Supplemental Information

Neural Circuit Inference

from Function to Structure

Esteban Real, Hiroki Asari, Tim Gollisch, and Markus Meister
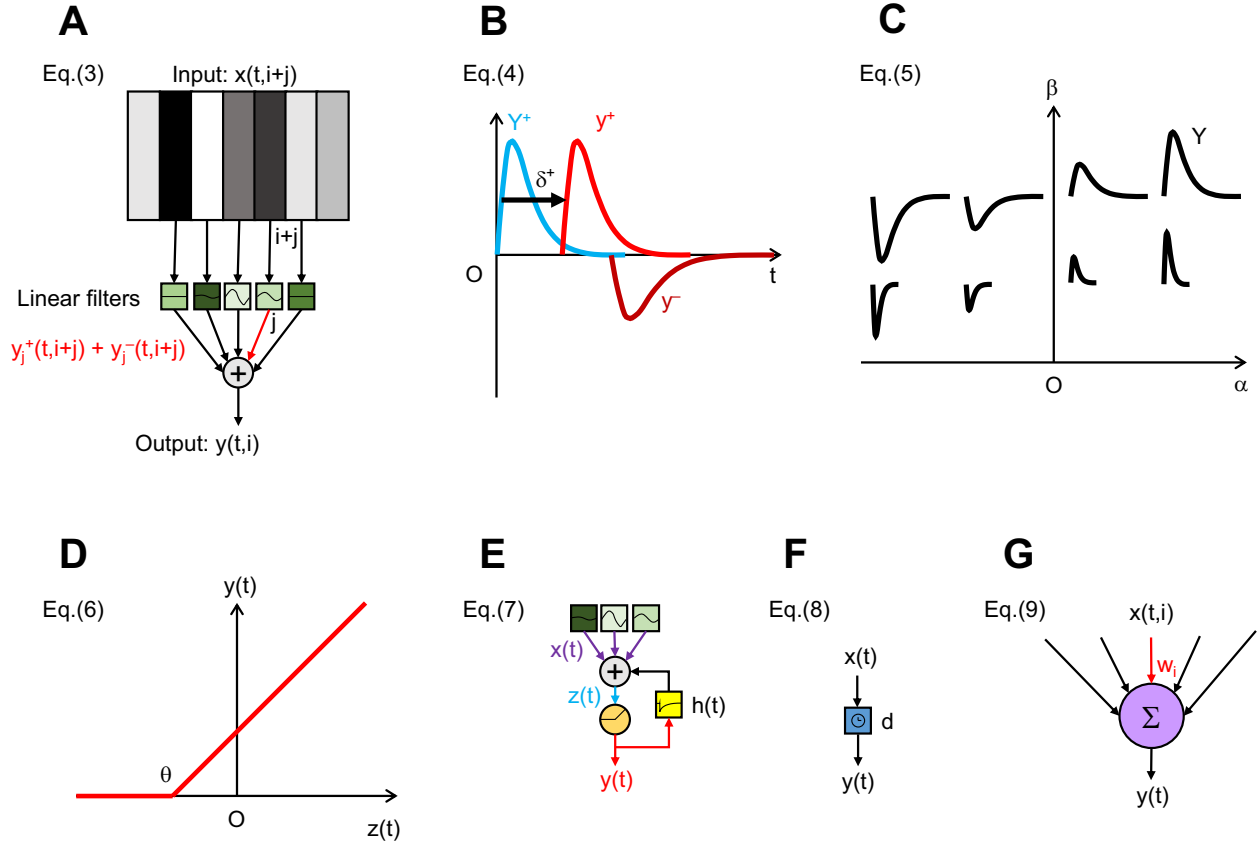
Figure S1, related to Figure 1: Schematics of LNFDSNF model components and related formulas.

(A) Linear filters ("L" stage), related to Eq.S3: $x(t, i+j)$, the input stimulus at location $i+j$; $y_j^+(t, i+j) + y_j^-(t, i+j)$, the output of linear filter at relative location j; and $y(t, i)$, the output of BCM at location i.

(B) Impulse response of the IIR filters, related to Eq.S4. The positive lobe of the BMC temporal processing y^+ is obtained by shifting Y^+ in time t by the amount δ^+. The negative lobe y is similarly obtained by the time warp of Y (by the amount δ^-; not shown). Indices for space and time are omitted for clarity.

(C) Dependence of the IIR filter Y (impulse response) on the free parameters (α, amplitude; $\beta \geq 0$, timescale), related to Eq.S5. All indices are omitted for clarity, but note $\alpha^+ \geq 0$ for Y^+ and $\alpha^- \leq 0$ for Y^-.

(D) Half-wave rectifier at threshold θ, related to Eq.S6.

(E) BCM feedback (together with nonlinearity; the first "NF" stage), related to Eq.S7. Note that the output of BCM linear filter is denoted as the input $x(t)$ to this stage.

(F) Delay function ("D" stage), related to Eq.S8.

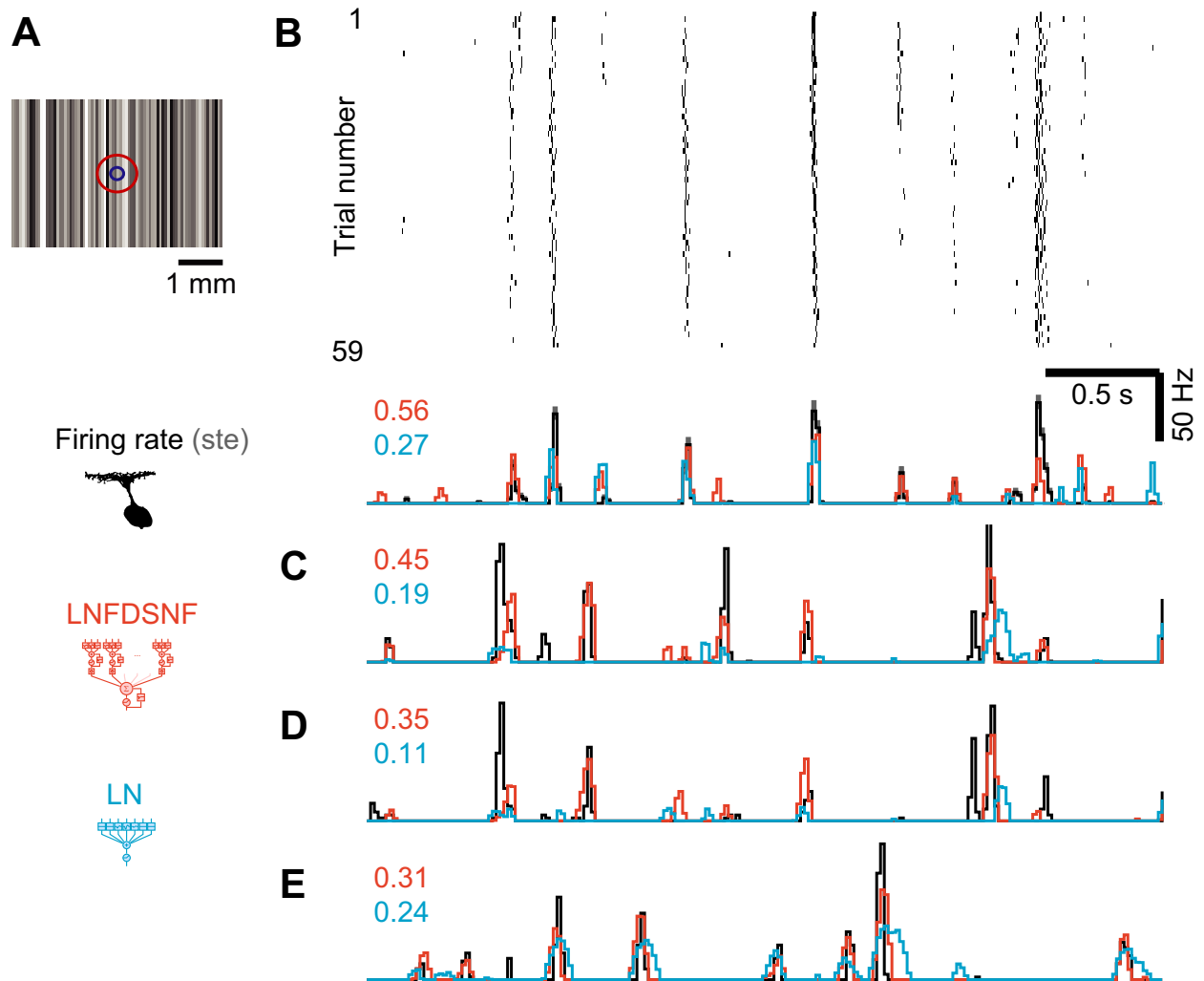**(G)** GCM spatial pooling ("S" stage), related to Eq.S9.

