Task-Driven Convolutional Recurrent Models of the Visual System

Aran Nayebi[1,*], Daniel Bear[2,*], Jonas Kubilius[5,7,*], Kohitij Kar[5], Surya Ganguli[4], David Sussillo[8], James J. DiCarlo[5,6], and Daniel L. K. Yamins[2,3]

[1]Neurosciences PhD Program, Stanford University, Stanford, CA 94305
[2]Department of Psychology, Stanford University, Stanford, CA 94305
[3]Department of Computer Science, Stanford University, Stanford, CA 94305
[4]Department of Applied Physics, Stanford University, Stanford, CA 94305
[5]McGovern Institute for Brain Research, MIT, Cambridge, MA 02139
[6]Department of Brain and Cognitive Sciences, MIT, Cambridge, MA 02139
[7]Brain and Cognition, KU Leuven, Leuven, Belgium
[8]Google Brain, Google, Inc., Mountain View, CA 94043
[*]Equal contribution

Abstract

Feed-forward convolutional neural networks (CNNs) are currently state-of-the-art for object classification tasks such as ImageNet. Further, they are quantitatively accurate models of temporally-averaged responses of neurons in the primate brain's visual system. However, biological visual systems have two ubiquitous architectural features not shared with typical CNNs: local recurrence within cortical areas, and long-range feedback from downstream areas to upstream areas. Here we explored the role of recurrence in improving classification performance. We found that standard forms of recurrence (vanilla RNNs and LSTMs) do not perform well within deep CNNs on the ImageNet task. In contrast, custom cells that incorporated two structural features, bypassing and gating, were able to boost task accuracy substantially. We extended these design principles in an automated search over thousands of model architectures, which identified novel local recurrent cells and long-range feedback connections useful for object recognition. Moreover, these task-optimized ConvRNNs explained the dynamics of neural activity in the primate visual system better than feedforward networks, suggesting a role for the brain's recurrent connections in performing difficult visual behaviors.

1 Introduction

The sensory systems of the brain must detect meaningful patterns in noisy and complex sense data [James, 1890]. Visual surroundings can reveal objects of positive or negative value, including types of food, signs of danger, and memorable social partners. These stimuli vary widely in position, pose, contrast, background, and foreground from one occasion to the next. As a result, object identity is not easily discerned from low-level image properties [Pinto et al., 2008]. In primates, the visual system *encodes* relevant high-level properties so that they may readily guide behavior [Majaj et al., 2015]. This process can be modeled as a transformation from the original pixel values of an image into an internal representation [DiCarlo et al., 2012]. A good encoding algorithm is one that exposes task-relevant features to simple decoding procedures, such as linear classifiers [Hung et al., 2005, Majaj et al., 2015].

Preprint. Work in progress.

Recent work has shown that task-optimized deep convolutional neural networks (CNNs) are quantitatively accurate models of the primate brain's visual encoding [Yamins et al., 2014, Khaligh-Razavi and Kriegeskorte, 2014, Güçlü and van Gerven, 2015]. CNNs trained to recognize ImageNet objects currently explain the temporally-averaged responses of neurons in the visual system better than any other model class. Model units from lower, middle, and upper convolutional layers, respectively, provide the best-known linear predictions of neural evoked responses occurring in the early (area V1 [Khaligh-Razavi and Kriegeskorte, 2014, Cadena et al., 2017]), intermediate (area V4 [Yamins et al., 2014]), and higher visual cortical areas (inferotemporal cortex, IT, [Khaligh-Razavi and Kriegeskorte, 2014, Yamins et al., 2014]).

However, the primate visual system has additional anatomical structure that is not modeled with strictly feedforward CNNs. This includes dense local recurrent connections within each cortical region and long-range connections between different regions, such as feedback from a higher part of the visual hierarchy to a lower part [Gilbert and Wu, 2013]. The functional roles of recurrence in the primate brain's visual system are not well understood. Several conjectures are that recurrence "fills in" missing data, [Spoerer et al., 2017, Michaelis et al., 2018, Rajaei et al., 2018, Linsley et al., 2018] such as object parts occluded by other objects; that it "sharpens" representations by top-down attentional feature refinement, allowing for easier decoding of certain stimulus properties or performance of certain tasks [Gilbert and Wu, 2013, Lindsay, 2015, McIntosh et al., 2017, Li et al., 2018]; that it allows the brain to "predict" future stimuli (such as the frames of a movie) [Rao and Ballard, 1999, Lotter et al., 2017, Issa et al., 2018]; or that recurrence "extends" a feedforward computation, reflecting the fact that an unrolled recurrent network is equivalent to a deeper feedforward network that conserves on neurons (and learnable parameters) by repeating transformations several times [Khaligh-Razavi and Kriegeskorte, 2014, Liao and Poggio, 2016, Zamir et al., 2017, Leroux et al., 2018].

Because existing neural data do not rule out any of these possibilities, computational models might help to assess them. We therefore attempted to extend the method of goal-driven modeling [Yamins and DiCarlo, 2016, Mante et al., 2013, Shi et al., 2018] from feedforward CNNs and temporal averages of neural signal to convolutional recurrent neural networks (ConvRNNs) and neural dynamics. In particular, we hypothesized that adding recurrence and feedback to CNNs would help these models perform some ethologically-relevant task and that such an augmented network would better account for the fine-timescale trajectories of neural responses in the visual pathway.

Although CNNs augmented with recurrence have been used to solve relatively simple variants of occlusion and future-prediction tasks [Spoerer et al., 2017, Lotter et al., 2017], these models have neither been shown to generalize to harder tasks already performed by feedforward CNNs — such as recognizing objects in the ImageNet dataset [Deng et al., 2009, Krizhevsky et al., 2012] — nor explained neural responses as well as ImageNet-optimized CNNs do. As of this writing, object recognition on ImageNet is the only task known to produce CNN activation patterns comparable to those of neurons throughout visual cortex [Khaligh-Razavi and Kriegeskorte, 2014, Yamins et al., 2014, Cadena et al., 2017]. In fact, because of its wide variety and complexity, ImageNet contains many images with properties that might make use of recurrent processing according to the hypotheses described above (e.g. heavy occlusion, the presence of multiple foreground objects, etc). Moreover, some of the most effective recent solutions to ImageNet (e.g. the ResNet models [He et al., 2016]) repeat the same structural motif across many layers, which suggests that they might be approximable by the temporal unrolling of shallower recurrent networks [Liao and Poggio, 2016]. We therefore chose to explore whether recurrence could improve classification performance on the ImageNet dataset. Although other work has used the output of CNNs as input to RNNs to solve visual tasks such as object segmentation [McIntosh et al., 2017], here we integrate recurrent structures within the CNN itself, as such structures have been ubiquitously observed in the neuroscience literature.

