Unifying sparse coding, predictive coding, and divisive normalization

Author Names and Affiliations: Yanbo Lian[1], Anthony N. Burkitt[1]

[1] Department of Biomedical Engineering, The University of Melbourne, Melbourne, VIC, 3010, Australia

Correspondence: yanbo.lian@unimelb.edu.au

Acknowledgements: This work received funding from the Australian Government, via grant AUS-MURIB000001 associated with ONR MURI grant N00014-19-1-2571.

Conflict of interest statement: The authors declare no competing financial interests.

Abstract

Sparse coding, predictive coding and divisive normalization are thought to be underlying principles of neural circuits across the brain, supported by much experimental evidence. However, the connection among these principles is still poorly understood. In this paper, we aim to unify sparse coding, predictive coding, and divisive normalization in the context of a two-layer neural model. This two-layer model is constructed that implements sparse coding with the structure originating from predictive coding. Our results show that a homeostatic function in the model can shape the nonlinearity of neural responses, which can replicate different forms of divisive normalization. We demonstrate that this model is equivalent to divisive normalization in a single-neuron scenario. Our simulation also shows that the model can learn simple cells with the property of contrast saturation that is previously explained by divisive normalization. In summary, our study demonstrates that the three principles of sparse coding, predictive coding, and divisive normalization can be unified, such that the model has the ability to learn as well as display more diverse response nonlinearities. This framework may also be potentially used to explain how the brain learns to integrate input from different sensory modalities.

1 Introduction

The brain can perform complex daily tasks with its highly sophisticated structure. However, the brain neural circuits consisting of numerous neurons are the fundamental unit underlying brain functions. Since the Nobel-winning discovery of simple cells and complex cells in cat's primary visual cortex (V1) (Hubel and Wiesel, 1959, 1962, 1968), many experimental studies of the visual system and other areas of the brain have been conducted to discover complicated brain functions. During the meantime, different models have also been proposed to account for experimental data, hoping that the principles of proposed models can uncover the underlying mechanisms of the brain. Sparse coding (Olshausen and Field, 1996, 1997), predictive coding (Rao and Ballard, 1999) and divisive normalization (Heeger, 1992) are three significant models proposed in the last 30 years that has a huge impact on the neuroscience community because of their success in accounting for a lot of experimental data and providing a explainable mechanism for the neural circuits.

Divisive normalization, originally proposed by Heeger (1992), is a mathematical model that normalizes

the response of individual model units (representing neurons in the real neural circuits) by the division of population response of the whole neural network; i.e., that the response of each neuron not only depends on itself but also on the responses of other neurons in the network. Divisive normalization is initiated by the fact that responses of cells in V1 display significant nonlinearities and a linear model failed to account for experimental data. Heeger (1992) showed that this simple mathematical model can explain many nonlinear response properties of V1 cells such as contrast saturation and contrast invariance. Due to the wide diversity of nonlinear structure introduced through divisive normalization with different parameters of the model, divisive normalization, has been used to explain a wide range of physiological data, from the visual system (Heeger, 1992) to the olfactory system (Olsen et al., 2010) and even multisensory integration (Ohshiro et al., 2011, 2017). For a review of divisive normalization see Carandini and Heeger (2012).

Sparse coding, originally proposed by Olshausen and Field (1996, 1997), finds an efficient representation of the sensory input using a linear combination of a set of basis vectors. Sparse coding has two important assumptions, namely that the model generates a linear reconstruction of the input and this linear reconstruction is efficient. The notion of efficiency adopted in sparse coding is based on the principle of redundancy reduction proposed by Barlow (1961, 1989). Olshausen and Field (1996, 1997) applied sparse coding onto natural images and showed that this model can learn Gabor-like features after learning, which resemble receptive fields (RFs) of simple cells in V1. An appealing feature that comes with sparse coding is its learning ability, namely that the model can learn based on its input statistics, which is similar to synaptic plasticity, a key mechanism, of the brain. Apart from its success of learning RFs of cells in the visual cortex, sparse coding has also been successfully applied to other brain areas such as auditory cortex (Smith and Lewicki, 2006) and hippocampus (Lian and Burkitt, 2021, 2022). For a review see Beyeler et al. (2019).

Predictive coding, originally proposed by Rao and Ballard (1999), provides a predictive structure inspired by the fact that there is extensive feedback in the brain. Predictive coding assumes a hierarchical structure where units of the lower layer project to the upper layer via feedforward connection while units of the higher layer project back to the lower layer via feedback connection. The residual error, defined as the difference between the input to lower layer and feedback from upper layer, is computed and used as the response of the units on lower layer that project to upper layer. The feedback from higher layer represents the prediction the input and thus the residual error is often referred to as prediction error in predictive

coding. The essence of predictive coding is that the model constantly makes prediction of the current input, computes the mismatch between input and prediction, and only transmits that mismatch (prediction error) to the higher layer. The notion of only transmitting mismatch to the higher layer is similar to the principle of efficiency in sparse coding. Rao and Ballard (1999) showed that predictive coding, when used to model the visual area of the brain, can learn simple-cell-like RFs and display some non-classical RF properties such as end-stopping. Later on, Bayesian framework has also been connected with predictive coding, which proves to be an appealing model for many different brain areas (Knill and Pouget, 2004; Friston, 2010; Bastos et al., 2012).

Built upon these three significant models - sparse coding, predictive coding and divisive normalization, enormous computational and experimental studies have been extensively investigated to enhance our understanding of brain functions. However, a unified understanding of these three models is still lacking. Though these three models seem not to have a direct connection with each other, we propose that these three models can be related. In this paper, we aim to unify sparse coding, predictive coding and divisive normalization from the its first principles.

2 Methods

2.1 Sparse coding

Sparse coding finds an efficient representation of the sensory input using a linear combination of a set of basis vectors (Olshausen and Field, 1996, 1997). Denote $\mathbf{x} = [x_1, x_2, \cdots, x_L]^T$ as a $L \times 1$ vector that represents the sensory input such as visual or auditory stimuli that has been pre-processed upstream. Denote $\mathbf{s} = [s_1, s_2, \cdots, s_M]^T$ as a $M \times 1$ vector that represents the response of M model units and $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_M]$ as a $L \times M$ matrix that contains M basis vectors $\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_M$. Each basis vector, $\mathbf{a}_i$ $(i = 1, 2, \cdots, M)$ is a $L \times 1$ vector that represents one feature used to reconstruct the input such that the input, $\mathbf{x}$, can be represented by a linear combination of basis vectors weighted by model responses:

$$\mathbf{x} \approx \mathbf{A}\mathbf{s} = \sum_{i=1}^{M} s\, \mathbf{a}_i. \tag{1}$$

Sparse coding finds the linear representation (Eq. 1) using sparse responses, namely that only part of the model responses, s_i ($i = 1, 2, \cdots, M$), will be active. This can be formulated as an optimization problem of minimizing the following cost function:

$$C^{\mathrm{SC}}(\mathbf{A}, \mathbf{s}) = \frac{1}{2}\|\mathbf{x} - \mathbf{A}\mathbf{s}\|^2 + \lambda Q(\mathbf{s}), \tag{2}$$

where $Q(\mathbf{s}) = \sum_{i=1}^{M} Q(s_i)$, $Q(\cdot)$ is a function that penalizes large responses, and λ is a constant parameter that scales the penalty function. In other words, the first term on the right-hand-side of Eq. 2 represents the reconstruction error and the second term determines the sparsity of model units. Therefore, by minimizing $C^{\mathrm{SC}}(\mathbf{A}, \mathbf{s})$, one finds the linear reconstruction of the input using sparse responses. However, there is always a trade-off between the reconstruction fidelity and the response sparsity. Note that Eq. 2 is the same as Eq. 2 in Olshausen and Field (1996) and Eq. 14 in Olshausen and Field (1997).

Assuming that $Q(\cdot)$ is differentiable, given $\mathbf{A}$, the dynamics of $\mathbf{s}$ can be derived from taking the partial derivative of $C^{\mathrm{SC}}(\mathbf{A}, \mathbf{s})$:

$$\dot{\mathbf{s}} \propto -\frac{\partial C^{\mathrm{SC}}(\mathbf{A}, \mathbf{s})}{\partial \mathbf{s}} = \mathbf{A}^T(\mathbf{x} - \mathbf{A}\mathbf{s}) - \lambda Q'(\mathbf{s}), \tag{3}$$

where $Q'(\cdot)$ represents the derivative of $Q(\cdot)$ and $Q'(\mathbf{s}) = [Q'(s_1), Q'(s_2), \cdots, Q'(s_M)]^T$. Eq. 3 describes how, given input $\mathbf{x}$, an iterative procedure enables us to calculate the response of model units. The response $\mathbf{s}$ may or may not reach a stable equilibrium, depending on the properties of this dynamic system.

The learning rule of $\mathbf{A}$ can also be derived by taking the partial derivative of $C^{\mathrm{SC}}(\mathbf{A}, \mathbf{s})$:

$$\dot{\mathbf{A}} \propto -\frac{\partial C^{\mathrm{SC}}(\mathbf{A}, \mathbf{s})}{\partial \mathbf{A}} = (\mathbf{x} - \mathbf{A}\mathbf{s})\mathbf{s}^T. \tag{4}$$

Generally, when simulating sparse coding models, the learning rule in Eq. 4 is applied once only after some iterations of computing the model response according to the model dynamics described by Eq. 3, with the assumptions that (i) the time scale of learning / plasticity is longer than that of model responses, and (ii) the model response iterates to some stable value. The learning rule in Eq. 4 may cause basis vectors in $\mathbf{A}$ to grow without bound, so L2 norm normalization of each vector, $\mathbf{a}_i$, is used in the original work of sparse coding (Olshausen and Field, 1997) and some subsequent studies (Zhu and Rozell, 2013; Lian et al., 2019).