Figure S2, related to Figure 2: More examples for the measured and predicted ganglion cell visual responses.

(A) One stimulus frame. The stimulus was an array of adjacent vertical bars (66 μm width), whose gray intensities flickered simultaneously and independently at 60 Hz. The overlaid circles indicate the typical extent of a ganglion cell's receptive field (red) and its center (blue).

(B) Typical responses of a ganglion cell to repetitions of the stimulus in the same format as in Figure 2A (top, raster graph; bottom, time course of measured and predicted ganglion cell firing rate; E.V. values are shown for each model in corresponding color).

(C–E) Three more examples of ganglion cell visual responses (in the same format as in panel **B**, bottom).
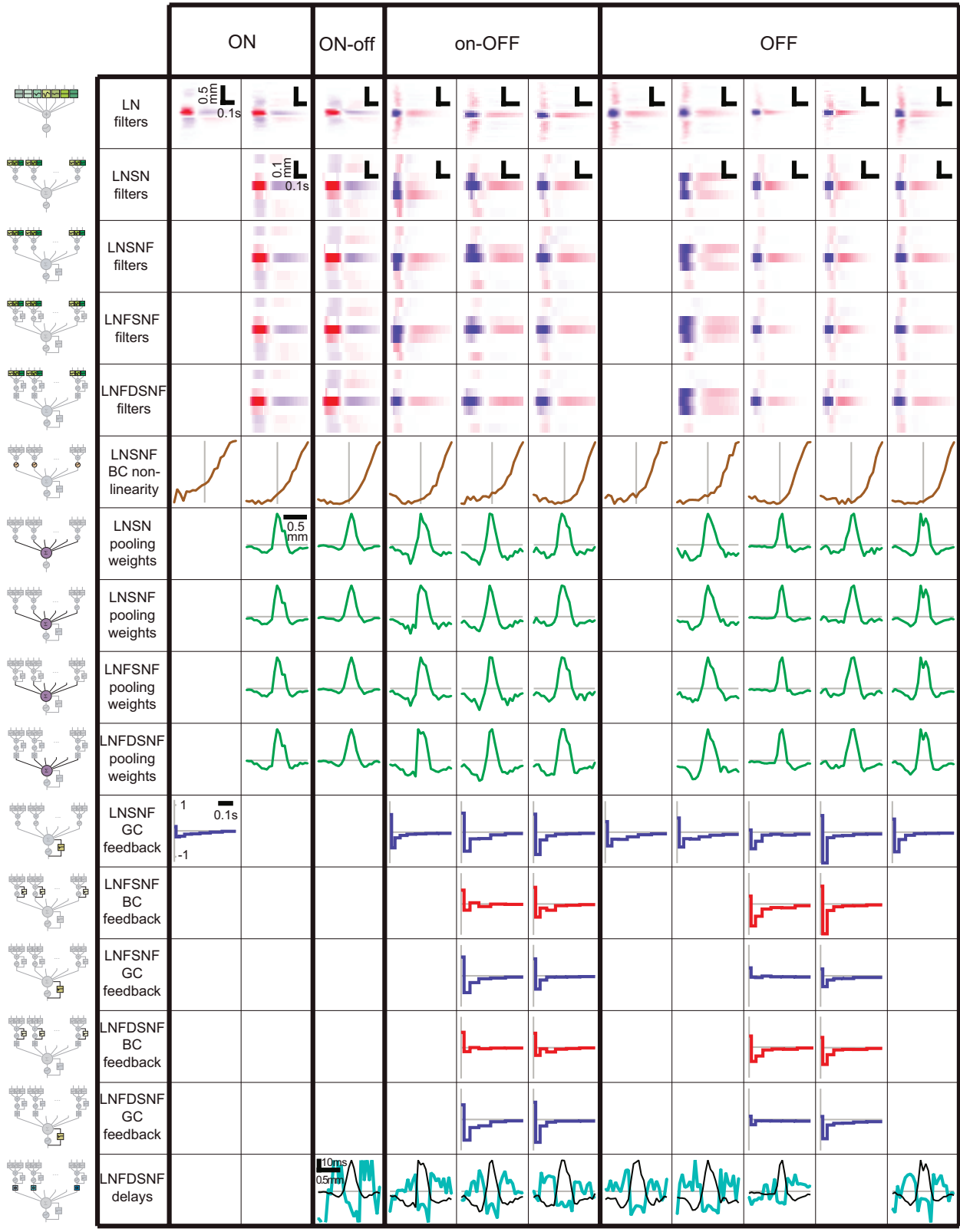
Figure S3, related to Figure 1: Optimized parameters across cells and models. Fitted model parameters for 11 representative ganglion cells of various types. Entries for parameters that did not lead to at least 5% improvement have been left blank.

Fitting some of these circuit models involved optimizing well over 100 parameters (Figure 1G). This is less of a challenge under special conditions where the objective function is convex and has just one optimum. Such restrictions apply, for example, to the simplest cascade model (LN) and some other models attractive to neuroscience [S1, S2]. However, the most general neural circuit involves stages of feedback and recurrence, and one cannot expect convexity in the system parameters. It is then a major concern whether optimization in such a large space can converge to a global solution.

Of course it always helps to limit the number of parameters at the outset. We thus kept the retinal circuit models as simple and basic as possible in their structure. Specifically, we took all BCs in a given GC circuit to have the same properties, and constrained the BCM and GCM nonlinearities to a half-wave rectifier (Figure S1). The resulting parameter estimates were indeed robust, as verified by various tests (Figure S5). When convergence was problematic we changed the structure of the model; for example this occurred in an attempt to fit the full shape of the GCM nonlinearity, which was then replaced by a simpler rectifier (see *Supplemental Experimental Procedures*).