We found that standard recurrent cell types (e.g. vanilla RNNs, LSTMs [Elman, 1990, Hochreiter and Schmidhuber, 1997]) do not increase ImageNet performance above parameter-matched feedforward baselines. However, we designed novel local cell architectures, embodying structural properties useful for integration into CNNs, that do. To identify even better structures within the large space of model architectures, we then performed an automated search over thousands of models that varied in their local recurrent cells and long-range feedback connections. Strikingly, this procedure discovered new patterns of recurrence not found in conventional RNNs: for instance, the most successful models exclusively used depth-separable convolutions for their local recurrent connections, which multiplicatively gated the networks' ResNet-like feedforward backbone. In addition, a small subset

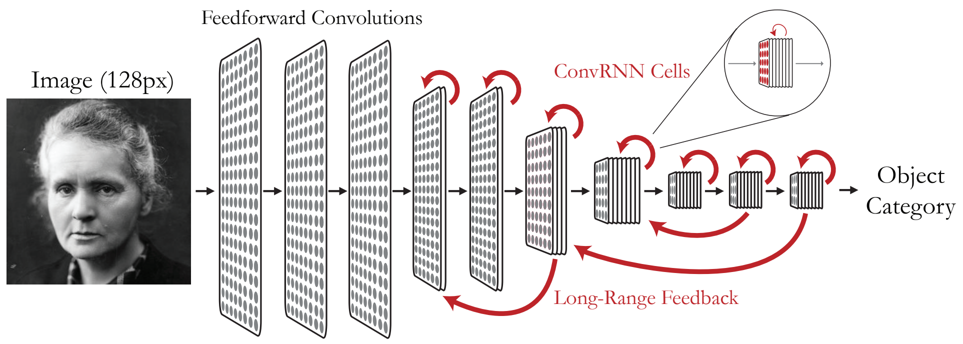


Figure 1: **Schematic of model architecture.** Convolutional recurrent networks (ConvRNNs) have a combination of local recurrent cells and long-range feedback connections added on top of a CNN backbone. In our implementation, propagation along each black or red arrow takes one time step (10 ms) to mimic conduction delays between cortical layers.

of long-range feedback connections boosted task performance even as the majority had a neutral or negative effect. Overall this search resulted in recurrent models that matched the performance of a much deeper feedforward architecture (ResNet-34) while using only $\sim 75\%$ as many parameters. Finally, comparing recurrent model features to neural responses in the primate visual system, we found that ImageNet-optimized ConvRNNs provide a quantitatively accurate model of neural dynamics at a 10 millisecond resolution across intermediate and higher visual cortical areas. These results therefore offer a model of how local and long-range recurrence in the visual system might be adapted for performing object recognition in the primate visual system.

2 Methods

Computational Architectures. To explore the architectural space of ConvRNNs, we used the Tensorflow library [Abadi et al., 2016] to augment standard CNNs with both local and long-range recurrence (Figure 1). Because we sought eventually to compare these models with neural data, it was critical that time flow realistically. Conduction from one area to another in visual cortex takes approximately 10 milliseconds (ms) [Mizuseki et al., 2009], with signal from photoreceptors reaching IT cortex at the top of the ventral stream by 70-100 ms. On the other hand, neural dynamics indicating potential recurrent connections take place over the course of 100-200 ms [Issa et al., 2018]. A single feedforward volley of responses thus cannot be treated as if it were instantaneous relative to the timescale of recurrence and feedback. Hence, rather than treating each entire feedforward pass from input to output as one integral time step, as is normally done with RNNs [Spoerer et al., 2017], each time step in our models corresponds to a single feedforward layer processing its input and passing it to the next layer. This choice required an unrolling scheme different from that used in the standard Tensorflow RNN library, the code for which (and for all of our models) can be found at `https://github.com/neuroailab/tnn`.

Within each ConvRNN layer, feedback inputs from higher layers are resized to match the spatial dimensions of the feedforward input to that layer. Both types of input are processed by standard 2-D convolutions. If there is any local recurrence at that layer, the output is next passed to the recurrent cell as input. Feedforward and feedback inputs are combined within the recurrent cell (see Supplemental Methods.) Finally, the output of the cell is passed through any additional nonlinearities, such as max-pooling. The generic update rule for the discrete-time trajectory of such a network is thus $H_{t+1}^{\ell} = C_{\ell}\left(F_{\ell}\left(\bigoplus_{j \neq \ell} R_t^j\right), H_t^{\ell}\right)$, and $R_t^{\ell} = A_{\ell}(H_t^{\ell})$, where R_t^{ℓ} is the output of layer ℓ at time t and H_t^{ℓ} is the hidden state of the locally recurrent cell C_{ℓ} at time t. A_{ℓ} is the activation function and any pooling post-memory operations, and $\oplus$ denotes concatenation along the channel dimension with appropriate resizing to align spatial dimensions. The learned parameters of such a network consist of F_{ℓ}, comprising any feedforward and feedback connections coming into layer $\ell = 1, \ldots, L$, and any of the learned parameters associated with the local recurrent cell C_{ℓ}.

In this work, all forms of recurrence add parameters to the feedforward base model. Because this could improve task performance for reasons unrelated to recurrent computation, we trained two types of control model to compare to ConvRNNs: (1) Feedforward models with more convolution filters ("wider") or more layers ("deeper") to approximately match the number of parameters in a recurrent model; and (2) Replicas of each ConvRNN model unrolled for a minimal number of time steps, defined as the number that allows all model parameters to be used at least once. A control model from this second class has exactly the same number of parameters as its fully unrolled comparison ConvRNN, so any increase in performance from unrolling longer can be attributed to recurrent computation. Fully and minimally unrolled ConvRNNs were trained separately with identical learning hyperparameters.

3 Results

Novel RNN Cell Structures Improve Task Performance. We first tested whether augmenting CNNs with standard RNN cells, such as vanilla RNNs and LSTMs, could improve performance on ImageNet object recognition (Figure 2a). We found that introducing these cells added a small amount of accuracy when introduced into an AlexNet-like 6-layer feedforward backbone (Figure 2b). However, there were two problems with these recurrent architectures: first, these ConvRNNs did not perform substantially better than parameter-matched minimally unrolled controls, suggesting that the observed performance gain was due to an increase in the number of unique parameters rather than recurrent computation. Second, making the feedforward model wider or deeper yielded an even larger performance gain than adding these standard RNN cells, but with fewer parameters. Adding other recurrent cell structures from the literature, including the UGRNN and IntersectionRNN, had similar effects to adding LSTMs (data not shown) [Collins et al., 2017]. This suggested that standard RNN cell structures, although well-suited for a range of temporal tasks, are less well-suited for inclusion within deep CNNs.