2.2 Predictive coding

Predictive coding, originally proposed by Rao and Ballard (1999), is a hierarchical model in which prediction errors are propagated to the higher levels of the hierarchy. Predictive coding can have multiple layers, but in this study we only look at a two-layer predictive coding model because it is the fundamental block and can relate to sparse coding and divisive normalization. Combining Eqs. 4 and 5 in Rao and Ballard (1999), the cost function of predictive coding can be written as

$$C^{\mathrm{PC}}(\mathbf{A}, \mathbf{s}) = \frac{1}{2}\|\mathbf{x} - f(\mathbf{A}\mathbf{s})\|^2 + \lambda Q(\mathbf{s}) + \beta W(\mathbf{A}), \tag{5}$$

where $f(\cdot)$ can be interpreted as the neuronal activation function, $f(\mathbf{A}\mathbf{s})$ is the prediction of the input, $Q(\mathbf{s})$ is the prior constraint on the model responses, and $W(\mathbf{A})$ puts some regularization on the basis vectors that are the columns of $\mathbf{A}$.

Similarly, the model dynamics and learning rule of predictive coding can be derived by taking the partial derivative of $C^{\mathrm{PC}}(\mathbf{A}, \mathbf{s})$:

$$\dot{\mathbf{s}} \propto -\frac{\partial C^{\mathrm{PC}}(\mathbf{A}, \mathbf{s})}{\partial \mathbf{s}} = \mathbf{A}^T \frac{\partial f^T(\mathbf{r})}{\partial \mathbf{r}}(\mathbf{x} - f(\mathbf{A}\mathbf{s})) - \lambda Q'(\mathbf{s}) \tag{6}$$

and

$$\dot{\mathbf{A}} \propto -\frac{\partial C^{\mathrm{PC}}(\mathbf{A}, \mathbf{s})}{\partial \mathbf{A}} = \frac{\partial f^T(\mathbf{r})}{\partial \mathbf{r}}(\mathbf{x} - f(\mathbf{A}\mathbf{s}))\mathbf{s}^T - \beta W'(\mathbf{A}), \tag{7}$$

where $\mathbf{r} = \mathbf{x} - f(\mathbf{A}\mathbf{s})$. The learning rule of predictive coding is generally applied after some iterations of computing the model responses.

2.3 Divisive normalization

Divisive normalization (DN) was first proposed by Heeger (1992) to account for the nonlinear physiological responses of neurons in V1. One variant of DN is given by

$$\begin{aligned} s_i^{\mathrm{DN}} &= \gamma \frac{(s_i^{\mathrm{L}})^n}{\rho^n + \sum_k (s_k^{\mathrm{L}})^n} \\ s_i^{\mathrm{L}} &= \max(\mathbf{a}_i^T \mathbf{x}, 0), \end{aligned} \tag{8}$$

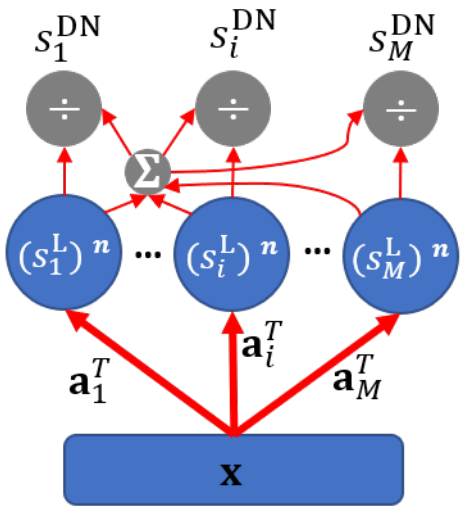

Figure 1: **Model structure of divisive normalization (DN).** The linear response is first generated by the ; some form of nonlinearity (exponential function with exponent n) is then applied; next, the response of each unit is divided by the sum of responses of all the units in the network.

where $\mathbf{x}$ is the sensory input, $\mathbf{a}_i$ represents the synaptic weights connecting input with neuron i, s_i^{L} is the linear response rectified by a threshold of zero to give nonnegative firing rates, s_i^{DN} is the response of neuron i after DN, and γ, ρ and n are the parameters that determine the shape of the response curve of DN. The model structure of DN is shown in Fig 1.

DN introduces nonlinearity into the model through the calculation of the neural responses in Eq.1, which also introduces competition between model units through the scaling of the response of each unit by the response of the whole population.

3 Results

3.1 Sparse/Predictive coding: sparse coding and predictive coding can be identical

Predictive coding is related with sparse coding, or efficient coding more broadly speaking, because only prediction error is fed to the higher level. Furthermore, from the first principles, sparse coding and predictive coding can be identical in a two-layer neural network.

In the original cost function of predictive coding (Eq. 5), $\beta W(\mathbf{A})$ puts some penalties on the strength of elements of $\mathbf{A}$ during the learning process and its derivative appears in the learning rule described by Eq. 7.

Rao and Ballard (1999) used the sum of squared response in the cost function that regulates L2 norm of learned basis vectors, which is essentially similar to L2 normalization used in sparse coding models during the learning process. Therefore, $\beta W(\mathbf{A})$ in Eq. 5 can be removed and $-\beta W'(\mathbf{A})$ in Eq. 7 can be replaced by a L2 normalization. Additionally, in the case where $f(\cdot)$ is linear function, $f(y) = y$, then the cost function (Eq. 5), model dynamics (Eq. 6) and learning rule (Eq. 7) of predictive coding become identical to those of sparse coding (Eqs. 2, 3 and 4). $f(y) = y$ was used to in end-stopping simulations in the work by Rao and Ballard (1999).

Above all, given that the weigh regularization term is replaced by L2 normalization and the feedback activation function is linear, sparse coding is identical to predictive coding. A unified framework with model diagram and equations is given below.

3.1.1 Model structure

The exposition above shows how sparse coding and predictive coding can be described in the same framework, although their original formulation had different goals, namely sparse coding is about the sparsity in the network model while predictive coding is concerned with the goal of the resulting neural processing. Here a unified framework for sparse coding and predictive coding is illustrated by implementing the principle of sparse coding in a two-layer model with the structure of predictive coding. We refer to this model as sparse/predictive coding (SPC) model, similar to Boutin et al. (2020). The structure of the SPC model is shown in Fig. 2.

The similarity between sparse coding (Eq. 3) and predictive coding (Eq. 6) enables the dynamics and learning rule of the SPC model to be described as a stage of *response iteration*:

$$\begin{aligned} \dot{\mathbf{s}} &= \eta_s(\mathbf{A}^T\mathbf{e} - H(\mathbf{s}))\ , \quad \mathbf{s} \geq \mathbf{0} \\ \mathbf{e} &= \mathbf{x} - \mathbf{A}\mathbf{s}, \end{aligned} \tag{9}$$

followed by the *learning step*:

$$\dot{\mathbf{A}} = \eta_a \mathbf{e}\mathbf{s}^T \ \text{ with } \ \|\mathbf{A}\|_2 = 1\ , \tag{10}$$

where $\mathbf{e}$ is the prediction error, $H(\mathbf{s}) = \lambda Q'(\mathbf{s}) = [H(s_1), H(s_1), \cdots, H(s_M)]^T$ is the mechanism that

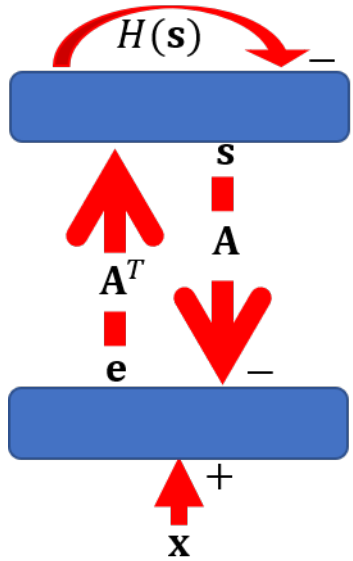

Figure 2: **Model structure of sparse/predictive coding (SPC).** The first layer computes the prediction error between the input and reconstruction of the model, and then sends error signals to the second layer. The second layer takes input from the first layer and incorporates the response-regulating mechanism, $H(\cdot)$. This two-layer network implements the dynamics described in Eq. 9.

regulates the responses of model units, η_s is the update rate when computing the responses, η_a is the learning rate of updating connection weights, and the strength of $\mathbf{A}$ is regulated by weight normalization ($\|\mathbf{A}\|_2 = 1$) at the end of each learning step, similar to previous studies (Olshausen and Field, 1997; Rozell et al., 2008; Zhu and Rozell, 2013; Lian et al., 2019). Note that model responses are constrained to be nonnegative here ($\mathbf{s} \geq \mathbf{0}$ in Eq. 9) to incorporate the nonnegativity of neural firing rate. However, the nonnegativity introduced here is known to keep the same expressiveness as standard sparse coding (Papyan et al., 2017).

The model responses are computed in response iteration using Eq. 9 and then the learning rule of SPC is applied in the learning step using Eq. 10. Note that Eq. 10 is a local Hebbian learning rule in this model structure. η_s needs to be chosen carefully to ensure reasonable convergence of the model responses (see Fig. 8 for an example). η_a also needs to be chosen such that $\mathbf{A}$ can learn meaningful structure at a reasonable speed.

When this SPC model is related to the neural circuits of the brain, neurons in the first layer compute the prediction error and feed the error signals to the second layer, where the responses are computed according to the dyanmics in Eq. 9. $\mathbf{A}^T$ and $\mathbf{A}$ can be interpreted as the feedforward and feedback synaptic connections

between two layers. This structure was used by Lian et al. (2019) to implement sparse coding and explain numerous experimental phenomena observed in the LGN-V1 pathway.

An alternative way of implementing sparse coding in a feedforward network is through inhibitory lateral connections, which promote competition that eventually leads to sparse responses (Rozell et al., 2008; Zylberberg et al., 2011; Brito and Gerstner, 2016), and the lateral connections are supposed to decorrelate responses across the network such that the model can learn different features. However, the mathematical equations underlying different implementations of sparse coding are essentially the same. Therefore, the framework of SPC proposed here can unify a range of different models related with sparse coding and predictive coding.

