Shallow Unsupervised Models Best Predict Neural Responses in Mouse Visual Cortex

Aran Nayebi* Nathan C. L. Kong*
Chengxu Zhuang Justin L. Gardner
Anthony M. Norcia Daniel L. K. Yamins

Stanford University

Abstract

Task-optimized deep convolutional neural networks are the most quantitatively accurate models of the primate ventral visual stream. However, such networks are implausible as models of the mouse visual system because mouse visual cortex has both lower retinal resolution and a shallower hierarchy than the primate. Moreover, the category supervision deep networks typically receive is neither ethologically relevant to the mouse in semantic content, nor realistic in quantity. As a result, standard supervised deep neural networks have proven quantitatively ineffective at modeling mouse visual data. Here, we develop and evaluate models that remedy these structural and functional gaps. We first demonstrate that shallow hierarchical architectures applied to lower resolution images improve match to neural responses, both in electrophysiological and calcium imaging data. We then show that networks trained using contrastive embedding methods, a recent unsupervised learning objective that requires no semantic labeling, achieve neural prediction performance that substantially exceed that of the same architectures trained in a supervised manner, across a wide variety of architecture types. Combining these better structural and functional priors yields models that are the most quantitatively accurate match to mouse visual responses to natural scenes, significantly surpassing that of prior attempts using primate-specific models, and approaching the inter-animal consistency level of the data itself. We further find that these shallow unsupervised models transfer to a wide variety of non-categorical visual tasks better than categorization-trained models. Taken together, our results suggest that mouse visual cortex is a low-resolution, shallow network that makes best use of the mouse's limited resources to create a light-weight, general-purpose visual system – in contrast to the deep, high-resolution, and more task-specific visual system of primates.

1 Introduction

In systems neuroscience, the mouse has become an indispensable model organism, allowing unprecedented genetic and experimental control at the level of cell-type specificity in individual circuits (Huberman and Niell, 2011). Beyond fine-grained control, studies of mouse visual behavior have revealed a multitude of abilities, ranging from stimulus-reward associations, to goal-directed navigation, and object-centric discriminations. These behaviors suggest that the mouse visual system is capable of supporting higher-order functions, and prior physiological studies provide evidence that higher visual cortical areas might subserve such behaviors (Glickfeld and Olsen, 2017). A natural question, therefore, is what these populations of neurons code for during visually-guided

*Equal contribution. Correspondence: {anayebi;nclkong}@stanford.edu.

behaviors. Formal computational models are needed to test these hypotheses: if optimizing a model for a certain task leads to accurate predictions of neural responses, then that task may provide a unified, normative account for why those population responses occur in the brain.

Deep convolutional neural networks (CNNs) are a class of models that have had immense success as predictive models of the human and non-human primate ventral visual stream (e.g., Yamins et al., 2014; Khaligh-Razavi and Kriegeskorte, 2014; Güçlü and van Gerven, 2015; Cichy et al., 2016; Cadena et al., 2019a; Bashivan et al., 2019). In contrast with the strong correspondence between task-optimized CNNs and the primate visual system, these CNNs are poor predictors of neural responses in mouse visual cortex (Cadena et al., 2019b).

Three fundamental problems, each grounded in the goal-driven modeling approach (Yamins and DiCarlo, 2016), confront these primate ventral stream models as potential models of the mouse visual system. Firstly, these models are too deep to be plausible models of the mouse visual system, since mouse visual cortex is known to be more parallel and much shallower than primate visual cortex (Harris et al., 2019; Siegle et al., 2021; Felleman and Van Essen, 1991). Secondly, they are trained in a supervised manner on ImageNet (Schrimpf et al., 2018; Conwell et al., 2020), which is an image set containing over one million images belonging to one thousand, mostly human-relevant, semantic categories (Deng et al., 2009). While such a dataset is an important technical tool for machine learning, it is highly implausible as a biological model particularly for rodents, who do not receive such category labels over development. Finally, mice are known to have lower visual acuity than that of primates (Prusky et al., 2000; Kiorpes, 2019), suggesting that the resolution of the inputs to mouse models should be lower than that of the inputs to primate models. Given these three differences between the visual system of primates and of mice, one cannot simply use current supervised primate ventral stream models as models of the mouse visual system.

The failure of these current models may therefore be tied to a failure in the application of the principles of goal-driven modeling to mouse vision, having to do with a mismatch between the model's architecture and task and those of the system being investigated. We addressed these three differences between the primate and mouse visual system by training shallower CNN architectures in an *unsupervised* manner using lower-resolution images. First, we noticed that AlexNet (Krizhevsky et al., 2012), which had the shallowest hierarchical architecture, provided strong correspondence to neural responses in mouse visual cortex. However, the deepest layers of AlexNet did not correspond well in neural predictivity to any mouse visual area, suggesting that even this architecture is too deep to be a completely physically matched model of the system. Therefore, we developed a class of novel shallower architectures with multiple parallel streams ("StreamNets") based on the AlexNet architecture. The parallel streams mimic the intermediate and higher visual areas identified in mice, informed by empirical work on the mouse visual hierarchy (Harris et al., 2019; Siegle et al., 2021). These StreamNets were able to achieve neural predictive performance competitive with that of AlexNet, while also maintaining a match between each model layer and a mouse visual area.

We then addressed the strong supervision signals used in the standard ImageNet categorization task by turning to a spectrum of unsupervised objectives including sparse autoencoding (Olshausen and Field, 1996), image-rotation prediction (Gidaris et al., 2018), and contrastive embedding objectives (Wu et al., 2018; Chen et al., 2020a,b; Chen and He, 2020), as well as supervised tasks with less category labels (CIFAR-10) or ethologically relevant labels that might be available to the mouse (e.g., depth information provided by whiskers; Quist et al., 2014; Huet and Hartmann, 2016; Zhuang et al., 2017), all while operating on lower-resolution images.

Finally, we found that lowering the resolution of the inputs during model training led to improved correspondence with the neural responses across model architectures, including current deep CNNs (VGG16 and ResNet-18) used in prior comparisons to mouse visual data (Cadena

et al., 2019b; Shi et al., 2019; de Vries et al., 2020; Conwell et al., 2020). Thus, strong constraints even at the level of input transformations improve model correspondence to the mouse visual system, although there remains a small gap between these models and the inter-animal consistency ceiling.

Overall, shallow architectures (our StreamNet variants and AlexNet) trained on unsupervised contrastive objectives using lower-resolution inputs yielded the best match to neural response patterns in mouse visual cortex, substantially improving the matches achieved by any of the supervised models we considered and approached the inter-animal consistency ceiling. Moreover, we show the resulting system is behaviorally different from a deep supervised network in a key qualitative fashion: unlike deep supervised neural networks, which are (comparatively speaking) categorization specialists, these shallow unsupervised networks are "general purpose" visual machines, achieving better transfer performance to a variety visual tasks.

Taken together, our best models of the mouse visual system suggest that it is a shallow, general-purpose system operating on lower-resolution inputs. These identified factors therefore provide interpretable insight into the confluence of evolutionary constraints that gave rise to the system in the first place, suggesting that these factors were crucially important given the ecological niche in which the mouse is situated, and the resource limitations to which it is subject.

2 Determining the animal-to-animal mapping transform

How should we map a neural network to mouse visual responses? What firing patterns of mouse visual areas are common across multiple animals, and thus worthy of computational explanation? A natural approach would be to map neural network features to mouse neural responses in the same manner that different animals can be mapped to each other. Specifically, we aimed to identify the best performing class of similarity transforms needed to map the firing patterns of one animal's neural population to that of another (inter-animal consistency; Figure 1A). We took inspiration from methods that have proven useful in modeling primate and human visual, auditory, and motor cortex (Yamins and DiCarlo, 2016; Kell et al., 2018; Michaels et al., 2020; Nayebi et al., 2021). As with other cortical areas, this transform class likely cannot be so strict as to require fixed neuron-to-neuron mappings between cells. However, the transform class for each visual area also cannot be so loose as to allow an unconstrained nonlinear mapping, since the model already yields an image-computable nonlinear response.

We explored a variety of linear mapping transform classes (fit with different constraints) between the population responses for each mouse visual area, as illustrated in Figure 1. The mouse visual responses to natural scenes were collected previously using both two-photon calcium imaging and Neuropixels by the Allen Institute (de Vries et al., 2020; Siegle et al., 2021). For all methods, the corresponding mapping was trained on 50% of all the natural scene images, and evaluated on the remaining held-out set of images (Figure 1B, see supplement for more details). We also included representational similarity analyses (RSA, Kriegeskorte et al., 2008) as a baseline measure of population-wide similarity across animals, corresponding to no selection of individual units, unlike the other mapping transforms. For the strictest mapping transform (One-to-One), each target unit was mapped to the single most correlated unit in the source animal. Overall, the One-to-One mapping and linear regression with sparseness priors (Lasso and ElasticNet) tended to yield the lowest inter-animal consistency among the maps considered. In some cases, they led to large error bars in some visual areas, implying an inconsistent fit. However, Ridge regression (L2-regularized), and PLS (Partial Least Squares) regression were more effective at the inter-animal mapping, yielding the most consistent fits across visual areas, with PLS regression providing the

highest inter-animal consistency. This result implies that an appropriate transform between visual areas in different mice (at least in these datasets) is a linear transform that incorporates a substantial proportion of source animal units to map to each unit in the target animal. We therefore use this *same* transform class by which to evaluate candidate models.

We further noticed a large difference between the inter-animal consistency obtained via RSA and the consistencies achieved by any of the other mapping transforms for the responses in VISrl of the calcium imaging dataset (green in Figure 1B). However, this difference was not observed for responses in VISrl in the Neuropixels dataset. This discrepancy suggested that there was a high degree of population-level heterogeneity in the responses collected from the calcium imaging dataset, which may be attributed to the fact that the two-photon FOV for VISrl spanned the boundary between the visual and somatosensory cortex, as originally noted by de Vries et al. (2020). We therefore excluded it from further analyses, following Siegle et al. (2020), who systematically compared these two datasets. Thus, this analysis provided insight into the experiments from which the data were collected.

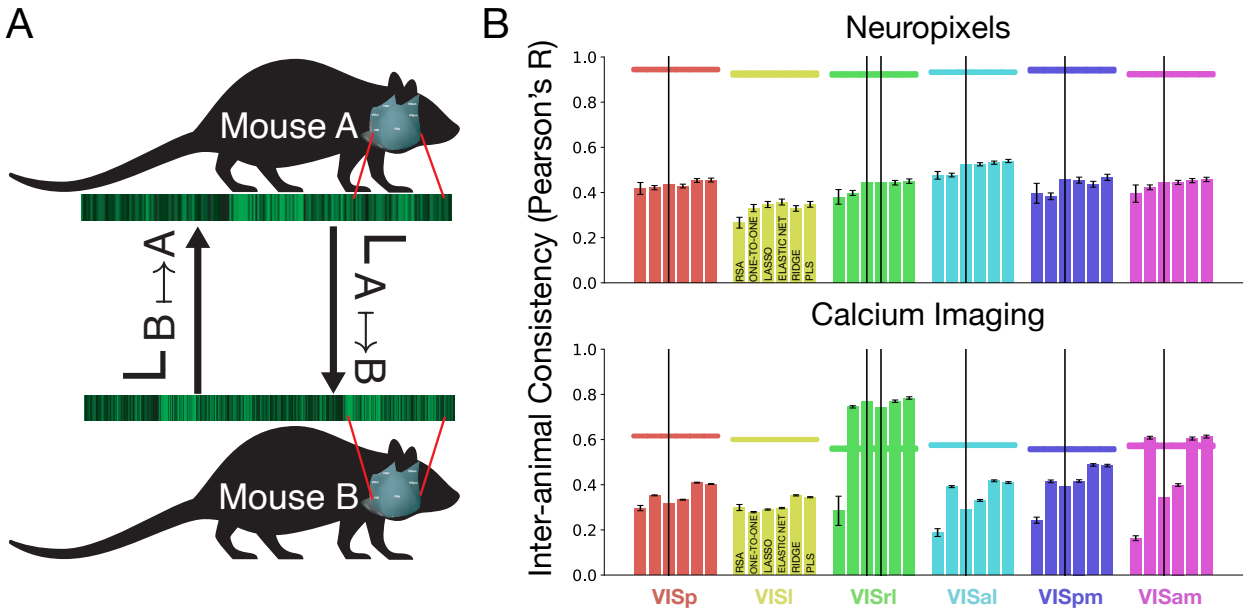

Figure 1: **Evaluating the inter-animal consistency of the neural data.** **A**. Calcium imaging and Neuropixels data were collected by the Allen Institute for six mouse visual areas: VISp, VISl, VISal, VISrl, VISam, VISpm. We assessed the neural data for their internal consistency (split-half reliability) and their inter-animal consistency, which tells us how well one animal corresponds to a pseudo-population of pooled source animals. Obtaining these metrics further allows us to determine how well *any* model can be expected to match the neural data, whereby each animal's visual responses are mapped onto other animal's visual responses. **B**. Inter-animal consistency was computed using different linear maps, showing that PLS regression provides the highest consistency. Horizontal bars at the top are the median and s.e.m. of the internal consistencies of the neurons in each visual area. Refer to Table S1 for N units per visual area.

3 Three key factors of quantitatively accurate goal-driven models of mouse visual cortex

We considered three primary ingredients that, when combined, yielded quantitatively accurate goal-driven models of mouse visual cortex: architecture (analogous to the wiring diagram), task (analogous to the visual behavior), and input resolution at which the system operates. Having established a consistent similarity transform class between animals across visual areas (PLS regression), we proceeded to map artificial neural network responses which varied in these three factors to mouse neural response patterns under this transform class. We delved into each factor individually before adjoining them, leading to the overall conclusion that the mouse visual system is most consistent with a low-resolution, shallow, and general-purpose visual system. These models approached 90% of the inter-animal consistency, significantly improving over the prior high-resolution, deep, and task-specific models (VGG16) which attained only 56.27% of this ceiling.

3.1 Architecture: Shallow architectures better predict mouse visual responses than deep architectures

The mouse visual system has a shallow hierarchy, in contrast to the primate ventral visual stream (Harris et al., 2019; Siegle et al., 2021; Felleman and Van Essen, 1991). We further corroborated this observation by examining the internal consistencies (i.e., split-half reliability) of the neurons in each visual area from the Neuropixels dataset at each 10-ms time bin, shown in the right panel of Figure 2A. The peak internal consistencies occurred in quick succession from 100-130 ms, starting from VISp (hierarchically the lowest visual area) and was consistent with the normalized hierarchy criterion of Siegle et al. (2021), reproduced in the left panel of Figure 2A, suggesting an overall three to four level architecture.

