Learning Divisive Normalization in Primary Visual Cortex

Max F. Burg,[1,2,*] Santiago A. Cadena,[1-3] George H. Denfield[3,4]
Edgar Y. Walker,[3,4] Andreas S. Tolias,[2-5,§] Matthias Bethge[1-3,§]
Alexander S. Ecker[1-3,§,†]

[1] Inst. for Theoretical Physics and Centre for Integrative Neuroscience, University of Tübingen, Germany
[2] Bernstein Center for Computational Neuroscience, Tübingen, Germany
[3] Center for Neuroscience and Artificial Intelligence, Baylor College of Medicine, Houston, TX, USA
[4] Department of Neuroscience, Baylor College of Medicine, Houston, TX, USA
[5] Department of Electrical and Computer Engineering, Rice University, Houston, TX, USA

[*] max.burg@bethgelab.org
[§] These authors contributed equally
[†] Present address: Department of Computer Science, University of Göttingen, Germany

July 24, 2020

Abstract

Divisive normalization (DN) is a prominent computational building block in the brain that has been proposed as a canonical cortical operation. Numerous experimental studies have verified its importance for capturing nonlinear response properties to simple, artificial stimuli, and computational studies suggest that DN is also an important component for processing natural stimuli. However, we lack quantitative models of DN that are directly informed by empirical data and applicable to arbitrary stimuli. Here, we developed an image-computable DN model and tested its ability to predict spiking responses of a large number of neurons to natural images. In macaque primary visual cortex (V1), we found that our model outperformed linear-nonlinear and wavelet-based feature representations and performed on par with state-of-the-art convolutional neural network models. Our model learns the pool of normalizing neurons and the magnitude of their contribution end-to-end from the data, answering a long-standing question about the tuning properties of DN: within the classical receptive field, oriented features were normalized preferentially by features with similar orientations rather than non-specifically as currently assumed. Overall, our work refines our view on gain control within the classical receptive field, quantifies the relevance of DN under stimulation with natural images and provides a new, high-performing, and compactly understandable model of V1.

Author summary

Divisive normalization is a computational building block apparent throughout sensory processing in the brain. Numerous studies in the visual cortex have highlighted its importance by explaining nonlinear neural response properties to synthesized simple stimuli like overlapping gratings with varying contrasts. However, we do not know if and how this normalization mechanism plays a role when processing complex stimuli like natural images. Here, we applied modern machine learning methods to build a general divisive normalization model that is directly informed by data and quantifies the importance of divisive normalization. By learning the normalization mechanism from a data set of natural images and neural responses from macaque primary visual cortex, our model made predictions as accurately as current stat-of-the-art convolutional neural networks. Moreover, our model has fewer parameters and offers direct interpretations of them. Specifically, we found that neurons that respond strongly to a specific orientation are preferentially normalized by other neurons that are highly active for similar orientations. Overall, we propose a biologically motivated model of primary visual cortex that is compact, more interpretable, performs on par with standard convolutional neural networks and refines our view on how normalization operates in visual cortex when processing natural stimuli.

1 Introduction

A crucial step towards understanding the visual system is to build models that predict neural responses to arbitrary stimuli with high accuracy (Carandini et al., 2005). The classical standard models of primary visual cortex (V1) are based on linear-nonlinear models (Simoncelli et al., 2004), energy models (Adelson and Bergen, 1985) and subunit (LN-LN) models (Rust et al., 2005; Touryan et al., 2005; Willmore et al., 2008; Butts et al., 2011; McFarland et al., 2013; Vintch et al., 2015). Fueled by advances in machine learning technology, recent studies have shown that multi-layer convolutional neural networks (CNNs) can significantly improve prediction of neural responses to complex images and videos at several stages of the visual pathway (Yamins et al., 2014; Khaligh-Razavi and Kriegeskorte, 2014; McIntosh et al., 2016; Zhang et al., 2018; Cadena et al., 2019; Kindel et al., 2019; Walker et al., 2019; Sinz et al., 2018), outperforming classical models. The current state-of-the-art data-driven model of single-unit activity in monkey V1 is a three-layer CNN (Cadena et al., 2019). However, it is challenging to gain insights into V1 function from the feature space produced by deeper layers of such models. In particular, we do not have first principles explaining the kind of nonlinear mapping the CNNs' composition of layers approximates as a whole.

A promising candidate to facilitate a more succinct description of V1 neurons is to replace the CNN's computations in deeper, hardly understandable feature spaces by a divisive normalization nonlinearity (Heeger, 1992) that is easier to interpret. Divisive normalization has been proposed to be a canonical neural computation present throughout the visual pathway (Carandini and Heeger, 2012) because it (1) explains a wide variety of neurophysiological phenomena using simple stimuli (Carandini and Heeger, 2012; Sawada and Petrov, 2017) and (2) can be derived

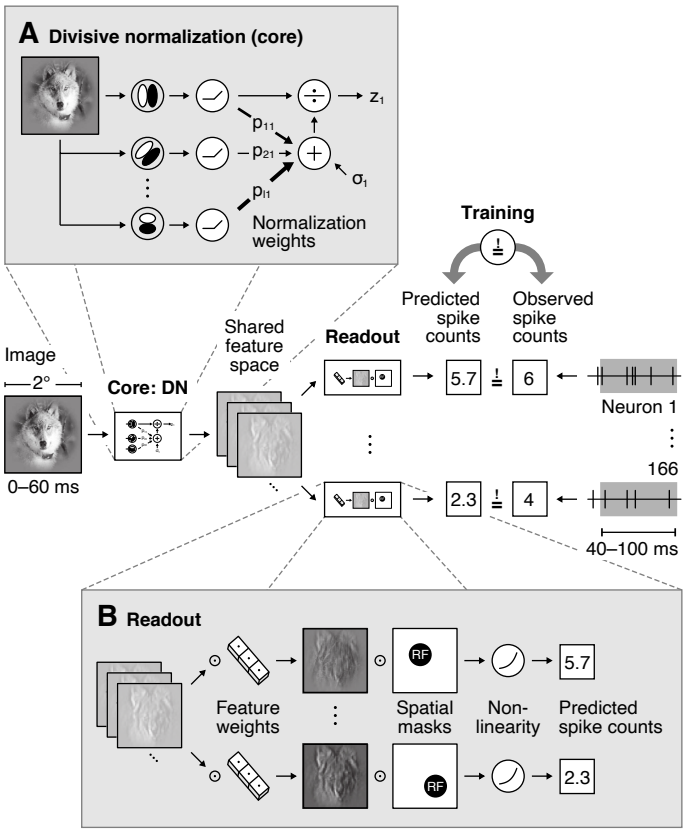

Figure 1: Overview of our divisive normalization (DN) model. The model takes as input an image and predicts neurons' spike counts in response to this image (details in Fig. 2). The model is split into two parts: a *core* that computes a shared nonlinear feature space and a *readout* that maps the shared feature space individually to each neuron's spike count. **A.** Divisive normalization mechanism (simplified). The visual input is convolved with 32 filters and then rectified to produce an excitatory output. The output of each filter is then divided by a weighted sum of the excitatory outputs of all filters with normalization weights p_{kl} and a semi-saturation constant σ_l. In our general formulation, all weights and constants are learned from the data. **B.** Readout that maps the shared feature space to each neuron's spike count through an individual weighted sum over the entire shared feature space and a pointwise output nonlinearity. The readout weights are factorized into a feature vector – capturing the nonlinear feature(s) that a neuron computes – and a spatial mask – localizing each neuron's receptive field (RF).

from first principles of redundancy reduction (Schwartz and Simoncelli, 2001; Sinz and Bethge, 2008). At the same time, a variant of DN was successfully used in computer vision yielding state-of-the-art natural image compression with high perceptual quality (Ballé et al., 2017). A prominent example of normalization phenomena in V1 is cross-orientation inhibition. Here, the response of a neuron to a driving grating stimulus in the receptive field (RF) is suppressed by superimposing a second grating that does not elicit a response when presented alone: for instance, a grating with orientation orthogonal to the neuron's preferred orientation (Bonds, 1989; Morrone et al., 1982; DeAngelis et al., 1992; Heeger, 1992; Carandini et al., 1997; Busse et al., 2009).

The basic idea of divisive normalization (Fig. 1A) is that a neuron's driving input is normalized divisively by a weighted sum over nearby neurons' responses (Heeger, 1992; Carandini and Heeger, 2012). While the general idea is simple, elegant and powerful, our current knowledge of DN is limited in two important ways: (1) DN has been studied mostly using simple stimuli but for natural images we do not know whether incorporating DN into predictive models of neural responses to natural stimuli improves these models' performance, and (2) we currently do not know how response properties of neurons with overlapping receptive field determine whether a neuron contributes to the normalization pool and, if so, with what normalization weight (Carandini et al., 1997).

To explain normalization phenomena within the classical receptive field like cross-orientation inhibition, current models of divisive normalization assume that all nearby neurons with diverse orientation tuning preferences and with similar receptive field locations contribute equally to the normalization pool (Heeger, 1992; Carandini et al., 1997; Busse et al., 2009). However, some earlier experimental studies suggest that this assumption may not be correct for some neurons (Bonds, 1989; DeAngelis et al., 1992), and normative models of normalization predict that the magnitude with which a given neuron contributes to another neuron's normalization should depend on the relationship of their response properties (Schwartz and Simoncelli, 2001).

In this paper, we address the two main questions raised above: (1) When predicting V1 neural responses to natural images, will divisive normalization added to simple subunit models match the performance of state-of-the-art CNNs and (2) what stimulus features are preferred for normalizing a V1 neuron's response in relation to its receptive field features?

We focus on responses to stimuli mostly restricted to the classical receptive field and on models that account only for normalization by neurons with overlapping receptive field locations. We developed an end-to-end trainable divisive normalization model to predict V1 spike counts from natural stimuli. Our model learns the filter coefficients of all neurons as well as their normalization weights directly from the data.