We speculated that this was because standard cells lack a combination of two key properties: 1) **Gating**, in which the value of a hidden state determines how much of the bottom-up input is passed through, retained, or discarded at the next time step; and 2) **Bypassing**, where a zero-initialized hidden state allows feedforward input to pass on to the next layer unaltered, as in the identity shortcuts of ResNet-class architectures (Figure 2a, top left.) Importantly, both of these features are thought to address the problem of gradients vanishing as they are backpropagated to early time steps of a recurrent network or to early layers of a deep feedforward network, respectively. LSTMs employ gating, but no bypassing, as their inputs must pass through several nonlinearities to reach their output; whereas vanilla RNNs do bypass a zero-initialized hidden state, but do not gate their input (Figure 2a).

We thus implemented recurrent cell structures with both features to determine whether they function better than standard cells within CNNs. One example of such a structure is the "Reciprocal Gated Cell", which bypasses its zero-initialized hidden state and incorporates LSTM-like gating (Figure 2a, bottom right, see Supplemental Methods for the cell equations). Adding this cell to the 6-layer CNN improved performance substantially relative to both the feedforward baseline and minimally unrolled, parameter-matched control version of this model. Moreover, the Reciprocal Gated Cell used substantially fewer parameters than the standard cells to achieve greater accuracy (Figure 2b).

However, this advantage over LSTMs could have resulted from an accidentally better choice of training hyperparameters for the Reciprocal Gated Cells. It has been shown that different RNN structures can succeed or fail to perform a given task because of differences in trainability rather than differences in capacity [Collins et al., 2017]. To test whether the LSTM models underperformed for this reason, we searched over training hyperparameters and common structural variants of the LSTM to better adapt this local structure to deep convolutional networks, using hundreds of second generation Google Cloud Tensor Processing Units (TPUv2s). We searched over learning hyperparameters (e.g. gradient clip values, learning rate), as well as structural hyperparameters separately for each layer (e.g. gate convolutional filter sizes, channel depth, whether or not to use peephole connections, etc). While this did yield a better LSTM-based ConvRNN, it did not eclipse the performance of the smaller Reciprocal Gated Cell-based ConvRNN; moreover, applying the same hyperparameter optimization procedure to the Reciprocal Gated Cell models equivalently increased that architecture class' performance and further reduced its parameter count (Figure 2b). Taken together, these results suggest that the choice of local recurrent structure strongly affects the accuracy of a ConvRNN on a challenging object recognition task. Although previous work has shown that

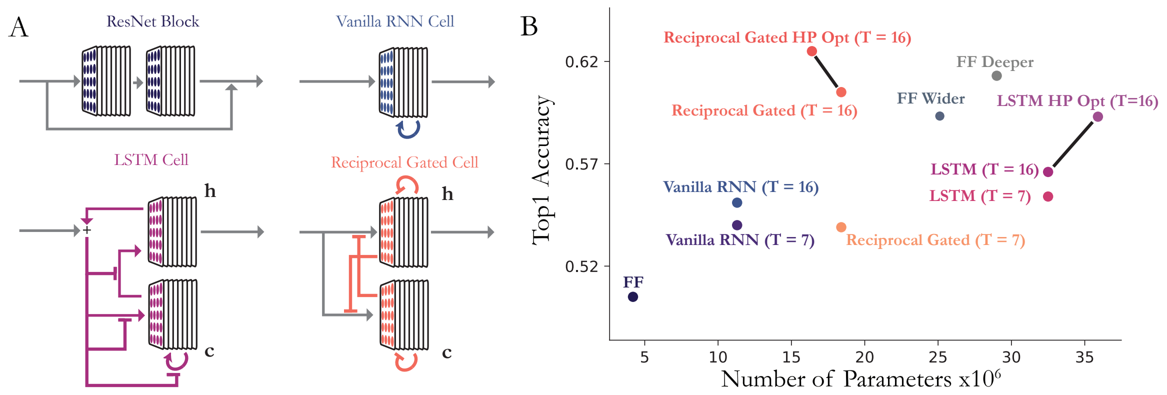


Figure 2: **Comparison of local recurrent cell architectures**. **(a) Architectural differences between ConvRNN cells.** Standard ResNet blocks and vanilla RNN Cells have bypassing (see text). The LSTM cell has gating, denoted by 'T'-junctions, but no bypassing. The Reciprocal Gated Cell has both. **(b) Performance of various ConvRNN and feedforward models as a function of number of parameters.** Colored points incorporate the respective RNN cell into the the 6-layer feedforward architecture ("FF"). 'T' denotes number of unroll steps. Hyperparameter-optimized versions of the LSTM and Reciprocal Gated Cell ConvRNNs are connected to their non-optimized versions by black lines.

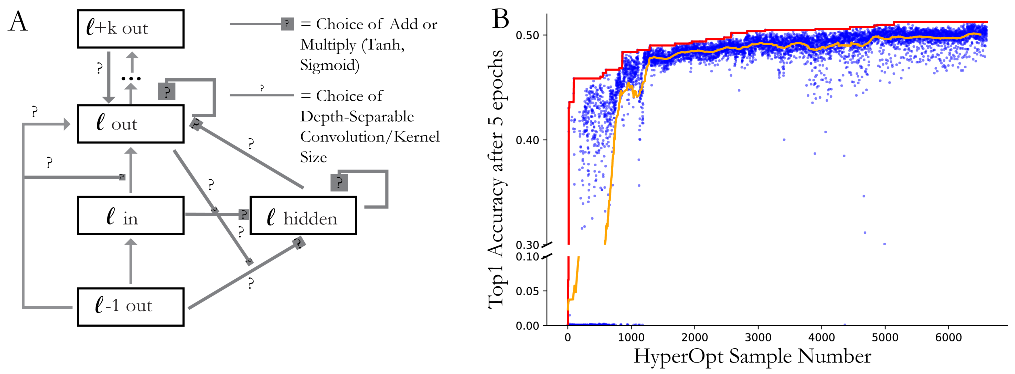


Figure 3: **ConvRNN hyperparametrization and search results.** **(a) Hyperparametrization of local recurrent cells.** Arrows indicate connections between cell input, hidden states, and output. Question marks denote optional connections, which may be conventional or depth-separable convolutions with a choice of kernel size. Feedforward connections between layers ($\ell - 1$ out, ℓ in, and ℓ out) are always present. Boxes with question marks indicate a choice of multiplicative gating with sigmoid or tanh nonlinearities, addition, or identity connections (as in ResNets.) Finally, long-range feedback connections from $\ell + k$ out may enter at either the local cell input, hidden state, or output. **(b) ConvRNN search results.** Each blue dot represents a model, sampled from hyperparameter space, trained for 5 epochs. The orange line is the average performance of the last 50 models. The red line denotes the top performing model at that point in the search.

standard RNN structures can improve performance on simpler object recognition tasks, these gains have not been shown to generalize to more challenging datasets such as ImageNet [Liao and Poggio, 2016, Zamir et al., 2017]. Difficult tasks may therefore be needed to differentiate the capacity of different recurrent architectures.

