Patterns

Article

Unraveling neural coding of dynamic natural visual scenes via convolutional recurrent neural networks

Highlights

- Learning relationship between retinal response and complex visual scenes
- We evaluate quantitatively the complexity of stimuli and spatiotemporal regularity of RFs
- The proposed CRNN can reveal the shapes and locations of receptive fields of RGCs
- The proposed CRNN outperforms CNNs in predicting large populations of RGCs

Authors

Yajing Zheng, Shanshan Jia, Zhaofei Yu, Jian K. Liu, Tiejun Huang

Correspondence

yuzf12@pku.edu.cn (Z.Y.),
j.liu9@leeds.ac.uk (J.K.L.)

In brief

Visual neuroscience is an immensely popular topic in AI, such that numerous methodologies developed in visual computing have broad applications and provide inspiration for other domains. The retina is one of the best-understood examples in neuroscience for visual computing. Here, we use retinal data to demonstrate how to use deep-learning models to encode dynamic visual scenes. The proposed models demonstrate that recurrence plays a critical role in encoding complex natural scenes and learning the biological computational underpinning of the neural circuits.

Zheng et al., 2021, Patterns 2, 100350
October 8, 2021 © 2021 The Author(s).
https://doi.org/10.1016/j.patter.2021.100350

CellPress

Patterns

CellPress
OPEN ACCESS

Article

Unraveling neural coding of dynamic natural visual scenes via convolutional recurrent neural networks

Yajing Zheng,[1] Shanshan Jia,[1] Zhaofei Yu,[1,2,*] Jian K. Liu,[3,4,*] and Tiejun Huang[1,2]

[1]Department of Computer Science and Technology, National Engineering Laboratory for Video Technology, Peking University, Beijing 100871, China
[2]Institute for Artificial Intelligence, Peking University, Beijing 100871, China
[3]School of Computing, University of Leeds, Leeds LS2 9JT, UK
[4]Lead contact
*Correspondence: yuzf12@pku.edu.cn (Z.Y.), j.liu9@leeds.ac.uk (J.K.L.)

https://doi.org/10.1016/j.patter.2021.100350

THE BIGGER PICTURE Understanding surrounding environments perceived by eyes requires unraveling the computational principle embedded in the neural system. Recently, deep learning has been implemented to develop useful models of the visual system for studying simple and static scenes. Yet, we perceive continuous dynamic scenes in an ever-changing environment, which cannot be captured by standard convolutional neural networks (CNNs). Here, we use the retina as a model system to demonstrate how recurrence helps to explain the relationship between neural response and complex natural scenes. Leveraging CNNs with different types of recurrence, we highlight the role of recurrence in the neural coding of dynamic visual scenes, not only better predicting the neural response, but also revealing the corresponding biological counterparts. Our results shed new light on unraveling the coding principle of visual neurons for dynamic scenes and provide a way of using recurrence for understanding visual computing.

1 2 3 4 5 **Proof-of-Concept:** Data science output has been formulated, implemented, and tested for one domain/problem

SUMMARY

Traditional models of retinal system identification analyze the neural response to artificial stimuli using models consisting of predefined components. The model design is limited to prior knowledge, and the artificial stimuli are too simple to be compared with stimuli processed by the retina. To fill in this gap with an explainable model that reveals how a population of neurons work together to encode the larger field of natural scenes, here we used a deep-learning model for identifying the computational elements of the retinal circuit that contribute to learning the dynamics of natural scenes. Experimental results verify that the recurrent connection plays a key role in encoding complex dynamic visual scenes while learning biological computational underpinnings of the retinal circuit. In addition, the proposed models reveal both the shapes and the locations of the spatiotemporal receptive fields of ganglion cells.

INTRODUCTION

Unraveling the neural system of the brain is one of the key questions of both neuroscience and artificial intelligence, as understanding the structure of neural systems could help to develop novel methodologies of artificial intelligence. The visual system constantly receives highly complex and dynamic visual scenes with a high order of spatiotemporal correlations. To cope with these inputs, it is necessary to develop an explainable neural network model, either for explaining the data of neuroscience, e.g., the neural response to input scenes,[1] or for developing an efficient computational framework for analyzing dynamic visual scenes for artificial vision.[2]

The retina, as the first stage of the visual system, encodes visual information from the external environment in both spatial and temporal domains.[1,3] It consists of three layers of neurons, namely, excitatory photoreceptors (input), bipolar cells, and ganglion cells (output), with inhibitory horizontal and amacrine cells communicating within the bipolar and ganglion cell layers, respectively. At the output side of the retina, i.e., the retinal ganglion cells (RGCs), all input signals are transformed into a sequence of spikes. These spikes are then transmitted via the optic nerve to the visual processing center of the brain. The retina receives approximately 100 MB per second of visual input[4] and sends approximately 1 MB per second of visual data to the brain from 10^6 RGCs.[5] Therefore, the retina must be "smart" enough to efficiently encode the input stimuli.[1] Exploring the encoding mechanism of the retina is essential to unravel the computational principles of other visual systems.

Recent achievements in deep learning have led to renewed interest among researchers using convolutional neural networks (CNNs) to investigate topics in systems neuroscience.[6–8] CNNs have been used to build the most quantitatively accurate models in predicting neural responses.[9–13] In addition, deep-learning-based methods[14–17] have been proposed to model retinal systems and have made remarkable progress in analyzing visual scenes, including those composed of artificial stimuli (e.g., moving bars) and static natural images. These studies have revealed that novel functional neural networks can encode simple and static visual scenes by analyzing the patterns of dynamic responses of RGCs. However, modeling the retina to process dynamics of rather complex natural scenes by deep neural networks remains unclear.[18]

Studies on models of the visual cortex have highlighted the role of recurrent connections in visual processing[19–21] within the models themselves. These connections help "fill in" missing data,[22–25] indicating that the real visual cortex allows the brain to "predict" future stimuli.[26–28] In addition, the retina, known as an efficient encoder, can anticipate motion with recurrent connections.[29] The RGCs can be connected laterally by electric synapses, i.e., gap junctions[30–33] or specific amacrine cells. The lateral connection allows the retina to detect the differential motion of the object and background,[34] while specific asymmetric connectivity of the amacrine cells helps the RGCs show direction selectivity.[35] These characteristics of gap junctions and recurrent connections play a critical role in the efficient encoding of dynamic visual scenes by the retina.[36,37]

Therefore, recurrent connections can be a potential element for understanding the neuronal encoding of visual scenes in the retina, which is beyond the capability of the feedforward approach.[14,15] The disadvantage of the CNN is that the final fully connected layer maps the convolutional feature space to individual cells' responses, leading to a dramatic increase in the number of model parameters with the increase in the number of neuron inputs. In addition, the CNN models of retinal encoding[14–16] typically learn only the relationship between a stimulus covering a small field of view and the subsequent response of the RGCs. Traditional models for learning retinal coding, such as the generalized linear models,[38] incorporate several linear or nonlinear filters that model each neuron and a set of coupling filters that capture the neurons' dependencies in the recent activity of other cells. This type of model is more closely related to the way in which a population of the RGCs encodes an external stimulus. Some recent studies have explored the role of recurrence,[39] using recurrent neural networks (RNNs) to model the shared feature space within the population of neurons. However, the performance of this approach depends critically on the initial location estimate.

To fill in this gap with an explainable model that reveals how the population of neurons work together to encode a larger field of dynamic natural scenes, in this study, we propose that the computations carried out by the retina could be better explained by a convolutional RNN (CRNN) rather than a feedforward CNN. We explore deep CNN and CRNN models with natural scenes consisting of a larger field as input. This approach allows us to determine the shared features of the RGCs and the way they cooperate to encode an external stimulus. The CRNN utilizes many fewer model parameters than the feedforward CNN to directly map out the receptive fields (RFs) of each neuron in the population from dynamic natural visual scenes, producing an outcome that is robust to individual stimulus videos and RFs comparable to those recorded by experiments. Visualization of the results shows that output neurons of the model can learn both the underpinning spatiotemporal RFs of the corresponding RGC and their locations. Furthermore, using a novel pruning strategy for convolution kernels, we find that the CRNN produces a highly effective subset of kernels that capture the performance of the full model. Altogether, these results would inspire researchers to improve the deep-learning strategies for modeling and analyzing dynamic visual scenes.

RESULTS

RGC-encoding model with recurrent connections

In this work, we propose a model consisting of both feedforward convolutional filters and recurrent units. The inputs and target outputs of the model are the natural scene movie stimuli and the responses simultaneously recorded from a population of the RGCs with an electrode array[40] (see experimental procedures for details of the data). To better study the working principle of the encoding of external input stimuli by the retina, we introduce recurrent connections based on a CNN to get closer to the anatomy of the retina, i.e., the lateral connection, between the RGCs by gap junction or amacrine cells. The CRNN model consists of a four-layer network, including two convolutional layers (model bipolar cells and amacrine cells) that use a rectified linear unit as the activation function. The recurrent connection layer (model lateral connection by gap junction or amacrine cells) is added before the last fully connected layer (model RGCs). The framework of the proposed CRNN model is shown in Figure 1, where one recurrent layer is added to the CNN to capture the temporal dynamics shown in the continuous natural videos and neural responses. The units in the recurrent layer can have a variety of structural forms (e.g., vanilla RNN, LSTM, or GRU, see below for more detailed comparison). Except for special instructions, we use the long-short-term memory (LSTM) units throughout our results. The complexity of the unit structure of the recurrent layer does not have much influence on the performance of the CRNN model.

RF subunits learned by the encoding models

To quantify the performance of the models, we evaluated the predicting performance of the neural response against various input stimuli, and explored whether the model parameters can conform to one of the critical characteristics of the retinal cell, namely, its RFs. Analysis of the RFs of the hidden layer parameters and the neurons in the last fully connected layer would help

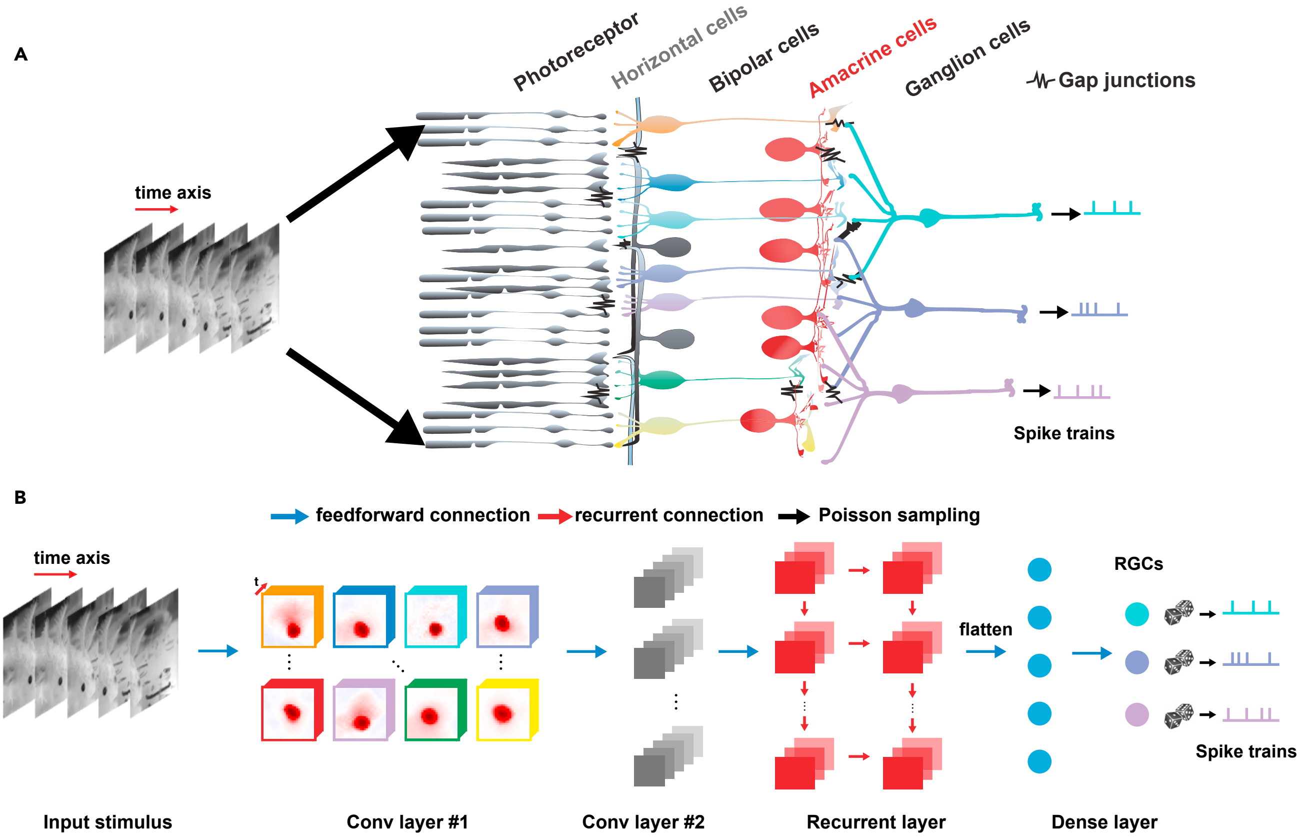

Figure 1. Illustration of the CRNN model architecture

(A) Schematic diagram of the retinal circuit.