We also restricted the stimulus to one dimension in space and one in time (Figure S2A). This gave enough power to resolve both spatial and temporal structure of the circuit components. Many retinal circuits are isotropic to good approximation, so that sampling of one spatial dimension is sufficient. There are some exceptions, though, such as direction-selective GCs that form distinct circuits for their specific functions [S3]. Consequently, our circuit models did not perform well for all cells (Figure 2C,D). This suggests that the circuit models (and visual stimuli) will need to be tailored for each GC type to better probe the underlying circuits. Such tailored models may require a larger number of free parameters. We expect that future development of efficient search algorithms will make it possible to apply a machine learning approach even to those more complicated models [S4].
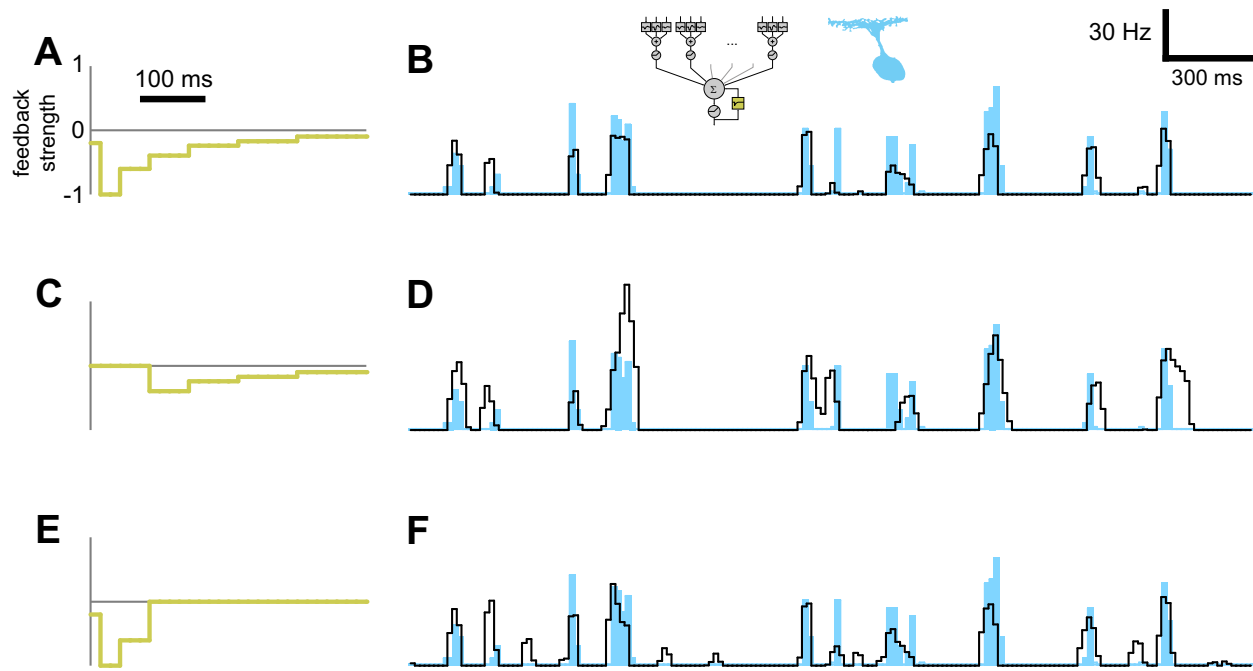
Figure S4, related to Figure 5: Detailed effects of GCM feedback on the model output.

(A) LNSNF feedback function fitted to a representative cell.

(B) The corresponding output of the model (black) and the data (blue).

(C) The same feedback function as in **A** but with the early portion up to 100 ms removed.

(D) The model output using the modified feedback function in **C** shows that many firing events become broader than appropriate.

(E) The same feedback function as in **A** but with the long tail beyond 100 ms removed.

(F) The model output using the modified feedback function in **E** shows many superfluous firing events at inappropriate times.

A

B

C

scaled output

1 sd

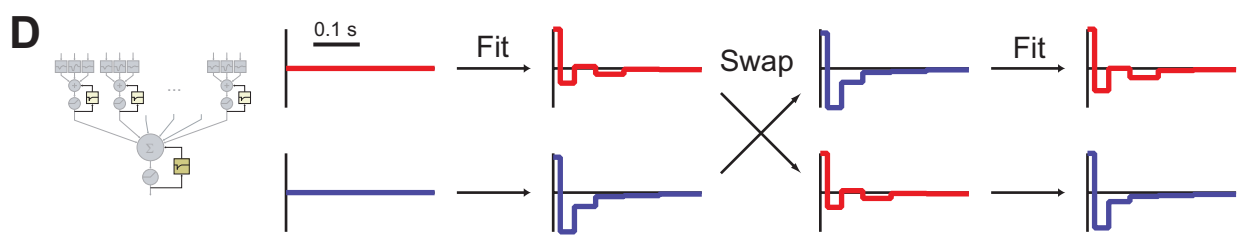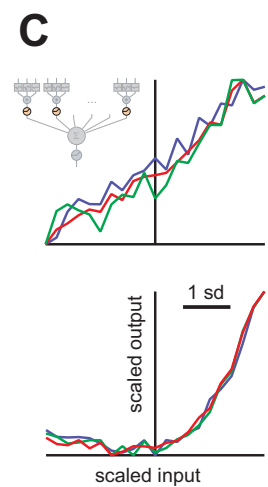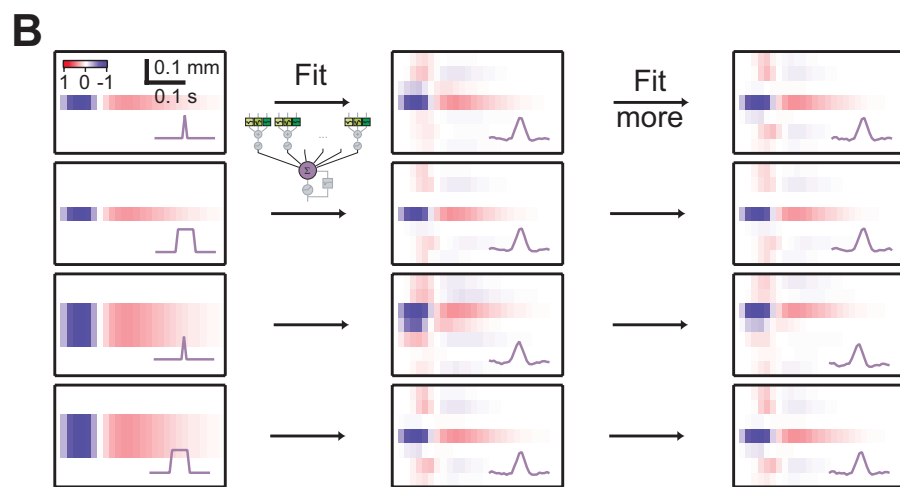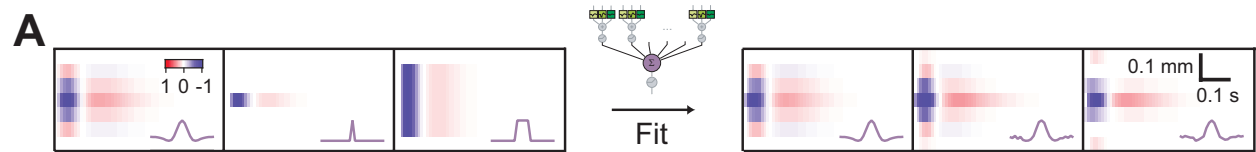scaled input

D

0.1 s

Fit

Swap

Fit

Figure S5, related to Figures 3–5: Numerical tests of the fitting algorithm. Unlike the simple LN model, whose convexity properties guarantee a single optimum in parameter space [S1], the more complex models considered here may allow for multiple local optima. Moreover, the fitting algorithm we used (the Polak-Ribière variant of conjugate gradient ascent) [S5] is not guaranteed to converge to a globally optimal solution. To test the fitting results thoroughly, we thus carried out various kinds of convergence tests.