We found that the neural response predictions of a standard deep CNN model (VGG16), used in prior comparisons to mouse visual areas (Cadena et al., 2019b; Shi et al., 2019; de Vries et al., 2020), were quite far from the inter-animal consistency (56.27%). Retraining this model with images of resolution closer to the visual acuity of mice (64×64 pixels) improved the model's neural predictivity, reaching 67.7% of the inter-animal consistency. We dive deeper into the image resolution issue in Section 3.3.

We also reasoned that the substantial gap with the inter-animal consistency was partly due to the mismatch between the shallow hierarchy of the mouse visual system and the deep hierarchy of the model. Work by Shi et al. (2020) investigated the construction of a parallel pathway model based on information provided by large-scale tract tracing data, though this model was neither task-optimized nor compared to neural responses. We trained this network on (64×64 pixels) ImageNet categorization, and conducted a hyperparameter sweep to identify the learning parameters that yielded the best performance on the task. We also trained a variant of this network where the task readout was at the final model layer ("VISpor"), and found that this yielded an approximately 2% improvement in ImageNet categorization performance over the original model with its best hyperparameters. The neural predictivity of the original MouseNet and this variant were comparable on both datasets (see Figure 2C for neural predictivity on the Neuropixels dataset and Figure S3B for neural predictivity on the calcium imaging dataset).

For each visual area, the maximum neural predictivity of all of these models was worse than that of AlexNet (trained on 64×64 pixels images), which was the best (and shallowest) model among these architectures (Figure 2C). By examining neural predictivity of the best performing model (AlexNet) as a function of model layer, we found that peak neural predictivity did not

occur past the fourth convolutional layer (Figure 3B; orange lines), suggesting that an even shallower network architecture might be more appropriate (Figures 2B and S3A; orange lines). This result motivated the development of an architecture that is shallower than AlexNet (which we call "StreamNet"), that is more physically matched to the known shallower hierarchy of the mouse visual system, and has *no* model layers that are unassigned to any visual area.

Our StreamNet was based on the AlexNet architecture, up to the model layer of maximum predictivity across all visual areas, but allowed for potentially multiple parallel pathways with three levels based on the peak internal consistency timing observations (right panel of Figure 2A). This yielded an architecture of four convolutional layers, divided into three levels. The first level consisted of one convolutional layer and the intermediate level consisted of two convolutional layers. The final level has two "areas" in parallel (where each area consists of one convolutional layer), inspired by the observation that VISpor and VISam comprise the top-most levels of the mouse visual hierarchy (left panel of Figure 2A; Harris et al., 2019; Siegle et al., 2021). We additionally included dense skip connections from shallow levels to deeper levels, known from the feedforward connectivity of the mouse connectome (Harris et al., 2016; Knox et al., 2018). Taking into consideration the observation of Conwell et al. (2020) that thinner task-optimized networks yielded better fits to mouse visual data[1], we allowed the number of parallel streams N to be an architectural variable. We set the number of parallel streams to be one (as a control; denoted "single-stream"), two (to mimic a potential ventral/dorsal stream distinction; denoted "dual-stream"), and six (to more closely match the known number of intermediate visual areas: VISl, VISli, VISal, VISrl, VISpl, and VISpm; denoted "six-stream"). Figure 2B shows a schematic of our StreamNet architecture. We found that our StreamNet model variants were always more predictive of the neural data across all the visual areas than the MouseNet of Shi et al. (2020) and attain comparable predictivity as AlexNet (Figure 2C).

3.2 Task: Unsupervised, contrastive objectives, instead of supervised categorization, improves predictions of mouse visual responses

Training neural network models on 1000-way ImageNet categorization is useful for obtaining visual representations that are well-matched to those of the primate visual system (Schrimpf et al., 2018; Zhuang et al., 2021) and seems to result in the best *supervised* models for rodent visual cortex (Conwell et al., 2020). However, it is unclear that rodents can perform well on large-scale object recognition tasks when trained (attaining approximately 70% on a two-alternative forced-choice object classification task, Froudarakis et al., 2020), such as those where there are hundreds of labels. Furthermore, the categories of the ImageNet dataset are rather human-centric and therefore not entirely ethologically relevant for rodents to begin with.

We therefore considered unsupervised losses instead, as these may provide more general goals for the models beyond the specifics of (human-centric) object categorization. Advances in computer vision have yielded algorithms that are powerful unsupervised visual representation learners, and models trained in those ways are quantitatively accurate models of the primate ventral visual stream (Zhuang et al., 2021). Reducing the image size during task training based on rodent visual acuity, which we show in Section 3.3 to be important to provide good neural predictivity when controlling for task and architecture, further set a constraint on the type of unsupervised algorithms we considered. Specifically, algorithms that involved crops were unlikely candidates, as the resultant crop would be too small to be effective or too small to have non-trivial features downstream of the network due to the architecture (e.g., relative location prediction or contrastive

[1]We note, however, that this conclusion was made without using models that were trained on images with the adjusted size.

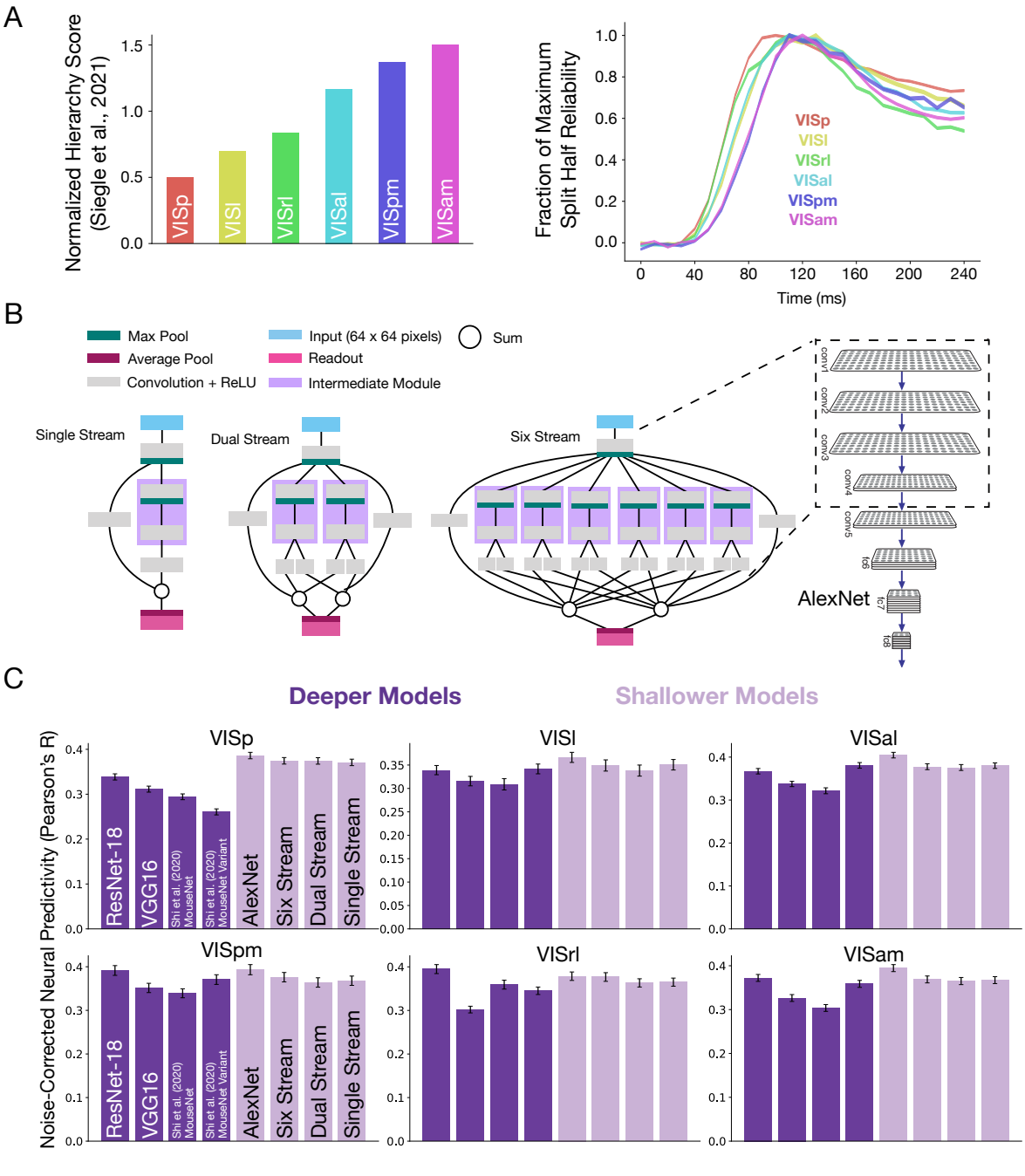

Figure 2: **Hierarchically shallow models achieve competitive neural predictivity performance (Neuropixels dataset).** **A**. Left: Normalized hierarchy scores for each mouse visual area from Figure 2a of Siegle et al. (2021). Higher score indicates that the visual area is higher in the mouse visual hierarchy. Right: Fraction of maximum split-half reliability for each visual area as a function of time computed from the Neuropixels dataset. **B**. We found that the first four convolutional layers of AlexNet best corresponded to all the mouse visual areas (panel C). These convolutional layers were used as the basis for our StreamNet architecture variants. **C**. AlexNet and our StreamNet variants (light purple) provide neural predictivity on the Neuropixels dataset that is better or at least as good as those of deeper architectures (dark purple). Refer to Table S1 for N units per visual area. See Figure S3 for neural predictivity on the calcium imaging dataset.

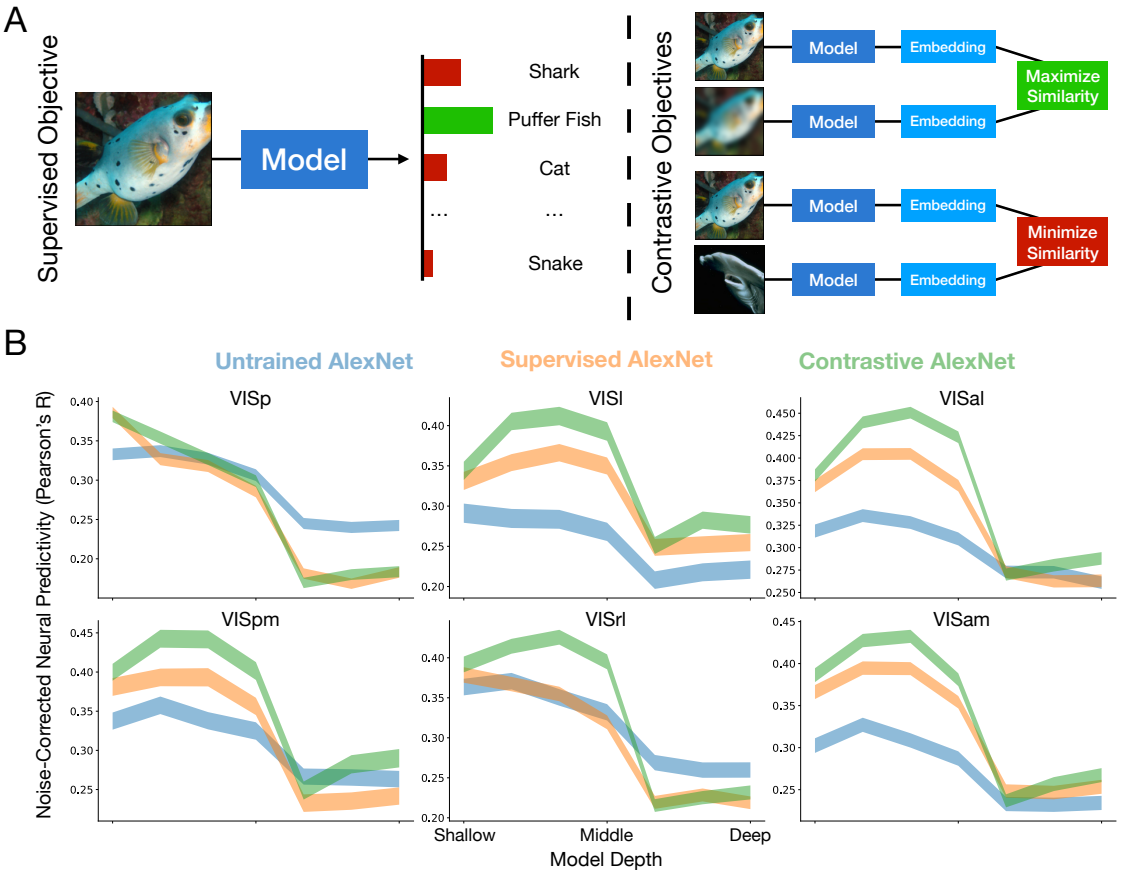

Figure 3: **Unsupervised models better predict the neural responses in mouse visual cortex (Neuropixels dataset). A**. Models can be trained in either a supervised or an unsupervised contrastive manner. In supervised training (left), an image is used as input for a model and the model's prediction (bars) is compared with the labels. In unsupervised contrastive training (right), models are trained so that embeddings of augmentations of an image are more similar to each other (upper two rows) than to the embeddings of another image (lower two rows). **B**. Neural predictivity, using PLS regression, on the Neuropixels dataset across AlexNet architectures trained in two different ways (supervised [orange] and unsupervised [green]). We observe that the first four convolutional layers provide the best fits to the neural data while the latter three layers are not very predictive for any visual area, suggesting that an even shallower architecture may be suitable. This is further corroboration for our architectural decision in Figure 2B. See Figure S4 for neural predictivity on the calcium imaging dataset.

predictive coding (CPC) for static images, Doersch et al., 2015; Oord et al., 2018). We instead considered objective functions that use image statistics from the *entire* image. As control models, we used relatively less powerful unsupervised algorithms including the sparse autoencoder (Olshausen and Field, 1996), depth-map prediction, and image-rotation prediction (RotNet, Gidaris et al., 2018). Advances in unsupervised learning have shown that training models on contrastive objective functions yields representations that can support strong performance on downstream object categorization tasks. Thus, the remaining four algorithms we used were from the family of contrastive objective functions: instance recognition (IR, Wu et al., 2018), simple framework for contrastive learning (SimCLR, Chen et al., 2020a), momentum contrast (MoCov2, Chen et al., 2020b), and simple siamese representation learning (SimSiam, Chen and He, 2020).

