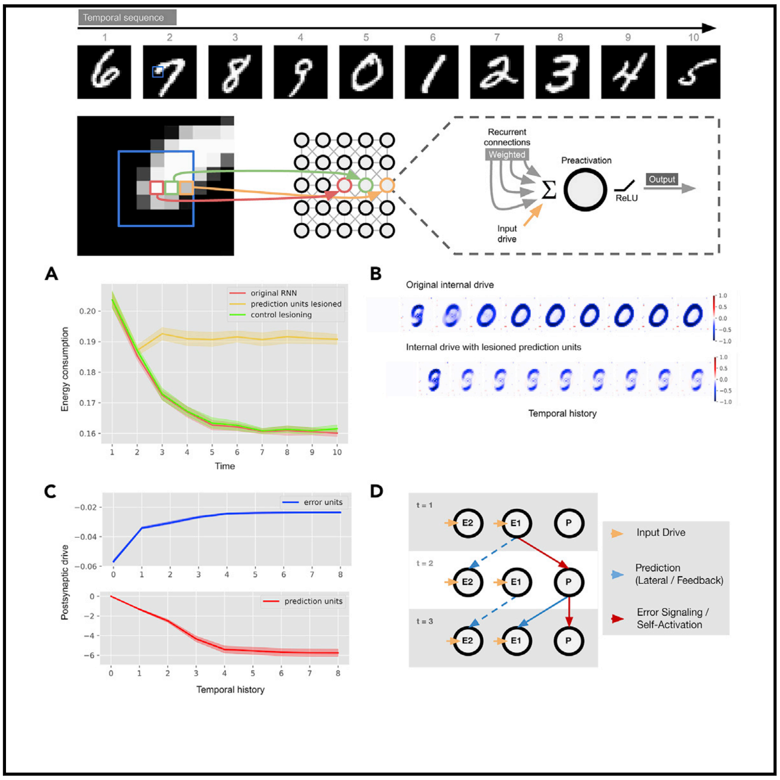
Article

Patterns

Predictive coding is a consequence of energy efficiency in recurrent neural networks

Graphical abstract



Authors

Abdullahi Ali, Nasir Ahmad,
Elgar de Groot,
Marcel Antonius Johannes van Gerven,
Tim Christian Kietzmann

Correspondence

abdullahi.ali@donders.ru.nl (A.A.),
tim.kietzmann@uni-osnabrueck.de (T.C.K.)

In brief

Connecting the computational principles of brain function to the physical constraints of the brain is important for understanding neural information processing. This is demonstrated by showing that the hallmarks of the predictive coding framework can emerge in a recurrent neural network model that is constrained in its energy usage.

Highlights

- Neural networks are optimized for energy efficiency in predictable environments
- Trained networks exhibit hallmarks of predictive coding as an emergent phenomenon
- Energy-efficient networks separate into subpopulations of prediction and error units
- Lesion studies indicate two types of prediction: fast and slow

Ali et al., 2022, Patterns *3*, 100639
December 9, 2022 © 2022 The Author(s).
https://doi.org/10.1016/j.patter.2022.100639

Patterns

CellPress
OPEN ACCESS

Article

Predictive coding is a consequence of energy efficiency in recurrent neural networks

Abdullahi Ali,[1,4,*] Nasir Ahmad,[1] Elgar de Groot,[1,3] Marcel Antonius Johannes van Gerven,[1] and Tim Christian Kietzmann[2,*]

[1]Donders Institute for Brain, Cognition and Behaviour, Radboud University, Nijmegen, the Netherlands
[2]Institute of Cognitive Science, University of Osnabrück, Osnabrück, Germany
[3]Department of Experimental Psychology, Utrecht University, Utrecht, the Netherlands
[4]Lead contact
*Correspondence: abdullahi.ali@donders.ru.nl (A.A.), tim.kietzmann@uni-osnabrueck.de (T.C.K.)

https://doi.org/10.1016/j.patter.2022.100639

THE BIGGER PICTURE In brain science and beyond, predictive coding has emerged as a ubiquitous framework for understanding sensory processing. It postulates that the brain continuously inhibits predictable sensory input, sparing computational resources for input that promises high information gain. Using artificial neural network models, we here ask whether hallmarks of predictive coding can arise from other, perhaps simpler principles. We report that predictive coding naturally emerges as a simple consequence of energy efficiency; networks trained to be efficient not only predict and inhibit upcoming sensory input but spontaneously separate into distinct populations of "error" and "prediction" units. Our results raise the intriguing question which other core computational principles of brain function may be understood as a result of physical constraints posed by the biological substrate and highlight the usefulness of bio-inspired, machine learning-powered neuroscience research.

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

SUMMARY

Predictive coding is a promising framework for understanding brain function. It postulates that the brain continuously inhibits predictable sensory input, ensuring preferential processing of surprising elements. A central aspect of this view is its hierarchical connectivity, involving recurrent message passing between excitatory bottom-up signals and inhibitory top-down feedback. Here we use computational modeling to demonstrate that such architectural hardwiring is not necessary. Rather, predictive coding is shown to emerge as a consequence of energy efficiency. When training recurrent neural networks to minimize their energy consumption while operating in predictive environments, the networks self-organize into prediction and error units with appropriate inhibitory and excitatory interconnections and learn to inhibit predictable sensory input. Moving beyond the view of purely top-down-driven predictions, we demonstrate, via virtual lesioning experiments, that networks perform predictions on two timescales: fast lateral predictions among sensory units and slower prediction cycles that integrate evidence over time.

INTRODUCTION

In computational neuroscience and beyond, predictive coding has emerged as a prominent theory for how the brain encodes and processes sensory information.[1–4] It postulates that higher-level brain areas continuously keep track of the causes of lower-level sensory input and actively inhibit incoming sensory signals that match expectation. Over time, the brain's higher-level representations are shaped to yield increasingly accurate predictions that, in turn, minimize the surprise the system encounters in its inputs. That is, the brain creates an increasingly accurate model of the external world and focuses on processing unexpected sensory events that yield the highest information gain.

Adding to the increasing, albeit often indirect, experimental evidence for predictive coding in the brain,[5–17] computational modeling has investigated explicit implementations of predictive coding, indicating that they can reproduce experimentally observed neural phenomena.[3,18–20] For example, Rao and Ballard[18] demonstrated that a hierarchical network with top-down inhibitory feedback can explain extra-classical field effects in primary visual cortex (V1). In addition, deep neural network architectures wired to implement predictive coding have been shown to work at scale in real-world tasks,[21–25] adding more support for the computational benefits of recurrent message passing.[26–30] The common rationale of predictive coding-focused modeling work is to test its computational and representational effects by hardwiring the model circuitry to mirror hierarchical and inhibitory connectivity between units that drive predictions and units that signal deviations from said predictions (also called sensory or error units).

Contrasting this approach, here we ask a different question: can computational mechanisms of predictive coding naturally emerge from other, perhaps simpler computational principles? In search for such principles, we take inspiration from the organism's limited energy budget and previous established links between prediction and energy efficiency.[31–36] Expanding previous work on efficient coding and sparse coding,[37–43] we subject recurrent neural networks (rate-based recurrent neural networks [RNNs]) to predictable sequences of visual input and optimize their synaptic weights to minimize the largest sources of energy consumption in the mammalian cortex: action potential generation and synaptic transmission.[44–49] We then test the resulting networks for phenomena typically associated with predictive coding. We compare the energy budget of the networks with two baseline models trained with more conventional efficient coding objectives, search for inhibitory connectivity profiles that mirror sensory predictions, and explore whether the networks automatically separate their neural resources into prediction and error units.

RESULTS

To better understand whether, and under which conditions, predictive coding may emerge in energy-constrained neural networks, we first trained a set of RNNs on predictable streams of visual input. For this, we created sequences of handwritten digits taken from the MNIST dataset,[50] iterating through 10 digits in numerical (category) order. Starting from a random digit, sequences wrapped around from nine to zero. The image sequences shown were deterministic in the sequence of digit categories and therefore (increasingly) predictable when the position in the sequence was extracted by the network. Such predictability of sensory input from the environment is a central assumption of the predictive coding framework because prediction is the primary mechanism through which the framework reduces uncertainty. A temporal autocorrelation of sensory signals is evident in the real world and the consequential assumption of related coding principles, such as slowness[51] and temporal stability.[52] Although this work considers the simpler case of predictable sequences of categories, we expect that temporal autocorrelation in natural video data, which is similarly predictable, will yield very similar results. The visual input to each network unit was stimulus dependent, with each unit receiving an input that mirrored its corresponding pixel intensity (i.e., 784 network units corresponding to 784 input pixels from MNIST). These visual inputs were not weighted with modifiable parameters, preventing the models from learning to ignore the input as a shortcut to a solution for energy efficiency. Crucially, because we are using an RNN, the next image in a sequence can be predicted. In other words, we trained the neural network not just to encode a set of images but the particular sequence that was conserved over training. We hypothesized that the models would learn to use their recurrent connectivity to counteract the input drive where possible and, therefore, to minimize their energy consumption. Energy costs were defined to arise from synaptic transmission and action potentials, mirroring the two most dominant sources of energy consumption in the mammalian cortex.[44–49] Figure 1 shows an overview of the experimental pipeline. More details are provided under ''Data generation'' and ''RNN architecture and training procedure.''