Hyperparameter Optimization for Deeper Recurrent Architectures. The success of Reciprocal Gated Cells on ImageNet prompted us to search for even better-adapted recurrent structures, in the context of deeper, more realistic feedforward backbones. In particular, we designed a novel search space over ConvRNN architectures that generalized that cell's use of gating and bypassing. To measure the quality of a large number of architectures, we installed RNN cell variants in a convolutional network based on the ResNet-18 model [He et al., 2016]. Deeper ResNet models,

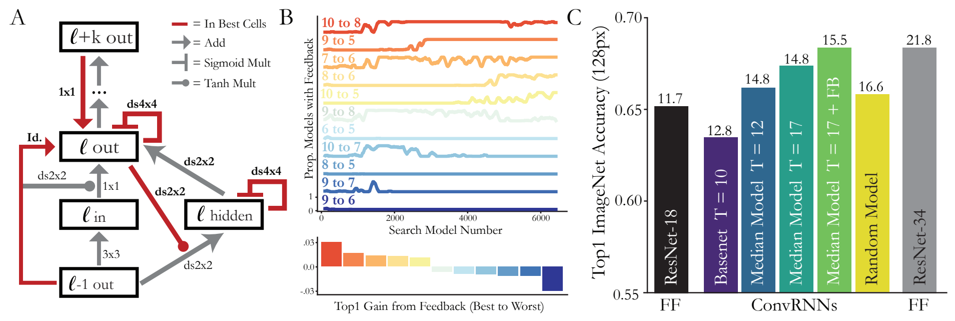


Figure 4: **Optimal local recurrent cell motif and global feedback connectivity. (a) RNN Cell structure from the top-performing search model.** Red lines indicate that this hyperparameter choice (connection and filter size) was chosen in each of top unique models the search (red line in 3b.) KxK denotes a convolution and dsKxK a depth-separable convolution with filter size KxK. **(b) Long-range feedback connections from the search.** (Top) Each trace shows the proportion of models in a 100-sample window that have a particular feedback connection. (Bottom) Each bar indicates the difference between the median performance of models with a given feedback and the median performance of models without that feedback. Colors correspond to the same feedbacks as above. **(c) Performance of models fully trained on 128px-size ImageNet.** We compared the performance of ResNet-18, an 18-layer feedforward base model (basenet) modeled after ResNet-18, the median model from the search with or without global feedback connectivity, and its minimally-unrolled control ($T = 12$). The "Random Model" was selected randomly from the initial, random phase of the model search. Parameter counts in millions are shown on top of each bar. ResNet models were trained as in [He et al., 2016] but with 128px images to compare to ConvRNNs.

such as ResNet-34, reach higher performance on ImageNet and therefore provide a benchmark for how much recurrent computation could substitute for feedforward depth in this model class (see below). Recurrent architectures varied in (a) how multiple inputs and hidden states were combined to produce the output of each cell; (b) which linear operations were applied to each input or hidden state; (c) which point-wise nonlinearities were applied to each hidden state; and (d) the values of several additive and multiplicative constants (Figure 3a). The original Reciprocal Gated Cell corresponded to a single point within this large hyperparameter space. We simultaneously searched over all possible long-range feedback connections, which use the output of a later layer as input to an earlier ConvRNN layer or cell hidden state. Finally, we varied learning hyperparameters of the network, such as learning rate and number of time steps to present the input image.

Since we did not want to make assumptions about how hyperparameters interact, we jointly optimized architectural and learning hyperparameters using a Tree-structured Parzen Estimator (TPE) [Bergstra et al., 2011], a type of Bayesian algorithm which searches over continuous and categorical variable configurations (see Supplemental Methods for details of the search.) This procedure trained hundreds of models simultaneously, each for 5 epochs. To reduce the amount of search time, we trained all models on 128 px ImageNet images. For CNNs, performance on images this size nearly equals performance on full size images [Mishkin et al., 2016].

Most models in this architecture space failed. In the initial TPE phase of randomly sampling hyperparameters, more than 80% of models either failed to improve task performance above chance or had exploding gradients (Figure 3b, bottom left of scatter plot.) However, over the course of the search, an increasing fraction of the models reached higher performance. The median and peak sample performance increased over the next $\sim$ 7000 model samples, with the best model reaching performance >15% higher than the top model from the initial phase of the search (Figure 3b). Like the contrast between LSTMs and Reciprocal Gated Cells, these dramatic improvements further support our hypothesis that some recurrent architectures are substantially better than others for object recognition at scale.

Sampling thousands of recurrent architectures allowed us to ask which structural features contribute most to task performance and which are relatively unimportant. Because TPE does not sample models according to a stationary parameter distribution, it is difficult to infer the causal roles of individual hyperparameters. Still, several clear patterns emerged in the sequence of best-performing architectures over the course of the search (Figure 3b, red line). These models used both bypassing and gating within their local RNN cells; depth-separable convolutions for updating hidden states, but conventional convolutions for connecting bottom-up input to output (Figure 4a); and several key long-range feedbacks, which were "discovered" early in the search, then subsequently used in the majority of models (Figure 4b). Overall, these recurrent structures act like feedforward ResNet blocks on their first time step but incorporate a wide array of operations on future time steps, including many of the same ones selected for in other work that optimized feedforward modules [Zoph et al., 2017]. Thus, this diversity of operations and pathways from input to output may reflect a natural solution for object recognition, whether implemented through recurrent or feedforward structures.