3.1.2 Homeostasis function

In the cost functions of sparse coding (Eq. 2) and predictive coding (Eq. 5), $Q(\cdot)$ is the sparsity term that promotes sparse response of the model units and its derivative is denoted as $H(\cdot)$, as appeared in the SPC model (Eq. 9). Since the model responses, $\mathbf{s}$, represent the firing rates of model neurons, $\mathbf{s}$ should be nonnegative, as described in Eq. 9. Given the nonnegativity of any model unit response s, $H(s)$ in Eq. 9 typically returns a nonnegative value that can be interpreted as the penalty or cost of a model neuron generating neural responses. Consequently, $H(\cdot)$ can be understood as a homeostasis function that regulates neuronal responses.

In the original papers of sparse coding (Olshausen and Field, 1997) and predictive coding (Rao and Ballard, 1999), the choices of $Q(s)$ considered were

$$Q(s) = \ln\left[1 + \left(\frac{s}{\sigma}\right)^2\right] \ , \ \ -e^{-\left(\frac{s}{\sigma}\right)^2} \ , \ |s| \ \text{ or } \ s^2 \ , \tag{11}$$

where σ is a positive scaling factor. With the assumption of nonnegative responses, consider now three different functions for $Q(s)$:

$$Q(s) = -e^{-(\frac{s}{\sigma})^2} \ , \ \ \ln[1 + (\frac{s}{\sigma})^2] \ \text{ or } \ s^{\alpha}, \tag{12}$$

where α is a positive power index. The corresponding homeostasis functions are $\lambda Q'(s)$, namely:

$$H(s) = \lambda \frac{s}{1 + (s/\sigma)^2} \ , \ \ \lambda s e^{-(\frac{s}{\sigma})^2} \ \text{ or } \ \lambda s^{\alpha - 1} \quad (\text{with } s \geq 0) \ , \tag{13}$$

where the constant parameters are absorbed into the single constant parameter, λ.

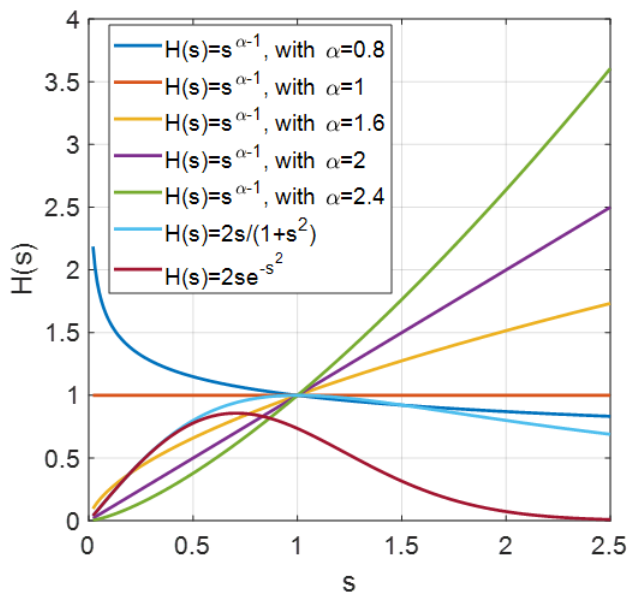

Figure 3: **Examples of homeostasis functions.** These functions are examples of Eq. 13. λ is set to 1 for power index functions. For the remaining functions, λ is set to 2 and σ is set to 1 so that all functions can be plotted in the same range.

Some examples of homeostasis functions, $H(\cdot)$, are shown in Fig. 3. For power functions, when $H(s) = 1$ ($\alpha = 1$), the cost the homeostasis puts on model neurons is the same for different firing rates; when α is smaller than 1, the cost of generating small firing rates is extremely high (infinite cost for zero firing rate) and the cost decreases dramatically for larger firing rates; when α is larger than 1, the cost of generating firing rates starts from 0 and increases without bound as the firing rate increases. For $H(s) = 2s/(1 + s^2)$ and $H(s) = 2se^{-s^2}$, the cost starts from zero, reaches the maximum and then reduces to zero as the firing rate increases. Overall, the choice of $H(\cdot)$ is crucial because it partly determines the dynamics of the model, as described by Eq. 9.

3.1.3 Stability analysis

The dynamics of SPC model described in Eq. 9 can be reformulated into one equation:

$$\dot{\mathbf{s}} = \eta_s \left[\mathbf{A}^T \left(\mathbf{x} - \mathbf{A}\mathbf{s} \right) - H(\mathbf{s}) \right]. \tag{14}$$

The equilibrium of this differential equation is reached when $\dot{\mathbf{s}} = \mathbb{0}$, namely when

$$\mathbf{A}^T(\mathbf{x} - \mathbf{A}\mathbf{s}) - H(\mathbf{s}) = \mathbb{0} = \mathbf{A}^T\mathbf{x} - \mathbf{A}^T\mathbf{A}\mathbf{s} - H(\mathbf{s}), \tag{15}$$

where $\mathbf{s}^{\mathrm{L}} := \mathbf{A}^T\mathbf{x}$ is the linear response given the feedforward connection, $\mathbf{A}$.

If there is no homeostasis (i.e., $H(s) = 0$), the above equation becomes

$$\mathbf{A}^T\mathbf{A}\mathbf{s} = \mathbf{A}^T\mathbf{x}. \tag{16}$$

Since $\mathbf{A}^T\mathbf{A}$ is positive semi-definite, $\mathbf{A}^T\mathbf{A}$ will be invertible (i.e., positive definite) if we assume $\mathbf{A}\mathbf{z} \neq \mathbf{0}$ for any vector $\mathbf{z} \neq \mathbf{0}$. This condition is generally met for sensory system in the brain, since there will be neurons responding to a non-zero input stimulus. The equilibrium, $\mathbf{s}^{\mathrm{EQ}}$ can then be written as

$$\mathbf{s}^{\mathrm{EQ}} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{x}. \tag{17}$$

The equilibrium of SPC model with no homeostasis is still linear, but the equivalent feedforward connection becomes $(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$. If $\mathbf{A}^T\mathbf{A} = \mathbb{1}$, the equilibrium of the SPC model will be identical to the response of a linear feedforward model. In addition, the amplitude of the model responses will increase linearly with the amplitude of the input. Given that the dynamic equations Eq. 14 describe a first order differential equation,

the model responses, $\mathbf{s}$, will approach the equilibrium, $\mathbf{s}^{\mathrm{EQ}}$, as

$$\mathbf{s}(t) = [\mathbf{v}_1\ \mathbf{v}_2\ \cdots \mathbf{v}_M] \begin{bmatrix} \mathbf{v}_1^T \mathbf{s}(0) e^{\lambda_1 t} \\ \mathbf{v}_2^T \mathbf{s}(0) e^{\lambda_2 t} \\ \vdots \\ \mathbf{v}_M^T \mathbf{s}(0) e^{\lambda_M t} \end{bmatrix} + \mathbf{s}^{\mathrm{EQ}} , \tag{18}$$

where $\mathbf{v}_i$ $(i = 1, 2, \cdots, M)$ are the eigenvectors and λ_i $(i = 1, 2, \cdots, M)$ are the corresponding eigenvalues of $-\mathbf{A}^T\mathbf{A}$, and $\mathbf{s}(0)$ is the initial response at time 0. Because $-\mathbf{A}^T\mathbf{A}$ is negative definite with the assumptions above, λ_i $(i = 1, 2, \cdots, M)$ will be all negative, meaning that the first term on the right hand side of Eq. 18 will reduce to zero exponentially.

Figs. 4 & 5 show the simulation results to validate the analysis above when there is no homeostasis in the model. In the simulation there are 256 units in the first layer ($L = 256$) and 256 units in the second layer ($M = 256$). Therefore $\mathbf{A}$ is a 256×256 matrix and each column is a basis vector used to reconstruct the input image. The input is a 16×16 image of sinusoidal grating (Fig. 4B). There are 3000 iterations of Eq. 9 to compute the model responses with $\eta_s = 0.05$ and there is no learning process of $\mathbf{A}$ in this simulation. $\mathbf{A}$ is chosen differently for Figs. 4 and 5.

The connection $\mathbf{A}$ in Fig. 4 is chosen as

$$\mathbf{A} = \mathbb{1} + \mathbb{N} \ \text{ with } \ \|\mathbf{A}\|_2 = 1 , \tag{19}$$

where $\mathbb{1}$ is an 256×256 identity matrix and $\mathbb{N}$ is an 256×256 matrix whose elements are sampled from a Gaussian distribution with mean 0 and standard deviation 0.05. $\mathbf{A}$ is then normalized to have l_2 norm 1. $\mathbf{A}$ is chosen this way to ensure that $\mathbf{A}^T\mathbf{A}$ is a full-rank matrix and the eigenvalues of $\mathbf{A}^T\mathbf{A}$ are all different but separated from 0. Fig. 4A displays the 256 basis vectors and these vectors are used to reconstruct the input image (Fig. 4B). Fig. 4C shows that the model response becomes stable within 3000 iterations. The analytical equilibrium $\mathbf{s}^{\mathrm{rm}}$ (Eq. 17) is plotted as red spikes and the model response $\mathbf{s}^{\mathrm{SP}}$ after 3000 iterations is plotted in blue spikes in Fig. 4D. Fig. 4D shows that the model response is identical to the analytical equilibrium. The summed squared error between $\mathbf{s}^{\mathrm{EQ}}$ and $\mathbf{s}^{\mathrm{SP}}$ is only 0.0036% and the summed squared

