

# Inception loops discover what excites neurons most using deep predictive models

Edgar Y. Walker <sup>1,2,8\*</sup>, Fabian H. Sinz <sup>1,2,3,4,8\*</sup>, Erick Cobos <sup>1,2</sup>, Taliah Muhammad <sup>1,2</sup>, Emmanouil Froudarakis <sup>1,2</sup>, Paul G. Fahey <sup>1,2</sup>, Alexander S. Ecker <sup>1,3,5,6</sup>, Jacob Reimer <sup>1,2</sup>, Xaq Pitkow <sup>1,2,7</sup> and Andreas S. Tolias <sup>1,2,7\*</sup>

**Finding sensory stimuli that drive neurons optimally is central to understanding information processing in the brain. However, optimizing sensory input is difficult due to the predominantly nonlinear nature of sensory processing and high dimensionality of the input. We developed ‘inception loops’, a closed-loop experimental paradigm combining *in vivo* recordings from thousands of neurons with *in silico* nonlinear response modeling. Our end-to-end trained, deep-learning-based model predicted thousands of neuronal responses to arbitrary, new natural input with high accuracy and was used to synthesize optimal stimuli—most exciting inputs (MEIs). For mouse primary visual cortex (V1), MEIs exhibited complex spatial features that occurred frequently in natural scenes but deviated strikingly from the common notion that Gabor-like stimuli are optimal for V1. When presented back to the same neurons *in vivo*, MEIs drove responses significantly better than control stimuli. Inception loops represent a widely applicable technique for dissecting the neural mechanisms of sensation.**

Since the work of Adrian and Bronk<sup>1</sup> and Hartline<sup>2</sup>, finding stimuli that optimally drive neurons has been fundamental for understanding information processing in the brain. In linear systems, linear filters elicit responses optimally; for instance, linear-nonlinear (LN) models with center-surround filters have high predictive power in the retina<sup>3</sup> and these patterns also strongly drive retinal activity. However, the response selectivity of many cortical neurons is inherently nonlinear, and even in V1, the predictive power of LN or energy models is low, especially for responses to natural stimuli<sup>4–6</sup>. Accordingly, identifying optimal sensory input for neurons with nonlinear sensitivity is difficult because of the intrinsically high-dimensional space of possible images. Proposed active learning approaches<sup>7–10</sup> are impractical because experimental constraints limit the number of responses that can be measured from any single cell or restrict the dimensionality of the stimulus space. Model-driven stimulus optimization, on the other hand, requires functional models that can faithfully predict the responses of neurons to arbitrary stimuli, including natural images, to guide an unrestricted search through a high dimensional space. Recently, deep learning-based models have set new standards in predicting cortical responses to natural images<sup>5,6,11–14</sup>. In the present study, we used end-to-end trained, deep-learning-based models to synthesize and search for optimal stimuli in silico that we verified back in the brain.

## Results

We designed a closed-loop experimental paradigm we call an inception loop that combines *in vivo* recordings with *in silico* modeling to synthesize stimuli that evoke a desired response that we confirm *in vivo* (Fig. 1a). Briefly, on day 1 of an inception loop experiment, we recorded the neural responses of large neuronal

populations to thousands of natural images, trained a convolutional neural network (CNN) to predict these responses based on the presented images<sup>6,11–13</sup>, and optimized images to maximize the model responses of selected model neurons<sup>15</sup>. Over subsequent days, we presented these tailored images to the corresponding neurons in the brain to test whether they indeed produced the strongest responses among all control stimuli.

We recorded the responses to natural images of over 2,000 excitatory neurons in layer 2/3 (L2/3) of the V1 (V1 L2/3) for each of five awake mice using two-photon imaging with a wide-field mesoscope (Fig. 1b)<sup>16</sup>. Before each functional imaging session, we recorded a high-resolution, anatomical, three-dimensional stack (Fig. 1c). Later we registered all recording planes into each of these stacks to locate the neurons of interest across multiple days (Supplementary Figs. 1 and 2; see Methods for details).

On the first day we collected the population responses to a set of 5,000 unique natural images, which we used to fit a predictive model of neural responses to visual stimuli (Fig. 1b). Another set of 100 images, repeated 10 times each, was used as the test set for the model and to evaluate response reliability. Each image was shown for 500 ms and we extracted each neuron’s response by integrating the deconvolved fluorescence trace over a time window of 50–550 ms after image onset. We ignored temporal dynamics and focused purely on the spatial response characteristics of the neurons.

Next, we trained a deep CNN to predict the recorded responses (Fig. 1d). Recent work established deep CNNs as state-of-the-art models for neural response prediction, outperforming classical models of V1 such as LN, subunit or energy models<sup>13,17–19</sup>. We used a core network that consisted of three convolutional layers shared among all recorded neurons, followed by a neuron-specific linear

<sup>1</sup>Center for Neuroscience and Artificial Intelligence, Baylor College of Medicine, Houston, TX, USA. <sup>2</sup>Department of Neuroscience, Baylor College of Medicine, Houston, TX, USA. <sup>3</sup>Bernstein Center for Computational Neuroscience, University of Tübingen, Tübingen, Germany. <sup>4</sup>Institute for Bioinformatics and Medical Informatics, University of Tübingen, Tübingen, Germany. <sup>5</sup>Centre for Integrative Neuroscience, University of Tübingen, Tübingen, Germany. <sup>6</sup>Institute for Theoretical Physics, University of Tübingen, Tübingen, Germany. <sup>7</sup>Department of Electrical and Computer Engineering, Rice University, Houston, TX, USA. <sup>8</sup>These authors contributed equally: Edgar Y. Walker, Fabian H. Sinz. \*e-mail: eywalker@bcm.edu; fabian.sinz@uni-tuebingen.de; astolias@bcm.edu

readout stage. The network also accounted for eye position and behavioral state (pupil dilation and running) of the animal<sup>6</sup>, which we could measure but not control experimentally (see Methods for details on the model). In line with earlier work, we found that the CNN model outperformed an LN model, which had the same architecture but all nonlinearities removed from the core. Pooled over all mice and all neurons, the CNN model achieved 77.8% relative to the achievable performance (oracle) given the noise ceiling in the recordings (Fig. 1e and Methods). We also verified that the CNN model was a nontrivial, nonlinear extension of an LN model (Supplementary Fig. 3).

Our final step for day 1 was to obtain the MEIs for a subset of 150 neurons whose responses were reliable and reasonably well-predicted by both the CNN and the LN model. We achieved this via a simple optimization procedure<sup>15</sup>: to find the image that maximally excites a target neuron, we started with a random image and performed regularized gradient ascent until convergence (Fig. 1f and Methods).

The resulting MEIs deviated substantially from the widespread notion of Gabor-shaped V1 receptive fields (RFs). They exhibited complex spatial features such as sharp corners, checkerboard patterns, irregular pointillist textures and a variety of curved strokes (Fig. 2a and Supplementary Fig. 4). The shape of the optimized MEIs was also stable against different initializations of the optimization (Supplementary Figs. 1 and 5) and across days (Supplementary Fig. 1). The former indicates that there are not many obvious invariances in the selected V1 cells and that the optimization procedure does not suffer from local maxima.

We confirmed that MEIs reflected neuronal selectivity by presenting the generated MEIs back to the same neurons during the next days. Before presentation, we matched their mean luminance and root mean square (RMS) contrast to a common value. MEIs were highly specific, consistently eliciting higher activity in their target neurons (Fig. 2b and Supplementary Fig. 6) and evoked sparse responses with few images activating a neuron above baseline (Fig. 2c). We also confirmed that model predictions correlated highly with observed responses (Fig. 2c,d; average Pearson correlation = 0.68).

We repeated the same optimization for the LN model to get a corresponding estimate of the linear RF for each target neuron. RFs often appeared slightly smaller than the MEIs for the same neuron and notably lacked the higher-frequency details present in MEIs (Supplementary Fig. 7). While some RFs exhibited an atypical structure, others looked qualitatively similar to Gabor filters (Fig. 3a and Supplementary Fig. 8), consistent with conventional wisdom about V1.

When both MEIs and RFs were presented during the same experiment, MEIs evoked significantly stronger activity in their associated neurons than the linear RF in most cells (Fig. 4a). To exclude

the possibility that MEIs represent better linear models than RFs, we compared the predictive performance of MEIs and RFs when used as a linear filter. As expected, predicted responses of the linear RF model correlated better with neuronal responses (Supplementary Fig. 9). This does not mean that RFs predict neuronal responses better (see Fig. 1e). On the contrary, it highlights the inherent nonlinearity of neuronal processing in mouse V1 exploited by MEIs to produce higher activations and shows that they should not be thought of as linear filters. To further emphasize that point, we fitted an LN model to the responses of the CNN model and found that the RFs computed from the brain's neurons and the RFs of this linearized CNN model were virtually identical (Supplementary Fig. 10), implying that the nonlinear nature of the CNN model is responsible for the enhanced ability of MEIs to drive cells.

Comparing MEIs to linear RFs is a stringent test of current standard models of V1, including nonlinear extensions such as energy models<sup>20</sup>. For instance, the optimal stimulus for an energy model is any linear combination of the two filters in the quadrature pair, that is, a Gabor; based on these models, the linear RF and MEI should be identical. To directly demonstrate that Gabor-like stimuli are not the optimal stimuli for these mouse V1 cells, we identified the optimal Gabor for each model neuron in the CNN using a grid search over a fine-grained parameter space (Fig. 3b and Supplementary Fig. 11; see Methods). In an additional inception loop experiment, we found that MEIs drive the responses of their target neurons significantly stronger than the optimal Gabor stimuli (Fig. 4b).

