cone equivalent photons μm^{-2} s^{-1} using the factor that converts from lumens to photon flux for rods, which is 1500, as given in ref. 31. Multiplying 3.67 lumens/mm^2 by 1500 gives a value of 5505 photoreceptor-equivalent-photons μm^{-2} s^{-1}, consistent with reports of rod-equivalent photons μm^{-2} s^{-1}under similar conditions (35).

Taking into account chromatic aberration:

With respect to chromatic aberration, there are two potential issues. One is the fact that image size is affected by wavelength. This is relevant to image presentation *in vivo*. A grating presented will be magnified or reduced depending on the wavelength of the light used to present it. As mentioned in the main text, we determined image size on the animal's retina following Remtulla and Hallet (1985) (36). This reference reports 31 μ/degree for mouse for wavelengths from 544-550 nm. The same reference also shows measurements for other wavelengths, specifically, 488 nm to 655 nm. At 488, image size

is 2.4% greater; at 655 nm, it is 2.6% smaller. This means that our choice of 31 μ/degree has a maximum error of ~2.5 % in each direction. This error is too small to affect our conclusions about neural codes. To see this, one can turn to Fig 2 in the main text. It would take a shift of about 25%, not 2.5%, to overturn the results for the spike timing code, and, for the spike count code, the shift would have to be much greater - several fold. Thus, the results very comfortably clear any errors introduced by chromatic aberration.

The second issue has to do with contrast. Because of the wavelength dependence on magnification, when the animal views a grating in black and white, it is receiving several gratings superimposed (the slightly magnified and reduced images that occur from the different light wavelengths). This reduces the effective contrast of the image on the retina in *the vivo* case. Moreover, because the effective focal length of the lens is also wavelength-dependent, not all wavelengths will be in perfect focus. This too will reduce effective contrast. Since these problems do not occur *in vitro*, the retina *in vitro* has the advantage; this only strengthens, rather than weakens, our conclusions.

G. Performance of the spike count code as a function of cell number, presented spatial frequency by spatial frequency.

In Fig. 3a, Column 1 we show performance of the spike count code as a function of number of cells. Here we show the same result, presented spatial frequency by spatial frequency (Fig S6).
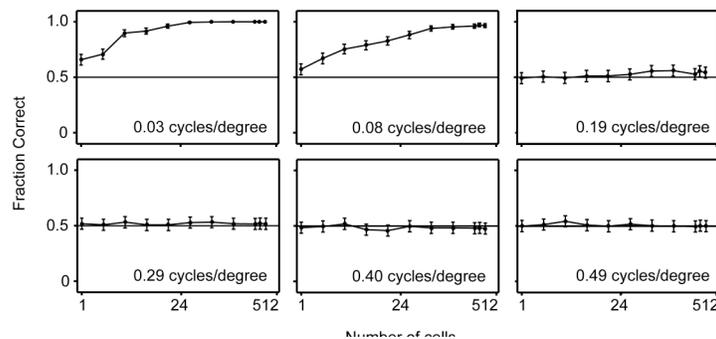

Fig S6. Performance of the spike count code as a function of cell number, shown spatial frequency by spatial frequency.