At a high-level, the goal of these contrastive objectives is to learn a representational space where embeddings of augmentations for one image (i.e., embeddings for two transformations of the *same* image) are more "similar" to each other than to embeddings of other images (schematized in Figure 3A). We found that a model trained with these contrastive objectives resulted in higher neural predictivity across all the visual areas than a model trained on supervised object categorization, for the best model architecture class in Section 3.1 (i.e., AlexNet) (Figure 3B). We systematically explored the space of architecture and objective function combinations in Section 3.4, and found that this observation holds more generally.

3.3 Data stream: Task-optimization on images of lower resolution improves predictions of mouse visual responses

The visual acuity of mice is known to be lower than the visual acuity of primates (Prusky et al., 2000; Kiorpes, 2019). We briefly mentioned previously that task-optimization with images of lower-resolution is important in building models of the mouse visual system. Here we delved deeper and investigated how neural predictivity performances of two (shallower) architectures varied as a function of the image resolution at which models were trained. A schematic of this is shown in Figure 4A. We trained our dual stream variant in an unsupervised manner (instance recognition) using image resolutions that varied from 32×32 pixels to 224×224 pixels. Similarly, we trained AlexNet on instance recognition using image resolutions that varied from 64×64 pixels to 224×224 pixels. 64×64 pixels was the minimum image size for AlexNet due to its additional max-pooling layer. In both cases, 224×224 pixels is the image resolution that is typically used to train neural network models of the primate ventral visual stream.

Training models using resolutions lower than what is used for primate models indeed improves neural predictivity across all visual areas, shown in Figure 4B. Although the input resolution of 64×64 pixels may not be optimal for different architectures, it is the resolution that we used to train all the models. This was motivated by the observation that the upper bound on mouse visual acuity is 0.5 cycles / degree (Prusky et al., 2000), corresponding to 2 pixels / cycle $\times$ 0.5 cycles / degree $=$ 1 pixel / degree, so that a simplified correction for the appropriate visual acuity would correspond to 64×64 pixels, which was also used in Shi et al. (2019) and in the MouseNet of Shi et al. (2020). A more thorough investigation into the appropriate image transformations, however, may be needed.

Overall, our observations suggest that a change in the task via a simple change in the image statistics (i.e., data stream) is crucial to obtain an appropriate model of mouse visual encoding. This further suggests that mouse visual encoding is the result of task-optimization at a lower "visual acuity" than what is typically used for primate ventral stream models.

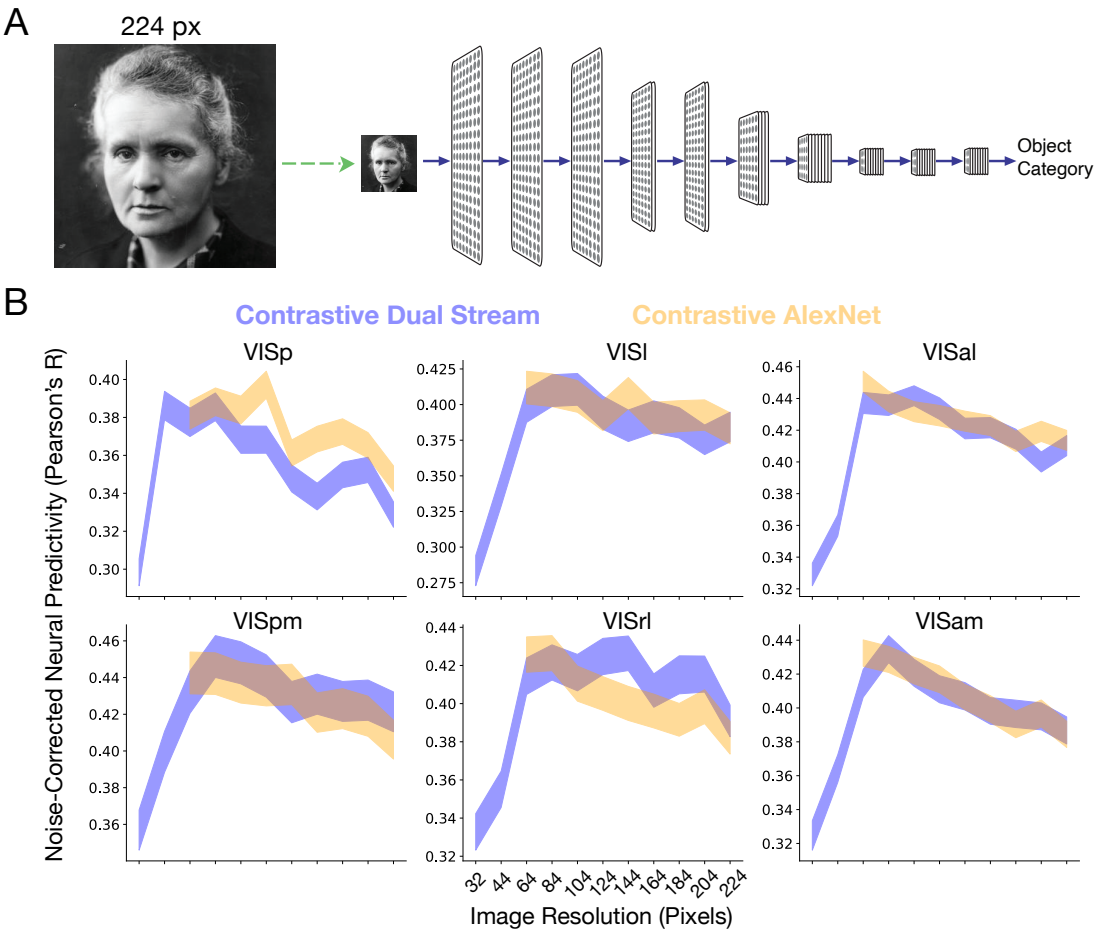

Figure 4: **Lower image resolution during model training improves task-optimized neural predictivity (Neuropixels dataset). A**. Models with "lower visual acuity" were trained using lower-resolution ImageNet images. Each image was downsampled from 224×224 pixels, which is the size typically used to train primate ventral stream models, to various image sizes. **B**. We trained our dual stream variant (blue) and AlexNet (orange) on instance recognition using various image sizes ranging from 32×32 pixels to 224×224 pixels and computed their neural predictivity performance for each mouse visual area. Training models on resolutions lower than 224×224 pixels generally led to improved correspondence with the neural responses for both models. The median and s.e.m. across neurons in each visual area is reported. Refer to Table S1 for N units per visual area. See Figure S5 for neural predictivity on the calcium imaging dataset.

3.4 Putting it all together: Shallow architectures trained on contrastive objectives with low-resolution inputs best capture neural responses throughout mouse visual cortex

Here we combined all three ingredients, varying the architecture and task at the visual acuity of the rodent.

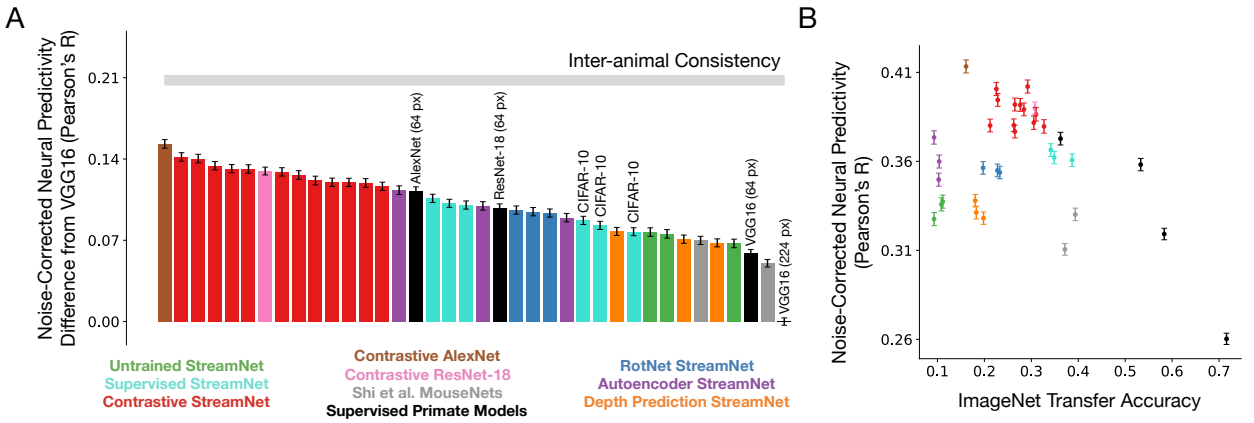

Figure 5: **Shallow architectures trained with contrastive objective functions yield the best matches to the neural data (Neuropixels dataset). A**. The median and s.e.m. neural predictivity, using PLS regression, across units in all mouse visual areas. $N = 1731$ units in total. Red denotes our StreamNet models trained on contrastive objective functions, blue denotes our StreamNet models trained on RotNet, turquoise denotes our StreamNet models trained in a supervised manner on ImageNet and on CIFAR-10, green denotes untrained models (random weights), orange denotes our StreamNet models trained depth prediction, purple denotes our StreamNet models trained on autoencoding, brown denotes contrastive AlexNet, pink denotes contrastive ResNet-18 (both trained on instance recognition), black denotes the remaining ImageNet supervised models (primate ventral stream models), and grey denotes the MouseNet of Shi et al. (2020) and our variant of this architecture. Actual neural predictivity performance can be found in Table S3. **B**. Each model's performance on ImageNet is plotted against its median neural predictivity across all units from each visual area. All ImageNet performance numbers can be found in Table S3. Color scheme as in **A**. See Figure S1 for neural predictivity on the calcium imaging dataset.

We found that AlexNet and our unsupervised StreamNet model variants outperformed all the other models (Figure 5A). Furthermore, when those models were trained with contrastive objectives, they had the highest neural predictivity, as shown by the red and brown bars on the left of Figure 5A, attaining close to 90% of the inter-animal consistency ceiling. However, there was no clear separation in neural predictivity among the different contrastive objectives. Among the unsupervised algorithms, contrastive objectives had the highest 64×64 pixels ImageNet transfer performance (red vs. blue/orange/purple in Figures 5B and S2B), indicating that powerful unsupervised loss functions are crucial for explaining the variance in the neural responses.

Higher ImageNet categorization performance also did not correspond to higher neural predictivity, in contrast to findings in models of the primate ventral visual stream (Yamins et al., 2014; Schrimpf et al., 2018). Specifically, deeper, purely supervised models that attain greater than 40% accuracy had, on average, the least match to the neural data (black dots in Figures 5B and S2B). Moreover, object recognition tasks with less categories (e.g., 10 categories in CIFAR-10, Krizhevsky et al., 2009) did not improve neural predictivity for the *same* architecture trained on ImageNet

(turquoise bars in Figure 5A).

As a positive control, we optimized ResNet-18 on a contrastive objective function (pink in Figure 5) and found that although changing the objective function improved neural predictivity for ResNet-18 over its supervised counterpart, it was still worse than the shallower AlexNet trained using a contrastive objective (compare pink and brown points in Figures 5A and 5B). This indicates that having an appropriately shallow architecture contributes to neural predictivity, but even with a less physically realistic deep architecture such as ResNet-18, you can greatly improve neural predictivity with a contrastive embedding loss function. These findings are consistent with the idea that appropriate combinations of objective functions and architectures are necessary to build quantitatively accurate models of neural systems, with the objective function providing a strong constraint especially when coupled with a shallow architecture (Yamins and DiCarlo, 2016).

4 Mouse visual cortex is a general-purpose visual system

Given the strong correspondence between unsupervised models trained with contrastive embedding objectives, especially relative to their supervised and untrained counterparts (the latter being an objective-function-independent control), here we delve into why this might be the case. In other words, given that improvement on ImageNet categorization was *not* related to improved neural predictivity (Figure 5B), are the representations in these unsupervised networks better serving some other downstream ecological niche compared to more task-specific objectives (e.g., object categorization)?

Our hypothesis was that non-task-specific unsupervised training resulted in more *general* visual representations. Specifically, we explored whether contrastive models provided visual representations useful for different downstream tasks that rodents *might* perform, such as object-centric visual behaviors (Zoccolan et al., 2009) and non-object-centric behaviors that may be useful for guiding navigation, typically attributed to dorsal areas (Wang and Burkhalter, 2013). We considered two primary datasets for these tasks. The first image set was previously used to assess both neural and behavioral consistency of supervised and unsupervised neural networks (Yamins et al., 2014; Rajalingham et al., 2018; Schrimpf et al., 2018; Zhuang et al., 2021), on which we evaluated transfer performance on object categorization and other object-centric visual tasks independent of object category, including object position localization, size estimation, and pose estimation. The second dataset was focused on texture discrimination (Cimpoi et al., 2014), which we used as a proxy for the subset of non-object-centric visual behaviors which may be relevant to navigation and the exploration of novel environments.

We assessed the transfer performance of the unsupervised models by adding a single fully-connected linear readout layer to each layer of the three StreamNet variants, trained with the supervised categorization and the unsupervised loss functions. Since different tasks could, in theory, be best supported by different layers of the unsupervised networks, we reported the cross-validated performance values for the best model layer. We note that these performance values are all on held-out images not used during the training of any of the original networks.