(B) A continuous input stimulus is convolved with the first convolutional layer consisting of several spatiotemporal filters, followed by another convolutional layer that integrates the resulting feature maps. A recurrent layer is incorporated after the last convolutional layer to capture the relationship between the dynamic natural scene stimulus and the retinal response. The activity sequence of the recurrent layer is linearly combined and passed to the final nonlinear activation function for the prediction of the individual RGC responses. Conv layer #1, first convolutional layer; Conv layer #2, second convolutional layer.

us gain a greater understanding of the influence of the recurrent connection layer on model parameter learning.

To verify whether the models developed an intermediate computational mechanism similar to a biological retinal circuit by learning the transformation between the input stimulus and the neural response, we generated eight RF subunits and grouped them to create a network model of two RGCs, as shown in Figure 2A, with handcrafted spatial and temporal filters. The RFs of these subunits and the subsequent composition of the RGCs are shown in Figure 2B. To simultaneously encode the response of the two RGC units and their subunits, the first convolutional layer is created with eight spatiotemporal filters, and the dense layer is constructed with two neurons. In addition, 8 × 8 pixel white-noise images are generated as the input of the network. We visualize the RFs of the kernels in the first convolution layer, and the neurons in the dense layer.

To explore the effect of the recurrent connections on the output of the models, we compare the performances of the CRNN and the CNN. To ensure a fair comparison, all the parameters and structure settings of the CNN are kept consistent with the CRNN except for the inclusion of the recurrent layer in the latter. As there is no temporal correlation within the white-noise stimuli, the correlations between the responses of the RGCs and the outputs of both CRNN and CNN reach approximately 0.99 without a significant difference between the models, and both models can learn the spatiotemporal RFs of the two RGCs (Figure 2C), which are computed by the standard techniques of spike-triggered average (STA).[41] As shown in Figure 2D, the subunits obtained by the CNN and CRNN closely match those given in the model cell. We also altered the size of the kernels in the models and found that we can more effectively map out the RF subunits with relatively large kernels than those with smaller kernels (Figure S1). If the size of the convolution kernel is set relatively small, certain subunits with similar shapes but distributed in different spatial locations can be multiplexed by certain convolution kernels, for example, the subunits shaped like square blocks in the first and fifth subunits.

To further verify the properties of the convolutional kernels learned from the models with different settings, we calculate the spatial autocorrelation as an index of spatial regularity.[42] We then propose a novel index to describe the temporal regularity (Equation 3 in experimental procedures) of the kernels. Figure 2E shows the distribution of both indices of the RF subunits of model data and for the filters learned in the CNN and CRNN,

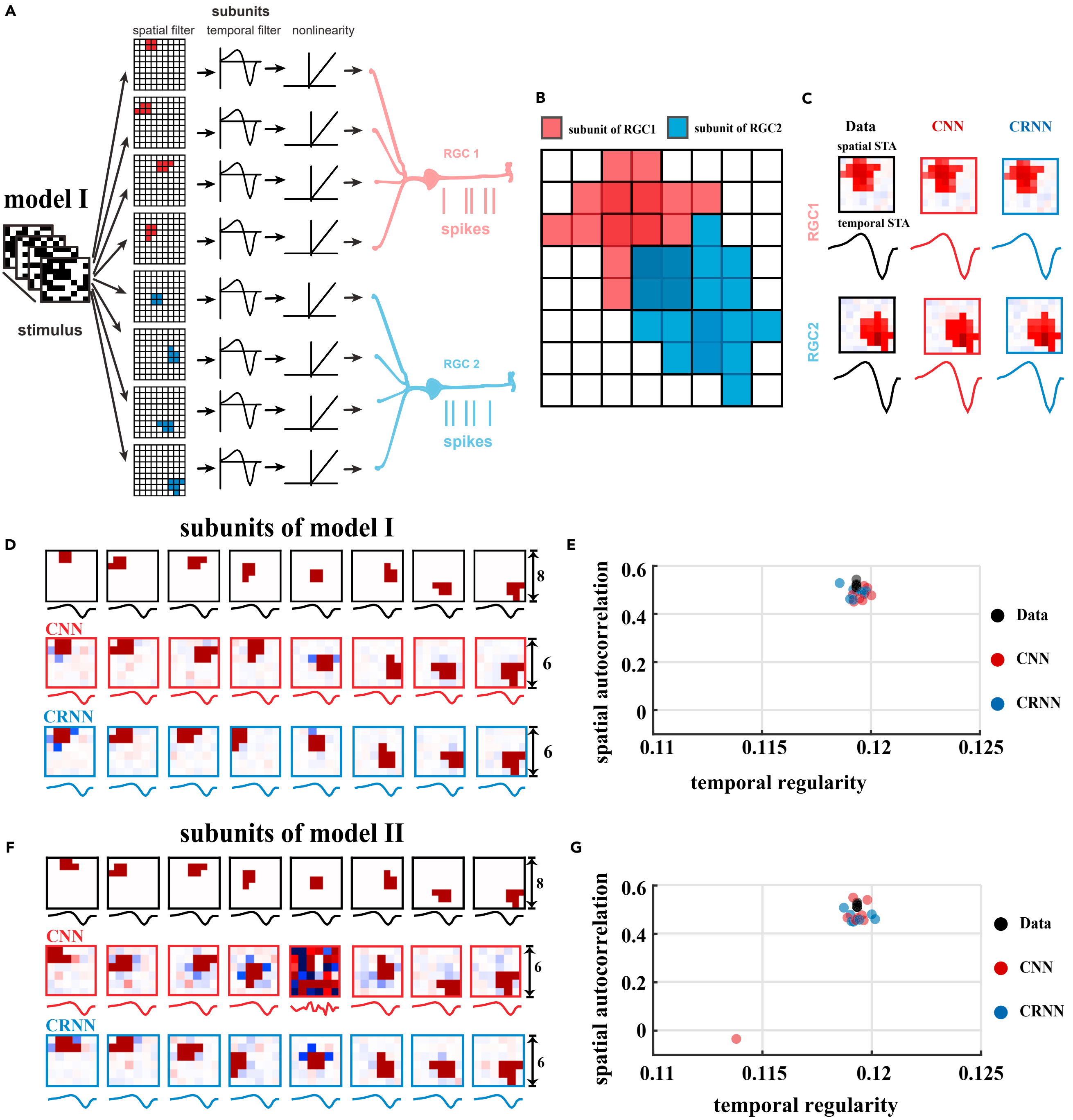

Figure 2. Subunit structures of the two RGC models revealed by the CNN and CRNN models

(A) RFs of the two RGCs.

(B) Overlaid RF subunits of the two RGCs.

(C) Comparison of the measured spatial and temporal STAs of the two RGCs with those predicted by the CNN and CRNN.

(D) The RF subunits of model I with the convolutional filters learned in the CNN and CRNN models, with a kernel size of 6.

(E) Spatial autocorrelation versus temporal regularity of the model and convolutional filters in the models.

(F and G) Similar to (D) and (E), respectively, but for model II, in which the first subunit is changed, followed by altering all eight RF subunits in the models. RF, receptive field; STA, spike-triggered average.

demonstrating that both models preserved the subunits of the model cell well, indicating that the filters can be effectively learned and that the indices of spatial and temporal regularities can characterize the importance of the RF subunits well.

Our results are robust to the use of ℓ_1 weight regularization, which results in the regularization of the subunits with more compact shapes (Figure S1). Thus, we apply regularization throughout the techniques below. To further examine the robustness of our models, we simulate another network model of the RGCs with different subunit shapes. We manipulated the first subunit (model II) shown in Figure 2F, and found that CRNN is more robust to subunit variations than CNN, in that the CRNN can robustly learn the eight spatiotemporal filters corresponding to the subunits with proper model settings, while the CNN fails to do so to the same degree (Figures 2F and 2G). These results indicate the critical role of recurrent connections in the CRNN to better capture the underlying computational components of the RF subunits of the retinal neural network.

CRNN enhances the encoding of retinal responses to dynamic natural stimuli

To verify the performance of the CRNN model regarding the electrophysiological data and evaluate whether the models can learn the adaptability of the retina to dynamic visual scenes, we further built models for the prediction of the response of a population of RGCs to natural movies. Two natural movies approximately 60 s long were used to train the models. The first movie (movie 1) was relatively simple, consisting merely of scenes of salamanders swimming in a tank. In contrast, the second (movie 2) was more complex, showing a tiger hunting its prey, in a backdrop of grass and trees, and with fast transitions between scenes. Example frames of the two natural movies and the corresponding RGC responses in terms of individual trials of spike trains as well as trial-averaged firing rate are shown in Figure 3A, together with the model output in the format of firing rate, from which one can sample individual spikes using the Poisson process (Table 1).[38]

After $\boldsymbol{h}_t$ network training, we evaluated the correlation coefficient (CC) between the response of each RGC and the output of the neurons in the dense layer, as well as between the average firing rate of the output neurons in the models and the average response of the RGCs. The performance of the models with respect to the RGC population is shown in Figures 3B and 3C, where the CRNN markedly outperforms the CNN on both movies (average CC of RGCs 0.857 versus 0.698 on movie 1, 0.718 versus 0.623 on movie 2), independent of the size of the training data (Figure S2). This improvement is also true when training the models with individual trials of spike trains (Figure S3). Particularly, for movie 1, the CRNN performs notably better than CNN, which may be because movie 1 is visually less complex than movie 2. Subsequently, to evaluate the complexity of dynamic natural scenes quantitatively, we characterized the spatial and temporal complexities of the scenes by calculating the structural similarity index (SSIM) between patches of the frames of movies, and compared them between the two movies in Figure 3D, which indicates that movie 2 was more complex than movie 1. We then examined the relationship between the complexity of the dynamic visual scenes (see Figure S4 and experimental procedures) and the performance of the models in terms of the CCs between the individual RGCs and the model outputs. As shown in Figure 3E, the performance of the models was lower for the more complex movie 2 than for movie 1, which suggests that the complexity of the visual scenes is indeed a major driving force for modeling prediction.

CRNN recovers the neuronal RF using dynamic natural scenes

In addition to evaluating the ability of the model to predict responses, we further show whether the structural components of the models can capture the intermediate computational mechanism of the retinal encoding circuit for dynamic scenes. Figure 4 shows the RFs of the RGCs and those that the CNN and CRNN learned when trained on movie 1 and movie 2. Experimentally, the RFs were computed with the STA obtained from white-noise stimuli to obtain a 3D spatiotemporal RF filter. Thereafter, we applied singular-value decomposition (SVD) to the 3D filters to obtain a temporal filter and spatial filter (Figure 4A). Two-dimensional Gaussian functions were fitted to the components of the spatial RFs obtained from both data and models to determine their center, size, and shape. Figures 4A and 4B show the fitted 2D Gaussian function of each RGC and model neuron as ellipses.