We applied our divisive normalization model to natural image responses in monkey V1 and found that it outperforms linear-nonlinear and subunit models, and is on par with that of state-of-the-art CNNs while requiring much fewer parameters. This result suggests that DN is an important nonlinear computation not only under stimulation with simple stimuli, but with natural images, too. Furthermore, we found that oriented features were normalized stronger by features with similar orientation, which leads to a quantitatively better model of the data in contrast to the current standard model of nonspecific normalization (Heeger, 1992; Busse et al., 2009). Our work thus advances our understanding of V1 function by establishing a new compact, state-of-the-art divisive normalization model and predicting an orientation-specific gain control mechanism under stimulation with natural stimuli.

2 Results

2.1 Learning divisive normalization

The basic idea of divisive normalization (Fig. 1A) is that the response of neuron l

$$z_l(x) = \frac{y_l(x)}{\sigma_l + \sum_{k \in \mathcal{K}} p_{kl} \cdot y_k(x)} \tag{1}$$

is given by its driving input activity $y_l(x)$ divisively normalized by a weighted sum over nearby neurons' responses $y_k(x)$ (Heeger, 1992; Carandini and Heeger, 2012), where x represents the stimulus and σ_l is a semi-saturation constant. Here, the set of normalizing neurons $\mathcal{K}$ and the normalization weights p_{kl} define which neurons contribute to the normalization pool of neuron l and with what strength, respectively.

While this formulation is straightforward to write down, it is challenging to build quantitative models based on it that are applicable to arbitrary inputs. The denominator depends on a potentially large population of neurons – which is unknown in general – and the structure of the normalization weights has been studied only using very restricted sets of simple stimuli such as oriented gratings and bars. Previous modeling work on divisive normalization has therefore made specific assumptions about the filter properties of the underlying neuronal population and either modeled only a closed set of stimuli such as gratings of different orientation (Heeger, 1992; Carandini et al., 1997; Freeman et al., 2002; Heuer and Britten, 2002; Busse et al., 2009) or evaluated models only qualitatively (Schwartz and Simoncelli, 2001; Wainwright et al., 2002; Froudarakis et al., 2014).

We developed a general, image-computable predictive model of divisive normalization following Eq. (1), which is applicable to arbitrary images and whose parameters are learned by optimizing the accuracy of the model in predicting the spiking activity of a large number of neurons in response to natural images (see Fig. 1). Our model builds on a recent innovation in predictive modeling (Antolík et al., 2016; Klindt et al., 2017; Batty et al., 2016; McIntosh et al., 2016; Cadena et al., 2019), jointly modeling all recorded neurons instead of learning a predictive model for each neuron individually. Because many neurons perform similar computations – up to shifts in receptive field location – jointly modeling them makes more efficient use of the data and we can learn more complex models. The basic idea is to split the model into two parts (Fig. 1): (1) a *core* that transforms the input image into nonlinear features shared among all neurons, and (2) a *readout* that linearly combines the features to produce a prediction of each neuron's response.

We use a convolutional network for the core, whose architecture lends itself very well to model divisive normalization. By construction, we have a model that contains all filters necessary to account for the recorded neurons' responses. All of these filter responses are automatically extracted at each location, providing a good approximation of the underlying population of neurons in the brain although it is only sparsely sampled during the experiment. As a consequence, we can optimize the pool of neurons providing normalizing inputs and their corresponding weights p_{kl} (Eq. 1) to account for the neural responses. In this work we mostly focus on normalization effects from neurons with overlapping receptive fields, rather than investigating interactions with the RFs' surround.

In summary, our model's core (Fig. 1A) consists of a set of convolutional filters (we use 32) that provide the driving inputs, followed by a DN stage (Eq. 1). This core is shared among all neurons and converts the image into a set of feature maps. These feature maps are converted into response predictions by a linear readout step (Fig. 1B) that picks the relevant features and spatial locations for each neuron, followed by an output nonlinearity. To ensure that the readout does not model any complex computation, we constrain its weights to be non-negative. The non-negativity ensures that activations can only add, preventing the readout stage from accounting for any suppressive effects. The readout can, however, account for response invariances such as phase invariance of complex cells; see Methods for an in-depth explanation. While our model reflects the general formulation of divisive normalization, mathematically it is not identical to the classical formulation of DN due to the additional linear-nonlinear readout after the model's core.

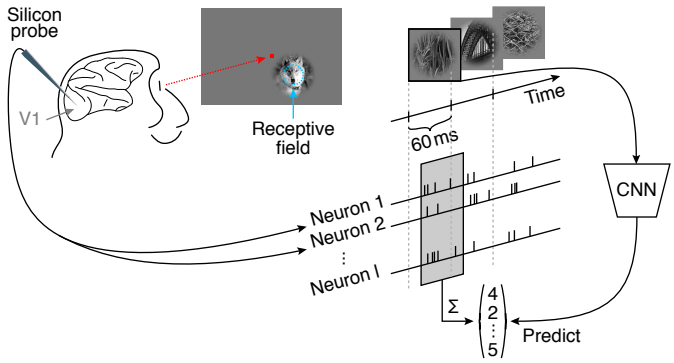

Figure 2: Experimental paradigm from Cadena et al. (2019). Natural images were flashed to a monkey covering 2° of their visual angle, and located at the center of the multi-unit receptive field. Multiple neurons were isolated from recordings with silicon probes inserted into V1 (Denfield et al., 2018). Natural images were shown in a fast sequence without blanks, each presented for 60 ms. Spike counts from all isolated neurons corresponding to each image were extracted from a window 40 ms after the image onset lasting 60 ms.

2.2 DN model achieves competitive accuracy with fewer parameters

We fit the model described above to a dataset of 166 neurons recorded in V1 of two awake, fixating monkeys (data from Cadena et al. 2019), who viewed a fast sequence of localized natural images and textures (Fig. 2). The stimuli were centered on the neurons' receptive fields and covered about twice the area of the classical receptive fields, mostly stimulating the near vicinity of the RFs' center. Images were shown for 60 ms each, without blanks in between. Single unit activity was recorded with laminar silicon probes sampling from all cortical layers. We fit the model jointly to the responses of all neurons. As neurons were recorded in 17 recording sessions, the dataset sampled a diverse range of preferred orientations. The objective function during training was to minimize the difference between the model's prediction and the observed spike counts of the neurons in a time window 40–100 ms after image onset (to account for response latency).

To evaluate model performance, we estimated the fraction of explainable variance explained (FEV), which quantifies the fraction of the stimulus-driven response variance that is accounted for by the model, and ignores unexplainable trial-to-trial variability in the response of the neurons (see Methods). A perfect model would reach a FEV of 100%.

Subunit models are an established approach to model primary visual cortex responses (Rust et al., 2005; Touryan et al., 2005; Willmore et al., 2008; Butts et al., 2011; McFarland et al., 2013; Vintch et al., 2015). In addition to capturing a fair portion of the explainable variance (Cadena et al., 2019), their computations are easily understandable because they are rather shallow models consisting of a first stage of rectified linear filtering followed by a static nonlinearity, and then a pooling stage with a subsequent output nonlinearity. Structurally, it is the same as our DN model, but without the normalization stage. Therefore, we considered a convolutional subunit model as a strong baseline for our model. This subunit model accounted for 46.6%

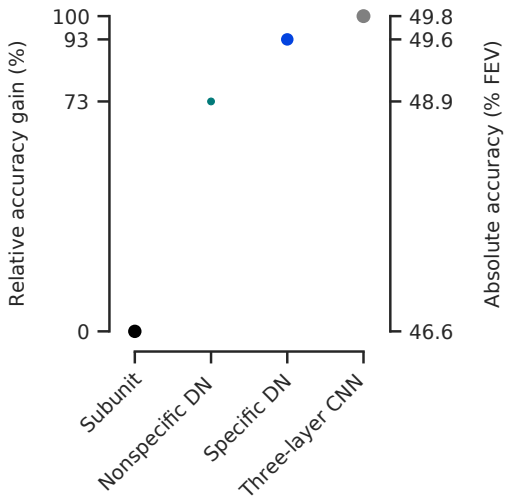

Figure 3: Performance comparison of our models fitted to the data from Cadena et al. (2019) relative to the gap between the best shallow model – a subunit one layer convolutional neural network (CNN) – and the deeper data-driven state-of-the-art three-layer CNN (Cadena et al., 2019). Non-specific divisive normalization (DN) accounts for 73% of this gap, while specific DN improves it up to 93%. Absolute values in terms of percentage of explainable variance explained (FEV) on the right (mean over the ten best models selected in terms of validation set accuracy). Diameters of dots correspond to standard error of the mean except for CNN (value not provided in original study).

FEV. In comparison, a regularized linear nonlinear Poisson model (LNP) only accounted for 16.3% FEV on the same dataset (Cadena et al., 2019) due to its inability to model complex cells.

As recent developments in machine learning technology have allowed us to improve predictive performance, we used the current best data-driven model as a gold standard. This model is a convolutional neural network with three convolutional layers and a linear-nonlinear readout, reaching a performance of 49.8% FEV (Cadena et al., 2019). However, although this model outperforms the simpler subunit model, it is of higher complexity requiring a higher number of parameters.

To evaluate how well our DN model accounts for the data, we placed its performance on a scale between 0% (baseline: subunit model) and 100% (gold standard: CNN). On that scale, our DN model achieved a score of 93% between the baseline and gold standard (49.6% FEV on test set, mean over the ten best models selected in terms of validation set accuracy, Fig. 3), being on par with the state-of-the-art model. Notably, we achieved this performance gain by simply adding the trainable DN stage to the convolutional subunit model, which quantifies the importance of divisive normalization as computational mechanism to predict neural responses in V1 under stimulation with complex, natural images.