As shown in Figure 6, we found that, across all evaluated objective functions and StreamNet architectural variants, the contrastive embedding objectives (red bars) showed substantially better linear task transfer performance than other unsupervised methods (purple and blue bars), as well as ethologically-relevant supervision such as depth prediction (orange bars). Furthermore, the contrastive embedding objectives either approached or exceeded, the performance of networks trained on supervised ImageNet categorization (turquoise bars). Taken together, these results suggest that contrastive embedding methods have achieved a generalized improvement in the quality

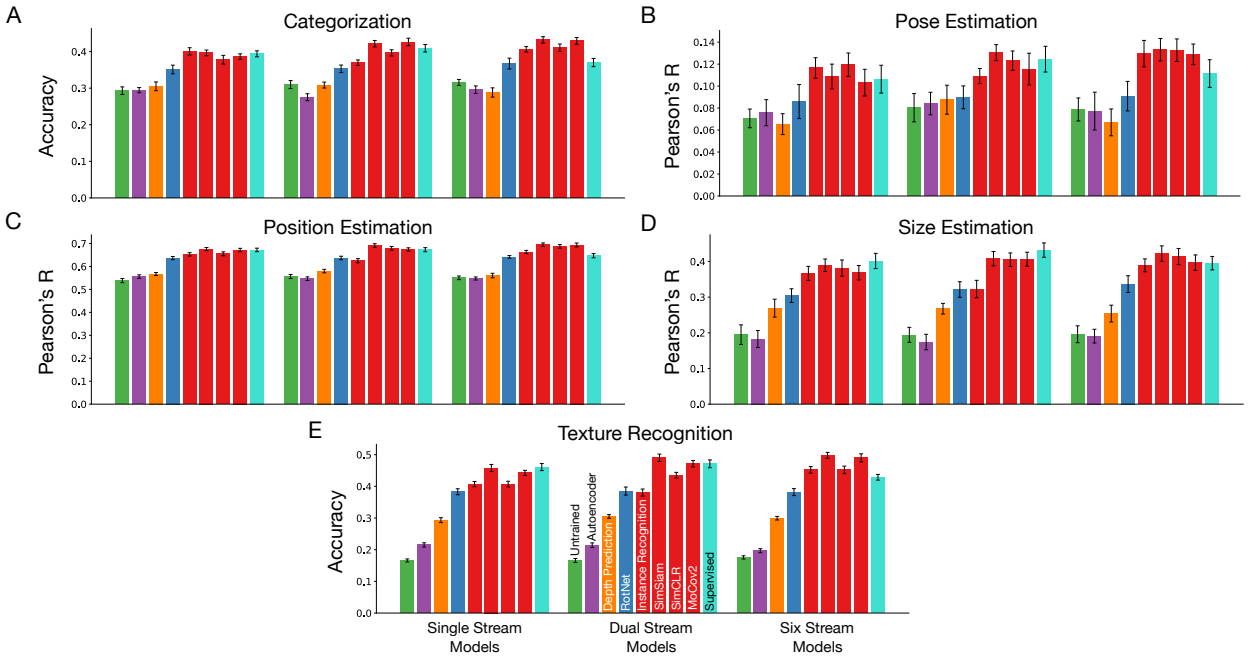

Figure 6: **Evaluating visual representations of StreamNet variants learned in an unsupervised manner on object-centric and non-object-centric visual tasks.** Red denotes our StreamNet variants trained on contrastive objective functions. Blue denotes our models trained on rotation prediction (RotNet), orange denotes our depth prediction models, purple denotes autoencoding models, green are the untrained model. The average performance and its standard deviation (mean and s.t.d.) across 10 train-test image splits is reported for each transfer task. **A.** Maximum linear transfer performance across model layers on the categorization of objects that are highly varied in terms of their rotation, sizes, and positions in the image. **B.** Object pose estimation accuracy. **C.** Object position estimation accuracy. **D.** Object size estimation accuracy. **E.** Maximum linear transfer performance on 47-way texture classification (a non-object-centric task).

of the visual representations they create, enabling a diverse range of visual behaviors, providing evidence for their potential as computational models of mouse visual cortex, across ventral and dorsal areas.

5 Discussion

In this work, we showed that shallow architectures trained with contrastive embedding methods operating on lower-resolution images most accurately predict image-evoked neural responses across visual areas in mice, surpassing the predictive power of supervised methods. Deep CNNs supervised on ImageNet categorization, which were previously used as models of mouse visual cortex and are quantitatively the best models of the *primate* ventral stream, were comparatively poor predictors of neural responses throughout mouse visual cortex. Our best models approached the computed inter-animal consistency of all measured units on both neural datasets. Taken together with recent work done in primates (Zhuang et al., 2021), the results indicate that contrastive objectives appear to best explain responses across both rodent and primate species, suggesting that these objectives may be part of a species-general toolkit.

Unlike the situation in the primate ventral visual stream, however, contrastive objectives *surpassed* the neural predictivity of their supervised counterparts, as increased categorization performance lead to overall *worse* correspondence to mouse visual areas. We observe that the advantage of these contrastive objectives is that they provide representations that are generally improved over those obtained by supervised methods in order to enable a diverse range of visual behaviors. We additionally found that neural networks of larger sizes, either measured by network parameters (analogous to the number of synapses) or network units (analogous to the number of neurons), had comparatively lower neural predictivity (Figure S6). These results suggest that the mouse visual cortex is a light-weight, shallow, low-resolution, and general-purpose visual system in contrast to the deep, high-resolution, and more task-specific visual system in primates. Our improved models of the mouse visual system therefore provide a different view of its goals and constraints than that provided by (comparatively) high-resolution, deep feedforward categorization models. Furthermore, the generic nature of these unsupervised contrastive objective functions suggests the intriguing possibility that they might be used by other sensory systems, such as in barrel cortex or the olfactory system.

In fact, these results, coupled with the fact that larger, deeper networks (which are relatively better models of primate ventral visual responses than shallow networks) are among the worst models of mouse visual cortex, demonstrates a double dissociation between the mouse-like architectures and tasks and the primate-like architectures and tasks. Thus, the failure of the "blind application" of deep networks to capture mouse data well – and the subsequent success of our more structurally-and-functionally tuned approach – illustrates not a weakness of the goal-driven neural network approach, but instead a strength of this methodology's ability to be appropriately sensitive to salient biological differences.

Overall, we have made progress in modeling the mouse visual system in three core ways: the choice of architecture class, objective function, and the data stream.

On the architectural front, we introduced StreamNets, a novel shallow and multi-stream model, which we think is a reasonable starting point for building more accurate mouse vision models. Specifically, our focus in this work was on feedforward models, but there are many feedback connections from higher visual areas to lower visual areas (Harris et al., 2019). Incorporating these architectural motifs into our models and training these models using dynamic inputs may be useful for modeling temporal dynamics in mouse visual cortex, as has been recently done in

primates (Nayebi et al., 2018; Kubilius et al., 2019; Nayebi et al., 2021).

We also demonstrated that unsupervised, contrastive embedding functions are critical goals for a system to accurately match responses in the mouse visual system. Thus, towards incorporating recurrent connections in the architecture, we would also like to probe the functionality of these feedback connections in scenarios with temporally-varying, dynamic inputs. Concurrent work of Bakhtiari et al. (2021) used the unsupervised predictive objective of CPC (Oord et al., 2018) to model neural responses of mouse visual cortex to natural movies. Given that our best performing unsupervised methods obtained good representations on static images by way of contrastive learning, it would be interesting to explore a larger spectrum of more object-centric unsupervised signals operating on dynamic inputs, such as in the context of forward prediction (e.g., Mrowca et al., 2018; Haber et al., 2018; Lingelbach et al., 2020).

Moreover, we found that constraining the input data so that they are closer to those received by the mouse visual system, was important for improved correspondence – specifically, resizing the images to be smaller during training as a proxy for low-pass filtering. We believe that future work could investigate other appropriate low-pass filters and ethologically relevant pixel-level transformations to apply to the original image or video stream. These additional types of input transformations will likely also constrain the types of unsupervised objective functions that can be effectively deployed in temporally-varying contexts, as it did in our case for static images.

Finally, our inter-animal consistency measurements make a clear recommendation for the type of future neural datasets that are likely to be helpful in more sharply differentiating future candidate models. In Figure S7A, we observed that when fitting linear maps between animals to assess inter-animal consistency, fitting values are significantly higher in training than on the evaluation (test) set, indicating that the number of stimuli is not large enough to prevent overfitting when identifying source animal neuron(s) to match any given target neuron. Furthermore, as a function of the number of stimuli, the test set inter-animal consistencies steadily increases (see Figure S7B), and likely would continue to increase substantially if the dataset had more stimuli. Thus, while much focus in methods has been on increasing the number of neurons contained in a given dataset (Steinmetz et al., 2021), our analysis indicates that the main limiting factor in model identification is *number of stimuli* in the dataset, rather than the number of neurons. In our view, future experiments should preferentially focus more resources on increasing stimulus count. Doing so would likely raise the inter-animal consistency, in turn providing substantially more dynamic range for separating models in terms of their ability to match the data, and thereby increasing the likelihood that more specific conclusions about precisely which circuit structure(s) (Collins et al., 2017; Bergstra et al., 2015) and which specific (combinations of) objectives (e.g., Wu et al., 2018; Chen and He, 2020; Chen et al., 2020a) best describe mouse visual cortex.

Acknowledgements

We thank Shahab Bakhtiari, Katherine L. Hermann, and Akshay Jagadeesh for helpful discussions, and Eshed Margalit and Xiaoxuan Jia for helpful feedback on the manuscript. N.C.L.K. is supported by the Stanford University Ric Weiland Graduate Fellowship. J.L.G. acknowledges the generous support of Research to Prevent Blindness and Lions Club International Foundation (`https://www.rpbusa.org/rpb/low-vision/`). A.M.N. is supported by the Stanford Institute for Human Centered Artificial Intelligence. D.L.K.Y. is supported by the James S. McDonnell Foundation (Understanding Human Cognition Award Grant No. 220020469), the Simons Foundation (Collaboration on the Global Brain Grant No. 543061), the Sloan Foundation (Fellowship FG-2018-10963), the National Science Foundation (RI 1703161 and CAREER Award 1844724), the DARPA Machine Common Sense program, and hardware donation from the NVIDIA Corporation.

References

Shahab Bakhtiari, Patrick Mineault, Timothy Lillicrap, Christopher Pack, and Blake Richards. The functional specialization of visual cortex emerges from training parallel pathways with self-supervised predictive learning. *bioRxiv*, 2021.

Pouya Bashivan, Kohitij Kar, and James J DiCarlo. Neural population control via deep image synthesis. *Science*, 364(6439), 2019.

James Bergstra, Brent Komer, Chris Eliasmith, Daniel Yamins, and David D Cox. Hyperopt: a python library for model selection and hyperparameter optimization. *Computational Science & Discovery*, 8(1), 2015.

Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

Santiago A Cadena, George H Denfield, Edgar Y Walker, Leon A Gatys, Andreas S Tolias, Matthias Bethge, and Alexander S Ecker. Deep convolutional models improve predictions of macaque v1 responses to natural images. *PLoS Computational Biology*, 15(4):e1006897, 2019a.

Santiago A Cadena, Fabian H Sinz, Taliah Muhammad, Emmanouil Froudarakis, Erick Cobos, Edgar Y Walker, Jacob Reimer, Matthias Bethge, Andreas Tolias, and Alexander S Ecker. How well do deep neural networks trained on object recognition characterize the mouse visual system? *NeurIPS Neuro AI Workshop*, 2019b.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pages 1597–1607, 2020a.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. *arXiv preprint arXiv:2011.10566*, 2020.

Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020b.

Radoslaw Martin Cichy, Aditya Khosla, Dimitrios Pantazis, Antonio Torralba, and Aude Oliva. Comparison of deep neural networks to spatio-temporal cortical dynamics of human visual object recognition reveals hierarchical correspondence. *Scientific Reports*, 6(1):1–13, 2016.

M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

Jasmine Collins, Jascha Sohl-Dickstein, and David Sussillo. Capacity and trainability in recurrent neural networks. In *ICLR*, 2017.

Colin Conwell, Michael Buice, Andrei Barbu, and George Alvarez. Model zoology and neural taskonomy for better characterizing mouse visual cortex. *ICLR Bridging AI and Cognitive Science (BAICS) Workshop*, 2020.

Saskia EJ de Vries, Jerome A Lecoq, Michael A Buice, Peter A Groblewski, Gabriel K Ocker, Michael Oliver, David Feng, Nicholas Cain, Peter Ledochowitsch, Daniel Millman, et al. A large-scale standardized physiological survey reveals functional organization of the mouse visual cortex. *Nature Neuroscience*, 23(1):138–151, 2020.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1422–1430, 2015.

Daniel J Felleman and David C Van Essen. Distributed hierarchical processing in the primate cerebral cortex. *Cerebral Cortex*, 1(1):1–47, 1991.

Emmanouil Froudarakis, Uri Cohen, Maria Diamantaki, Edgar Y Walker, Jacob Reimer, Philipp Berens, Haim Sompolinsky, and Andreas S Tolias. Object manifold geometry across the mouse cortical visual hierarchy. *bioRxiv*, 2020.

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations*, 2018.

Lindsey L Glickfeld and Shawn R Olsen. Higher-order areas of the mouse visual cortex. *Annual Review of Vision Science*, 3:251–273, 2017.

Umut Güçlü and Marcel AJ van Gerven. Deep neural networks reveal a gradient in the complexity of neural representations across the ventral stream. *Journal of Neuroscience*, 35(27):10005–10014, 2015.

Nick Haber, Damian Mrowca, Stephanie Wang, Li Fei-Fei, and Daniel LK Yamins. Learning to play with intrinsically-motivated, self-aware agents. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 8398–8409, 2018.

Julie A Harris, Stefan Mihalas, Karla E Hirokawa, Jennifer D Whitesell, Hannah Choi, Amy Bernard, Phillip Bohn, Shiella Caldejon, Linzy Casal, Andrew Cho, et al. Hierarchical organization of cortical and thalamic connectivity. *Nature*, 575(7781):195–202, 2019.

Kameron D Harris, Stefan Mihalas, and Eric Shea-Brown. High resolution neural connectivity from incomplete tracing data using nonnegative spline regression. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

Ha Hong, Daniel LK Yamins, Najib J Majaj, and James J DiCarlo. Explicit information for category-orthogonal object properties increases along the ventral stream. *Nature neuroscience*, 19(4):613, 2016.

Andrew D Huberman and Cristopher M Niell. What can mice tell us about how vision works? *Trends in Neurosciences*, 34(9):464–473, 2011.

Lucie A Huet and Mitra JZ Hartmann. Simulations of a vibrissa slipping along a straight edge and an analysis of frictional effects during whisking. *IEEE transactions on haptics*, 9(2):158–169, 2016.

Alexander JE Kell, Daniel LK Yamins, Erica N Shook, Sam V Norman-Haignere, and Josh H McDermott. A task-optimized neural network replicates human auditory behavior, predicts brain responses, and reveals a cortical processing hierarchy. *Neuron*, 98(3):630–644, 2018.

Seyed-Mahdi Khaligh-Razavi and Nikolaus Kriegeskorte. Deep supervised, but not unsupervised, models may explain it cortical representation. *PLoS Computational Biology*, 10(11):e1003915, 2014.

Lynne Kiorpes. Understanding the development of amblyopia using macaque monkey models. *Proceedings of the National Academy of Sciences*, 116(52):26217–26223, 2019.

