



ELSEVIER



Going in circles is the way forward: the role of recurrence in visual inference

Ruben S van Bergen¹ and Nikolaus Kriegeskorte^{1,2,3,4}

Biological visual systems exhibit abundant recurrent connectivity. State-of-the-art neural network models for visual recognition, by contrast, rely heavily or exclusively on feedforward computation. Any finite-time recurrent neural network (RNN) can be unrolled along time to yield an equivalent feedforward neural network (FNN). This important insight suggests that computational neuroscientists may not need to engage recurrent computation, and that computer-vision engineers may be limiting themselves to a special case of FNN if they build recurrent models. Here we argue, to the contrary, that FNNs are a special case of RNNs and that computational neuroscientists and engineers should engage recurrence to understand how brains and machines can (1) achieve greater and more flexible computational depth (2) compress complex computations into limited hardware (3) integrate priors and priorities into visual inference through expectation and attention (4) exploit sequential dependencies in their data for better inference and prediction and (5) leverage the power of iterative computation.

Addresses

¹ Zuckerman Mind Brain Behavior Institute, Columbia University, New York, NY, United States

² Department of Psychology, Columbia University, New York, NY, United States

³ Department of Neuroscience, Columbia University, New York, NY, United States

⁴ Affiliated member, Electrical Engineering, Columbia University, New York, NY, United States

Current Opinion in Neurobiology 2020, 65:xx-yy

This review comes from a themed issue on **Whole-brain interactions between neural circuits**

Edited by **Karel Svoboda** and **Laurence Abbott**

<https://doi.org/10.1016/j.conb.2020.11.009>

0959-4388/© 2020 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

Introduction

The primate visual cortex uses a recurrent algorithm to process sensory input [1–3]. Anatomically, connectivity is cyclic. Neurons are connected in cycles within local cortical circuits [4–6]. Global inter-area connections are dense and mostly bidirectional [7–9]. Physiologically, the dynamics of neural responses bear temporal signatures indicative of recurrent processing [10,1,11]. Behaviorally,

visual perception can be disturbed by carefully timed interventions that coincide with the arrival of re-entrant information to a visual area [12–15]. The evidence for recurrent computation in the primate brain, thus, is unequivocal. What is less obvious, however, is *why* the brain uses a recurrent algorithm.

This question has recently been brought into sharper focus by the successes of deep feedforward neural network models (FNNs) [16,17]. These models now match or exceed human performance on certain visual tasks [18–20], and better predict primate recognition behavior [21,22,23] and neural activity [24–29] than current alternative models.

Although computer vision and computational neuroscience both have a long history of recurrent models [30–33], feedforward models have earned a dominant status in both fields. How should we account for this discrepancy between brains and models?

One answer is that the discrepancy reflects the fact that brains and computer-vision systems operate on different hardware and under different constraints on space, time, and energy. Perhaps we have come to a point at which the two fields must go their separate ways. However, this answer is unsatisfying. Computational neuroscience must still find out how visual inference works in brains. And although engineers face *quantitatively* different constraints when building computer-vision systems, they, too, must care about the spatial, temporal, and energetic limitations their models must operate under when deployed in, for example, a smartphone. Moreover, as long as neural network models continue to dominate computer vision, more efficient hardware implementations are likely to be more similar to biological neural networks than current implementations using conventional processors and graphics processing units (GPUs).

A second explanation for the discrepancy is that the abundance of recurrent connections in cortex belies a superficial role in neural computation. Perhaps the core computations can be performed by a feedforward network [34], while recurrent processing serves more auxiliary and modulatory functions, such as divisive normalization [35] and attention [36–39]. This perspective is convenient because it enables us to hold on to the feedforward model in our minds. The auxiliary and modulatory functions let us acknowledge recurrence without fundamentally changing the way we envision the algorithm of recognition.

However, there is a third and more exciting explanation for the discrepancy between recurrent brains and feedforward models: although feedforward computation is powerful, a recurrent algorithm provides a fundamentally superior solution to the problem of visual inference, and this algorithm is implemented in primate visual cortex. This recurrent algorithm explains how primate vision can be so efficient in terms of space, time, energy, and data, while being so rich and robust in terms of the inferences and their generalization to novel environments.

In this review, we argue for the latter possibility, discussing a range of potential computational functions of recurrence and citing the evidence suggesting that the primate brain employs them. We aim to distinguish established from more speculative, and superficial from more profound forms of recurrence, so as to clarify the most exciting directions for future research that will close the gap between models and brains.

Unrolling a recurrent network

What exactly do we mean when we say that a neural network — whether biological or artificial — is recurrent rather than feedforward? This may seem obvious, but it turns out that the distinction can easily be blurred. Consider the simple network in [Figure 1a](#). It consists of three processing stages, arranged hierarchically, which we will refer to as *areas*, by analogy to cortex. Each area contains a number of neurons (real or artificial) that apply fixed operations to their input. Visual input enters in the first area, where it undergoes some transformation, the result of which is passed as input to the second area, and so forth. Information travels exclusively in one direction — the ‘forward’ direction, from input to output — and so this is an example of a feedforward architecture. Notably, the number of transformations between input and output is fixed, and equal to the number of areas in the network.

Now compare this to the architecture in [Figure 1b](#). Here, we have added lateral and feedback connections to the network. Lateral connections allow the output of an area to be fed back into the same area, to influence its computations in the next processing step. Feedback connections allow the output of an area to influence information processing in a lower area. There is some freedom in the order in which computations may occur in such a network. The order we illustrate here starts with a full feed-forward pass through the network. In subsequent time steps, neural activations are updated in ascending order through the hierarchy, based on the activations that were computed in the previous time step.

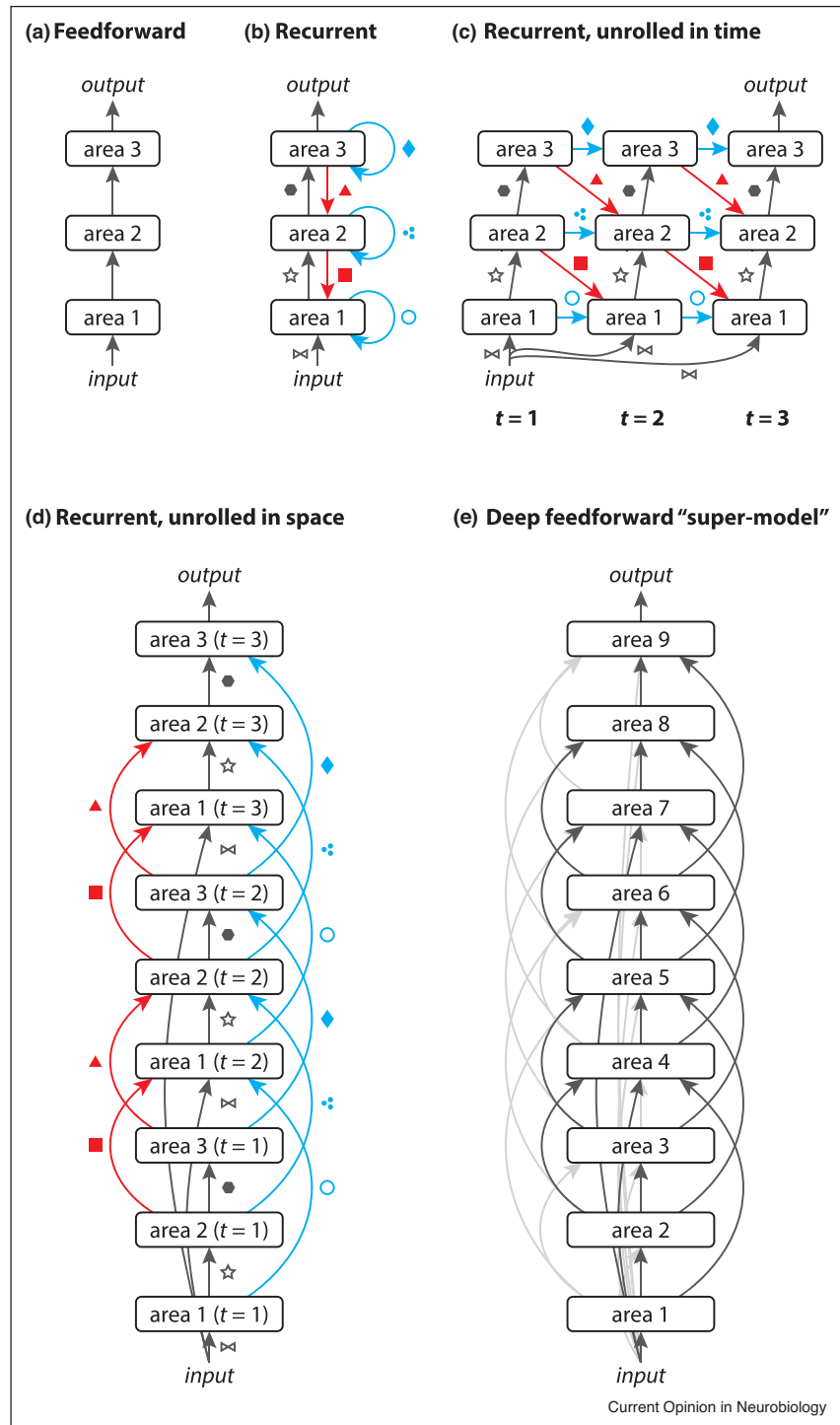
This order of operations can be seen more clearly if we ‘unroll’ the network in time, as shown in [Figure 1c](#). In this illustration, the network is unrolled for a fixed number of time steps (3). In fact, recurrent processing can be run for any desired duration before its output is read out — a

notion we will return to later. Notice how this temporally unrolled, small network resembles a larger feedforward neural network with more connections and areas between its input and output. We can emphasize this recurrent-feedforward equivalence by interpreting the computational graph over time as a spatial architecture, and visually arranging the induced areas and connections in a linear spatial sequence — an operation we call *unrolling in space* ([Figure 1d](#)). This results in a deep feedforward architecture with many *skip connections* between areas that are separated by more than one level in this new hierarchy, and with many connections that are exact copies of one another (sharing identical connection weights).

Thus, any finite-time RNN can be transformed into an equivalent FNN. But this should not be taken to mean that RNNs are a special case of FNNs. In fact, FNNs are a special case of finite-time RNNs ([Figure 2a](#)), comprising those which happen to have no cycles. More practically, not every unrolled finite-time RNN is a *realistic* FNN ([Figure 2b](#)). By realistic networks, we mean networks that conform to the real-world constraints the system must operate under. For computational neuroscience, a realistic network is one that fits in the brain of the animal and does not require a deeper network architecture or more processing steps than the animal can accommodate. For computer vision, a realistic network is one that can be trained and deployed on available hardware at the training and deployment stages. For example, there may be limits on the storage and energy available, which would limit the complexity of the architecture and computational graph. A realistic finite-time RNN, when unrolled, can yield an unworkably deep FNN. Although the most widely used current method for training RNNs (backpropagation through time) requires unrolling, an RNN is not equivalent to its unrolled FNN twin at the stage of real-world deployment: the RNN’s recurrent connections need not be physically duplicated, but can be reused across cycles of computation.