14

H. Information carried by spike count code for all cells in the data set

Fig. S7 shows the information carried by the spike count code for all the cells in the data set, sorted by retina. This is presented to reveal retina-to-retina variation. To test whether this introduces errors in the conclusions, we measured code performance after systematically eliminating the retinas with the lowest performing cells. (This amounts to enriching for the higher performing cells.) Because performance is so slowly growing when the number of cells is at 300, this has very little effect. Conclusions about the spike count code remain the same, that is, the spike count code still fails ($p<0.0001$) (Fig. S8).
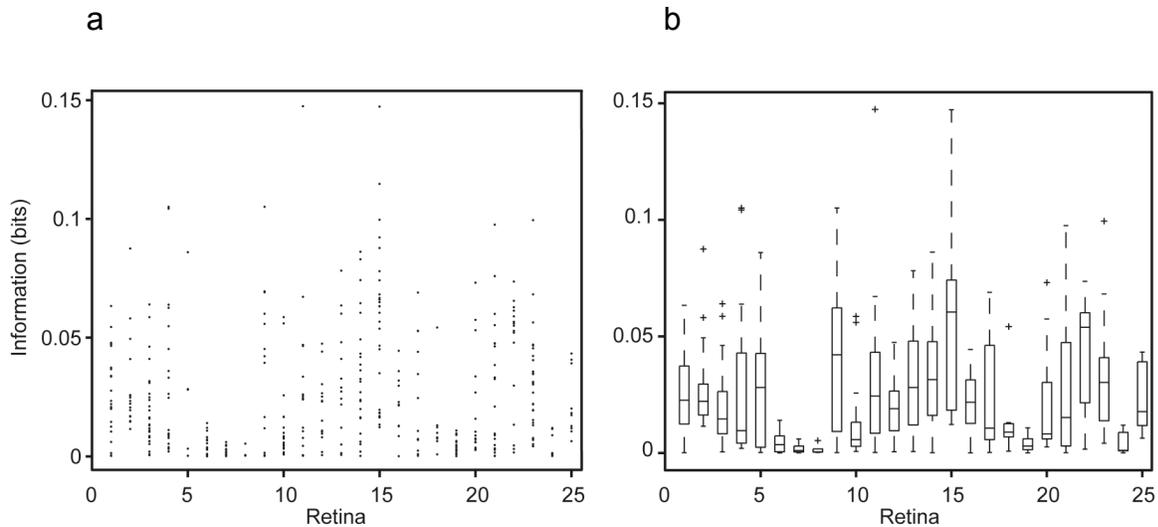

Fig. S7. Information carried by the spike count code for all cells in the data set, sorted by retina. a. Scatter plot. Each vertical array of points corresponds to a retina in the data set. The points correspond to the individual cells in the retina. **b**. Box plot of the same data. The box plot format allows the median for each retina and the quartiles above and below it to be seen. *Horizontal line* indicates the median, *boxes* indicate the quartiles above and below it, *dotted lines* indicate the data ranges; +'s indicate outliers. *Methods*: Information was computed using the standard expression $I(r;s) = \Sigma_{r,s} p(r|s) \, p(s) \log_2[p(r|s)/p(r)]$ (37, 38) where the stimulus, as indicated in the main text, was either a gray field or a grating (each occurring with probability 0.5), and the response was spike count.
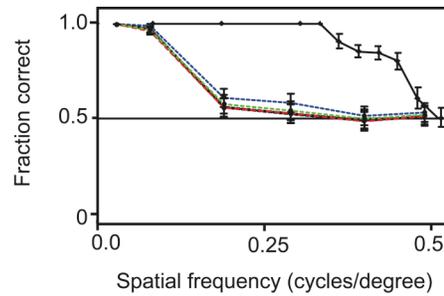
Fig. S8. Performance of the spike count code after the retinas with the lowest performing cells were systematically eliminated. *Solid black trace*, performance of the animal. *Dotted black trace*, performance using 300 cells, with cells drawn from the complete cell population. *Dotted red trace,* performance using 300 cells, with cells drawn from the population after the two lowest performing retinas were removed. *Dotted green trace*, performance with cells drawn after the three lowest were removed. *Dotted blue trace*, performance with cells drawn after the six lowest were removed. As mentioned in the text on the previous page, because performance is so slowly growing at 300 cells, eliminating these retinas has very little effect. The spike count code continues to fail ($p<0.0001$).

I. Decoding with the spike count code using responses shorter than the complete stimulus period (i.e., 50 ms and 100 ms, rather than the complete 300 ms.)

In the main text, when we evaluated the spike count code, we used the spike count for the whole stimulus presentation, which was 300 ms long. Since not all of the 300 ms response may contain informative spikes, we might be "diluting" the performance of the spike count code, that is, we might be underestimating its performance by including response periods that are essentially only adding noise. To address this, we measured performance using epochs within the 300 ms response, that is, we measured performance when we only counted spikes in the first 100 ms, then the second 100 ms, etc. We repeated this using the first 50 ms, the second 50 ms, etc. The results show that with some epochs the performance showed an increase relative to the performance with the complete 300 ms response, but the increase was very small: In all cases the performance fell substantially short of the performance of the animal (Fig. S9) ($p<<-0.001$ for all tests), supporting the notion that the failure of the spike count code is a robust result.