Preactivation as a proxy for the energy demands of the brain

Sparse coding is often implemented with an L1 sparsity constraint alongside a reconstruction objective on the unit outputs; i.e., networks are optimized to reduce the overall unit activity post non-linearity by having as few units active as possible while reconstructing the input as closely as possible.[39] Here we take a different approach by making energy minimization the target objective instead. However, given that synaptic transmission contributes considerable energy cost alongside action potentials, a modified learning objective is required to approximate total energy consumption; for example, minimization of unit outputs alone could be implemented via unreasonably strong inhibitory connections, which themselves might increase the overall energy budget. To solve this issue, we propose to instead train energy-efficient RNNs by minimizing absolute (L1) unit preactivation. In real neuronal networks, one can regard the preactivation as presynaptic input; in other words, the summed afferent input from all other neurons. This has two desired properties: first, it drives unit output toward zero, mirroring minimization of unit firing rates (dependent naturally on the form of activation function used). Second, we show that this objective also leads to minimal synaptic weight magnitudes in cases of noise in the system, mirroring an overall reduction in synaptic transmission (see the supplemental text and Figure S5 in the supplemental information for a theoretical derivation and visualization of why this is the case). We consider this link between a biologically realistic notion of energy efficiency and synaptic weight magnitude an important first result of this paper. Optimizing for minimal preactivation, we trained 10 network instances with different random weight initializations.[53] To empirically assess whether minimizing preactivation leads to better energy consumption outcomes, we need to establish that minimizing preactivation indeed minimizes activity and synaptic transmission and show that it does this better than other, more standard efficient coding objectives. With this in mind, we devised two candidate baseline models: a more conventional model, where the unit outputs (post non-linearity) are minimized, and a model where the unit outputs and weights are minimized. Inclusion of these candidates allows us to explore how overemphasis on minimizing unit outputs can affect synaptic transmission and how to penalize the weights to correct this overemphasis. The results

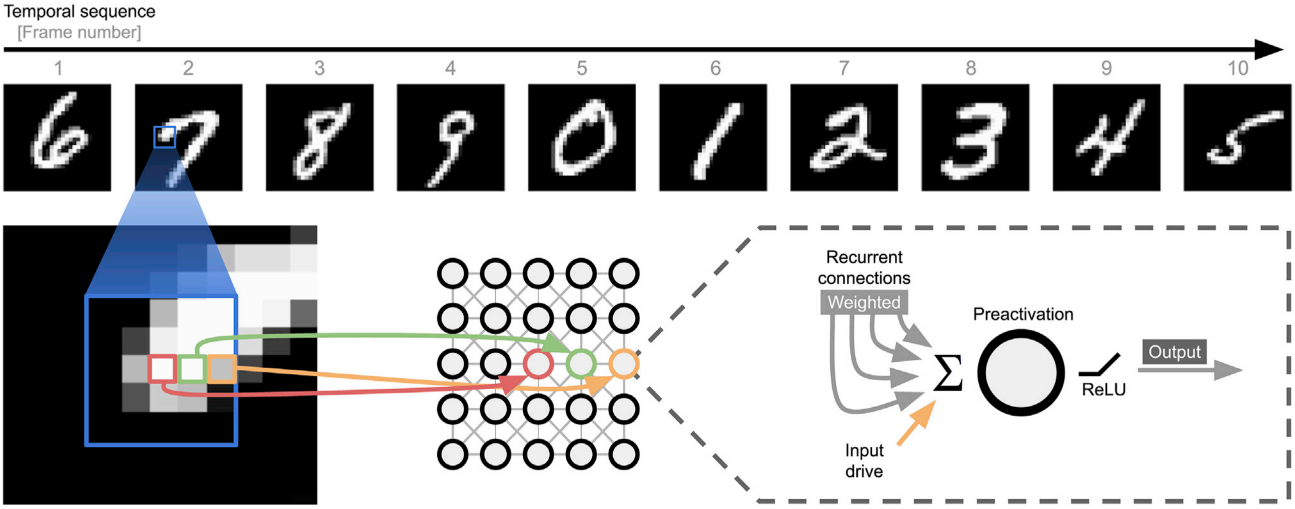


Figure 1. Overview of the model architecture and input pipeline

The top part of the figure shows an example input sequence fed to a recurrent network. In each run, 10 randomly sampled MNIST digits are presented in ascending order to the network (with wraparound after digit 9). Each input pixel drives exactly one unit of the RNN. The preactivation of each RNN unit is computed from the recurrent input, provided by other networks units, as well as the input drive determined by the intensity of the pixel to which it is connected. The output of the network units is determined by taking the rectified linear unit (ReLU) activation function of the preactivations. Only the recurrent connections in the network are learned.

of this comparison can be found in Figure 2A. As postulated, the preactivation models outperform the models that minimize unit output alone.

If we break down the network's energy consumption into contributions from activities and synaptic transmissions, we see that this divergence is a result of the model minimizing unit output at the cost of high synaptic transmission. Although the preactivation model has higher network activity, it performs much better in terms of synaptic transmission and, hence, in terms of overall energy efficiency. This is due to the fact that the preactivation model penalizes excessively large (negative) weights because they would massively increase synaptic transmission even when that achieves low unit outputs. What would happen if we added a penalty on the weight magnitudes? We tested networks with L2 penalties on the weights and found that these networks can be competitive in terms of energy demands. However, they achieve this through broad inhibition in a similar fashion to the networks trained on the outputs alone (see Figure S1 for a more in-depth breakdown).

The results in Figure 2A are based on a point estimate of energy demands in an animal brain model; i.e., biological assumptions about the relative energy cost of action potentials and synaptic transmission. Our estimate is in line with biological estimates for the mammalian cortex, which commonly estimate the total synaptic transmission to induce higher energy costs than action potentials.[44–49] The results do not depend on the particular weighting of action potentials and synaptic transmission and can be changed to increase the impact of action potentials (from zero to 10-fold) on the system without significantly changing the results. In short, minimizing unit preactivation consistently outperformed the minimization of unit output (L1 on the unit activity) in terms of total energy budget.

RNNs trained for energy efficiency learn to predict and inhibit upcoming sensory signals

Having established that the trained RNNs successfully reduce their energy consumption over time, we confirmed targeted inhibition of predicted sensory input, a hallmark of predictive coding. This was accomplished by separately visualizing the input drive (i.e., the unit activation driven by sensory input) and the network drive (i.e., the unit activation because of network-internal connectivity alone). Figure 2B depicts the aforementioned drives elicited from an example sequence sampled from test data (see "RNN architecture and training procedure" for more details). Two observations with respect to the network trained with preactivation loss are of importance in this example. First, the network trained with preactivation loss clearly predicts and inhibits the upcoming digit category. Second, as time progresses, the network is able to integrate information, leading to better knowledge of the current sequence position and more targeted inhibition (e.g., the inhibition of later digits appears more strong/targeted than earlier predictions that appear weaker/smoother). Better predictions, in turn, result in progressively lower network activity, as quantified previously. This observation follows from the shape of our loss function under "RNN architecture and training procedure." Replacing p_t with $-p_t$ in Equation 1 will yield a prediction error loss. This tight link has been established before.[31–36] In our work, the link can be explained in the following way: the preactivation allows "predictive inhibition" while forcing the network weights to be as small possible so that irreducible noise has as little impact as possible on the network activity. We refer the reader to thesupplemental information for a derivation of why this is the case. Another important observation is that the network trained on unit outputs alone is not able to generate sharp predictions, showcasing that this objective is not a good proxy for energy consumption. In addition

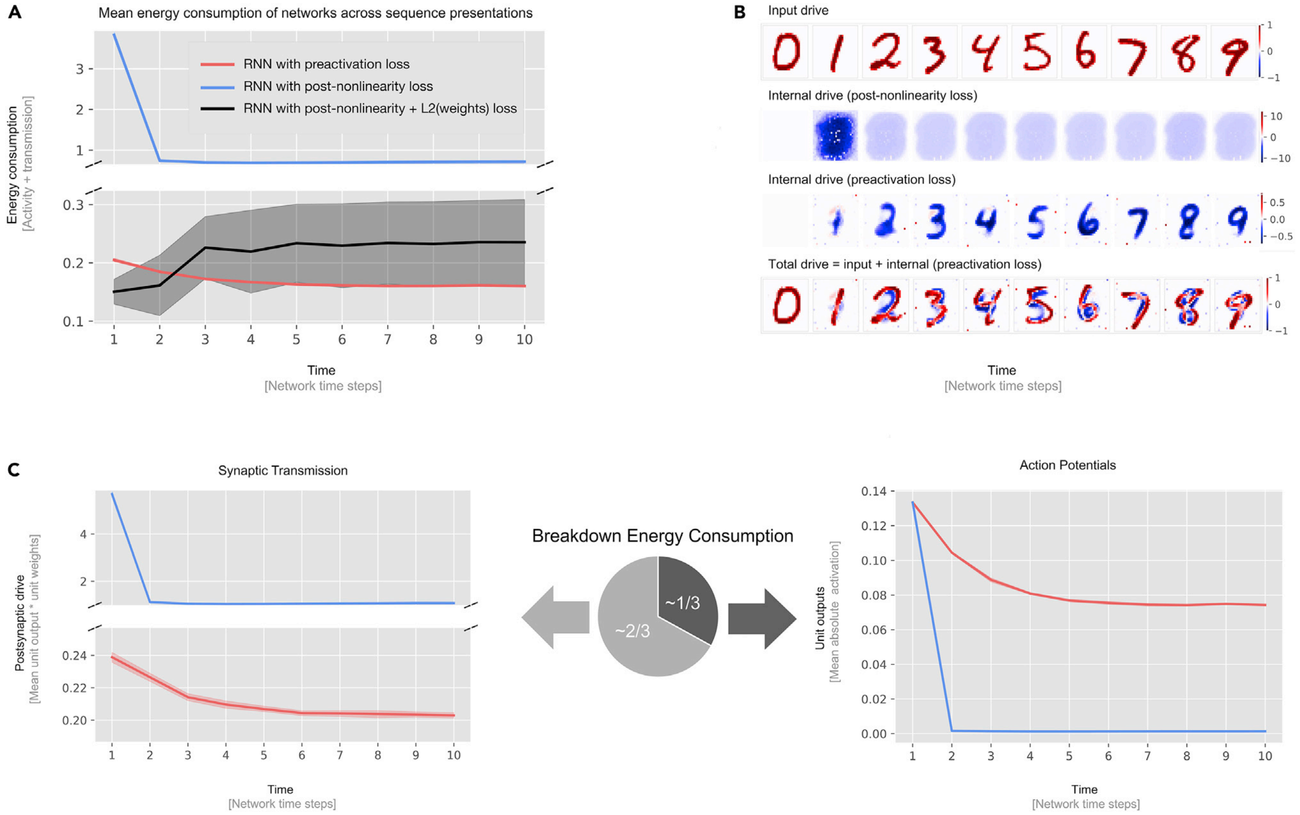


Figure 2. Networks trained with preactivation minimize their energy consumption and learn to predict their input

(A) Evolution of the "energy consumption" of trained networks. The models trained to minimize preactivation are compared with control models that are trained to minimize network output (post non-linearity) with an L1 norm on the network output plus an L2 norm on the network weights. Each plot is shown with 95% CIs, bootstrapped across 10 network instances. Networks trained to minimize preactivation achieve lower energy when presented with sequences.