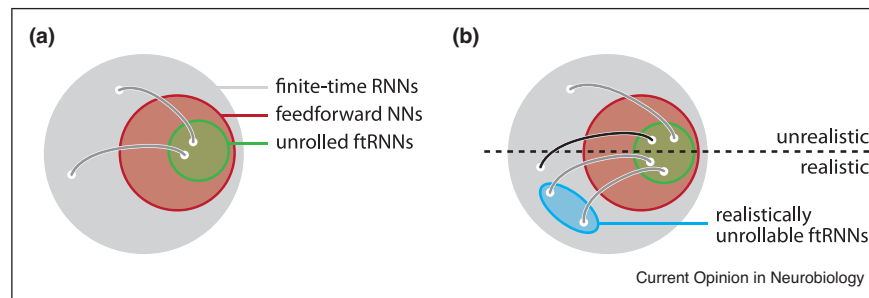
An important recent observation [[40](#)•, [41](#), [42](#)] is that the architecture that results from spatially unrolling a recurrent network, resembles the architectures of state-of-the-art FNNs used in computer vision, which similarly contain skip connections and can be very deep. These deep FNNs may form a super-class of models ([Figure 1e](#)), which reduce to (recurrent-equivalent) architectures when certain subsets of weights are constrained to be identical. Liao and Poggio [[40](#)•] showed that deep feedforward architectures known as *residual networks* (ResNets) [[19](#)] are formally equivalent to recurrent architectures when certain connection weights are constrained to be identical. Moreover, when ResNets were trained with such recurrent-equivalent weight-sharing constraints, their performance on computer vision benchmarks was similar to unconstrained ResNets (even though the weight sharing drastically reduces the parameter count and limits the component computations that the network can perform). This is especially

Figure 1



Unrolling recurrent neural networks. (a) A simple feedforward neural network. (b) The same network with lateral (blue) and feedback (red) connections added, to make it recurrent. (c) ‘Unrolling’ the network in time clarifies the order of its computations. Here, the network is unrolled for three time steps before its output is read out, but we could choose to run the network for more or fewer steps. Areas are staggered from left to right to show the order in which their neural activities are updated. (d) Alternatively, we can unroll the recurrent network’s time steps in space, by arranging the areas and connections from different time steps in a linear spatial sequence. Note how all arrows now once again point in the same (forward) direction, from input to output. Throughout panels (a-d), connections that are identical (sharing the same weight matrices) are indicated by corresponding symbols. (e) If we lift the weight-sharing constraints from the previous network, this induces a deep feedforward ‘super-model’, which can implement the spatially unrolled recurrent network as a special case. This more general architecture may include additional connections (examples shown as light gray arrows) not present in the spatially unrolled recurrent net.

Figure 2



Relationships between recurrent and feedforward networks. This figure illustrates relationships between discrete-time feedforward (FNN) and discrete-time recurrent (RNN) neural network models. (a) The architecture of any RNN can be reduced to an FNN by removing all its recurrent connections (e.g. going from Figure 1b back to Figure 1a), or equivalently, setting the weights of these connections to zero. Vice versa, any FNN can be expanded to an infinite variety of RNNs by adding lateral or feedback connections. Feedforward networks, thus, form an architectural subset of RNNs. Here we specifically consider RNNs that accomplish their task in a finite number of time steps. These finite-time RNNs (ftRNNs) have the special property that they can be unrolled into equivalent FNNs. White points linked by arcs indicate pairs of computationally equivalent architectures. Thus, the feedforward NNs contain a subset of architectures that can be obtained by unrolling a ftRNN. (b) These sets of networks can be further subdivided into subsets that are or are not *realistic* to implement with the computational resources available for a brain or engineered device (areas below and above the dotted line, respectively). Deeper networks and, more generally, networks with more neurons and connections tend to require more memory and computation to train and run. Some realistic ftRNNs remain realistic when expressed as an FNN (blue ellipse). Others, however, become too complex, when unrolled, to be feasible (black arc crossing the realism line). This is because the unrolling operation induces a much deeper architecture with many more neural connections to be stored. These not-realistically-unrollable ftRNNs are especially interesting, since they correspond to recurrent solutions that cannot be replaced by feedforward architectures.

noteworthy given that ResNets, and architecturally related DenseNets, are currently among the top-ranking FNNs on prominent computer vision benchmarks [19,43], as well as measures of brain-similarity [29]. Today's best artificial vision models, thus, actually implement computational graphs closely related to those of recurrent networks, even though these models are strictly feedforward architectures.

Continuous-time versus discrete-time dynamics

FNNs used in computer vision do not have meaningful dynamics. Each unit in the network instantaneously transforms its input into an output. This is in contrast to a feedforward network of biological neurons. When given a static input, biological neurons do not immediately produce their final responses. The movement of electric charges and neurotransmitters, and the opening and closing of ion channels takes time, so the network will gradually transition from its initial to its final state, with its trajectory continually perturbed by noise. Such continuous-time dynamics can be described by differential equations. When these cannot be solved analytically (as is typically the case), the dynamics can be simulated in discrete steps. In each step, the current state of each simulated neuron is updated. The future state of the network thus depends on its current state, as it does in an RNN. Consequently, the computational graph of the simulation algorithm contains loops from each neuron back to itself. Running the simulation over time amounts to unrolling this loopy computational graph, even though the network architecture did not contain loops.

Computational neuroscientists commonly study models of feedforward and recurrent neural networks with continuous-time dynamics [44]. Here our focus is on neural network models that are motivated by the goal to capture computations, rather than their precise neural implementation. The discrete-time behavior of such a model is not derived from a continuous-time description in differential equations. Moreover, the model is optimized in its discrete-time implementation. However, an implicit assumption in the field is that such models could be implemented in biological brains, and thus in continuous-time dynamical systems.

Reasons to recur

We have described how a recurrent network can be unrolled into a deep feedforward architecture. The resulting feedforward super-model offers greater computational flexibility, since weight-sharing constraints can be omitted and additional skip connections added to the network (Figure 1e). So what would be the benefit of restricting ourselves to recurrent architectures? We will first discuss the benefits of recurrence in terms of overarching principles, before considering more specific implementations of these principles.

Recurrence provides greater and more flexible computational depth

Recurrence enables arbitrary computational depth

One important advantage of recurrent algorithms is that they can be run for any desired length of time before their output is collected. We can define computational depth as the maximum path length (i.e. number of successive connections and nonlinear transformations) between

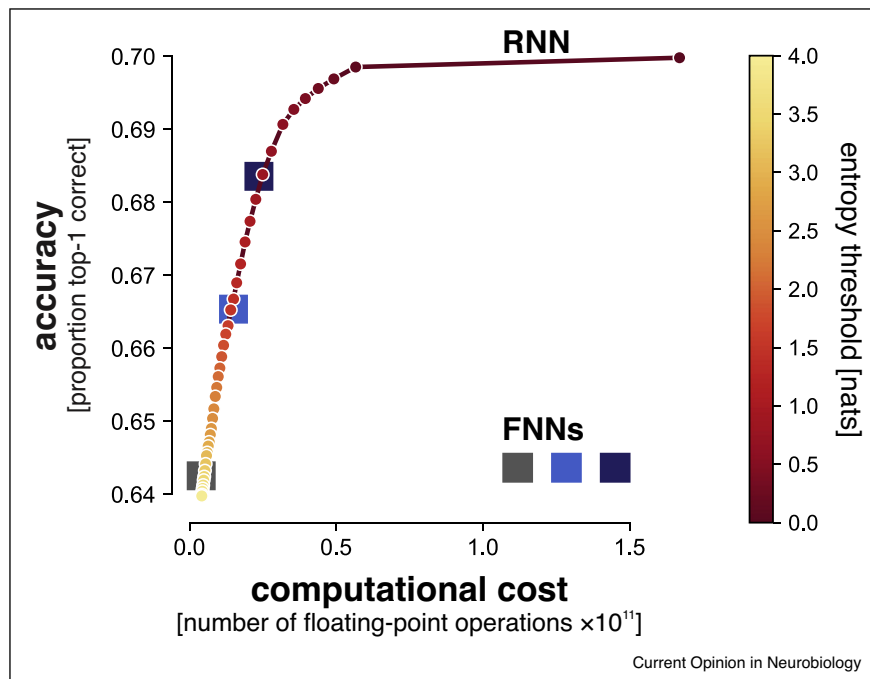
input and output. A recurrent neural network (RNN) can achieve arbitrary computational depth despite having a finite count of parameters and being limited to finite spatial components. In other words, it can multiply its limited spatial resources along time. These deeper computations can serve to expand on the number of hypotheses considered (in generative inference) or on the number of nonlinear features computed (in discriminative inference), or to extend the representation into the future or past, or to iteratively converge to a good estimate of some latent variable of interest.

Recurrence enables more flexible expenditure of energy and time in exchange for inferential accuracy

In addition to enabling an arbitrarily deep computation given enough time, an RNN can *adjust* its computational depth to the task at hand. The computational depth of a feedforward net, by contrast, is a fixed number determined by the architecture.

Spoerer *et al.* implemented a recurrent model that terminates computations when it reaches a confidence threshold (defined by the entropy of the posterior, a measure of the model's uncertainty) [23**]. The model terminates rapidly for many images, but expends more time and energy on hard images to reach its confidence threshold. Adjusting the confidence threshold enables trading off speed for accuracy in terms of average performance. When compared to a range of FNNs requiring different amounts of computation, the RNN achieved roughly the same accuracy as each of the FNNs when its confidence threshold was set to match the FNN's computational cost (number of floating point operations) on average across images (Figure 3). Flexible computational depth would be advantageous for animals, who may need to respond rapidly in some situations, must limit metabolic expenditures in general, and may benefit from slower and more energetically costly inferences when high accuracy is required. Computer vision faces similar requirements in certain applications. For example, a vision algorithm

Figure 3



Recurrence enables a network to trade speed for accuracy while approximately emulating the accuracies of feedforward models on average at matched computational cost. Circles denote the performance of a recurrent neural network (RNN) that was run for different numbers of time steps, until it achieved a desired threshold of image classification confidence (quantified by the entropy of the class probabilities in the final network layer). Squares correspond to three architecturally similar feedforward networks (FNN) with different computational costs. On the x-axis is the computational cost of running these models, measured by the number of floating point operations. For the feedforward models, this cost is fixed by the architecture. For the recurrent models, it is the average number of operations that was required to meet the given entropy threshold. The y-axis shows the classification accuracy achieved by each model. The performance of the recurrent model for different certainty thresholds follows a smooth curve, trading off computational cost (and thus computational speed) and accuracy. Note that this curve passes almost exactly through the cost-accuracy combinations achieved by the feedforward models. Thus, a single recurrent model can emulate the performance of multiple feedforward models as it trades off speed and accuracy. When the confidence threshold of termination was set such that the RNN matched the accuracy of a given FNN, the RNN required a similar number of floating-point operations on average as the FNN. (Figure adapted with permission from the authors [23**].)

in a smartphone should respond rapidly and conserve energy in general, but should also be able to recognize hard images, and it should allow trading off mean accuracy for speed and energy (e.g. when the battery is low).

Recurrent architectures can compress complex computations in limited hardware

Another benefit of recurrent solutions is that they require fewer components in space when physically implemented in recurrent circuits, such as brains. Compare Figure 1b and 1e: the recurrent network is anatomically more compact than the feedforward network and has fewer connections. It is easy to see why evolution might have favored a recurrent implementations for many brain functions: Space, neural projections, and the energy to develop and maintain them are all costly for the organism. In addition, synaptic efficacies must be either learned from limited experience or encoded in a limited-capacity genome. Beyond saving space, material, and energy, thus, smaller descriptive complexity (or parameter count) might ease development and learning.

Engineered devices face the same set of costs, although their relative weighting changes from application to application. In particular, a larger number of units and weights must either be represented in the memory of a conventional computer or implemented in specialized (e.g. neuromorphic) hardware. The connection weights in an NN model need to be learned from limited data. This requires extensive training, for example, in a supervised setting, with millions of hand-labeled examples that show the network the desired output for a given input. The larger number of parameters associated with a feedforward solution might overfit the training data. The learned parameters then do not generalize well to new examples of the same task.

FNNs often turn out to generalize surprisingly well even when they have very large numbers of parameters [45–47]. This phenomenon is thought to reflect a regularizing effect of the learning algorithm, stochastic gradient descent. Indeed, the trend is towards ever deeper networks with more connections to be optimized, and this trend is associated with continuing gains in performance on computer vision benchmarks [48]. Nevertheless, it could turn out that recurrent architectures that achieve high computational depth with fewer parameters bring benefits not only in terms of their storage, but also in terms of statistical efficiency, the ability generalize accurately based on limited experience. This would imply that recurrent networks have an inductive bias that makes up for the limited experiential data. This is explored further in subsequent sections, where we discuss how RNNs can exploit temporal dependency structures, and enable iterative inference.