Note that these performances were expected to fall short of animal performance, because, barring data limitation issues, the spike count code must do worse than the spike timing code, and the spike timing falls short of the animal (as shown in Figs. 2 and 3 of

16

the main text). The data presented here are just to show the magnitude of the failure with different epochs.
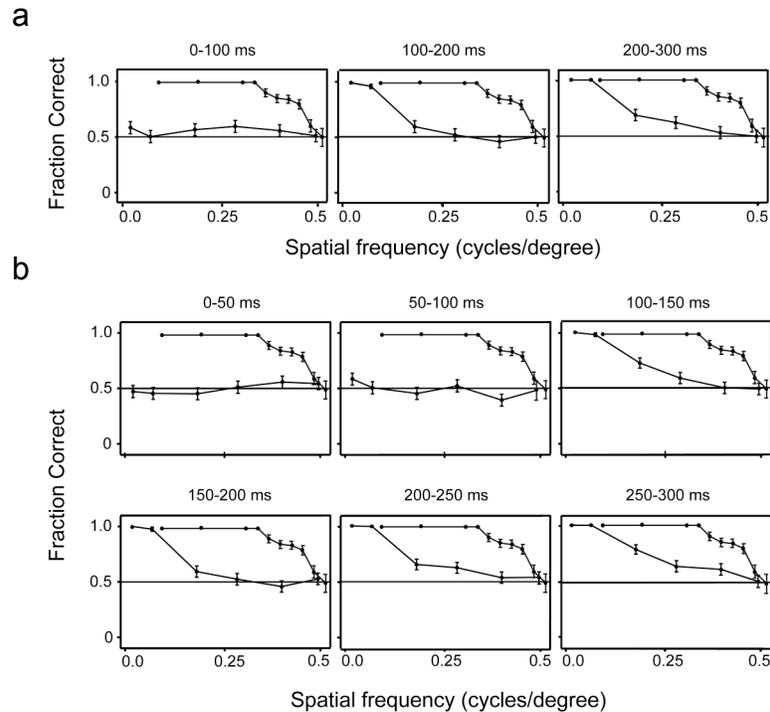
a

b

Fig. S9. Performance of the spike count code when spikes were counted for periods shorter than the complete stimulus period. a. Performance when spikes were counted only for the first 100 ms, then only for the second 100 ms, etc. **b**. Performance when spikes were counted only for the first 50 ms, then only for the second 50 ms, etc. *Solid black* trace, performance of the animal, *dotted black trace*, performance when spikes were counted only for the period indicated at the top of the panel.

REFERENCES

1.   Truccolo W, Eden UT, Fellows MR, Donoghue JP, Brown EN (2005) A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects. *J Neurophysiol* 93: 1074-1089.

2.   Kass RE, Ventura V (2001) A spike-train probability model. *Neural Comput* 13: 1713-1720.

3.   DiMatteo I, Genovese CR, Kass RE (2001) Bayesian curve-fitting with free-knot splines. *Biometrika* 88: 1055-1071.

4.   Bayesian Adaptive Regression Splines (BARS) software, http://lib.stat.cmu.edu/~kass/bars/bars.html.

5.   Berry MJ 2nd, Meister M (1998) Refractoriness and neural precision. *J Neurosci* 18: 2200-2211.

6.   Miller MI, Mark KE (1992) A statistical study of cochlear nerve discharge patterns in response to complex speech stimuli. *J Acoust Soc Am* 92: 202-209.

7.   Wichmann FA, Hill NJ (2001) The psychometric function: I. Fitting, sampling, and goodness of fit. *Percept Psychophys* 63: 1293-1313.

8.   Wichmann FA, Hill NJ (2001) The psychometric function: II. Bootstrap-based confidence intervals and sampling. *Percept Psychophys* 63: 1314-1329.

9.   Psignifit software, http://bootstrap-software.org/psignifit/.

10.   Nirenberg S, Carcieri SM, Jacobs AL, Latham PE (2001) Retinal ganglion cells act largely as independent encoders. *Nature* 411: 698-701.

11.   Golledge HD, *et al.* (2003) Correlations, feature-binding and population coding in primary visual cortex. *Neuroreport* 14: 1045-1450.

12.   Petersen RS, Panzeri S, Diamond ME (2001) Population coding of stimulus location in rat somatosensory cortex. *Neuron* 32: 503-514.

13.   Averbeck BB, Lee D (2003) Neural noise and movement-related codes in the macaque supplementary motor area. *J Neurosci* 23: 7630-7641.