(A) As the first minimal test of the present methods, a specific parameterization of the LNSN model was used to generate artificial data. Starting with various initial conditions, we then fitted the model to such artificial data. The parameters converged to the ones used to generate the artificial data, even for widely different initial conditions, confirming that the search algorithm can find a parameter set with known "ground truth".

Shown here are the convergence results for artificially generated data, using the LNSN model. The three initial conditions for the BCM filters are on the left (red hue, ON-polarity; blue hue, OFF-polarity) and the initial pooling functions appear in the violet insets. The leftmost initial condition was the one used to generate the data. The corresponding results after fitting are on the right. The BCM filters and pooling function converge to the values used to generate the data, regardless of the initial condition.

(B,C) When using real data, a different type of test is necessary, because the ideal values of the parameters are not known. One strategy was to vary the initial values of the parameters and see if the search converges on a consistent set of final values. These tests covered the BCM filters and the pooling weights in the LNSNF model (**B**) and the BCM output nonlinearity in the LNSN model (**C**). The initial condition for the feedback was set to zero in all applicable cases. While the fitted parameters converged to the same values in most cases, some initial conditions ended up with different parameters from the rest. On most of these occasions, however, the attained explained variance was much lower than for the optimal parameter set. Presumably these initial conditions were too far from the global optimum and led to an inferior local optimum.

Shown in **B** are the convergence results for real data, using the LNSNF model. Each row corresponds to a different set of initial conditions (left column) for the BCM filters and the pooling weights. After 100 iterations, the results have converged (middle column), and this is unchanged by subsequent 100 iterations (right column).

Shown in **C** are the convergence results for the BCM output nonlinearities of a representative linear cell (top) and nonlinear cell (bottom). The three colors correspond to different initial conditions for the nonlinearity: a half-wave rectifier, a linear function, and a step function. Due to degeneracies of the model, an overall additive constant and an overall multiplicative factor are inconsequential. The functions shown here have therefore been rescaled.

(D) In seven of 30 GCs, the transition from LNSNF to LNFSNF resulted in >5% fractional improvement in the explained variance (Figure 5C). This suggests that the second feedback function improves the circuit model in a substantial way. Because the two feedback functions, the one around the GCM and the other around the BCM, often attained distinct shapes (Figure 5A), we tested if these two shapes are interchangeable due to a degeneracy or specific to their locations within the circuit model. We restricted the test to those seven GCs, and re-ran the LNFSNF model on each of them as follows: the fitted BCM and GCM feedback functions were exchanged for each other and fed back into the model as the new initial conditions, while simultaneously resetting all the other free parameters.

The result was that the feedback functions reverted back to the original fitting results in all cases. An example is shown here for the independent convergence of the BCM feedback (top row) and GCM feedback (bottom row) of the LNFSNF model. Fitting identically-zero initial conditions yields typical shapes for both feedback functions (left two columns). Swapping them for each other, resetting the other free parameters, and fitting again restores the feedback functions that had been found in the first place (right two columns). This suggests that the two feedback elements around the pre- and post-pooling parts of the model do indeed have distinct properties, each playing a unique role in retinal processing.
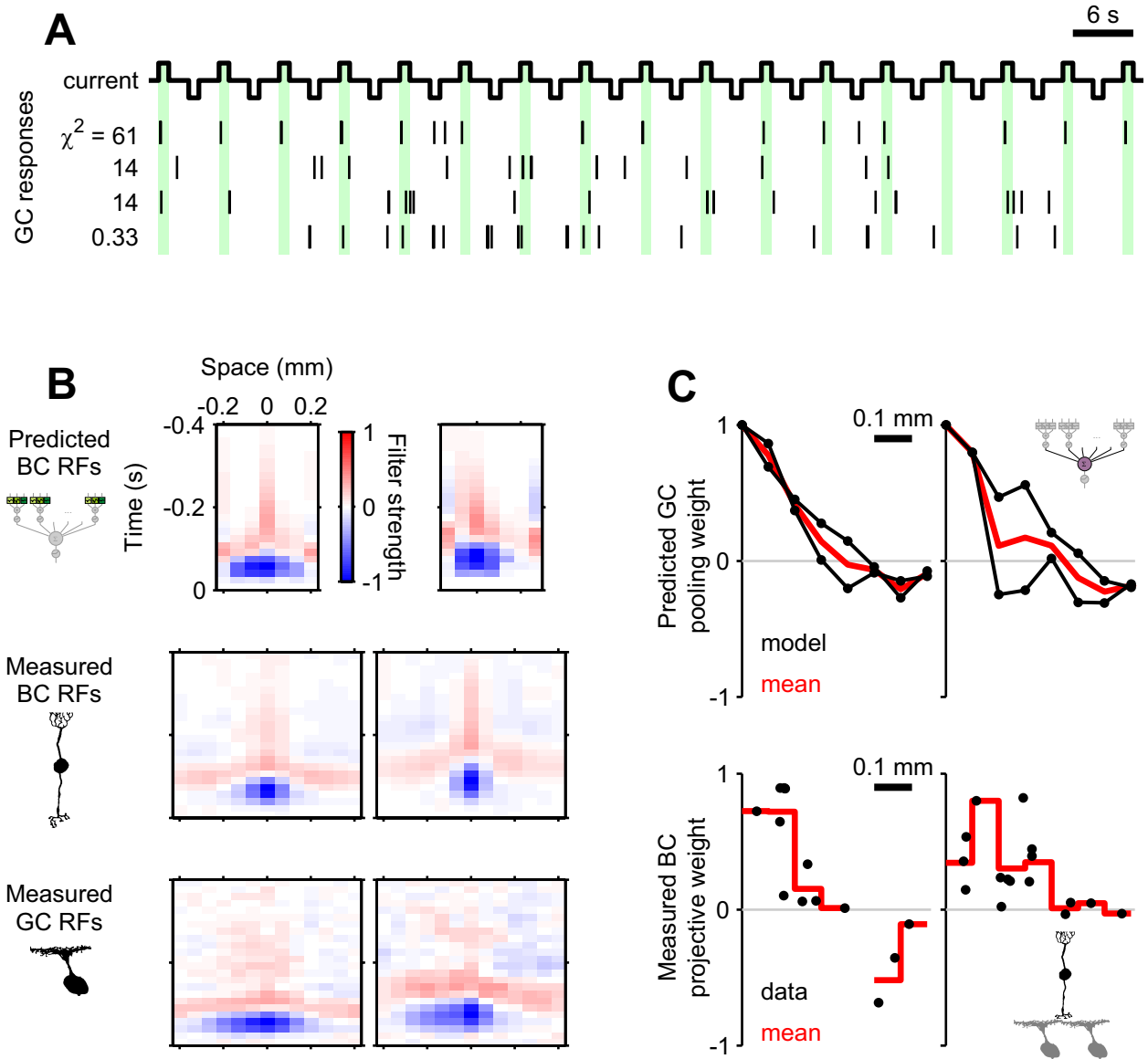
Figure S6, related to Figure 7: More examples for the experimental tests of the models.

(A) Spiking responses of four GCs in response to current injections (±500 pA square pulses) into a single BC (top row, injected current trace). The first three GCs (rows 2–4) are more likely to fire than by chance during a depolarizing current injection (green shade), suggesting a significant projection from that BC ($\chi^2 \gg 1$). For comparison, the last row shows a fourth GC that did not receive a projection from the source BC. The first GC corresponds to the example in Figure 7A.