Energy is another factor to consider in both biology and engineering. Larger FNNs take longer to train on bigger computing clusters, while drawing greater amounts of

power—a trend that is not sustainable. In the long run, therefore, computer vision too may benefit from the anatomical compression that can be achieved through clever use of recurrence.

Importantly, however, not every deep feedforward model can be compressed into an equivalent recurrent implementation. This anatomical compression can only be achieved when the same function may be applied iteratively or recursively within the network. The crucial question, therefore, is: what are these functions? What operations can be applied repeatedly in a productive manner? The remainder of this paper will reflect on the various roles that have been proposed for recurrent processing for visual inference, from superficial to increasingly profound forms of recurrence.

Feedback connections are required to integrate information from outside the visual hierarchy

A key, established role of recurrent connections in biological vision is to propagate information from outside the visual cortex, so that it can aid visual inference [49]. Here, we will briefly discuss two such outside influences: attention and expectations.

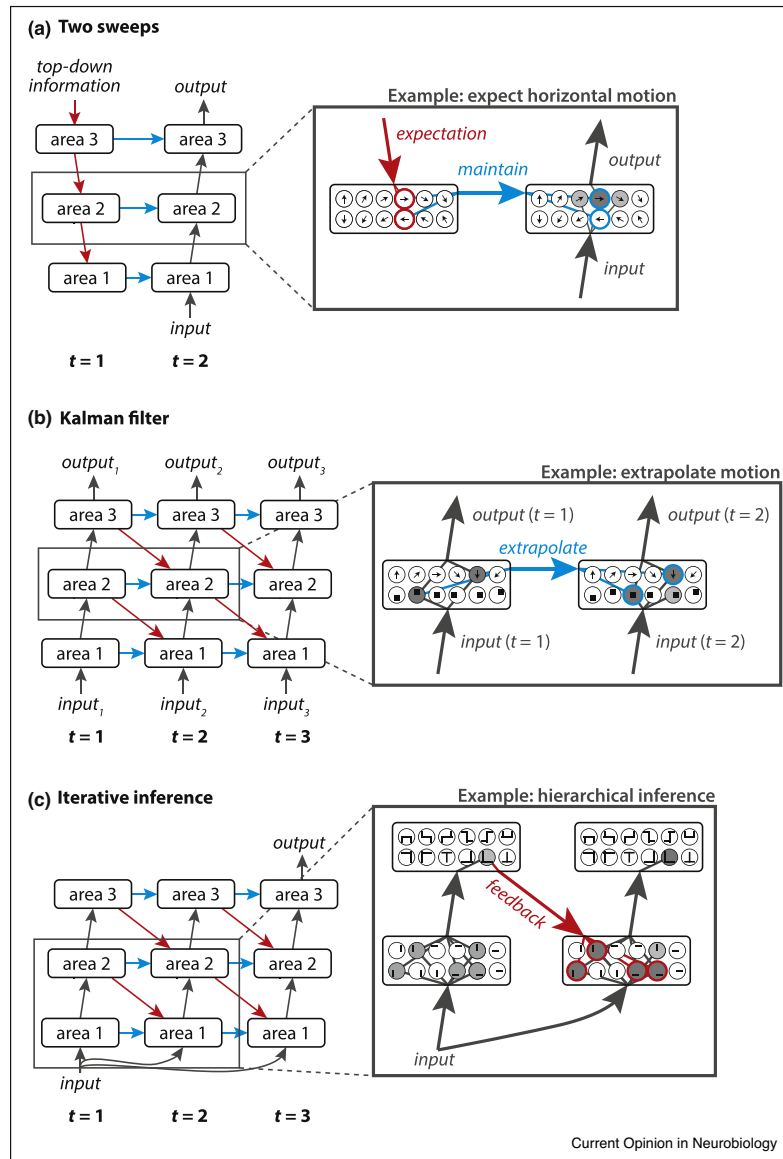
Attentional prioritization requires feedback connections

Animals have needs and goals that change from moment to moment. Perception is attuned to an animal's current objectives. For instance, a primate foraging for red berries may be more successful if its visual perception apparatus prioritizes or enhances the processing of red items. Since current goals are represented outside the visual cortex (e.g. in frontal regions), top-down connections are clearly required for this information to influence visual processing. Such top-down effects have been grouped under the label 'attention', and they have been the subject of an entire subfield of study. For our purposes, it is sufficient to note that the effects and mechanisms of top-down attention are well-documented and pervasive in visual cortex (for review, see [36–38]), and thus there is no question that this is one important function of recurrent connections.

Integrating prior expectations into visual inference requires feedback connections

Organisms may constrain their visual inferences by expectations [50]. Visual input can be ambiguous and unreliable, and thus open to multiple interpretations. To constrain the inference, an observer can make use of prior knowledge [51–53]. One form of prior knowledge is environmental constants (e.g. 'light tends to come from above' [54]). Such unvarying knowledge may be stored within visual cortex, especially when it pertains to the overall prevalence of basic visual features (e.g. local edge orientations [55]). Another form of prior knowledge is contextual information specific to the current situation. Such time-varying knowledge may require a flexible representation outside visual cortex (e.g. 'I rang the

Figure 4



Increasingly profound modes of recurrent processing, unrolled in time. Visual cortex likely combines all three modes of recurrence illustrated here. The left side of each panel shows the computational graph induced by each form of recurrence, while the right side illustrates a (simplified) example of how this recurrence can be used. In these examples, circles correspond to neurons (or neural assemblies) encoding the feature illustrated within the circle, and lines that connect to circles indicate neural connections with significant activity. **(a)** Top-down influences from outside the visual processing hierarchy may be incorporated through two computational sweeps: a feedback sweep priming the network with top-down information and a feedforward sweep to interpret visual input and combine this interpretation with the top-down signal. Note that the lateral connections here merely copy neural activities in each area to the next time point; this identity transformation could also be implemented in other ways, such as slow membrane time constants or other forms of local memory. In the example on the right, a top-down signal communicates the expectation that the upcoming input will be horizontal motion. This primes neurons encoding this direction of motion to be more easily or strongly activated, and sharpens the interpretation of the subsequent (ambiguous) visual input. **(b)** To efficiently perform inference on time-varying visual input, recurrent connections may implement a fixed temporal prediction function akin to the transition kernel in a Kalman filter, extrapolating the ongoing dynamics of the world one time step into the future. For instance, in the example on the right, a downward moving square was perceived at $t = 1$. This motion is predicted to continue, and this prediction constrains the interpretation of the (ambiguous) visual input at the next time point. For simplicity, only lateral recurrence is shown in this example. Note that each input is mapped onto its corresponding output in a single recurrent time step. **(c)** Static input may also benefit from recurrent processing that iteratively refines an initial, coarse feedforward interpretation. In this mode of recurrence, there are several processing time steps between input and output, whereas in **(b)** there was one input and output for each time step. Illustrated on the right is an iterative hierarchical inference algorithm. Here, a higher-level hypothesis, generated in the first time step, refines the underlying lower-level representation in the next time step, which in turn improves the higher-level hypothesis, and so forth, until the network converges to an optimal interpretation of the input across the entire hierarchy. For simplicity, lateral recurrent interactions are not shown in this example.

doorbell at my mother's house, so I expect to see her open the door'). Such expectations, represented in higher cortical regions, require feedback connections to affect processing in visual cortex [50].

The top-down imposition of attention and expectation must be mediated by feedback connections. However, it is unclear whether these influences fundamentally change the nature of visual representations or merely modulate these representations, adjusting the gain depending on the current relevance of different features of the visual input. As illustrated in Figure 4a, for a given input this would require only two 'sweeps' of computation through the visual processing hierarchy: a feedback sweep that primes visual areas with top-down information, and a bottom-up sweep to interpret the visual input and integrate or modify this interpretation with the top-down signal (not necessarily in that order). Importantly, if the feedback signal merely enhances or suppresses some visual features, then the core inference algorithm need not be fundamentally recurrent — one can imagine that the bottom-up part of such a network is modeled perfectly by an FNN, while an optional recurrent module could be added in order to implement top-down contextual influences.

Recurrent networks can exploit temporal dependency structure

Contextual constraints on visual inference include not only information from outside the visual hierarchy, such as information from other sensory modalities and memory, as discussed in the previous section. The recent stimulus history within the visual modality also provides context, likely represented within the visual system.

Recurrent networks can dynamically compress the stimulus history

The primate visual system is thought to contain a hierarchy, not only of processing stages and spatial scales, but also of temporal scales [56,57]. Visual representations track the environment moment by moment. However, the duration of a visual moment, the temporal grain, may depend on the level of representation. These principles apply to all sensory modalities and have been empirically explored, in particular, for audition and speech perception. At the simplest level, a neural network could use delay lines to detect spatiotemporal, rather than purely spatial, patterns. Recurrent neural networks have internal states and can represent temporal context across units tuned to different latencies. An RNN could represent a fixed temporal window, by replicating units tuned to different patterns for multiple latencies. However, RNNs trained on sequence processing tasks, such as language translation, learn more sophisticated representations of temporal context [58]. They can represent context at multiple time scales, learning a latent representation that enables them to dynamically compress whatever information from the past is needed for the task. In contrast to

a feedforward network, a recurrent network is not limited by spatial constraints in terms of its retrospective time horizon. It can maintain task-relevant information indefinitely, integrating long-term memory into its inferences.

Recurrent dynamics can simulate and predict the dynamics of the world

Dynamic compression of the past exploits the temporal dependency structure of the sensory data. The purpose of representing the past is to act well in the future. This suggests that a neural network should exploit temporal dependencies not just to compress the past, but also to predict the future. In fact, an optimal representation of even just the present requires prediction, because the sensory data is delayed and noisy.

Changes in the world are governed by laws of dynamics, which by definition are temporally invariant. An ideal observer will exploit these laws in visual inference and optimally combine previous with present observations to estimate the current state. This implies an extrapolation of the past to generate predictions that improve the interpretation of the present sensory input. When the dynamics are linear and noise is Gaussian, the optimal way to infer the present state by combining past and present evidence is the *Kalman filter* [59] — an algorithm widely used in engineering applications. A number of authors [60–63] have proposed that the visual cortex may implement an algorithm similar to a Kalman filter. This theory is consistent with temporal biases that are evident in human perceptual judgments [64–66].

Kalman filters employ a fixed temporal transitional kernel. This kernel takes a representation of the world (e.g. variables encoding the present state of a physical system, such as positions and velocities) at time t , and transforms it into a predicted representation for time $t + 1$, to be integrated with new sensory evidence that arrives at that time. While the resulting prediction varies as a function of the kernel's input, the kernel itself is constant, reflecting the temporal shift-invariance of the laws governing the dynamics. Recurrent neural networks provide a generalization of the Kalman filter and can represent nonlinear dynamical systems with non-Gaussian noise.

Note that this type of recurrent processing is more profound than the two-sweep algorithm (Figure 4a) that incorporated top-down influences on visual inference. The two-sweep algorithm is trivial to unroll into a feedforward architecture. In contrast, unrolling a Kalman filter-like recurrent algorithm would induce an infinitely deep feedforward network, with a separate set of areas and connections for each time point to be processed. A finite-depth feedforward architecture can only approximate the recurrent algorithm. While the feedforward approximation will have a finite temporal window of memory to constrain its present inferences, the recurrent

network can in principle integrate information over arbitrarily long periods.

Due to their advantages for dealing with time-varying (or otherwise ordered) inputs, recurrent neural networks are in fact widely employed in the broader field of machine learning for tasks involving sequential data. Speech recognition and machine translation are prominent applications that RNNs excel at [58,67–70]. Computer vision, too, has embraced RNNs for recognition and prediction of video input [71–73]. Note that these applications all exploit the dynamics in RNNs to model the dynamics in the data.