While our DN model's accuracy is practically on par to that of the state-of-the-art CNN, it requires substantially fewer parameters to achieve this performance (Table 1): The DN model's

Model	Number of parameters Core	Readout per neuron
Subunit model	5 440	867
Nonspecific divisive normalization	5 536	867
Divisive normalization	**6 528**	**867**
Three-layer CNN (Cadena et al., 2019)	23 936	867

Table 1: Number of parameters for different models.

core – i. e. the shared computational backend before the linear readout – uses only 27.3% of the parameters of the CNN model's core. This saving in parameters suggests that the DN model captures important structure in the data, which we elaborate in the next section. Compared to the number of parameters required by the subunit model's core, the DN model requires 20.0% more parameters (allocated to the divisive normalization module). The number of readout parameters – i. e. the part that turns the shared nonlinear feature representation into individual neurons' responses – is the same for all models.

2.3 Normalization is feature-specific

Having established that the DN model outperforms the rather simple subunit model and performs as well as the gold standard CNN, we next investigated the structure of the normalizing input, i. e. the sum in the denominator of Eq. (1) and how strongly different features contribute to it. For this analysis, we focused on the question if DN is specific to the orientation difference between orientation-selective features, i. e. features with an anisotropic structure. Visually inspecting the strength of the normalizing inputs suggests that oriented features are normalized preferentially by features with similar orientation preference (Fig. 4A). In contrast, orthogonal features seem to contribute less.

To quantify the difference with which the two groups contribute to normalization, we split the sum in Eq. (1) into two parts and collect the contribution of normalizing features with similar ($< 45°$) orientation as the driving feature and that of features with dissimilar ($\geq 45°$) orientations. Analyzing the normalization of each oriented feature individually, we found that most oriented features are more strongly normalized by features with similar orientations (Fig. 4B). To assess whether our qualitative observation above is a general property of the data or a spurious characteristic of that one particular model we selected, we repeated this analysis for the top-10 models (assessed in terms of performance on the validation set, only differing in the initialization of the model parameters before training) and observed similar behavior. Averaging over the features, we found that, for all of these models, similar orientations contributed more strongly than dissimilar orientations. Taking the data of all top-10 models into account, we found that, on average, similarly oriented features contributed 49% more normalizing input than dissimilar features (Wilcoxon signed rank test; $p < 0.006$, $N = 10$ models; Cohen's $d = 1.2$).

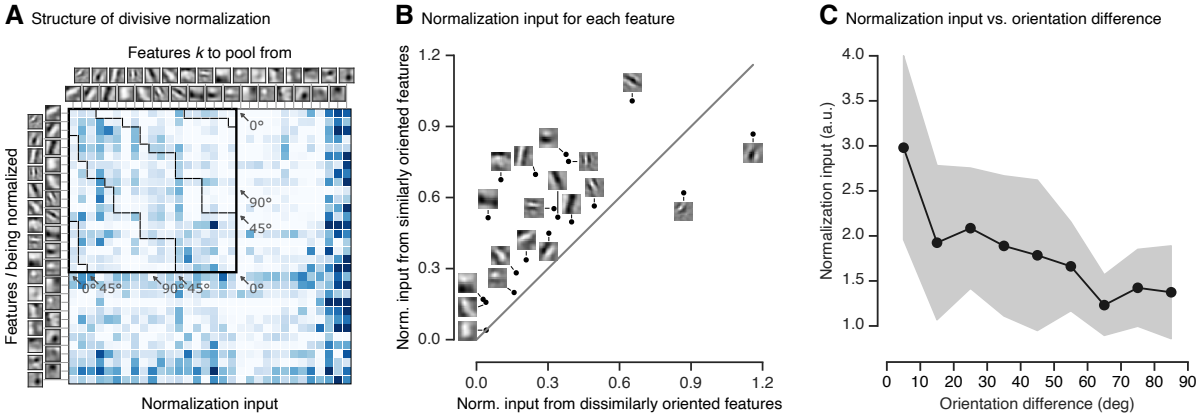

Figure 4: Structure of divisive normalization. **A.** The matrix shows the average strength (over images) of the normalizing inputs for each combination of filter response being normalized (rows) and filter response providing normalizing input (columns). Darker shades of blue indicate stronger normalization. Orientation-selective filters are grouped at the top, ordered by preferred orientation and marked by the black square. The dashed black lines within the square separate pairs of filters with similar ($< 45°$) and dissimilar ($> 45°$) orientations. Normalizing inputs are stronger for similarly tuned filters. Unoriented filters are sorted by total normalization input. Darkest blue color corresponds to the maximum normalization input for the group of oriented filters, higher normalization input values for the unoriented filters are clipped to that value. Data of the model with highest accuracy on the validation set is shown. **B.** Normalization input from similar orientations ($< 45°$) compared to the normalization input from dissimilar orientations ($\geq 45°$) for each feature. Grey line: identity. Most features are normalized preferentially by the responses of filters with similar preferred orientations. Data of the model with highest accuracy on the validation set is shown. **C.** Normalization input, binned into orientation difference of 10°. Each bin was averaged over the top-10 models (assessed on the validation set). The shaded area depicts the standard deviation per bin.

Having established that normalizing inputs are orientation-specific, we analyzed this specificity in more quantitative detail. Instead of using just two groups as before, we split up the normalizing inputs into nine bins of 10° width each and averaged those bins across the top-10 models. This analysis revealed that the strength of the normalizing inputs decreased as the difference in orientation increased (Fig. 4C). Hence, the more similar a normalizing feature's orientation was to the feature to be normalized, the stronger was its contribution to normalization. In fact, features in the group most similar to the driving input contributed 117% more than those in the orthogonal group (Cohen's $d = 1.9$).

2.3.1 Control: Nonspecific divisive normalization reduces accuracy

To determine how important orientation-specific normalization is, we performed a control experiment: For each feature l being normalized, we constrained all of its incoming normalization weights p_{kl} to be identical. This model is an instantiation of the previous models (Heeger, 1992; Carandini et al., 1997; Busse et al., 2009) assuming non-specific normalization from all features. Note that, since the feature orientations were non-uniformly distributed (Fig. 4A), this model's normalization was weakly orientation tuned despite equal weights for all features (orientation

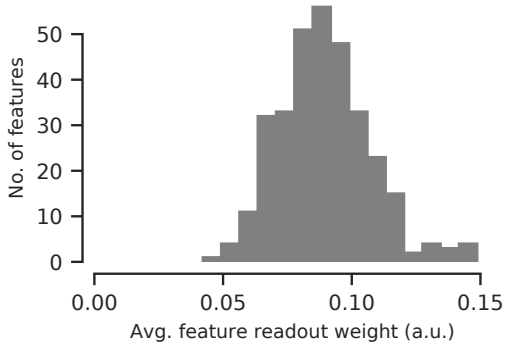

Figure 5: Histogram of feature readout weights of the ten best performing models in terms of validation set accuracy. For each model, feature weights are normalized across channels and averaged across individual neurons. All model's channels are used to predict neural activity.

difference $< 45°$ contributed 14% stronger than $\geq 45°$; Wilcoxon signed rank test; $p < 0.007$, $N = 10$ models; Cohen's $d = 0.3$). This model achieved a performance of 73% between the baseline and gold standard (48.9% FEV). While it does not match the performance of our more general DN model, it does outperform the subunit baseline. Thus, stronger orientation-specific normalization is necessary to achieve full performance, but a simpler form of uniform divisive normalization can account for a large fraction of the improvement over the subunit baseline.

2.3.2 Control: All channels contribute to our model's prediction

One potential caveat of our analyses so far is that we analyzed the orientation specificity of DN in terms of the convolutional feature maps in our model's core rather than the actual neurons we recorded. These features provide a much more compact view of the population of neurons, because they are invariant to the receptive field locations and the neural responses are simple linear combinations of those features. However, it is not clear a-priori whether all features are equally important for predicting the activity of the neurons in our population. Thus, considering convolutional features instead of actual neurons may lead to a skewed view of the population. To verify that this is not an issue, we quantified how much each feature contributed to the overall activity of all neurons by normalizing the feature readout weights across channels and averaging across neurons. The resulting distribution (Fig. 5) containing these averaged feature readout weights for the best ten models had a coefficient of variation of 0.2. We therefore concluded that all features were read out by roughly the same number of neurons and hence were similarly important to predict neural activity. Thus, our interpretation of orientation-specific normalization is unlikely to be an artifact of analyzing the convolutional features rather than the actual neurons.

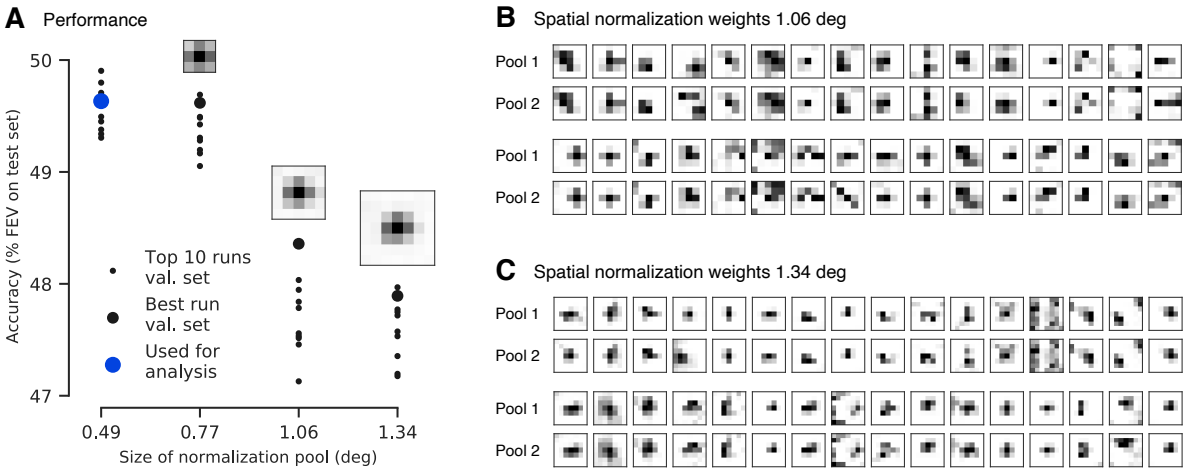

Figure 6: A. Test set performance of the ten best performing DN models (chosen by validation set accuracy). The model's performance rapidly decreases for spatially increasing normalization pool size (in units of visual angle in degrees). For the analysis in this paper, we used the best model on the validation set, indicated by a blue dot. **B. & C.** Weights of the spatial normalization pool for the best performing model with pool size of (**B.**) 1.06° of visual field / 5 px × 5 px and (**C.**) 1.34° of visual field / 7 px × 7 px (evaluated in terms of the validation set accuracy). For each feature (columns), the two components (rows) of the in total 32 spatial normalization pools are shown. Darker color corresponds to higher weights. Both components are similar. **A. insets:** Average across features and normalization pool components. The model learned normalization from the receptive field center (on average).