To better quantify the similarity of the RFs between the RGC data and the models obtained for each cell, we calculated the cosine distance between the 2D Gaussian distribution of the RFs of the model neurons and those of the recorded RGCs (Figure 4C). While calculating the RF distance, we considered only the neurons that are able to learn an RF. For example, the neurons indicated in the third column of Figure 4A, which do not learn an effective spatial RF, are not included in the statistical analysis. The model with the best performance (the CRNN model trained on movie 1) best reproduced the RFs of a large number of the RGCs with a notably small distance between the RFs of the data and the model, while the CNN model trained on movie 2 was unable to learn either the spatial or the temporal filters. Moreover, as the temporal correlations in movies of natural scenes are much higher than those found in white-noise stimuli, the temporal filters obtained by the models trained on these movies adapted better than the filters calculated by white-noise stimuli. The models that learned the temporal filters usually produced filters whose first peak had a low temporal latency, while the peaks of the temporal filters obtained using white-noise stimuli had much longer latency, which is a peculiar feature of the temporal adaptation induced in the retina by stimulus images with different statistics.[43] These results indicate that the CRNN, and not the CNN, can model the rich computational structures of the retinal neural circuit while learning the complex dynamic visual scenes.

Efficient learning of the CRNN model

In the previous sections, we have described how the introduction of the recurrent layer can improve the model performance in predicting the retinal response to dynamic sequence stimuli and the robustness of learning to infer the subunits. In addition to the response prediction performance, quantification of the effectiveness of CRNN on the retinal electrophysiological data is a key issue in evaluating a neural coding model. We compare the CNN and CRNN from two aspects: inferring the subunits of the retinal circuit and learning to predict the responses of large-scale

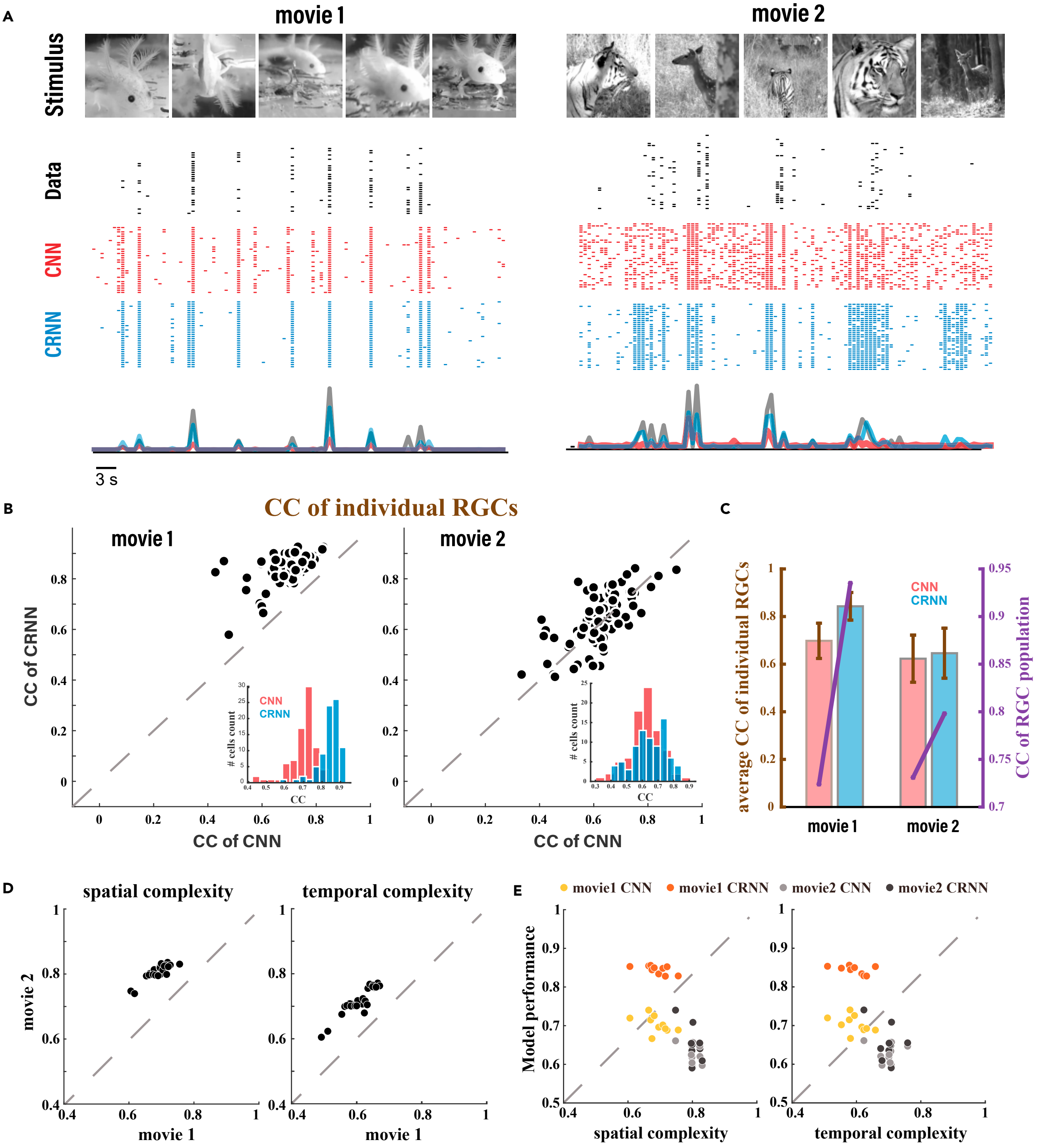

Figure 3. Performance of the models in response to natural movies

(A) Spikes and firing rate of the responses of a representative recorded RGC and the predicted CNN and CRNN responses to two natural movies.

(B) Scatterplots of the CCs between the electrophysiological data of all RGCs and the responses of both CNN and CRNN models to the two movies.

(C) Average CC in response to both movies between the RGCs and the models (left y axis), and CC between the population RGC response and the average of the output of models (purple line belongs to the right y axis).

(D) Different levels of complexity between the two movie stimuli. Each dot represents a slice patch used for computing the correlation.

(E) Relationship between the complexity of the movies and the performance of the models. The points in (B), (D), and (E) refer to different cells.

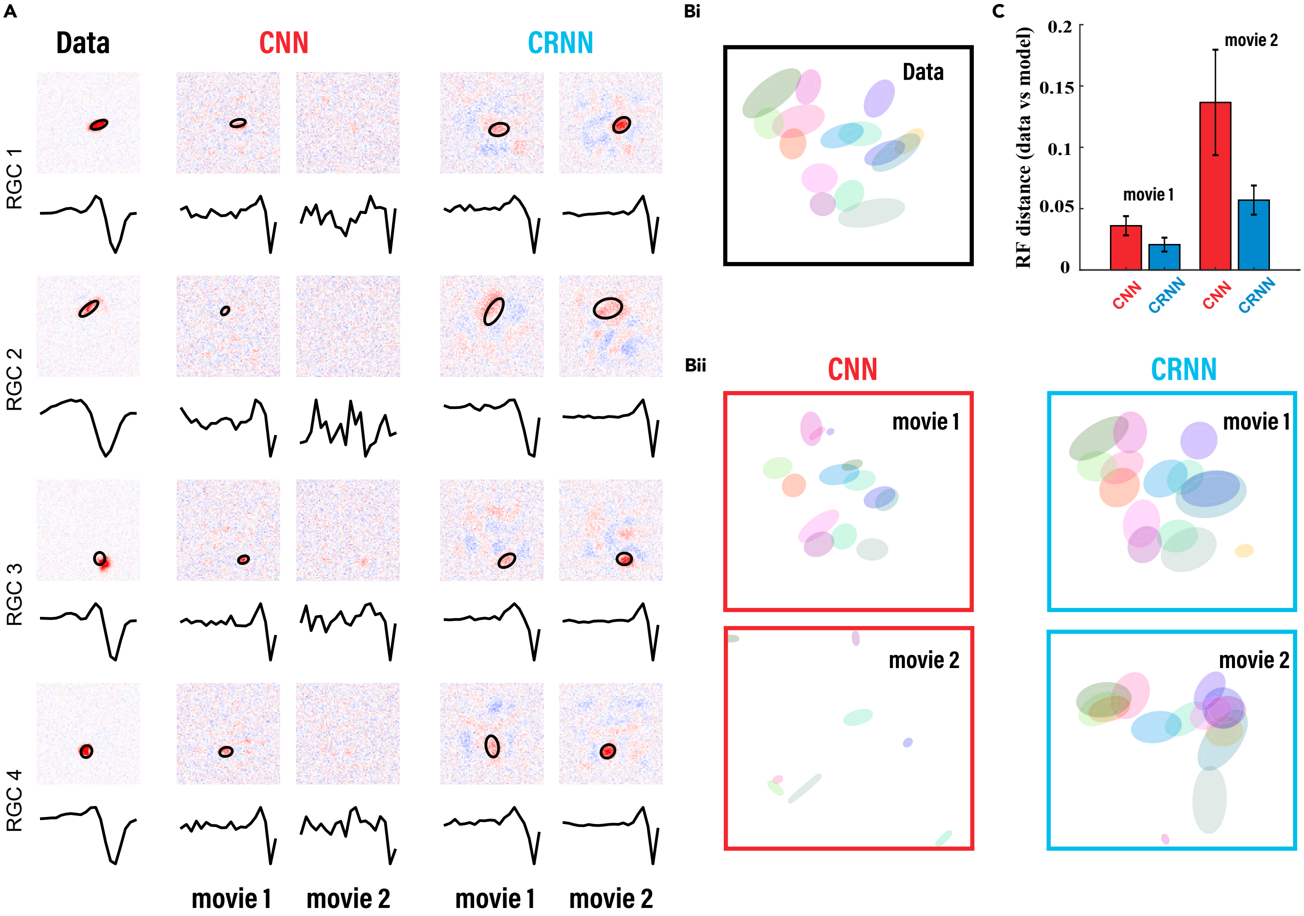

Figure 4. The CRNN reproduces the spatial RFs and temporal filters of RGCs

(A) The spatial RFs and temporal filters of four representative RGCs and the corresponding neurons learned by the CNN and CRNN. The ellipses indicate the RFs fitted by a 2D Gaussian function.

(B) The RFs of 15 RGCs randomly selected from a population of 80 cells from the dataset (i) and those generated by the models (ii) computed from white-noise stimuli.

(C) RF distance between the data and the models for all RGCs.

population ganglion cells. They are used to evaluate whether the introduction of the recurrent layer can improve the effectiveness of the retinal coding model.

First, we evaluated the kernel parameters of the first convolutional layer trained on the movies according to the subunit importance indices, the spatial autocorrelation, and the temporal regularity (shown in Figure 5A). The spatial autocorrelation can measure whether the spatial filter of the convolutional kernel is relatively concentrated in a certain area, while the temporal regularity can measure the adaption regularity of the temporal filter. When we construct reduced/pruned models using fewer subunits selected according to the value of either index, the performance is found to be better preserved in models pruned based on the temporal regularity index. The pruning results with different numbers of convolutional filters quantified by both the spatial and the temporal indices are shown in Figure 5B. The performance of the reduced models can be maintained at a good level regardless of the temporal regularity indices of the remaining subunits. In contrast, when using the spatial autocorrelation, the performance of the reduced CNN model significantly drops when the number of convolution kernels is less than 32. Moreover, it is interesting to note that by using temporal filter regularity as a quantified index, the reduced CNN model can still achieve high performance on both movies. For the CRNN, the convolution kernels learned in the model are better than those in the CNN model; thus, the prediction performance can be maintained at a better level when unimportant convolution kernels are removed, especially for the model with the best performance in movie 1. A few examples of selected filters sorted in terms of decreasing temporal regularity are illustrated in Figure 5C, indicating that the filters are more organized in movie 1 than in movie 2. Altogether, these results signify that the temporal regularity is prioritized in learning dynamic visual scenes, and the CRNN enables us to implement efficient learning with a superb representation of the retina, even with a much smaller set of learned parameters.