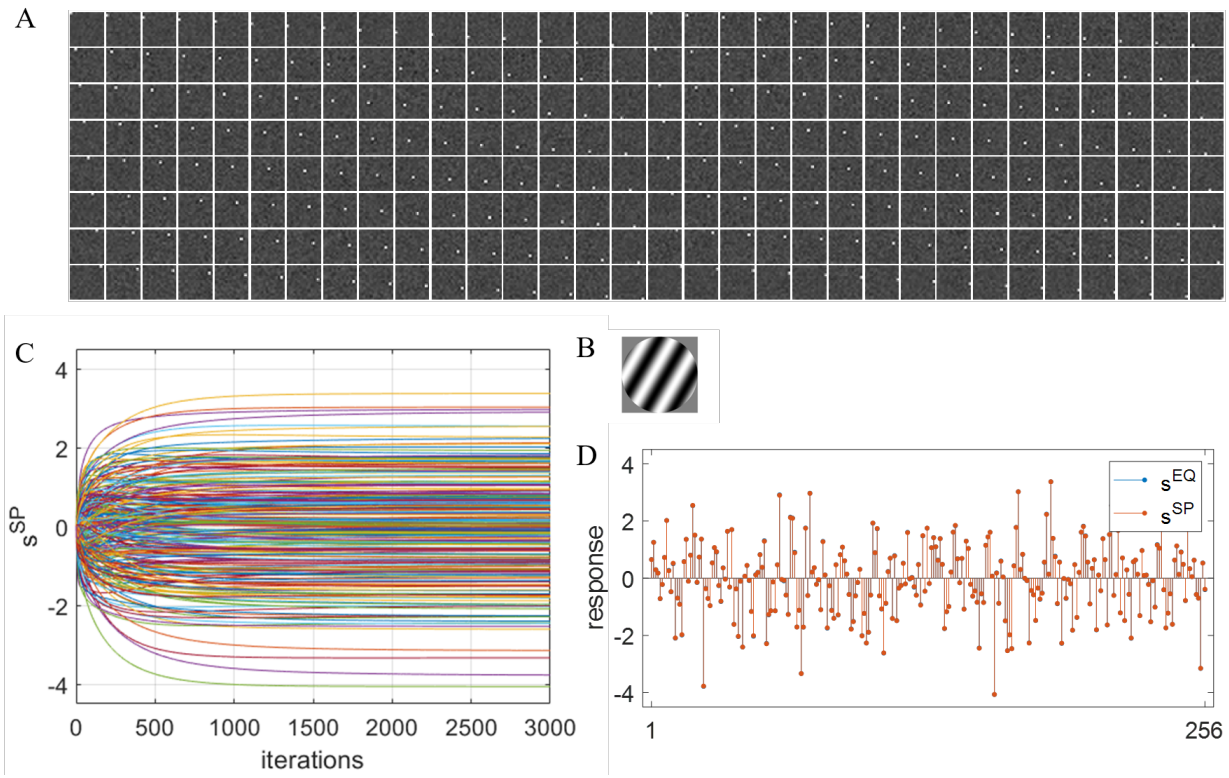

Figure 4: **Model response converges to the analytical equilibrium.** There is no homeostasis in this simulation. (A) Basis vectors of $\mathbf{A}$. Each square block is a 16×16 image. $\mathbf{A}$ is defined in Eq 19 and close to an identity matrix, so each basis vector has one dominant element. (B) The 16×16 input image. (C) Response trajectory. (D) The response of analytical equilibrium and response of SPC model after 3000 iterations of computing model responses with $\eta_s = 0.05$.

error between input ($\mathbf{x}$) and reconstruction ($\mathbf{As}$) is 0. In other words, given the current input and fixed connection, the model can essentially represent the input perfectly and the model response converges to the analytical value.

Although the model has an analytical equilibrium described by Eq. 17, it does not guarantee that model response will converge to the analytical equilibrium within any reasonable time. On the contrary, the iterated model response might be far away from the equilibrium while the model response can still reconstruct the input very well, which is illustrated in Fig. 5.

In Fig. 5, the connection weights $\mathbf{A}$ are chosen to be some Gabor-like filters that are learnt by a standard sparse coding model where there is no nonnegative constraint on the model response (Olshausen and Field, 1996, 1997; Rozell et al., 2008) and $\|\mathbf{A}\|_2 = 1$. Fig. 5A shows the 256 basis filters of $\mathbf{A}$, which look similar

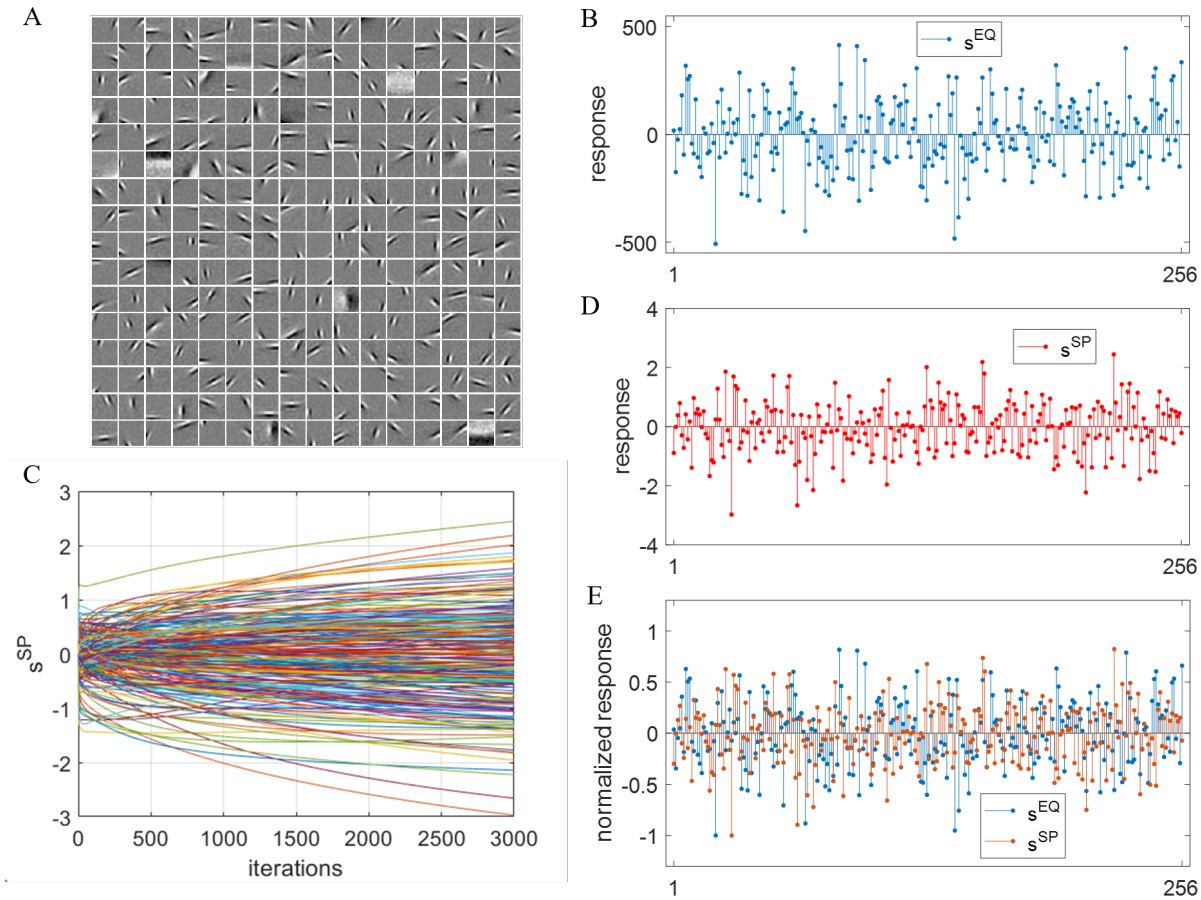

Figure 5: **Model response does not converge to the analytical equilibrium.** There is no homeostasis in this simulation and the input to the model is the same as Fig. 4B. (A) Basis vectors of $\mathbf{A}$. Each square block is a 16×16 image. $\mathbf{A}$ is learned using a standard sparse coding model and each basis filter is similar to simple cells in V1. (B) $\mathbf{s}^{\mathrm{EQ}}$, the analytical equilibrium determined by Eq. 17. (C) Response trajectory. (D) Model response, $\mathbf{s}^{\mathrm{SP}}$, after 3000 iterations. (E) The normalized response of analytical equilibrium and model response after 3000 iterations with $\eta_s = 0.05$.

to simple cells found in the visual cortex (Hubel and Wiesel, 1959). Though $\mathbf{A}^T\mathbf{A}$ is invertible with all positive eigenvalues, some eigenvalues are very small (the smallest is 4×10^{-10}). Therefore, the analytical response described by Eq. 17 may become very large. This is illustrated in Fig. 5B, which shows that the analytical equilibrium is quite large and the amplitude in this example can grow as large as 500. Fig. 5C shows the trajectory of model response over these 3000 iterations of implementing model dynamics with $\eta_s = 0.05$ (the same as Fig. 4). However, it appears that model responses do not converge after 3000 iterations. The model responses after 3000 iterations are plotted in Fig. 5D. Compared with the analytical

equilibrium shown in Fig. 5B, the amplitude of model response is much smaller than the analytical equilibrium. Apart from the different amplitudes of responses, the analytical equilibrium and model response also differ in the population responses, which are illustrated in Fig. 5E, that shows that the responses are very different even after normalizing the responses to the same scale. The summed squared error between the equilibrium and model responses is 99.99%. The difference between analytical and simulated results may be caused by the discrete implementation of the model and the very small eigenvalues of $\mathbf{A}^T\mathbf{A}$, which takes much longer for the model response to converge, as described by Eq. 18. Despite the significant difference between the analytical equilibrium and model simulation response, the model can still reconstruct the input well, with a small summed squared error of 0.8% and the average energy of the model response (mean of the squared response) is 0.77. However, if there are only 300 iterations, the reconstruction error of the model is 1.5% and the average energy is 0.21. Therefore, there seems to be a trade-off between the model accuracy and computation cost (including both time and energy). In practice, although $\mathbf{A}^T\mathbf{A}$ is mostly invertible, some of the eigenvalues can easily be very small, as illustrated in Fig. 5. Also, if the standard deviation of $\mathbb{N}$ in Eq. 19 becomes 0.1, the smallest eigenvalues of $\mathbf{A}^T\mathbf{A}$ would be 9.9×10^{-7}. Therefore, though the discrete online implementation of the model is not as accurate as the analytical result, it nevertheless provides a reasonable response within a reasonable time.