Joseph E Knox, Kameron Decker Harris, Nile Graddis, Jennifer D Whitesell, Hongkui Zeng, Julie A Harris, Eric Shea-Brown, and Stefan Mihalas. High-resolution data-driven model of the mouse connectome. *Network Neuroscience*, 3(1):217–236, 2018.

Nikolaus Kriegeskorte, Marieke Mur, and Peter A Bandettini. Representational similarity analysis-connecting the branches of systems neuroscience. *Frontiers in Systems Neuroscience*, 2:4, 2008.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25:1097–1105, 2012.

Jonas Kubilius, Martin Schrimpf, Kohitij Kar, Rishi Rajalingham, Ha Hong, Najib Majaj, Elias Issa, Pouya Bashivan, Jonathan Prescott-Roy, Kailyn Schmidt, et al. Brain-like object recognition with high-performing shallow recurrent anns. *Advances in Neural Information Processing Systems*, 32: 12805–12816, 2019.

Michael Lingelbach, Damian Mrowca, Nick Haber, Li Fei-Fei, and Daniel LK Yamins. Towards curiosity-driven learning of physical dynamics. *ICLR Bridging AI and Cognitive Science (BAICS) Workshop*, 2020.

Najib J Majaj, Ha Hong, Ethan A Solomon, and James J DiCarlo. Simple learned weighted sums of inferior temporal neuronal firing rates accurately predict human core object recognition performance. *Journal of Neuroscience*, 35(39):13402–13418, 2015.

Jonathan A Michaels, Stefan Schaffelhofer, Andres Agudelo-Toro, and Hansjörg Scherberger. A goal-driven modular neural network predicts parietofrontal neural dynamics during grasping. *Proceedings of the National Academy of Sciences*, 117(50):32124–32135, 2020.

Damian Mrowca, Chengxu Zhuang, Elias Wang, Nick Haber, Li Fei-Fei, Joshua B Tenenbaum, and Daniel LK Yamins. Flexible neural representation for physics prediction. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 8813–8824, 2018.

Aran Nayebi, Daniel M Bear, Jonas Kubilius, Kohitij Kar, Surya Ganguli, David Sussillo, James J DiCarlo, and Daniel LK Yamins. Task-driven convolutional recurrent models of the visual system. *Advances in Neural Information Processing Systems*, 31:5295—5306, 2018.

Aran Nayebi, Javier Sagastuy-Brena, Daniel M Bear, Kohitij Kar, Jonas Kubilius, Surya Ganguli, David Sussillo, James J DiCarlo, and Daniel LK Yamins. Goal-driven recurrent neural network models of the ventral visual stream. *bioRxiv*, 2021.

Bruno A Olshausen and David J Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Glen T Prusky, Paul WR West, and Robert M Douglas. Behavioral assessment of visual acuity in mice and rats. *Vision Research*, 40(16):2201–2209, 2000.

Brian W Quist, Vlad Seghete, Lucie A Huet, Todd D Murphey, and Mitra JZ Hartmann. Modeling forces and moments at the base of a rat vibrissa during noncontact whisking and whisking against an object. *Journal of Neuroscience*, 34(30):9828–9844, 2014.

Rishi Rajalingham, Elias B Issa, Pouya Bashivan, Kohitij Kar, Kailyn Schmidt, and James J DiCarlo. Large-scale, high-resolution comparison of the core visual object recognition behavior of humans, monkeys, and state-of-the-art deep artificial neural networks. *Journal of Neuroscience*, 38 (33):7255–7269, 2018.

J. Brendan Ritchie, Stefania Bracci, and Hans Op de Beeck. Avoiding illusory effects in representational similarity analysis: What (not) to do with the diagonal. *NeuroImage*, 148:197–200, 2017.

Martin Schrimpf, Jonas Kubilius, Ha Hong, Najib J Majaj, Rishi Rajalingham, Elias B Issa, Kohitij Kar, Pouya Bashivan, Jonathan Prescott-Roy, Kailyn Schmidt, et al. Brain-score: Which artificial neural network for object recognition is most brain-like? *bioRxiv*, page 407007, 2018.

Jianghong Shi, Eric Shea-Brown, and Michael Buice. Comparison against task driven artificial neural networks reveals functional properties in mouse visual cortex. *Advances in Neural Information Processing Systems*, 32:5764–5774, 2019.

Jianghong Shi, Michael A Buice, Eric Shea-Brown, Stefan Mihalas, and Bryan Tripp. A convolutional network architecture driven by mouse neuroanatomical data. *bioRxiv*, 2020.

Joshua H Siegle, Peter Ledochowitsch, Xiaoxuan Jia, Daniel Millman, Gabriel K Ocker, Shiella Caldejon, Linzy Casal, Andrew Cho, Daniel J Denman, Séverine Durand, et al. Reconciling functional differences in populations of neurons recorded with two-photon imaging and electrophysiology. *BioRxiv*, 2020.

Joshua H Siegle, Xiaoxuan Jia, Séverine Durand, Sam Gale, Corbett Bennett, Nile Graddis, Greggory Heller, Tamina K Ramirez, Hannah Choi, Jennifer A Luviano, et al. Survey of spiking in the mouse visual system reveals functional hierarchy. *Nature*, pages 1–7, 2021.

Nicholas A Steinmetz, Cagatay Aydin, Anna Lebedeva, Michael Okun, Marius Pachitariu, Marius Bauza, Maxime Beau, Jai Bhagat, Claudia Böhm, Martijn Broux, et al. Neuropixels 2.0: A miniaturized high-density probe for stable, long-term brain recordings. *Science*, 372(6539), 2021.

Quanxin Wang and Andreas Burkhalter. Stream-related preferences of inputs to the superior colliculus from areas of dorsal and ventral streams of mouse visual cortex. *Journal of Neuroscience*, 33(4):1696–1705, 2013.

Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018.

Daniel LK Yamins and James J DiCarlo. Using goal-driven deep learning models to understand sensory cortex. *Nature Neuroscience*, 19(3):356–365, 2016.

Daniel LK Yamins, Ha Hong, Charles F Cadieu, Ethan A Solomon, Darren Seibert, and James J DiCarlo. Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the National Academy of Sciences*, 111(23):8619–8624, 2014.

Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*, 2017.

Yinda Zhang, Shuran Song, Ersin Yumer, Manolis Savva, Joon-Young Lee, Hailin Jin, and Thomas Funkhouser. Physically-based rendering for indoor scene understanding using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5287–5295, 2017.

Chengxu Zhuang, Jonas Kubilius, Mitra JZ Hartmann, and Daniel Yamins. Toward goal-driven neural network models for the rodent whisker-trigeminal system. *Advances in Neural Information Processing Systems*, 2017:2556–2566, 2017.

Chengxu Zhuang, Siming Yan, Aran Nayebi, Martin Schrimpf, Michael C Frank, James J DiCarlo, and Daniel LK Yamins. Unsupervised neural network models of the ventral visual stream. *Proceedings of the National Academy of Sciences*, 118(3), 2021.

Davide Zoccolan, Nadja Oertelt, James J DiCarlo, and David D Cox. A rodent model for the study of invariant visual object recognition. *Proceedings of the National Academy of Sciences*, 106(21): 8748–8753, 2009.

6 Methods

6.1 Neural Response Datasets

We used the Allen Brain Observatory Visual Coding dataset (de Vries et al., 2020; Siegle et al., 2021) collected using both two-photon calcium imaging and Neuropixels from areas VISp (V1), VISl (LM), VISal (AL), VISrl (RL), VISam (AM), and VISpm (PM) in mouse visual cortex. We focused on the natural scene stimuli, consisting of 118 images, each presented 50 times (i.e., 50 trials per image).

We list the number of units and specimens for each dataset in Table S1, after units are selected, according to the following procedure: For the calcium imaging data, we used a similar unit selection criterion as in Conwell et al. (2020), where we sub-selected units that attain a Spearman-Brown corrected split-half consistency of at least 0.3 (averaged across 100 bootstrapped trials), and whose peak responses to their preferred images are not significantly modulated by the mouse's running speed during stimulus presentation ($p > 0.05$).

For the Neuropixels dataset, we separately averaged, for each specimen and each visual area, the temporal response (at the level of 10-ms bins up to 250 ms) on the largest contiguous time interval when the median (across the population of units in that specimen) split-half consistency reached at least 0.3. This procedure helps to select the most internally-consistent units in their temporally-averaged response, and accounts for the fact that different specimens have different time courses along which their population response becomes reliable.

Finally, after subselecting units according to the above criteria for both datasets, we only keep specimens that have at least the 75th percentile number of units among all specimens for that given visual area. This final step helped to ensure we have enough internally-consistent units per specimen for the inter-animal consistency estimation (derived in Section 6.3).

Dataset Type	Visual Area	Total Units	Total Specimens
Calcium Imaging	VISp	7080	29
	VISl	4393	24
	VISal	2064	9
	VISrl	1116	8
	VISam	847	9
	VISpm	1844	19
Neuropixels	VISp	442	8
	VISl	162	6
	VISal	396	6
	VISrl	299	7
	VISam	257	7
	VISpm	175	5

Table S1: **Descriptive statistics of the neural datasets.** Total number of units and specimens for each visual area for the calcium imaging and Neuropixels datasets.

6.2 Noise Corrected Neural Predictivity

6.2.1 Linear Regression

When we perform neural fits, we choose a random 50% set of natural scene images (59 images in total) to train the regression, and the remaining 50% to use as a test set (59 images in total), across ten train-test splits total. For Ridge, Lasso, and ElasticNet regression, we use an $\alpha = 1$, following the `sklearn.linear_model` convention. For ElasticNet, we use an `l1_ratio=0.5`. PLS regression was performed with 25 components, as in prior work (e.g., Yamins et al., 2014; Schrimpf et al., 2018). When we perform regression with the One-to-One mapping, as in Figure 1B, we identify the top correlated (via Pearson correlation on the training images) unit in the source population for each target unit. Once that source unit has been identified, we then fix it for that particular train-test split, evaluated on the remaining 50% of images.

Motivated by the justification given in Section 6.3 for the noise correction in the inter-animal consistency, the noise correction of the model to neural response regression is a special case of the quantity defined in Section 6.3.2, where now the source animal is replaced by model features, separately fit to each target animal (from the set of available animals $\mathcal{A}$). Let L be the set of model layers, let r^ℓ be the set of model responses at model layer $\ell \in L$, M be the mapping, and let s be the trial-averaged pseudo-population response.

$$\max_{\ell \in L} \operatorname{median} \bigoplus_{\mathsf{B} \in \mathcal{A}} \left\langle \frac{\mathsf{Corr}\left(M\left(r^\ell_{\mathsf{train}}; \mathrm{s}^{\mathsf{B}}_{1,\mathsf{train}}\right)_{\mathsf{test}}, \mathrm{s}^{\mathsf{B}}_{2,\mathsf{test}}\right)}{\sqrt{\widetilde{\mathsf{Corr}}\left(M\left(r^\ell_{\mathsf{train}}; \mathrm{s}^{\mathsf{B}}_{1,\mathsf{train}}\right)_{\mathsf{test}}, M\left(r^\ell_{\mathsf{train}}; \mathrm{s}^{\mathsf{B}}_{2,\mathsf{train}}\right)_{\mathsf{test}}\right) \times \widetilde{\mathsf{Corr}}\left(\mathrm{s}^{\mathsf{B}}_{1,\mathsf{test}}, \mathrm{s}^{\mathsf{B}}_{2,\mathsf{test}}\right)}} \right\rangle,$$

where the average is taken over 100 bootstrapped split-half trials, $\oplus$ denotes concatenation of units across animals $\mathsf{B} \in \mathcal{A}$ followed by the median value across units, and $\mathsf{Corr}(\cdot, \cdot)$ denotes the Pearson correlation of the two quantities. $\widetilde{\mathsf{Corr}}(\cdot, \cdot)$ denotes the Spearman-Brown corrected value of the original quantity (see Section 6.3.5).

Prior to obtaining the model features of the stimuli for linear regression, we preprocessed each stimulus using the image transforms used on the validation set during model training, resizing the shortest edge of the stimulus in both cases to 64 pixels, preserving the aspect ratio of the input stimulus. Specifically, for models trained using the ImageNet dataset, we first resized the shortest edge of the stimulus to 256 pixels, center-cropped the image to 224×224 pixels, and finally resized the stimulus to 64×64 pixels. For models trained using the CIFAR-10 dataset, this resizing yielded a 64×81 pixels stimulus.

6.2.2 Representational Similarity Analysis (RSA)

In line with prior work (Shi et al., 2019; Conwell et al., 2020), we also used representational similarity analysis (RSA, Kriegeskorte et al., 2008) to compare models to neural responses, as well as to compare animals to each other. Specifically, we compared (via Pearson correlation) only the upper-right triangles of the representational dissimilarity matrices (RDMs), excluding the diagonals to avoid illusory effects (Ritchie et al., 2017).

For each visual area and a given model, we defined the predictivity of the model for that area to be the maximum RSA score across model layers after the suitable noise correction is applied, which is defined as follows. Let r^ℓ be the model responses at model layer ℓ and let s be the trial-averaged pseudo-population response (i.e., responses aggregated across specimens). The metric used here is a specific instance of Equation (10), where the single source animal A is the trial-wise,

deterministic model features (which have a mapping consistency of 1 as a result) and a single target animal B, which is the pseudo-population response:

$$\max_{\ell \in L} \left\langle \frac{\mathsf{RSA}\left(r^{\ell}, \mathrm{s}_2\right)}{\sqrt{\widetilde{\mathsf{RSA}}\left(\mathrm{s}_1, \mathrm{s}_2\right)}} \right\rangle, \qquad \widetilde{\mathsf{RSA}}\left(\mathrm{s}_1, \mathrm{s}_2\right) := \frac{2\,\mathsf{RSA}\left(\mathrm{s}_1, \mathrm{s}_2\right)}{1 + \mathsf{RSA}\left(\mathrm{s}_1, \mathrm{s}_2\right)}, \tag{1}$$

where L is the set of model layers, $\{\mathrm{s}_i\}_{i=1}^{2}$ are the animal's responses for two halves of the trials (and averaged across the trials dimension), the average is computed over 100 bootstrapped split-half trials, and $\widetilde{\mathsf{RSA}}\left(\mathrm{s}_1, \mathrm{s}_2\right)$ denotes Spearman-Brown correction applied to the internal consistency quantity, $\mathsf{RSA}\left(\mathrm{s}_1, \mathrm{s}_2\right)$, defined in Section 6.3.5.