What if we trained a Kalman filter or sequence-to-sequence RNN (Figure 4b) on a train of independently sampled *static* inputs to be classified? The memory of the preceding inputs would not be useful then, so we expect the recurrent model to revert to using essentially only its feedforward weights. The type of recurrent processing we described in this section, thus uses memory to improve visual inference. In the next section, we consider how recurrent processing can help with the inferential computations themselves, even for static inputs.

Recurrence enables iterative inference

Recurrent processing can contribute even to inference on static inputs, and regardless of the agent's goals and expectations, by means of an *iterative* algorithm. An iterative algorithm is one that employs a computation that improves an initial guess. Applying the computation again to the improved guess yields a further improvement. This process can be repeated until a good solution has been achieved or until we run out of time or energy. Recurrent networks can implement iterative algorithms, with the same neural network functions applied successively to some internal pattern of activity (Figure 4c).

In many fields, iterative algorithms are used to solve estimation and optimization problems. In each iteration, a small adjustment is made to the problem's proposed solution, to improve a mathematically formulated objective. A locally optimal solution is found by making small improvements until further progress is not required or not possible. The algorithm navigates a path in the space of the values to be estimated or the parameters to be optimized, that leads to a good solution (albeit not necessarily the global optimum).

Much of machine learning involves iterative methods. Gradient descent is an iterative optimization method, whose stochastic variant is the most widely used method for training FNNs. Many discrete optimization techniques are iterative. Iterative algorithms are also central to inference in machine learning, for example in variational inference (where inference is achieved by optimization), sampling methods (where steps are chosen stochastically

such that the distribution of samples converges on the posterior distribution), and message passing algorithms (such as loopy belief propagation). In particular, such iterative inference algorithms are used in probabilistic approaches to computer vision [31,33]. It is somewhat surprising, then, that iterative computation is not widely exploited to perform visual inference in FNNs.

Visual inference is naturally understood as an optimization problem, where the goal is to find hypotheses that can explain the current visual input [51]. A hypothesis, in this case, is a proposed set of *latent* (i.e. unobserved) causes that can jointly explain the image. The hypothesized latent causes could be the identities and positions of objects in the scene. Visual hypotheses are hierarchical, being subdivided into smaller hypotheses about lower or intermediate-level features, such as the local edges that make up a larger contour. An iterative visual inference algorithm starts with an initial hypothesis, and refines it by incremental improvements. These improvements may include eliminating hypotheses that are mutually exclusive, strengthening compatible causes, or adjusting a hypothesis based on its ability to predict the data (the visual input). In a probabilistic framework, the optimization objective would be the likelihood (probability of the image given the latent representation) or the posterior probability (probability of the latent representation given the image).

Incompatible hypotheses can compete in the representation

There are often multiple plausible explanations for a given sensory input that are mutually exclusive. The distributed, parallel nature of neural networks enables them to initially activate and represent all of these possible hypotheses simultaneously. Recurrent connectivity between neurons can then implement competitive interactions among hypotheses, so as to converge on the best overall explanation.

There is some evidence that sensory representations are probabilistic [74–76] — in this case, the probabilities assigned to a set of mutually exclusive hypotheses must sum to 1. A strengthening of belief in one hypothesis, thus, should entail a reduction of the probability of other hypotheses in the representation. If neurons encode point estimates rather than probability distributions, then only one hypothesis can win (although that hypothesis may be encoded by a population response involving multiple neurons). The winning hypothesis could be the maximum a posteriori (MAP) hypothesis or the maximum likelihood hypothesis. Influential models of visual inference involving competitive recurrent interactions include *divisive normalization* [35], biased competition [36], and *predictive coding* [30,32,77].

Recent theoretical work has demonstrated that lateral competition can give rise to a robust neural code, and can explain certain puzzling neural response properties [77,78]. This theory considers a spiking neural network setting, in which different neurons encode highly

overlapping or even identical features in their input. This degeneracy means that the same signal can be encoded equally well by a range of different response patterns. When a particular neuron spikes, lateral inhibition ensures that other competing neurons do not encode the same part of the input again. Which neuron gets to do the encoding thus depends on which neuron fires first, because its membrane potential happened to be closest to a spiking threshold. This leads to trial-to-trial variability in neural responses that reflects subtle differences in initial conditions — conditions that may not be known to an experimenter, who may thus mistake this variability for random noise. This could explain the puzzling observation that individual neurons reliably reproduce the same output given the same electrical stimulation, but populations of neurons, wired together, display apparently random variability under sensory stimulation [79–81]. Since multiple neurons can encode the same feature, the resulting code is also robust to neurons being lost or temporarily inactivated.

FNNs do not incorporate lateral connections for competitive interactions, although they very often include computations that serve a similar purpose. Chief among these are operations known as *max-pooling* and *local response normalization* (LRN) [82,16]. In max-pooling, only the strongest response within a pool of competing neurons is forwarded to the next processing stage. In LRN, each neuron has its response divided by a term that is computed from the sum of activity in its normalization pool. While neither of these mechanisms is mediated by explicit lateral connections in a FNN, a strictly connectionist implementation of these mechanisms (e.g. in biological neurons or neuromorphic hardware) would have to include lateral recurrence. This, then, is another way in which apparently feedforward FNNs can exhibit a (limited) form of recurrent processing ‘under the hood’. Note, though, that each of these operations is carried out only once, rather than allowing competitive dynamics to converge over multiple iterations. Furthermore, in contrast to the lateral interactions in predictive coding or other normative models, LRN and max-pooling are not derived from normative principles, and do not necessarily select (or enhance) the best hypothesis (however ‘best’ is defined).

Compatible hypotheses can strengthen each other in the representation

In feedforward models of hierarchical visual inference, neurons at higher stages selectively respond to combinations of simpler features encoded by lower-level neurons. Higher-level neurons thus are sensitive to larger-scale patterns of correlation between subsets of lower-level features. But such larger-scale statistical regularities may not be most efficiently captured by a set of larger-scale building blocks. Instead, they may be more compactly captured by local association rules. Consider, for instance, the problem of contour detection. Many combinations of local edges in an image can form a continuous contour. The resulting space of contours may

be too complex to be efficiently represented with larger-scale templates. What all these contours have in common, however, is that they consist of pairs of edges that are locally contiguous, with sharper angles occurring with lower probability. Thus, the criteria for ‘contour-ness’ may be compactly expressed by a set of local association rules: *these edges go together; those do not* [83,84]. Contours may then be pieced together by repeatedly applying the same local association rules. Those edge pairs which are most clearly connected would be identified in early iterations. Later inferences can benefit from the context provided by earlier inferences, enabling the process to recognize continuity even where it is less locally apparent.

This insight has inspired network models of visual inference that implement local association rules through lateral connections, to aid contour integration and other perceptual grouping operations [85]. Recent examples include Linsley *et al.*, who developed *horizontal gated-recurrent units* (hGRUs) that learn local spatial dependencies [86**]. A network equipped with this particular recurrent connectivity was competitive with state-of-the-art feedforward models on a contour integration task, while using far fewer free parameters. George *et al.* [87] similarly leveraged lateral interactions to recognize contiguous contours and surfaces, by modeling these with a conditional random field (CRF), using a message-passing algorithm for inference. This approach enabled their Recursive Cortical Network (RCN) to reliably beat CAPTCHAs — images of letter sequences under a variety of distortions, noise and clutter, that are widely used to verify that queries to a user interface are made by a person, and not an algorithm. CRFs were also used by Zheng *et al.* [88], who incorporated them as a recurrent extension of a convolutional neural network for image segmentation. The model surpassed state-of-the-art performance at the time. Association rules enforced through lateral connections may also help to fill in missing information, such as when objects are partially hidden from view by occluders. Lateral connectivity has been shown to improve recognition performance in such settings [23**,89*,90]. Montobbio *et al.* showed that lateral diffusion of activity between neurons with correlated feedforward filter weights improves robustness to image perturbations including occlusions [90].

Enhancement of mutually compatible hypotheses (this section) and competition between mutually exclusive hypotheses (previous section) can both contribute to inference. A more general perspective is provided by the insight that prior knowledge about what features in a scene are mutually compatible or exclusive may be part of an overarching *generative model*, which iterative algorithms can exploit for inference.

Iterative algorithms can leverage generative models for inference

Perceptual inference aims to converge on a set of hypotheses that best explain the sensory data. Typically, a

hypothesis is considered to be a good explanation if it is consistent with both our prior knowledge and the sensory data. A generative model is a model of the joint distribution of latent causes and sensory data. Generative models can powerfully constrain perceptual inference because they capture prior knowledge about the world. In machine learning, defining generative models enables us to express and exploit what we know about the domain. A wide range of inference algorithms can be used to compute posterior distributions over variables of interest, given observed variables. The algorithms include variational inference, message passing, and Markov Chain Monte Carlo sampling, all of which require iterative computation.

In this section, we focus on a particular approach to leveraging generative models in visual inference, in which the joint distribution $p(x, z)$ of the image x and the latents z is factorized as $p(x, z) = p(z) \cdot p(x|z)$, which we refer to as the top-down factorization. The architecture contains components that model $p(x|z)$ and predict the image from the latents (or more generally lower-level latent representations from higher-level latent representations). Compared to the alternative factorization $p(x, z) = p(x) \cdot p(z|x)$, the top-down factorization has the potential advantage that the model operates in the causal direction, matching the causal process in the world that generated the image. The top-down model predicts what visual input is likely to result from a scene that has the hypothesized properties. This is somewhat similar to the graphics engine of a video game or image rendering software. This top-down model can be implemented via feedback connections that translate higher-level hypotheses in the network to representations at a lower level of abstraction.

Using generative models implemented with top-down predictions for inference is known as *analysis-by-synthesis* — an approach that has a long history in theories of perception [51,30,32]. Arguably, the goal of perceptual inference, by definition, is to reason back from effects (sensory data) to their causes (unobserved variables of interest), and thus invert the process that generated the effects. The crucial question, however, is whether the causal process is explicitly represented in the inference algorithm. The alternative, which can be achieved with feedforward inference, is to directly approximate the inverse, without ever making predictions in the causal direction. The success of the feedforward approach then depends on how well the inverse can be approximated by a fixed mapping of inputs to hypotheses. To iteratively invert the causal process, a neural network can evaluate the causal model for a current hypothesis and update the hypothesis in a beneficial direction. This process can then be repeated until convergence. This process of analysis by repeated synthesis may be preferable to directly approximating the inverse mapping if the causal process that generates the sensory data is easier to model than its inverse. In particular, the causal process may be more

compactly represented, more easily learned, more efficient to compute, and more generalizable beyond the training distribution than its inverse.

Another potential advantage of generative inference lies in robustness to variations in the input. While FNNs can accurately categorize images drawn from the same distribution that the training images were drawn from, it does not take much to fool them. A slight alteration imperceptible to humans can cause a FNN to misclassify an image entirely, with high confidence [91]. State-of-the-art FNNs rely more strongly on texture than humans, who rely more on shape [92]. More generally, FNNs seem to ignore many image features that are relevant to human perception [93]. One hypothesized reason for this is that these networks are trained to discriminate images, but not to generate them. Thus, any visual feature that reliably discriminates categories in the training data will be weighted heavily in the network's classification decisions. Importantly, this weight is unrelated to how much variance the feature explains in the image, and to the likelihood — that is, the probability of the image given either of the categories. An ideal observer should evaluate the likelihood for each hypothesis and adjudicate according to their ratio [94]. A feedforward network may instead latch on to a few highly discriminative, but subtle image features that don't explain much and may not generalize to images from a different data set [93,95]. In contrast, visual features that are important for generating or reconstructing images of a given class may be more likely to generalize to other examples of the same category. In support of this intuition, two novel RNN architectures that employ generative models for inference were found to be more robust to adversarial perturbations [96,97]. Generative inference networks were also shown to better align with human perception, compared to discriminative models, when presented with *controversial stimuli* — images synthesized to evoke strongly conflicting classifications from different models [98*].