2.3.3 Control: No surround influence in our results and dataset

We have observed orientation-specific divisive normalization in the classical receptive field. Surround suppression is known to be orientation-specific (Blakemore and Tobin, 1972; DeAngelis et al., 1994; Cavanaugh et al., 2002; Coen-Cagli et al., 2015), so a potential concern would be that some of the extra-classical surround of a unit's RF contributed to the results presented above. To rule out this possibility, we fit a more general DN model, where we additionally learn the spatial structure of the normalization pool instead of just limiting it to neurons with overlapping receptive fields (see Methods). This extended DN model included two normalization pools that could have different patterns of weights along the feature dimension. It is therefore general enough to account for the standard model of DN with a nonspecific center normalization pool and orientation-specific surround suppression.

In contrast to what one may expect, spatially expanding the normalization pool to cover larger surround areas did not increase our model's accuracy; in fact, for lager surrounds the performance even decreased (Fig. 6A). The best performance was achieved for models with the smallest normalization pool mostly restricted to units' classical RFs (approximately 0.5° diameter). The normalization weights of the extended spatial normalization pool showed no visible separation into center and surround and exhibited no or only weak contributions from the classical RF's surround (Fig. 6). If leakage from the surround were the source of orientation-specific normalization, we would expect to observe spatial profiles that cover the

surround, which is not the case (except for a few exceptions, which are mostly untuned features). We thus concluded that our model is unlikely to capture the influence from the RF surround. The reason for this limitation could be that our stimuli were mostly restricted to the classical receptive field (see Methods), preventing any model to capture information from the surround.

3 Discussion

To improve our understanding of primary visual cortex, we asked what condensed nonlinear function state-of-the-art CNNs might implement for predicting V1 responses to localized natural stimuli. To answer this question, we developed an end-to-end learnable divisive normalization model and fit it to neural responses. Both the unspecific control model and the full model that learned the normalization pool outperformed the subunit model. The full DN model improved performance even further, reaching an accuracy competitive with the gold standard CNN while having fewer parameters. This result of increased performance quantifies the relevance of DN to predict V1 responses to natural images.

Along those lines, one may ask whether the difference between the non-specific DN model and the full model learning orientation specific normalization weights is relevant, because the full model may simply be able to better account for some insignificant biological heterogeneity due to its additional parameters. Although it is possible, we believe that this explanation is unlikely, because oriented features are preferentially normalized by channels with similar orientation. If the model was simply picking up some biological imperfection, we would expect the normalization weights not to depend systematically on preferred orientation.

Previous experimental work investigated suppressive phenomena within the receptive field only qualitatively or used simple stimuli that mainly consisted of a combination of driving and mask gratings. Morrone et al. (1982) find suppression at all orientations, but do not investigate orientations similar to preferred orientation. Bonds (1989) report predominantly orientation-nonspecific suppression, although three of fourteen cells exhibit stronger suppression with masks oriented similarly to the neurons' preferred orientations, and a few other cells are suppressed most strongly by mask orientations orthogonal to the preferred orientation. Similarly, DeAngelis et al. (1992) find suppression to be predominantly independent of orientation, although for some cells an increased suppression for a range of orientations near the optimal excitatory orientation is apparent. Heeger (1992) explains those results by proposing an orientation-nonspecific divisive normalization model. Carandini et al. (1997) consider the possibility of orientation-specific normalization which provides a marginal improvement in the quality of their model fits to the data. However, they conclude that their dataset was not specifically designed to provide a strong test of this question and their results are inconclusive in this respect. Busse et al. (2009) develop a quantitative model for the response of a population of neurons to a combination of gratings. Assuming nonspecific normalization by overall contrast, their model predicts the collective action of the whole neuron population better than linear and winner-take-all baselines, but they do not test against an orientation-specific alternative model. To summarize, these studies find phenomena that are predominantly explained by nonspecific normalization (Heeger, 1992), some of them encountering only weak orientation-specific phenomena and only

in relatively few cells. Until now, a quantitative analysis of orientation-specificity on a data set of natural images was missing.

Our findings are largely consistent with previous experimental results and quantitatively refine them using a larger dataset, place them in the context of other models of V1 and show that the same mechanisms observed with simple stimuli also apply under more natural stimulus conditions. Interestingly, and somewhat in contrast to earlier work, channels with preferred orientations within 10° of the driving feature provided twice as strong normalizing input than those with orthogonal preferred orientations. The reason for this difference between our findings and previous studies could be that we used natural stimuli, which have different image statistics compared to simple stimuli used in earlier studies. Furthermore, most previous studies of divisive normalization were performed in cats (Morrone et al., 1982; Bonds, 1989; DeAngelis et al., 1992; Busse et al., 2009) and the results therein may not generalize to monkeys, for which preceding studies are inconclusive regarding orientation specificity (Carandini et al., 1997).

Recent work modeling a large set of classical psychophysical data also suggests an orientation-specific divisive normalization: Schütt and Wichmann (2017) developed an image-computable model of early vision very similar in structure to ours, and found that in order to explain classical data on contrast detection, contrast discrimination and oblique masking, their model required divisive normalization to be orientation-specific. Similar results had been reported in an earlier study (Itti et al., 2000).

Following a normative approach, Schwartz and Simoncelli (2001) derive an ecologically justified divisive normalization model from the efficient coding hypothesis (Barlow, 1961) that is able to qualitatively describe the orientation masking data of Bonds (1989). Reducing the statistical redundancy of responses to natural stimuli predicts that normalization should be stronger for neurons that exhibit a higher dependency in their unnormalized responses. This theoretical result implies that normalization weights should not be uniform, consistent with our empirical findings. Iyer and Burge (2019) compare the output statistics of a divisive normalization model with broadband and narrowband normalization to natural images, $1/f$ stimuli and white noise stimuli. They report that feature-specific divisive normalization improves the discriminability among natural images compared to unspecific normalization.

Is our discovery of divisive normalization by similar orientations actually implemented by the connectivity of neurons in primary visual cortex? The answer to this question could be reflected in the connectivity from inhibitory parvalbumin-expressing (PV) interneurons to pyramidal cells and their relation to neurons' tuning properties. Hofer et al. (2011) find that, in the mouse, pyramidal cells and PV cells are homogeneously connected. Although a weak bias towards orientation tuning is apparent, they conclude that local inhibition in V1 is primarily non-specific. However, despite the connection probability between PV and pyramidal cells being homogeneous, connection strengths are quite heterogeneous: Individual PV cells strongly inhibit those pyramidal cells that share their visual selectivity (Znamenskiy et al., 2018). This result is in line with our finding of orientation-specific normalization. Similarly, recent work employs single-neuron perturbations in layer 2/3 of mouse V1 and directly measures higher suppression of neighbouring excitatory neurons with a similar preferred orientation (Chettih and Harvey, 2019).

A limitation of our study is that the stimuli in our dataset are spatially restricted to approximately twice the size of the classical receptive field, which prevented us from learning the influence of the surround on normalization. Moreover, we here focused on single images to predict a spike count in a relatively short time window covering the transient response, and ignored any temporal aspects or more sustained periods of the response. These limitations, however, are imposed by the available data – the modeling approach generalizes very well to cover both the surround and the temporal structure – and thus should be addressed in future work.

In conclusion, we developed a model consisting of one layer of subunits followed by learned orientation-specific divisive normalization, which accounted remarkably well for the V1 data. We hope that this quantitative approach of evaluating theories of computation in the brain by formalizing them as (components of) trainable predictive models will be used more widely in the future, so the field will (slowly) converge to an accurate general-purpose model of the visual system applicable to natural inputs.

4 Methods

4.1 Experimental details

We used the dataset described in detail in Cadena et al. (2019) and provide a summary of the most important characteristics here. Electrophysiological recordings from two healthy, male rhesus macaque monkeys aged 12 and 9 years were performed with a 32-channel linear silicon probe. The monkeys were head-fixed and placed in front of a screen. They were trained to fixate on a target located at the center of the screen. The start of a trial was determined by maintained fixation on the target for 300 ms. The fixation tolerance was set to 0.42° around the center of the target. At the beginning of each recording session, population receptive fields were mapped with a sparse random dot stimulus. Each dot was of size 0.12° of visual angle and was presented over a uniform gray background, changing location and light intensity (black or white) randomly every 30 ms. The receptive field profiles per electrode channel were then obtained via reverse correlation (i. e. spike-triggered average). The center location of the population receptive field was subsequently estimated by averaging over channels and fitting a two-dimensional Gaussian to the reverse correlation profiles. Afterwards, this location was used to place the images of the natural stimulus paradigm.

The dataset in Cadena et al. (2019) consists of 7 250 distinct natural, greyscale images which were presented two to four times each. A fifth of these images (1 450) were taken from ImageNet (Russakovsky et al., 2015). Four additional texturized images were synthesized from each of them, preserving varying degrees of higher-order statistics. The images were cropped to $140\,\text{px} \times 140\,\text{px}$ covering two degrees of visual angle. Before displaying the images on the screen, the images were normalized such that the central 1° (70 px) of each image had the same mean and standard deviation. The mean was set to the screen's mean gray intensity (128) and the standard deviation was set to the average standard deviation of the original images. Pixels with an intensity that fell outside the display's range $[0, 255]$ where clipped. Afterwards, all

images were overlaid with a circular mask with a soft cosine fade-out and an aperture with a diameter of 1°.