If the fits are performed separately for each animal, then B corresponds to each animal among those for a given visual area (defined by the set $\mathcal{A}$), and we compute the median across animals $\mathsf{B} \in \mathcal{A}$:

$$\max_{\ell \in L} \operatorname*{median}_{\mathsf{B} \in \mathcal{A}} \left\langle \frac{\mathsf{RSA}\left(r^{\ell}, \mathrm{s}_2^{\mathsf{B}}\right)}{\sqrt{\widetilde{\mathsf{RSA}}\left(\mathrm{s}_1^{\mathsf{B}}, \mathrm{s}_2^{\mathsf{B}}\right)}} \right\rangle. \tag{2}$$

Similar to the above, Spearman-Brown correction is applied to the internal consistency quantity, $\mathsf{RSA}\left(\mathrm{s}_1^{\mathsf{B}}, \mathrm{s}_2^{\mathsf{B}}\right)$.

6.3 Inter-Animal Consistency Derivation

6.3.1 Single Animal Pair

Suppose we have neural responses from two animals A and B. Let t_i^p be the vector of true responses (either at a given time bin or averaged across a set of time bins) of animal $p \in \mathcal{A} = \{\mathsf{A}, \mathsf{B}, \dots\}$ on stimulus set $i \in \{\mathsf{train}, \mathsf{test}\}$. Of course, we only receive noisy observations of t_i^p, so let $\mathrm{s}_{j,i}^p$ be the jth set of n trials of t_i^p. Finally, let $M(x; y)_i$ be the predictions of a mapping M (e.g., PLS) when trained on input x to match output y and tested on stimulus set i. For example, $M\left(\mathrm{t}_{\mathsf{train}}^{\mathsf{A}}; \mathrm{t}_{\mathsf{train}}^{\mathsf{B}}\right)_{\mathsf{test}}$ is the prediction of mapping M on the test set stimuli trained to match the true neural responses of animal B given, as input, the true neural responses of animal A on the train set stimuli. Similarly, $M\left(\mathrm{s}_{1,\mathsf{train}}^{\mathsf{A}}; \mathrm{s}_{1,\mathsf{train}}^{\mathsf{B}}\right)_{\mathsf{test}}$ is the prediction of mapping M on the test set stimuli trained to match the trial-average of noisy sample 1 on the train set stimuli of animal B given, as input, the trial-average of noisy sample 1 on the train set stimuli of animal A.

With these definitions in hand, the inter-animal mapping consistency from animal A to animal B corresponds to the following true quantity to be estimated:

$$\mathsf{Corr}\left(M\left(\mathrm{t}_{\mathsf{train}}^{\mathsf{A}}; \mathrm{t}_{\mathsf{train}}^{\mathsf{B}}\right)_{\mathsf{test}}, \mathrm{t}_{\mathsf{test}}^{\mathsf{B}}\right), \tag{3}$$

where $\mathsf{Corr}(\cdot, \cdot)$ is the Pearson correlation across a stimulus set. In what follows, we will argue that Equation (3) can be approximated with the following ratio of measurable quantities, where we split in half and average the noisy trial observations, indexed by 1 and by 2:

$$\mathsf{Corr}\left(M\left(\mathrm{t}_{\mathsf{train}}^{\mathsf{A}}; \mathrm{t}_{\mathsf{train}}^{\mathsf{B}}\right)_{\mathsf{test}}, \mathrm{t}_{\mathsf{test}}^{\mathsf{B}}\right) \sim \frac{\mathsf{Corr}\left(M\left(\mathrm{s}_{1,\mathsf{train}}^{\mathsf{A}}; \mathrm{s}_{1,\mathsf{train}}^{\mathsf{B}}\right)_{\mathsf{test}}, \mathrm{s}_{2,\mathsf{test}}^{\mathsf{B}}\right)}{\sqrt{\mathsf{Corr}\left(M\left(\mathrm{s}_{1,\mathsf{train}}^{\mathsf{A}}; \mathrm{s}_{1,\mathsf{train}}^{\mathsf{B}}\right)_{\mathsf{test}}, M\left(\mathrm{s}_{2,\mathsf{train}}^{\mathsf{A}}; \mathrm{s}_{2,\mathsf{train}}^{\mathsf{B}}\right)_{\mathsf{test}}\right) \times \mathsf{Corr}\left(\mathrm{s}_{1,\mathsf{test}}^{\mathsf{B}}, \mathrm{s}_{2,\mathsf{test}}^{\mathsf{B}}\right)}}. \tag{4}$$

In words, the inter-animal consistency (i.e., the quantity on the left side of Equation (4)) corresponds to the predictivity of the mapping on the test set stimuli from animal A to animal B on two different (averaged) halves of noisy trials (i.e., the numerator on the right side of Equation (4)), corrected by the square root of the mapping reliability on animal A's responses to the test set stimuli on two different halves of noisy trials multiplied by the internal consistency of animal B.

We justify the approximation in Equation (4) by gradually replacing the true quantities (t) by their measurable estimates (s), starting from the original quantity in Equation (3). First, we make the approximation that:

$$\mathrm{Corr}\left(M\left(\mathrm{t}^{\mathsf{A}}_{\mathsf{train}};\mathrm{t}^{\mathsf{B}}_{\mathsf{train}}\right)_{\mathsf{test}},\mathrm{s}^{\mathsf{B}}_{2,\mathsf{test}}\right)\sim\mathrm{Corr}\left(M\left(\mathrm{t}^{\mathsf{A}}_{\mathsf{train}};\mathrm{t}^{\mathsf{B}}_{\mathsf{train}}\right)_{\mathsf{test}},\mathrm{t}^{\mathsf{B}}_{\mathsf{test}}\right)\times\mathrm{Corr}\left(\mathrm{t}^{\mathsf{B}}_{\mathsf{test}},\mathrm{s}^{\mathsf{B}}_{2,\mathsf{test}}\right),\tag{5}$$

by the transitivity of positive correlations (which is a reasonable assumption when the number of stimuli is large). Next, by transitivity and normality assumptions in the structure of the noisy estimates and since the number of trials (n) between the two sets is the same, we have that:

$$\begin{aligned}\mathrm{Corr}\left(\mathrm{s}^{\mathsf{B}}_{1,\mathsf{test}},\mathrm{s}^{\mathsf{B}}_{2,\mathsf{test}}\right)&\sim\mathrm{Corr}\left(\mathrm{s}^{\mathsf{B}}_{1,\mathsf{test}},\mathrm{t}^{\mathsf{B}}_{\mathsf{test}}\right)\times\mathrm{Corr}\left(\mathrm{t}^{\mathsf{B}}_{\mathsf{test}},\mathrm{s}^{\mathsf{B}}_{2,\mathsf{test}}\right)\\&\sim\mathrm{Corr}\left(\mathrm{t}^{\mathsf{B}}_{\mathsf{test}},\mathrm{s}^{\mathsf{B}}_{2,\mathsf{test}}\right)^2.\end{aligned}\tag{6}$$

In words, Equation (6) states that the correlation between the average of two sets of noisy observations of n trials each is approximately the square of the correlation between the true value and average of one set of n noisy trials. Therefore, combining Equations (5) and (6), it follows that:

$$\mathrm{Corr}\left(M\left(\mathrm{t}^{\mathsf{A}}_{\mathsf{train}};\mathrm{t}^{\mathsf{B}}_{\mathsf{train}}\right)_{\mathsf{test}},\mathrm{t}^{\mathsf{B}}_{\mathsf{test}}\right)\sim\frac{\mathrm{Corr}\left(M\left(\mathrm{t}^{\mathsf{A}}_{\mathsf{train}};\mathrm{t}^{\mathsf{B}}_{\mathsf{train}}\right)_{\mathsf{test}},\mathrm{s}^{\mathsf{B}}_{2,\mathsf{test}}\right)}{\sqrt{\mathrm{Corr}\left(\mathrm{s}^{\mathsf{B}}_{1,\mathsf{test}},\mathrm{s}^{\mathsf{B}}_{2,\mathsf{test}}\right)}}.\tag{7}$$

From the right side of Equation (7), we can see that we have removed $\mathrm{t}^{\mathsf{B}}_{\mathsf{test}}$, but we still need to remove the $M\left(\mathrm{t}^{\mathsf{A}}_{\mathsf{train}};\mathrm{t}^{\mathsf{B}}_{\mathsf{train}}\right)_{\mathsf{test}}$ term, as this term still contains unmeasurable (i.e., true) quantities. We apply the same two steps, described above, by analogy, though these approximations may not always be true (they are, however, true for Gaussian noise):

$$\begin{aligned}\mathrm{Corr}\left(M\left(\mathrm{s}^{\mathsf{A}}_{1,\mathsf{train}};\mathrm{s}^{\mathsf{B}}_{1,\mathsf{train}}\right)_{\mathsf{test}},\mathrm{s}^{\mathsf{B}}_{2,\mathsf{test}}\right)&\sim\mathrm{Corr}\left(\mathrm{s}^{\mathsf{B}}_{2,\mathsf{test}},M\left(\mathrm{t}^{\mathsf{A}}_{\mathsf{train}};\mathrm{t}^{\mathsf{B}}_{\mathsf{train}}\right)_{\mathsf{test}}\right)\\&\quad\times\mathrm{Corr}\left(M\left(\mathrm{t}^{\mathsf{A}}_{\mathsf{train}};\mathrm{t}^{\mathsf{B}}_{\mathsf{train}}\right)_{\mathsf{test}},M\left(\mathrm{s}^{\mathsf{A}}_{1,\mathsf{train}};\mathrm{s}^{\mathsf{B}}_{1,\mathsf{train}}\right)_{\mathsf{test}}\right)\end{aligned}$$

$$\begin{aligned}&\mathrm{Corr}\left(M\left(\mathrm{s}^{\mathsf{A}}_{1,\mathsf{train}};\mathrm{s}^{\mathsf{B}}_{1,\mathsf{train}}\right)_{\mathsf{test}},M\left(\mathrm{s}^{\mathsf{A}}_{2,\mathsf{train}};\mathrm{s}^{\mathsf{B}}_{2,\mathsf{train}}\right)_{\mathsf{test}}\right)\\&\sim\mathrm{Corr}\left(M\left(\mathrm{s}^{\mathsf{A}}_{1,\mathsf{train}};\mathrm{s}^{\mathsf{B}}_{1,\mathsf{train}}\right)_{\mathsf{test}},M\left(\mathrm{t}^{\mathsf{A}}_{\mathsf{train}};\mathrm{t}^{\mathsf{B}}_{\mathsf{train}}\right)_{\mathsf{test}}\right)^2,\end{aligned}$$

which taken together implies the following:

$$\mathrm{Corr}\left(M\left(\mathrm{t}^{\mathsf{A}}_{\mathsf{train}};\mathrm{t}^{\mathsf{B}}_{\mathsf{train}}\right)_{\mathsf{test}},\mathrm{s}^{\mathsf{B}}_{2,\mathsf{test}}\right)\sim\frac{\mathrm{Corr}\left(M\left(\mathrm{s}^{\mathsf{A}}_{1,\mathsf{train}};\mathrm{s}^{\mathsf{B}}_{1,\mathsf{train}}\right)_{\mathsf{test}},\mathrm{s}^{\mathsf{B}}_{2,\mathsf{test}}\right)}{\sqrt{\mathrm{Corr}\left(M\left(\mathrm{s}^{\mathsf{A}}_{1,\mathsf{train}};\mathrm{s}^{\mathsf{B}}_{1,\mathsf{train}}\right)_{\mathsf{test}},M\left(\mathrm{s}^{\mathsf{A}}_{2,\mathsf{train}};\mathrm{s}^{\mathsf{B}}_{2,\mathsf{train}}\right)_{\mathsf{test}}\right)}}.\tag{8}$$

Equations (7) and (8) together imply the final estimated quantity given in Equation (4).

6.3.2 Multiple Animals

For multiple animals, we consider the average of the true quantity for each target in B in Equation (3) across source animals A in the ordered pair (A, B) of animals A and B:

$$\left\langle \mathsf{Corr}\left(M\left(\mathrm{t}^{\mathsf{A}}_{\mathsf{train}};\mathrm{t}^{\mathsf{B}}_{\mathsf{train}}\right)_{\mathsf{test}},\mathrm{t}^{\mathsf{B}}_{\mathsf{test}}\right)\right\rangle_{\mathsf{A}\in\mathcal{A}:(\mathsf{A},\mathsf{B})\in\mathcal{A}\times\mathcal{A}}$$
$$\sim\left\langle\frac{\mathsf{Corr}\left(M\left(\mathrm{s}^{\mathsf{A}}_{1,\mathsf{train}};\mathrm{s}^{\mathsf{B}}_{1,\mathsf{train}}\right)_{\mathsf{test}},\mathrm{s}^{\mathsf{B}}_{2,\mathsf{test}}\right)}{\sqrt{\widetilde{\mathsf{Corr}}\left(M\left(\mathrm{s}^{\mathsf{A}}_{1,\mathsf{train}};\mathrm{s}^{\mathsf{B}}_{1,\mathsf{train}}\right)_{\mathsf{test}},M\left(\mathrm{s}^{\mathsf{A}}_{2,\mathsf{train}};\mathrm{s}^{\mathsf{B}}_{2,\mathsf{train}}\right)_{\mathsf{test}}\right)\times\widetilde{\mathsf{Corr}}\left(\mathrm{s}^{\mathsf{B}}_{1,\mathsf{test}},\mathrm{s}^{\mathsf{B}}_{2,\mathsf{test}}\right)}}\right\rangle_{\mathsf{A}\in\mathcal{A}:(\mathsf{A},\mathsf{B})\in\mathcal{A}\times\mathcal{A}}.$$

We also bootstrap across trials, and have multiple train/test splits, in which case the average on the right hand side of the equation includes averages across these as well.

Note that each neuron in our analysis will have this single average value associated with it when *it* was a target animal (B), averaged over source animals/subsampled source neurons, bootstrapped trials, and train/test splits. This yields a vector of these average values, which we can take median and standard error of the mean (s.e.m.) over, as we do with standard explained variance metrics.