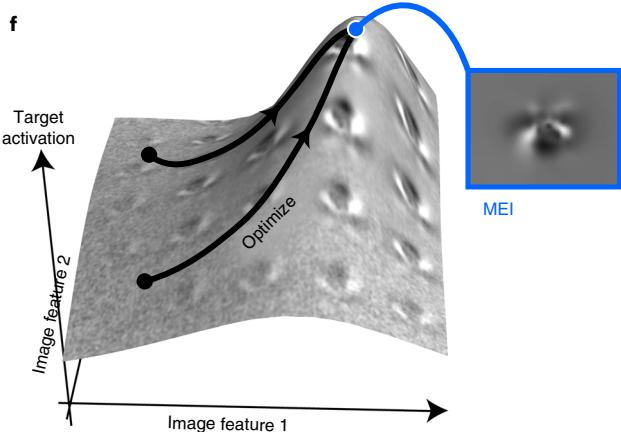
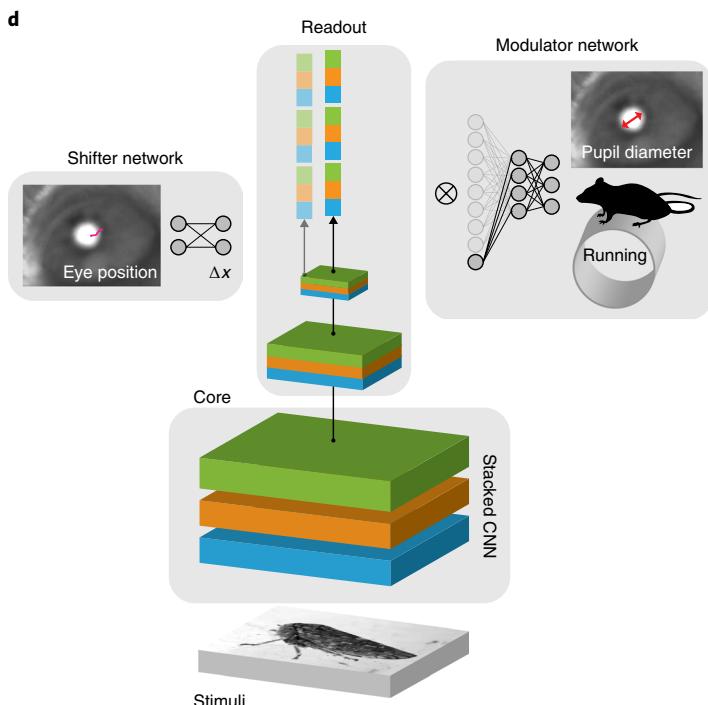
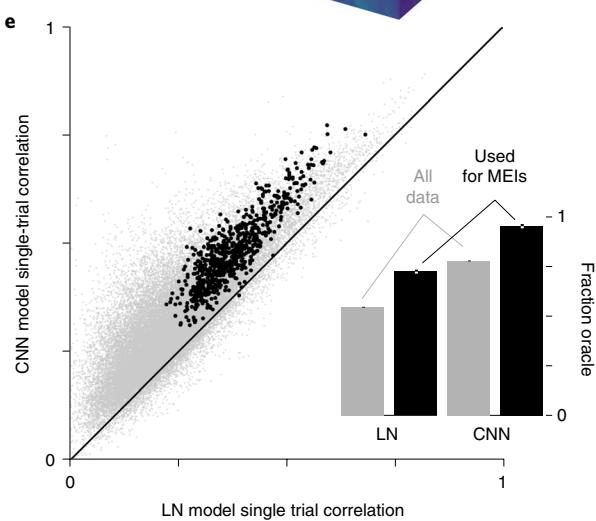
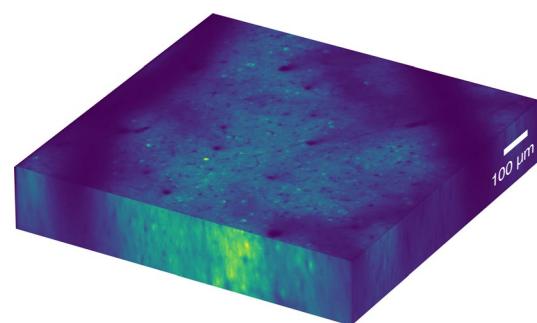
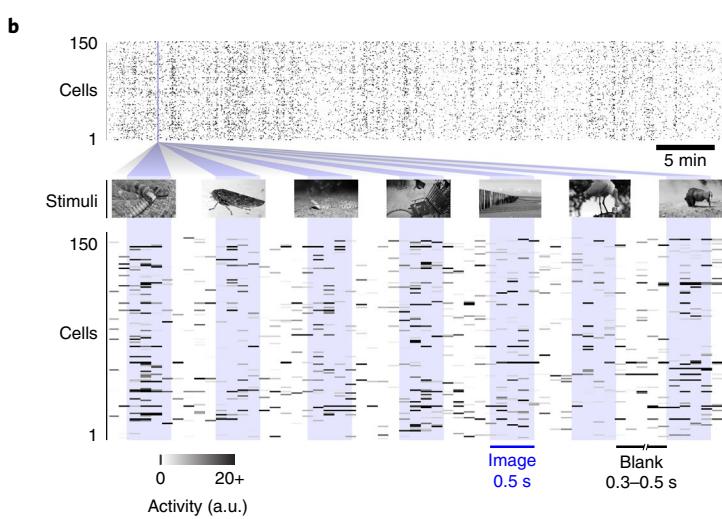
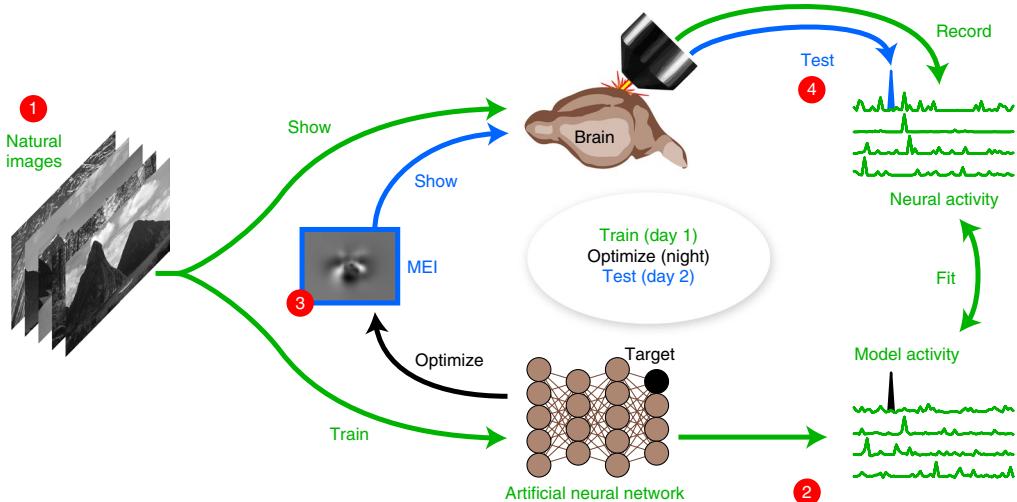
We next wanted to determine how the activation elicited by the MEIs compared to the activations in response to optimal natural images. We screened a completely new set of 5,000 natural images, not used in any prior experiment, to find the most exciting natural images when restricted to match the size, location and contrast of the MEIs. We found that (1) the most exciting masked natural image patches and the MEIs show a striking perceptual resemblance even when searching over only 5,000 images (Fig. 3b and Supplementary Fig. 11) and (2) MEIs still drive biological neurons better than the corresponding most exciting masked natural images (Fig. 4c). While (2) is another strong test for our model, (1) suggests that the features to which neurons are responsive are remarkably pervasive in natural images. Finally, we found that the full-field counterpart of the best masked natural images elicits a weaker response than MEIs (Fig. 4d). Few masked natural images evoked strong responses, so the distribution of good stimuli was sparse over our ensemble of natural images (only 1.6% of images produced activations above half that of the MEI response and only 0.04% above 0.75 of it) (Fig. 4e).

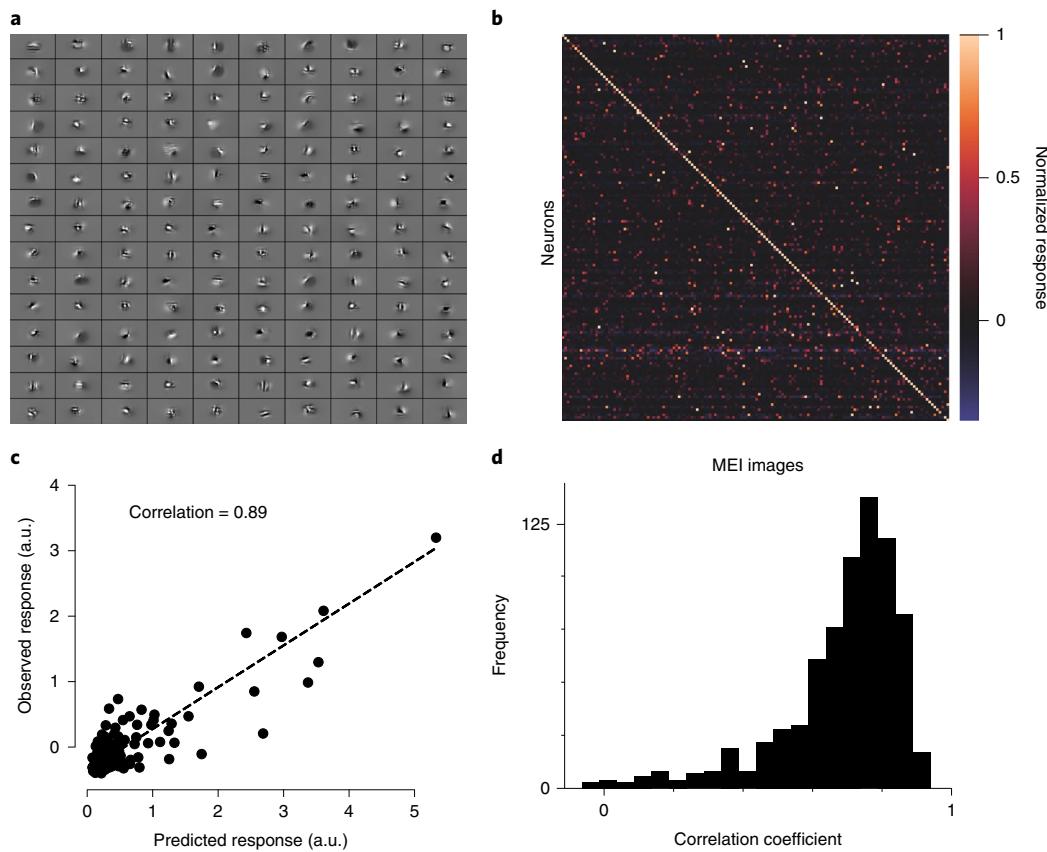
## Discussion

Our work shows that high-performing, end-to-end trained, black-box models of the visual system generalize and can make *in silico*