Images were presented for 60 ms with no blanks in between. Neural responses were extracted in time windows of 40–100 ms after image onset (Fig. 2), accounting for typical response latencies in primary visual cortex. The image sequence was randomized with the restriction that consecutive images do not belong to the same type (i. e. natural or one of the four texturized versions).

We discarded a few isolated neurons if their stimulus driven variability was too low. The explainable variance in a dataset is smaller than the total variance because the observation noise prevents even a perfect model to account for all the variance in the data. Thus, targeting neurons that have sufficient explainable variance is necessary to train meaningful models of visually driven responses. For a neuron's spike count r, the explainable variance $\mathrm{Var}_{\mathrm{exp}}[r]$ is the difference between the the total variance $\mathrm{Var}[r]$ and the variance of the observational noise $\sigma^2_{\mathrm{noise}}$,

$$\mathrm{Var}_{\mathrm{exp}}[r] = \mathrm{Var}[r] - \sigma^2_{\mathrm{noise}} \ . \tag{2}$$

We estimated the variance of the observational noise by computing the variance of a neuron's response r_t in multiple trials t in which we presented the same stimulus x_j and subsequently taking the expectation E_j over all images,

$$\sigma^2_{\mathrm{noise}} = \mathrm{E}_j \left[\mathrm{Var}_t \left[r_t | x_j\right]\right] \ . \tag{3}$$

We removed data of neurons if the ratio between the explainable to total variance was below 0.15. The resulting dataset includes spike count data for 166 isolated neurons, with an average ratio of explainable to total variance of 0.285. These neurons were recorded at $1° - 3°$ eccentricities and had receptive field size diameters between 0.25° and 0.75°.

To keep the results of our models consistent and comparable to the gold standard baseline from Cadena et al. (2019), we down-sampled the images by a factor of two to train our models. Likewise, images were cropped symmetrically, keeping the 40×40 central pixels. This size covers all of the recorded neurons' receptive fields, with a slight variability in their spatial location. Furthermore, the stimuli light intensities across all pixels and all images were centered around zero and normalized to have unit standard deviation. Additionally, we used the same random dataset splits of Cadena et al. (2019) into training (64%), validation (16%) and testing (20%). We assessed our models' accuracy for a specific architecture or set of hyper-parameters in the validation set and we report performance on the test set. We consistently used the same split throughout our study.

4.2 Divisive normalization model

Our model consists of two parts, a nonlinear core and a linear readout (Section 2.1 and Fig. 1). The core (Fig. 1A) processes the input stimulus x by convolving it with 32 filters w_k of size $13\,\mathrm{px} \times 13\,\mathrm{px}$ without padding, defining a bank of features indexed by k. Subsequently, we

apply batch normalization without re-scaling (BN*) leading to responses of unit variance (Ioffe and Szegedy, 2015), followed by a rectified linear unit (ReLU) nonlinearity

$$f(\,\cdot\,) = \max(0,\,\cdot\,)\ . \tag{4}$$

Hence, the resulting 32 feature maps of size $28\,\mathrm{px} \times 28\,\mathrm{px}$ for the excitatory drive are given by

$$y_k = f(\mathrm{BN}^*(w_k * x))\ . \tag{5}$$

Many neurons perform similar computations but respond at different localized areas of the visual field. Those receptive fields are represented by the kernels w_k, which we implemented convolutionally to make use of this knowledge. Furthermore, the ReLU nonlinearity (Eq. 4) ensures that all feature maps are positive, $y_k \geq 0$, which is coherent to the biological interpretation of an excitatory drive.

The feature maps y_k are then normalized divisively to produce 32 output feature maps

$$z_l = \frac{y_l^{n_l}}{\sigma_l^{n_l} + \sum_k p_{kl}\,\langle y_k^{n_k}\rangle} \tag{6}$$

shared by all neurons. Here, all operations are element-wise and the scalar semi-saturation constant $\sigma_l \geq 0$ is learned from the data. To include normalization by other channels k, we first exponentiate the excitatory feature maps y_k by the scalar $n_k \geq 0$ element-wise, which is learned from the data as well. Subsequently, low-pass filtering is performed through average pooling in space with pool-size $5\,\mathrm{px} \times 5\,\mathrm{px}$, denoted by $\langle y_k^{n_k}\rangle$. We perform this pooling in order to achieve (approximate) phase invariance of the normalizing input without requiring a large number of filters with different phases. Subsequently, the results of the low-pass filtering are summed up, weighted by the normalization weights p_{kl}, and added into the denominator, resembling Eq. (1). Furthermore, the normalization weights are constrained to be non-negative, $p_{kl} \geq 0$. Together with $y_k \geq 0$ and $\sigma_l \geq 0$, this ensures that the denominator in Eq. (6) is non-negative, hence having a well-defined biological interpretation.

We converted the core's output feature maps z_l, shared by all neurons, to the activity of individual neurons via a readout for each of them (Fig. 1B). To do so, we factorized the readout into spatial readout weights $a_{uv,i} \geq 0$ and feature readout weights $b_{l,i} \geq 0$ that pick the relevant locations and features plus a bias q_i,

$$g_i = \sum_{u,v,l} (a_{uv,i}\, b_{l,i})\, z_{uvl} + q_i\ . \tag{7}$$

Here, u, v index space and i indexes neurons. This factorization is beneficial because it reduces the number of parameters in the readout. Also, we wanted to ensure that the readout does not model any complex computations, which we achieved by this factorization and the non-negativity of the readout weights. Additionally, we limited complexity by imposing a sparseness prior on both weights, because each neuron should only respond to its receptive field which is represented by a sparse spatial readout weight and should not mix many different features which corresponds to a sparse feature readout weight. The readout can, however, model a

complex cell (Hubel and Wiesel, 1962) by linearly combining multiple channels of the shared feature space.

We fitted an output nonlinearity to obtain the final prediction of a neuron's activity $\hat{r}_i$ as suggested by Cadena et al. (2019) from which we paraphrase the description here. Optimizing the output nonlinearity improves data-driven models, but has to be done carefully when simultaneously learning the shared core and the readout weights end-to-end. We therefore construct the output nonlinearity as

$$\hat{r}_i = \mathrm{ELU}^*(g_i)\, h(g_i) \ . \tag{8}$$

Here, ELU* is an exponential linear unit (ELU; Clevert et al. (2015)) that is shifted up and to the right by one unit to keep the resulting activity non-negative,

$$\mathrm{ELU}^*(x) = \mathrm{ELU}(x-1) + 1 \ , \tag{9}$$

$$\mathrm{ELU}(x) = \begin{cases} x & \text{if } x \geq 0 \\ \exp(x) - 1 & \text{otherwise} \ , \end{cases} \tag{10}$$

which is multiplied by a non-negative function h with trainable parameters α_j

$$h(x) = \sum_j t_j(x)\ \mathrm{e}^{\alpha_j} \ . \tag{11}$$

The basis functions $t_j(x)$ in the argument of h are tent functions leading to a piecewise linear function inside the exponential. The tent functions are defined as

$$t_j(x) = \min\left(\max\left(0, \frac{x - x_{j-1}}{\Delta x}\right), \max\left(0, \frac{x_{j+1} - x}{\Delta x}\right)\right) \ , \tag{12}$$

with interpolation points $x_j = -3, -2.82, ..., 6$ and $\Delta x = 0.18$. By regularizing the coefficients α_j with a L_2 and a smoothness penalty, the model is pushed towards using the ELU* nonlinearity unless the data provides strong evidence in favor of a different output nonlinearity, in which case the piecewise function h can be used to modify it.

To optimize our model's parameters, we maximized the log-likelihood of the model's predictions given the data. To do so, we assumed that neurons' spikes are produced by a Poisson process. Our model predicts the average spike count $\hat{r}$ of a neuron, hence the probability of observing r spikes in the experiment is

$$P(r|\hat{r}) = \frac{\hat{r}^r}{r!}\, \mathrm{e}^{-\hat{r}} \ . \tag{13}$$

From that follows the Poisson log-likelihood

$$\ln P(r|\hat{r}) = \sum_{i,j} (r_i(x_j) \ln \hat{r}_i(x_j) - \ln(r_i(x_j)!) - \hat{r}_i(x_j)) \tag{14}$$

for all neurons i and all stimuli x_j. A neuron's response $r_i \equiv r_i(x_j)$ depends on the stimulus x_j, which we suppress in our further notation for better readability. For implementation reasons,

we wanted to minimize the Poisson loss function

$$\mathcal{L}_{\text{Poisson}} = \sum_{i,j} (\hat{r}_i - r_i \ln \hat{r}_i) \ , \tag{15}$$

which is the negative of the Poisson log-likelihood (Eq. 14), where we omitted $\ln(r_i!)$ since this term does not depend on our model.