14.   Oram MW, Hatsopoulos NG, Richmond BJ, Donoghue JP (2001) Excess synchrony in motor cortical neurons provides redundant direction in formation with that from coarse temporal measures. *J Neurophysiol* 86: 1700-1716.

15.   Gawne TJ, Richmond BJ (1993) How independent are the messages carried by adjacent inferior temporal cortical neurons? *J Neurosci* 13: 2758-71.

16.   Rolls ET, Franco L, Aggelopoulos NC, Reece S (2003) An information theoretic approach to the contributions of the firing rates and the correlations between the firing of neurons. *J Neurophysiol* 89: 2810-22.

17.   Nirenberg S, Latham PE (2003) Decoding neuronal spike trains: how important are correlations? *Proc Natl Acad Sci USA* 100: 7348-7353.

18.   Jeon CJ, Strettoi E, Masland RH (1998) The major cell populations of the mouse retina. *J Neurosci* 18: 8936-8946.

19. Williams RW, Strom RC, Rice DS, Goldowitz D (1996) Genetic and environmental control of variation in retinal ganglion cell number in mice. *J Neurosci* 16: 7193-7205.

20. Zhou G, Williams RW (1999) Mouse models for the analysis of myopia: an analysis of variation in eye size of adult mice. *Optom Vis Sci* 76: 408-418.

21. Carcieri SM, Jacobs AL, Nirenberg S (2003) Classification of retinal ganglion cells: a statistical approach. *J Neurophysiol* 90: 1704-1713.

22. Stone C, Pinto LH (1993) Response properties of ganglion cells in the isolated mouse retina. *Vis Neurosci* 10: 31-39.

23. Tian N, Copenhagen DR (2003) Visual stimulation is required for refinement of ON and OFF pathways in postnatal retina. *Neuron* 39: 85-96.

24. Balkema GW, Jr, Pinto LH (1982) Electrophysiology of retinal ganglion cells in the mouse: a study of a normally pigmented mouse and a congenic hypopigmentation mutant, pearl. *J Neurophysiol* 48: 968-980.

25. Sagdullaev BT, McCall MA (2005) Stimulus size and intensity alter fundamental receptive-field properties of mouse retinal ganglion cells in vivo. *Vis Neurosci* 22: 649-659.

26. Doi M, Uji Y, Yamamura H (1995) Morphological classification of retinal ganglion cells in mice. *J Comp Neurol* 356: 368-386.

27. Treves A, Panzeri S (1995) The upward bias in measures of information derived from limited data samples. *Neural Computation* 7: 399-407.

28. http://www.microscopyu.com/articles/optics/objectiveproperties.html

29. Lucas RJ, *et al.* (2003) Diminished pupillary light reflex at high irradiances in melanopsin-knockout mice. *Science* 299: 245-247.

30. Pennesi ME, Lyubarsky AL, Pugh EN, Jr (1998) Extreme responsiveness of the pupil of the dark-adapted mouse to steady retinal illumination. *Invest Ophthalmol Vis Sci* 39: 2148-2156.

31. Lyubarsky AL, Daniele LL, Pugh EN, Jr (2004) From candelas to photoisomerizations in the mouse eye by rhodopsin bleaching in situ and the light-rearing dependence of the major components of the mouse ERG. *Vision Res* 44: 3235–3251.

32. Alpern M, Fulton AB, Baker BN (1987) "Self-screening" of rhodopsin in rod outer segments. *Vision Res* 27: 1459-1470.

33. Ulmino, Y, Solessio, E, Barlow, RB (2008) Speed, spatial, and temporal tuning of rod and cone vision in mouse. *J Neurosci* 28: 189-98.

34. Cavonius CR, Robbins DO (1973) Relationships between luminance and visual acuity in the rhesus monkey. *J Physiol* 232: 239-246.

35. Lyubarsky AL, *et al.* (2001) RGS9-1 is required for normal inactivation of mouse cone phototransduction. *Mol Vis* 7: 71-78.

36. Remtulla S, Hallett PE (1985) A schematic eye for the mouse, and comparisons with the rat. *Vision Res* 25: 21-31.

37. Shannon, CE (1963) *The Mathematical Theory of Communication* (University of Illinois Press, Illinois).

38. Cover TM, Thomas JA (1991) *Elements of Information Theory* (Wiley, New York).