6.3.3 RSA

We can extend the above derivations to other commonly used metrics for comparing representations that involve correlation. Since $\mathsf{RSA}(x,y) := \mathsf{Corr}(\mathsf{RDM}(x),\mathsf{RDM}(y))$, then the corresponding quantity in Equation (4) analogously (by transitivity of positive correlations) becomes:

$$\left\langle \mathsf{RSA}\left(M\left(\mathrm{t}^{\mathsf{A}}_{\mathsf{train}};\mathrm{t}^{\mathsf{B}}_{\mathsf{train}}\right)_{\mathsf{test}},\mathrm{t}^{\mathsf{B}}_{\mathsf{test}}\right)\right\rangle_{\mathsf{A}\in\mathcal{A}:(\mathsf{A},\mathsf{B})\in\mathcal{A}\times\mathcal{A}}$$
$$\sim\left\langle\frac{\mathsf{RSA}\left(M\left(\mathrm{s}^{\mathsf{A}}_{1,\mathsf{train}};\mathrm{s}^{\mathsf{B}}_{1,\mathsf{train}}\right)_{\mathsf{test}},\mathrm{s}^{\mathsf{B}}_{2,\mathsf{test}}\right)}{\sqrt{\widetilde{\mathsf{RSA}}\left(M\left(\mathrm{s}^{\mathsf{A}}_{1,\mathsf{train}};\mathrm{s}^{\mathsf{B}}_{1,\mathsf{train}}\right)_{\mathsf{test}},M\left(\mathrm{s}^{\mathsf{A}}_{2,\mathsf{train}};\mathrm{s}^{\mathsf{B}}_{2,\mathsf{train}}\right)_{\mathsf{test}}\right)\times\widetilde{\mathsf{RSA}}\left(\mathrm{s}^{\mathsf{B}}_{1,\mathsf{test}},\mathrm{s}^{\mathsf{B}}_{2,\mathsf{test}}\right)}}\right\rangle_{\mathsf{A}\in\mathcal{A}:(\mathsf{A},\mathsf{B})\in\mathcal{A}\times\mathcal{A}}. \tag{9}$$

Note that in this case, each *animal* (rather than neuron) in our analysis will have this single average value associated with it when *it* was a target animal (B) (since RSA is computed over images and neurons), where the average is over source animals/subsampled source neurons, bootstrapped trials, and train/test splits. This yields a vector of these average values, which we can take median and s.e.m. over, across animals $\mathsf{B}\in\mathcal{A}$.

For RSA, we can use the identity mapping (since RSA is computed over neurons as well, the number of neurons between source and target animal can be different to compare them with the identity mapping). As parameters are not fit, we can choose train = test, so that Equation (9) becomes:

$$\left\langle \mathsf{RSA}\left(\mathrm{t}^{\mathsf{A}},\mathrm{t}^{\mathsf{B}}\right)\right\rangle_{\mathsf{A}\in\mathcal{A}:(\mathsf{A},\mathsf{B})\in\mathcal{A}\times\mathcal{A}}\sim\left\langle\frac{\mathsf{RSA}\left(\mathrm{s}^{\mathsf{A}}_{1},\mathrm{s}^{\mathsf{B}}_{2}\right)}{\sqrt{\widetilde{\mathsf{RSA}}\left(\mathrm{s}^{\mathsf{A}}_{1},\mathrm{s}^{\mathsf{A}}_{2}\right)\times\widetilde{\mathsf{RSA}}\left(\mathrm{s}^{\mathsf{B}}_{1},\mathrm{s}^{\mathsf{B}}_{2}\right)}}\right\rangle_{\mathsf{A}\in\mathcal{A}:(\mathsf{A},\mathsf{B})\in\mathcal{A}\times\mathcal{A}}. \tag{10}$$

6.3.4 Pooled Source Animal

Often times, we may not have enough neurons per animal to ensure that the estimated inter-animal consistency in our data closely matches the "true" inter-animal consistency. In order to address this

issue, we holdout one animal at a time and compare it to the pseudo-population aggregated across units from the remaining animals, as opposed to computing the consistencies in a pairwise fashion. Thus, B is still the target heldout animal as in the pairwise case, but now the average over A is over a sole "pooled" source animal constructed from the pseudo-population of the remaining animals.

6.3.5 Spearman-Brown Correction

The Spearman-Brown correction can be applied to each of the terms in the denominator individually, as they are each correlations of observations from half the trials of the *same* underlying process to itself (unlike the numerator). Namely,

$$\widetilde{\mathsf{Corr}}\,(X, Y) := \frac{2\,\mathsf{Corr}\,(X, Y)}{1 + \mathsf{Corr}\,(X, Y)}.$$

Analogously, since $\mathsf{RSA}(X, Y) := \mathsf{Corr}(\mathsf{RDM}(x), \mathsf{RDM}(y))$, then we define

$$\begin{aligned}\widetilde{\mathsf{RSA}}\,(X, Y) &:= \widetilde{\mathsf{Corr}}(\mathsf{RDM}(x), \mathsf{RDM}(y)) \\ &= \frac{2\,\mathsf{RSA}\,(X, Y)}{1 + \mathsf{RSA}\,(X, Y)}.\end{aligned}$$

6.4 StreamNet Architecture Variants

We developed shallow, multiple-streamed architectures for mouse visual cortex, shown in Figure 5A. There are three main modules in our architecture: shallow, intermediate, and deep. The shallow and deep modules each consist of one convolutional layer and the intermediate module consists of a block of two convolutional layers. Thus, the longest length of the computational graph, excluding the readout module, is four (i.e., $1+2+1$). Depending on the number of parallel streams in the model, the intermediate module would contain multiple branches (in parallel), each receiving input from the shallow module. The outputs of the intermediate modules are then passed through one convolutional operation (deep module). Finally, the outputs of each parallel branch would be summed together, concatenated across the channels dimension, and used as input for the readout module. Table S2 describes the parameters of three model variants, each containing one ($N = 1$), two ($N = 2$), or six ($N = 6$) parallel branches.

6.5 Neural Network Training Objectives

In this section, we briefly describe the supervised and unsupervised objectives that were used to train our models.

6.5.1 Supervised Training Objective

The loss function $\mathcal{L}$ used in supervised training is the cross-entropy loss, defined as follows:

$$\mathcal{L}(\boldsymbol{X}; \boldsymbol{\theta}) = -\frac{1}{N}\sum_{i=1}^{N} \log\left(\frac{\exp(\boldsymbol{X}_i[\boldsymbol{c}_i])}{\sum_{j=0}^{C-1}\exp(\boldsymbol{X}_i[j])}\right), \tag{11}$$

where N is the batch size, C is the number of categories for the dataset, $\boldsymbol{X} \in \mathbb{R}^{N\times C}$ are the model outputs (i.e., logits) for the N images, $\boldsymbol{X}_i \in \mathbb{R}^C$ are the logits for the ith image, $\boldsymbol{c}_i \in [0, C-1]$ is the category index of the ith image (zero-indexed), and $\boldsymbol{\theta}$ are the model parameters. Equation (11) was minimized using stochastic gradient descent (SGD) with momentum (Bottou, 2010).

Module Name	**Output Size**	**Single** $(N = 1)$	**Dual** $(N = 2)$	**Six** $(N = 6)$
Input	64×64	N/A	N/A	N/A
Shallow	7×7	$(64, 11, 4, 2)$	$(64, 11, 4, 2)$	$(64, 11, 4, 2)$
Intermediate	3×3	$\begin{bmatrix}(192, 5, 1, 2)\\(384, 3, 1, 1)\end{bmatrix}$	$\begin{bmatrix}(192, 5, 1, 2)\\(384, 3, 1, 1)\end{bmatrix} \times 2$	$\begin{bmatrix}(192, 5, 1, 2)\\(384, 3, 1, 1)\end{bmatrix} \times 6$
Deep	3×3	If inputs are from intermediate: $(256, 3, 1, 1)$, otherwise: $(256, 3, 2, 0)$	If inputs are from intermediate: $(256, 3, 1, 1)$, otherwise: $(256, 3, 2, 0)$	If inputs are from intermediate: $(256, 3, 1, 1)$, otherwise: $(256, 3, 2, 0)$

Table S2: **Neural network parameters and output sizes for the convolutional layers of our StreamNet model variants containing one, two, and six parallel branches in the intermediate module.** One convolutional layer is denoted by a tuple: (number of filters, filter size, stride, padding). A block of convolutional layers is denoted by a list of tuples, where each tuple in the list corresponds to a single convolutional layer. When a list of tuples is followed by "$\times N$", this means that the convolutional parameters for each of the N parallel branches are the same.

ImageNet (Deng et al., 2009) This dataset contains approximately 1.3 million images in the train set and $50\,000$ images in the validation set. Each image was previously labeled into $C = 1000$ distinct categories.

CIFAR-10 (Krizhevsky et al., 2009) This dataset contains $50\,000$ images in the train set and $10\,000$ images in the validation set. Each image was previously labeled into $C = 10$ distinct categories.

6.5.2 Unsupervised Training Objectives

Sparse Autoencoder (Olshausen and Field, 1996) The goal of this objective is to reconstruct an image from a sparse image embedding. In order to generate an image reconstruction, we used a mirrored version of each of our StreamNet variants. Concretely, the loss function was defined as follows:

$$\mathcal{L}(\boldsymbol{x}; \boldsymbol{\theta}) = \frac{1}{2 \cdot 64^2} \|f(\boldsymbol{x}) - \boldsymbol{x}\|_2^2 + \frac{\lambda}{128} \|\boldsymbol{v}\|_1, \tag{12}$$

where $\boldsymbol{v} \in \mathbb{R}^{128}$ is the image embedding, f is the (mirrored) model, $f(\boldsymbol{x})$ is the image reconstruction, $\boldsymbol{x}$ is a 64×64 pixels image, λ is the regularization coefficient, and $\boldsymbol{\theta}$ are the model parameters.

Our single-, dual-, and six-stream variants were trained using a batch size of 256 for 100 epochs using SGD with momentum of 0.9 and weight decay of 0.0005. The initial learning rate was set to 0.01 for the single- and dual-stream variants and was set to 0.001 for the six-stream variant. The learning rate was decayed by a factor of 10 at epochs 30, 60, and 90. For all the StreamNet variants, the embedding dimension was set to 128 and the regularization coefficient was set to 0.0005.

Depth Prediction (Zhang et al., 2017) The goal of this objective is to predict the depth map of an image. We used a synthetically generated dataset of images known as PBRNet (Zhang et al., 2017). It contains approximately $500\,000$ images and their associated depth maps. Similar to the loss function used in the sparse autoencoder objective, we used a mean-squared loss to train the

models. The output (i.e., depth map prediction) was generated using a mirrored version of each of our StreamNet variants. In order to generate the depth map, we appended one final convolutional layer onto the output of the mirrored architecture in order to downsample the three image channels to one image channel. During training, random crops of size 224×224 pixels were applied to the image and depth map (which were both subsequently resized to 64×64 pixels). In addition, both the image and depth map were flipped horizontally with probability 0.5. Finally, prior to the application of the loss function, each depth map was normalized such that the mean and standard deviation across pixels were zero and one respectively.

Each of our single-, dual-, and six-stream variants were trained using a batch size of 256 for 50 epochs using SGD with momentum of 0.9, and weight decay of 0.0001. The initial learning rate was set to 10^{-4} and was decayed by a factor of 10 at epochs 15, 30, and 45.

RotNet (Gidaris et al., 2018) The goal of this objective is to predict the rotation of an image. Each image of the ImageNet dataset was rotated four ways ($0°$, $90°$, $180°$, $270°$) and the four rotation angles were used as "pseudo-labels" or "categories". The cross-entropy loss was used with these pseudo-labels as the training objective (i.e., Equation (11) with $C = 4$).

Our single-, dual-, and six-stream variants were trained using a batch size of 192 (which is effectively a batch size of $192 \times 4 = 768$ due to the four rotations for each image) for 50 epochs using SGD with nesterov momentum of 0.9, and weight decay of 0.0005. An initial learning rate of 0.01 was decayed by a factor of 10 at epochs 15, 30, and 45.

Instance Recognition (Wu et al., 2018) The goal of this objective is to be able to differentiate between embeddings of augmentations of one image from embeddings of augmentations of other images. Thus, this objective function is an instance of the class of contrastive objective functions.

A random image augmentation is first performed on each image of the ImageNet dataset (random resized cropping, random grayscale, color jitter, and random horizontal flip). Let $\boldsymbol{x}$ be an image augmentation, and $f(\cdot)$ be the model backbone composed with a one-layer linear multi-layer perceptron (MLP) of size 128. The image is then embedded onto a 128-dimensional unit-sphere as follows:

$$\boldsymbol{z} = f(\boldsymbol{x})/\|f(\boldsymbol{x})\|_2, \qquad \boldsymbol{z} \in \mathbb{R}^{128}.$$

Throughout model training, a memory bank containing embeddings for each image in the train set is maintained (i.e., the size of the memory bank is the same as the size of the train set). The embedding $\boldsymbol{z}$ will be "compared" to a subsample of these embeddings. Concretely, the loss function $\mathcal{L}$ for one image $\boldsymbol{x}$ is defined as follows:

$$\begin{aligned} h(\boldsymbol{u}) &= \frac{\exp(\boldsymbol{u} \cdot \boldsymbol{z}/\tau)/Z}{\exp(\boldsymbol{u} \cdot \boldsymbol{z}/\tau)/Z + (m/N)}, \\ \mathcal{L}(\boldsymbol{x}; \boldsymbol{\theta}) &= -\log h(\boldsymbol{v}) - \sum_{j=1}^{m} \log\left(1 - h(\boldsymbol{v}_j)\right), \end{aligned} \tag{13}$$

where $\boldsymbol{v} \in \mathbb{R}^{128}$ is the embedding for image $\boldsymbol{x}$ that is currently stored in the memory bank, N is the size of the memory bank, $m = 4096$ is the number of "negative" samples used, $\{\boldsymbol{v}_j\}_{j=1}^{m}$ are the negative embeddings sampled from the memory bank uniformly, Z is some normalization constant, $\tau = 0.07$ is a temperature hyperparameter, and $\boldsymbol{\theta}$ are the parameters of f. From Equation (13), we see that we want to maximize $h(\boldsymbol{v})$, which corresponds to maximizing the similarity between $\boldsymbol{v}$ and $\boldsymbol{z}$ (recall that $\boldsymbol{z}$ is the embedding for $\boldsymbol{x}$ obtained using f). We can also see that we want

to maximize $1 - h(\boldsymbol{v}_j)$ (or minimize $h(\boldsymbol{v}_j)$). This would correspond to minimizing the similarity between $\boldsymbol{v}_j$ and $\boldsymbol{z}$ (recall that $\boldsymbol{v}_j$ are the negative embeddings).