(B) Visualization of RNN unit drives and activity for an example sequence. Units are organized according to their input pixel location. Excitatory signals are depicted in red, and inhibitory signals are depicted in blue. The darker the color, the stronger the signal. The sequence images shown were not used for training. The observable improvements in inhibitory trained network drive indicate that the RNN trained for preactivation minimization integrates sensory evidence over time, leading to better internal knowledge of the state of the world and more targeted inhibitory predictions. This is in stark contrast with the drive from the network trained with post-non-linearity loss, which displays imprecise inhibition that also does not improve over the sequence. If we look at the total drive (input + internal drive; i.e., preactivation), then we see that the internal drive is slightly off in matching the input. This is likely a result of the network not being able to match variation of individual writing styles in a particular digit category.

(C) Breakdown of energy consumption into activity and synaptic transmission for networks trained with preactivation versus networks trained on activation only. The networks trained with activation outperform the preactivation networks by a substantial margin but do so by incurring a massive cost in synaptic transmission, leading to an overall increase in synaptic transmission.

to the expected emergence of prediction in our model, this pattern of results is in line with the hierarchical predictive coding framework, in which feedback from higher-order areas is subtracted from sensory evidence of lower areas. However, a hierarchical organization was not imposed in our RNNs. Rather, it emerges because of optimization of the networks for energy efficiency.

Separate error units and prediction units as an emergent property of RNN self-organization

Our previous results demonstrate that RNNs, when optimized to reduce their energy consumption (action potentials and synaptic transmission), exhibit key phenomena associated with predictive coding. Predictive coding emerges, without architectural hardwiring, as a result of energy efficiency in predictable sensory environments. A second key component postulated by the predictive coding framework is the existence of distinct neural populations that provide sensory predictions or carry the deviations from such predictions; i.e., prediction and error units.[54] Given the non-hierarchical, non-constrained nature of the current setup, we next wanted to determine whether a similar separation could also be observed in our energy-efficient networks.

If the networks had indeed developed prediction units, we could potentially identify them by looking at biases in their (median) preactivation at the latter time points of the sequence (i.e., the time when network dynamics should be most stable). The rationale is as follows: because the networks are trained to minimize absolute preactivation, we would expect that the median preactivation for a unit is close to zero. However, if a units' median preactivation is significantly different from zero, then that would imply that this unit has some functional role in the network, which, in this case, is prediction. Following this rationale, we

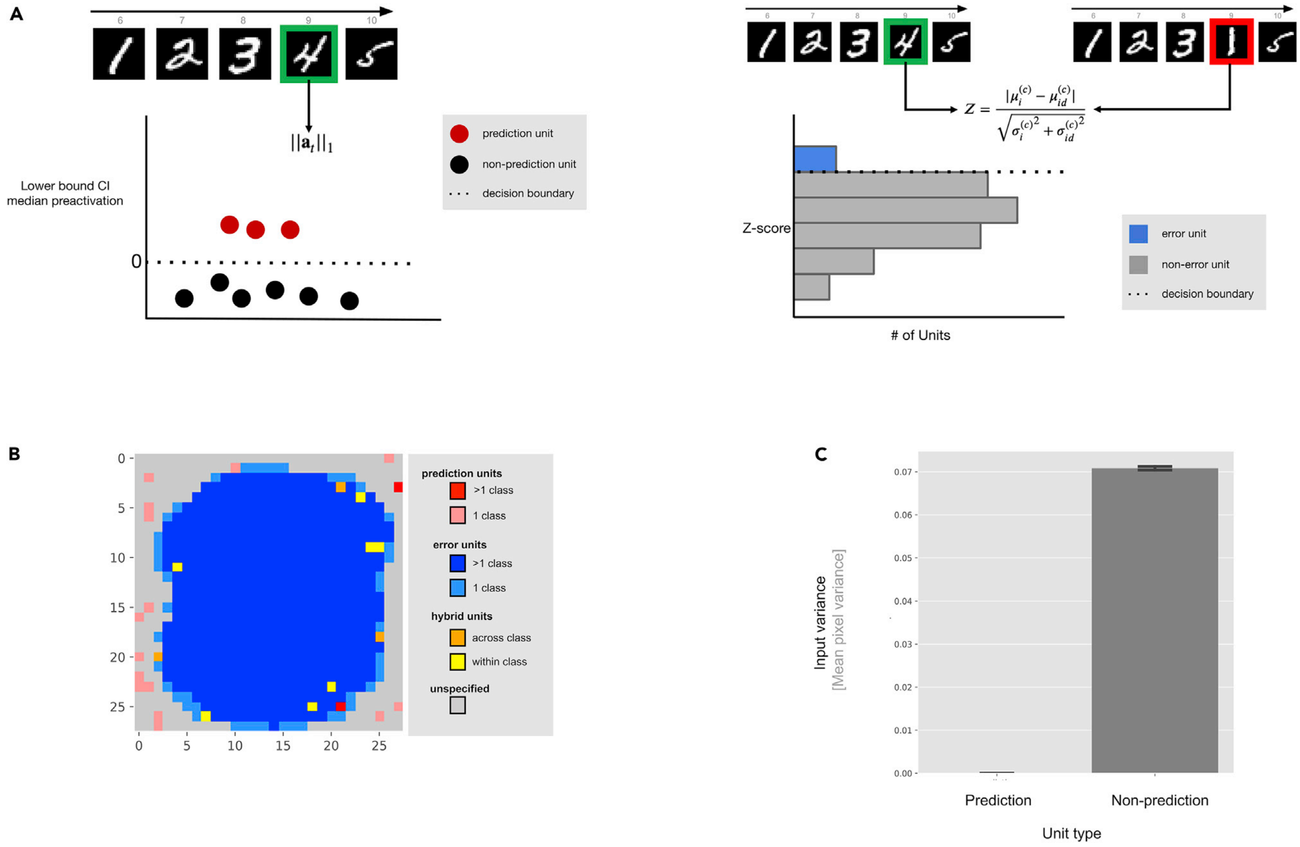


Figure 3. Trained RNNs exhibit populations of prediction and error units

(A) Illustration of how prediction and error units are determined. The left illustration shows the process of prediction error selection. For each unit, the median preactivation and a CI around that median are determined (for a particular class). If zero is not within the CI, then our method identifies the unit as being predictive (for that class). The right illustration shows how error-signaling units are determined. The network is subjected to two sets of sequences. In one set, the sequence contains a distractor digit that breaks the order in the sequence. If a unit's (class-wise) mean response differs significantly between the two sets of sequences (99% CI), then we label them as error signaling.

(B) Topographical distribution of unit types in MNIST space. Gray units have no functional role in the network. Units that only have one functional role (i.e., prediction or error signaling) are stained red or blue, respectively, with the particular shade depending on whether the units respond to a single class or multiple classes. Units that are identified as performing prediction and error signaling (i.e., hybrid) are stained orange or yellow, depending on whether they fulfill both roles in a class or across classes.

(C) Input pixel variance as a function of unit type. Prediction units emerge in areas with low pixel variance (error bars shown with 95% CI).

identified candidates for prediction units by calculating the median preactivation of each unit across multiple instances of each digit category during presentation of the final element of the sequences.

Units were labeled to be potentially predictive when, for any one class, the 99% confidence interval (CI) around the median preactivation did not include zero. That is, they showed a consistently deviation from a "zero activity" profile, which would have been inhibited if it did not serve a functional role in the overall objective of energy efficiency. See "Determination of prediction and error units" for details about the exact procedure for categorizing units and calculation of CIs for the median preactivation and median absolute deviation (MAD) around the median preactivation.