Thus far, we have used a population of 80 RGCs from a single recording to serve as our electrophysiological dataset. To

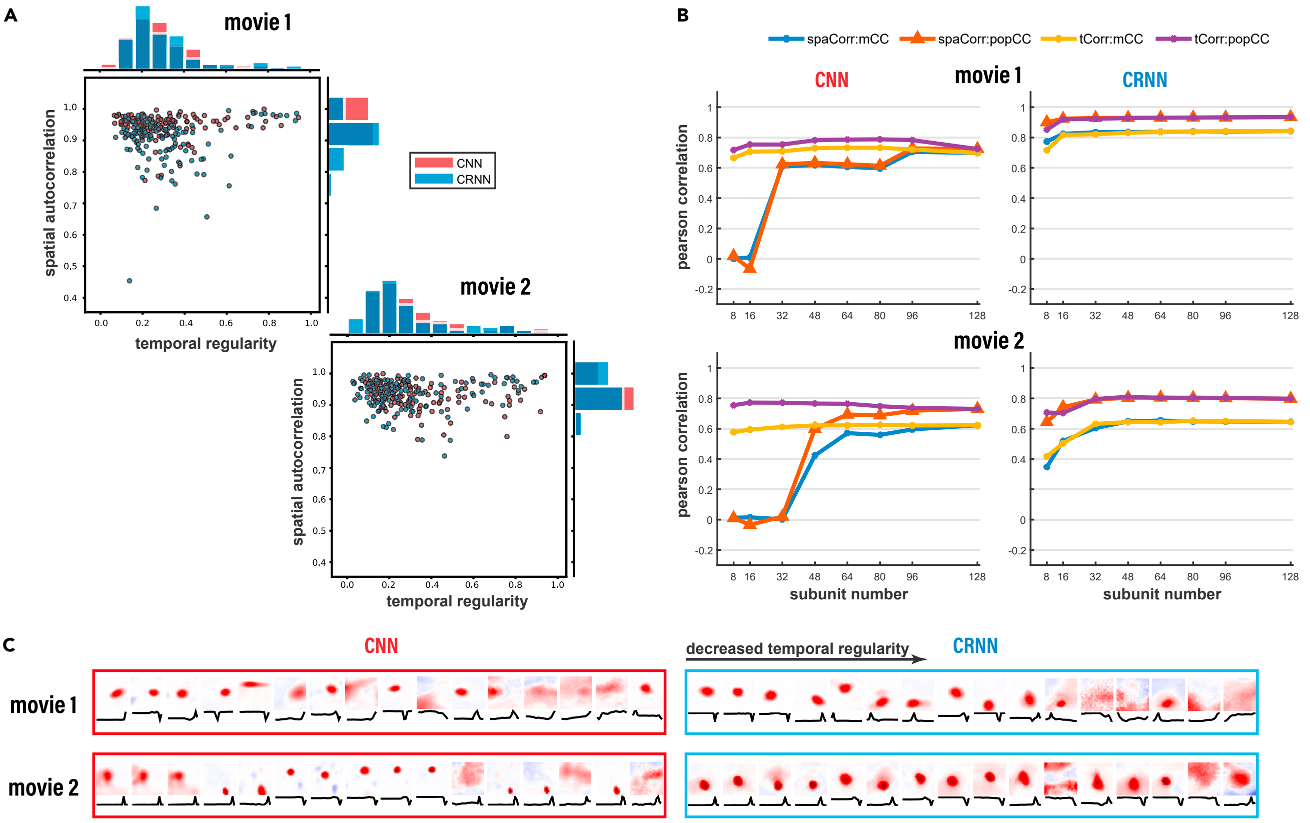

Figure 5. Model construction using effective components with highly temporal regularity or spatial autocorrelation

(A) Distribution of the spatial autocorrelation and temporal regularity indices of all filters learned from both movies by the CNN and CRNN.

(B) Performance (CC) on both movies with reduced models incorporating rank-selected filters. The average individual CCs (mCC) and the population response CC (popCC) between the data and the pruned models are shown.

(C) Convolutional filters of the CNN and CRNN models sorted by decreasing the temporal regularity. spaCorr, spatial autocorrelation; tCorr, temporal regularity.

further explore the effect of the amount of data size on model learning, we obtained a second dataset involving 14 recordings from 1,218 RGCs for movie 1 and 500 RGCs for movie 2, and used this dataset to again train the models. In addition, we evaluated the influence of the number of hidden units in the recurrent layers on the performance of the models. Table S1 shows the number of parameters used in the models constructed for this set of experiments: the existing CNN model described above and CRNN models constructed with 32, 64, 128, and 256 recurrent units. Following the training of these models using the second dataset mentioned above, the CRNN models were found to outperform the CNN models in both movies (Figure 6A), similar to the results reported in the above subsections. However, we found that more recurrent units are not always better; eventually, a large number of recurrent units result in a deteriorated model performance. For movie 1, the CRNN model achieved good performance with 32 recurrent units. As the number of units increased to 64 and then 128, the performance of the CRNN models slowly increased. For movie 2, the CRNN model did not perform well with 32 recurrent units and achieved the best performance with 64 units, implying that the optimal CRNN model for movie 2 requires 64 recurrent units, while the performance for movie 1 is relatively equitable with 32–128 units.

To examine the RFs learned by these models, we first computed them using white-noise images as described earlier, followed by calculating the spatial autocorrelation and temporal regularity of the RFs of the output neurons. The average values of these indices are shown in Figure 6B. Some examples of the spatiotemporal RFs learned by the models with the highest spatial autocorrelation (left) and temporal regularity indices are shown in Figure 6C. The CRNN models with the best performance on movie 1 (CRNN-128) and movie 2 (CRNN-64) exhibit more regularized RFs (spatial centralized, regular oscillatory temporal wave) than other models. For the CRNN with 256 recurrent nits, the centralized area of the spatial filters was much larger than that of other models, while the spatial filter had less diversity, or in other words, was more uniform. This observation suggests that using more recurrent units can yield output neurons with similar RFs, which can be combined with several small RFs in the same region. In addition, when training with the complex scenes of movie 2, some spatial RFs of the CRNN exhibited complex tuning beyond spatially localized center-surround tuning, which could be due to the overly dense representation in the collated population of the RGCs used for training the model. Taken together, these results suggest that a CRNN with more RGC samples could achieve a nearly 100% perfect

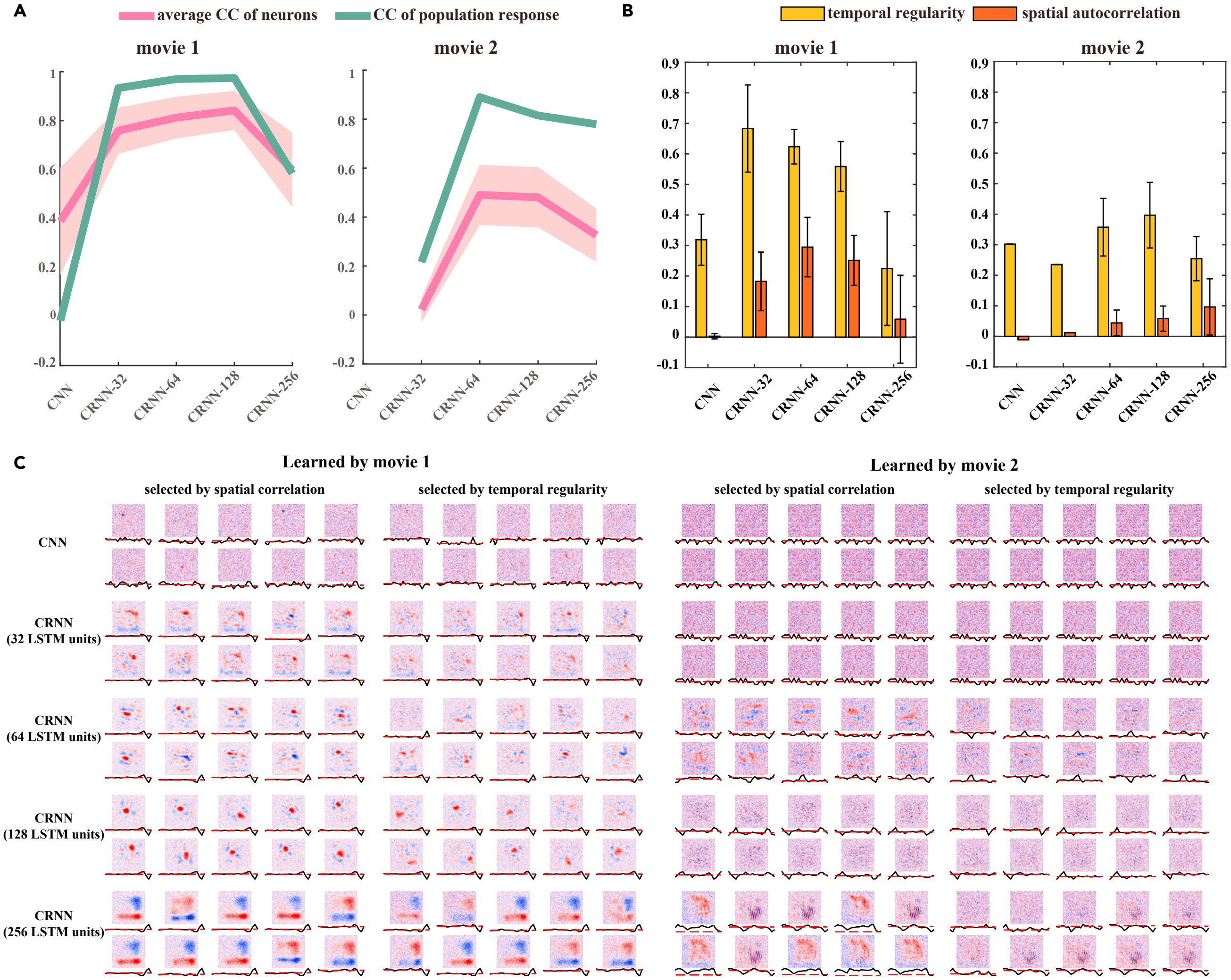

Figure 6. Model behavior of a large population of RGCs

(A) Performance of CNNs and CRNNs with different numbers of recurrent units. Red and green indicate the average CC of individual cells and the CC of the population response, respectively.

(B) Distribution of the temporal regularity and spatial autocorrelation of the RF filters from all neurons learned from the different models. The error bars represent SD.

(C) Examples of the RFs learned from the different models. For each model, 10 spatial and 10 temporal filters were selected and ranked by the spatial autocorrelation and temporal regularity, respectively, for each movie.

performance, while a feedforward CNN would be unable to learn the response of a large group of RGCs. Moreover, the CRNN could be capable of demonstrating robust performance with a small number of recurrent units.