If there is a homeostasis term (i.e., $H(s) \neq 0$), the analysis is somewhat more difficult, but it is still possible to give some analytical results for special choices of $H(s)$. If $H(s) = \lambda s$ (i.e., $Q(s) = s^2$), the dynamics of the SPC model, described by Eq. 14, becomes reformulated into one equation:

$$\dot{\mathbf{s}} = \eta_s \left[\mathbf{A}^T(\mathbf{x} - \mathbf{A}\mathbf{s}) - \lambda\mathbf{s}\right] = \eta_s \left[\mathbf{A}^T\mathbf{x} - (\mathbf{A}^T\mathbf{A} - \lambda\mathbb{1})\mathbf{s}\right] \ , \tag{20}$$

and the equilibrium of this differential equation is reached when

$$(\mathbf{A}^T\mathbf{A} - \lambda\mathbb{1})\mathbf{s} = \mathbf{A}^T\mathbf{x} \ . \tag{21}$$

If $\mathbf{A}^T\mathbf{A} - \lambda\mathbb{1}$ is invertible then the equilibrium responses are

$$\mathbf{s}^{\mathrm{EQ}} = (\mathbf{A}^T\mathbf{A} - \lambda\mathbb{1})^{-1}\mathbf{A}^T\mathbf{x} \ , \tag{22}$$

and the amplitude is also linearly dependent on the amplitude of the input. If $H(s)$ takes a more complex form, the equilibrium of the model is generally more difficult to calculate analytically. Additionally, $H(s)$ may alter the linear dependence of model responses on the input (Eq. 17) and potentially introduce some nonlinearity into the model dynamics.

Note that the above analysis assumes a linear system and there is no constraint of nonnegativity for $\mathbf{s}$. If $\mathbf{s} \geq \mathbf{0}$ is incorporated in the above analysis, it generally becomes intractable.

3.1.4 Nonlinearity

The nonlinearity introduced into the SPC model is two-fold: through the nonnegativity of the firing rates and through the homeostasis function.

As analyzed in the previous section, the equilibrium of the SPC model is linearly dependent on the input when there is no constraint of nonnegative firing rate and homeostasis (Eq. 17), given that $\mathbf{A}^T\mathbf{A}$ is invertible. However, $\mathbf{s} \geq \mathbf{0}$ has a dramatic effect on the network, because this constraint introduces great nonlinearity. Somewhat surprisingly, the nonnegativity constraint accelerates the convergence of the model responses and it becomes stable after 1500 iterations (Fig. 6A) compared with the case without the nonnegativity constraint (Fig. 5C). However, because of the nonnegativity of $\mathbf{s}$, the model cannot reconstructs the input with negative responses though the basis vectors $\mathbf{A}$ in Fig. 5 are learned in a standard sparse coding that allows model responses to be negative, which leads to a larger reconstruction error of 11.1% after incorporating $\mathbf{s} \geq \mathbf{0}$. Fig. 6B illustrates the model responses after 3000 iterations when $\mathbf{s} \geq \mathbf{0}$ is incorporated and it shows a clear difference from the analytical response ($\mathbf{s}^{\mathrm{EQ}}$ in Fig. 5B) and the model responses without the nonnegativity constraint (Fig. 5D). After the nonnegativity of model response is incorporated, one way to preserve the expressiveness of the basis vectors is to extend the basis vectors to be $[\mathbf{A} \quad -\mathbf{A}]$; i.e., similar to the study by Zhu and Rozell (2013). Therefore, there are 512 model units, instead of 256 model units in this case. Figs. 6C & D show that the convergence of model units is lost again and more model units are active, which leads to a better reconstruction of the input with 0.8% error.

Apart from the nonnegativity of the model response, the homeostasis, $H(s)$, also introduces nonlinearity into the model, since the system becomes nonlinear if $H(s)$ is not a first-order polynomial of s. Even

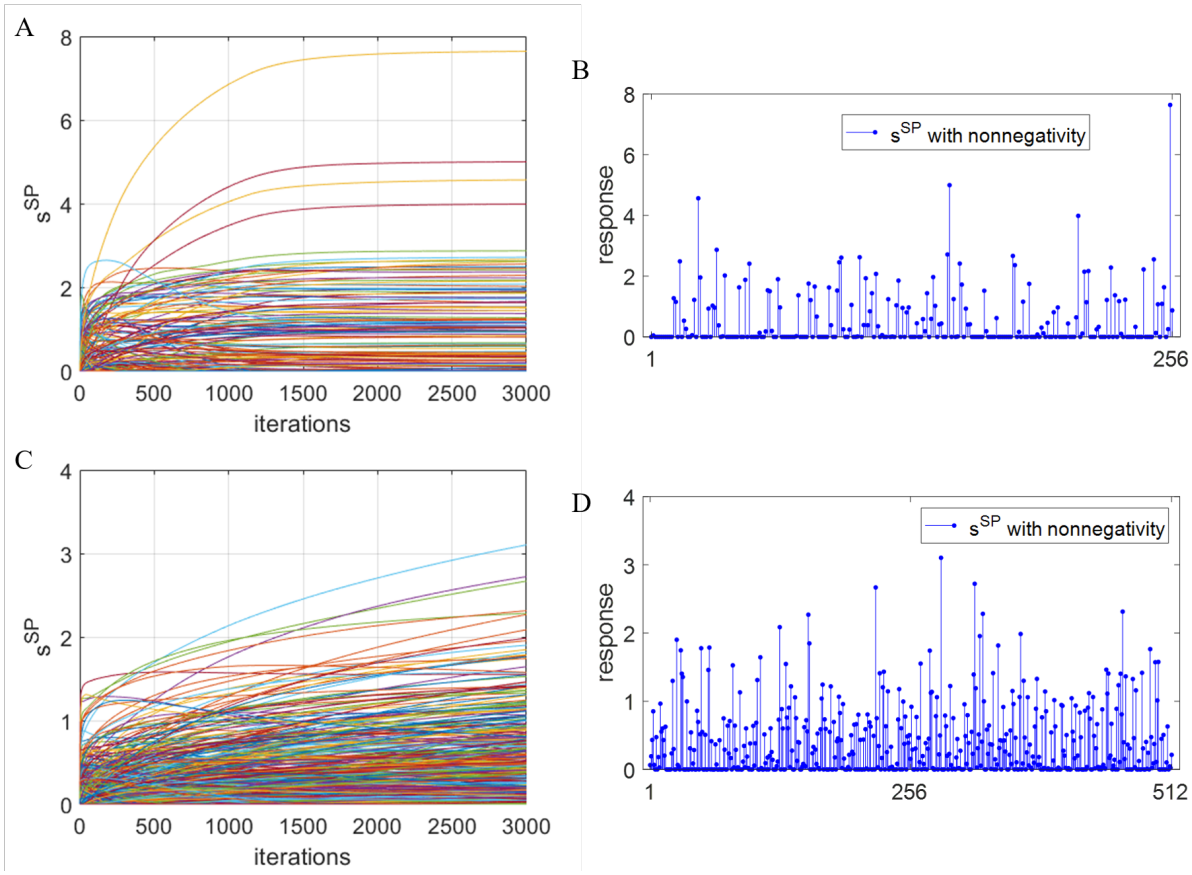

Figure 6: **Model responses with nonnegative constraint.** The input to the model is the same as Fig. 4B. (A, C) Response trajectory. (B, D) $\mathbf{s}^{\mathrm{SP}}$, after 3000 iterations. For (A) and (B), the same basis vectors $\mathbf{A}$ as in Fig. 5A are used. For (C) and (D), the basis vectors are extend to be $[\mathbf{A} \ \ -\mathbf{A}]$ and the number of model units also doubles, so there are 512 model units in (C) and (D).

when there is no nonnegative constraint on the model response, the choice of $H(s)$ can determine the model responses and ultimately determines whether the model can learn structures from the input. For example, only suitable choices of sparsity functions can lead to Gabor-like simple cells after learning (Olshausen and Field, 1996, 1997; Rao and Ballard, 1999). We will later show in the paper that, a suitable choice of $H(\cdot)$ can bring significant nonlinearity into the SPC model, which is similar to the nonlinearity brought by DN.

3.1.5 Competition

SPC model incorporates competition into the neural network via the homeostasis function and the model structure.

The homeostasis function, $H(\cdot)$ in Eq. 9, introduces competition because it directly regulates the response of model units and it can put different penalties on different units based on their response level.

Apart from the homeostasis function, the structure of SPC model itself (Fig. 2) has a competitive nature because it explicitly implements a form of competition, as demonstrated by Spratling (2008). Suppose there is no homeostasis (i.e., $H(s) = 0$) in the dynamic equation of SPC, Eq. 9 essentially becomes the implementation of Harpur's "negative feedback network" (Harpur and Prager, 1996) that causes competition between model units such that they compete to represent the input. That competition can also be seen from the analytical equilibrium described in Eq. 17, where $(\mathbf{A}^T\mathbf{A})^{-1}$ will always scale the linear response $(\mathbf{A}^T\mathbf{x})$ based on the correlation between different basis vectors. Lian et al. (2021) show that the competition incorporated by this structure can be very strong such that learning complex cells from simple cells can be very challenging.

Because of this competitive nature of SPC model, sparse coding has been used to explain some nonclassical receptive field properties of V1 such as end-stopping, contrast invariance of orientation tuning (Zhu and Rozell, 2013). However, this study failed to explain contrast saturation, which is another important property of V1 neurons.

3.2 SPC and DN can be identical for the single-neuron scenario

Divisive normalization (DN) is fundamentally different from sparse/predictive coding (SPC). Unlike SPC, DN is a physiological model that was introduced to explain neural response properties in the brain, rather than a learning model with neural dynamics and synaptic plasticity. Specifically, for the model responses, SPC applies an iterative procedure described by Eq. 9, while the response of DN can be directly calculated by Eq. 8; for the learning rule, SPC incorporates learning and can describe how some brain properties can emerge via learning (Olshausen and Field, 1996, 1997; Zhu and Rozell, 2013; Lian et al., 2019), while DN has no learning rule and can only be used for explaining physiological data. However, we show that the response of DN can be identical to SPC in one of the simplest scenarios.