To identify error-signaling units, a different approach is necessary. If the networks developed error units, then they could be found by analyzing the unit responses during presentation of surprising inputs. In this light, we constructed an additional set of sequences in which the digit at the second-to-last time point is swapped out with a random digit from another digit class. Response distributions of each network were constructed for these surprising sequence events and compared with the response distribution for normal sequences. Units were labeled to be potentially error-signaling, if the means of the respective distributions were statistically different (99% CI). An illustration of this analysis is shown in Figure 3A. When all prediction units are collected (16.7 ± 0.48 units, depending on the particular network instance), they occupy a different topographic part of the RNN relative to the error units (see Figure 3B, which shows the general distribution of postulated prediction and error units across all 10 digit categories). Units with consistent non-zero activation (i.e., the postulated prediction units) reside in locations that are in low-pixel variance areas typically not strongly driven by sensory input, as evident in the structure of the MNIST dataset. Experiments on CIFAR-10(Figure 5B) with networks with and without additional latent resources confirm that prediction units

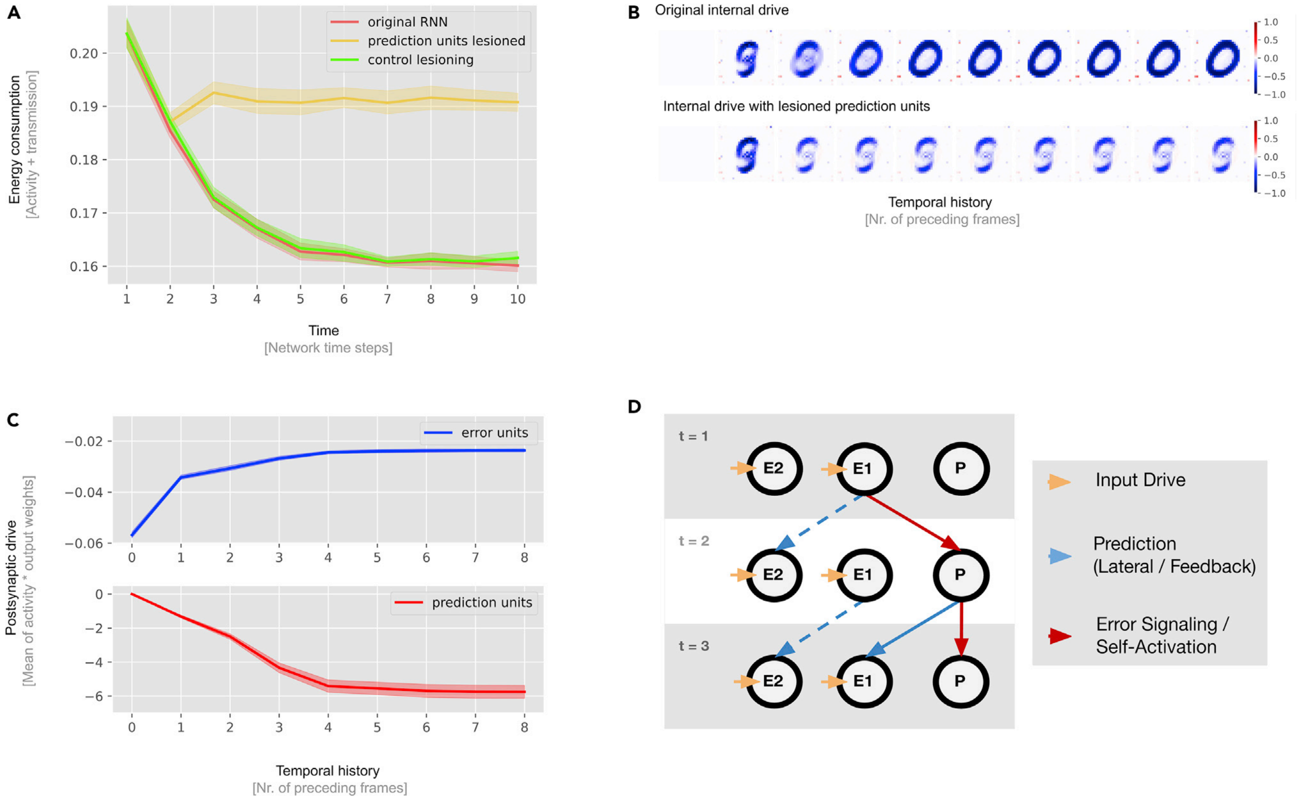


Figure 4. Lesion studies reveal the causal role of prediction units

(A) Mean energy consumption of normal RNNs (red line), prediction unit lesioned RNNs (yellow line), and control lesioned RNNs (green line). In contrast to regular RNNs, the prediction unit lesioned RNNs do not reduce their energy consumption throughout the sequence, indicating that they are not capable of integrating evidence over time. Control experiments demonstrate that lesioning non-prediction units instead of prediction units has little effect on network dynamics (green line). From this, we can conclude that the prediction units have a unique functional role in prediction. Mean energy consumption is shown with 95% CIs, bootstrapped across 10 network instances with replacement.

(B) Internal drive (p_t) in the RNN for the “0” digit category as a result of varying temporal history prior to stimulus presentation. Each square shows the internal drive of the network following various sequence lengths prior to the stimulus of interest; e.g., the sixth image displays the internal drive after the sequence “5-6-7-8-9” was presented to the network. The first row shows the predictions of a normal RNN, and the second row shows the result of lesioning RNN prediction units. The internal drive of the lesioned network does not develop better predictions with longer sequence lengths.

(C) Prediction and error units have different postsynaptic drive dynamics. Initially, error units inhibit the network, but this inhibitory drive diminishes as prediction units take over. This is in line with the hypothesis of two mechanisms of prediction that operate on different timescales.

(D) Illustration of how the two predictive mechanisms might interact in a toy network of two error units (E1 and E2) and one prediction unit (P). Initial input drive excites an error unit, which then inhibits neighboring error units (lateral inhibition) and excites adjacent prediction units. In the next time step, the prediction unit inhibits its target error-signaling units (feedback inhibition). In this example, E1 one might be a target because its activity persists in the next time step.

tend to emerge in areas of lower input variance. The peripheral locations of the putative prediction units occupy parts of the input space that are typically black (i.e., unused) or lower variance when such “empty” input space does not exist. Such secondary use of neural resources that would otherwise remain unused is in line with neuroscientific evidence for cortical plasticity and reorganization.[55,56] In addition to prediction and error units, we also observe a population of units that behave as prediction and error units; i.e., hybrid units. Investigating the role of these (few) units in the network dynamics will be the focus of future work. We refer the reader to Figure S4 for a visualization of the topography of an untrained network.

Until now, we have not established a causal effect or functional role of the postulated prediction units in the network dynamics. This will be addressed in depth in the next section.

Prediction units integrate evidence over time for more targeted inhibition

To explicitly test their functional role in RNN dynamics, we performed virtual lesion experiments with the candidate prediction units, as identified in the previous section. As shown in Figure 4B, top row, an unlesioned RNN is capable of more fine-grained predictions with increasing sequence length exposure. In stark contrast to this, networks with lesioned prediction units (Figure 4B, bottom row) remain fixed at the initial and immediate prediction and lose the ability to integrate evidence over time. This prediction resembles the median of all images of the MNIST data (Figure S3), which the is the optimal solution when the network cannot integrate category-specific information. Figure S2 shows more lesion experiments targeting other digit categories. Next we quantified the effects of lesioning prediction units (Figure 4A). After an initial

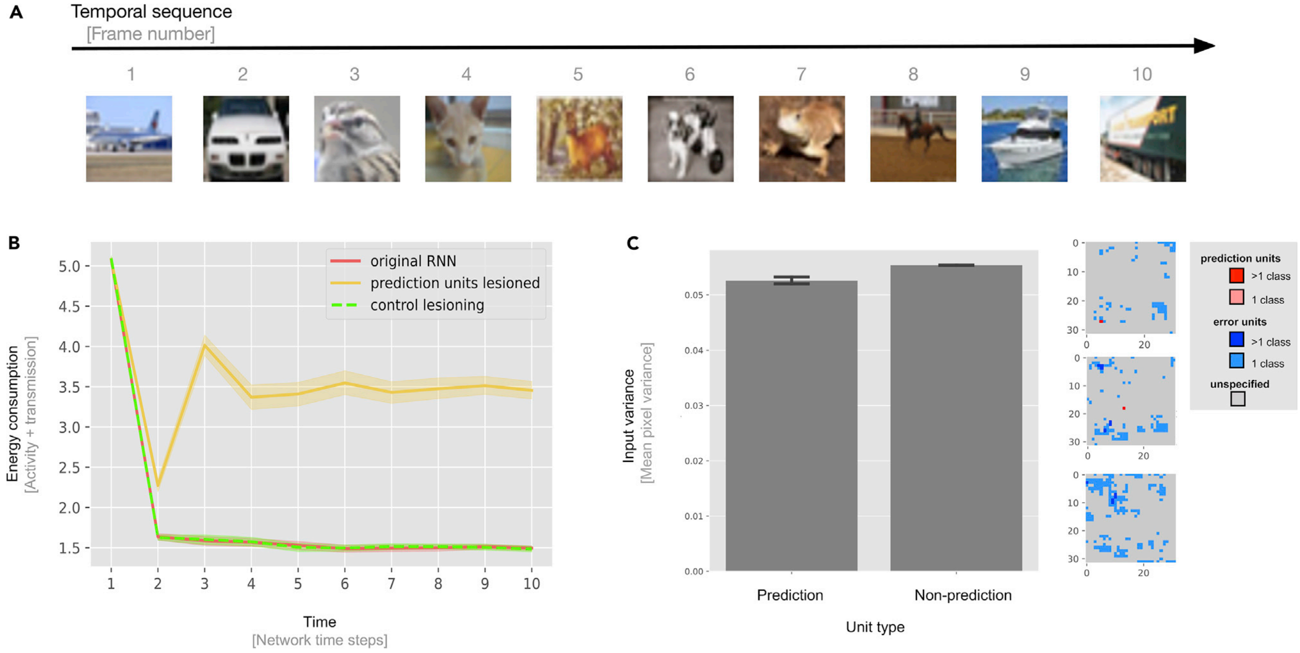


Figure 5. Results generalize to CIFAR-10

(A) Example input sequence fed to the network. In each run, 10 randomly sampled CIFAR-10 images are presented in ascending order to the network (with wraparound after image class 9).

(B) Mean energy consumption of normal RNNs (red line), prediction unit lesioned RNNs (yellow line), and control lesioned RNNs (green line). In contrast to regular RNNs, the prediction unit lesioned RNNs do not reduce their energy consumption throughout the sequence, indicating that they are not capable of integrating evidence over time. Control experiments demonstrate that lesioning non-prediction units instead of prediction units has little effect on network dynamics (green line). From this, we can conclude that the prediction units have a unique functional role in prediction. Mean energy consumption is shown with 95% CIs, bootstrapped across 10 network instances with replacement.