Furthermore, three terms regularizing the model's parameters were applied to the loss. We imposed a smoothness prior on the kernels w_k to ensure the spatial continuity of the predictors' receptive fields. The according penalty on the loss for not-smooth weights was determined with a Laplace filter L to be

$$\mathcal{L}_{\text{smooth}} = \sqrt{\sum_{u,v,k} (L * w_k)_{uv}^2} \ , \quad L = \begin{pmatrix} 0.25 & 0.5 & 0.25 \\ 0.5 & -3 & 0.5 \\ 0.25 & 0.5 & 0.25 \end{pmatrix} \ . \tag{16}$$

Due to their receptive fields, neurons only respond to a small, localized area of the visual field, which is why we imposed a sparsity regularizer on the spatial readout weights a_{uv}. Furthermore, neurons should only pool from a small set of feature maps to ensure that the readout does not perform complex computations. Thus we imposed a sparsity regularizer on the feature readout weights b_l as well. We achieved this by adding the L_1-norm of both weights

$$\mathcal{L}_{\text{sparse}} = \sum_i \sum_{u,v,l} |a_{uv}| \cdot |b_l| \tag{17}$$

to the loss function. As described by Cadena et al. (2019), we regularized the output nonlinearity by penalizing the sum of squares of the first and second finite differences of the weights α_j to keep the learnable exponential function $h(x)$ smooth and close to 1,

$$\mathcal{L}_{\text{out}} = \lambda_{\text{out}} \left(\sum_j (\alpha_j - \alpha_{j-1})^2 + \sum_j (2\alpha_j - \alpha_{j-1} - \alpha_{j+1})^2 \right) \ . \tag{18}$$

The final loss function to minimize with respect to our model's parameters is

$$\mathcal{L} = \mathcal{L}_{\text{Poisson}} + \lambda_{\text{smooth}} \, \mathcal{L}_{\text{smooth}} + \lambda_{\text{sparse}} \, \mathcal{L}_{\text{sparse}} + \lambda_{\text{out}} \, \mathcal{L}_{\text{out}} \ , \tag{19}$$

where λ_{smooth}, λ_{sparse} and λ_{out} are hyper-parameters which set the strength of the smoothness, the sparsity and the output nonlinearity regularizer, respectively.

4.3 Divisive normalization model extended to normalization from surround

To extend our DN model to capture normalization from the spatial surround of a unit's classical RF, we replaced the weighted sum accounting for normalization (Eq. 6) by a convolution that

also covers space, keeping the rest of the original DN model unchanged,

$$z_l = \frac{y_l^{n_l}}{\sigma_l^{n_l} + s_l} \ , \quad s_l = \sum_k p_{kl\cdot\cdot} * \langle y_k^{n_k} \rangle \ . \tag{20}$$

The new shared feature space z_l consist of all element-wise operations where the normalization feature maps s_l represent the strength by which the excitatory drive $y_l^{n_l}$ is normalized. The normalization feature maps are the result of a convolution between $\langle y_l^{n_l} \rangle$ and normalization pool kernels p_{kluv}. These kernels encode which features (indexed by k) are pooled over what spatial extent (indexed by u, v). Note that for an $u \times v = 1 \times 1$ convolutional kernel p, this is equal to the DN model without normalization from the surround (Eq. 6).

For a larger convolutional kernel p, the feature maps s have smaller spatial dimensions than the excitatory feature maps y due to the convolution without padding. To be able to perform the element-wise division, we symmetrically cropped the excitatory feature maps y so that the resulting feature maps had the same spatial dimensions as s. Additionally, we wanted to keep the complexity (number of parameters) of the linear readout constant for all the size choices of the normalization kernel p. To this end, we symmetrically cropped the input images to a size that corresponds – after a forward pass through our model – to a shared feature space of spatial dimensions $28\,\text{px} \times 28\,\text{px}$. Overall, this process enabled a fair comparison across all sizes of p.

To keep the kernel size of p computationally tractable, we used convolutions with a dilation factor of five to be able to pool from a relatively large extra-classical RF while using few parameters. If we would compute the convolution directly on the feature maps $y_k^{n_k}$, the dilation would lead to a situation in which some elements in the feature map $y_k^{n_k}$ are not accounted for by the convolution's inner product for one specific position of the convolutional kernel, i. e. one specific element in the suppression feature maps s_l. To consider all those elements in the inner product computation of the convolution, we introduced a preceding average pooling with a $5\,\text{px} \times 5\,\text{px}$ pool size (same as the dilation factor) and stride one. Then, all the information is pooled over and weighted by exactly one weight of the convolutional kernel. In this view, the pools of neighbouring weights of the dilated kernel have coinciding boundaries. So in addition to implementing shift invariance (see Section 4.2), the average pooling makes sure that we do not loose information for the extended DN model. Due to this pooling, a normalization kernel p of spatial size $3\,\text{px} \times 3\,\text{px}$ would spatially cover a normalization pool of size $15\,\text{px} \times 15\,\text{px}$. We further reduced the number of parameters by a rank-two decomposition separating spatial integration c and the feature weighting d,

$$p_{kluv} = \sum_{m=1}^{2} c_{luv,m} \cdot d_{kl,m} \ . \tag{21}$$

Like before, u, v index space and k indexes the features to pool from. We constrained c and d to be non-negative to make sure the denominator in Eq. (20) is strictly non-negative (recall that in Eq. (20) $\langle y_k^{n_k} \rangle \geq 0$). Our motivation to use two normalization pools (indexed by m) was to allow for both a localized feature-non-specific normalization pool and a feature-specific surround normalization as suggested by the standard model of DN. We investigated models with normalization kernel sizes of $1\,\text{px} \times 1\,\text{px}$, $3\,\text{px} \times 3\,\text{px}$, $5\,\text{px} \times 5\,\text{px}$ and $7\,\text{px} \times 7\,\text{px}$ which

spatially covered a five times larger normalization pool due to dilation. Those normalization pools covered visual angles of 0.49°, 0.77°, 1.06° and 1.34°, respectively.

4.4 Baseline models

4.4.1 Convolutional neural network

Since the divisive normalization computation in our model was completely learned from the data, we wanted to compare to a baseline model that is purely data-driven as well. For this, the current state-of-the-art model is a convolutional neural network with three layers (Cadena et al., 2019). Its first convolutional layer consists of a kernel with spatial size of 13 px × 13 px and for the second and third layer of size 3 px × 3 px each. All layers use 32 channels, batch normalization (Ioffe and Szegedy, 2015) and ELU nonlinearity (Clevert et al. (2015); Eq. 10). Similar to our model's architecture, the core part of the CNN model results in a nonlinear feature space shared by all neurons which is mapped to each neuron's activity with individual readout weights factorized in spatial and feature weightings. Sparseness of both of them is achieved by adding an L_1-penalty to the according loss function. This readout employs the same final output nonlinearity but differs from ours in having no non-negativity constraints on the weights.

4.4.2 Convolutional subunit model

Our convolutional subunit baseline model is structurally a one-layer convolutional neural network with multiple filters followed by a readout. It is exactly the same as our divisive normalization model (Section 4.2) but with the normalization function (Eq. 6) replaced by the identity function

$$z_l = \mathrm{id}(y_l) = y_l \ . \tag{22}$$

Hence, the only difference to our DN model is the lack of normalization. The shared feature space z_l consists of rectified outputs of linear filters (Eq. 5) which approximate simple cells. The subsequent readout can sum up those simple cell responses with additional weightings, enabling the model to approximate complex cells (Hubel and Wiesel, 1962). We trained the model with the same loss function (Eq. 19) as the divisive normalization model.

4.5 Number of learned parameters

In our model and the baseline models, parameters belong either to the core part that is shared by all neurons (Table 1) or to the readout part in which parameters are specific for each individual neuron.

Our DN model's first convolution consists of kernels w_k of spatial size 13 px × 13 px for 32 output channels and batch-normalization without re-scaling, adding 32 bias weights. To learn the normalization pool, 32 normalization weights p_{kl} were learned for each of the 32 output channels l. Additionally, we learned 32 semi-saturation constants σ_l and 32 exponents n_l (one

for each channel l). Hence, we get $13 \cdot 13 \cdot 32 + 32 + 32 \cdot 32 + 32 + 32 = 6\,528$ weights for the core. The resulting 32 feature maps are of spatial size of $28\,\text{px} \times 28\,\text{px}$ due to no padding in the convolution. Our readout is factorized in $28 \cdot 28$ spatial and 32 feature weights. The final output nonlinearity adds 50 weights due to the weighted sum of tent functions and one bias parameter. Thus, the readout consists of $28 \cdot 28 + 32 + 51 = 867$ parameters per neuron. Note additionally that all weights except for the convolution kernel and the bias are constrained to be non-negative, halving the according weight-space.

In the nonspecific divisive normalization model (Section 2.3.1) the normalization weights are constant for a given feature l, that is $p_{kl} = p_l$. Hence, it requires 32 instead of the $32 \cdot 32$ normalization weights of the full divisive normalization model. The other parts of the core remain the same, leading to $6\,528 - 32 \cdot 32 + 32 = 5\,536$ weights for the core. The readout and the number of 867 readout weights per neuron is the same for both models.

The spatially extended DN model covering the classical receptive field surround requires more parameters. As before, we get from the first convolution, bias weights, exponents and semi-saturation constants $13 \cdot 13 \cdot 32 + 32 + 32 + 32 = 5\,504$ weights. For each normalization pool component (indexed by m), the factorized convolution learning the normalization contains $32 \cdot 32 = 1\,024$ feature normalization weights $d_{kl,m}$ and $32\,u\,v$ spatial normalization weights $c_{luv,m}$. For the control experiments, we used 2 normalization pool components. So, the core consists in total of $5\,504 + 2 \cdot 1\,024 + 2 \cdot 32\,u\,v = 7\,552 + 64\,u\,v$ weights. For the fitted spatial normalization kernel sizes $u = v = 1, 3, 5, 7$ this results in 7 616, 8 128, 9 152 and 10 688 parameters for the core, respectively. The readout and the number of 867 readout weights per neuron is the same as in the other models.

The convolutional subunit model is the same as the divisive normalization model but with the divisive normalization function (Eq. 6) replaced by the identity function (Eq. 22). Hence, compared to the DN model, it saves $32 \cdot 32$ normalization weights p_{kl}, 32 semi-saturation constants σ_l and 32 exponents n_l, leading to $6\,528 - (32 \cdot 32 + 32 + 32) = 5\,440$ parameters for the core. The number of 867 readout parameters per neuron stays the same as for the divisive normalization model.