Despite these promising developments, generative inference remains rare in visual FNN models. The exceptions mentioned above are rather simple networks trained on easy classifications problems, and are not (yet) competitive with state-of-the-art performance on more challenging computer vision benchmarks. Within computational neuroscience, by contrast, generative feedback connections appear in many network models of visual inference. Prominent examples are predictive coding [30,32] and hierarchical Bayesian inference [99]. However, these models have not had much success in explaining visual inference beyond its earliest stages. A notable exception is work by Wen *et al.* [100*], which shows that extending supervised convolutional FNNs with the recurrent dynamics of predictive coding can improve classification performance. The fields of computer vision and computational neuroscience both stand to benefit from the

development of more powerful generative inference models.

Iteration is necessary to close the amortization gap

Iterative inference has many advantages. A drawback of iteration, however, is that it takes time for the algorithm to converge during inference. This is unattractive for animals who need to perform visual inference under time pressure. It is also a challenge when training a FNN, which already requires many iterations of optimization. If each update of the network's connections additionally includes an iterative inner loop to perform inference on each training example, this lengthens the time required for training.

A complementary inference mechanism is *amortized inference* [101,102], where a feedforward model approximates the mapping from images to their latent causes. FNNs are eminently suited for learning complicated input–output mappings. A single transformation then replaces the trajectories that would be navigated by an iterative inference algorithm. In some cases, the iterative solution and the best amortized mapping may be exactly equivalent. A linear model, for instance, can be estimated iteratively, by performing gradient descent on the sum of squared prediction errors. However, if a unique solution exists, it can equivalently be found by a linear transformation that directly maps from the data to the optimal coefficients.

In general, however, amortized inference incurs some error, compared to the optimal solution that might be found through iterative optimization. This error has been called the *amortization gap* [103,104*]. It is analogous to the poor fit that may result from buying clothes ‘off the rack’, compared to a tailored version of the same garment. The amortization gap is defined in the context of variational inference, when the iterative optimization of the variational approximation to the posterior is replaced by a neural network that maps from the image to the parameters of the variational distribution. The resulting model suffers from two types of error: (1) error caused by the choice of the variational approximation (variational approximation gap) and (2) error caused by the model mapping from images to variational parameters (amortization gap). One recent study has argued that the amortization gap is often the main source of error in amortized inference models [103]. Amortized and iterative inference define a continuum. At one extreme, iterative inference until convergence reaches a solution through a trajectory of small improvements, explicitly evaluating the quality of the current solution at every iteration. At the other extreme, fully amortized inference takes a single leap from input to output. In between these extremes lies a space for algorithms that use intermediate numbers of steps, to approximate the optimal solution through a computational path that is more refined than a leap, but more efficient than full-fledged iterative optimization. Models that occupy this space include

explicit hybrids of iterative and amortized inference [104*,105,106], as well as RNNs with arbitrary dynamics that are trained to converge to a desired objective in a limited number of time steps (e.g. [23*,107,108**,109*]).

Recurrence is required for active vision

Vision is an active exploratory process. Our eye movements scan the scene through a sequence of well-chosen fixations that bring objects of interest into foveal vision. Moving our heads and our bodies enables us to bring entirely new parts of the scene into view, and closer for inspection at high resolution. Active control of our eyes, heads, and bodies can also help disambiguate 3D structure as fixation on points at different depths changes binocular disparity, and head and body movements create motion parallax. Active vision involves a recurrent cycle of sensory processing and muscle control, a cycle that runs through the environment.

Our focus here has been on the internal computational functions of recurrent processing, and active vision has been reviewed elsewhere [110–112]. However, it is important to note that the internal recurrent processes of visual inference from a single glimpse are embedded within the larger recurrent process of active visual exploration. Active vision provides not just the larger behavioral context of visual inference. It also provides a powerful illustration of the fundamental advantages that recurrent algorithms offer in general. It illustrates how limited resources (the fovea) can be dynamically allocated (eye movements) to different portions of the evidence (the visual scene) in temporal sequence. A sensory system limited to a finite number of neurons, thus, can multiply its resources along time to achieve a detailed analysis. The cycle may start with an initial rough analysis of the entire visual field, followed by fixations on locations likely to yield valuable information. This is an example of an essentially recurrent process whose efficiency cannot be emulated with a feedforward system. The internal mechanisms of visual inference are faced with qualitatively similar challenges: Just like our retinae cannot afford foveal resolution throughout the visual field, the ventral stream cannot afford to perform all potentially relevant inferences on the evidence streaming in through the optic nerve in a single feedforward sweep. Internal shifts of attention, like eye movements, can sequentialize a complex computation and avoid wasting energy on portions of the evidence that are uninformative or irrelevant to the current goals of the animal.

Whereas the outer loop of active vision is largely about positioning our eyes relative to the scene and bringing important content into foveal vision, the inner loop of visual inference on each glimpse is far more flexible. Beyond covert attentional shifts that select locations, features, or objects for scrutiny, a recurrent network can *decide what computations to perform* so as to most efficiently reduce uncertainty about the important parts of the scene. In a game of twenty questions, we choose a question that most

reduces our remaining uncertainty at each step. The budget of twenty would not suffice if we had to decide all the questions before seeing any answers. The visual system similarly has limited computational resources for processing a massive stream of evidence. It must choose what inferences to pursue on the basis of their computational cost and uncertainty-reducing benefit as it forages for insight [113–115].

Closing the gap between biological and artificial vision

We have reviewed a number of advantages that recurrence can bring to neural networks for visual inference. Going forward, neural network models of vision should incorporate recurrence; not just to better understand visual inference in the brain, but also to improve its implementation in machines.

Recurrence already improves performance on challenging visual tasks

Efforts in this direction are already underway, and turning up promising results. Some of this work has been described in previous sections, such as the use of lateral connections to impose local association rules [86**,87, 88] and generative inference for more robust performance outside the training distribution [96,97]. Several other recent findings are worth highlighting here, as they have shown improved performance on visual tasks, better approximations to biological vision, or both, through recurrent computations.

In particular, several studies have found that recurrence is required in order to explain or improve visual inference in challenging settings. Kar and colleagues [108**] identified a set of ‘challenge images’ that required recurrent processing in order to be accurately recognized. A feedforward FNN struggled to interpret these images, whereas macaque monkeys recognized them as accurately as a set of control images. Challenge images were associated with longer processing times in the macaque inferior temporal (IT) cortex, consistent with recurrent computations. Neural responses in IT for images that took longer were well accounted for by a brain-inspired RNN model. In a different study [116*], this same recurrent architecture was found to account for behavior, and neural data from macaque visual cortex, in object recognition tasks, while also achieving good performance on an important computer vision benchmark (ImageNet [117]). In human visual cortex, recurrent interactions were also found to be crucial to model the neural dynamics underlying object recognition, as measured through magnetoencephalography (MEG) [118].

One prominent challenge to visual inference is posed by partial occlusions, which hide part of a target object from view. In two recent studies, recurrent architectures were shown to be more robust to occlusions than their feedforward counterparts [89*,119*]. Interestingly, in both

human observers and in an RNN model, object recognition under occlusion was impaired by *backward masking* [119*] (the presentation of a meaningless noise image, shortly after a target stimulus, to disrupt recurrent processing [120,15,13]). Neural responses to partially occluded shapes in macaque visual cortex are also consistent with recurrent processing, and were well explained by a predictive coding model in which prefrontal cortex provide a feedback signal to visual area V4 [121,122].

Another challenge for human perception is *crowding*, which occurs when the detailed perception of a target stimulus is disrupted by nearby flanker stimuli [123]. In certain instances, the target stimulus can be released from crowding if further flankers are added that form a larger, coherent structure with the original flankers. This *uncrowding* effect may be due to the flankers being ‘explained away’, thus reducing their interference with the target representation [124,125]. Recent work [126] has shown that both effects can be explained by architectures known as *Capsule Nets* [127**,128], which include recurrent information routing mechanisms that may be similar to perceptual grouping and segmentation processes in the visual cortex.

Note that, in all of these cases, it may be possible to develop a feedforward architecture that performs the task equally well or better. Trivially, and as we discussed previously, a successful recurrent architecture can always be unrolled (for a finite number of time steps) into a deep feedforward network with many more learnable connections. However, a *realistic* recurrent model, when unrolled, may map onto an *unrealistic* feedforward model (Figure 2), where realism could refer to the real-world constraints faced by either biological or artificial visual systems. Future studies should compare RNN and FNN implementations for the same visual inference task, while matching the complexity of the models in a meaningful way. Setting a realistic budget of units, connections, and computational operations is one important approach. To understand the computational differences between RNN and FNN solutions, it is also interesting to (1) match the parameter count (number of connection weights that must be learned and stored), which requires granting the FNN larger feature kernels, more feature maps per layer, or more layers, or (2) match the computational graph, which equates the distribution of path lengths from input to output and all other statistics of the graph, but grants the FNN a much larger number of parameters [23**].

Freeing ourselves from the feedforward framework

Deep feedforward neural networks constitute an essential building block for visual inference, but they are not the whole story. The missing element, recurrent dynamics, is central to a range of alternative conceptions of visual inference that have been proposed [110,129,31,111,112,130]. These ideas have a long history, they are essential to

understanding biological vision, and they have great potential for engineering, especially in the context of modern hardware and software. The promise of active vision and recurrent visual inference is, in fact, boosted by the power of feedforward networks.

However, the beauty, power, and simplicity of feedforward neural networks also makes it difficult to engage and develop the space of recurrent neural network algorithms for vision. The feedforward framework, embellished by recurrent processes that serve auxiliary and modulatory functions like normalization and attention, enables computational neuroscientists to hold on to the idea of a hierarchy of feature detectors. This idea might not be entirely mistaken. However, it is likely to be severely incomplete and ultimately limiting.

The insight that any finite-time recurrent network can be unrolled compounds the problem by suggesting that the feedforward framework is essentially complete. More practically, the fact that we train RNNs by unrolling them for finite time steps might in some ways impede our progress. FNNs are usually trained by stochastic gradient descent using the *backpropagation* algorithm. This method retraces in reverse the computational steps that led to the response in the output layer, so as to estimate the influence that each connection in the network had on the response. Each connection weight is then adjusted, to bring the network output closer to a desired output. The deeper the network, the longer the computational path that needs to be retraced. RNNs for visual inference typically are trained through a variation on this method, known as *backpropagation through time* (BPTT) [131]. To retrace computations in reverse through cycles, the RNN is unrolled along time, so as to convert it into a feedforward network whose depth depends on the number of time steps as shown in Figure 1b–d. This enables the RNN to be trained like an FNN.

BPTT is attractive for enabling us to train RNNs like FNNs on arbitrary objectives. When it comes to learning recurrent dynamics, however, BPTT strictly optimizes the output at the specific time points evaluated by the objective (e.g. the output after exactly N steps). Outside of this time window, there is no guarantee that the network's response will be well-behaved. The RNN might reach the desired objective at the desired time, but diverge immediately after. Ideally, we would like a visual RNN presented with a stable image to converge to an attractor that represents the image and behave stably for arbitrary lengths of time. This would be consistent with iterative optimization, in which each step improves the network's approximation to its objective. While it is not impossible for BPTT to give rise to such dynamics, it does not specifically favor them.

From a theory perspective, BPTT is limiting because it shackles RNNs to the feedforward framework, in which the goal is still to map inputs to outputs, rather than to

discover useful dynamics. From a practical and implementational perspective, BPTT is computationally cumbersome, as every additional recurrent time step extends the computational path that must be retraced in order to update the connections. This complication also renders BPTT biologically implausible. Although the case for backpropagation as potentially biologically plausible has recently been strengthened [132–134], its extension *through time* is difficult to reconcile with biology [135] or implement efficiently in a finite engineered system for online learning — precisely because it requires unrolling and keeping track of separate copies of each weight as computational cycles are retraced in reverse.