(C) Prediction units have mildly lower input variance. Left: input pixel variance as a function of unit type. Prediction units tend to emerge in areas with low pixel variance. Error bars are shown with 95% CIs. Right: topographical distribution of populations in CIFAR-10 space across the three color channels (R, G, and B). Gray units have no functional role in the network. Units that only have one functional role (i.e., prediction or error signaling) are stained red or blue, respectively, with the particular shade depending on whether the units respond to a single class or multiple classes. No hybrid units were identified in CIFAR-10.

reduction in energy consumption (investigated in depth in the next section), the lesioned networks fail to reduce their activity over time (yellow line). As a control experiment, we lesioned the same number of units from outside the population of identified prediction units rather than prediction units (green line). This had a minimal effect on network energy consumption, verifying the special role of prediction units in overall network dynamics. See “procedure for prediction unit lesions and control lesions” for a more detailed description of the lesion procedures.

Two distinct inhibitory mechanisms work at different timescales

As can be surmised from Figures 4A and 4B, the energy consumption of the network drops from the first to the second timestep by means of an imprecise but category-specific inhibition. This network behavior is of interest because potential prediction loops, from input to prediction units and back, require two time steps to come into effect (one to activate prediction units and one for these prediction units to provide some feedback). To confirm this observation, we looked at how prediction and error units drive other units with increasing temporal history (Figure 4C). We observe that their postsynaptic drive dynamics show opposite trends. The initial inhibitory drive results from the error units, but as temporal history increases (i.e., more of the sequence is observed), their inhibitory drive weakens. Contrary to this, prediction units show a continual increase in inhibition as more of the sequence is observed, followed later by a saturation period. This suggests that the observed predictions in time step two are rather driven by more immediate lateral connections among error units; to our knowledge, a mode of predictive coding not considered previously. We observe two different modes of predictive inhibitions: one operating at a faster timescale among error units and one on a slower timescale involving prediction units with the potential to integrate evidence over time. See Figure 4D for a schematic of how these two mechanisms of predictions may operate in the network.

Replication of main results: CIFAR-10

Having established that predictive coding can emerge as a result of training RNNs for energy efficiency, we tested the generality of our results by testing on a separate dataset with more realistic image statistics (e.g., full image coverage, unlike MNIST, in which information is predominantly present in central pixels). In particular, we performed experiments as before but used sequences of CIFAR-10 images (Figure 5A). Replicating our previous results, we again observe the emergence of two distinct populations of units, with lesions to prediction units exhibiting strong effects on predictive performance and energy consumption of the network (Figure 5B). One notable difference is that that we observe far fewer prediction and error units in CIFAR-10 (1.8 ± 0.42 and 179.2 ± 137.77 prediction units on average) than in MNIST (16.7 ± 0.48 and 552 ± 1.7 prediction and error units, respectively, on average) despite the networks being bigger (784 units versus 3072 units). We attribute this to differences in the pixel variance profile of the data for CIFAR-10 versus MNIST, as can be seen in Figures 3B, 3C, and 5C. Prediction units tend to congregate in areas with lower pixel variance. MNIST clearly separates into high and low variance pixels, whereas input variance in the case of CIFAR-10 is much more varied because of not having perpetually dark pixels. In MNIST, the prediction units appear in areas with low variance (the edges), whereas error units appear in areas with high pixel variance (the center). In CIFAR-10, although less pronounced because of the less extreme fluctuations in pixel variance, we observe a similar trend. We also conducted analyses of models with additional "latent" (i.e., excess) units that are not driven by any input. In these models, the prediction units consistently emerged in the latent part of the network, which, by architectural design, is part of the network with the lowest input variance. This pattern could be explained by the fact that error-signaling units seem to be characterized by shifts in response distributions, whereas prediction units benefit from low variance to slow down their dynamics and respond to fluctuations caused by error signaling.

The networks are still able to minimize preactivation and reproduce the lesioning results seen in the MNIST dataset (Figure 5B). See "data generation" for specific details about the CIFAR-10 dataset.

DISCUSSION

Here, we demonstrate that RNN models, trained to minimize biologically realistic energy consumption (action potentials and synaptic transmissions), spontaneously develop hallmarks of predictive coding without the need to hardwire predictive coding principles into the network architecture. This opens up the possibility that predictive coding can be understood as a natural consequence of recurrent networks operating under a limited energy budget in predictive environments. In addition to category-specific inhibition and increasingly sharpened predictions with time, we observed that the RNNs self-organize into mainly two distinct unit populations: error and prediction units. Beyond these processes, we observe two distinct mechanisms of prediction: a faster inhibitory loop among error units and a slower predictive loop that enables RNNs to integrate information over longer timescales and, therefore, generate increasingly fine-grained predictions. This observation can be interpreted as a rudimentary form of hierarchical self-organization in which the predictive units can be viewed as a higher-order cortical population operating on longer timescales and the error units as a lower-order cortical population operating on shorter timescales. This interpretation is consistent with hierarchical predictive coding architectures[18,22] as well as the organization of the brain in terms of a hierarchy of timescales.[57,58] The phenomenon of specialization in the circuit has also been demonstrated to be a useful notion by Zeldenrust et al.,[59] who analytically derived a class of spiking recurrent predictive coding networks with neural heterogeneity and have shown that these networks can better explain away prediction errors than homogeneous networks. This may indicate that neural heterogeneity and specialization are integral parts of neural systems that have to be efficient in predictive environments.

Our loss function is computed from the network preactivation and, therefore, fully depends on the input drive and the recurrent drive. This implies an optimal solution in which the recurrent drive corresponds to the negative input drive. The network could therefore be seen as optimizing for the negative target, and, therefore, in this context, there is little to no difference between energy minimization and prediction of network input.

Although it is true that minimizing the preactivation is equivalent to predicting the (negative) input drive, this does not necessarily imply a predictive coding framework. In particular, predictive coding as a framework consists of a functionally distinct population of units interacting in a hierarchical fashion.[54] Thus, the loss function minimized, despite its functional relationship to prediction, does not build in these assumptions of structure. This allows us to conclude that our loss function, although intimately linked to prediction, does not assume the predictive coding framework and that this is instead an emergent property.

One might wonder why we opted for RNNs over feedforward networks, given their popularity in brain data modeling. We believe that the time-varying nature of the network input requires a network architecture that can take time into account during its computation through a form of recurrence. In principle, this can be implemented in feedforward networks by temporally unrolling an equivalent RNN. However, without weight sharing, such a network would not be able to implement lateral or top-down information flow and would be expensive in terms of parametric complexity.

This work builds on a number of studies investigating the relationship between predictive systems, environments, and their effects on efficiency. From a thermodynamics perspective, Still et al.[34] demonstrated that a system with memory exposed to a stochastic signal must be predictive to operate at maximal energy efficiency. Candadai and Izquierdo[35] showed, information theoretically, that predictable environments produce neural networks that exhibit predictive information. In the work by Sengupta et al.,[60] the minimization of thermodynamic energy was linked to information processing and efficiency using the Jarzynski equality. This formulation highlights the deep connection between the thermodynamic energy and the computational cost of (Bayesian) belief updating. In this instance, the authors were able to show that the thermodynamic equilibrium, which minimizes thermodynamic free energy, coincides with the minimum of variational free energy that underwrites approximate Bayesian inference.[31–33,36] This is important because predictive coding can be cast as Bayesian filtering (e.g., Kalman filtering), which entails a minimization of variational free energy.[61] This variational free energy is also known as an evidence lower bound and is the objective function used in variational autoencoders.[32]

Although we have framed the choice of the preactivation energy function in terms of thermodynamic free energy, we could have also chosen an information theoretic cost function based on variational free energy. Interestingly, variational free energy can also be derived in terms of minimum message length or minimum description length in the setting of Kolmogorov complexity and Solomonov induction and related formulations.[31,62,63] In turn, this leads to formulations of universal computation that speak again to the minimization of (variational free) energy in affording the most compressed representation of some input. This was the original motivation for linear predictive coding in the 1950s[64] and fits very comfortably with the formalism presented in this work.

On a neural level, it was demonstrated that tightly balanced excitation-inhibition can be understood as neural networks efficiently coding information, with membrane potentials of neurons interpretable as a prediction error of a global signal.[65–68] Masumori et al.[69] demonstrate that a spiking neural network, solely trained based on spike timings, learns to predict temporal sequences, proposing that predictive coding arises to avoid stimulation of biological networks. This current work extends this body of evidence by demonstrating that the framework for predictive coding can emerge in recurrent networks that are trained for a simple consideration of energy efficiency, reproducing functional components of predictive coding in a biologically inspired network setup. To concretely link the impact of our loss function to energy efficiency in terms of action potentials and synaptic weights, we derive the impact of noise and input variability (zero mean but with arbitrary distribution) as aiding in minimization of the size of the synaptic weights used to minimize the network's activity and precluding a solution where synaptic weights are large but cancelling one anothers' network impact.

The current work, because of its reliance on smaller-scale datasets (MNIST and CIFAR-10), represents a proof of concept. The reliable findings across two separate datasets and the aforementioned findings regarding the link between prediction and energy efficiency indicates, however, that the observed results are likely more general. Larger categorical datasets, such as CIFAR-100 or ImageNet, would increase the complexity of the problem by the larger number of categories available. This would allow training and testing on longer sequences. We do not see principled reasons why these longer sequences should not yield a similar separation into prediction and error units. We also must take note of the fact that our loss function does not cover all metabolic sources in the brain, such as the maintenance associated with longer wire lengths,[70] which can induce more energy constraints on the model. It will be interesting for future work to investigate the interplay of both sources of energy cost, information transfer and wiring length, in a joint setting.