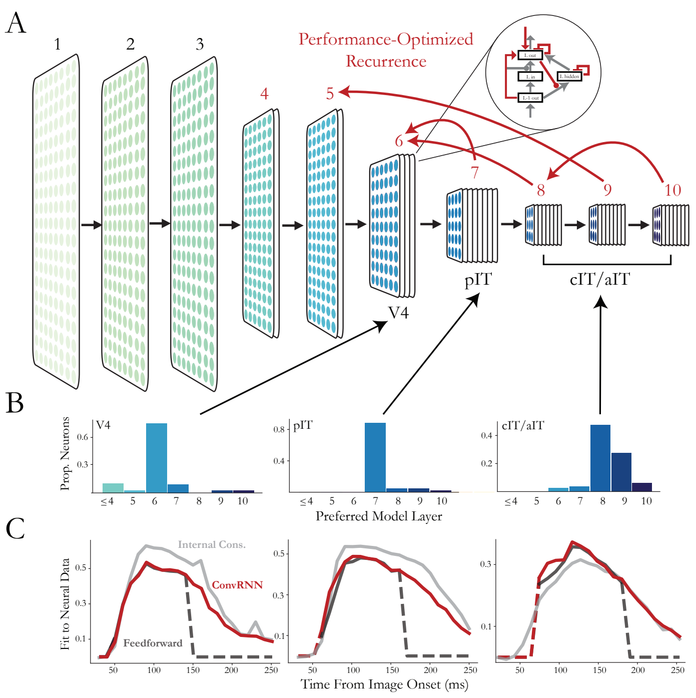
If the primate visual system uses recurrence in lieu of greater network depth to perform object recognition, then a shallower recurrent model with a suitable form of recurrence should achieve recognition accuracy equal to a deeper feedforward model [Liao and Poggio, 2016]. We therefore tested whether our search had identified such well-adapted recurrent architectures by fully training a representative ConvRNN, the model with the median five-epoch performance after 5000 TPE samples. This median model reached a final Top1 ImageNet accuracy equal to a ResNet-34 model with nearly twice as many layers, even though the ConvRNN used only $\sim 75\%$ as many parameters (ResNet-34, 21.8 M parameters, .684 Validation Top1; Median ConvRNN, 15.5 M parameters, .684 Validation Top1.) Neither the minimally unrolled control version of this model nor a fully unrolled model from the random phase of the search performed substantially better than ResNet-18, despite using as many or or more parameters (Figure 4d). Together these results demonstrate that the median model uses recurrent computations to perform object recognition and that particular features of its recurrent architecture, found through hyperparameter optimization, are required for its improved accuracy. Thus, given well-selected forms of local recurrence and long-range feedback, a ConvRNN can do the same challenging visual task as a deeper feedforward CNN.

ConvRNNs Capture Neuronal Dynamics in the Primate Visual System. ConvRNNs produce a dynamic time series of outputs given a static input, unlike feedforward networks. While these recurrent dynamics could be used for tasks involving time, here we optimized the ConvRNNs to perform the "static" task of object classification on ImageNet. It is possible that the primate visual system is optimized for such a task, because even static images produce reliably dynamic neural response trajectories at temporal resolutions of tens of milliseconds [Issa et al., 2018]. The object content of some images becomes decodable from the neuronal population significantly later than the content of other images, even though animals recognize both object sets equally well. Interestingly, late-decoding images are not well classified by feedforward CNNs, raising the possibility that they are encoded in animals through recurrent computations [Kar et al., 2017]. If this were the case, then recurrent networks trained to perform a difficult, but static object recognition task might explain neural dynamics in the primate visual system, just as feedforward models explain time-averaged responses [Yamins et al., 2014, Khaligh-Razavi and Kriegeskorte, 2014]. We therefore asked whether our task-optimized ConvRNN model could predict the dynamic firing rates of primate neurons recorded during encoding of visual stimuli. These data included multi-electrode array recordings from the ventral visual pathway of rhesus macaque monkeys described in [Majaj et al., 2015] (see Supplemental Methods.)

We presented the fully trained median ConvRNN and its feedforward baseline with the same images shown to monkeys and collected the time series of features from each model layer. To decide which layer should be used to predict which neural responses, we fit linear models from each feedforward layer's features to the neural population and measured where explained variance on held-out images peaked (see Supplemental Methods.) Units recorded from distinct arrays – placed in the successive V4, posterior IT, and central/anterior IT cortical areas of the macaque – were fit best by the successive 6th, 7th, and 8/9th layers of the feedforward model, respectively (Figure 3a,b). This result extends and refines the mapping of successive CNN layers to defined anatomical areas within the ventral visual pathway [Yamins et al., 2014]. Finally, we measured how well the ConvRNN features from these layers predicted the dynamics of each unit. Critically, we used a single linear mapping across all times so that the intrinsic dynamics of the recurrent network had to account for the dynamics in the neural response, consistent with a fixed physical relationship between real neurons and model units.

On held-out images, the ConvRNN feature dynamics fit the neural response trajectories as well as or better than the feedforward baseline features for almost every 10 ms time bin (Figure 3c). Moreover, because the feedforward model has the same square wave dynamics as its 100ms visual input, it

Figure 5: **Modeling primate ventral stream neural dynamics with ConvRNNs.** (a) The ConvRNN model used to fit neural dynamics has local recurrent cells at layers 4 through 10 and the long-range feedbacks denoted by the red arrows. (b) Consistent with the brain's ventral hierarchy, most units from V4 are best fit by layer 6 features; pIT by layer 7; and cIT/aIT by layers 8/9. (c) Fitting model features to neural dynamics approaches the noise ceiling of these responses. The y-axis indicates the median across units of the correlation between predictions and ground-truth responses on held-out images.



cannot predict neural responses after image offset plus a fixed delay, corresponding to the number of layers (Figure 3c dashed black lines.) In contrast the activations of ConvRNN units have persistent dynamics, yielding strong predictions of the entire neural response trajectories. This improvement is especially clear when comparing the predicted temporal responses of the feedforward and ConvRNN models on single images (Figure S1a): ConvRNN predictions have dynamic trajectories that resemble smoothed versions of the ground truth responses, whereas feedforward predictions necessarily have the same time course for every image. Crucially, this results from the nonlinear dynamics learned on ImageNet, as both models are fit to neural data with the same form of temporally-shared linear model the same number of parameters. Across all recorded units and held-out images, the ConvRNN shows a substantial advantage over the feedforward model in its prediction of single-image temporal responses (Figure S1b). Thus, a recurrent object recognition model trained on ImageNet may account for neural dynamics in the primate ventral pathway.

4 Conclusion and Future Directions

Here, we show that careful introduction of recurrence into CNNs can improve performance on a challenging object recognition task, as even exhaustive hyperparameter optimization of traditional recurrent cells demonstrated inferior ImageNet performance compared to hand-designed custom cells. In light of suggestions that cortical microcircuits might be well-modeled by LSTMs [Costa et al., 2017], our findings raise the possibility that different local recurrent structures may be adapted for different behaviors within the brain. In fact, we found strong selection for the combination gating, bypassing, and depth-separability in a local recurrent cell. With the proper local recurrence in place, specific patterns of long-range feedback connections also performance. Furthermore, although all nonlinear dynamics in these models were optimized only to perform a task, these dynamics provide strong estimates of neural dynamics in the primate ventral stream. Thus we extend a core principle from the feedforward to the recurrent regime: given an appropriate architecture class, task-optimized recurrent neural networks provide the quantitatively best normative models of visual encoding dynamics in the primate brain.