Suppose there is only one neuron in the first layer and one neuron in the second layer for both SPC model and DN, that the input is nonnegative, and the connection weight is 1. The model of DN computes the

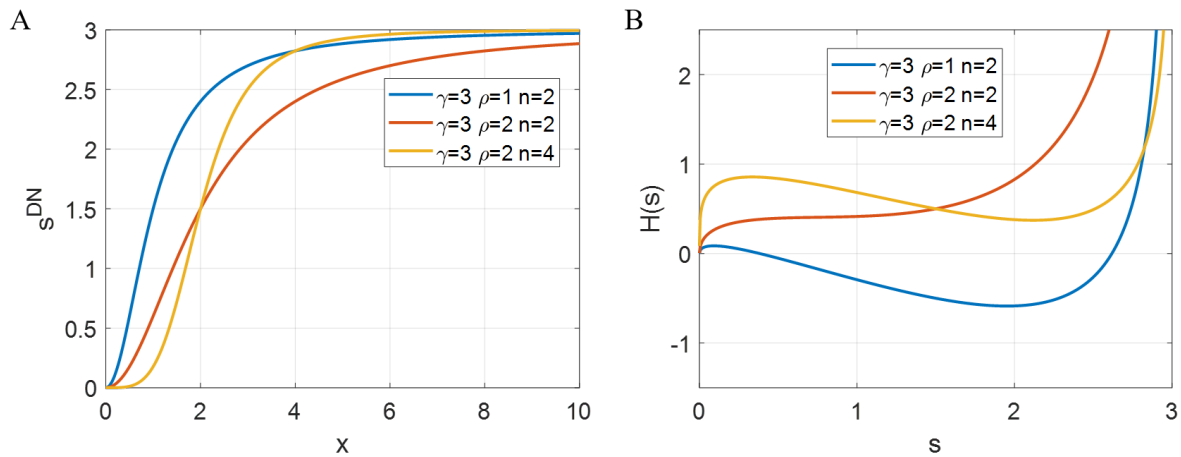

Figure 7: **Divisive normalization (DN) can be identical to sparse/predictive coding (SPC) for single-neuron scenario.** (A) DN described by Eq. 23. (B) Homeostasis function of the equivalent SPC model.

response as

$$s^{\mathrm{DN}} = \gamma \frac{x^n}{\rho^n + x^n} \ , \tag{23}$$

where γ determines the saturation value for the response, ρ determines the point where the response is $\gamma/2$, and n determines the curvature of the normalization curve. Some examples of DN described by Eq. 23 are shown in Fig. 7A.

The dynamics of the SPC model becomes

$$\dot{s}^{\mathrm{SP}} = \eta_s \left[x - s^{\mathrm{SP}} - H(s^{\mathrm{SP}}) \right] \ . \tag{24}$$

Then the equilibrium of the output neuron, s^{SP}, in the SPC model can be computed from the implicit relationship

$$x - s^{\mathrm{SP}} - H(s^{\mathrm{SP}}) = 0. \tag{25}$$

From Eq. 23

$$x = \rho \left(\frac{\gamma}{s^{\mathrm{DN}}} - 1 \right)^{-\frac{1}{n}} . \tag{26}$$

If DN and SPC give the same nonlinearity for this scenario (i.e., $s^{\mathrm{DN}} = s^{\mathrm{SP}}$), the homeostasis function in

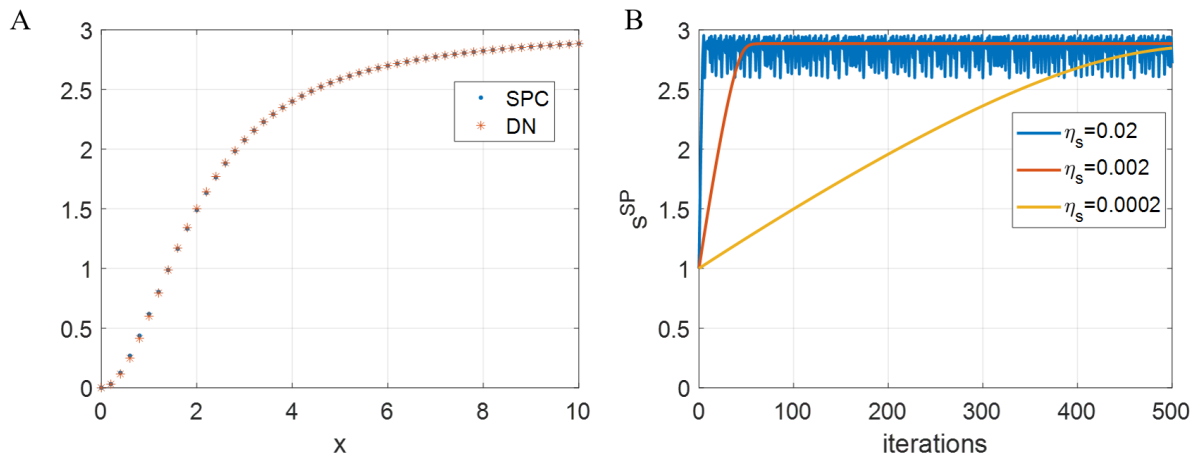

Figure 8: **Divisive normalization (DN) and sparse/predictive coding (SPC) in the single-neuron scenario.** (A) Responses of DN and SPC with $\eta_s = 0.002$ given different input. (B) Response of SPC vs. number of iterations when the input, x, is 10 with different values of η_s.

SPC can be derived from Eqs. 25 & 26:

$$H(s) = x - s = \rho \left(\frac{\gamma}{s} - 1 \right)^{-\frac{1}{n}} - s \ . \tag{27}$$

Some homeostasis functions of SPC described by Eq. 27 are shown in Fig. 7B. When s is small, $H(s)$ has a small cost on responses, though it displays various shapes for different sets of parameters. When s is approaches 3 (the value of ρ), $H(s)$ has a very high cost that prevents the response from exceeding the saturation. Therefore, for a single neuron scenario, the homeostasis function in SPC can generate response nonlinearity as well as response saturation.

Note that the homeostasis function (Eq. 27) is derived from Eq. 23 where s^{DN} is always smaller than γ. Therefore, $H(s)$ in Eq. 27 only applies to values of s that are smaller than the saturation value γ.

The simplest scenario described above was implemented with $\gamma = 3$, $\rho = 2$ and $n = 2$. DN is implemented using Eq. 23. SPC is implemented by an iterative procedure described by Eq. 24 with $H(s)$ given by Eq. 27 and 500 iterations to computing the response. Simulation results are shown in Fig. 8.

SPC and DN can generate equivalent results in this scenario with a suitable choice of homeostasis function of SPC, as shown in Fig. 8A. When the input is 10, the response of the SPC model vs. number of iterations

is plotted in Fig. 8. Fig. 8B shows that the response of SPC gradually evolves to the equilibrium and its dynamics depends on the value of η_s. s^{SP} start with with a value of 1 because of it is initialized as 1 prior to the iterative process. When computing the response of the SPC model, the value of η_s needs to be chosen carefully. If η_s is too small, it takes more iterations to reach the equilibrium (yellow line in Fig. 8B). However, if η_s is too large, s^{SP} may become oscillatory (blue line in Fig. 8B). Additionally, s^{SP} may go beyond the saturation value, γ, which causes the SPC model to break down because $H(s)$ described by Eq. 27 is well-defined only for $s < \gamma$.

The analysis of this simple scenario illustrates that although SPC and DN are different in terms of their model dynamics and model structure, they can nevertheless generate equivalent network activity given a suitable choice of the homeostasis function in the SPC model. However, the homeostasis function in SPC has a rigid assumption on the response, namely that the response has to be always smaller than γ during the iterative procedure of computing responses. When the network has more complexity as the number of neurons increases, this assumption can easily be violated.

3.3 SPC can learn Gabor-like simple cells with contrast saturation previously explained by DN

One notable difference between DN and SPC is the learning ability: DN is only a physiological model while SPC model has a learning process; i.e., SPC model can learn important features given a suitable choice of homeostasis function (Olshausen and Field, 1996, 1997; Rao and Ballard, 1999) while DN cannot. However, DN, as a physiological model, can account for many nonlinear properties because of its design of the mathematical equation. As demonstrated by Zhu and Rozell (2013), sparse coding can be used to explain a number of nonlinear receptive field properties of V1 cells. However, a classical nonlinear property, contrast saturation of V1 cells (Ohzawa et al., 1982, 1985; Freeman et al., 2002), can not be captured in the sparse coding model of Zhu and Rozell (2013), while it can easily be captured by DN Heeger (1992). In a simple scenario in the section above, DN can be equivalent to SPC with suitable choice of the homeostasis function, $H(s)$ (Eq. 27 and Fig. 7B), which suggests that SPC can account for saturation at least for a single-neuron scenario (Fig. 8A). Saturation in the SPC model is built in the homeostasis function, $H(s)$ (Eq. 27), that has a very large penalty (infinity in this case) at the saturation point.

In this section, we show that SPC with a novel homeostasis function can learn Gabor-like simple cells as well as displaying the property of contrast saturation that is previously explained by DN.

3.3.1 A novel homeostasis function for SPC

When using sparse coding to learn Gabor-like features that are similar to simple cells in V1 (Olshausen and Field, 1996, 1997), homeostasis functions were chosen to be

$$H(s) = \lambda \frac{s}{1 + (s/\sigma)^2}\ ,\ \ \lambda s e^{-(\frac{s}{\sigma})^2} \ \text{ or } \ \lambda s^{\alpha-1} \quad (\text{with } s \geq 0)$$

where $\alpha = 1$ (Olshausen and Field, 1996, 1997; Zhu and Rozell, 2013; Lian et al., 2019). $H(s) = \lambda s^{\alpha-1}$ with $\alpha = 2$ is used by Rao and Ballard (1999), but the model does not Gabor-like receptive field. When $H(s)$ takes the form of $\lambda s^{\alpha-1}$ (with $s \geq 0$), from our simulation, we find that $1 \leq \alpha < 2$ is necessary to enable the model to learn Gabor-like receptive field.