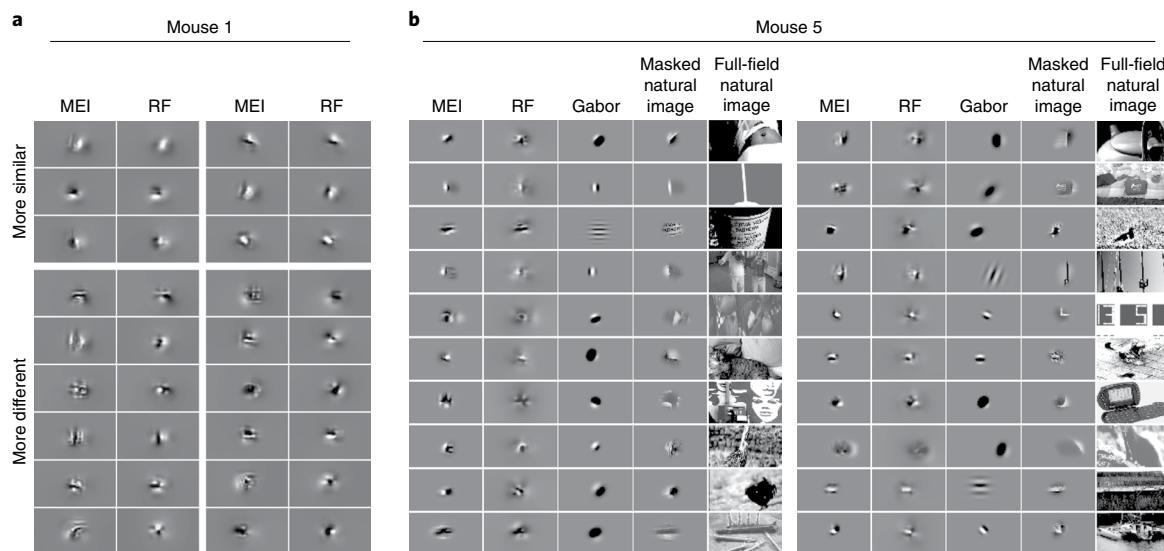
**Fig. 1 | Experimental paradigm and model.** **a**, Schematic of an inception loop. On day 1 (green) we showed sequences of natural images to a mouse and recorded neural activity by two-photon calcium imaging. Overnight (black), we trained linear models and CNNs to reproduce those measured neural responses and generated MEIs for each target neuron. On day 2 (blue) we showed these MEIs from linear and nonlinear models back to the same neurons in the brain and compared their responses. **b**, We presented 5,100 unique natural images to an awake mouse for 500 ms, interleaved with gray screen gaps of random length between 300 and 500 ms. A subset of 100 images were repeated 10 times each to estimate the reliability of neuronal responses (see Methods). Neuronal activity was recorded at 8 Hz in V1 L2/3 using a wide-field two-photon microscope. a.u., arbitrary unit. **c**, To identify the same cells across days, structural stacks were recorded each day. **d**, CNN trained to predict neuronal responses. Our network consisted of a core computing nonlinear features from the image, a readout predicting the neuronal responses from these features, a shifter accounting for eye movements by predicting a global gaze shift,  $\Delta x$  and  $\Delta y$ , and a modulator predicting an adaptive gain for each neuron based on behavioral variables (see Methods). **e**, CNN versus LN model performance. Each point denotes the correlation between the model predictions and single-trial responses. The CNN model significantly outperforms the LN model (two-tailed paired *t*-test,  $t(37378)=224.98$  with  $P < 10^{-9}$ ) over a total of  $n=37,379$  neurons pooled across five mice. The black points depict the performance for neurons used to generate MEIs ( $n=750$  neurons). Data from all mice are combined. Inset: performances for models without shifter and modulator signals, relative to an upper bound estimated from repeated presentations of identical stimuli, termed 'oracle'. The values are fraction oracle performances averaged across all neurons, with the error bars depicting the s.e.m. **f**, Illustration of the optimization over all possible images. The vertical axis represents activation of a model neuron as a function of two example image dimensions. The black curves depict optimization trajectories converging to the same MEI from different initializations.

a

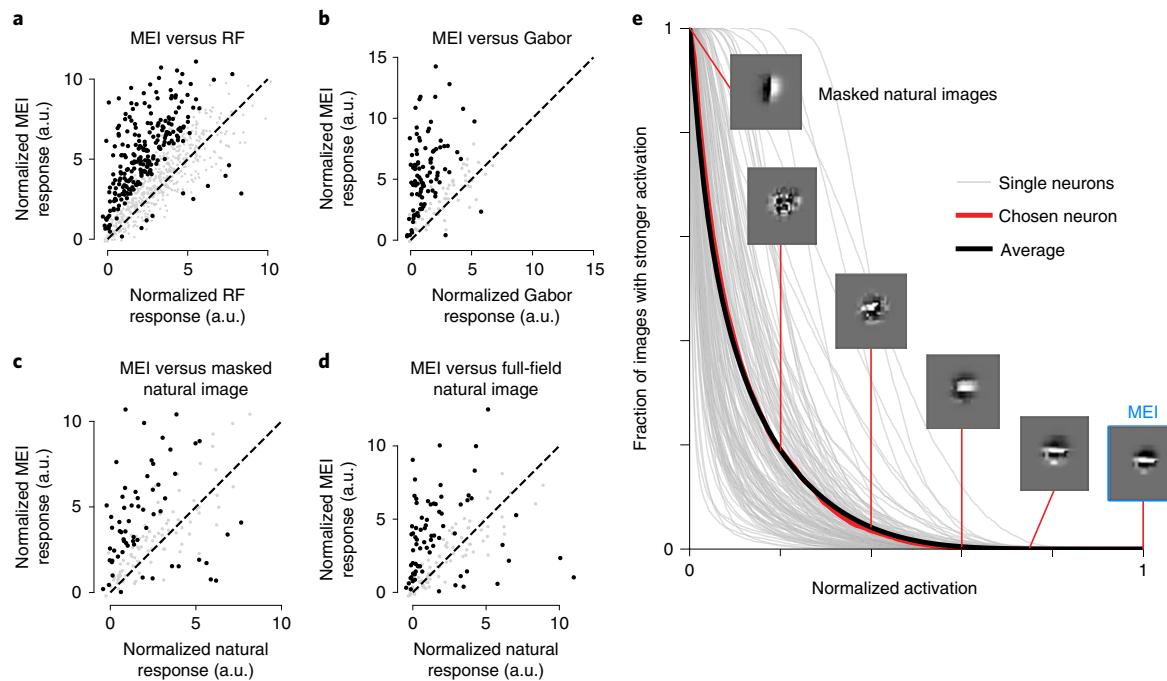




**Fig. 2 | MEIs.** **a**, Examples of MEIs for one mouse (mouse 1). MEIs exhibit complex, high-frequency features. (See Supplementary Fig. 4 for examples from other mice.) **b**, MEIs activate neurons with high specificity. The confusion matrix shows the responses of each neuron to the MEIs of all neurons. The responses of each neuron were normalized and pooled across days, and each row was scaled so the maximum response across all images equals 1. Data shown for mouse 1. **c**, Predicted versus observed responses of one example neuron (from mouse 1) to all 150 MEIs presented to a single cell (Pearson correlation = 0.89). **d**, Single-trial Pearson correlations between model predictions and actual responses to presented MEIs for all neurons (750) across five mice selected for MEI generation (average Pearson correlation coefficient = 0.68).



**Fig. 3 | Comparison of MEIs and other types of stimuli.** **a**, Examples of MEIs and RFs from one mouse (mouse 1). The images are grouped based on their similarity, measured by correlation between RF and MEI within a center mask. Strongly correlated MEIs and RFs are shown in the top block. **b**, Examples of MEIs and other control stimuli. MEIs, RFs, best Gabor filters (Gabor), best masked natural images and full-field natural images ('unmasked' version of the best masked natural image) are shown for 30 neurons of mouse 5. (See Supplementary Fig. 11 for the images for the rest of the neurons.).



**Fig. 4 | Neurons respond more to MEIs than other types of stimuli.** **a-d**, Each point corresponds to the normalized activity of a single neuron, averaged over repeats, in response to its MEI versus its RF (**a**), best Gabor filter (**b**), best masked natural image (**c**) or full-field natural image (**d**). Neurons with a significant difference in their mean responses are colored black ( $P < 0.05$ , two-tailed Welch's  $t$ -test with 55.5, 28.9, 32.0 and 30.3 average d.f., respectively). MEIs activated their target neurons significantly stronger than their corresponding RF (two-sided Wilcoxon signed-rank test,  $W = 47139$ ,  $P < 10^{-9}$ ), Gabor filter (two-sided Wilcoxon signed-rank test,  $W = 579$ ,  $P < 10^{-9}$ ), masked natural image (two-sided Wilcoxon signed-rank test,  $W = 2271$ ,  $P < 10^{-9}$ ) and full-field natural image (two-sided Wilcoxon signed-rank test,  $W = 2911$ ,  $P = 2.43 \times 10^{-7}$ ). **a**, Data are pooled over 8 scans from 5 mice, displaying a total of 750 neurons; 569 neurons showed stronger response to their MEIs than their RFs, of which 228 were statistically significant. In contrast, only ten had a statistically significantly stronger response to RFs, consistent with random choice. **b-d**, Each plot corresponds to a single closed loop scan from mouse 5. **b**, We found that 131 of 150 neurons showed a stronger response to MEIs (93 statistically significant); 2 responded more strongly to Gabor filters with statistical significance. **c**, We found that 114 of 150 neurons showed a stronger response to MEIs (52 were statistically significant); 11 responded more strongly to its best masked natural image with statistical significance. **d**, We found that 102 of 150 neurons showed a stronger response to MEIs (62 were statistically significant); 11 responded more strongly to its full-field natural image with statistical significance. **e**, Neuron responses to natural images are sparse and smaller than those to MEIs. The gray lines show the fraction out of 5,000 images that elicit a given activation or higher for 150 model target neurons in mouse 5; black is the average. Responses from each cell are divided by the response to its MEI; on average, 1.6% of images produced activations above half and only 0.04% above 0.75 of the MEI activation. For a representative cell (in red), we show images at different activation levels, along with its MEI (blue box).

inferences about nontrivial computational properties of V1 neurons. We find that even mouse V1 neurons prefer features that are more complex than the classical oriented edges described by Hubel and Wiesel<sup>21</sup> and predicted by many theories of early visual processing<sup>22</sup>. Intriguingly, searching for optimal natural image patches in as few as 5,000 images often yielded natural images that exhibited a striking perceptual similarity to MEIs, showing that the perceptual attributes of MEIs occur often in natural scenes. These complex feature selectivities in V1 could provide a faster, albeit less general, method to extract task-relevant causal variables from natural inputs. This might benefit small animals like mice, whose size may impose stringent constraints on computation<sup>23</sup>.