These results differ from related work in which CNNs were supplemented with recurrence at several layers or reconfigured into recurrent structures. In those cases, recurrence served either to (a) perform a dynamic task with time-varying inputs and outputs (e.g. future prediction [Lotter et al., 2017]), (b) solve object recognition with substantially fewer parameters than in a deeper feedforward structure [Leroux et al., 2018], (c) strictly improved object recognition performance, but on tasks substantially less challenging than ImageNet categorization [Spoerer et al., 2017, Liao and Poggio, 2016, Zamir et al., 2017], or (d) achieved only very small performance increases with recurrence [Li et al., 2018]. Despite their merits, none of these approaches has (to our knowledge) produced quantitatively strong models of neural dynamics, nor substantially improved performance on ImageNet above the feedforward backbones they extend (although see Cichy et al. [2016], which predicts later timepoints of neural dynamics with deep layers of a feedforward CNN). However, it will be important to test whether such models can predict aspects of neural dynamics even without achieving higher ImageNet accuracy, for instance by looking at responses specific to occluded objects.

We believe that applying the approach we took here in concert with CNNs that use more sophisticated feedforward motifs, such as those in NASNet [Zoph et al., 2017], could lead to improvements to these state-of-the-art baselines. While training such models on full-sized ImageNet images would require resources beyond those that we currently possess, such an effort could be fruitful. Further experiments will explore whether different tasks, such as those that do not rely on difficult-to-obtain category labels, can substitute for supervised object recognition in matching ConvRNNs to neural responses. We will also test whether our models can withstand other forms of temporal noise or nonrandom movement in an input movie, or even use them as a form of “data augmentation” to build spatio-temporally invariant representations of objects.

5 Acknowledgments

This project has received funding from the James S. McDonnell foundation (grant #220020469, D.L.K.Y.), the Simons Foundation (SCGB grants 543061 (D.L.K.Y) and 325500/542965 (J.J.D)), the US National Science Foundation (grant iis-ri1703161, D.L.K.Y.), the European Union’s Horizon 2020 research and innovation programme (agreement #705498, J.K.), the US National Eye Institute (grant R01-EY014970, J.J.D.), and the US Office of Naval Research (MURI-114407, J.J.D). We thank Google (TPUv2 team) and the NVIDIA corporation for generous donation of hardware resources.

References

M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. Tensorflow: A system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.

J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl. Algorithms for hyper-parameter optimization. In *NIPS*, 2011.

J. Bergstra, B. Komer, C. Eliasmith, D. Yamins, and D. D. Cox. Hyperopt: a python library for model selection and hyperparameter optimization. *Computational Science & Discovery*, 8(1), 2015.

S. A. Cadena, G. H. Denfield, E. Y. Walker, L. A. Gatys, A. S. Tolias, M. Bethge, and A. S. Ecker. Deep convolutional models improve predictions of macaque v1 responses to natural images. *bioRxiv*, page 201764, 2017.

R. M. Cichy, A. Khosla, D. Pantazis, A. Torralba, and A. Oliva. Comparison of deep neural networks to spatio-temporal cortical dynamics of human visual object recognition reveals hierarchical correspondence. *Sci Rep.*, 6, 2016. doi: 10.1038/srep27755.

J. Collins, J. Sohl-Dickstein, and D. Sussillo. Capacity and trainability in recurrent neural networks. In *ICLR*, 2017.

R. Costa, I. A. Assael, B. Shillingoford, N. de Freitas, and T. Vogels. Cortical microcircuits as gated-recurrent neural networks. In *NIPS*, 2017.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

J. J. DiCarlo, D. Zoccolan, and N. C. Rust. How does the brain solve visual object recognition? *Neuron*, 73(3):415–34, 2012.

J. L. Elman. Finding structure in time. *Cognitive Science*, 14(2):179–211, 1990.

C. D. Gilbert and L. Wu. Top-down influences on visual processing. *Nat. Rev. Neurosci.*, 14(5): 350–363, 2013.

U. Güçlü and M. A. van Gerven. Deep neural networks reveal a gradient in the complexity of neural representations across the ventral stream. *The Journal of Neuroscience*, 35(27):10005–10014, 2015.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. doi: https://doi.org/10.1109/CVPR.2016.90.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

C. P. Hung, G. Kreiman, T. Poggio, and J. J. DiCarlo. Fast readout of object identity from macaque inferior temporal cortex. *Science*, 310(5749):863–6, 2005.

E. B. Issa, C. F. Cadieu, and J. J. DiCarlo. Neural dynamics at successive stages of the ventral visual stream are consistent with hierarchical error signals. *bioRxiv preprint*, 2018. doi: 10.1101/092551.

W. James. The principles of psychology (vol. 1). *New York: Holt*, 474, 1890.

K. Kar, J. Kubilius, E. B. Issa, K. M. Schmidt, and J. J. DiCarlo. Evidence that feedback is required for object identity inferences computed by the ventral stream. *COSYNE (Computational and Systems Neuroscience) 2017 Conference Poster II-37*, 2017.

S.-M. Khaligh-Razavi and N. Kriegeskorte. Deep supervised, but not unsupervised, models may explain it cortical representation. *PLoS computational biology*, 10(11):e1003915, 2014.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

S. Leroux, P. Molchanov, P. Simoens, B. Dhoedt, T. Breuel, and J. Kautz. Iamnn: iterative and adaptive mobile neural network for efficient image classification. In *ICLR Workshop 2018*, 2018.

X. Li, Z. Jie, J. Feng, C. Liu, and S. Yan. Learning with rethinking: recurrently improving convolutional neural networks through feedback. *Pattern Recognition*, 79:183–194, 2018.

Q. Liao and T. Poggio. Bridging the gaps between residual learning, recurrent neural networks and visual cortex. *arXiv preprint arXiv:1604.03640*, 2016.

G. W. Lindsay. Feature-based attention in convolutional neural networks. *arXiv preprint arXiv:1511.06408*, 2015.

D. Linsley, J. Kim, V. Veerabadran, and T. Serre. Learning long-range spatial dependencies with horizontal gated-recurrent units. *arXiv preprint arXiv:1805.08315*, 2018.

W. Lotter, G. Kreiman, and D. Cox. Deep predictive coding networks for video prediction and unsupervised learning. In *ICLR*, 2017.

N. J. Majaj, H. Hong, E. A. Solomon, and J. J. DiCarlo. Simple learned weighted sums of inferior temporal neuronal firing rates accurately predict human core object recognition performance. *Journal of Neuroscience*, 35(39):13402–13418, 2015.

V. Mante, D. Sussillo, K. V. Shenoy, and W. T. Newsome. Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature*, 503:78–84, 2013.

L. McIntosh, N. Maheswaranathan, D. Sussillo, and J. Shlens. Recurrent segmentation for variable computational budgets. *arXiv preprint arXiv:1711.10151*, 2017.