After examining the homeostasis functions that can learn Gabor-like receptive fields for SPC models, one thing in common is that the homeostasis function is concave starting from 0. In order to introduce response saturation into the homeostasis function, the homeostasis function should approach to infinity when the input keeps increasing and a convex curvature of the function is preferred. Therefore, we propose a homeostasis function for SPC model described as follows:

$$H(s) = \begin{cases} \lambda \frac{2s/x_0}{1+(s/x_0)^2}, & s \leq x_0 \\ \lambda e^{(\frac{s-x_0}{\sigma})^n}, & s > x_0 \end{cases} \tag{28}$$

where $x_0 > 0$ is the turning point where the $H(s)$ becomes convex instead of concave, σ determines the steepness of $H(s)$ when s is larger than x_0, and n is the index that determines the steepness of the curve when s is large. An example of $H(s)$ given by Eq. 28 is shown in Fig. 9. This homeostasis function has a similar shape to the corresponding homeostasis function of DN in the single-neuron scenario (Eq. 27 and Fig. 7B).

The proposed homeostasis function, $H(s)$, in Eq. 28 has the following properties: (1) it is concave when

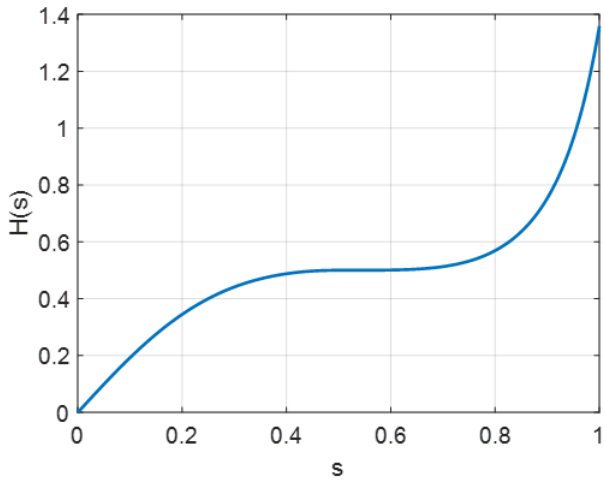

Figure 9: **The proposed homeostasis function described by Eq. 28.** In this example, parameters are chosen as follow: $\lambda = 0.5$, $x_0 = 0.5$, $\sigma = 0.5$ and $n = 4$.

$s < x_0$ and hopefully this will lead to the emergence of features after learning; (2) it is convex when $s > x_0$; (3) the derivative at $s = x_0$ is 0 such that the transition between concave and convex is smooth and continuous; (4) it approaches to large values fairly quickly when s is large and hopefully this will lead to response saturation.

3.3.2 The model can learn Gabor-like simple cells

The SPC model can learn Gabor-like simple cells when $H(s)$ described in Fig. 9 is used in the model dynamics (Eq. 9) and learning rule (Eq. 10). For training, the input is 16×16 image patches of pre-whitened natural images (Olshausen and Field, 1997), the model has 512 model units and model response are constrained to be nonnegative. After the learning process, the model learns Gabor-like features like simple cells in v1 (shown in Fig. 10), similar to previous modelling studies based on sparse coding (Olshausen and Field, 1996, 1997; Zylberberg et al., 2011; Zhu and Rozell, 2013; Lian et al., 2019). This shows that SPC with a suitable homeostasis function can learn features and it is not confined to a specific function as used in previous models.

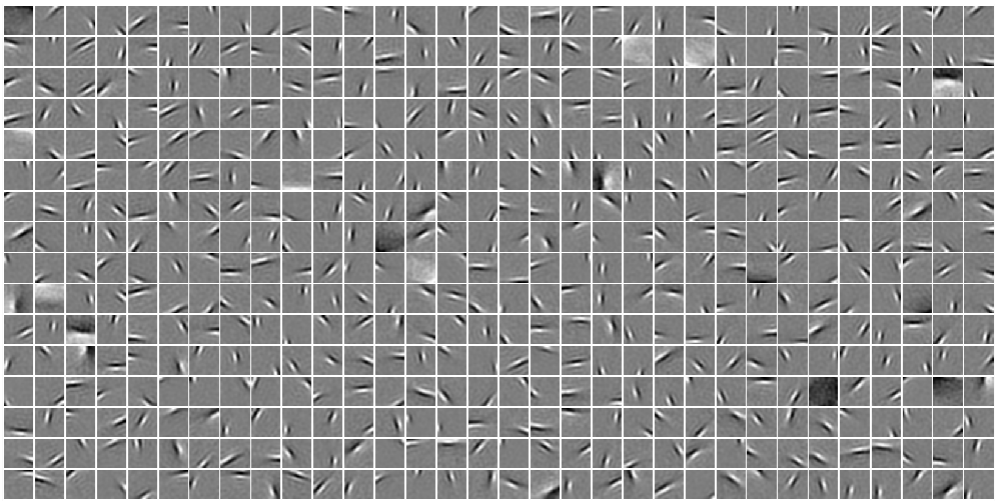

Figure 10: **Sparse/Predictive coding (SPC) model can learn Gabor-like features.** There are 512 model units and each square is a 16×16 image that shows the corresponding learnt feature. The homeostasis function used here is defined in Eq. 28.

3.3.3 The response of model units shows contrast saturation that has not been observed in previous sparse coding models

After learning, we test the response vs. contrast curve to investigate the response nonlinearity of the model using the above proposed homeostasis function. Fig. 11 shows that SPC with the homeostasis function (Eq. 28) can not only learn Gabor-like features but also display contrast saturation.

For the learnt 512 simple cells, we exclude cells whose Gabor-like receptive fields are not well within the 16×16, which leave us 228 learnt cells for further analysis. Next, we perform a exhaustive search for each cell to find the preferred sinusoidal grating with preferred spatial frequency, orientation and spatial phase that gives the cell maximal response. Then we change the contrast (amplitude of the preferred sinusoidal grating) and collect the response of the cell to obtain its response vs. contrast curve. The response vs. contrast curve of each cell is then fitted by the equation of single-neuron DN (Eq. 23), similar to previous studies (Heeger, 1992; Busse et al., 2009). Fig. 11A shows the response vs. contrast curve of one example unit of the SPC model with homeostasis function described in Fig. 9, which shows response saturation and the response curve can well be fitted by the equation of single-neuron DN (Eq.23). Fig. 11B shows response vs. contrast curves of 228 out of 512 learnt simple cells whose Gabor-like receptive fields are well within

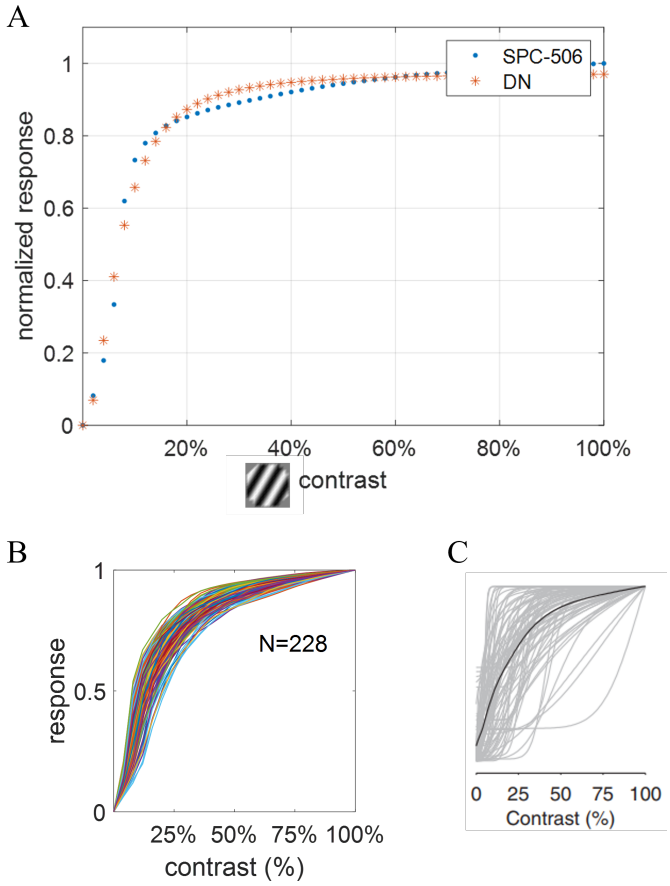

Figure 11: **Sparse/Predictive coding (SPC) model can generate contrast saturation similar to divisive normalization (DN).** (A) 'SPC-506' represents normalized response of the 506-th unit of the SPC model using the above homeostasis. 'DN' shows the closest DN fit by Eq. 23. The fitted parameters of DN are: $\gamma = 0.97$, $\rho = 0.35$ and $n = 2.05$. (B) Response vs. contrast curves for 228 out of 512 learnt simple cells whose receptive fields are well within the 16×16 visual field. The response is scaled to a maximum of 1 for each learnt cell. (C) Response vs. contrast curves for cells in cat V1 from an experimental study (Busse et al., 2009). The response is also scaled to a maximum of 1 for each cell. The black line represents the population average.

the 16×16 visual field. These 228 learnt cells display different degrees of contrast saturation, which is similar to the majority of experimental cells shown in Fig. 11C.

Overall, using the homeostasis function defined in Eq. 28 (Fig. 9), SPC model possess two important features: learning useful features and displaying nonlinearities. In our simulation here, when trained on natural images, SPC model can learn Gabor-like simple cells as well as display contrast-saturation. Our brain is a learning organism, so this SPC framework has a great potential of explaining how the neural circuits learns

to possess nonlinear properties.

4 Discussion

In this study, we aim to unify three significant principles: sparse coding, predictive coding and divisive normalization. Sparse coding and predictive coding can be identical in a two-layered neural network and unified as a sparse/predictive coding (SPC) framework whose model dynamics is determined by the predictive structure and a homeostasis function. In a single-neuron scenario, SPC can be identical to divisive normalization given the corresponding homeostasis function. We then show that in a neural network with 512 model cells, SPC with a specific homeostasis function can learn Gabor-like simple cells, similar to previous sparse and predictive coding model (Olshausen and Field, 1996, 1997; Rao and Ballard, 1999), as well as display contrast saturation, which is previously explained by DN (Heeger, 1992). Given the success of divisive normalization in modelling data of multisensory integration (Ohshiro et al., 2011, 2017), this SPC framework might be able to uncover how the brain learns to integrate input from multiple sensory systems.