Strong predictive models allow for a nearly unlimited number of in silico experiments that can be tailored to individual neurons or populations. This is especially important when studying high-dimensional tuning properties, such as invariances<sup>17,24</sup> or equivariances<sup>25</sup>: searching directly for these dimensions in the brain is slow, costly and unlikely to reveal relevant stimulus manifolds. Using flexible neural network models to optimize stimuli has been proposed before<sup>18,26,27</sup>. However, given that deep neural networks do not necessarily generalize well beyond the typical statistics of their training set, it is not clear *a priori* whether in silico synthesized MEIs affect *in vivo* responses as predicted. Therefore, predictions derived from

these models require experimental verification. Our current work with inception loops shows that such verification is indeed feasible.

Inception loops provide many opportunities for future neuroscience studies. We performed this experiment in V1 since it constitutes a particularly rigorous test because of the large existing literature on V1 RF structure. This approach might prove even more useful for revealing feature selectivities in extrastriate or nonvisual areas, about which we currently understand much less<sup>28–31</sup>. Although we executed the inception loop once, further passes could test stimuli neighboring the MEI to refine estimates of tuning and invariance in a focused, efficient way. Combining emerging neurotechnologies with modeling and machine learning in an inception loop would provide a powerful tool to control, probe and evaluate brain transformations that will probably lead to a much richer understanding of neural computation.

#### Online content

Any methods, additional references, Nature Research reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information, details of author contributions and competing interests, and statements of data and code availability are available at <https://doi.org/10.1038/s41593-019-0517-x>.

Received: 11 January 2019; Accepted: 16 September 2019;  
Published online: 4 November 2019

## References

- Adrian, E. D. & Bronk, D. W. The discharge of impulses in motor nerve fibres: Part I. Impulses in single fibres of the phrenic nerve. *J. Physiol.* **66**, 81–101 (1928).
- Hartline, H. K. The response of single optic nerve fibers of the vertebrate eye to illumination of the retina. *Am. J. Physiol.* **121**, 400–415 (1938).
- Chichilnisky, E. J. A simple white noise analysis of neuronal light responses. *Network* **12**, 199–213 (2001).
- Olshausen, B. A. & Field, D. J. in *Problems in Systems Neuroscience* (eds Sejnowski, T. J. & van Hemmen, L.) 182–211 (Oxford Univ. Press, 2004).
- Antolík, J., Hofer, S. B., Bednar, J. A. & Mrsic-flogel, T. D. Model constrained by visual hierarchy improves prediction of neural responses to natural scenes. *PLoS Comput. Biol.* **12**, e1004927 (2016).
- Sinz, F. et al. Stimulus domain transfer in recurrent models for large scale cortical population prediction on video. In *Proc. Advances in Neural Information Processing Systems 31* (eds Bengio, S. et al.) 7199–7210 (Curran Associates, 2018).
- Harth, E. & Tzanakou, E. ALOPEX: a stochastic method for determining visual receptive fields. *Vision Res.* **14**, 1475–1482 (1974).
- Földiák, P. Stimulus optimisation in primary visual cortex. *Neurocomputing* **38–40**, 1217–1222 (2001).
- Paninski, L., Pillow, J. & Lewi, J. in *Computational Neuroscience: Theoretical Insights into Brain Function* (eds Cisek, P. et al.) 493–507 (Elsevier, 2007).
- Benda, J., Gollisch, T., Machens, C. K. & Herz, A. V. From response to stimulus: adaptive sampling in sensory physiology. *Curr. Opin. Neurobiol.* **17**, 430–436 (2007).
- Yamins, D. L. K. & DiCarlo, J. J. Using goal-driven deep learning models to understand sensory cortex. *Nat. Neurosci.* **19**, 356–365 (2016).
- Cadieu, C. F. et al. Deep neural networks rival the representation of primate IT cortex for core visual object recognition. *PLoS Comput. Biol.* **10**, e1003963 (2014).
- Klindt, D., Ecker, A. S., Euler, T. & Bethge, M. Neural system identification for large populations separating “what” and “where”. *Adv. Neural Inf. Process. Syst.* **30**, 3506–3516 (2017).
- McIntosh, L. T., Maheswaranathan, N., Nayebi, A., Ganguli, S. & Baccus, S. A. Deep learning models of the retinal response to natural scenes. *Adv. Neural Inf. Process. Syst.* **29**, 1369–1377 (2016).
- Erhan, D. & Bengio, Y. & Courville, A. & Vincent, P. Visualizing higher-layer features of a deep network. *Technical Report 1341* (University of Montreal, 2009).
- Sofroniew, N. J., Flickinger, D., King, J. & Svoboda, K. A large field of view two-photon mesoscope with subcellular resolution for *in vivo* imaging. *eLife* **5**, e14472 (2016).
- Cadena, S. A. et al. Deep convolutional models improve predictions of macaque V1 responses to natural images. *PLoS Comput. Biol.* **15**, e1006897 (2019).
- Kindel, W. F., Christensen, E. D. & Zylberberg, J. Using deep learning to probe the neural code for images in primary visual cortex. *J. Vis.* **19**, 29 (2019).
- Zhang, Y., Lee, T. S., Li, M., Liu, F. & Tang, S. Convolutional neural network models of V1 responses to complex patterns. *J. Comput. Neurosci.* **46**, 33–54 (2019).
- Adelson, E. H. & Bergen, J. R. Spatiotemporal energy models for the perception of motion. *J. Opt. Soc. Am. A* **2**, 284–299 (1985).
- Hubel, D. H. & Wiesel, T. N. Receptive fields of single neurones in the cat's striate cortex. *J. Physiol.* **148**, 574–591 (1959).
- Simoncelli, E. P. & Olshausen, B. A. Natural image statistics and neural representation. *Annu. Rev. Neurosci.* **24**, 1193–1216 (2001).
- Lindsey, J., Ocko, S. A., Ganguli, S. & Deny, S. A unified theory of early visual representations from retina to cortex through anatomically constrained deep CNNs. Preprint at *bioRxiv* <https://www.biorxiv.org/content/10.1101/511535v1.full> (2019).
- DiCarlo, J. J. & Cox, D. D. Untangling invariant object recognition. *Trends Cogn. Sci.* **11**, 333–341 (2007).
- Sabour, S., Frosst, N. & Hinton, G. E. Dynamic routing between capsules. In *Proc. Advances in Neural Information Processing Systems 30* (eds Guyon, I. et al.) 3856–3866 (2017).
- Lehky, S. R. & Sejnowski, T. J. & Desimone, R. Predicting responses of nonlinear neurons in monkey striate cortex to complex patterns. *J. Neurosci.* **12**, 3568–3581 (1992).
- Ecker, A. S. et al. A rotation-equivariant convolutional neural network model of primary visual cortex. *International Conference on Learning Representations (ICLR) 2019 Conference Poster* <https://openreview.net/forum?id=H1fU8iAqKX> (2018).
- Pasupathy, A. & Connor, C. E. Population coding of shape in area V4. *Nat. Neurosci.* **5**, 1332–1338 (2002).
- Abbasi-Asl, R. et al. The DeepTune framework for modeling and characterizing neurons in visual cortex area V4. Preprint at *bioRxiv* <https://www.biorxiv.org/content/biorxiv/early/2018/11/09/465534.full.pdf> (2018).
- Bashivan, P., Kar, K. & DiCarlo, J. J. Neural population control via deep image synthesis. *Science* **364**, eaav9436 (2019).
- Ponce, C. R. et al. Evolving images for visual neurons using a deep generative network reveals coding principles and neuronal preferences. *cell* **177**, 999–1009.e10 (2019).

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

© The Author(s), under exclusive licence to Springer Nature America, Inc. 2019

## Methods

**Neurophysiological experiments.** All procedures were approved by the Institutional Animal Care and Use Committee of Baylor College of Medicine. Briefly, five mice (*Mus musculus*: three male, two female) aged 57, 62, 91, 102 and 128 d (mouse 1–5, respectively) expressing GCaMP6s in excitatory neurons via *Slc17a7-Cre* and *Ai162* transgenic lines (stock nos. 023527 and 031562, respectively; The Jackson Laboratory) were anesthetized and a 4 mm craniotomy was made over the visual cortex as described previously<sup>32,33</sup>. Mice were head-mounted above a cylindrical treadmill and calcium imaging was performed using a Chameleon Ti-Sapphire laser (Coherent) tuned to 920 nm and a large field of view mesoscope<sup>16</sup> equipped with a custom objective (0.6 numerical aperture, 21 mm focal length). Laser power after the objective was kept below 60 mW. Visual stimuli were presented to the left eye with a 25" LCD monitor and a resolution of 2,560 × 1,440 px positioned 15 cm away from the eye. The rostro-caudal treadmill movement was measured using a rotary optical encoder with a resolution of 8,000 pulses per revolution. We used light diffusing from the laser through the pupil to capture eye movements. Images of the pupil were reflected through a hot mirror and captured with a GigE CMOS camera (Genie Nano C1920M; Teledyne Dalsa) at 20 fps at a 1,920 × 1,200 px resolution. The contour of the pupil was extracted semiautomatically for each frame and the center and major radius of a fitted ellipse were used as the position and dilation of the pupil.