C. Michaelis, M. Bethge, and A. S. Ecker. One-shot segmentation in clutter. *arXiv preprint arXiv:1803.09597*, 2018.

D. Mishkin, N. Sergievskiy, and J. Matas. Systematic evaluation of cnn advances on the imagenet. *arXiv preprint arXiv:1606.02228*, 2016.

K. Mizuseki, A. Sirota, E. Pastalkova, and G. Buzsáki. Theta oscillations provide temporal windows for local circuit computation in the entorhinal-hippocampal loop. *Neuron*, pages 267–280, 2009.

N. Pinto, D. D. Cox, and J. J. Dicarlo. Why is real-world visual object recognition hard? *PLoS Computational Biology*, 2008.

K. Rajaei, Y. Mohsenzadeh, R. Ebrahimpour, and S.-M. Khaligh-Razavi. Beyond core object recognition: recurrent processes account for object recognition under occlusion. *bioRxiv preprint*, 2018. doi: 10.1101/302034.

R. P. Rao and D. H. Ballard. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nat. Neurosci.*, 2(1):79–87, Jan 1999.

J. Shi, H. Wen, Y. Zhang, K. Han, and Z. Liu. Deep recurrent neural network reveals a hierarchy of process memory during dynamic natural vision. *Hum Brain Mapp.*, 39:2269–2282, 2018.

C. J. Spoerer, P. McClure, and N. Kriegeskorte. Recurrent convolutional neural networks: a better model of biological object recognition. *Front. Psychol.*, 8:1–14, 2017.

D. L. Yamins and J. J. DiCarlo. Using goal-driven deep learning models to understand sensory cortex. *Nature neuroscience*, 19(3):356, 2016.

D. L. K. Yamins, H. Hong, C. F. Cadieu, E. A. Solomon, D. Seibert, and J. J. DiCarlo. Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the National Academy of Sciences*, 111(23):8619–8624, 2014. doi: 10.1073/pnas.1403112111.

A. R. Zamir, T.-L. Wu, L. Sun, W. B. Shen, B. E. Shi, J. Malik, and S. Savarese. Feedback networks. In *CVPR*, 2017. doi: 10.1109/CVPR.2017.196.

B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le. Learning transferable architectures for scalable image recognition. *arXiv preprint arXiv:1707.07012*, 2017.

6 Supplemental Methods

6.1 CNN architectures

For all of our CNN architectures, we use Elu nonlinearities. The 6-layer feedforward CNN, referenced in Figure 2b as “FF”, had the following architecture:

Layer	Kernel Size	Channels	Stride	Max Pooling
1	7×7	64	2	2×2
2	3×3	128	1	2×2
3	3×3	256	1	2×2
4	3×3	256	1	2×2
5	3×3	512	1	2×2
6	2×2	1000	1	No

The variant of the above 6-layer feedforward CNN, referenced in Figure 2b as “FF wider” is given below:

Layer	Kernel Size	Channels	Stride	Max Pooling
1	7×7	128	2	2×2
2	3×3	512	1	2×2
3	3×3	512	1	2×2
4	3×3	512	1	2×2
5	3×3	1024	1	2×2
6	2×2	1000	1	None

The “FF deeper” model referenced in Figure 2b is given below:

Layer	Kernel Size	Depth	Stride	Max Pooling
1	7×7	64	2	2×2
2	3×3	64	1	None
3	3×3	64	1	None
4	3×3	128	1	2×2
5	3×3	128	1	None
6	3×3	256	1	2×2
7	3×3	256	1	2×2
8	3×3	512	1	None
9	3×3	512	1	None
10	3×3	512	1	2×2
11	None (Avg. Pool FC)	1000	None	None

The 18 layer CNN modeled after ResNet-18 (using MaxPooling rather than stride-2 convolutions to perform downsampling) is given below:

Block	Kernel Size	Depth	Stride	Max Pooling	Repeat
1	7×7	64	2	2×2	$\times 1$
2	3×3	64	1	None	$\times 2$
3	3×3	64	1	None	$\times 2$
4	3×3	128	1	2×2	$\times 2$
5	3×3	128	1	None	$\times 2$
6	3×3	256	1	2×2	$\times 2$
7	3×3	256	1	2×2	$\times 2$
8	3×3	512	1	None	$\times 2$
9	3×3	512	1	None	$\times 2$
10	3×3	512	1	2×2	$\times 2$
11	None (Avg. Pool FC)	1000	None	None	$\times 1$

6.2 Reciprocal Gated Cell Equation

We will provide the update equations for the "Reciprocal Gated Cell", diagrammed in Figure 2a (bottom right). Let $\circ$ denote Hadamard (elementwise) product, let $*$ denote convolution, and let x_t^ℓ denote the input to the cell at layer ℓ. Then the update equations are as follows:

The update equation for the output of the cell, h_t^ℓ, is given by a gating of both the input and memory c_t^ℓ.

$$\begin{aligned} a_{t+1}^\ell &= (1 - \sigma(W_{ch}^\ell * c_t^\ell)) \circ x_t^\ell + (1 - \sigma(W_{hh}^\ell * h_t^\ell)) \circ h_t^\ell \\ h_t^\ell &= f\left(a_t^\ell\right). \end{aligned} \tag{1}$$

The update equation for the memory c_t^ℓ is given by a gating of the input and the output of the cell h_t^ℓ.

$$\begin{aligned} \tilde{c}_{t+1}^\ell &= (1 - \sigma(W_{hc}^\ell * h_t^\ell)) \circ x_t^\ell + (1 - \sigma(W_{cc}^\ell * c_t^\ell)) \circ c_t^\ell \\ c_t^\ell &= f(\tilde{c}_t^\ell). \end{aligned} \tag{2}$$

6.3 RNN Cell Search Parametrization

Inspired by the Reciprocal Gated Cell equations above, there are a variety of possibilities for how $h_{t-1}^\ell, x_t^\ell, c_t^\ell$, and h_t^ℓ can be connected to one another. Figure 3a schematizes these possibilities.

Mathematically, Figure 3a can be formalized in terms of the following update equations.