Different structures of the recurrent layer

To show that our results are not dependent on one specific type of recurrent structure, we tested and compared three types of recurrence: vanilla RNN, gated recurrent unit (GRU),[44] and LSTM.[45] Their structures are shown in Figure 7A. The predicting performances of these three structures for natural movies are similar (Figure 7C). In addition, we examined whether the models could obtain the spatiotemporal RF of the RGCs, and some example results are shown in Figure 7B. By calculating the cosine distance between the 2D Gaussian distribution of the RFs of the models and those of the recorded RGCs, we found that the models with LSTM units could obtain RFs with higher similarity with the RGC data (Figure 7D). Overall, these results demonstrate that CRNN models with different kinds of recurrent units can achieve comparable performance and outperform the CNN model. In other words, the difference in performance between CNN and CRNN is not due to the complexity of the LSTM/GRU units, but the recurrent layer is essential. However, considering the maintenance of the model's ability to learn long-term input stimuli, the vanilla RNN models are prone to gradient disappearance/explosion when receiving long-term stimuli, and the performance of the model based on LSTM is better than that based on GRU, as well. Therefore, the results of the CRNN models obtained in the above section were constructed with the LSTM. These results indicate that the recurrent layer

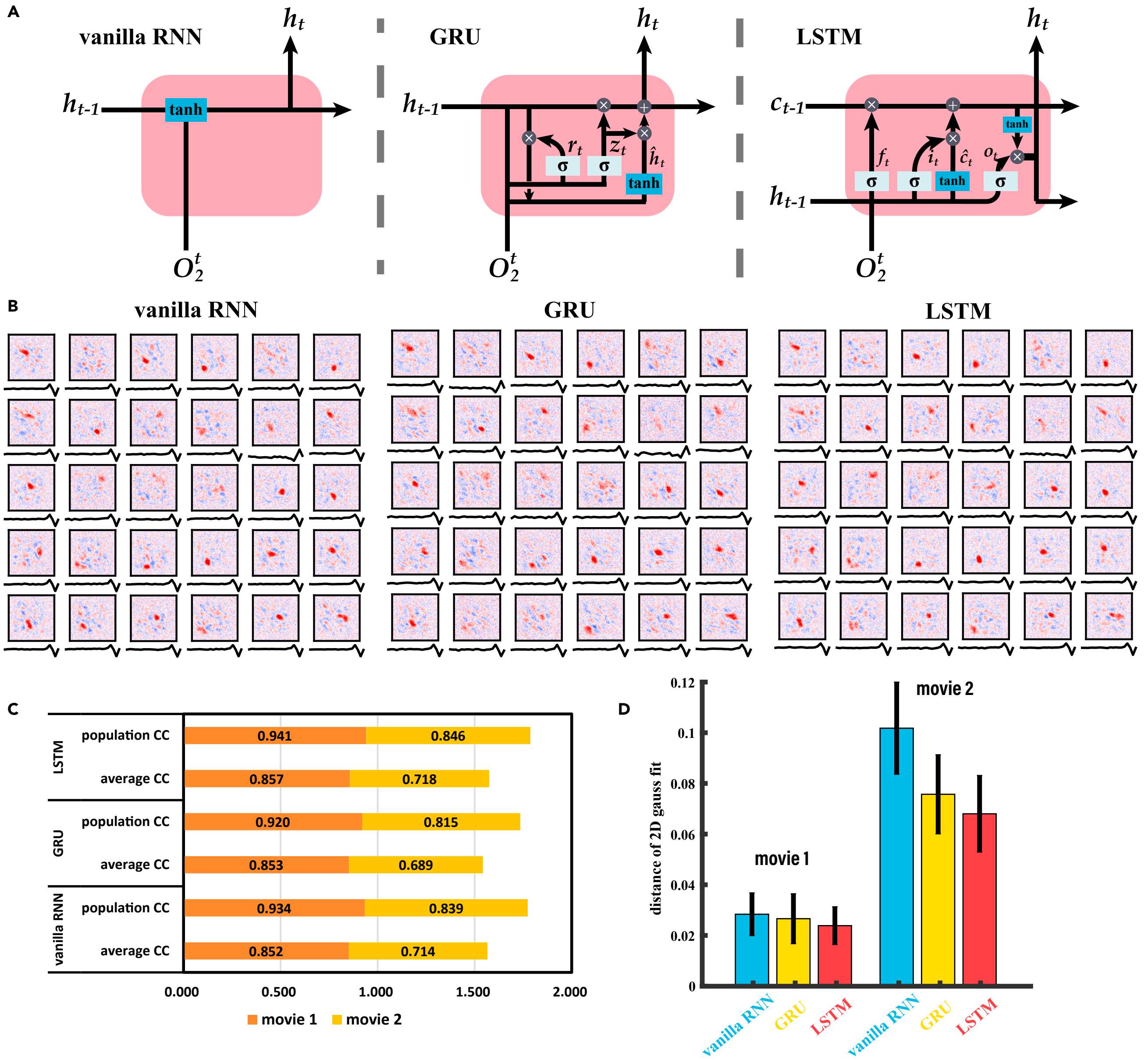

Figure 7. The CRNN models with different recurrent layers

(A) Structures of different kinds of recurrent units.

(B) Spatial RFs and temporal filters of neurons learned by the CRNN with vanilla RNN, GRU, and LSTM.

(C) Performance of the CRNN models based on different recurrent units to predict the neural response to two natural video stimuli (movie 1 and movie 2). Population CC, the correlation coefficient between population RGCs and the average output of the models; average CC, the correlation coefficient between the RGCs and the models' outputs.

(D) RF distance of each cell between the data and the model outputs.

as a general form, rather than specific models of recurrent units, plays a functional role in explaining neural responses to dynamic visual scenes.

DISCUSSION

To unravel how the retina computes dynamic natural visual scenes, we have investigated the role of recurrent connection in encoding complex dynamic visual scenes in the retina in this study. Using both simulated and experimental data, we observed that the CRNN was more effective than the CNN in predicting the response to the movies, as well as at generating the effective and comparable RFs shown in experiments. This observation, independent of the specific choice of recurrent units modeled, is a general feature that emerged from the neural response to dynamic visual scenes.

The role of the recurrent connection

The CRNN models capture several important properties of the biological retinal circuits, while the CNN cannot. First, the convolutional filters in the first hidden layer of the CRNN are more consistent with the model subunits in terms of the shapes of the RFs, regardless of kernel size changes. Second, the CRNN better predicts the response of the animal retina to natural scene movies. Last, the CRNN provides an estimate of the location and shape of the RF of each RGC.

Inspired by experimental observation in neuroscience,[46] typical neural networks have a hierarchical architecture with several layers. Some of these layers include a block of convolutional filters, and consequently, each filter serves as a feature detector to extract an important property of the input images.[47] Thus, training with a large set of images allows the convolutional filters to play functional roles as neurons in the retina and the elements of other visual systems to encode the complex statistical properties of natural images. The filter shapes are sparse and localized, like the RF of the visual neurons. Therefore, it would be reasonable to use similar neural network approaches to investigate the central question of neuronal encoding in neuroscience.[8]

In visual coding, the ventral stream of visual processing in the brain starts in the retina and passes to the lateral geniculate nucleus via the optic nerve, and finally the layered visual cortex, to reach the inferior temporal gyrus. This visual pathway has been suggested as the ‘‘what pathway,’’ used to recognize and identify visual objects.[48,49] A deep neural network was used to model and predict with reasonable accuracy the activity of the neurons in the inferior temporal cortex in monkeys.[8,50–52] Therefore, the biological underpinnings of the ventral stream of visual processing in the brain can be related to the structure components used in deep neural networks. However, interpretation of this relationship is not straightforward, since the pathway from the retina to the inferior temporal cortex is complicated,[8] although in the retina the neuronal organization is relatively simple.[53]

Previously, a few studies have taken this approach by applying different kinds of CNNs to model earlier elements of the visual systems on the brain, such as the retina,[15,39] V1,[54–58] and V2.[59] Similar to the current study, most of these studies sought to demonstrate that a better performance in terms of neural response could be achieved by using either a feedforward CNN or an RNN, or both. The results presented in this study provide a promising direction in which to reinvestigate the functional role of feedforward and recurrent approaches for different types of visual scenes. The recurrent layer plays an important role in modeling neuronal nonlinearity, which is a unique feature of neural computation.[60] By incorporating recurrent connections, many models have shown advantages in recognizing static images.[22,25,61–64] An unrolled recurrent network is equivalent to a deeper or wider network that saves on neurons by repeating data transformation several times,[24,65,66] but it improves the flexibility trading of speed and accuracy in biological vision.[67] Our results, together with those from other recent studies, provide new insights into the underlying mechanisms of neuronal encoding for dynamic visual scenes, as well as the design of better models for analyzing dynamic visual scenes.

Another potential approach is to model recurrence with an additional layer of neurons to mimic the role of inhibitory amacrine cells of the retina. One simple way is to add inhibition using local normalization, such that each inhibition cell suppresses a local group of cells, similar to the retinal biologic mechanism. However, how to add grouping constraints to make lateral suppression work is still an open problem. As early as in the AlexNet[68] on the image classification task, the authors introduced local response normalization to make neurons in the same local region inhibit one another. However, the divisive normalization did not bring too much gain to the network performance. Hence, this idea is not widely used in deep neural networks so far. Given the rich neuroscience knowledge from the retina and other visual pathways, there may be other ways to add inhibitory neurons to artificial neural networks. It is worthy of more detailed investigation and future work.

Model parameter pruning using temporal filters

To evaluate whether the elements of the models play the roles of intermediate computational mechanisms similar to those used in biological retinal circuits, several subunit importance evaluation indices have been introduced. In one study,[17] the authors reduced the number of hidden neurons or stimulus attribution of the trained models based on an importance index and finally tested the effectiveness of CNN subnetwork models through their response-based performance in experimental protocols, including omitted stimulus response, motion reversal, etc. This method is used to quantify the importance of the model units according to their contribution to the final neuronal firing rate and exploit stimulus invariance to reduce computational dimensionality. However, this is infeasible when the natural dynamic stimulus is not spatially invariant, and the prediction result is a population response rather than a single neuron response; furthermore, it is difficult to measure the contribution of each subunit to the individual RGCs.

In other studies, the effectiveness of model subunits has been quantified by the spatial autocorrelation of the convolution kernels,[16,42] and was determined by the Moran index. However, this method can be verified only on the white-noise stimuli, since they are not spatiotemporally correlated. Hence, the spatial RF of the convolution kernel is relatively concentrated in a small area, representing only the center effect of the RF without the surrounding inhibition effect.[41] Such a simple RF allows the selected subunit to achieve desirable results. However, in dynamic natural scenes, each pixel has a comparatively high correlation with spatially adjacent areas, leading to a large spatial autocorrelation for the convolution kernel, with no possibility of reduction of the correlation as represented in the Moran index. Consequently, we took advantage of the biphasic response via the ON and OFF polarity in the temporal filter of the RF produced by temporal adaption[43,69] and evaluated it according to a relatively regular oscillatory wave with some peak sensitivity and period of adaption. Temporal adaption is ubiquitous not only for the neural computation of sensory input,[43] but also for controlling and adjusting the dynamic range of single cells and neural populations in investigations of general neural dynamics.[70,71] Without limitation of the properties of visual stimulus, e.g., spatial invariation, our importance index, an important feature for modeling of the retinal encoding, used to evaluate the temporal filter enables us to incorporate neuronal adaption in response to complex and dynamic natural visual scenes.[72,73] Thus, our filter pruning approach could help reduce the effective number of parameters in other deep-learning models while processing the dynamic visual scenes, expanding their performance beyond that for static natural images.

Application to other systems

Here we use the retina as a model system to explain the role of recurrence in the network modeling the relationship between neural response and dynamic visual scenes. It is well known that the retina is one of the best-understood examples in neuroscience for visual computing.[1] The methodologies for the retina generally work well for other visual pathways, from the lateral geniculate nucleus and primary visual cortex to the inferior temporal cortex, as well as neural coding in other parts of the brain.[8,38] Recent studies also emphasize the role of recurrence in visual computing.[20,63,64] Our work aligns with this line of showing the unique feature of recurrence in neural network models: the recurrent connection plays a role similar to maintaining the *memory* of lasting external stimulation and ongoing neural dynamics. In other words, the recurrent layer stores the previously input computation information in the hidden node state, and then a new prediction judgment can be made by combining the previously stored information when a new input is introduced. This functional role is generally shown in various brain areas.[7,8]

The topic of our work focused on dynamic visual scenes, e.g., continuous videos. Video analysis is of great interest to data science researchers, not only for neuroscience, but also for other domains of applied vision, including machine vision, neuromorphic computing, and brain-machine interface, where a large chunk of data in the format of videos is analyzed.[2] Analysis of static natural images is relatively easy,[18] while videos span multiple scales in space and time, which raises tremendous difficulties for analyzing the contexts themselves,[74] as well as for characterizing the underlying neural dynamics.[72] We show that movies with different levels of complexity show different behaviors in models. Such difference calls for a further investigation of the level of scene complexity and how it affects neural dynamics and the network modeling approach.