Pixelwise response across a 2,400 × 2,400  $\mu\text{m}^2$  region of interest (0.2 px  $\mu\text{m}^{-1}$ ) at 200  $\mu\text{m}$  depth from the cortical surface to a drifting bar stimuli was used to generate a sign map to delineate visual areas<sup>34</sup>. We chose an imaging site in V1 with minimal blood vessel occlusion and maximal stability. The craniotomy window was leveled with regard to the objective with six d.f., five of which were locked between days to allow us to return to the same imaging site using the z axis. Imaging was performed at approximately 8 Hz for all scans, using a remote objective to sequentially image ten 630 × 630  $\mu\text{m}^2$  fields per frame at 0.4 px  $\mu\text{m}^{-1}$  xy resolution. Fields were spaced 5  $\mu\text{m}$  apart in depth to achieve dense imaging coverage of a 630 × 630 × 50  $\mu\text{m}^3$  xyz volume, with the most superficial plane positioned in L2/3 at around 200  $\mu\text{m}$  from the surface of the cortex. At this z resolution, cells in the imaged volume were oversampled, often appearing in 3 or more imaging planes and allowing matching across days with  $\leq 2.5 \mu\text{m}$  vertical distance between masks. Imaging data were motion-corrected, automatically segmented and deconvolved using the CNMF algorithm<sup>35</sup>; cells were further selected by a classifier trained to detect somata based on the segmented cell masks. This resulted in 5,300–8,500 soma masks per scan. A structural stack encompassing the volume and imaged at 0.6 × 0.6 × 1 px<sup>3</sup>  $\mu\text{m}^{-3}$  xyz resolution with 100 repeats was used to register the scan average image into a shared xyz frame of reference between scans (see further on).

No statistical methods were used to predetermine sample sizes but our sample sizes are similar to those reported in previous publications<sup>30</sup>. Data collection and analysis were not performed blind to the conditions of the experiments. In performing the analysis, no animal or collected data point was excluded. Additional details may be found in Nature Research Reporting Summary.

**Monitor positioning across days.** To have a consistent monitor placement relative to the mouse across all imaging sessions, we placed the aggregate RF for each session at the center of the monitor. To map the RF, we tiled the center of the screen in a 10 × 10 grid with single dark dots over bright background (approximately 5°) and averaged the calcium trace of an approximately 150 × 150  $\mu\text{m}^2$  window in the center of our field of view from 0.5–1.5 s after stimulus onset across all repetitions of the stimulus for each location. We fitted the resulting two-dimensional (2D) map using an elliptic 2D Gaussian and we centered the position of the RF by displacing the monitor with approximately ±2.5° precision between imaging sessions.

**Cell registration across days.** We registered each 2D scanning plane to the three-dimensional stack using an affine transformation matrix with 9 d.f. estimated via gradient ascent on the correlation between the average recorded plane and the extracted plane from the stack. We matched cells across days using their estimated centroids in the stack, matching each cell of interest to the closest cell in the stack from a different day. We repeated this matching over multiple stacks and selected the pairings that occurred most often. To confirm that our target cells were matched correctly, we correlated their responses across days to a set of repeated images. The resulting confusion matrix is strongly diagonal demonstrating that matched cells have similar functional properties as expected (Supplementary Fig. 2).

**Presentation of natural stimuli.** Stimuli consisted of 5,100 natural images from ImageNet (ILSVRC2012)<sup>36</sup>, cropped to fit a 16:9 monitor aspect ratio and converted to gray scale. In each scan, we showed 5,000 unique images and 100 images repeated 10 times each. Each image was presented for 500 ms followed by a blank screen lasting between 300 and 500 ms, sampled uniformly from that range.

**Preprocessing of neural and behavioral data.** Neural responses were first deconvolved using constrained nonnegative calcium deconvolution<sup>35</sup>. We subsequently extracted the accumulated activity of each neuron between 50 and 550 ms after stimulus onset using a Hamming window. Behavioral traces, used as auxiliary signals in model training (see further on), were extracted using the same

temporal offset and integration window. To train our models, we isotropically downsampled stimuli images to 64 × 36 px. Input images, the target neuronal activities, behavioral traces and pupil positions were normalized across the training set during training.

**Network architecture.** The network considered in the present study consists of four components (Fig. 1d): a common core for all neurons providing nonlinear features computed from static images; a dedicated readout for each neuron mapping the core features to their responses; a modulator predicting a gain factor for each neuron depending on the running state and the pupil dilation of the animal recorded during the experiment; and a shifter predicting RF shifts from pupil position changes.

The CNN core consists of three layers (Fig. 1d, each layer is illustrated as a differently colored block) whose outputs are concatenated to yield a rich feature set used by the readout module<sup>6</sup>. Each layer in the core consists of a convolution layer without a bias term, a batch normalization layer with an affine function term<sup>37</sup> and an exponential linear unit (ELU) nonlinearity<sup>38</sup>. The LN model and the nonlinear (CNN) model differ only in their core: the core for the LN network is identical to the CNN core except that all nonlinearities were removed. This ensures maximal similarity between the two networks in terms of network components, maximal RF size and behavioral modulation.

For the readout, we model the neural response as an instantaneous affine function of the core features followed by an ELU nonlinearity and an offset of 1 to make the response positive. For each image, the output of the core is a tensor  $v \in \mathbb{R}^{w \times h \times c}$ . We model the location of the  $i$ th neuron's RF with a spatial transformer layer reading from a single grid point  $v_{x_p y_p}$ <sup>6,39</sup>, that is, we extract a vector  $v_{x_p y_p} \in \mathbb{R}^c$  via bilinear interpolation of neighboring pixels of the location  $(x_p, y_p)$  in the tensor  $v$ . To facilitate the learning of the grid points via gradient descent, we decompose  $v_{ijk}$  into  $\ell$  spatial scales through repeated application of a 5 × 5 Gaussian low-pass filter:  $v^{(j)} = \text{lowpass}^{(j)}(v)$  for  $j \in [0, \ell - 1]$  (Fig. 1d, represented as multiple copies of the three-layered block at different scales). Like in a Gaussian pyramid, we keep the difference between the filtered and original component at each step of filtering. However, we include readout variants where we do not explicitly downsample the feature channels since we empirically found it performs slightly better on some datasets. The exact variant used was determined by model selection on a validation set (see further on). The spatial transformer layer then extracts the feature vector from the same learned relative location  $(x_p, y_p)$  at each scale. The set of feature vectors is then fed to the final affine function and nonlinearity, yielding:

$$o^{(i)} = f \left( \sum_{j=1}^{\ell} w_{(i,j)}^T v_{x_p y_p}^{(j)} + b_i \right) \quad (1)$$

where  $o^{(i)}$  is the readout layer's output for the  $i$ th neuron,  $f(\cdot)$  is the output nonlinearity and  $w_{(i,j)} \in \mathbb{R}^c$  is the weight vector for the  $i$ th neuron for the scale  $j$  (Fig. 1d, depicting feature vectors at two readout locations).

To account for fluctuations in neural responses unrelated to the visual stimuli, we used pupil dilations and their temporal derivative, as well as the absolute running speed of the animal, which are known correlates of brain state changes<sup>32,40,41</sup>. Using these three variables, we computed a gain factor for each neuron that scales the output of the readout layer  $o^{(i)}$  (equation (1)). The modulator predicts the gain using a two-layer fully connected multilayer perceptron (MLP) with rectified linear unit nonlinearities at all hidden layers and a shifted exponential nonlinearity at the last layer to enforce positive outputs.

Unlike primates, training mice to fixate their gaze in a single position is impractical. To model the responses of thousands of neurons in a free viewing experiment, the shifter estimates a RF shift for all neurons from the tracked pupil position based on the predictive performance of the network<sup>6</sup>. In the model, this is reflected as trial-by-trial shift  $\Delta x$  and  $\Delta y$  applied to  $x_i$  and  $y_i$  in equation (1) across all neurons. Note that pupil location is measured in coordinates of the camera recording the eye, while the shift needs to be applied in monitor coordinates. This transformation can be estimated by a calibration procedure<sup>42–44</sup> or learned from the data using regression on pairs of eye camera–monitor coordinates. We used a three-layer MLP with a tanh nonlinearity for predicting the joint receptive displacement for all neurons.

Both modulator and shifter components were included in the linear-nonlinear network to allow for a fair comparison between the models.

**Training and model selection.** We trained two different network architectures: an LN model and a CNN model. Both models used a point readout, and shifter and modulator networks, as described earlier. We trained four instances of each network configuration, corresponding to four random weight initializations.

The first layer was regularized by penalizing the L2 norm of the 3 × 3 Laplace-filtered weights, weighted by an inverse Gaussian profile of the form:

$$\alpha_{x,y} = \gamma_L \left( b_L - \exp \left( -\frac{1}{2\sigma_L^2} (x^2 + y^2) \right) \right) \quad (2)$$

where  $x$  and  $y$  are the spatial pixel positions of the convolutional weights relative to the center of the filter. The Gaussian profile encourages the filter to be smoother at the edge. Empirically, we observed that this profile also helps to center the RF of the neuron within the convolutional kernel's spatial extent. The values of  $\sigma_l$  and  $b_l$  were set to 0.5 and 1, respectively. For each hidden convolutional layer, the convolutional weight was penalized by group sparsity, and computed as the  $L_1$  sparsity regularizer over the  $L_2$  norm of weight values for each channel with the regularization weight  $\gamma_r$ . The values of  $\gamma_l$  and  $\gamma_r$  were chosen from a discrete set of values as part of the hyperparameter search (Supplementary Table 1).

The dataset was split into training, validation and test set. The test set consisted of 100 images with 10 repeats each. The remaining unique 5,000 images were randomly split into 4,477 training images and 523 validation images. The network was trained to minimize Poisson loss  $\frac{1}{m} \sum_{i=1}^m (\hat{r}^{(i)} - r^{(i)}) \log \hat{r}^{(i)}$  where  $m$  denotes the number of neurons,  $\hat{r}$  the predicted neuronal response and  $r$  the experimentally recorded one. We used early stopping on the correlation between predicted and measured neuronal responses on the validation set<sup>45</sup>; if the correlation failed to increase during any consecutive ten passes through the entire training set, we stopped the training and restored the model to the best-performing model over the course of training. Empirically, we found that this combination of Poisson objective and early stopping on correlation yielded the best results. After the first stop, we decreased the learning rate from  $5 \times 10^{-3}$  to  $10^{-3}$  and resumed training until it was stopped again. Network parameters were optimized iteratively via stochastic gradient descent with the Adam optimizer<sup>46</sup> with a batch size of 60. Once training completed, the trained network was evaluated on the validation set to yield the score used for hyperparameter selection.