4.1 Winner-take-all

Winner-take-all, the mechanism that only the neuron with the highest firing rate of the neural network is allowed to fire, has been used in many studies of computational neuroscience. The nice property of winner-take-all is that a strong competition is incorporated into the neural network, which will normally help with learning useful features from the input (Einhäuser et al., 2002; Rennó-Costa and Tort, 2017). Coultrip et al. (1992) proposed that a network with lateral inhibition can be used to perform winner-take-all competition. However, SPC framework proposed here is also closely related with winner-take-all mechanism, because winner-take-all can be seen as an extreme sparse coding model where only few neurons fire and this can be implemented in the SPC model with a large sparsity constraint. For example, when the homeostasis function, $H(\cdot)$, is a constant function, the value of the constant will change the percentage of active neurons in the network.

4.2 Homeostasis

Homeostasis is one of a prominent features of the brain neural circuits, namely that the brain employs a balancing mechanism to regulate neuronal activities (Turrigiano, 2011). In the SPC framework proposed in this study, the homeostasis function in the model dynamics (Eq. 9) regulates the response of each single neuron in a dynamic fashion, similar to the concept of homeostatic regulation of neuronal firing of biological neural circuits (Turrigiano, 2011). In the learning step of the SPC framework, weights are normalized after each update, similar to the concept of synaptic homeostasis (Turrigiano, 2011). However, the homeostasis function in the SPC model comes from the mathematical derivation of the original sparse and predictive coding model and is not meant to explicitly describe the homeostasis of real neural circuits. Our SPC framework combines Hebbian learning with homeostasis at a conceptual level and may serve as a fundamental mathematical model of understanding neural circuits (Turrigiano, 2017).

4.3 Biological plausibility

SPC framework proposed in this paper unifies sparse coding and predictive coding, and bridges its connection with divisive normalization. A predictive structure with feedforward input and feedback is adopted by SPC to preserve the essence of predictive coding and the models dynamics of SPC (Eq. 9) implements the network using an iterative procedure. However, as an mathematical framework directly derived from original sparse coding and predictive coding, a biologically plausible implementation of this model still requires much work to be done. For example, biological plausibility includes obedience to Dale's Law, biological realization of error neurons that compute prediction errors which can be both positive and negative, using spiking neurons in the network and spike-timing-dependant-plasticity for learning, reasonable choice of time constants of the dynamic system, synaptic delay between first and second layer, biological realization of homeostasis function, and many other. Though part of these aspects have been considered by some previous works (Denève and Machens, 2016; Lian et al., 2019), much still remains unexplored.

4.4 Prediction in future

The original predictive coding provide a reconstruction of the current input; i.e., that the model predicts the current input instead of future input. However, in real life, the input to the brain is mostly moving stimulus

that changes over time and the brain is constantly making predictions of the future. In this paper, the input to the model is static. However, this framework can also be used to investigate the problem of 'prediction in future' by incorporating synaptic delay between first and second layer of the model, which is closely related to recent studies of investigating predictive coding with temporal alignment (Hogendoorn and Burkitt, 2019; Burkitt and Hogendoorn, 2021). This aspect is left for our future work.

4.5 Mathematical analysis of the relationship between SPC and divisive normalization

In this paper, we analytically showed that SPC model can be identical to divisive normalization for a single-neuron scenario, namely that SPC with a homeostasis function can be derived from any functions of divisive normalization. Due to the difficulties of mathematically analyzing the relationship between SPC and divisive normalization in a network with multiple neurons, we only provide simulation results of a network implementation of SPC. The mathematical analysis in this aspect is part of our onging research.

References

Barlow HB (1961) Possible principles underlying the transformation of sensory messages. *Sens. Commun.* 1.

Barlow HB (1989) Unsupervised learning. *Neural Comput.* 1:295–311.

Bastos AM, Usrey WM, Adams RA, Mangun GR, Fries P, Friston KJ (2012) Canonical microcircuits for predictive coding. *Neuron* 76:695–711.

Beyeler M, Rounds EL, Carlson KD, Dutt N, Krichmar JL (2019) Neural correlates of sparse coding and dimensionality reduction. *PLoS Comput. Biol.* 15:e1006908.

Boutin V, Franciosini A, Ruffier F, Perrinet L (2020) Effect of top-down connections in hierarchical sparse coding. *Neural Comput.* 32:2279–2309.

Brito CS, Gerstner W (2016) Nonlinear hebbian learning as a unifying principle in receptive field formation. *PLoS Comput. Biol.* 12:e1005070.

Burkitt AN, Hogendoorn H (2021) Predictive visual motion extrapolation emerges spontaneously and with-

out supervision at each layer of a hierarchical neural network with spike-timing-dependent plasticity. *J. Neurosci.* 41:4428–4438.

Busse L, Wade AR, Carandini M (2009) Representation of concurrent stimuli by population activity in visual cortex. *Neuron* 64:931–942.

Carandini M, Heeger DJ (2012) Normalization as a canonical neural computation. *Nat. Rev. Neurosci.* 13:51–62.

Coultrip R, Granger R, Lynch G (1992) A cortical model of winner-take-all competition via lateral inhibition. *Neural Netw.* 5:47–54.

Denève S, Machens CK (2016) Efficient codes and balanced networks. *Nat. Neurosci.* 19:375–382.

Einhäuser W, Kayser C, König P, Kor̈ding KP (2002) Learning the invariance properties of complex cells from their responses to natural stimuli. *Eur. J. Neurosci.* 15:475–486.

Freeman TC, Durand S, Kiper DC, Carandini M (2002) Suppression without inhibition in visual cortex. *Neuron* 35:759–771.

Friston K (2010) The free-energy principle: a unified brain theory? *Nat. Rev. Neurosci.* 11:127–138.

Harpur GF, Prager RW (1996) Development of low entropy coding in a recurrent network. *Network: Comput. Neural Syst.* 7:277–284.

Heeger DJ (1992) Normalization of cell responses in cat striate cortex. *Vis. Neurosci.* 9:181–197.

Hogendoorn H, Burkitt AN (2019) Predictive coding with neural transmission delays: a real-time temporal alignment hypothesis. *eNeuro* 6.

Hubel DH, Wiesel TN (1959) Receptive fields of single neurones in the cat's striate cortex. *J. Physiol.* 148:574–591.

Hubel DH, Wiesel TN (1962) Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J. Physiol.* 160:106–154.

Hubel DH, Wiesel TN (1968) Receptive fields and functional architecture of monkey striate cortex. *J. Physiol.* 195:215–243.

Knill DC, Pouget A (2004) The Bayesian brain: the role of uncertainty in neural coding and computation. *Trends Neurosci.* 27:712–719.

Lian Y, Almasi A, Grayden DB, Kameneva T, Burkitt AN, Meffin H (2021) Learning receptive field properties of complex cells in V1. *PLoS Comput. Biol.* 17:e1007957.

Lian Y, Burkitt AN (2021) Learning an efficient hippocampal place map from entorhinal inputs using non-negative sparse coding. *eNeuro* 8:1–19.

Lian Y, Burkitt AN (2022) Learning spatiotemporal properties of hippocampal place cells. *eNeuro* 9.

Lian Y, Grayden DB, Kameneva T, Meffin H, Burkitt AN (2019) Toward a biologically plausible model of LGN-V1 pathways based on efficient coding. *Front. Neural Circuits* 13:13.

Ohshiro T, Angelaki DE, DeAngelis GC (2011) A normalization model of multisensory integration. *Nat. Neurosci.* 14:775–782.

Ohshiro T, Angelaki DE, DeAngelis GC (2017) A neural signature of divisive normalization at the level of multisensory integration in primate cortex. *Neuron* 95:399–411.

Ohzawa I, Sclar G, Freeman RD (1985) Contrast gain control in the cat's visual system. *J. Neurophysiol.* 54:651–667.

Ohzawa I, Sclar G, Freeman R (1982) Contrast gain control in the cat visual cortex. *Nature* 298:266–268.

Olsen SR, Bhandawat V, Wilson RI (2010) Divisive normalization in olfactory population codes. *Neuron* 66:287–299.

Olshausen BA, Field DJ (1996) Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature* 381:607–609.

Olshausen BA, Field DJ (1997) Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Res.* 37:3311–3325.

Papyan V, Romano Y, Elad M (2017) Convolutional neural networks analyzed via convolutional sparse coding. *J. Mach. Learn. Res.* 18:2887–2938.

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

Rennó-Costa C, Tort AB (2017) Place and grid cells in a loop: implications for memory function and spatial coding. *J. Neurosci.* 37:8062–8076.

Rozell CJ, Johnson DH, Baraniuk RG, Olshausen BA (2008) Sparse coding via thresholding and local competition in neural circuits. *Neural Comput.* 20:2526–2563.

Smith EC, Lewicki MS (2006) Efficient auditory coding. *Nature* 439:978–982.

Spratling MW (2008) Reconciling predictive coding and biased competition models of cortical function. *Front. Comput. Neurosci.* 2:4.

Turrigiano G (2011) Too many cooks? Intrinsic and synaptic homeostatic mechanisms in cortical circuit refinement. *Ann. Rev. Neurosci.* 34:89–103.

Turrigiano GG (2017) The dialectic of hebb and homeostasis. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* 372:20160258.

Zhu M, Rozell CJ (2013) Visual nonclassical receptive field effects emerge from sparse coding in a dynamical system. *PLoS Comput. Biol.* 9:e1003191.

Zylberberg J, Murphy J, DeWeese M (2011) A sparse coding model with synaptically local plasticity and spiking neurons can account for the diverse shapes of V1 simple cell receptive fields. *PLoS Comput. Biol.* 7:e1002250.