Visual neuroscience is an immensely popular topic in machine learning, such that numerous methodologies developed have a broad application to and inspiration for other topics.[7,46] From the perspective of deep learning, the introduction of recurrent connections affects the parameter adjustment of the model during backpropagation, subsequently affecting the learning results. When modeling the neural response of simulated data, the convolution kernels of the CRNN are more consistent with neural subunits. When the model is trained with videos, the spatial autocorrelation and temporal regularity of the spatiotemporal filters in the CRNN model are stronger, suggesting that the lateral connections routed by inhibitory cells or gap junctions play a functional role. In particular, it is very beneficial to use temporal regularity to reduce the model parameters to a subset while maintaining the model performance. It implies that the effect of the temporal domain in videos is more prominent than that of the spatial domain.[43,74] These implications based on a neuroscience-inspired approach could provide inspiration for algorithm designs of artificial intelligence.[46]

Limitations

In this work, the CRNN model mainly simulates the structure of the retina, with a three-layer feedforward network with some interneurons and gap junctions. Thus the proposed model was studied to simulate the encoding process of the retina with a relatively simple setting. However, in addition to the interneurons such as amacrine cells and gap junctions, there are other interactions between cells in the retinal circuit, for instance, feedback from RGCs to the inner retina.[75] The current architecture of our proposed models is based on the simple assumption of the retinal circuit. These retinal components that we have not simulated may play an indispensable role in the process of encoding the external environment by the retina. Future work is needed to include these feedback factors, which can improve our modeling approach beyond the retina to other higher visual pathways.

When learning to predict the response of the same 80 ganglion cells to the two natural scene videos, CRNN can train an effective model on each stimulus. However, it is still unable to transfer the models between videos, e.g., a model trained on movie 1 cannot predict well the response of RGCs to movie 2. Thus, future work is needed to introduce some strategies to make the model show good generalization performance on different visual stimuli. Furthermore, dealing with the higher complexity of movies may need additional mechanisms. One possible mechanism is attention or feedback, which has engendered significant efforts in modeling visual computing.

The added recurrent layer could have an attention mechanism whereby the recurrent unit will pay attention to different features of the input. Except for the recurrent-based attention network, recently the transformer,[76] based solely on attention mechanisms, has outperformed many convolutional recurrent networks on processing sequential tasks, e.g., natural language processing. Ideally, in the future, we hope to build models utilizing these deep-learning architectures for neural encoding and decoding of the visual pathway rather than the retina. Another direction is to use graph neural networks processing non-Euclidean data.[77] The current models of retinal coding usually receive frame-based input, which belongs to the distribution of the Euclidean domain. The traditional convolutional layer can extract features well on regular Euclidean data. In the future, we will explore how to build a model to predict the retinal response against stimuli that belong to different data distributions in the non-Euclidean domain. These strategies could reinforce our consideration of the role of recurrence in visual computing, either data raised from neuroscience or applications of neuromorphic computing, brain-machine interface, and video analysis.

EXPERIMENTAL PROCEDURES

Resource availability

Lead contact

The lead contact for this study is Jian K. Liu: j.liu9@leeds.ac.uk.

Materials availability

This study did not generate new unique materials.

Data and code availability

The data are available at https://doi.org/10.5061/dryad.4ch10. The code is available at https://github.com/Zyj061/retina-crnn_model.

Table 1. Models' parameter settings

Name	Description	Size/value
Inputs	spatiotemporal stimulus	90 × 90 × 20
conv1 num	number of kernels in the first convolutional layer	128
ℓ_1^1	weight regularization in the first convolutional layer	5×10^{-4}
conv2 num	number of kernels in the second convolutional layer	64
ℓ_2	weight regularization	10^{-3}
ℓ_1	activity regularization on the dense layer	10^{-3}
Outputs	dynamic responses of the population RGCs	80

The RGC encoding models

To untangle the underpinnings of the retinal system and gain a clearer understanding of the stimulus/response relationship for dynamic scenes, we propose two deep-learning models to describe RGC encoding: one based on a CNN and another based on a CRNN.

CNN

The structure of the proposed CNN encoding model is the same as that of the models used in the references.[14,16] In the CNN model, we establish two convolutional layers to extract the spatiotemporal information of the input stimulus. The outputs of these convolutional layers are obtained as follows:

$$\boldsymbol{O}_l = g(\varphi(\boldsymbol{W}_l * \boldsymbol{O}_{l-1} + \boldsymbol{b}_l)), \qquad \text{(Equation 1)}$$

where $\boldsymbol{W}_l$ and $\boldsymbol{b}_l$ are the convolutional weights and biases of layer l, respectively. $*$ denotes the convolution operation, and $g(\bullet)$ is the activation function, which is set as the ReLU function $g(x) = max(0, x)$ in this work. $\varphi(\bullet)$ denotes all the operations that follow the convolution, e.g., batch normalization and the addition of Gaussian noise. After the second convolutional layer, we flatten the output $\boldsymbol{O}_2$ into a one-dimensional vector $\tilde{\boldsymbol{O}}_2$, and pass it through a dense layer with n output neurons corresponding to the population RGCs. The outputs of the dense layer denote the firing rates of the RGCs, which are obtained as follows:

$$\hat{\boldsymbol{y}} = \varphi(\boldsymbol{W}_d \cdot \tilde{\boldsymbol{O}}_2 + \boldsymbol{b}_d), \qquad \text{(Equation 2)}$$

where $\varphi(x)$ is the parametric softplus function $\varphi(x) = \alpha \cdot \log(1 + \exp(\beta x))$, and $\boldsymbol{W}_d$ and $\boldsymbol{b}_d$ are the connected weights and biases, respectively, of the dense layer. The α and β are trainable parameters. Taking the actual firing rate $\boldsymbol{y}$ of the RGCs as the fitting target, the models are optimized to jointly minimize the Poisson loss function and regularization as follows:

$$L(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{1}{N}(\hat{\boldsymbol{y}} - \boldsymbol{y}\log\hat{\boldsymbol{y}}) + ||\boldsymbol{W}_d||_2 + ||\hat{\boldsymbol{y}}||_1, \qquad \text{(Equation 3)}$$

where N is the batch size of the samples used at each iteration. $||\cdots||_1$ and $||\cdots||_2$ represent ℓ_1 and ℓ_2 norm regularization, respectively. To avoid overfitting and ensure that the neurons are sparsely firing, we apply ℓ_2 norm regularization to the weights and ℓ_1 norm regularization to the neuron activation. In each layer of the network, we add the ℓ_2 norm to regularize the weights of the layer. The output of each layer is normalized using batch normalization prior to the nonlinear activation function. In the last fully connected layer, the softplus activation function is used with trainable parameters α and β. The weight of the first convolutional layer is regularized by the ℓ_1 norm to let the RFs of the convolutional kernels have more compact shapes. The nonnegative loss and Adam optimization strategies are used for multivariate regression training.

CRNN

In the CRNN model, we add an additional recurrent layer between the second convolutional layer and the final dense layer of the previous CNN model. We have examined the effects of different kinds of units in the recurrent layer on predicting the retinal response to natural movie stimuli, including vanilla RNN, GRU, and LSTM. We use 32 recurrent units as the components of the special recurrent layer (the number of recurrent units can be adjusted when the number of RGCs increases), which has been shown to be powerful and efficient in modeling sequence dependencies. We take the output of the second convolutional layer $\boldsymbol{O}_2 = \{\boldsymbol{O}_2^1, \ldots, \boldsymbol{O}_2^t\}$ as the input sequence to the recurrent layer, where each feature map $\boldsymbol{O}_2^t$ is the input at each time step. In the following, we introduce the details of vanilla RNN, GRU, and LSTM.

Vanilla RNN. In the vanilla RNN unit, output state vector $\boldsymbol{h}_t$ is obtained by passing through the multiplication of the output of the second convolutional layer $\boldsymbol{O}_2^t$ and the previous state $\boldsymbol{h}_{t-1}$ to the **Tanh** activation function:

$$\boldsymbol{h}_t = \tanh(\boldsymbol{W} \cdot \boldsymbol{O}_2^t + \boldsymbol{U} \cdot \boldsymbol{h}_{t-1} + \boldsymbol{b}), \qquad \text{(Equation 4)}$$

where $\boldsymbol{W}$, $\boldsymbol{U}$, and $\boldsymbol{b}$ are the feedforward weight matrix, recurrent weight matrix, and bias vector, respectively, which need to be learned during training. $\boldsymbol{W}$, $\boldsymbol{U}$, and $\boldsymbol{b}$ in the following formulas also have the same meaning.

GRU. Compared with the vanilla RNN, update gate $\boldsymbol{z}_t$ and reset gate $\boldsymbol{r}_t$ are introduced into the GRU unit to avoid gradient vanishing/exploding with long-term stimuli. The hidden state $\boldsymbol{h}_t$ of the GRU unit is obtained by:

$$\boldsymbol{z}_t = \sigma(\boldsymbol{W}_z \cdot \boldsymbol{O}_2^t + \boldsymbol{U}_z \cdot \boldsymbol{h}_{t-1} + \boldsymbol{b}_z), \qquad \text{(Equation 5)}$$

$$\boldsymbol{r}_t = \sigma(\boldsymbol{W}_r \cdot \boldsymbol{O}_2^t + \boldsymbol{U}_r \cdot \boldsymbol{h}_{t-1} + \boldsymbol{b}_r),$$

$$\hat{\boldsymbol{h}}_t = \tanh(\boldsymbol{W}_h \cdot \boldsymbol{O}_2^t + \boldsymbol{U}_h(\boldsymbol{r}_t \circ \boldsymbol{h}_{t-1}) + \boldsymbol{b}_{\hat{h}}),$$

$$\boldsymbol{h}_t = (1 - \boldsymbol{z}_t) \circ \boldsymbol{h}_{t-1} + \hat{\boldsymbol{h}}_t \circ \boldsymbol{z}_t,$$

where $\cdot$ refers to dot production, and $\circ$ denotes element-wise multiplication.

LSTM. Each LSTM unit consists of an input gate $\boldsymbol{i}_t$, a forget gate $\boldsymbol{f}_t$, and an output gate $\boldsymbol{o}_t$, while one hidden unit means maintaining one time-step memory at t. The states of these gates and cells are as follows:

$$\boldsymbol{f}_t = \sigma(\boldsymbol{W}_f \cdot \boldsymbol{O}_2^t + \boldsymbol{U}_f \cdot \boldsymbol{h}_{t-1} + \boldsymbol{b}_f), \qquad \text{(Equation 6)}$$

$$\boldsymbol{i}_t = \sigma(\boldsymbol{W}_i \cdot \boldsymbol{O}_2^t + \boldsymbol{U}_i \cdot \boldsymbol{h}_{t-1} + \boldsymbol{b}_i),$$

$$\boldsymbol{o}_t = \sigma(\boldsymbol{W}_o \cdot \boldsymbol{O}_2^t + \boldsymbol{U}_o \cdot \boldsymbol{h}_{t-1} + \boldsymbol{b}_o),$$

$$\hat{\boldsymbol{c}}_t = \tanh(\boldsymbol{W}_c \cdot \boldsymbol{O}_2^t + \boldsymbol{U}_c \cdot \boldsymbol{h}_{t-1} + \boldsymbol{b}_c),$$

$$\boldsymbol{c}_t = \boldsymbol{f}_t \circ \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \circ \hat{\boldsymbol{c}}_t,$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t \circ \tanh(\boldsymbol{c}_t),$$

where $\hat{\boldsymbol{c}}_t$, $\boldsymbol{c}_t$, and $\boldsymbol{h}_t$ represent the cell input activation vector, cell state vector, and output vector, respectively, and $\circ$ denotes element-wise multiplication.