Exact architectural details, including the weights of regularizers, or whether to downsample the spatial pyramid, were selected using grid search on the validation performance. We repeated our model selection for each mouse and used the best CNN and LN architecture for all subsequent experiments (Supplementary Table 2). Remaining network configurations, namely, the input convolution kernel size ( $k_{in} = 15$ ), the number of hidden layers ( $n = 2$ ), hidden layer convolution kernel size ( $k_{hidden} = 7$ ) and number of channels ( $c = 32$ ) were manually selected based on previous experience.

Since the imaging volume was densely scanned (10 planes spaced 5  $\mu\text{m}$  apart in depth), the same soma would appear in several planes. During training, we used masks from every plane, including potentially duplicate ones.

**Oracle correlation and fraction of oracle.** Cortical neurons naturally exhibit substantial response variability. To estimate a bound for maximally achievable correlation, we computed an 'oracle' per cell by correlating the leave-one-out mean response with the response to the remaining trial across repeated images in the test set, and averaged that correlation across all cells. We estimated the fraction of oracle performance achieved by a model as the slope of a line without offset fitted on the oracle correlation to the model's test set correlation across all neurons. Since the oracle is only conditioned on repeats of the same image and not factors relating to brain state, a network with shifter and modulator components that utilize brain state parameters, could in principle achieve a fraction of oracle score  $>1$ . To avoid this confounding element, we froze the shifter and modulator input to the training set mean when computing the fraction of oracle (Fig. 1e).

**Selection of neurons for MEI generation.** We selected neurons to generate MEIs based on the following criteria. First, select neurons in the top 50th percentile of oracle correlation. This restricts us to cells with reliable responses to visual stimuli. Second, exclude neurons within 10  $\mu\text{m}$  of the edge of the scanning fields. This avoids artifacts near the edge of the fields and avoids missing a cell on subsequent days if the scanning field of view moves slightly in  $xy$ . Next, select neurons in the intersection of the top 30th percentile of the CNN model's fraction of oracle score  $\rho_{CNN}$  and the top 30th percentile of the LN model's fraction of oracle score  $\rho_{LN}$ . This selects cells that are predicted reasonably well by both CNN and LN models and ensures that each neuron has a significant linear part that can be predicted by an LN model using the linear RF. Last, iterate through the remaining neurons, from largest to smallest in  $\rho_{CNN} - \rho_{LN}$ , placing the visited neuron into a final to-keep set and removing any unvisited neuron that falls within 20  $\mu\text{m}$  distance of it. This avoids selecting neurons that are too close to each other, reducing the chance of picking two masks that belong to the same cell. These criteria yielded a total of 206, 304, 311, 309 and 346 neurons for mouse 1–5. Among these, for each mouse, we selected the top 150 neurons according to largest  $\rho_{CNN} - \rho_{LN}$  for MEI generation.

**Generation of synthetic stimuli.** To find stimuli that optimally drive particular cells in V1, we were inspired by deep learning approaches to visualize the hidden features of an artificial neural network<sup>15,47–57</sup>.

For each model neuron, we generated the image that most strongly activates it<sup>55</sup>, subject to a number of regularization constraints to encourage stable results. We started the optimization with a Gaussian white noise image  $I_0 \in \mathbb{R}^{w \times h \times c}$  and iteratively added the gradient of the target neuron's activity with respect to the image  $\nabla_I f$ , averaged over four instances of the network; we trained four instances of each selected architecture, each initialized with different random parameters. Optimizing on four networks simultaneously better estimates the gradient and reduces high-frequency noise. High-frequency noise obscures the gradient signal

and depends highly on the starting image; thus, it is more powerful during the start of the optimization<sup>50</sup>. We used two additional strategies to dampen its effect. First, we blurred the image after every gradient ascent step using a Gaussian filter with an s.d. that decreases gradually after each iteration<sup>58</sup>. Second, we preconditioned the gradient before adding it to the image with a low-pass filter  $G(\omega_x, \omega_y) = \frac{1}{(2\pi)^2} (\omega_x^2 + \omega_y^2)^{-\alpha}$  in the Fourier domain that preferentially suppresses the higher-frequency content of the gradient<sup>50</sup>; we selected  $\alpha = 0.1$  by visual inspection of the resulting images.

When this image generation technique is applied to a CNN model, it creates the MEIs for the target neuron. When it is applied to an LN model, the resulting image is equivalent to a (highly regularized) linear RF. After optimization, we matched the mean luminance and RMS contrast of all MEIs and RFs to a common value.

**Presentation of synthetic stimuli.** For closed-loop scans designed to compare MEI to RF responses (day 2 – N), we presented 150 MEIs and 150 RFs with 20 repetitions each. During the presentation, the images were upscaled to the monitor resolution using third-order spline interpolation. We also included 100 images repeated 10 times to compute the oracle score. Just as with natural images (day 1), images were presented for 500 ms followed by a blank screen, with duration uniformly distributed between 300 and 500 ms; the order of presentation was chosen at random.

**Statistical significance of MEI comparisons.** Recorded responses were normalized across all presented images per scan. The normalized responses of the matched neurons were then averaged across MEI (or RF) repetitions in the same scan and, if available, across multiple scans (2, 2, 1, 2, 1 scans for each of our five mice); all scans were recorded on separate days.

We used these aggregate responses to assess whether MEIs generated for their target neurons elicited higher responses than their corresponding linear RF, Gabor image, masked natural image or full-field natural image. For single neurons, the statistical significance of the difference in response was assessed using a two-tailed Welch's *t*-test across the pooled repeats (average d.f. of 55.5, 28.9, 32.0 and 30.3 for MEI versus RF, Gabor, masked natural and full-field natural, respectively). The overall difference in average responses pooled across all neurons (750 for MEI versus RF, 150 for other comparisons) was assessed using a two-sided Wilcoxon signed-rank test.

**Generation of MEI mask.** We created a weighted mask for each MEI to capture the region containing most of its variance so that the resulting masked MEI activates the model neuron only slightly less than the original MEI. We generated this mask as follows: (1) starting with an MEI image  $I_{MEI}$ , compute the absolute deviation of the pixels from the mean image intensity  $\mu_I$ ,  $\Delta I_{xy} = |I_{MEI,x,y} - \mu_I|$ , and threshold it at 1 s.d. of  $\Delta I$  to identify highly active pixels; (2) compute the convex hull over pixels identified in the previous step, producing a mask image  $M^{(0)}$  with a single connected region.  $M = 1$  for all pixels inside the convex hull and 0 otherwise; (3) Gaussian blur the mask  $M^{(k)}$  ( $\sigma = 2 \text{ px}$ ) to smooth its edges, resulting in  $\hat{M}^{(k)}$ ; (4) compute masked MEI as  $I_{MEI,\text{masked}}^{(k)} = \hat{M}^{(k)} \odot I_{MEI} + \mu_{I_{MEI}} (1 - \hat{M}^{(k)})$ ; (5) run  $I_{MEI,\text{masked}}^{(k)}$  through the model to yield the model neuron response  $\hat{r}^{(k)}$ ; (6) binary erode  $M^{(k)}$  with a  $3 \times 3$  square structuring element<sup>59</sup> to yield  $M^{(k+1)}$ ; and (7) repeat steps 3–6 until  $\hat{r}^{(k)}$  goes below 90% of  $\hat{r}^{(0)}$  for the first time. We take  $\hat{M}^{(k)}$  to be the final MEI mask;  $\odot$  denotes the Hadamard product (elementwise product).

**Selection of Gabor stimuli.** To directly compare the effectiveness of MEI against Gabor stimuli, we selected the Gabor image with the highest predicted activation for each target cell. A Gabor image was generated according to the following equation:

$$I_{\text{Gabor}}(x, y) = \exp\left(-\frac{1}{2\sigma^2}((x - \mu_x)^2 + (y - \mu_y)^2)\right) \cos\left(\frac{2\pi}{\lambda}(x \cos\theta + y \sin\theta) + \phi\right) \quad (3)$$

in the luminance space, where  $\mu_x$  and  $\mu_y$  control the center of the Gabor,  $\sigma$  controls the fall off of the Gaussian window and  $\theta$ ,  $\lambda$  and  $\phi$  control the orientation, spatial frequency and phase of the grating, respectively. We defined a set of discrete values for each of the six parameters and searched over all combinations to find the most exciting Gabor image (Supplementary Table 3). For each candidate Gabor image, the image mean and scaling was adjusted to ensure that all images shared the same mean luminance and RMS contrast (Fig. 3b and Supplementary Fig. 11).

In one closed-loop experiment for mouse 5, we presented the 150 best Gabor filter for each of our target cells and their corresponding MEIs exactly as we did in scans comparing MEIs to RFs.

**Selection of masked and full-field natural images.** For each target cell with an index, we searched for the natural image that yielded the highest predicted activation after masking it with the neuron's MEI mask (see earlier); we used 5,000 images not used during training. This allowed us to compare responses for MEIs to those for natural images and assess the prevalence of MEI-like features in natural scenes.

For a neuron with MEI mask  $M$ , we masked each original full-field natural image  $I_j$  ( $j$  is the index of the natural image in the set of 5,000 images), yielding the masked natural image  $\bar{I}_j$  as follows:

$$\bar{I}_j = M \odot (\alpha I_j + \beta) + (1 - M)\mu \quad (4)$$

where the scalars  $\alpha$  and  $\beta$  are adjusted so that the mean luminance  $\mu$  and RMS contrast of the masked images for this neuron are kept constant. We then selected the masked natural image  $\bar{I}_j$  that gives rise to the largest model neuron activation among the 5,000 new natural images, where  $j$  is the index of the best natural image; we also compared it against the corresponding full-field natural image  $I_j$  (Fig. 3b and Supplementary Fig. 11).