Given these drawbacks, we speculate that a true breakthrough in recurrent vision models will require a training regime that does not rely on BPTT. Rather than optimizing an RNN's state in a finite time window, future RNN training methods might directly target the network's dynamics, or the states that those dynamics are encouraged to converge to. This approach has some history in RNN models of vision. Predictive coding models, for instance, are designed with dynamics that explicitly implement iterative optimization. Such models can update their connections through learning rules that require only the converged network state as input [30], rather than the entire computational path to this state. Marino *et al.* [104*] recently proposed *iterative amortized inference*, training inference networks to have recurrent dynamics that improve the network's hypotheses in each iteration, without constraining these dynamics to a particular form (such as predictive coding). More generally, RNNs whose dynamics converge to a steady state can be optimized through variations on an algorithm known as *recurrent backpropagation* [136–138], which avoids retracing the computational graph through time. However, it is often difficult to design RNNs such that their dynamics converge to a steady state (within the time window for which the model is trained), while maintaining *expressivity* (the ability of the model to learn a wide range of functions). This challenge is addressed by the recently developed *contractor recurrent backpropagation* method [139], which introduces a mathematical penalty that can be imposed while training any RNN, to encourage it to learn convergent dynamics.

Going forward, in circles

We started this review with the puzzling observation that, whereas biological vision is implemented in a profoundly recurrent neural architecture, the most successful neural network models of vision to date are feedforward. We have argued, theoretically and empirically, that vision models will eventually converge to their biological roots and implement more powerful recurrent solutions. This is an appealing prospect, as it suggests that neuroscientists and engineers can continue to work synergistically, to make progress on common challenges. After all, visual inference, and intelligence more generally, were solved

once before, and so discovering nature's solutions should go hand in hand with building artificial ones.

Conflict of interest statement

Nothing declared.

Acknowledgements

We thank Samuel Lippel, Heiko Schütt, Andrew Zaharia, Tal Golan and Benjamin Peters for detailed comments on a draft of this paper. This work was supported by a Rubicon grant from the Dutch Research Council (to RSvB).

References and recommended reading

Papers of particular interest, published within the period of review, have been highlighted as:

- of special interest
- of outstanding interest

1. Lamme VA, Roelfsema PR: **The distinct modes of vision offered by feedforward and recurrent processing.** *Trends Neurosci* 2000, **23**:571-579 [http://dx.doi.org/10.1016/S0166-2236\(00\)01657-X](http://dx.doi.org/10.1016/S0166-2236(00)01657-X).
2. Kreiman G, Serre T: **Beyond the feedforward sweep: feedback computations in the visual cortex.** *Ann N Y Acad Sci* 2020, **1464**:222-241 <http://dx.doi.org/10.1111/nyas.14320>.
3. Angelucci A, Bressloff PC: **Contribution of feedforward, lateral and feedback connections to the classical receptive field center and extra-classical receptive field surround of primate V1 neurons.** *Progress in Brain Research*, vol 154 2006:93-120 [http://dx.doi.org/10.1016/S0079-6123\(06\)54005-1](http://dx.doi.org/10.1016/S0079-6123(06)54005-1).
4. Anderson JC, Douglas RJ, Martin KAC, Nelson JC: **Synaptic output of physiologically identified spiny stellate neurons in cat visual cortex.** *J Comp Neurol* 1994, **341**:16-24 <http://dx.doi.org/10.1002/cne.903410103>.
5. Martin KA: **Microcircuits in visual cortex.** *Curr Opin Neurobiol* 2002, **12**:418-425 [http://dx.doi.org/10.1016/S0959-4388\(02\)00343-4](http://dx.doi.org/10.1016/S0959-4388(02)00343-4).
6. Douglas RJ, Martin KA: **Recurrent neuronal circuits in the neocortex.** *Curr Biol* 2007, **17**:496-500 <http://dx.doi.org/10.1016/j.cub.2007.04.024>.
7. Felleman DJ, Van Essen DC: **Distributed hierarchical processing in the primate cerebral cortex.** *Cereb Cortex* 1991, **1**:1-47 <http://dx.doi.org/10.1093/cercor/1.1.1>.
8. Salin PA, Bullier J: **Corticocortical connections in the visual system: structure and function.** *Physiol Rev* 1995, **75**:107-154 <http://dx.doi.org/10.1152/physrev.1995.75.1.107>.
9. Markov NT, Ercsey-Ravasz MM, Ribeiro Gomes AR, Lamy C, Magrou L, Vezoli J, Misery P, Falchier A, Quilodran R, Gariel MA, Sallet J, Gamanut R, Huissoud C, Clavagnier S, Giroud P, Sappey-Marinié D, Barone P, Dehay C, Toroczkai Z, Knoblauch K, Van Essen DC, Kennedy H: **A weighted and directed interareal connectivity matrix for macaque cerebral cortex.** *Cereb Cortex* 2014, **24**:17-36 <http://dx.doi.org/10.1093/cercor/bhs270>.
10. Douglas RJ, Koch C, Mahowald M, Martin KA, Suarez HH: **Recurrent excitation in neocortical circuits.** *Science* 1995, **269**:981-985 <http://dx.doi.org/10.1126/science.7638624>.
11. Supér H, Spekreijse H, Lamme VA: **Two distinct modes of sensory processing observed in monkey primary visual cortex (V1).** *Nat Neurosci* 2001, **4**:304-310 <http://dx.doi.org/10.1038/85170>.
12. Di Lollo V, Enns JT, Rensink RA: **Competition for consciousness among visual events: the psychophysics of reentrant visual processes.** *J Exp Psychol: Gen* 2000, **129**:481-507 <http://dx.doi.org/10.1037/0096-3445.129.4.481>.
13. Lamme VA, Zipser K, Spekreijse H: **Masking interrupts figure-ground signals in V1.** *J Vision* 2001, **1**:1044-1053 <http://dx.doi.org/10.1167/1.3.32>.
14. Heinen K, Jolij J, Lamme VA: **Figure-ground segregation requires two distinct periods of activity in V1: a transcranial magnetic stimulation study.** *NeuroReport* 2005, **16**:1483-1487 <http://dx.doi.org/10.1097/01.wnr.0000175611.26485.c8>.
15. Fahrenfort JJ, Scholte HS, Lamme VA: **Masking disrupts reentrant processing in human visual cortex.** *J Cogn Neurosci* 2007, **19**:1488-1497 <http://dx.doi.org/10.1162/jocn.2007.19.9.1488>.
16. Lecun Y, Bengio Y, Hinton G: **Deep learning.** *Nature* 2015, **521**:436-444 <http://dx.doi.org/10.1038/nature14539>.
17. Schmidhuber J: **Deep learning in neural networks: an overview.** *Neural Netw* 2015, **61**:85-117 <http://dx.doi.org/10.1016/j.neunet.2014.09.003>.
18. He K, Zhang X, Ren S, Sun J: **Delving deep into rectifiers: surpassing human-level performance on ImageNet classification.** *2015 IEEE International Conference on Computer Vision (ICCV)*, vol. 2015 International 2015:1026-1034 <http://dx.doi.org/10.1109/ICCV.2015.123>.
19. He K, Zhang X, Ren S, Sun J: **Deep residual learning for image recognition.** *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2016-December 2016:770-778* <http://dx.doi.org/10.1109/CVPR.2016.90>.
20. Kemelmacher-Shlizerman I, Seitz SM, Miller D, Brossard E: **The MegaFace benchmark: 1 million faces for recognition at scale.** *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2016-December 2016:4873-4882* <http://dx.doi.org/10.1109/CVPR.2016.527>.
21. Kubilius J, Bracci S, Op de Beeck HP: **Deep neural networks as a computational model for human shape sensitivity.** *PLOS Comput Biol* 2016, **12**:e1004896 <http://dx.doi.org/10.1371/journal.pcbi.1004896>.
22. Majaj NJ, Pelli DG: **Deep learning-Using machine learning to study biological vision.** *J Vision* 2018, **18**:1-13 <http://dx.doi.org/10.1167/18.13.2>.
23. Spoerer CJ, Kietzmann TC, Mehrer J, Charest I, Kriegeskorte N: **Recurrent neural networks can explain flexible trading of speed and accuracy in biological vision.** *PLOS Comput Biol* 2020, **16**:e1008215 <http://dx.doi.org/10.1371/journal.pcbi.1008215>.
- Showed that RNNs can flexibly prioritize speed or accuracy, without having to sacrifice either when compared to FNNs with the same computational cost.
24. Cadieu CF, Hong H, Yamins DLK, Pinto N, Ardila D, Solomon EA, Majaj NJ, DiCarlo JJ: **Deep neural networks rival the representation of primate IT cortex for core visual object recognition.** *PLoS Comput Biol* 2014, **10**:e1003963 <http://dx.doi.org/10.1371/journal.pcbi.1003963>.
25. Khaligh-Razavi SM, Kriegeskorte N: **Deep supervised, but not unsupervised, models may explain IT cortical representation.** *PLoS Comput Biol* 2014, **10** <http://dx.doi.org/10.1371/journal.pcbi.1003915>.
26. Guclu U, van Gerven MAJ: **Deep neural networks reveal a gradient in the complexity of neural representations across the ventral stream.** *J Neurosci* 2015, **35**:10005-10014 <http://dx.doi.org/10.1523/JNEUROSCI.5023-14.2015>.
27. Kriegeskorte N: **Deep neural networks: a new framework for modeling biological vision and brain information processing.** *Annu Rev Vision Sci* 2015, **1**:417-446 <http://dx.doi.org/10.1146/annurev-vision-082114-035447>.
28. Kheradpisheh SR, Ghodrati M, Ganjtabesh M, Masquelier T: **Deep networks can resemble human feed-forward vision in invariant object recognition.** *Sci Rep* 2016, **6**:32672 <http://dx.doi.org/10.1038/srep32672>.
29. Schrimpf M, Kubilius J, Hong H, Majaj NJ, Rajalingham R, Issa EB, Kar K, Bashivan P, Prescott-Roy J, Geiger F, Schmidt K, Yamins DLK, DiCarlo JJ: **Brain-score: Which artificial neural network for object recognition is most brain-like?** *bioRxiv* 2020 <http://dx.doi.org/10.1101/407007>.