Each recurrent unit generates a sequence output $\mathbf{h} = \{\boldsymbol{h}_1, \ldots, \boldsymbol{h}_t\}$ $t = 64$, and by stacking $n_l = 32$ recurrent units in one layer, the output of the entire recurrent layer is $\mathcal{H} = \mathbf{h}^1, \ldots, \mathbf{h}^{n_l}$. Similar to the outputs of the convolutional layer, we also flatten the recurrent layer output $\mathcal{H}$ into a one-dimensional vector $\tilde{\mathcal{H}}$, and pass it through a dense layer with n output neurons. The final output of the CRNN model $\hat{\boldsymbol{y}}$ is the following:

$$\hat{\boldsymbol{y}} = \varphi(\boldsymbol{W}_d \cdot \widetilde{\mathcal{H}} + \boldsymbol{b}_d), \qquad \text{(Equation 7)}$$

where $\varphi(\bullet)$ is identical to the parametric softplus function used in Equation 2. Similar to the training process of the CNN model, the CRNN model is optimized by minimizing the Poisson loss, with ℓ_1 regularization on neuron activity $\hat{\boldsymbol{y}}$ and ℓ_2 regularization on the connected weight $\boldsymbol{W}_d$.

Model implementation

All of the models were implemented with Keras using TensorFlow as the back end and trained on NVIDIA K80 GPUs. The training epoch was set to 1,000, but the training would terminate early if the loss converged. To model biophysical

RGC responses using an entire frame as input, we used a filter size of 25 × 25 in the first convolutional layer and a filter size of 11 × 11 in the second layer. For the CRNN models referred to in the results, which were trained on natural movies, we used LSTM units in the recurrent layer with ℓ_2 norm regularization in the kernels. In addition to modeling larger amounts of RGC data with CRNN models constructed with 32, 64, 128, and 256 LSTM units, we used 32 units in all the other CRNN models. Except for the recurrent layer, all the other units and hyperparameters were the same for the CNN and the CRNN. The settings of the CNNs and CRNNs used for learning the relationships between the dynamic responses of the population RGCs and the natural movies are shown in Table 1.

Model pruning

To examine whether the deep-learning-based mode just learns the relationship without explainable hidden units, we developed a novel pruning strategy to evaluate the importance of the parameters of the convolution kernels, and to evaluate whether the models act as intermediate computational mechanisms similar to those used in biological retinal circuits. According to previous analyses of temporal filters of neural circuits, the RF of an effective temporal filter is a relatively regular oscillatory wave with a certain peak sensitivity and an adaptation period, such as that for ON or OFF bipolar cells.[69] Therefore, we propose a novel subunit importance index to quantify the wave regularity of a temporal filter. The calculation formula is as follows:

$$I_{temporal} = \frac{1}{T}\sum_{i=1}^{T}\left|\left||w_i|-\max|\mathbf{w}|\right|\right| - \frac{\varepsilon}{\mathbf{w}-\overline{\mathbf{w}}_2+\varepsilon}, \qquad \text{(Equation 8)}$$

where **w** is the weight of the temporal filter obtained by SVD of the first convolutional kernel, T is the length of the temporal filter, and w_i is the element of the temporal filter at position i. The first term of the formula determines whether there are regular wave peaks in the temporal filter. To eliminate the influence of the corresponding ON and OFF subunits, the first term is calculated using the absolute value of the given parameter of the kernel. In the second term, the Euclidean distance between each temporal filter parameter and its average value is used as the denominator to improve the diversity of the temporal filter. ε is a small value, which is set to 5×10^{-4}.

RGC experimental data

To verify the performance and effectiveness of our models with biological data, we used public datasets recorded from the ganglion cells of isolated salamander retinas using multi-electrode arrays with natural movies as the input stimuli.[40] Briefly, each frame of the movies covered an area of 2,700 × 2,700 μm^2 on the retinas with a spatial resolution of 360 × 360 pixels. The multi-electrode arrays were used to record the responses of 80 RGCs to 31 and 33 trials of the presentation of movie 1 (simple scenes of swimming salamander) and movie 2 (complex scenes of animals), respectively (described earlier under "CRNN enhances the encoding of retinal responses to dynamic natural stimuli"). For model training, the target output was created by averaging the response from each cell over all trials and binning with a bin width of 33 ms. To have the model learn the spatial and temporal filters, at each of the time bins created for the target model output, the corresponding frame of the movie downsampled to 90 × 90 pixels, along with the frames of the preceding 20 time bins, was fed as the input.

Complexity of the natural scenes

Natural movies have different scene contexts, which can be described in the pixel space on spatial and temporal scales. To characterize the spatiotemporal complexities of the scenes, first, each frame of the movie is sliced into patches of equal size, and the similarity between each patch and its neighboring patches is computed. For spatial complexity, we first calculate the SSIM between a patch and the eight neighboring patches in each movie frame. Next, we average the SSIM values across the neighboring patches. By averaging this value across all frames, we take the spatial correlation of the patch; the spatial complexity of the patch is 1 minus this value, i.e., a higher correlation means that the patch has a lower complexity. For example, for patch i (as shown in Figure S2), the spatial complexity *SC* is:

$$SC_i = 1 - \frac{1}{T}\sum_{t=1}^{T}\frac{1}{n}\sum_{j\in neig(i)}^{n} SSIM\left(\boldsymbol{p}_i^t,\boldsymbol{p}_j^t\right), \qquad \text{(Equation 9)}$$

where T is the number of frames in the movie, n is the patch number of j, which specifies its location as one of the eight neighbors of patch i, and ***p*** denotes the slice patch.

The calculation for the temporal complexity is similar; however, instead of comparing the patches in the same frame, the SSIM is calculated between a patch in frame t and its eight neighboring patches in the corresponding positions in frame t + 1, as well as between the patch at time t and the same patch at time t + 1. The temporal complexity of the patch is then obtained by performing similar steps as described for the spatial complexity. The formula for temporal complexity *TC* of patch i is as follows:

$$TC_i = 1 - \frac{1}{T-1}\sum_{t=1}^{T-1}\frac{1}{n+1}\left(\sum_{j\in neig(i)}^{n} SSIM\left(\boldsymbol{p}_i^t,\boldsymbol{p}_j^{t+1}\right) + SSIM\left(\boldsymbol{p}_i^t,\boldsymbol{p}_i^{t+1}\right)\right). \qquad \text{(Equation 10)}$$

Using a patch size of 18 × 18 pixels, space and time complexities of both movies are shown in Figure 3D. Such complexities can affect the performance of the encoding models. To relate the performance of each model RGC to the complexity of an individual movie patch, we overlap the RF of each RGC with each patch, collocate those RGCs in that patch, and average all the CCs of the individual RGCs as the performance of the CNN and CRNN models for that particular image patch. Finally, we obtain a relationship between the complexity of each movie and the performance of each model, as shown in Figure 3E.

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.patter.2021.100350.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (62088102, 61961130392); the International Youth Talent Fund of Zhejiang Lab (ZJ2020JS017), Zhejiang Lab, China (2019KC0AB03 and 2019KC0AD02); and the Royal Society Newton Advanced Fellowship, UK (NAF-R1-191082).

AUTHOR CONTRIBUTIONS

Conceptualization, Y.Z., Z.Y., and J.K.L.; data curation, S.J. and J.K.L.; formal analysis, Y.Z., S.J., Z.Y., and J.K.L.; investigation, Y.Z.; methodology, Y.Z. and J.K.L.; writing, Y.Z., Z.Y., and J.K.L.; funding acquisition, Y.Z., Z.Y., J.K.L., and T.H.; supervision, Z.Y., J.K.L., and T.H.

DECLARATION OF INTERESTS

The authors declare no competing interests.

Received: June 3, 2021
Revised: June 22, 2021
Accepted: August 23, 2021
Published: September 17, 2021

REFERENCES

1. Gollisch, T., and Meister, M. (2010). Eye smarter than scientists believed: neural computations in circuits of the retina. Neuron *65*, 150–164.
2. Shah, N.P., and Chichilnisky, E.J. (2020). Computational challenges and opportunities for a bi-directional artificial retina. J. Neural Eng. *17*, 055002.
3. Zhang, Y., Jia, S., Zheng, Y., Yu, Z., Tian, Y., Ma, S., Huang, T., and Liu, J.K. (2020). Reconstruction of natural visual scenes from neural spikes with deep neural networks. Neural Networks *125*, 19–30.
4. Kelly, D. (1962). Information capacity of a single retinal channel. IRE Trans. Inf. Theor. *8*, 221–226.

5. Zhaoping, L., and Li, Z. (2014). Understanding Vision: Theory, Models, and Data (Oxford University Press).

6. Richards, B.A., Lillicrap, T.P., Beaudoin, P., Bengio, Y., Bogacz, R., Christensen, A., Clopath, C., Costa, R.P., de Berker, A., Ganguli, S., et al. (2019). A deep learning framework for neuroscience. Nat. Neurosci. *22*, 1761–1770.

7. Kriegeskorte, N. (2015). Deep neural networks: a new framework for modeling biological vision and brain information processing. Annu. Rev. Vis. Sci. *1*, 417–446.

8. Yamins, D.L.K., and Dicarlo, J.J. (2016). Using goal-driven deep learning models to understand sensory cortex. Nat. Neurosci. *19*, 356–365.

9. Cadieu, C.F., Hong, H., Yamins, D.L., Pinto, N., Ardila, D., Solomon, E.A., Majaj, N.J., and DiCarlo, J.J. (2014). Deep neural networks rival the representation of primate it cortex for core visual object recognition. PLoS Comput. Biol. *10*, e1003963.

10. Khaligh-Razavi, S.M., and Kriegeskorte, N. (2014). Deep supervised, but not unsupervised, models may explain it cortical representation. PLoS Comput. Biol. *10*, e1003915.

11. Yamins, D.L., Hong, H., Cadieu, C.F., Solomon, E.A., Seibert, D., and DiCarlo, J.J. (2014). Performance-optimized hierarchical models predict neural responses in higher visual cortex. Proc. Natl. Acad. Sci. U S A *111*, 8619–8624.

12. Güçlü, U., and van Gerven, M.A. (2015). Deep neural networks reveal a gradient in the complexity of neural representations across the ventral stream. J. Neurosci. *35*, 10005–10014.

13. Rajalingham, R., Schmidt, K., and DiCarlo, J.J. (2015). Comparison of object recognition behavior in human and monkey. J. Neurosci. *35*, 12127–12136.

14. McIntosh, L., Maheswaranathan, N., Nayebi, A., Ganguli, S., and Baccus, S. (2016). Deep learning models of the retinal response to natural scenes. Adv. Neural Inf. Process. Syst. 1369–1377.

15. Maheswaranathan, N., McIntosh, L.T., Kastner, D.B., Melander, J., Brezovec, L., Nayebi, A., Wang, J., Ganguli, S., and Baccus, S.A. (2018). Deep learning models reveal internal structure and diverse computations in the retina under natural scenes. bioRxiv, 340943. https://doi.org/10.1101/340943.

16. Yan, Q., Zheng, Y., Jia, S., Zhang, Y., Yu, Z., Chen, F., Tian, Y., Huang, T., and Liu, J.K. (2020). Revealing fine structures of the retinal receptive field by deep-learning networks. IEEE Trans. Cybernetics, 1–12. https://doi.org/10.1109/TCYB.2020.2972983.

17. Tanaka, H., Nayebi, A., Maheswaranathan, N., McIntosh, L., Baccus, S., and Ganguli, S. (2019). From deep learning to mechanistic understanding in neuroscience: the structure of retinal prediction. Adv. Neural Inf. Process. Syst. 8535–8545.