The three-layer CNN was trained on the same data with input stimuli of size $40\,\text{px} \times 40\,\text{px}$ (Cadena et al., 2019). The CNN's first convolution uses kernels with spatial size of $13\,\text{px} \times 13\,\text{px}$ and 32 channels as well as 32 biases due to batch normalization, leading to $13 \cdot 13 \cdot 32 + 32 = 5440$ parameters. The two subsequent convolutions use kernels of spatial size $3\,\text{px} \times 3\,\text{px}$ with 32 input and output channels as well as 32 biases each, $3 \cdot 3 \cdot 32 \cdot 32 + 32 = 9248$ parameters for each convolution. In total, the core consists of $5440 + 2 \cdot 9248 = 23\,936$ parameters. The readout is the same as in our models, consisting of 867 readout weights per neuron.

4.6 Hyper-parameter optimization

Our model's accuracy depends on several hyper-parameters. We set the initial learning-rate to 10^{-3} and used an early stopping training scheme: We evaluated the Poisson loss (Eq. 15) every 100 training steps and after ten iterations of no improvement we decayed the learning-rate by a factor of three. We repeated this four times to follow the same procedure used by Cadena et al.

(2019), because they find best validation set accuracy for this approach. For the filters w_k in the first convolution, we found that a size of $13\,\text{px} \times 13\,\text{px}$ was optimal, the same is true for the number of 32 channels.

The weights λ_{smooth} of the smoothness penalty (Eq. 16), λ_{sparse} of the readout sparsity penalty (Eq. 17) and λ_{out} of the output nonlinearity's regularization (Eq. 18) in the full loss function (Eq. 19) were extensively cross-validated using the validation set of our data (Section 4.1).

We randomly sampled the smooth-weight λ_{smooth} from a logarithmic uniform distribution in the interval $\left[10^{-9}, 10^{-3.5}\right]$ and the readout sparse-weight λ_{sparse} from a logarithmic uniform distribution in the interval $\left[10^{-9}, 10^{-4.5}\right]$ for all models. For the divisive normalization model constrained to the receptive field center and the subunit model, we drew the output nonlinearity regularization weight λ_{out} from a logarithmic uniform distribution in the interval $\left[10^{-8}, 10^{2}\right]$. For those two models, we sampled 1 000 runs. For the non-specific divisive normalization model and the DN models extended to the spatial surround, we narrowed down the relevant parameter space of λ_{out} and sampled them from a logarithmic uniform distribution in the interval $\left[10^{-5}, 10^{0}\right]$. Since this halved the search space of the hyper-parameters, we halved the number of samples drawn for those models. Thus, the density of sample points in the search space was the same for all models and should lead to a fair comparison between their performance.

4.7 Accuracy evaluation

4.7.1 Average correlation

For architecture search, hyper-parameter optimization and the selection of specific models for analysis we evaluated models' accuracies on the validation set with the Pearson correlation coefficient between the measured spike counts and our models' predictions, averaged over neurons. If the prediction for one neuron is constant, the according standard deviation is zero. Hence, the correlation coefficient was not computable due to division by zero. For those neurons, we set the correlation coefficient to zero before averaging. This average correlation measure does not consider observational noise (Eq. 3).

4.7.2 Fraction of explainable variance explained

For reporting accuracy values in this paper, we used the data's test set to compute the fraction of explainable variance explained (FEV)

$$\text{FEV} = 1 - \frac{\text{Var}_{\text{res}}[r]}{\text{Var}_{\text{exp}}[r]} \tag{23}$$

which utilizes the variance that is explainable in principle, $\mathrm{Var}_{\mathrm{exp}}[r]$ (Eq. 2), and the variance of the residuals corrected by the observation noise,

$$\mathrm{Var}_{\mathrm{res}}[r] = \sum_{i}^{N} (r_i - \hat{r}_i)^2 / N - \sigma^2_{\mathrm{noise}} \ . \tag{24}$$

This measure corrects for observation noise, which variance $\sigma^2_{\mathrm{noise}}$ we estimated with Eq. (3).

4.8 Evaluation of orientation-specific normalization

To analyse how the preferred orientation of the features being normalized depend on that of the features providing normalizing inputs (Fig. 4), we determined for each feature map whether it extracts oriented features and – if so – its preferred orientation. To do so, we windowed each convolutional kernel with a Gaussian window (standard deviation: 3 px), normalized it and then computed its 2D power spectrum (using the discrete Fourier transform with 64×64 samples). We then quantified how power spectral density is distributed across orientations by computing a mean resultant vector m given by:

$$m = \sum_{u,v \in R} F_{uv} e^{2i\phi}, \tag{25}$$

where F_{uv} is the Fourier transformed kernel, $R = \{(u, v) : 0.3 < \sqrt{u^2 + v^2} < 0.7\}$ contains all frequencies between 0.3 and 0.7 (with 1.0 being the Nyquist frequency), $\phi = \mathrm{atan2}(v, u)$ is the orientation, i the imaginary unit and the factor 2 in the complex exponential accounts for the fact that we are interested in orientation, which is periodic in 180° or π. If all power in a kernel is concentrated in one orientation, the mean resultant vector will be long, whereas an unoriented kernel will have a mean resultant vector near zero. Based on visual inspection of the kernels in model fits of the top 11 to 20 model runs (in terms of validation set accuracy), we found $|m| = 0.125$ to be a reasonable threshold for separating oriented from unoriented features and used it as a heuristic for further analyses of the best ten model runs, avoiding issues with post-hoc statistical testing.

To quantify how strong a feature l is normalized by other features k, we computed the average normalizing input, which is given as the expected value (over images) of the product $p_{kl} \cdot y_{kuv}(x)$ in Eq. (1). Since this normalization input depends on the stimulus, we computed its expected value of all images in the validation set. We removed the dependence on space by averaging over all locations within the feature map.

4.9 Control: All channels contribute to our model's prediction

To verify that all features contribute to normalization, we analyzed the readout feature weights for the best ten models (assessed in terms of performance on the validation set). However, there are two issues that prevent a direct comparison across models and neurons of the feature weightings. First, the factorization of the readout into spatial and feature weightings is not

unique: scaling the spatial weights (a in Eq. (7)) by a factor β whilst scaling the feature weights (b in Eq. (7)) by $1/\beta$ yields the same output limiting comparisons across neurons. Second, a similar exercise between the normalization weights p and the semi-saturation constant σ (Eq. 6) impedes comparison across models. To solve these issues, we normalized the feature readout weights across channels for this control analysis so that the resulting vectors for each neuron and model convey how much a certain channel contributes to predict a neuron's response compared to the other channels, making the feature readout weights comparable across neurons. Next, we averaged these weights across neurons to assess the importance of the channels in a model. Since these normalized feature readout weights were comparable across both neurons and models, we calculated a collective distribution of the averaged feature readout weights from the best ten models. To make sense of this distribution's absolute values, we evaluated its width in terms of the coefficient of variation, which is the standard deviation in units of the mean.

4.10 Implementation details

We used Tensorflow (Abadi et al., 2015) to implement models as well as Python, which we additionally used for data analysis. We optimized models with the Adam optimizer (Kingma and Ba, 2014) using mini-batches of size 256. In addition, we used the Python packages Numpy/Scipy (Walt et al., 2011), Pandas (McKinney, 2010), Matplotlib (Hunter, 2007), Seaborn (Waskom et al., 2017) and the tools Jupyter(Kluyver et al., 2016) and Docker (Merkel, 2014).

Acknowledgements

We thank Fabian H. Sinz, Felix A. Wichmann, David A. Klindt, Alexander Böttcher, Matthias Kümmerer, Dylan M. Paiton, Robert Geirhos, and Claudio Michaelis for valuable discussions. M.F.B. and S.A.C. thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS). The research was supported by the German Federal Ministry of Education and Research (BMBF) via the Competence Center for Machine Learning (FKZ 01IS18039A); the German Research Foundation (DFG) grant EC 479/1-1 (A.S.E.), the Collaborative Research Center (SFB 1233, Robust Vision) and the Cluster of Excellence "Machine Learning – New Perspectives for Science" (EXC 2064/1, project number 390727645); the Bernstein Center for Computational Neuroscience (FKZ 01GQ1002); the National Eye Institute of the National Institutes of Health under Award Numbers R01EY026927 (A.S.T.), DP1 EY023176 (A.S.T.), and NIH-Pioneer award DP1-OD008301 (A.S.T). This research was also supported by NEI/NIH Core Grant for Vision Research (EY-002520-37), NEI training grant T32EY00700140 (G.H.D) and F30EY025510 (E.Y.W.), DARPA grant N66001-17-C-4002, and the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/Interior Business Center (DoI/IBC) contract number D16PC00003. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/IBC, or the U.S. Government. The funders had no

role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests

The authors declare no competing interests.

References

M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. Tensorflow: Large-scale machine learning on heterogeneous distributed systems, 2015.

E. H. Adelson and J. R. Bergen. Spatiotemporal energy models for the perception of motion. *Journal of the Optical Society of America A*, 2: 284–299, 1985.

J. Antolík, S. B. Hofer, J. A. Bednar, and T. D. Mrsic-Flogel. Model constrained by visual hierarchy improves prediction of neural responses to natural scenes. *PLoS Computational Biology*, 12(6): e1004927, 2016.

J. Ballé, V. Laparra, and E. P. Simoncelli. End-to-end optimized image compression. In *International Conference on Learning Representations (ICLR2017)*, Toulon, France, April 2017.

H. B. Barlow. Possible principles underlying the transformations of sensory messages. In W. A. Rosenblith, editor, *Sensory Communication*, pages 217–234. MIT Press, Cambridge, Massachusetts, 1961.

E. Batty, J. Merel, N. Brackbill, A. Heitman, A. Sher, A. Litke, E. Chichilnisky, and L. Paninski. Multilayer recurrent network models of primate retinal ganglion cell responses. 2016.

C. Blakemore and E. A. Tobin. Lateral inhibition between orientation detectors in the cat's visual cortex. *Experimental Brain Research*, 15: 439–440, 1972.