(B) Predicted (top) and measured (middle) BC receptive fields (RFs), with the corresponding GC RF (bottom) obtained by a simultaneous BC-GC recording. The left and right columns correspond to the second and the third examples in **A**, respectively.

(C) The pooling function of the two representative GCs (top) and the projective weights of the simultaneously recorded BC (bottom); left and right from the corresponding BC-GC pairs in **B**.
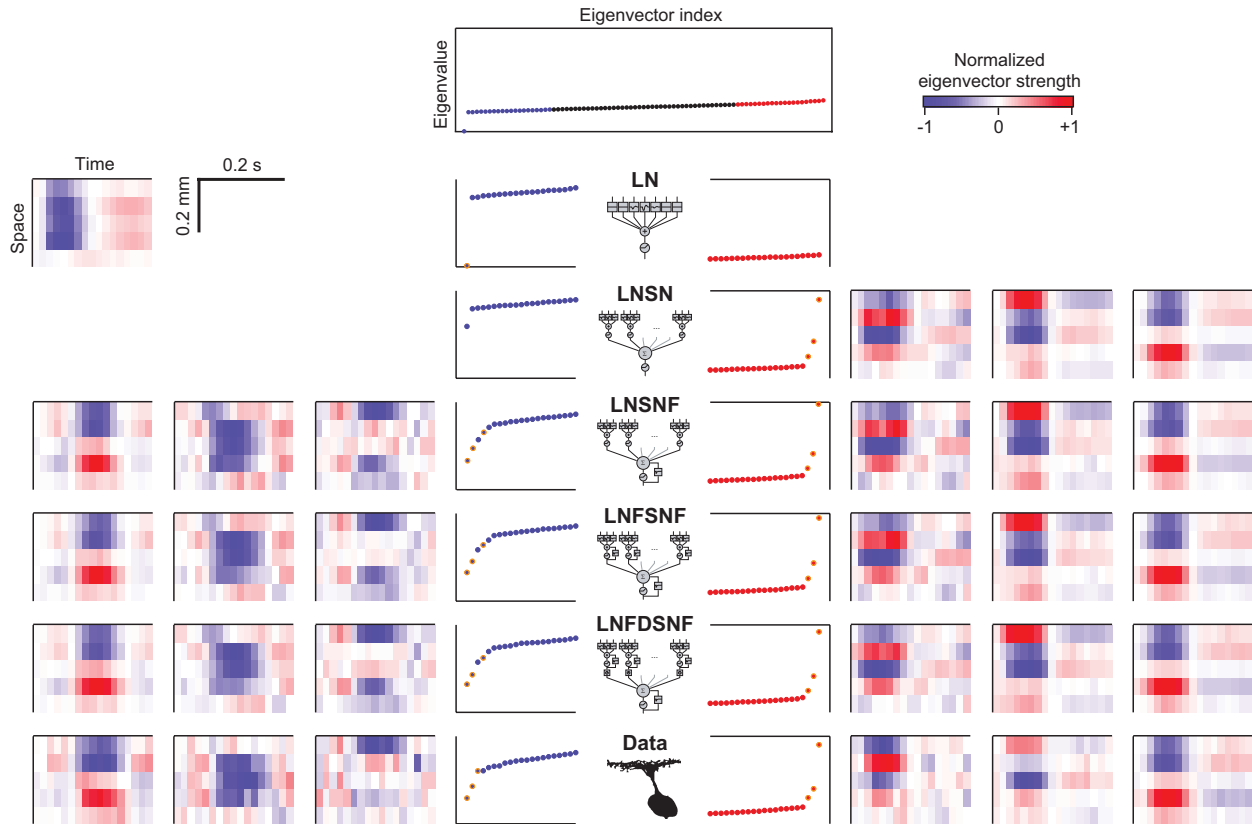
Figure S7, related to Figure 2: Spike-triggered covariance analysis for a typical cell across models. We evaluated the model performance using the explained variance of the model, as defined in Eq.S10, which compares the full time courses between the firing rate responses of GCs and the model outputs. There are, however, many other statistical quantities that could be used instead, which extract and emphasize certain aspects of the stimulus-response relationship. As an alternative technique for assessing the models, we performed a standard spike-triggered covariance (STC) analysis [S6]. Since the models do not produce spikes, the STC matrix for the models was computed by considering the contribution of every bin and weighing it by the model output for that bin. This is analogous to spike-triggering, which weighs spike-containing bins with a value of 1 and all other bins with a value of 0.

The STC analysis limits itself to the study of the second-order statistics of the stimulus after removing the mean, and thus focuses only on the sections of stimulus space that were most successful in causing a response from the cells (or the models). STC is especially sensitive to nonlinearities that may exist in the pooling of information from different spatial locations [S6]. Moreover, the eigenvalues and eigenvectors of the STC matrix can be related to linear filters in a multi-filter LN model [S6]. These features make it appealing as an alternative way to assess the improvement of the models in the successive stages.

The topmost graph shows the full eigenvalue spectrum for the LN model. Only those eigenvalues matter that are significantly larger or smaller than expected. Thus the central column shows exclusively the low-end (blue dots) and high-end (red dots) tails of the spectrum. Each row corresponds to a different model; the last row corresponds to the data. To the sides, the eigenvectors for the most significant eigenvalues are plotted (left, low-end; right, high-end; the corresponding eigenvalues are highlighted by orange circles). These vectors, being representations of the stimulus space, are best displayed as two-dimensional space-time surface plots, akin to receptive fields (red hue, positive values; blue hue, negative values). An improvement in the models is reflected in how much the eigenvalues and eigenvectors of their STC analysis match those of the data. In most cases, such as in this example, this improvement is evident from LN to LNSN, to LNSNF. Beyond LNSNF, the eigenvalues and eigenvectors do not change much. In contrast, the explained variance shows that the improvement continues through to LNFSNF and LNFDSNF (Figure 2C,D), indicating a limitation of the STC analysis that exploits only the second order stimulus statistics.

Supplemental Experimental Procedures

Electrophysiology

We isolated the retina of a larval tiger salamander (*Ambystoma tigrinum*) in the dark, and placed a piece (2-4 mm in diameter) on a flat array of 61 extracellular electrodes with the ganglion cell (GC) side down. The retina was superfused with oxygenated Ringer's medium (in mM: NaCl, 110; NaHCO$_3$, 22; KCl, 2.5; MgCl$_2$, 1.6; CaCl$_2$, 1; and D-glucose, 10; equilibrated with 95% O$_2$ and 5% CO$_2$ gas) at room temperature. The electrode array recorded the extracellular signals from GCs, while the photoreceptors were visually stimulated [S7, S8]. A computer stored the waveform of the signal from each electrode, sampling them at 10 kHz. Further offline processing with custom software extracted the spike trains for the individual GCs [S9]. In particular, we discarded any spike train with inter-spike intervals of less than 4 ms because it likely represents multi-unit activity [S10].

We made intracellular recordings from bipolar cells (BCs) using a sharp glass electrode filled with 2 M potassium acetate and 3% Rhodamine Dextran 10,000 MW (final impedance 150–250 MΩ). Under infrared illumination, the electrode was blindly inserted into various cells until one with the response characteristics matching those of BCs was found [S11]. To measure the projection from individual BCs to their downstream GCs, the intracellular electrode was also used to stimulate the BCs directly by injecting current in current-clamp mode (Figure S6A) [S12, S13].