18. Simoncelli, E.P., and Olshausen, B.A. (2001). Natural image statistics and neural representation. Annu. Rev. Neurosci. *24*, 1193.

19. Kietzmann, T.C., Ehinger, B.V., Porada, D., Engel, A.K., and König, P. (2016). Extensive training leads to temporal and spatial shifts of cortical activity underlying visual category selectivity. NeuroImage *134*, 22–34.

20. Kar, K., Kubilius, J., Schmidt, K., Issa, E.B., and DiCarlo, J.J. (2019). Evidence that recurrent circuits are critical to the ventral stream's execution of core object recognition behavior. Nat. Neurosci. *22*, 974–983.

21. Kietzmann, T.C., Spoerer, C.J., Sörensen, L.K., Cichy, R.M., Hauk, O., and Kriegeskorte, N. (2019). Recurrence is required to capture the representational dynamics of the human visual system. Proc. Natl. Acad. Sci. U S A *116*, 21854–21863.

22. Spoerer, C.J., McClure, P., and Kriegeskorte, N. (2017). Recurrent convolutional neural networks: a better model of biological object recognition. Front. Psychol. *8*, 1551.

23. Michaelis, C., Bethge, M., and Ecker, A. (2018). One-shot segmentation in clutter. In Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, PMLR 80.

24. Rajaei, K., Mohsenzadeh, Y., Ebrahimpour, R., and Khaligh-Razavi, S.M. (2019). Beyond core object recognition: recurrent processes account for object recognition under occlusion. PLoS Comput. Biol. *15*, e1007001.

25. Linsley, D., Kim, J., Veerabadran, V., and Serre, T. (2018). Learning long-range spatial dependencies with horizontal gated-recurrent units. In NIPS'18: Proceedings of the 32nd International Conference on Neural Information Processing Systems, pp. 152–164.

26. Rao, R.P., and Ballard, D.H. (1999). Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. Nat. Neurosci. *2*, 79–87.

27. Lotter, W., Kreiman, G., and Cox, D. (2016). Deep predictive coding networks for video prediction and unsupervised learning. arXiv, 160508104 [cs.LG].

28. Issa, E.B., Cadieu, C.F., and DiCarlo, J.J. (2018). Neural dynamics at successive stages of the ventral visual stream are consistent with hierarchical error signals. Elife *7*, e42870.

29. Souihel, S., and Cessac, B. (2021). On the potential role of lateral connectivity in retinal anticipation. J. Math. Neurosci. *11*, 1–60.

30. Bloomfield, S.A., and Völgyi, B. (2009). The diverse functional roles and regulation of neuronal gap junctions in the retina. Nat. Rev. Neurosci. *10*, 495–506.

31. Grimes, W.N., Songco-Aguas, A., and Rieke, F. (2018). Parallel processing of rod and cone signals: retinal function and human perception. Annu. Rev. Vis. Sci. *4*, 123–141.

32. O'Brien, J., and Bloomfield, S.A. (2018). Plasticity of retinal gap junctions: roles in synaptic physiology and disease. Annu. Rev. Vis. Sci. *4*, 79–100.

33. Rivlin-Etzion, M., Grimes, W.N., and Rieke, F. (2018). Flexible neural hardware supports dynamic computations in retina. Trends Neurosci. *41*, 224–237.

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

35. Trenholm, S., Schwab, D.J., Balasubramanian, V., and Awatramani, G.B. (2013). Lag normalization in an electrically coupled neural network. Nat. Neurosci. *16*, 154–156.

36. Werblin, F.S. (2011). The retinal hypercircuit: a repeating synaptic interactive motif underlying visual function. J. Physiol. *589*, 3691–3702.

37. Yu, Z., Liu, J.K., Jia, S., Zhang, Y., Zheng, Y., Tian, Y., and Huang, T. (2020). Toward the next generation of retinal neuroprosthesis: visual computation with spikes. Engineering *6*, 449–461.

38. 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*, 995–999.

39. Batty, E., Merel, J., Brackbill, N., Heitman, A., Sher, A., Litke, A., Chichilnisky, E.J., Paninski, L. (2017). Multilayer recurrent network models of primate retinal ganglion cell responses. International Conference on Learning Representations.

40. Onken, A., Liu, J.K., Karunasekara, P.C.R., Delis, I., Gollisch, T., and Panzeri, S. (2016). Using matrix and tensor factorizations for the single-trial analysis of population spike trains. PLoS Comput. Biol. *12*, e1005189.

41. Chichilnisky, E.J. (2001). A simple white noise analysis of neuronal light responses. Netw. Comput. Neural Syst. *12*, 199–213.

42. Liu, J.K., Schreyer, H.M., Onken, A., Rozenblit, F., Khani, M.H., Krishnamoorthy, V., Panzeri, S., and Gollisch, T. (2017). Inference of neuronal functional circuitry with spike-triggered non-negative matrix factorization. Nat. Commun. *8*, 149.

43. Liu, J.K., and Gollisch, T. (2015). Spike-triggered covariance analysis reveals phenomenological diversity of contrast adaptation in the retina. PLoS Comput. Biol. *11*, e1004425.

44. Cho, K., Van Merriënboer, B., Bahdanau, D., and Bengio, Y. (2014). On the properties of neural machine translation: encoder-decoder approaches. arXiv, 1409.1259 [cs.CL].

45. Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. Neural Comput. *9*, 1735–1780.

46. Demis, H., Dharshan, K., Christopher, S., and Matthew, B. (2017). Neuroscience-inspired artificial intelligence. Neuron *95*, 245–258.

47. Simonyan, K., Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. International Conference on Learning Representations.

48. Ingle, D.J., Goodale, M.A., and Mansfield, R.J. (1982). Analysis of Visual Behavior (MIT Press Cambridge).

49. Mishkin, M., Ungerleider, L.G., and Macko, K.A. (1983). Object vision and spatial vision: two cortical pathways. Trends Neurosci. *6*, 414–417.

50. Yamins, D., Hong, H., Cadieu, C., and Dicarlo, J.J. (2013). Hierarchical modular optimization of convolutional networks achieves representations similar to macaque it and human ventral stream. Adv. Neural Inf. Process. Syst. 3093–3101.

51. Yamins, D.L.K., Hong, H., Cadieu, C.F., Solomon, E.A., Seibert, D., and DiCarlo, J.J. (2014). Performance-optimized hierarchical models predict neural responses in higher visual cortex. Proc. Natl. Acad. Sci. U S A *111*, 8619–8624.

52. Khaligh-Razavi, S.M., and Kriegeskorte, N. (2014). Deep supervised, but not unsupervised, models may explain IT cortical representation. PLoS Comput. Biol. *10*, e1003915.

53. Lindsey, J., Ocko, S.A., Ganguli, S., and Deny, S. (2019). A unified theory of early visual representations from retina to cortex through anatomically constrained deep CNNs. biorXiv. 190100945. https://doi.org/10.1101/511535.

54. Vintch, B., Movshon, J.A., and Simoncelli, E.P. (2015). A convolutional subunit model for neuronal responses in macaque V1. J. Neurosci. *35*, 14829–14841.

55. Antolík, J., Hofer, S.B., Bednar, J.A., and Mrsic-Flogel, T.D. (2016). Model constrained by visual hierarchy improves prediction of neural responses to natural scenes. PLoS Comput. Biol. *12*, e1004927.

56. Kindel, W.F., Christensen, E.D., and Zylberberg, J. (2017). Using deep learning to reveal the neural code for images in primary visual cortex. arXiv, 1706.06208 [q-bio.NC].

57. Cadena, S.A., Denfield, G.H., Walker, E.Y., Gatys, L.A., Tolias, A.S., Bethge, M., and Ecker, A.S. (2019). Deep convolutional models improve predictions of macaque V1 responses to natural images. PLoS Comput. Biol. *15*, e1006897.

58. Klindt, D., Ecker, A.S., Euler, T., and Bethge, M. (2017). Neural system identification for large populations separating "what"and "where ". Adv. Neural Inf. Process. Syst. 3509–3519.

59. Rowekamp, R.J., and Sharpee, T.O. (2017). Cross-orientation suppression in visual area V2. Nat. Commun. *8*, 1–9.

60. Jia, S., Yu, Z., Onken, A., Tian, Y., Huang, T., and Liu, J.K. (2021). Neural system identification with spike-triggered non-negative matrix factorization. IEEE Trans. Cybernetics, 1–12.

61. Liang, M., Hu, X. (2015). Recurrent convolutional neural network for object recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 3367–3375.

62. Liao, Q., and Poggio, T. (2016). Bridging the gaps between residual learning, recurrent neural networks and visual cortex. arXiv, 1604.03640 [cs.LG].

63. Nayebi, A., Bear, D., Kubilius, J., Kar, K., Ganguli, S., Sussillo, D., DiCarlo, J.J., and Yamins, D.L. (2018). Task-driven convolutional recurrent models of the visual system. arXiv, 1807.00053 [q-bio.NC].

64. Nayebi, A., Sagastuy-Brena, J., Bear, D.M., Kar, K., Kubilius, J., Ganguli, S., Sussillo, D., DiCarlo, J.J., and Yamins, D.L. (2021). Goal-driven recurrent neural network models of the ventral visual stream. bioRxiv. https://doi.org/10.1101/2021.02.17.431717.

65. Zamir, A.R., Wu, T.L., Sun, L., Shen, W.B., Shi, B.E., Malik, J., Savarese, S. (2017). Feedback networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 1308–1317.

66. Leroux, S., Molchanov, P., Simoens, P., Dhoedt, B., Breuel, T., and Kautz, J. (2018). Iamnn: iterative and adaptive mobile neural network for efficient image classification. arXiv, 1804.10123 [cs.CV].

67. Spoerer, C.J., Kietzmann, T.C., Mehrer, J., Charest, I., and Kriegeskorte, N. (2020). Recurrent neural networks can explain flexible trading of speed and accuracy in biological vision. PLoS Comput. Biol. *16*, e1008215.

68. Krizhevsky, A., Sutskever, I., and Hinton, G.E. (2012). ImageNet classification with deep convolutional neural networks. Adv. Neural Inf. Process. Syst. *1*, 1097–1105.

69. Weiss, Y. (2001). Comparing the mean field method and belief propagation for approximate inference in MRFs. In Advanced Mean Field Methods: Theory and Practice, M. Opper and D. Saad, eds. (MIT Press), pp. 229–240.

70. An, L., Tang, Y., Wang, D., Jia, S., Pei, Q., Wang, Q., Yu, Z., and Liu, J.K. (2020). Intrinsic and synaptic properties shaping diverse behaviors of neural dynamics. Front. Comput. Neurosci. *14*. https://doi.org/10.3389/fncom.2020.00026.

71. Tang, Y., An, L., Yuan, Y., Pei, Q., Wang, Q., and Liu, J.K. (2021). Modulation of the dynamics of cerebellar purkinje cells through the interaction of excitatory and inhibitory feedforward pathways. PLoS Comput. Biol. *17*, e1008670.

72. Heitman, A., Brackbill, N., Greschner, M., Sher, A., Litke, A.M., and Chichilnisky, E. (2016). Testing pseudo-linear models of responses to natural scenes in primate retina. bioRxiv, 045336.

73. Botella-Soler, V., Deny, S., Martius, G., Marre, O., and Tkačik, G. (2018). Nonlinear decoding of a complex movie from the mammalian retina. PLoS Comput. Biol. *14*, e1006057.

74. Lotter, W., Kreiman, G., and Cox, D. (2017). Deep predictive coding networks for video prediction and unsupervised learning. arXiv, 1605.08104v5 [cs.LG].

75. Vlasiuk, A., and Asari, H. (2021). Feedback from retinal ganglion cells to the inner retina. PLoS One *16*, e0254611.

76. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I. (2017). Attention is all you need. 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

77. Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., and Sun, M. (2020). Graph neural networks: a review of methods and applications. AI Open *1*, 57–81.