30. Rao RPN, Ballard DH: **Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects.** *Nat Neurosci* 1999, **2**:79-87 <http://dx.doi.org/10.1038/4580>.
31. Yuille A, Kersten D: **Vision as Bayesian inference: analysis by synthesis?** *Trends Cogn Sci* 2006, **10**:301-308 <http://dx.doi.org/10.1016/j.tics.2006.05.002>.
32. Friston K, Kiebel S: **Predictive coding under the free-energy principle.** *Philos Trans R Soc B: Biol Sci* 2009, **364**:1211-1221 <http://dx.doi.org/10.1098/rstb.2008.0300>.
33. Prince SJD: *Computer Vision: Models, Learning and Inference.* Cambridge: Cambridge University Press; 2012 <http://dx.doi.org/10.1017/CBO9780511996504>.
34. DiCarlo JJ, Zoccolan D, Rust NC: **How does the brain solve visual object recognition?** *Neuron* 2012, **73**:415-434 <http://dx.doi.org/10.1016/j.neuron.2012.01.010>.
35. Carandini M, Heeger DJ: **Normalization as a canonical neural computation.** *Nat Rev Neurosci* 2012, **13**:51-62 <http://dx.doi.org/10.1038/nrn3136>.
36. Desimone R, Duncan J: **Neural mechanisms of selective visual attention.** *Annu Rev Neurosci* 1995, **18**:193-222 <http://dx.doi.org/10.1146/annurev.neuro.18.1.193>.
37. Kastner S, Ungerleider LG: **Mechanisms of visual attention in the human cortex.** *Annu Rev Neurosci* 2000, **23**:315-341 <http://dx.doi.org/10.1146/annurev.neuro.23.1.315>.
38. Maunsell JH, Treue S: **Feature-based attention in visual cortex.** *Trends Neurosci* 2006, **29**:317-322 <http://dx.doi.org/10.1016/j.tics.2006.04.001>.
39. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Lu, Polosukhin I: **Attention is all you need.** *Advances in Neural Information Processing Systems* 2017:5998-6008.
40. Liao Q, Poggio T: *Bridging the Gaps Between Residual Learning, Recurrent Neural Networks and Visual Cortex.* 2016. arXiv:1604.03640
- This paper highlighted the important equivalence between ResNet architectures and unrolled RNNs, and showed that ResNets could be made more parameter-efficient (while retaining similar performance) by imposing recurrent-equivalent weight-sharing constraints.
41. Jastrzebski S, Arpit D, Ballas N, Verma V, Che T, Bengio Y: *Residual Connections Encourage Iterative Inference.* 2017. arXiv:1710.04773.
42. Greff K, Srivastava RK, Schmidhuber J: **Highway and residual networks learn unrolled iterative estimation.** *5th International Conference on Learning Representations, ICLR 2017 – Conference Track Proceedings (2015)* 2016:1-14 arXiv:1612.07771.
43. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ: **Densely connected convolutional networks.** *Proceedings – 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017 2017-January* 2017:2261-2269 <http://dx.doi.org/10.1109/CVPR.2017.243>.
44. Dayan P, Abbott LF: *Theoretical Neuroscience.* Cambridge, MA: MIT Press; 2001.
45. Advani MS, Saxe AM: *High-dimensional Dynamics of Generalization Error in Neural Networks.* 2017. arXiv:1710.03667.
46. Belkin M, Hsu D, Ma S, Mandal S: **Reconciling modern machine-learning practice and the classical bias-variance trade-off.** *Proc Natl Acad Sci U S A* 2019, **116**:15849-15854 <http://dx.doi.org/10.1073/pnas.1903070116>.
47. Nakkiran P, Kaplun G, Bansal Y, Yang T, Barak B, Sutskever I: *Deep Double Descent: Where Bigger Models and More Data Hurt.* 2019. arXiv:1912.02292.
48. Rawat W, Wang Z: **Deep convolutional neural networks for image classification: a comprehensive review.** *Neural Comput* 2017, **29**:2352-2449 http://dx.doi.org/10.1162/neco_a_00990.
49. Gilbert CD, Li W: **Top-down influences on visual processing.** *Nat Rev Neurosci* 2013, **14**:350-363 <http://dx.doi.org/10.1038/nrn3476>.
50. Summerfield C, Egnér T: **Expectation (and attention) in visual cognition.** *Trends Cogn Sci* 2009, **13**:403-409 <http://dx.doi.org/10.1016/j.tics.2009.06.003>.
51. von Helmholtz H: *Handbuch der physiologischen Optik.* New York: Dover (English translation); 1860/1962.
52. Weiss Y, Simoncelli EP, Adelson EH: **Motion illusions as optimal percepts.** *Nat Neurosci* 2002, **5**:598-604 <http://dx.doi.org/10.1038/nn858>.
53. Stocker AA, Simoncelli EP: **Noise characteristics and prior expectations in human visual speed perception.** *Nat Neurosci* 2006, **9**:578-585 <http://dx.doi.org/10.1038/nn1669>.
54. Mamassian P, Goutcher R: **Prior knowledge on the illumination position.** *Cognition* 2001, **81**:1-9 [http://dx.doi.org/10.1016/S0010-0277\(01\)00116-0](http://dx.doi.org/10.1016/S0010-0277(01)00116-0).
55. Girshick AR, Landy MS, Simoncelli EP: **Cardinal rules: visual orientation perception reflects knowledge of environmental statistics.** *Nat Neurosci* 2011, **14**:926-932 <http://dx.doi.org/10.1038/nn.2831>.
56. Hasson U, Yang E, Vallines I, Heeger DJ, Rubin N: **A hierarchy of temporal receptive windows in human cortex.** *J Neurosci* 2008, **28**:2539-2550 <http://dx.doi.org/10.1523/JNEUROSCI.5487-07.2008>.
57. Murray JD, Bernacchia A, Freedman DJ, Romo R, Wallis JD, Cai X, Padoa-Schioppa C, Pasternak T, Seo H, Lee D, Wang X-J: **A hierarchy of intrinsic timescales across primate cortex.** *Nat Neurosci* 2014, **17**:1661-1663 <http://dx.doi.org/10.1038/nn.3862>.
58. Sutskever I, Vinyals O, Le QV: **Sequence to sequence learning with neural networks.** *Adv Neural Inform Process Syst* 2014, **4**:3104-3112.
59. Kalman RE: **A new approach to linear filtering and prediction problems.** *J Basic Eng* 1960, **82**:35-45 <http://dx.doi.org/10.1115/1.3662552>.
60. Wolpert D, Ghahramani Z, Jordan M: **An internal model for sensorimotor integration.** *Science* 1995, **269**:1880-1882 <http://dx.doi.org/10.1126/science.7569931>.
61. Rao RPN, Ballard DH: **Dynamic model of visual recognition predicts neural response properties in the visual cortex.** *Neural Comput* 1997, **9**:721-763 <http://dx.doi.org/10.1162/neco.1997.9.4.721>.
62. Rao RPN: **Bayesian computation in recurrent neural circuits.** *Neural Comput* 2004, **16**:1-38.
63. Denève S, Duhamel J-R, Pouget A: **Optimal sensorimotor integration in recurrent cortical networks: a neural implementation of kalman filters.** *J Neurosci* 2007, **27**:5744-5756 <http://dx.doi.org/10.1523/JNEUROSCI.3985-06.2007>.
64. Orban de Vivry J-J, Coppe S, Blohm G, Lefevre P: **Kalman filtering naturally accounts for visually guided and predictive smooth pursuit dynamics.** *J Neurosci* 2013, **33**:17301-17313 <http://dx.doi.org/10.1523/JNEUROSCI.2321-13.2013>.
65. Kwon O-S, Tadin D, Knill DC: **Unifying account of visual motion and position perception.** *Proc Natl Acad Sci U S A* 2015, **112**:8142-8147 <http://dx.doi.org/10.1073/pnas.1500361112>.
66. van Bergen RS, Jehee JFM: **Probabilistic representation in human visual cortex reflects uncertainty in serial decisions.** *J Neurosci Off J Soc Neurosci* 2019, **39**:8164-8176 <http://dx.doi.org/10.1523/JNEUROSCI.3212-18.2019>.
67. Graves A, Mohamed A-R, Hinton G: **Speech recognition with deep recurrent neural networks.** *2013 IEEE International Conference on Acoustics, Speech and Signal Processing, no. 3* 2013:6645-6649 <http://dx.doi.org/10.1109/ICASSP.2013.6638947>.
68. Sak H, Senior A, Beaufays F: *Long Short-term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition.* 2014. arXiv:1402.1128.
69. Bahdanau D, Cho K, Bengio Y: **Neural machine translation by jointly learning to align and translate.** *3rd International Conference on Learning Representations, ICLR 2015 – Conference Track Proceedings* 2014 arXiv:1409.0473.

70. Cho K, van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y: **Learning Phrase Representations Using Rnn Encoder-decoder for Statistical Machine Translation**. 2014. arXiv:1406.1078.
71. Ranzato M, Sziám A, Bruna J, Mathieu M, Collobert R, Chopra S: **Video (language) modeling: a Baseline for Generative Models of Natural Videos**. 2014. arXiv:1412.6604.
72. Srivastava N, Mansimov E, Salakhutdinov R: **Unsupervised Learning of Video Representations Using LSTMs**. 2015. arXiv:1502.04681.
73. Lotter W, Kreiman G, Cox D: **Deep Predictive Coding Networks for Video Prediction and Unsupervised Learning**. 2016. arXiv:1605.08104.
74. Pouget A, Beck J, Ma WJ, Latham P: **Probabilistic brains: knowns and unknowns**. *Nat Neurosci* 2013, **16**:1170-1178 <http://dx.doi.org/10.1038/nn.3495>.
75. Ma WJ, Jazayeri M: **Neural coding of uncertainty and probability**. *Annu Rev Neurosci* 2014, **37**:205-220 <http://dx.doi.org/10.1146/annurev-neuro-071013-014017>.
76. Orbán G, Berkes P, Fiser J, Lengyel M: **Neural variability and sampling-based probabilistic representations in the visual cortex**. *Neuron* 2016, **92**:530-543 <http://dx.doi.org/10.1016/j.neuron.2016.09.038>.
77. Boerlin M, Machens CK, Denève S: **Predictive coding of dynamical variables in balanced spiking networks**. *PLoS Comput Biol* 2013, **9** <http://dx.doi.org/10.1371/journal.pcbi.1003258>.
78. Barrett DG, Denève S, Machens CK: **Optimal compensation for neuron loss**. *eLife* 2016, **5**:1-36 <http://dx.doi.org/10.7554/eLife.12454>.
79. Schiller PH, Finlay BL, Volman SF: **Short-term response variability of monkey striate neurons**. *Brain Res* 1976, **105**:347-349.
80. Dean A: **The variability of discharge of simple cells in the cat striate cortex**. *Exp Brain Res* 1981, **44** <http://dx.doi.org/10.1007/BF00238837>.
81. Mainen ZF, Sejnowski TJ: **Reliability of spike timing in neocortical neurons**. *Science* 1995, **268**:1503-1506.
82. Krizhevsky A, Sutskever I, Hinton GE: **ImageNet classification with deep convolutional neural networks**. *Advances in Neural Information Processing Systems* 2012 arXiv:1102.0183.
83. Field DJ, Hayes A, Hess RF: **Contour integration by the human visual system: evidence for a local "association field"**. *Vision Res* 1993, **33**:173-193 [http://dx.doi.org/10.1016/0042-6989\(93\)90156-q](http://dx.doi.org/10.1016/0042-6989(93)90156-q).
84. Geisler WS, Perry JS, Super BJ, Gallogly DP: **Edge co-occurrence in natural images predicts contour grouping performance**. *Vision Res* 2001, **41**:711-724 [http://dx.doi.org/10.1016/S0042-6989\(00\)00277-7](http://dx.doi.org/10.1016/S0042-6989(00)00277-7).
85. Roelfsema PR: **Cortical algorithms for perceptual grouping**. *Annu Rev Neurosci* 2006, **29**:203-227 <http://dx.doi.org/10.1146/annurev.neuro.29.051605.112939>.
86. Linsley D, Kim J, Veerabadran V, Windolf C, Serre T: **Learning long-range spatial dependencies with horizontal gated recurrent units**. *Advances in Neural Information Processing Systems* 2018:152-164
- Developed a new type of lateral recurrent connectivity to build RNNs that solve long-range path integration tasks far more efficiently than feedforward architectures.
87. George D, Lehrach W, Kansky K, Lázaro-Gredilla M, Laan C, Marthi B, Lou X, Meng Z, Liu Y, Wang H, Lavin A, Phoenix DS: **A generative vision model that trains with high data efficiency and breaks text-based CAPTCHAs**. *Science* 2017, **358** <http://dx.doi.org/10.1126/science.aag2612>.
88. Zheng S, Jayasumana S, Romera-Paredes B, Vineet V, Su Z, Du D, Huang C, Torr PH: **Conditional random fields as recurrent neural networks**. *Proceedings of the IEEE International Conference on Computer Vision* 2015 nation 2015:1529-1537 <http://dx.doi.org/10.1109/ICCV.2015.179>.
89. Spoerer CJ, McClure P, Kriegeskorte N: **Recurrent convolutional neural networks: a better model of biological object recognition**. *Front Psychol* 2017, **8**:1-14 <http://dx.doi.org/10.3389/fpsyg.2017.01551>
- Showed that lateral and feedback connections improve object recognition under clutter and occlusions in convolutional neural networks, compared to FNNs matched in complexity.
90. Montobbio N, Bonnasse-Gahot L, Citti G, Sarti A: **Kercnns: Biologically Inspired Lateral Connections for Classification of Corrupted Images**. 2019. arXiv:1910.08336.
91. Szegedy C, Zaremba W, Sutskever I, Bruna J, Erhan D, Goodfellow I, Fergus R: **Intriguing properties of neural networks**. *2nd International Conference on Learning Representations, ICLR 2014* 2014 arXiv:1312.6199.
92. Geirhos R, Michaelis C, Wichmann FA, Rubisch P, Bethge M, Brendel W: **Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness**. *7th International Conference on Learning Representations, ICLR 2019* 2019 arXiv:1811.12231.
93. Jacobsen JH, Behrmann J, Zemel R, Bethge M: **Excessive invariance causes adversarial vulnerability**. *7th International Conference on Learning Representations, ICLR 2019* 2019 arXiv:1811.00401.
94. Neyman J, Pearson ES: **IX. On the problem of the most efficient tests of statistical hypotheses**. *Philos Trans R Soc Lond Ser A, Containing Papers of a Mathematical or Physical Character* 1933, **231**:289-337 <http://dx.doi.org/10.1098/rsta.1933.0009>.
95. Ilyas A, Santurkar S, Tsipras D, Engstrom L, Tran B, Madry A: **Adversarial Examples Are Not Bugs, They Are Features**. 2019. arXiv:1905.02175.
96. Li Y, Bradshaw J, Sharma Y: **Are generative classifiers more robust to adversarial attacks?** *36th International Conference on Machine Learning, ICML 2019* 2019:6754-6783 arXiv:1802.06552.
97. Schott L, Rauber J, Bethge M, Brendel W: **Towards the first adversarially robust neural network model on MNIST**. *7th International Conference on Learning Representations, ICLR 2019* 2019 arXiv:1805.09190.
98. Golan T, Raju PC, Kriegeskorte N: **Controversial Stimuli: Pitting Neural Networks Against Each Other as Models of Human Recognition**. 2019. arXiv:1911.09288
- Showed that generative inference networks, compared to discriminative models, better aligned with human perception, when probed with stimuli designed to maximally distinguish between models.
99. Lee TS, Mumford D: **Hierarchical Bayesian inference in the visual cortex**. *J Opt Soc Am A Opt Image Sci Vision* 2003, **20**:1434-1448.
100. Wen H, Han K, Shi J, Zhang Y, Culurciello E, Liu Z: **Deep Predictive Coding Network for Object Recognition**. 2018. arXiv:1802.04762
- Demonstrated better object recognition performance when generic FNNs were augmented with the recurrent connectivity and dynamics of predictive coding.
101. Srikumar V, Kundu G, Roth D: **On amortizing inference cost for structured prediction**. *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning* 2012:1114-1124.
102. Stuhlmüller A, Taylor J, Goodman N: **Learning stochastic inverses**. *Advances in Neural Information Processing Systems* 2013:3048-3056.
103. Cremer C, Li X, Duvenaud D: **Inference suboptimality in variational autoencoders**. *35th International Conference on Machine Learning, ICML 2018* 2018:1749-1760 arXiv:1801.03558.
104. Marino J, Yue Y, Mandt S: **Iterative amortized inference**. *35th International Conference on Machine Learning, ICML 2018* 2018, **8**:5444-5462 arXiv:1807.09356
- Proposed a method to amortize iterative optimization dynamics in variational inference networks, to close the amortization gap.
105. Hjelm RD, Cho K, Chung J, Salakhutdinov R, Calhoun V, Jovic N: **Iterative refinement of the approximate posterior for directed belief networks**. *Advances in Neural Information Processing Systems (NIPS 2016)* 2016:4698-4706 arXiv:1511.06382.