Visual stimulation

We stimulated the isolated retina using a gamma-corrected cathode-ray tube monitor (DELL M783s) that produced white light in the photopic regime (approximately 10^{12} photons cm^{-2} s^{-1}). The stimulus consisted of a 1-dimensional array of adjacent bars 66 μm in width (Figure S2A), which corresponds approximately to the size of a dendritic field of BCs [S14, S15, S16]. Their gray intensities changed simultaneously, independently, and randomly with a refresh rate of 60 Hz. These intensities were drawn from a Gaussian distribution or from a binary black-or-white distribution. The projected image was focused on the photoreceptor layer and covered the entire retinal piece under study. The length of this random sequence varied between a few minutes and a few hours. The data collected in this manner constituted the stimulus for the training data set. Interleaved with the stimulus described above were a series of 60 s-long identical sequences with the same flickering bar structure and statistics. The number of repetitions ranged from 8 to 58. These repeated sequences comprised the stimulus for the testing data set.

We chose the white noise stimuli because of the convenience to generate a large unbiased ensemble to achieve efficient system identification. Because the nervous system is nonlinear, plastic, and dynamic, however, the response models will need to be adjusted if one moves from one ensemble to another, such as natural stimuli. It will be an interesting research direction for the future to develop models with multi-scale dynamics that generalize better to cover the retinal responses under a wider stimulus space.

Data selection

The raw data set contained about 200 retinal GCs from 6 isolated retinas. Of those, 30 well-isolated GCs were deemed appropriate for the subsequent modeling analyses according to the following three criteria:

1. Constant firing rate over an extended period of time, preferably over an hour, with more than 2,000 spikes in total.

2. Clear response to the stimulus, not simply spontaneous firing, so the spike-triggered average clearly reveals receptive field structure.

3. No sudden changes in response to the repeated stimulus sequences.

This selection of data was done entirely before starting to fit or evaluate the models. It should thus introduce no bias as to whether the cells chosen are especially suited to the specific models examined. Although imposing a lower limit on the total number of spikes and requiring a clear receptive field center may be biasing the selection toward certain cell types, these requirements were necessary for the gradient ascent algorithm to converge. The only goal of the selection was to provide high quality data for the model fitting process. No normalization or other post-processing was performed on the recorded data.

Cell-type classification

We identified the cell type from the flickering bar data as described previously [S17]. Briefly, we examined the shape of the nonlinearity associated with the most significant eigenvalue of the spike-triggered covariance matrix. The cells were then classified into four types (Figure 2D), according to how they respond to changes in luminance at their receptive field centers:

1. ON cells, which increase their activity only with an increase in luminance;

2. ON-off cells, which increase their activity with both an increase and a decrease in luminance, but biased towards ON;

3. on-OFF cells, which increase their activity with both an increase and a decrease in luminance, but biased towards OFF; and

4. OFF cells, which increase their activity only with a decrease in luminance.

For the population analysis in Figure 4, the first two are grouped as ON types, and the last two as OFF types.

Receptive field analysis

We used stimulus ensemble statistical techniques ("reverse correlation" methods) to calculate the spatio-temporal receptive fields. In the case of GCs, we computed a spike-triggered average (STA) of the stimulus (Figures 3C, 6C, 7A and S6B) [S7, S18]. The STA is the average over all spikes of the visual stimulus that occurred in a brief interval before the spike. It is generally indicative of what stimulus makes the cell fire action potentials. Intracellular BC recordings and model outputs do not have spikes but vary continuously

in their response. In this case, we computed the reverse correlation of the response, namely the average of the stimulus before each time bin, weighted by the response value for that bin (Figures 6C, 7A and S6B)

To determine the latencies of center and surround regions in a spatio-temporal receptive field (Figure 6C,D), we first computed the latency of the peak at each spatial location and then averaged these numbers, weighted by the peak amplitude, separately over the regions with positive and negative amplitude. To characterize the spatial profile (Figure 7B), we averaged the spatio-temporal receptive field over all time points between the center and surround peak latencies. The zero-crossing radius was then obtained by linear interpolation of the data points.

Projective field analysis

We analyzed the data from simultaneous BC-GC recordings similarly as in [S12, S13]. In brief, the projection strength was first calculated for each BC-GC pair as follows:

$$\text{projective weight} = \frac{N_{\mathrm{d}} - N_{\mathrm{h}}}{N_{\mathrm{d}} + N_{\mathrm{h}}}, \tag{S1}$$

where N_{d} and N_{h} are the total number of spikes fired by the GC when the depolarizing and hyperpolarizing current was injected into the BC, respectively. To obtain the BC's projective field, these weights were then plotted as a function of the distance from the BC to the GCs (Figures 7B and S6C). The BC-GC distance was estimated from their receptive field centers mapped by a randomly flickering checkerboard stimulus. The spatial profile of the projective field was then characterized by the zero-crossing radius (Figure 7D).

We also ran a χ^2 test to examine if the current injected into a BC affected the spiking response of a GC:

$$\chi^2 = \frac{(N_{\mathrm{d}} - \bar{N})^2}{\bar{N}} + \frac{(N_{\mathrm{h}} - \bar{N})^2}{\bar{N}}, \tag{S2}$$

where $\bar{N} = (N_{\mathrm{d}} + N_{\mathrm{h}})/2$ is taken as the predicted number of spikes under the null hypothesis of no projection. Together with other requirements on the GC data, this reduced the data set to 14 BC-GC pairs (from 6 BCs, each projecting to 1–4 GCs) for the modeling analyses. As before, the selection of these cell pairs was done entirely before fitting the models.

Model formalism

We employed the cascade model framework [S19, S20] and progressively extended its complexity (Figure 1), from the linear–nonlinear (LN) model to the linear–nonlinear–feedback–delayed–sum–nonlinear–feedback (LNFDSNF) model. Unlike in many other applications of machine learning, the goal here is not improved data fitting using arbitrary functions, but an interpretation of the fitting function itself in terms of biological structure. Therefore we chose as a reference model not the best existing mathematical functions for response prediction, but the LN model, which lends itself to developing increased biological realism. In this process we began by splitting the retina into two layers with bipolar cell modules (BCMs) as the spatial subunits. Then we introduced local feedback circuits and time delays. Figure 1G summarizes the

number of free parameters for each component of the final cascade. Note that the LN model has the most free parameters (186 in L and 1 in N) among the models we tested.

In the following, the input and output of any stage are denoted as $x(t, i)$ and $y(t, i)$, respectively, where the time t is binned at 1/60 s and i represents discrete spatial locations. In all models, a modeled GC covered a spatial window of 2.05 mm (31 stimulus bars).

Linear filters ("L" in LNFDSNf):

For modeling temporal processing, we used discrete time infinite impulse response (IIR) filters. This was essential to speed up the simulations required in fitting the model. The IIR filters were implemented with 6 free parameters at each spatial location to produce a biphasic function in time (see Figure 3A for example). The 6 numbers correspond to the amplitudes, timescales, and temporal locations of each of the two phases. This results in a total of 186 free parameters in this stage for the LN model where the linear filters of a modeled GC covered the entire 31 stimulus width. In contrast, all the other models employing a BCM have only 42 free parameters here because the space is tiled by identical BCMs, each covering 7 stimulus bars and overlapping with its nearest neighbor over 6 stimulus bars.