In one closed-loop experiment for mouse 5, we presented the 150 best masked natural images  $\bar{I}_j$  for each of our target cells and their corresponding MEIs exactly as we did in the closed-loop scans comparing MEIs to RFs. In a separate closed-loop scan for mouse 5, we presented the 150 full-field counterparts of the best masked natural images  $I_j$  and the 150 MEIs.

**Nontrivial nonlinearity of the CNN.** An LN model has the form  $r = g(\mathbf{w}^\top \mathbf{I} + b)$  where  $r$  is the neuronal response prediction,  $\mathbf{I}$  is the input image,  $\mathbf{w}$  is a weight vector of the same dimensions,  $b$  is an offset and  $g$  is a static nonlinearity usually chosen as part of the model design. In our case, we chose  $g(z) = \text{ELU}(z) + 1$ , where<sup>38</sup>:

$$\text{ELU}(x) = \begin{cases} x, & \text{if } x \geq 0 \\ e^x - 1, & \text{otherwise} \end{cases} \quad (5)$$

It is possible that the CNN model differs from the LN model only in trivial ways by effectively learning an LN model but achieving higher prediction scores because of more freedom in fitting the static nonlinearity  $g$  or because of an inbuilt architectural bias that makes learning  $\mathbf{w}$  easier.

To demonstrate that the CNN model deviates nontrivially from the LN model, we computed the gradient of both models with respect to the input image on the entire image dataset and computed the largest ten eigenvalues of the covariance matrix of these gradients. For the LN model, all gradients are proportional to  $\mathbf{w}$ . Thus, there must be exactly one eigenvalue greater than zero. If the CNN model behaves just like an LN model, the spectrum should look the same. However, if it is nonlinear in a nontrivial way, gradients should differ and the spectrum should have more nonzero eigenvalues.

We find the latter to be the case (Supplementary Fig. 3), indicating that the CNN model performs better because it can model interesting nonlinearities of cortical neurons that cannot be captured by the linear model.

**Difference in spatial frequency content between MEIs and RFs.** We compared the spatial frequency content of MEIs and RFs by computing the average differences in the amplitude of the spatial frequency spectrums of the MEI and RF images:

$$\bar{A}_{\text{diff}} = \frac{1}{N} \sum_i (|\mathcal{F}(h \odot I_{\text{MEI},i})| - |\mathcal{F}(h \odot I_{\text{RF},i})|) \quad (6)$$

where  $\mathcal{F}(\cdot)$  denotes the 2D Fourier transform,  $h$  is a Hamming window and  $N$  is the total number of MEI/RF image pairs (Supplementary Fig. 7).

**MEIs as linear filters.** CNNs model the nonlinear processing of the cells they encode. Intuitively, a CNN allows us to generate images (MEIs) that use this additional capacity to drive a cell more strongly than images generated from an LN model (RFs) by disregarding their ability to act as a linear filter. To test this intuition and assert that the observed higher activations for MEIs are not due to them being better linear filters than RFs, we compared their performance as a linear encoder. For each cell, we used its MEI or RF as a linear filter to predict its responses to 100 test set images and correlated the predicted responses with the real neuron responses. We used Spearman's rank correlation to sidestep the need to fit a monotonic nonlinear function to the output of the filter. Filters generated with an LN model produce better predictions (Supplementary Fig. 9), suggesting that the capacity of MEIs to excite cells depends on their ability to exploit subtle nonlinear processing in V1 cells.

**RFs of the linearized CNN model.** To further assess the importance of the nonlinear nature of our models, we approximated our CNN using an LN model and compared the original RFs to the RFs of this linearized CNN model. We fitted LN models on the same training set used for the rest of experiments but replaced the real cell responses with those predicted by a trained CNN, and followed the same procedures described earlier for model selection, training and RF generation. The resulting RFs looked virtually identical to the RFs learned directly from the neuronal responses, further corroborating the importance of a nonlinear model (Supplementary Fig. 10).

**Statistics.** All statistical tests used, including statistical values, sample sizes and  $P$  values are provided in the figure captions. Where a  $t$ -test was used, the underlying data distribution was assumed to be normal, although this was not formally tested. Exact  $P$  values less than  $10^{-9}$  were reported as  $P < 10^{-9}$ .

**Reporting Summary.** Further information on research design is available in the Nature Research Reporting Summary linked to this article.

## Data availability

All figures were generated from raw or processed data. The data generated and/or analyzed during the current study are available from the corresponding author upon request. No publicly available data was used in this study.

## Code availability

Experiments and analyses were performed using custom software developed using the following tools: ScanImage 2018a (ref. <sup>60</sup>), CaImAn v.1.0 (ref. <sup>61</sup>), DataJoint v.0.11.1 (ref. <sup>62</sup>), PyTorch v.0.4.1 (ref. <sup>63</sup>), NumPy v.1.16.4 (ref. <sup>64</sup>), SciPy v.1.3.0 (ref. <sup>65</sup>), Docker v.18.09.7 (ref. <sup>66</sup>), Matplotlib v.3.0.3 (ref. <sup>67</sup>), seaborn v.0.9.0 (ref. <sup>68</sup>), pandas v.0.24.2 (ref. <sup>69</sup>) and Jupyter v.1.0.0 (ref. <sup>70</sup>). The code for carrying out the data collection and preprocessing is available at <https://github.com/cajal/pipeline>; the code to perform MEI generation and analysis is available at [https://github.com/cajal/inception\\_loop2019](https://github.com/cajal/inception_loop2019).

## References

32. Reimer, J. et al. Pupil fluctuations track fast switching of cortical states during quiet wakefulness. *Neuron* **84**, 355–362 (2014).
33. Froudarakis, E. et al. Population code in mouse v1 facilitates readout of natural scenes through increased sparseness. *Nat. Neurosci.* **17**, 851–857 (2014).
34. Garrett, M. E., Nauhaus, I., Marshel, J. H. & Callaway, E. M. Topography and areal organization of mouse visual cortex. *J. Neurosci.* **34**, 12587–12600 (2014).
35. Pnevmatikakis, E. A. et al. Simultaneous denoising, deconvolution, and demixing of calcium imaging data. *Neuron* **89**, 285–299 (2016).
36. Russakovsky, O. et al. ImageNet large scale visual recognition challenge. *Int. J. Comput. Vis.* **115**, 211–252 (2015).
37. Ioffe, S. & Szegedy, C. Batch normalization: accelerating deep network training by reducing internal covariate shift. in *Proceedings of the 32nd International Conference on Machine Learning, Lille, France* **37**, 448–456 (2015).
38. Clevert, D.-A., Unterthiner, T. & Hochreiter, S. Fast and accurate deep network learning by exponential linear units (ELUs). Preprint at [arXiv https://arxiv.org/pdf/1511.07289.pdf](https://arxiv.org/pdf/1511.07289.pdf) (2015).
39. Jaderberg, M., Simonyan, K., Zisserman, A. & Kavukcuoglu, K. Spatial transformer networks. In *Proc. Advances in Neural Information Processing Systems 28* (eds Cortes, C. et al.) 2017–2025 (Curran Associates, 2015).
40. McGinley, M. J. et al. Waking state: rapid variations modulate neural and behavioral responses. *Neuron* **87**, 1143–1161 (2015).
41. Fu, Y. et al. A cortical circuit for gain control by behavioral state. *Cell* **156**, 1139–1152 (2014).
42. Zoccolan, D., Graham, B. & Cox, D. A self-calibrating, camera-based eye tracker for the recording of rodent eye movements. *Front. Neurosci.* **4**, 193 (2010).
43. Stahl, J. S., van Alphen, A. M. & De Zeeuw, C. I. A comparison of video and magnetic search coil recordings of mouse eye movements. *J. Neurosci. Methods* **99**, 101–110 (2000).
44. van Alphen, B., Winkelmann, B. H. & Frens, M. A. Three-dimensional optokinetic eye movements in the C57BL/6J mouse. *Invest. Ophthalmol. Vis. Sci.* **51**, 623–630 (2010).
45. Prechelt, L. Early stopping — but when? in *Neural Networks: Tricks of the Trade* (eds Montavon, G., Orr, G., & Müller, K.-R.) 53–67 (Springer, 1998).
46. Kingma, D. P. & Ba, J. Adam: a method for stochastic optimization. Preprint at [arXiv https://arxiv.org/pdf/1412.6980.pdf](https://arxiv.org/pdf/1412.6980.pdf) (2017).
47. Nguyen, A., Dosovitskiy, A., Yosinski, J., Brox, T. & Clune, J. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. *Adv. Neural Inf. Process. Syst.* **29**, 3387–3395 (2016).
48. Nguyen, A. M., Yosinski, J. & Clune, J. Multifaceted feature visualization: uncovering the different types of features learned by each neuron in deep neural networks. Preprint at [arXiv https://arxiv.org/pdf/1602.03616.pdf](https://arxiv.org/pdf/1602.03616.pdf) (2016).
49. Wei, D., Zhou, B., Torralba, A. & Freeman, W. T. Understanding intra-class knowledge inside CNN. Preprint at [arXiv https://arxiv.org/pdf/1507.02379.pdf](https://arxiv.org/pdf/1507.02379.pdf) (2015).
50. Olah, C., Mordvintsev, A. & Schubert, L. Feature visualization: how neural networks build up their understanding of images. *Distill* <https://distill.pub/2017/feature-visualization> (2017).