First, we define our input sets and building block functions:

$$
\begin{aligned}
&minin = \{h_{t-1}^{\ell-1}, x_t^\ell, c_{t-1}^\ell, h_{t-1}^\ell\} \\
&minin_a = minin \cup \{c_t^\ell\} \\
&minin_b = minin \cup \{h_t^\ell\} \\
&S_a \subseteq minin_a \\
&S_b \subseteq minin_b \\
&\text{Affine}(x) \in \{+, 1 \times 1 \text{ conv}, K \times K \text{ conv}, K \times K \text{ depth-separable conv}\} \\
&K \in \{3, \ldots, 7\}
\end{aligned}
$$

With those in hand, we have the following update equations:

$$
\begin{aligned}
&\tau_a = v_1^\tau + v_2^\tau \sigma(\text{Affine}(S_a)) \\
&\tau_b = v_1^\tau + v_2^\tau \sigma(\text{Affine}(S_b)) \\
&gate_a = v_1^g + v_2^g \sigma(\text{Affine}(S_a)) \\
&gate_b = v_1^g + v_2^g \sigma(\text{Affine}(S_b)) \\
&a_t^\ell = \{gate_a\} \cdot in_t^\ell + \{\tau_a\} \cdot h_{t-1}^\ell \\
&h_t^\ell = f(a_t^\ell) \\
&b_t^\ell = \{gate_b\} \cdot in_t^\ell + \{\tau_b\} \cdot c_{t-1}^\ell \\
&c_t^\ell = f(b_t^\ell) \\
&f \in \{\text{elu}, \tanh, \sigma\}.
\end{aligned}
$$

For clarity, the following matrix summarizes the connectivity possibilities (with ? denoting the possibility of a connection):

$$
\begin{array}{c|cccccc}
 & h_{t-1}^{\ell-1} & x_t^\ell & c_{t-1}^\ell & c_t^\ell & h_{t-1}^\ell & h_t^\ell \\
h_{t-1}^{\ell-1} & 0 & 1 & 0 & ? & 0 & ? \\
x_t^\ell & 0 & 0 & 0 & ? & 0 & ? \\
c_{t-1}^\ell & 0 & 0 & 0 & ? & 0 & ? \\
c_t^\ell & 0 & 0 & 0 & 0 & 0 & ? \\
h_{t-1}^\ell & 0 & 0 & 0 & ? & 0 & ? \\
h_t^\ell & 0 & 0 & 0 & ? & 0 & 0
\end{array}
$$

6.4 Training ConvRNN Models

ConvRNN models and ResNet models in this work were trained by stochastic gradient descent using the same learning rate schedule and data augmentation methods as the original ResNet models [He et al., 2016] with the following modifications: we used a batch size of 128, an initial learning rate of 0.01, and Nesterov momentum of 0.9. Standard ImageNet cross-entropy loss was computed from the last time step of a ConvRNN. Gradients with absolute value greater than 0.7 were clipped.

6.5 Hyperparameter Optimization Methods

Models were trained synchronously 100 models at a time using the HyperOpt package [Bergstra et al., 2015], for 5 epochs each. Each model was trained on its own Tensor Processing Unit (TPUv2), and around 6000 models were sampled in total over the course of the search.

We employed a form of Bayesian optimization, a Tree-structured Parzen Estimator (TPE), to search the space of continuous and categorical hyperparameters [Bergstra et al., 2011]. This algorithm constructs a generative model of $P[score \mid configuration]$ by updating a prior from a maintained history H of hyperparameter configuration-loss pairs. The fitness function that is optimized over models is the expected improvement, where a given configuration c is meant to optimize $EI(c) = \int_{x<t} P[x \mid c, H]$. This choice of Bayesian optimization algorithm models $P[c \mid x]$ via a Gaussian mixture, and restricts us to tree-structured configuration spaces.

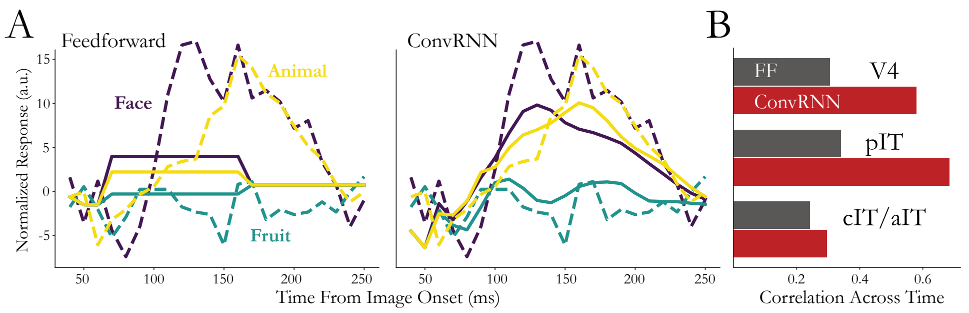


Figure S1: **Comparison of feedforward models and ConvRNNs to single images.** (a) Examples of a single pIT unit's response to three different images from the held-out validation set. Dashed lines are the ground truth response, solid lines are the model prediction. The feedforward model is forced to have the dynamics of a square wave. (b) Correlations of ground truth with predicted responses across time bins (50 to 250 ms.) The temporal correlation is computed for each image and for each unit. Plotted: the median across units from each array of the median temporal correlation across single held-out validation images.

6.6 Fitting Model Features to Neural Data

We fit trained model features to multi-unit array responses from [Majaj et al., 2015]. Briefly, we fit to 256 recorded sites from two monkeys. These came from three multi-unit arrays per monkey: one implanted in V4, one in posterior IT, and one in central and anterior IT. Each image was presented approximately 50 times, using rapid visual stimulus presentation (RSVP). Each stimulus was presented for 100 ms, followed by a mean gray background interleaved between images. Each trial lasted 250 ms. The image set consisted of 5120 images based on 64 object categories. Each image consisted of a 2D projection of a 3D model added to a random background. The pose, size, and x- and y-position of the object was varied across the image set, whereby 2 levels of variation were used (corresponding to medium and high variation from [Majaj et al., 2015].) Multi-unit responses to these images were binned in 10ms windows, averaged across trials of the same image, and normalized to the average response to a blank image. This produced a set of (5120 images x 256 units x 25 time bins) responses, which were the targets for our model features to predict. To estimate a noise ceiling for each mean response at each time bin, we computed the Spearman split-half reliability across trials.

The 5120 images were split 75-25 within each object category into a training set and a held-out testing set. All images were presented to the models for 10 time steps (corresponding to 100 ms), followed by a mean gray stimulus for the remaining 15 time steps, to match the image presentation to the monkeys.

ConvRNN or feedforward model features from each image (i.e. the activations of units in a given model layer) were linearly fit to the neural responses by stochastic gradient descent with a standard L2 loss. Each of the 256 units was fit independently. We stipulated that units from each multi-unit array must be fit by features from a single model layer. To determine which one, we fit the features from the feedforward model to each time bin of a given unit's response, averaged across time bins, and counted how many units had minimal loss for a given model layer. This yielded a mapping from the V4 array to model layer 6, pIT to layer 7, and cIT/aIT to layer 8. Finally, ConvRNN and feedforward model features were fit to the entire set of 25 time bins for each unit using a shared linear model: that is, a single set of regression coefficients was used for all time bins. The loss for this fitting was the average L2 loss across training images and 25 time bins for each unit.