The detailed implementation of the IIR filters was as follows: The output of the BCM at location i was computed as (Figure S1A)

$$y(t, i) = \sum_{j=-3}^{3} y_j^+(t, i + j) + y_j^-(t, i + j), \tag{S3}$$

where $y_j^+(t, i+j)$ and $y_j^-(t, i+j)$ are the outputs of the time-warped second-order IIR filters that respectively represent the positive and negative lobes of the BCM temporal processing at spatial location $i + j$. These IIR filters are identical in form, each with 3 free parameters (amplitude α_j^*, timescale β_j^*, and temporal location δ_j^* with "$*$" being either "$+$" or "$-$"), and written as follows:

$$y_j^*(t, i + j) = (1 - \{\delta_j^*\}) Y_j^*(t - \lfloor \delta_j^* \rfloor, i + j) + \{\delta_j^*\} Y_j^*(t - \lfloor \delta_j^* \rfloor - 1, i + j), \tag{S4}$$

$$Y_j^*(t, i + j) = \alpha_j^* x(t, i + j) + 2\beta_j^* Y_j^*(t - 1, i + j) - \beta_j^{*2} Y_j^*(t - 2, i + j), \tag{S5}$$

where $\alpha_j^+ \geq 0$, $\alpha_j^- \leq 0$, $\beta_j^* \geq 0$ and $\delta_j^* \geq 0$. The Eq.S4 represents the time-shifting of $Y_j^*(t, i + j)$ by the amount δ_j^* (Figure S1B), where the floor $\lfloor \delta_j^* \rfloor$ is the largest integer not greater than δ_j^*, and the fractional part $\{\delta_j^*\} = \delta_j^* - \lfloor \delta_j^* \rfloor$. The Eq.S5 is the difference equation of the IIR filter with the feed-forward filter coefficient α_j^* and the feedback filter coefficients $2\beta_j^*$ and $-\beta_j^{*2}$ (Figure S1C).

Preliminary runs confirmed that the parameterization of linear filters as in Eqs.S3–S5 is appropriate, even though the free parameters themselves do not have direct biological interpretations: A point-wise fit of the BCM linear filters (together with other parameters simultaneously) resulted in very similar outcomes. The point-wise fits, however, tended to overfit as the models became more complicated or as the amount of data was decreased. In contrast, this tendency was not observed when using the 6-parameter IIR filters.

BCM nonlinearity and feedback (the first "NF" in LNFDSNF):

In the LNSN model, we used a point-wise static nonlinearity (21 free parameters) for the BCM output (the first "N" in LNSN; Figure 4). In the other models, we approximated the BCM nonlinearity using a half-wave rectifier with a free threshold location θ (Figure S1D), implemented together with the feedback kernel $h(t)$ (Figure S1E) as follows:

$$y(t) = \begin{cases} 0, & \text{if } z(t) \leq \theta \\ z(t) - \theta, & \text{otherwise,} \end{cases} \tag{S6}$$

$$z(t) = x(t) + \sum_{s \geq 0} h(s)\, y(t - s - 1). \tag{S7}$$

The spatial index i is omitted for clarity. We parameterized $h(t)$ to achieve higher temporal resolutions at shorter times (7 free parameters; Figure 5). Specifically, the value of the feedback function at the first time bin was a free parameter, the second and third bins were another, the next three were another, and so on, giving a square root time dependence for the resolution.

Delay function ("D" in LNFDSNF):

We assigned the delays d_i independently to each BCM, resulting in 25 more free parameters ($i = 4, \ldots, 28$; Figure 6). When the delay d was not a multiple of the stimulus sampling interval, this required interpolation of the input signals $x(t)$ as in Eq.S4:

$$y(t) = (1 - \{d\})\, x(t - \lfloor d \rfloor) + \{d\}\, x(t - \lfloor d \rfloor - 1), \tag{S8}$$

where $\lfloor d \rfloor$ and $\{d\}$ are the integer and fractional part of d as measured in stimulus intervals (Figure S1F).

GCM spatial pooling ("S" in LNFDSNF):

Spatial pooling of the GCM is formulated as a weighted sum of the inputs $x(t, i)$ across BCMs ($i = 4, \ldots, 28$; Figure S1G):

$$y(t) = \sum_i w_i\, x(t, i). \tag{S9}$$

This results in a pooling function w_i with 25 free parameters (Figure 3B), with which a modeled GC covered a spatial window of 2.05 mm (31 stimulus bars).

GCM nonlinearity and feedback (the second "NF" in LNFDSNF):

We used Eqs.S6 and S7 for the GCM nonlinearity and feedback. In all models but LN, however, we used a fixed threshold $\theta = 0$ because the GCM nonlinearity proved very hard to fit as it did not converge. This function is nevertheless compatible with previous studies [S18]. The GCM feedback kernel was parameterized with 7 free parameters as in the BCM feedback (Figure 5).

Model fitting

To fit the model parameters, we wrote C++ code and ran it on the training data set for each of the 30 select GCs. The code was compiled and executed in Harvard University's Odyssey computer cluster and on a single computer with an NVIDIA Tesla C1060 card using the NVIDIA CUDA library.

For computing purposes, time was divided into a succession of identical bins. The data spike train was then represented as the number of spikes that was recorded in each time bin, and the output of the models was treated analogously. The objective function of the fitting algorithm was the fractional variance of the data spike train that is explained by the model output [S21, S22]:

$$\text{explained variance} = 1 - \frac{\sum_t (n_t - r_t)^2}{\sum_t (n_t - \bar{n})^2}. \tag{S10}$$

Here the sums are over all time bins, n_t is the number of data spikes in bin t, r_t is the output of the model in bin t, and $\bar{n}$ is the average spike count per bin of the data. The explained variance reaches its maximum of 1 in the case of an exact agreement between the two binned sequences, and is around 0 or less in the case of unrelated sequences. The bin size for the calculations was 1/60 s, which captures most of the dynamics of GC light responses in the amphibian retina [S10, S23]. Because the explained variance depends on the bin size and differs from cell to cell, the absolute values are not as important as the relative change in moving from one model to the other (Figure 2C,D). Specifically, the ratio of the variance explained by any given model to that of the LN model allows for a comparison of model performance across cells.

The explained variance in Eq.S10 is directly related to the signal power explained, that is, the part of the total power explained by the model that excludes the noise power [S21, S24]:

$$\frac{\text{signal power explained}}{\text{explained variance}} = \frac{\text{total power}}{\text{signal power}} = 1 + \frac{\text{noise power}}{\text{signal power}}. \tag{S11}$$

The total power is the variance of the observed spike train data (peri-stimulus time histogram; PSTH), the signal power is the variance of the deterministic part of the data (mean PSTH across trials under the assumption of additive independent and identically distributed noise), and the noise power is the variance of stochastic part of the data (trial-to-trial variability). The noise power is much smaller than the signal power in our data set (e.g., Figures 2A,B and S2B) and thus the signal power explained is nearly equal to the explained variance.

For each model on each GC, a free parameter search was carried out to maximize the objective function. Whereas many machine learning problems are solved by methods of stochastic gradient ascent, we chose a deterministic algorithm, because (1) the data set was small enough to be evaluated in its entirety at each step of the search and (2) the computation of gradients relative to the parameters is expensive, owing to the feedback loops in the networks. The Polak-Ribière variant of conjugate gradient ascent [S5] determines which direction in parameter space should be explored next. The code then performed a line minimization along that direction. The process of direction choosing and line minimizing was repeated iteratively until the objective function ceased to improve significantly. Each line minimization was accomplished in two stages. The first stage was "brute force": it proceeded by sampling 100 points spanning a domain of a carefully

determined length. This length was initially determined by our prior knowledge on the rough orders of magnitude of the various free parameters. Such choice of domain was not prescriptive, as the code would "zoom out" if it found that the explained variance near the edges was not low enough. In particular, it would zoom out if the maximum was too close to the edges. In addition, the code would "zoom in" if the points around the maximum did not approximate a parabola. The second stage of the line minimization employed the Brent's algorithm to narrow in on the exact optimal location along that line. This algorithm ran for a maximum of 20 iterations, but it rarely needed that many to converge.