106. Krishnan RG, Liang D, Hoffman MD: **On the challenges of learning with inference networks on sparse, high-dimensional data.** *International Conference on Artificial Intelligence and Statistics, AISTATS 2018* 84 2018:143-151 arXiv:1710.06085.
107. Liang M, Hu X: **Recurrent convolutional neural network for object recognition.** *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 07-12-June 2015*:3367-3375 <http://dx.doi.org/10.1109/CVPR.2015.7298958>.
108. Kar K, Kubilius J, Schmidt K, Issa EB, DiCarlo JJ: **Evidence that recurrent circuits are critical to the ventral stream's execution of core object recognition behavior.** *Nat Neurosci* 2019, **22**:974-983 <http://dx.doi.org/10.1038/s41593-019-0392-5>
Identified a set of 'challenge images' that, in both neural network models and primate visual cortex, required recurrent processing to be recognized as accurately as a set of control images.
109. Nayebi A, Bear D, Kubilius J, Kar K, Ganguli S, Sussillo D, DiCarlo JJ, Yamins DL: **Task-driven convolutional recurrent models of the visual system.** *Advances in Neural Information Processing Systems 2018*:5290-5301
Explored a large space of recurrent connectivity motifs and compared their effects on object recognition performance.
110. Ballard DH: **Animate vision.** *Artif Intell* 1991, **48**:57-86 [http://dx.doi.org/10.1016/0004-3702\(91\)90080-4](http://dx.doi.org/10.1016/0004-3702(91)90080-4).
111. Findlay JM, Gilchrist ID: *Active Vision.* Oxford University Press; 2003 <http://dx.doi.org/10.1093/acprof:oso/9780198524793.001.0001>.
112. Bajcsy R, Aloimonos Y, Tsotsos JK: **Revisiting active perception.** *Autonom Robots* 2018, **42**:177-196 <http://dx.doi.org/10.1007/s10514-017-9615-3>.
113. Russell SJ: **Rationality and intelligence.** *Artif Intell* 1997, **94**:57-77 [http://dx.doi.org/10.1016/S0004-3702\(97\)00026-X](http://dx.doi.org/10.1016/S0004-3702(97)00026-X).
114. Gershman SJ, Horvitz EJ, Tenenbaum JB: **Computational rationality: a converging paradigm for intelligence in brains, minds, and machines.** *Science* 2015, **349**:273-278 <http://dx.doi.org/10.1126/science.aac6076>.
115. Griffiths TL, Lieder F, Goodman ND: **Rational use of cognitive resources: levels of analysis between the computational and the algorithmic.** *Topics Cogn Sci* 2015, **7**:217-229 <http://dx.doi.org/10.1111/tops.12142>.
116. Kubilius J, Schrimpf M, Kar K, Rajalingham R, Hong H, Majaj N, Issa E, Bashivan P, Prescott-Roy J, Schmidt K, Nayebi A, Bear D, Yamins DL, DiCarlo JJ: **Brain-like object recognition with high-performing shallow recurrent ans.** *Advances in Neural Information Processing Systems 2019*:12805-12816
Developed a compact, brain-inspired RNN architecture that provides a good model of neural responses and behavior, while also scoring well on a prominent object recognition task.
117. Deng J, Dong W, Socher R, Li L-J, Li Kai, Fei-Fei Li: **ImageNet: a large-scale hierarchical image database.** *2009 IEEE Conference on Computer Vision and Pattern Recognition*; doi:10.1109/CVPR.2009.5206848: IEEE; 2009:248-255.
118. Kietzmann TC, Spoerer CJ, Sørensen LKA, Cichy RM, Hauk O, Kriegeskorte N: **Recurrence is required to capture the representational dynamics of the human visual system.** *Proc Natl Acad Sci U S A* 2019, **116**:201905544 <http://dx.doi.org/10.1073/pnas.1905544116>.
119. Tang H, Schrimpf M, Lotter W, Moerman C, Paredes A, Caro JO, Hardesty W, Cox D, Kreiman G: **Recurrent computations for visual pattern completion.** *Proc Natl Acad Sci U S A* 2018, **115**:8835-8840 <http://dx.doi.org/10.1073/pnas.1719397115>
Demonstrated robust object recognition under partial occlusions in recurrent, but not feedforward architectures. Like human observers, the RNNs were sensitive to backward masking.
120. Enns JT, Di Lollo V: **What's new in visual masking?** *Trends Cogn Sci* 2000, **4**:345-352 [http://dx.doi.org/10.1016/S1364-6613\(00\)01520-5](http://dx.doi.org/10.1016/S1364-6613(00)01520-5).
121. Fyall AM, El-Shamayleh Y, Choi H, Shea-Brown E, Pasupathy A: **Dynamic representation of partially occluded objects in primate prefrontal and visual cortex.** *eLife* 2017, **6**:1-25 <http://dx.doi.org/10.7554/eLife.25784>.
122. Choi H, Pasupathy A, Shea-Brown E: **Predictive coding in area V4: dynamic shape discrimination under partial occlusion.** *Neural Comput* 2018, **30**:1209-1257 http://dx.doi.org/10.1162/neco_a_01072.
123. Levi DM: **Crowding-An essential bottleneck for object recognition: a mini-review.** *Vision Res* 2008, **48**:635-654 <http://dx.doi.org/10.1016/j.visres.2007.12.009>.
124. Manassi M, Sayim B, Herzog MH: **Grouping, pooling, and when bigger is better in visual crowding.** *J Vision* 2012, **12**:13-13, doi:10.1167/12.10.13.
125. Manassi M, Lonchampt S, Clarke A, Herzog MH: **What crowding can tell us about object representations.** *J Vision* 2016, **16**:35 <http://dx.doi.org/10.1167/16.3.35>.
126. Doerig A, Bornet A, Choung O, Herzog M: **Crowding reveals fundamental differences in local vs. global processing in humans and machines.** *Vision Res* 2020, **167**:39-45 <http://dx.doi.org/10.1016/j.visres.2019.12.006>.
127. Sabour S, Frosst N, Hinton GE: **Dynamic routing between capsules.** *Advances in Neural Information Processing Systems 2017*:3856-3866
Introduced the influential Capsule Networks, which implement iterative recurrent information routing dynamics for visual segmentation and grouping.
128. Sabour S, Frosst N, Hinton GE: **Matrix capsules with EM routing.** *ICLR 2018, 2018* arXiv:1710.09829.
129. O'Regan JK, Noë A: **A sensorimotor account of vision and visual consciousness.** *Behav Brain Sci* 2001, **24**:939-973 <http://dx.doi.org/10.1017/S0140525X01000115>.
130. Buzsáki G: *The Brain from Inside Out.* Oxford University Press; 2019 <http://dx.doi.org/10.1093/oso/9780190905385.001.0001>.
131. Werbos P: **Backpropagation through time: what it does and how to do it.** *Proc IEEE* 1990, **78**:1550-1560 <http://dx.doi.org/10.1109/5.58337>.
132. Guerguiev J, Lillicrap TP, Richards BA: **Towards deep learning with segregated dendrites.** *eLife* 2017, **6**:1-37 <http://dx.doi.org/10.7554/eLife.22901>.
133. Sacramento J, Ponte Costa R, Bengio Y, Senn W: **Dendritic cortical microcircuits approximate the backpropagation algorithm.** *Advances in Neural Information Processing Systems 2018*:8721-8732.
134. Whittington JC, Bogacz R: **Theories of error back-propagation in the brain.** *Trends Cogn Sci* 2019, **23**:235-250 <http://dx.doi.org/10.1016/j.tics.2018.12.005>.
135. Lillicrap TP, Santoro A: **Backpropagation through time and the brain.** *Curr Opin Neurobiol* 2019, **55**:82-89 <http://dx.doi.org/10.1016/j.conb.2019.01.011>.
136. Almeida L: **A learning rule for asynchronous perceptrons with feedback in a combinatorial environment.** *Proceedings, 1st First International Conference on Neural Networks 1987*, **2**:609-618.
137. Pineda FJ: **Generalization of back-propagation to recurrent neural networks.** *Phys Rev Lett* 1987, **59**:2229-2232 <http://dx.doi.org/10.1103/PhysRevLett.59.2229>.
138. Liao R, Xiong Y, Fetaya E, Zhang L, Yoon KJ, Pitkow X, Urtasun R, Zemel R: **Revising and improving recurrent back-propagation.** *35th International Conference on Machine Learning, ICML 2018* 2018:4807-4820. arXiv:1803.06396.
139. Linsley D, Ashok AK, Govindarajan LN, Liu R, Serre T: *Stable and Expressive Recurrent Vision Models.* 2020. arXiv:2005.11362.