51. Simonyan, K., Vedaldi, A. & Zisserman, A. Deep inside convolutional networks: visualising image classification models and saliency maps. *International Conference on Learning Representations (ICLR) Workshop Paper* <https://arxiv.org/abs/1312.6034> (2014).
52. Kindermans, P.-J., Schütt, K. T., Alber, M., Müller, K.-R. & Dähne, S. Learning how to explain neural networks: PatternNet and PatternAttribution. Preprint at *arXiv* <https://arxiv.org/pdf/1705.05598.pdf> (2017).
53. Yosinski, J., Clune, J., Nguyen, A., Fuchs, T. & Lipson, H. Understanding neural networks through deep visualization. Preprint at *arXiv* <https://arxiv.org/pdf/1506.06579.pdf> (2015).
54. Gatys, L. A., Ecker, A. S. & Bethge, M. A neural algorithm of artistic style. Preprint at *arXiv* <https://arxiv.org/pdf/1508.06576.pdf> (2015).
55. Mahendran, A. & Vedaldi, A. Understanding deep image representations by inverting them. Preprint at *arXiv* <https://arxiv.org/pdf/1412.0035.pdf> (2015).
56. Lenc, K. & Vedaldi, A. Understanding image representations by measuring their equivariance and equivalence. Preprint at *arXiv* <https://arxiv.org/pdf/1411.5908.pdf> (2015).
57. Tsai, C.-Y. & Cox, D. D. Characterizing visual representations within convolutional neural networks: toward a quantitative approach. In *Proc. Workshop on Visualization for Deep Learning, 33rd International Conference on Machine Learning* (2016).
58. Øygard, A. Visualizing GoogLeNet classes. *Audun M. Øygard Blog* <https://www.auduno.com/2015/07/29/visualizing-googlenet-classes/> (2015).
59. Sreedhar, K. & Panlal, B. Enhancement of images using morphological transformations. *Int. J. Comput. Sci. Inf. Technol.* **4**, 33–50 (2012).
60. Pologruto, T. A., Sabatini, B. L. & Svoboda, K. Scanimage: flexible software for operating laser scanning microscopes. *Biomed. Eng. Online* **2**, 13 (2003).
61. Giovannucci, A. et al. Caiman: an open source tool for scalable calcium imaging data analysis. *eLife* **8**, e38173 (2019).
62. Yatsenko, D., Walker, E. Y. & Tolias, A. S. Datajoint: a simpler relational data model. Preprint at *arXiv* <https://arxiv.org/pdf/1807.11104.pdf> (2018).
63. Paszke, A. et al. Automatic differentiation in PyTorch. In *Proc. Advances in Neural Information Processing Systems (NIPS) 31 Workshop Autodiff* Submission (2017).
64. van der Walt, S., Colbert, S. C. & Varoquaux, G. The NumPy array: a structure for efficient numerical computation. *Comput. Sci. Eng.* **13**, 22–30 (2011).
65. Jones, E. et al. *SciPy: Open Source Scientific Tools for Python* <http://www.scipy.org> (SciPy.org, accessed 3 October 2019)
66. Merkel, D. Docker: lightweight Linux containers for consistent development and deployment. *Linux J.* **239**, 2 (2014).
67. Hunter, J. D. Matplotlib: a 2D graphics environment. *Comput. Sci. Eng.* **9**, 90–95 (2007).
68. Waskom, M. et al. mwaskom/seaborn: v0.8.1 (September 2017). *Zenodo* <https://zenodo.org/record/883859#.XZXljUZKguV> (2017).
69. McKinney, W. Data structures for statistical computing in Python. In *Proc. 9th Python in Science Conference* (eds van der Walt, S. & Millman, J.) 51–56 (2010).
70. Kluyver, T. et al. Jupyter notebooks: a publishing format for reproducible computational workflows. In *Proc. 20th International Conference on Electronic Publishing. Positioning and Power in Academic Publishing: Players, Agents and Agendas* (eds Loizides, F. & Schmidt, B.) 87–90 (IOS Press, 2016).

## Acknowledgements

We thank G. Denfield for comments on the manuscript. This research was supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/Interior Business Center (DoI/IBC) contract no. D16PC00003. The US Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation thereon. 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 US Government. This research was also supported by grant no. R01 EY026927 to A.S.T., National Eye Institute/National Institutes of Health Core Grant for Vision Research (no. T32-EY-002520-37), National Science Foundation NeuroNex grant no. 1707400 to X.P. and A.S.T., and grant no. F30EY025510 to E.Y.W. F.H.S. is supported by the Institutional Strategy of the University of Tübingen (ZUK 63) and the Carl-Zeiss-Stiftung. F.H.S. acknowledges the support from the German Federal Ministry of Education and Research (BMBF) through the Tübingen AI Center (FKZ: 01IS18039A), the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC-Number 2064/1 – Project number 390727645, and Amazon AWS through a Machine Learning Research Award. P.G.F. received support from the BCM Medical Scientist Training Program, no. F30-MH112312. The name of the authors' approach, inception loops, was inspired by the movie *Inception* directed by Christopher Nolan.

## Author contributions

All authors designed the experiments and developed the theoretical framework. E.Y.W. designed and implemented the inception loop framework with contributions from F.H.S. and E.C. T.M. performed the surgeries and conducted the recordings with contributions from E.F., P.G.F. and J.R. E.Y.W. performed data analyses on mice 1 and 2. E.Y.W. and E.C. performed the data analyses on mice 3–5. E.Y.W., F.H.S., A.S.E., X.P. and A.S.T. wrote the manuscript, with contributions from all authors. A.S.T. supervised all stages of the project.

## Competing interests

E.Y.W., J.R. and A.S.T. hold equity ownership in Vathes LLC, which provides development and consulting for the framework (DataJoint) used to develop and operate the data analysis pipeline for this publication.

## Additional information

**Supplementary information** is available for this paper at <https://doi.org/10.1038/s41593-019-0517-x>.

**Correspondence and requests for materials** should be addressed to E.Y.W., F.H.S. or A.S.T.

**Peer review information** *Nature Neuroscience* thanks Bruno Olshausen, Joel Zylberberg, and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

# Reporting Summary

Nature Research wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Research policies, see [Authors & Referees](#) and the [Editorial Policy Checklist](#).

## Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a Confirmed

- The exact sample size ( $n$ ) for each experimental group/condition, given as a discrete number and unit of measurement
- A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
- The statistical test(s) used AND whether they are one- or two-sided  
*Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
- A description of all covariates tested
- A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
- A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
- For null hypothesis testing, the test statistic (e.g.  $F$ ,  $t$ ,  $r$ ) with confidence intervals, effect sizes, degrees of freedom and  $P$  value noted  
*Give P values as exact values whenever suitable.*
- For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
- For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
- Estimates of effect sizes (e.g. Cohen's  $d$ , Pearson's  $r$ ), indicating how they were calculated

*Our web collection on [statistics for biologists](#) contains articles on many of the points above.*

## Software and code

Policy information about [availability of computer code](#)

Data collection

Experiments and analyses were performed using custom software developed using the following tools: ScanImage (2018a), CalmAn (1.0), DataJoint (0.11.1), PyTorch (0.4.1), NumPy (1.16.4), SciPy (1.3.0), Docker (18.09.7), matplotlib (3.0.3), seaborn (0.9.0), pandas (0.24.2), and Jupyter (1.0.0). All code necessary to reproduce the data collection are made available at <https://github.com/cajal/pipeline>.

Data analysis

Experiments and analyses were performed using custom software developed using the following tools: ScanImage (2018a), CalmAn (1.0), DataJoint (0.11.1), PyTorch (0.4.1), NumPy (1.16.4), SciPy (1.3.0), Docker (18.09.7), matplotlib (3.0.3), seaborn (0.9.0), pandas (0.24.2), and Jupyter (1.0.0). All code necessary to reproduce the data analysis are made available at [https://github.com/cajal/inception\\_loop2019](https://github.com/cajal/inception_loop2019).

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors/reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Research [guidelines for submitting code & software](#) for further information.

## Data

Policy information about [availability of data](#)

All manuscripts must include a [data availability statement](#). This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A list of figures that have associated raw data
- A description of any restrictions on data availability

All figures were generated from raw data or processed data. The datasets generated and/or analyzed during the current study are available from the corresponding author upon reasonable request. No publicly available datasets was used in this study.

# Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

- Life sciences       Behavioural & social sciences       Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see [nature.com/documents/nr-reporting-summary-flat.pdf](https://nature.com/documents/nr-reporting-summary-flat.pdf)

## Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

|                 |  |
|-----------------|--|
| Sample size     | A total of 5 mice were used with approximately 3000 neurons recorded and used during the training of the networks for each scan from each mouse. The size of the dataset was determined based on the capacity of the recording system as well as the number of subjects, sites and neurons recorded in a similar study (Bashivan, Kar and DiCarlo 2019). |
| Data exclusions | No data was excluded during the recording, training or analysis of the mouse V1 responses to natural images. In performing the closed loop analysis, 150 neurons were selected based on the criteria outlined in detail in the manuscript.   |
| Replication     | Data was collected over multiple scans from each of the 5 mice, and the findings were replicated across all mice.  |
| Randomization   | There was no randomization performed as the study does not involve multiple study groups, and all analyses were performed in identical fashion on all subjects.  |
| Blinding        | There was no blinding performed as the study does not involve multiple study groups, and all analyses were performed in identical fashion on all subjects.   |

## Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

| Materials & experimental systems    |   | Methods                             |   |
|-------------------------------------|---|-------------------------------------|---|
| n/a                                 | Involved in the study   | n/a                                 | Involved in the study                           |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> Antibodies                             | <input checked="" type="checkbox"/> | <input type="checkbox"/> ChIP-seq               |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> Eukaryotic cell lines                  | <input checked="" type="checkbox"/> | <input type="checkbox"/> Flow cytometry         |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> Palaeontology                          | <input checked="" type="checkbox"/> | <input type="checkbox"/> MRI-based neuroimaging |
| <input type="checkbox"/>            | <input checked="" type="checkbox"/> Animals and other organisms |                                     |   |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> Human research participants            |                                     |   |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> Clinical data                          |                                     |   |

## Animals and other organisms

Policy information about [studies involving animals](#); [ARRIVE guidelines](#) recommended for reporting animal research

|                         |  |
|-------------------------|--|
| Laboratory animals      | A total of five healthy mice ( <i>Mus Musculus</i> , three male, two female) aged 57, 62, 91, 102 and 128 days expressing GCaMP6s in excitatory neurons via SLC17a7-Cre and Ai162 transgenic lines (JAX stock #023527 and #031562, respectively) were used in the study. |
| Wild animals            | This study did not involve wild animals.   |
| Field-collected samples | This study did not involve samples collected from the field.   |
| Ethics oversight        | All procedures were approved by the Institutional Animal Care and Use Committee (IACUC) of Baylor College of Medicine.   |

Note that full information on the approval of the study protocol must also be provided in the manuscript.