Given that we hypothesized that energy efficiency is sufficient for the basic components of the predictive coding framework to emerge, we restricted ourselves to a single-layer recurrent network in which units can freely connect to each other. Future work would entail extending the current model to a multilayer network so that communication between prediction and error units between layers can be investigated in more detail. In terms of model details, future work could also explore energy efficiency in spiking network models where membrane voltages and spike times must be considered because the current set of results relies on a rate-coded neuron model. How a predictive system should optimally process differing spatial and temporal scales of dynamics and deal with unpredictable but information-less inputs (e.g., chaotic inputs and random noise) are key areas for future consideration.

Our current set of findings suggests that predictive coding principles do not necessarily have to be hard-wired into the biological substrate but can be viewed as an emergent property of a simple recurrent system that minimizes its energy consumption in a predictable environment. We have shown here that minimizing unit preactivations implies minimizing unit activity and synaptic transmission. This observation opens interesting avenues of research into efficient coding and neural network modeling.

EXPERIMENTAL PROCEDURES

Resource availability

Lead contact

The lead contact is A.A. (abdullahi.ali@donders.ru.nl).

Materials availability

There are no newly generated materials.

Data and code availability

All data and analysis code along with instructions are available at https://osf.io/c57d4/(https://doi.org/10.17605/OSF.IO/C57D4).

Data generation

MNIST

The input data are sequences of images drawn from the MNIST database of handwritten digits.[50] Images are of size 28×28 with pixel intensities in the range $[0, 1]$. There are 60,000 images in the training set and 10,000 images in the test set. Each set of images can be divided into 10 categories; one for each digit. The frequency of each digit category in the dataset varies slightly (frequencies lie within 9%–11% of the total dataset (70,000 samples)). Sequences are generated by choosing random digits as starting points. Digits from numerically subsequent categories are then randomly sampled and appended to the sequence until the desired sequence length is reached. The categories are wrapped around so that, after category "nine," the sequence continues from category "zero." The sequence length can be chosen in advance, but all sequences are constrained to have the same length (in our simulations, we take a sequence length of 10). The sequences are organized into batches. The images are drawn without replacement, and the process is stopped when there are no image samples available for the next digit. Incomplete batches, in which there are no remaining image samples to complete a sequence, are discarded.

CIFAR-10

The input data are sequences of images drawn from CIFAR-10, a labeled subset of the tiny image database.[71] CIFAR-10 consists of 10 classes. These classes are as follows: airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks. Images are colored and of size 32×32×3 with pixel intensities in the range $[0, 1]$. There are 50,000 images in the training set and 10,000 images in the test set. Each set of images can be divided into 10 categories; one for each class. The frequency of each category in the data set is exactly 6,000. Sequences are generated by choosing random images as starting points. Images from numerically subsequent categories are then randomly sampled and appended to the sequence until the desired sequence length is reached. The categories are wrapped around so that, after category "nine," the sequence continues from category "zero." The sequence length can be chosen in advance, but all sequences are constrained to have the same length (in our simulations, we take a sequence length of 10). The sequences are organized into batches. The images are drawn without replacement, and the process is stopped when there are no image samples available for the next class. Incomplete batches, in which there are no remaining image samples to complete a sequence, are discarded.

RNN architecture and training procedure

We created a fully connected RNN consisting of 784 units for MNIST and 3,072 units for CIFAR-10. Each unit is driven by exactly one input pixel, which means that the number of units exactly matches the number of pixels in the image. The equations that determine the RNN dynamics are

$$\mathbf{p_t} = \mathbf{W}\mathbf{h_{t-1}}, \quad \text{(Equation 1)}$$

$$\mathbf{a_t} = \mathbf{p_t} + \mathbf{x_t}, \quad \text{(Equation 2)}$$

$$\mathbf{h_t} = \mathbf{f}(\mathbf{a_t}), \quad \text{(Equation 3)}$$

where $\mathbf{x_t} \in \mathbb{R}^N$ denotes the input drive, $\mathbf{a_t} \in \mathbb{R}^N$ denotes the preactivation, $\mathbf{W} \in \mathbb{R}^{N \times N}$ denotes the recurrent weight matrix of the RNN (the only learnable parameters), $\mathbf{p_t} \in \mathbb{R}^N$ denotes the recurrent feedback, and $\mathbf{h_t} \in \mathbb{R}^N$ denotes the unit outputs (in our study, $\mathbf{f}$ are ReLU non-linearities). The subscripts t and $t-1$ refer to the discrete (integer) timestep of the system because all of these variables (aside from the weights) are iteratively updated. The weight matrix is uniformly initialized in $[-1, 1]$ and scaled by $N^{-\frac{1}{2}}$; i.e., proportionally scaled by the number of units in the weight matrix.

The objective of minimizing energy is captured through the following loss function:

$$\ell = \frac{1}{NT}\sum_{t=1}^{T}\|\mathbf{a}_t\|_1, \qquad \text{(Equation 4)}$$

where T is the number of time steps, N is the number of units, and $\mathbf{a}_t$ is the preactivation of the units when processing the t -th element (or timestep) in a given sequence. We trained 10 model instances for 200 epochs on MNIST and 10 model instances for 1,000 epochs on CIFAR-10 with batch size 32 (32 sequences per batch) with the Adam optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$, learning rate $= 0.0001$).[72] Model training differed in weight initialization and training sequence. We tested the same 20 model instances (lesioned models in Figure 4A and Figure 5C) but with unit activity $\mathbf{h_{t-1}}$ masked so that the feedback-driven units did not affect the computation of $\mathbf{a_t}$. No extensive hyperparameter search was performed.

In addition to the models trained with the preactivation loss, we also trained two baseline models (10 instances each) with equivalent architectures and training procedures with different losses. The first baseline was trained with a loss on the unit activity post non-linearity (i.e., network activity):

$$\ell_{b1} = \frac{1}{NT}\sum_{t=1}^{T}\mathbf{h}_t, \qquad \text{(Equation 5)}$$

The second baseline was trained with the post-non-linearity loss and L2 regularization on the weights

$$\ell_{b2} = \frac{1}{NT}\sum_{t=1}^{T}(\mathbf{h}_t + \lambda\|\mathbf{W}\|), \qquad \text{(Equation 6)}$$

Here λ controls the relative impact of the weight regularization and is picked (through a search) to be as close as possible to the energy consumption of the preactivation loss model (here $\lambda = 3708$). The models were trained using PyTorch 1.5.0 and Python 3.7.7.

Determination of energy consumption

To benchmark how well the model minimizes energy consumption, we use the following measure for energy consumption for a particular sample m on a particular time point t:

$$E_t^m = \alpha_1\overline{\mathbf{h_t^m}} + \alpha_2\overline{\mathbf{s_t^m}}, \qquad \text{(Equation 7)}$$

where

$$\mathbf{s_t^m} = |\mathbf{h_t^m}| \otimes |\mathbf{W}|, \qquad \text{(Equation 8)}$$

α_1, α_2 are constants that weigh the impact of activity and synaptic transmission (in this work, $\alpha_1 = \frac{1}{3}$, $\alpha_2 = \frac{2}{3}$) and $\overline{\mathbf{h_t^m}}$, $\overline{\mathbf{s_t^m}}$ denote the average network activity and network synaptic transmission for m at time point t. The mean energy consumption for a network at time point t over all samples in the test data set will be

$$\overline{E}_t = \sum_{m=1}^{M}\alpha_1\overline{\mathbf{h_t^m}} + \alpha_2\overline{\mathbf{s_t^m}}. \qquad \text{(Equation 9)}$$

Determination of prediction and error units

To determine whether units are predictive, we record their median preactivations for each category at the final time step of a sequence, when the dynamics of the network are most stable. We construct 99 % CIs around this median. To obtain a 99 % CI, we first compute the MAD around the category median preactivation. Supposing $\mu^{\frac{1}{2}}\left(a_i^{(c)}\right)$ is the category median preactivation at the final time point for a particular unit i, we can compute the MAD for a particular category and unit as follows:

$$MAD_i^{(c)} = \mu^{\frac{1}{2}}\left(\left|a_{i,m}^{(c)} - \mu^{\frac{1}{2}}\left(a_i^{(c)}\right)\right|\right), \qquad \text{(Equation 10)}$$

where $a_{i,m}^{(c)}$ is the preactivation of unit i for an arbitrary sample m of category c. This means that you can find the MAD for a particular unit and category by taking the median of the absolute deviations of the unit sample preactivations from the median preactivation.

CIs can only be analytically calculated for Gaussian distributed random variables using the standard deviation around the mean, so we have to use an approximation. We can do this by converting the obtained MADs to pseudo-standard deviations[73] in the following way:

$$\widehat{\sigma}_i^{(c)} = 1.4826 \cdot MAD_i^{(c)}, \qquad \text{(Equation 11)}$$

where $\widehat{\sigma}_i^{(c)}$ is the pseudo-standard deviation around the median preactivation for unit i and category c, and 1.4826 is the scaling constant for Gaussian distributed variables.

The 99 % CI will then be

$$CI_i^{(c)} = \left[\mu^{\frac{1}{2}}\left(a_i^{(c)}\right) - 2.576\widehat{\sigma}_i^{(c)}, \mu^{\frac{1}{2}}\left(a_i^{(c)}\right) + 2.576\widehat{\sigma}_i^{(c)}\right]. \qquad \text{(Equation 12)}$$

where 2.576 is the Z score that determines the bounds of the CI. A unit i will be identified as a prediction unit when $0 \notin CI_i^{(c)}$ for at least one c.