Empirically, the multi-dimensional search worked better if the Polak-Ribière algorithm acted on subspaces of comparable free parameters (examples are the subspace of filter amplitudes, that of pooling weights, and that of delays) and then cycled through the subspaces iteratively. This is in contrast to running the algorithm on all free parameters simultaneously. Even though our subspace approach slowed down the search by forcing it to take a zigzag-like path through parameter space, this substantially improved convergence. On each subspace, the Polak-Ribière algorithm was allowed to run for at most 5 iterations. This number is low, but it is not so important as each subspace was revisited many times as we cycled through the subspaces. The number of cycles was fixed so that each subspace was visited 100 times. This number was deemed to be enough by observing that there were only minimal changes in the values of the free parameters (and in the objective function) after about 20 iterations.

We selected the initial conditions of the free parameters as follows (Figure S5). For the BCM filter and GCM pooling weights, the initial conditions were very loosely based on the receptive field of the cell in question, but were still quite different from the final values. For the nonlinearities, feedback functions, and delays, the initial conditions bore no resemblance to the final values of the parameters. If the search started in an approximately parabolic region, the convergence of the search algorithm is guaranteed [S5]; however, the initial brute force stage of the line minimization may fail. Therefore, after the runs were finished, all the line searches carried out were roughly inspected by eye to check that their shapes had a single clear maximum. To test for convergence, we also performed various kinds of numerical tests (Figure S5).

The field of machine learning develops fast and we acknowledge that there are many other approaches to large-scale smooth nonconvex optimization problems, such as automatic differentiation methods, stochastic gradient descent, and online preconditioner. We have not, however, tested such algorithms in this study.

Model assessment

Model performance was assessed by measuring the fractional variance of the GC firing rate explained by the model's output (Figure 2C,D), using Eq.S10 in a manner similar to that for model fitting. To avoid any type of over-fitting concerns, this was done on a separate testing data set. The testing data set included many repeats of the identical flicker sequence. The model's output was compared to the average firing rate observed over all these trials.

Note that the present model predicts only the trial-averaged firing rate, and makes no statement about the noise that leads to fluctuations from trial to trial. In fact, the experimental variability of firing was not the limiting factor in these model fits. Even with optimal parameter settings, the model showed systematic deviations from the data that exceeded the noise (see e.g., Figure 2B). More sophisticated circuit models

will be able to narrow that gap, at which point it will become useful to engage an explicit formalism for noise sources and how they affect the firing of ganglion cells.

For the data from the simultaneous BC-GC recording, the models did not always converge from all different initial conditions used, possibly due to small data sizes. In such cases, we selected the fitting results as the parameter set that produced the highest explained variance on the training data set. This selection was done entirely before analyzing the intracellular recording data, so no bias was introduced in the process.

Supplemental References

S1. Paninski, L. (2003) Convergence properties of three spike-triggered analysis techniques. Network *14*, 437–464.

S2. Lewi, J., Butera, R., and Paninski, L. (2009) Sequential optimal design of neurophysiology experiments. Neural. Comput. *21*, 619–687.

S3. Vaney, D. I., Sivyer, B., and Taylor, W. R. (2012) Direction selectivity in the retina: symmetry and asymmetry in structure and function. Nat. Rev. Neurosci. *13*, 194–208.

S4. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015) Human-level control through deep reinforcement learning. Nature *518*, 529–533.

S5. Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T. (1992) Numerical recipes in C: The art of scientific computing, Second Edition (Cambridge University Press).

S6. Schwartz, O., Pillow, J. W., Rust, N. C., and Simoncelli, E. P. (2006) Spike-triggered neural characterization. J. Vis. *6*, 484–507.

S7. Meister, M., Pine, J., and Baylor, D. A. (1994) Multi-neuronal signals from the retina: acquisition and analysis. J. Neurosci. Methods *51*, 95–106.

S8. Segev, R., Goodhouse, J., Puchalla, J., and Berry, M. J. (2004) Recording spikes from a large fraction of the ganglion cells in a retinal patch. Nat. Neurosci. *7*, 1154–1161.

S9. Pouzat, C., Mazor, O., and Laurent, G. (2002) Using noise signature to optimize spike-sorting and to assess neuronal classification quality. J. Neurosci. Methods. *122*, 43–57.

S10. Berry, M. J.,Warland, D. K., and Meister, M. (1997) The structure and precision of retinal spike trains. Proc. Natl. Acad. Sci. U S A *94*, 5411–5416.

S11. Baccus, S. A., and Meister, M. (2002) Fast and slow contrast adaptation in retinal circuitry. Neuron *36*, 909–919.

S12. Asari, H., and Meister, M. (2012) Divergence of visual channels in the inner retina. Nat. Neurosci. *15*, 1581–1589.

S13. Asari, H., and Meister, M. (2014) The projective field of retinal bipolar cells and its modulation by visual context. Neuron *81*, 641–652.

S14. Borges, S., and Wilson, M. (1987) Structure of the receptive fields of bipolar cells in the salamander retina. J. Neurophysiol. *58*, 1275–1291.

S15. Baccus, S. A., Ölveczky, B. P., Manu, M., and Meister, M. (2008) A retinal circuit that computes object motion. J. Neurosci. *28*, 6807–6817.

S16. Zhang, A.-J., and Wu, S. M. (2009) Receptive fields of retinal bipolar cells are mediated by heterogeneous synaptic circuitry. J. Neurosci. *29*, 789–797.

S17. Gollisch, T., and Meister, M. (2008) Rapid neural coding in the retina with relative spike latencies. Science *319*, 1108–1111.

S18. Chichilnisky, E. J. (2001) A simple white noise analysis of neuronal light responses. Network *12*, 199–213.

S19. Meister, M., and Berry, M. J. (1999) The neural code of the retina. Neuron *22*, 435–450.

S20. Herz, A. V. M., Gollisch, T., Machens, C. K., and Jaeger, D. (2006) Modeling single-neuron dynamics and computations: a balance of detail and abstraction. Science *314*, 80–85.

S21. Sahani, M., and Linden, J. F. (2003) How linear are auditory cortical responses? In Advances in neural information processing systems, S. Becker, S. Thrun, and K. Obermayer, eds. (Cambridge, MA: MIT Press), pp. 109–116.

S22. Pillow, J. W., Paninski, L., Uzzell, V. J., Simoncelli, E. P., and Chichilnisky, E. J. (2005) Prediction and decoding of retinal ganglion cell responses with a probabilistic spiking model. J. Neurosci. *25*, 11003–11013.

S23. Keat, J., Reinagel, P., Reid, R. C., and Meister, M. (2001) Predicting every spike: a model for the responses of visual neurons. Neuron *30*, 803–817.

S24. Schoppe, O., Harper, N. S., Willmore, B. D. B., King, A. J., and Schnupp, J. W. H. (2016) Measuring the performance of neural models. Front. Comput. Neurosci. *10*, 10.