A. B. Bonds. Role of inhibition in the specification of orientation selectivity of cells in the cat striate cortex. *Visual Neuroscience*, 2: 41–55, 1989.

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

D. A. Butts, C. Weng, J. Jin, J.-M. Alonso, and L. Paninski. Temporal precision in the visual pathway through the interplay of excitation and stimulus-driven suppression. *Journal of Neuroscience*, 31(31): 11313–11327, 2011.

S. A. Cadena, G. H. Denfield, E. Y. Walker, L. A. Gatys, A. S. Tolias, M. Bethge, and A. S. Ecker. Deep convolutional models improve predictions of macaque v1 responses to natural images. *PLoS Computational Biology*, 15(4): e1006897, 2019.

M. Carandini and D. J. Heeger. Normalization as a canonical neural computation. *Nature Reviews Neuroscience*, 13: 51–62, 2012.

M. Carandini, D. J. Heeger, and J. A. Movshon. Linearity and normalization in simple cells of the macaque primary visual cortex. *Journal of Neuroscience*, 17: 8621–8644, 1997.

M. Carandini, J. B. Demb, V. Mante, D. J. Tolhurst, Y. Dan, B. A. Olshausen, J. L. Gallant, and N. C. Rust. Do we know what the early visual system does? *Journal of Neuroscience*, 25: 10577–10597, 2005.

J. R. Cavanaugh, W. Bair, and J. A. Movshon. Nature and Interaction of Signals From the Receptive Field Center and Surround in Macaque V1 Neurons. *Journal of Neurophysiology*, 88: 2530–2546, 2002.

S. N. Chettih and C. D. Harvey. Single-neuron perturbations reveal feature-specific competition in V1. *Nature*, 567(7748): 334–340, 2019.

D.-A. Clevert, T. Unterthiner, and S. Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

R. Coen-Cagli, A. Kohn, and O. Schwartz. Flexible gating of contextual influences in natural vision. *Nature Neuroscience*, 18: 1648–1655, 2015.

G. C. DeAngelis, J. G. Robson, I. Ohzawa, and R. D. Freeman. Organization of suppression in receptive fields of neurons in cat visual cortex. *Journal of Neurophysiology*, 68: 144–163, 1992.

G. C. DeAngelis, R. D. Freeman, and I. Ohzawa. Length and width tuning of neurons in the cat's primary visual cortex. *Journal of Neurophysiology*, 71: 347–374, 1994.

G. H. Denfield, A. S. Ecker, T. J. Shinn, M. Bethge, and A. S. Tolias. Attentional fluctuations induce shared variability in macaque primary visual cortex. *Nature Communications*, 9(1): 2654, 2018.

T. C. Freeman, S. Durand, D. C. Kiper, and M. Carandini. Suppression without Inhibition in Visual Cortex. *Neuron*, 35: 759–771, 2002.

E. Froudarakis, P. Berens, A. S. Ecker, R. J. Cotton, F. H. Sinz, D. Yatsenko, P. Saggau, M. Bethge, and A. S. Tolias. Population code in mouse v1 facilitates readout of natural scenes through increased sparseness. *Nature Neuroscience*, 17(6): 851, 2014.

D. J. Heeger. Normalization of cell responses in cat striate cortex. *Visual Neuroscience*, 9: 181–197, 1992.

H. W. Heuer and K. H. Britten. Contrast dependence of response normalization in area mt of the rhesus macaque. *Journal of Neurophysiology*, 88(6): 3398–3408, 2002.

S. B. Hofer, H. Ko, B. Pichler, J. Vogelstein, H. Ros, H. Zeng, E. Lein, N. A. Lesica, and T. D. Mrsic-Flogel. Differential connectivity and response dynamics of excitatory and inhibitory neurons in visual cortex. *Nature Neuroscience*, 14: 1045–1052, 2011.

D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160: 106–154, 1962.

J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9 (3): 90–95, 2007.

S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

L. Itti, C. Koch, and J. Braun. Revisiting spatial vision: Toward a unifying model. *Journal of the Optical Society of America A*, 17(11): 1899–1917, 2000.

A. Iyer and J. Burge. The statistics of how natural images drive the responses of neurons. *Journal of Vision*, 19(13): 4–4, 2019.

S.-M. Khaligh-Razavi and N. Kriegeskorte. Deep supervised, but not unsupervised, models may explain it cortical representation. *PLoS Computational Biology*, 10: e1003915, 2014.

W. F. Kindel, E. D. Christensen, and J. Zylberberg. Using deep learning to probe the neural code for images in primary visual cortex. *Journal of Vision*, 19(4): 29–29, 2019.

D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

D. Klindt, A. S. Ecker, T. Euler, and M. Bethge. Neural system identification for large populations separating " what" and " where". In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 3506–3516. Curran Associates, Inc., 2017.

T. Kluyver, B. Ragan-Kelley, F. Pérez, B. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. Hamrick, J. Grout, S. Corlay, P. Ivanov, D. Avila, S. Abdalla, and C. Willing. Jupyter notebooks – a publishing format for reproducible computational workflows. In F. Loizides and B. Schmidt, editors, *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, pages 87 – 90. IOS Press, 2016.

J. M. McFarland, Y. Cui, and D. A. Butts. Inferring nonlinear neuronal computation based on physiologically plausible inputs. *PLoS Computational Biology*, 9(7): e1003143, 2013.

L. McIntosh, N. Maheswaranathan, A. Nayebi, S. Ganguli, and S. Baccus. Deep learning models of the retinal response to natural scenes. In *Advances in Neural Information Processing Systems*, pages 1369–1377, 2016.

W. McKinney. Data structures for statistical computing in python. In S. van der Walt and J. Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 51 – 56, 2010.

D. Merkel. Docker: Lightweight linux containers for consistent development and deployment. *Linux J.*, 2014(239), Mar. 2014.

M. C. Morrone, D. C. Burr, and L. Maffei. Functional implications of cross-orientation inhibition of cortical visual cells. I. Neurophysiological evidence. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 216: 335–354, 1982.

O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115: 211–252, 2015.

N. C. Rust, O. Schwartz, J. A. Movshon, and E. P. Simoncelli. Spatiotemporal Elements of Macaque V1 Receptive Fields. *Neuron*, 46: 945–956, 2005.

T. Sawada and A. A. Petrov. The divisive normalization model of V1 neurons: A comprehensive comparison of physiological data and model predictions. *Journal of Neurophysiology*, 118: 3051–3091, 2017.

H. H. Schütt and F. A. Wichmann. An image-computable psychophysical spatial vision model. *Journal of Vision*, 17: 12–12, 2017.

O. Schwartz and E. P. Simoncelli. Natural signal statistics and sensory gain control. *Nature Neuroscience*, 4: 819–825, 2001.

E. P. Simoncelli, L. Paninski, J. Pillow, and O. Schwartz. Characterization of neural responses with stochastic stimuli. *The Cognitive Neurosciences*, 3: 327–338, 2004.

F. Sinz and M. Bethge. The conjoint effect of divisive normalization and orientation selectivity on redundancy reduction. In *Advances in Neural Information Processing Systems 21*, pages 1521–1528, 2008.

F. Sinz, A. S. Ecker, P. Fahey, E. Walker, E. Cobos, E. Froudarakis, D. Yatsenko, Z. Pitkow, J. Reimer, and A. Tolias. Stimulus domain transfer in recurrent models for large scale cortical population prediction on video. In *Advances in Neural Information Processing Systems 31 (NeurIPS)*, pages 7199–7210. 2018.

J. Touryan, G. Felsen, and Y. Dan. Spatial structure of complex cell receptive fields measured with natural images. *Neuron*, 45(5): 781–791, 2005.

B. Vintch, J. A. Movshon, and E. P. Simoncelli. A Convolutional Subunit Model for Neuronal Responses in Macaque V1. *Journal of Neuroscience*, 35: 14829–14841, 2015.

M. J. Wainwright, O. Schwartz, and E. P. Simoncelli. Natural image statistics and divisive normalization: Modeling nonlinearity and adaptation in cortical neurons. *Probabilistic Models of the Brain: Perception and Neural Function*, 2002.

E. Y. Walker, F. H. Sinz, E. Cobos, T. Muhammad, E. Froudarakis, P. G. Fahey, A. S. Ecker, J. Reimer, X. Pitkow, and A. S. Tolias. Inception loops discover what excites neurons most using deep predictive models. *Nature Neuroscience*, 22(12): 2060–2065, 2019.

S. v. d. Walt, S. C. Colbert, and G. Varoquaux. The numpy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2): 22–30, 2011.

M. Waskom, O. Botvinnik, D. O'Kane, P. Hobson, S. Lukauskas, D. C. Gemperline, T. Augspurger, Y. Halchenko, J. B. Cole, J. Warmenhoven, J. de Ruiter, C. Pye, S. Hoyer, J. Vanderplas, S. Villalba, G. Kunter, E. Quintero, P. Bachant, M. Martin, K. Meyer, A. Miles, Y. Ram, T. Yarkoni, M. L. Williams, C. Evans, C. Fitzgerald, Brian, C. Fonnesbeck, A. Lee, and A. Qalieh. mwaskom/seaborn: v0.8.1 (september 2017), Sept. 2017.

B. Willmore, R. J. Prenger, M. C.-K. Wu, and J. L. Gallant. The berkeley wavelet transform: a biologically inspired orthogonal wavelet transform. *Neural Computation*, 20(6): 1537–1564, 2008.

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

Y. Zhang, T. S. Lee, M. Li, F. Liu, and S. Tang. Convolutional neural network models of V1 responses to complex patterns. *Journal of Computational Neuroscience*, pages 1–22, 2018.

P. Znamenskiy, M.-H. Kim, D. R. Muir, M. F. Iacaruso, S. B. Hofer, and T. D. Mrsic-Flogel. Functional selectivity and specific connectivity of inhibitory neurons in primary visual cortex. *bioRxiv*, page 294835, 2018.