To determine whether units signal error, we provide the networks with additional sequences that have a late-time-point (in this work, $t = 8$) distractor (an image that breaks the order of the sequence) and record the unit responses (i.e., unit outputs post non-linearity). For each unit i and class c, a distribution over responses for distractor sequences $P_{id}^{(c)}(r_{id}^{(c)})$ is constructed and compared with $P_i^{(c)}(r_i^{(c)})$, the response distribution for normal sequences, where

$$P_i^{(c)}\left(r_i^{(c)}\right) = \mathcal{N}_i^{(c)}\left(r_i^{(c)}; \mu_i^{(c)}, {\sigma_i^{(c)}}^2\right), \qquad \text{(Equation 13)}$$

$$P_{id}^{(c)}\left(r_{id}^{(c)}\right) = \mathcal{N}_{id}^{(c)}\left(r_{id}^{(c)}; \mu_{id}^{(c)}, {\sigma_{id}^{(c)}}^2\right). \qquad \text{(Equation 14)}$$

Unit i is determined to be signaling error for class c when

$$\frac{\left|\mu_i^{(c)} - \mu_{id}^{(c)}\right|}{\sqrt{{\sigma_i^{(c)}}^2 + {\sigma_{id}^{(c)}}^2}} > 2.576, \qquad \text{(Equation 15)}$$

that is, we can say with 99% confidence that the distributions are different.

Procedure for prediction unit lesions and control lesions

The procedure for the lesion experiments on prediction units is as follows: using the method described in "Determination of prediction and error units," prediction units were detected for all digit classes. Subsequently, the set of activations and outputs of all prediction units was set to zero during inference to ensure that their activity cannot affect the other units in the network. A similar procedure was employed for the control lesion experiments. In this case, however, random sets of non-prediction units in the network were lesioned. The number of lesioned units in the control condition was set to be equal to the original number of prediction units lesioned.

Calculating postsynaptic drive dynamics of prediction and error units

To calculate the postsynaptic drive dynamics of prediction and error units, we first determined the prediction and error units (separately for each network instance). For each prediction and error unit, we calculate their synaptic transmission for each sequence and time point. The resulting matrix is then averaged over the number of sequences, yielding a mean postsynaptic drive curve for each prediction and error unit. The corresponding curves, shown in Figure 4C, indicate the average synaptic transmission originating from prediction and error units, averaged across network instances together with 95% Cis, obtained via bootstrapping.

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.patter.2022.100639.

ACKNOWLEDGMENTS

We thank Courtney J. Spoerer for feedback and helpful discussions during the early phase of the project.

AUTHOR CONTRIBUTIONS

T.C.K., N.A., and M.A.J.v.G. conceptualized the project and designed the methodology. A.A. and E.G. wrote the code and performed computational experiments. N.A. helped with the analysis. A.A. and T.C.K. wrote the manuscript. N.A. and M.A.J.v.G. reviewed and edited the manuscript.

DECLARATION OF INTERESTS

The authors declare no competing interests.

Received: November 25, 2021
Revised: December 24, 2021
Accepted: October 27, 2022
Published: November 23, 2022

REFERENCES

1. Srinivasan, M.V., Laughlin, S.B., and Dubs, A. (1982). Predictive coding: a fresh view of inhibition in the retina. Proc. R. Soc. Lond. B Biol. Sci. *216*, 427–459. https://doi.org/10.1098/rspb.1982.0085.

2. Mumford, D. (1992). On the computational architecture of the neocortex. Biol. Cybern. *66*, 241–251. https://doi.org/10.1007/BF00198477.

3. Friston, K. (2005). A theory of cortical responses. Philos. Trans. R. Soc. Lond. B Biol. Sci. *360*, 815–836. https://doi.org/10.1098/rstb.2005.1622.

4. Clark, A. (2013). Whatever next? predictive brains, situated agents, and the future of cognitive science. Behav. Brain Sci. *36*, 181–204. https://doi.org/10.1017/S0140525X12000477.

5. Alink, A., Schwiedrzik, C.M., Kohler, A., Singer, W., and Muckli, L. (2010). Stimulus predictability reduces responses in primary visual cortex. J. Neurosci. *30*, 2960–2966. https://doi.org/10.1523/JNEUROSCI.3730-10.2010.

6. Näätänen, R., Tervaniemi, M., Sussman, E., Paavilainen, P., and Winkler, I. (2001). Primitive intelligence in the auditory cortex. Trends Neurosci. *24*, 283–288. https://doi.org/10.1016/s0166-2236(00)01790-2.

7. Summerfield, C., Wyart, V., Johnen, V.M., and De Gardelle, V. (2011). Human scalp electroencephalography reveals that repetition suppression varies with expectation. Front. Hum. Neurosci. *5*, 67. https://doi.org/10.3389/fnhum.2011.00067.

8. Summerfield, C., Trittschuh, E.H., Monti, J.M., Mesulam, M.M., and Egner, T. (2008). Neural repetition suppression reflects fulfilled perceptual expectations. Nat. Neurosci. *11*, 1004–1006. https://doi.org/10.1038/nn.2163.

9. Squires, N.K., Squires, K.C., and Hillyard, S.A. (1975). Two varieties of long-latency positive waves evoked by unpredictable auditory stimuli in man. Electroencephalogr. Clin. Neurophysiol. *38*, 387–401. https://doi.org/10.1016/0013-4694(75)90263-1.

10. Hupé, J.M., James, A.C., Payne, B.R., Lomber, S.G., Girard, P., and Bullier, J. (1998). Cortical feedback improves discrimination between figure and background by v1, v2 and v3 neurons. Nature *394*, 784–787. https://doi.org/10.1038/29537.

11. Murray, S.O., Kersten, D., Olshausen, B.A., Schrater, P., and Woods, D.L. (2002). Shape perception reduces activity in human primary visual cortex. Proc. Natl. Acad. Sci. USA. *99*, 15164–15169. https://doi.org/10.1073/pnas.192579399.

12. Rao, H.M., Mayo, J.P., and Sommer, M.A. (2016). Circuits for presaccadic visual remapping. J. Neurophysiol. *116*, 2624–2636. https://doi.org/10.1152/jn.00182.2016.

13. Kok, P., Jehee, J.F.M., and De Lange, F.P. (2012). Less is more: expectation sharpens representations in the primary visual cortex. Neuron *75*, 265–270. https://doi.org/10.1016/j.neuron.2012.04.034.

14. Ekman, M., Kok, P., and de Lange, F.P. (2017). Time-compressed preplay of anticipated events in human primary visual cortex. Nat. Commun. *8*, 15276–15279. https://doi.org/10.1038/ncomms15276.

15. De Lange, F.P., Heilbron, M., and Kok, P. (2018). How do expectations shape perception? Trends Cogn. Sci. *22*, 764–779. https://doi.org/10.1016/j.tics.2018.06.002.

16. Dijkstra, N., Ambrogioni, L., Vidaurre, D., and Van Gerven, M. (2020). Neural dynamics of perceptual inference and its reversal during imagery. Elife *9*, e53588. https://doi.org/10.7554/eLife.53588.

17. Schwiedrzik, C.M., and Freiwald, W.A. (2017). High-level prediction signals in a low-level area of the macaque face-processing hierarchy. Neuron *96*, 89–97.e4. https://doi.org/10.1016/j.neuron.2017.09.007.

18. Rao, R.P., and Ballard, D.H. (1999). Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. Nat. Neurosci. *2*, 79–87. https://doi.org/10.1038/4580.

19. Lee, T.S., and Mumford, D. (2003). Hierarchical bayesian inference in the visual cortex. J. Opt. Soc. Am. Opt Image Sci. Vis. *20*, 1434–1448. https://doi.org/10.1364/JOSAA.20.001434.

20. Friston, K. (2010). The free-energy principle: a unified brain theory? Nat. Rev. Neurosci. *11*, 127–138. https://doi.org/10.1038/nrn2787.

21. Chalasani, R., and Principe, J.C. (2013). Deep predictive coding networks. Preprint at arXiv. https://doi.org/10.48550/arXiv.1301.3541.

22. Lotter, W., Kreiman, G., and Cox, D. (2016). Deep predictive coding networks for video prediction and unsupervised learning. Preprint at arXiv. https://doi.org/10.48550/arXiv.1605.08104.

23. Villegas, R., Yang, J., Zou, Y., Sohn, S., Lin, X., and Lee, H. (2017). Learning to generate long-term future via hierarchical prediction. In Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017 vol. 70 of Proceedings of Machine Learning Research (PMLR), pp. 3560–3569.

24. Linsley, D., Kim, J., Ashok, A., and Serre, T. (2020). Recurrent neural circuits for contour detection. Preprint at arXiv. https://doi.org/10.48550/arXiv.2010.15314.

25. Lotter, W., Kreiman, G., and Cox, D. (2020). A neural network trained for prediction mimics diverse features of biological neurons and perception. Nat. Mach. Intell. *2*, 210–219. https://doi.org/10.1038/s42256-020-0170-9.

26. Spoerer, C.J., Kietzmann, T.C., Mehrer, J., Charest, I., and Kriegeskorte, N. (2020). Recurrent neural networks can explain flexible trading of speed and accuracy in biological vision. PLoS Comput. Biol. *16*. e1008215. https://doi.org/10.1371/journal.pcbi.1008215.

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

28. Kar, K., Kubilius, J., Schmidt, K., Issa, E.B., and DiCarlo, J.J. (2019). Evidence that recurrent circuits are critical to the ventral stream's execution of core object recognition behavior. Nat. Neurosci. *22*, 974–983. https://doi.org/10.1038/s41593-019-0392-5.

29. Spoerer, C.J., McClure, P., and Kriegeskorte, N. (2017). Recurrent convolutional neural networks: a better model of biological object recognition. Front. Psychol. *8*, 1551. https://doi.org/10.3389/fpsyg.2017.015511.

30. van Bergen, R.S., and Kriegeskorte, N. (2020). Going in circles is the way forward: the role of recurrence in visual inference. Curr. Opin. Neurobiol. *65*, 176–193. https://doi.org/10.1016/j.conb.2020.11.009.

31. MacKay, D. (1995). Free energy minimisation algorithm for decoding and cryptanalysis. Electron. Lett. *31*, 446–447. https://doi.org/10.1049/el:19950331.

32. Winn, J., Bishop, C.M., and Jaakkola, T. (2005). Variational message passing. J. Mach. Learn. Res. *6*.

33. Dauwels, J. (2007). On variational message passing on factor graphs. In 2007 IEEE international symposium on information theory (IEEE), pp. 2546–2550. https://doi.org/10.1109/ISIT.2007.4557602.

34. Still, S., Sivak, D.A., Bell, A.J., and Crooks, G.E. (2012). Thermodynamics of prediction. Phys. Rev. Lett. *109*, 120604. https://doi.org/10.1103/PhysRevLett.109.120604.

35. Candadai, M., and Izquierdo, E.J. (2020). Sources of predictive information in dynamical neural networks. Sci. Rep. *10*, 16901–16912. https://doi.org/10.1038/s41598-020-73380-x.

36. Da Costa, L., Parr, T., Sengupta, B., and Friston, K. (2021). Neural dynamics under active inference: plausibility and efficiency of information processing. Entropy *23*, 454. https://doi.org/10.3390/e23040454.

37. Barlow, H.B., and Rosenblith, W.A. (1961). Possible Principles Underlying the Transformations of Sensory Messages, *1* (MIT Press).

38. Bell, A.J., and Sejnowski, T.J. (1995). An information-maximization approach to blind separation and blind deconvolution. Neural Comput. *7*, 1129–1159. https://doi.org/10.1162/neco.1995.7.6.1129.

39. Olshausen, B.A., and Field, D.J. (1996). Natural image statistics and efficient coding. Network *7*, 333–339. https://doi.org/10.1088/0954-898X/7/2/014.

40. Bialek, W., Van Steveninck, R.R.D.R., and Tishby, N. (2006). Efficient representation as a design principle for neural coding and computation. In 2006 IEEE international symposium on information theory (IEEE), pp. 659–663.

41. Berkes, P., and Wiskott, L. (2005). Slow feature analysis yields a rich repertoire of complex cell properties. J. Vis. *5*, 579–602. https://doi.org/10.1167/5.6.9.

42. Chalk, M., Marre, O., and Tkačik, G. (2018). Toward a unified theory of efficient, predictive, and sparse coding. Proc. Natl. Acad. Sci. USA *115*, 186–191. https://doi.org/10.1073/pnas.1711114115.

43. Eckmann, S., Klimmasch, L., Shi, B.E., and Triesch, J. (2020). Active efficient coding explains the development of binocular vision and its failure in amblyopia. Proc. Natl. Acad. Sci. USA *117*, 6156–6162. https://doi.org/10.1073/pnas.1908100.

44. Attwell, D., and Laughlin, S.B. (2001). An energy budget for signaling in the grey matter of the brain. J. Cereb. Blood Flow Metab. *21*, 1133–1145. https://doi.org/10.1097/00004647-200110000-00001.

45. Lennie, P. (2003). The cost of cortical computation. Curr. Biol. *13*, 493–497. https://doi.org/10.1016/s0960-9822(03)00135-0.

46. Alle, H., Roth, A., and Geiger, J.R.P. (2009). Energy-efficient action potentials in hippocampal mossy fibers. Science *325*, 1405–1408. https://doi.org/10.1126/science.1174331.

47. Carter, B.C., and Bean, B.P. (2009). Sodium entry during action potentials of mammalian neurons: incomplete inactivation and reduced metabolic efficiency in fast-spiking neurons. Neuron *64*, 898–909. https://doi.org/10.1016/j.neuron.2009.12.011.

48. Sengupta, B., Stemmler, M., Laughlin, S.B., and Niven, J.E. (2010). Action potential energy efficiency varies among neuron types in vertebrates and invertebrates. PLoS Comput. Biol. *6*. e1000840. https://doi.org/10.1371/journal.pcbi.1000840.

49. Howarth, C., Gleeson, P., and Attwell, D. (2012). Updated energy budgets for neural computation in the neocortex and cerebellum. J. Cereb. Blood Flow Metab. *32*, 1222–1232. https://doi.org/10.1038/jcbfm.2012.35.

50. LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. Proc. IEEE *86*, 2278–2324. https://doi.org/10.1109/5.726791.

51. Wiskott, L., and Sejnowski, T.J. (2002). Slow feature analysis: unsupervised learning of invariances. Neural Comput. *14*, 715–770. https://doi.org/10.1162/089976602317318938.

52. Körding, K.P., Kayser, C., Einhäuser, W., and König, P. (2004). How are complex cell properties adapted to the statistics of natural stimuli? J. Neurophysiol. *91*, 206–212. https://doi.org/10.1152/jn.00149.2003.

53. Mehrer, J., Spoerer, C.J., Kriegeskorte, N., and Kietzmann, T.C. (2020). Individual differences among deep neural network models. Nat. Commun. *11*, 5725–5812. https://doi.org/10.1038/s41467-020-19632-w.

54. Bastos, A.M., Usrey, W.M., Adams, R.A., Mangun, G.R., Fries, P., and Friston, K.J. (2012). Canonical microcircuits for predictive coding. Neuron *76*, 695–711. https://doi.org/10.1016/j.neuron.2012.10.038.

55. Kaas, J.H. (2002). Sensory loss and cortical reorganization in mature primates. Prog. Brain Res. *138*, 167–176. https://doi.org/10.1016/S0079-6123(02)38077-4.

56. Merabet, L.B., and Pascual-Leone, A. (2010). Neural reorganization following sensory loss: the opportunity of change. Nat. Rev. Neurosci. *11*, 44–52. https://doi.org/10.1038/nrn2758.

57. Kiebel, S.J., Daunizeau, J., and Friston, K.J. (2008). A hierarchy of timescales and the brain. PLoS Comput. Biol. *4*. e1000209. https://doi.org/10.1371/journal.pcbi.1000209.

58. Baldassano, C., Chen, J., Zadbood, A., Pillow, J.W., Hasson, U., and Norman, K.A. (2017). Discovering event structure in continuous narrative perception and memory. Neuron *95*, 709–721.e5. https://doi.org/10.1016/j.neuron.2017.06.041.

59. Zeldenrust, F., Gutkin, B., and Denéve, S. (2021). Efficient and robust coding in heterogeneous recurrent networks. PLoS Comput. Biol. *17*. e1008673. https://doi.org/10.1371/journal.pcbi.1008673.

60. Sengupta, B., Stemmler, M.B., and Friston, K.J. (2013). Information and efficiency in the nervous system—a synthesis. PLoS Comput. Biol. *9*. e1003157. https://doi.org/10.1371/journal.pcbi.1003157.

61. Friston, K. (2008). Hierarchical models in the brain. PLoS Comput. Biol. *4*. e1000211. https://doi.org/10.1371/journal.pcbi.1000211.

62. Wallace, C.S., and Dowe, D.L. (1999). Minimum message length and kolmogorov complexity. Comput. J. *42*, 270–283. https://doi.org/10.1093/comjnl/42.4.327.

63. Ikeda, S., Tanaka, T., and Amari, S.-i. (2004). Stochastic reasoning, free energy, and information geometry. Neural Comput. *16*, 1779–1810. https://doi.org/10.1162/0899766041336477.

64. Elias, P. (1955). Predictive coding–i. IEEE Trans. Inf. Theory *1*, 16–24. https://doi.org/10.1109/TIT.1955.1055126.

65. Boerlin, M., Machens, C.K., and Denève, S. (2013). Predictive coding of dynamical variables in balanced spiking networks. PLoS Comput. Biol. *9*. e1003258. https://doi.org/10.1371/journal.pcbi.1003258.

66. Denève, S., and Machens, C.K. (2016). Efficient codes and balanced networks. Nat. Neurosci. *19*, 375–382. https://doi.org/10.1038/nn.4243.

67. Denève, S., Alemi, A., and Bourdoukan, R. (2017). The brain as an efficient and robust adaptive learner. Neuron *94*, 969–977. https://doi.org/10.1016/j.neuron.2017.05.016.

68. Brendel, W., Bourdoukan, R., Vertechi, P., Machens, C.K., and Denève, S. (2020). Learning to represent signals spike by spike. PLoS Comput. Biol. *16*. e1007692. https://doi.org/10.1371/journal.pcbi.1007692.

69. Masumori, A., Ikegami, T., and Sinapayen, L. (2019). Predictive coding as stimulus avoidance in spiking neural networks. In 2019 IEEE Symposium Series on Computational Intelligence (SSCI) (IEEE), pp. 271–277. https://doi.org/10.1109/SSCI44817.2019.9003066.

70. Blauch, N.M., Behrmann, M., and Plaut, D.C. (2022). A connectivity-constrained computational account of topographic organization in primate high-level visual cortex. Proc. Natl. Acad. Sci. USA *119*. e2112566119. https://doi.org/10.1073/pnas.2112566119.

71. Krizhevsky, A., and Hinton, G. (2009). Learning multiple layers of features from tiny images. Tech. Rep.

72. Kingma, D.P., and Ba, J. (2014). Adam: a method for stochastic optimization. Preprint at arXiv. https://doi.org/10.48550/arXiv.1412.6980.

73. Rousseeuw, P.J., and Croux, C. (1993). Alternatives to the median absolute deviation. J. Am. Stat. Assoc. *88*, 1273–1283. https://doi.org/10.1080/01621459.1993.10476408.