After each iteration of training, the embeddings for the current batch are used to update the memory bank (at their corresponding positions in the memory bank) via a momentum update. Concretely, for image $\boldsymbol{x}$, its embedding in the memory bank $\boldsymbol{v}$ is updated using its current embedding $\boldsymbol{z}$ as follows:

$$\begin{aligned}\boldsymbol{v} &\leftarrow \lambda \boldsymbol{v} + (1-\lambda)\boldsymbol{z},\\ \boldsymbol{v} &\leftarrow \boldsymbol{v}/\|\boldsymbol{v}\|_2,\end{aligned}$$

where $\lambda = 0.5$ is the momentum coefficient. The second operation on $\boldsymbol{v}$ is used to project $\boldsymbol{v}$ back onto the 128-dimensional unit sphere.

Our single-, dual-, and six-stream variants were trained using a batch size of 256 for 200 epochs using SGD with momentum of 0.9, and weight decay of 0.0005. An initial learning rate of 0.03 was decayed by a factor of 10 at epochs 120 and 160.

SimSiam (Chen and He, 2020) The goal of this objective is to maximize the similarity between the embeddings of two augmentations of the same image. Thus, SimSiam is another instance of the class of contrastive objective functions.

Two random image augmentations (e.g., random resized crop, random horizontal flip, color jitter, random grayscale, and random Gaussian blur) are first generated for each image in the ImageNet dataset. Let $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ be the two augmentations of the same image, $f(\cdot)$ be the model backbone, $g(\cdot)$ be a three-layer non-linear MLP, and $h(\cdot)$ be a two-layer non-linear MLP. The three-layer MLP has hidden dimensions of 2048, 2048, and 2048. The two-layer MLP has hidden dimensions of 512 and 2048 respectively. Let $\boldsymbol{\theta}$ be the parameters for f, g, and h. The loss function $\mathcal{L}$ for one image $\boldsymbol{x}$ of a batch is defined as follows (recall that $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are two augmentations of one image):

$$\begin{gathered}\boldsymbol{p}_1 = h \circ g \circ f(\boldsymbol{x}_1), \qquad \boldsymbol{p}_2 = h \circ g \circ f(\boldsymbol{x}_2), \qquad \boldsymbol{z}_1 = g \circ f(\boldsymbol{x}_1), \qquad \boldsymbol{z}_2 = g \circ f(\boldsymbol{x}_2),\\ \mathcal{L}(\boldsymbol{x}_1, \boldsymbol{x}_2; \boldsymbol{\theta}) = -\frac{1}{2}\left(\frac{\boldsymbol{z}_1 \cdot \boldsymbol{p}_2}{\|\boldsymbol{z}_1\|_2 \|\boldsymbol{p}_2\|_2} + \frac{\boldsymbol{z}_2 \cdot \boldsymbol{p}_1}{\|\boldsymbol{z}_2\|_2 \|\boldsymbol{p}_1\|_2}\right), \qquad (14)\end{gathered}$$

where $\boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{p}_1, \boldsymbol{p}_2 \in \mathbb{R}^{2048}$. Note that $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ are treated as constants in this loss function (i.e., the gradients are not back-propagated through $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$). This "stop-gradient" method was key to the success of this objective function.

Our single-, dual-, and six-stream variants were trained using a batch size of 512 for 100 epochs using SGD with momentum of 0.9, and weight decay of 0.0001. An initial learning rate of 0.1 was used, and the learning rate was decayed to 0.0 using a cosine schedule (with no warm-up).

MoCov2 (He et al., 2020; Chen et al., 2020b) The goal of this objective is to be able to distinguish augmentations of one image (i.e., by labeling them as "positive") from augmentations of other images (i.e., by labeling them as "negative"). Intuitively, embeddings of different augmentations of the same image should be more "similar" to each other than to embeddings of augmentations of other images. Thus, this algorithm is another instance of the class of contrastive objective functions and is similar conceptually to instance recognition.

Two image augmentations are first generated for each image in the ImageNet dataset by applying random resized cropping, color jitter, random grayscale, random Gaussian blur, and random horizontal flips. Let $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ be the two augmentations for one image. Let $f_q(\cdot)$ be a query encoder, which is a model backbone composed with a two-layer non-linear MLP of dimensions 2048

and 128 respectively and let $f_k(\cdot)$ be a key encoder, which has the same architecture as f_q. $\boldsymbol{x}_1$ is encoded by f_q and $\boldsymbol{x}_2$ is encoded by f_k as follows:

$$\boldsymbol{v} = f_q(\boldsymbol{x}_1), \qquad \boldsymbol{k}_0 = f_k(\boldsymbol{x}_2), \qquad \boldsymbol{v}, \boldsymbol{k}_0 \in \mathbb{R}^{128}.$$

During each iteration of training, a dictionary of size K of image embeddings obtained from previous iterations is maintained (i.e., the dimensions of the dictionary are $K \times 128$). The image embeddings in this dictionary are used as "negative" samples. The loss function $\mathcal{L}$ for one image of a batch is defined as follows:

$$\mathcal{L}(\boldsymbol{x}_1, \boldsymbol{x}_2; \boldsymbol{\theta}_q) = -\log \frac{\exp(\boldsymbol{v} \cdot \boldsymbol{k}_0/\tau)}{\sum_{i=0}^{K} \exp(\boldsymbol{v} \cdot \boldsymbol{k}_i/\tau)}, \tag{15}$$

where $\boldsymbol{\theta}_q$ are the parameters of f_q, $\tau = 0.2$ is a temperature hyperparameter, $K = 65\,536$ is the number of "negative" samples, and $\{\boldsymbol{k}_i\}_{i=1}^{K}$ are the embeddings of the negative samples (i.e., the augmentations for other images which are encoded using f_k, and are stored in the dictionary). From Equation (15), we see that we want to maximize $\boldsymbol{v} \cdot \boldsymbol{k}_0$, which corresponds to maximizing the similarity between the embeddings of the two augmentations of an image.

After each iteration of training, the dictionary of negative samples is enqueued with the embeddings from the most recent iteration, while embeddings that have been in the dictionary for the longest are dequeued. Finally, the parameters $\boldsymbol{\theta}_k$ of f_k are updated via a momentum update, as follows:

$$\boldsymbol{\theta}_k \leftarrow \lambda \boldsymbol{\theta}_k + (1 - \lambda)\boldsymbol{\theta}_q,$$

where $\lambda = 0.999$ is the momentum coefficient. Note that only $\boldsymbol{\theta}_q$ are updated with back-propagation.

Our single-, dual-, and six-stream variants were trained using a batch size of 512 for 200 epochs using SGD with momentum of 0.9, and weight decay of 0.0005. An initial learning rate of 0.06 was used, and the learning rate was decayed to 0.0 using a cosine schedule (with no warm-up).

SimCLR (Chen et al., 2020a) The goal of this objective is conceptually similar to that of MoCov2, where the embeddings of augmentations of one image should be distinguishable from the embeddings of augmentations of other images. Thus, SimCLR is another instance of the class of contrastive objective functions.

Similar to other contrastive objective functions, two image augmentations are first generated for each image in the ImageNet dataset (by using random cropping, random horizontal flips, random color jittering, random grayscaling and random Gaussian blurring). Let $f(\cdot)$ be the model backbone composed with a two-layer non-linear MLP of dimensions 2048 and 128 respectively. The two image augmentations are first embedded into a 128-dimensional space and normalized:

$$\boldsymbol{z}_1 = f(\boldsymbol{x}_1)/\|f(\boldsymbol{x}_1)\|_2, \qquad \boldsymbol{z}_2 = f(\boldsymbol{x}_2)/\|f(\boldsymbol{x}_2)\|_2, \qquad \boldsymbol{z}_1, \boldsymbol{z}_2 \in \mathbb{R}^{128}.$$

The loss function $\mathcal{L}$ for a single pair of augmentations of an image is defined as follows:

$$\mathcal{L}(\boldsymbol{x}_1, \boldsymbol{x}_2; \boldsymbol{\theta}) = -\log \frac{\exp(\boldsymbol{z}_1 \cdot \boldsymbol{z}_2/\tau)}{\sum_{i=1}^{2N} \mathbb{1}[i \neq 1] \exp(\boldsymbol{z}_1 \cdot \boldsymbol{z}_i/\tau)}, \tag{16}$$

where $\tau = 0.1$ is a temperature hyperparameter, N is the batch size, $\mathbb{1}[i \neq 1]$ is equal to 1 if $i \neq 1$ and 0 otherwise, and $\boldsymbol{\theta}$ are the parameters of f. The loss defined in Equation (16) is computed for every pair of images in the batch (including their augmentations) and subsequently averaged.

Our single-, dual-, and six-stream variants were trained using a batch size of 4096 for 200 epochs using layer-wise adaptive rate scaling (LARS, You et al., 2017) with momentum of 0.9, and weight decay of 10^{-6}. An initial learning rate of 4.8 was used and decayed to 0.0 using a cosine schedule. A linear warm-up of 10 epochs was used for the learning rate with warm-up ratio of 0.0001.

6.6 Top-1 Validation Set Performance

6.6.1 Performance of primate models on 224×224 pixels and 64×64 pixels ImageNet

Here we report the top-1 validation set accuracy of models trained in a supervised manner on 64×64 pixels and 224×224 pixels ImageNet.

Architecture	Image Size	Objective Function	Top-1 Accuracy
AlexNet	224×224	Supervised (ImageNet)	56.52%
	64×64		36.22%
VGG16	224×224		71.59%
	64×64		58.32%
ResNet-18	224×224		69.76%
	64×64		53.31%

6.6.2 Performance of StreamNet Variants on 64×64 pixels CIFAR-10 and 64×64 pixels ImageNet

Here we report the top-1 validation set accuracy of our model variants trained in a supervised manner on 64×64 pixels CIFAR-10 and ImageNet.

Architecture	Dataset	Objective Function	Top-1 Accuracy
Single Stream	CIFAR-10	Supervised	76.52%
	ImageNet		34.87%
Dual Stream	CIFAR-10		81.13%
	ImageNet		38.68%
Six Stream	CIFAR-10		78.73%
	ImageNet		34.15%

6.6.3 Transfer Performance of StreamNet Variants on 64×64 pixels ImageNet Under Linear Evaluation for Models Trained with Unsupervised Objectives

In this subsection, we report the top-1 ImageNet validation set performance under linear evaluation for models trained with unsupervised objectives. After training each model on a unsupervised objective, the model backbone weights are then held fixed and a linear readout head is trained on top of the fixed model backbone. In the case where the objective function is "untrained", model parameters were randomly initialized and held fixed while the linear readout head was trained. The image augmentations used during transfer learning were random cropping and random horizontal flipping. The linear readout for every unsupervised model was trained with the cross-entropy loss function (i.e., Equation (11) with $C = 1000$) for 100 epochs, which was minimized using SGD with momentum of 0.9, and weight decay of 10^{-9}. The initial learning rate was set to 0.1 and reduced by a factor of 10 at epochs 30, 60, and 90.

6.7 Parameter and Unit Counts for Each Model

Table S4 summarizes the total number of trainable parameters and the total number of units for each model. The number of trainable parameters reported excludes those specific to the loss function itself (i.e., the embedding or classification head). The total number of units for each model was defined as the total number of features used in the neural response fitting procedure for each model layer, summed across all model layers used for neural response fitting.

Architecture	Objective Function	ImageNet Transfer Top-1 Accuracy	Neural Predictivity Neuropixels; Calcium Imaging
Single Stream	Untrained	9.28%	32.76%; 28.65%
	Supervised	34.87%	36.21%; 29.73%
	Autoencoder	10.37%	35.99%; 28.69%
	Depth Prediction	18.04%	33.79%; 27.54%
	RotNet	19.72%	35.63%; 29.27%
	Instance Recognition	21.22%	38.01%; 30.88%
	SimSiam	26.48%	39.19%; 30.48%
	MoCov2	27.63%	39.17%; 30.30%
	SimCLR	22.84%	39.45%; 29.50%
Dual Stream	Untrained	10.85%	33.58%; 29.24%
	Supervised	38.68%	36.07%; 29.43%
	Autoencoder	10.26%	34.97%; 28.74%
	Depth Prediction	19.81%	32.81%; 27.20%
	RotNet	23.29%	35.37%; 29.15%
	Instance Recognition	22.55%	40.07%; 30.64%
	SimSiam	29.21%	40.20%; 30.60%
	MoCov2	31.00%	38.64%; 30.33%
	SimCLR	26.25%	38.03%; 29.08%
Six Stream	Untrained	11.12%	33.74%; 29.26%
	Supervised	34.15%	36.64%; 29.79%
	Autoencoder	9.27%	37.34%; 31.12%
	Depth Prediction	18.27%	33.12%; 27.63%
	RotNet	22.78%	35.49%; 28.97%
	Instance Recognition	26.49%	37.67%; 31.18%
	SimSiam	30.52%	38.17%; 30.46%
	MoCov2	32.70%	37.96%; 30.44%
	SimCLR	28.42%	38.92%; 29.19%
AlexNet	Supervised	36.22%	37.28%; 30.34%
AlexNet	Instance Recognition	16.09%	41.33%; 31.60%
ResNet-18	Supervised	53.31%	35.82%; 28.93%
ResNet-18	Instance Recognition	30.75%	38.99%; 30.11%
VGG16	Supervised	58.32%	31.92%; 27.09%
VGG16 (224 px)	Supervised	71.59%	26.03%; 20.40%
MouseNet of Shi et al. (2020)	Supervised	37.14%	31.05%; 25.89%
MouseNet Variant	Supervised	39.37%	33.02%; 26.53%

Table S3: **ImageNet top-1 validation set accuracy via linear transfer or via supervised training and neural predictivity for each model.** We summarize here the top-1 accuracy for each unsupervised and supervised model on ImageNet as well as their noise-corrected neural predictivity obtained via the PLS map (aggregated across all visual areas). These values are plotted in Figures 5C and S1. Unless otherwise stated, each model is trained and validated on 64×64 pixels images.

Architecture	Objective Function	Parameter Count	Unit Count
Single Stream	Untrained	2029632	8896
	Supervised (CIFAR-10)	2029632	11712
	Supervised (ImageNet)	2029632	8896
	Autoencoder	2029632	8896
	Depth Prediction	2029632	8896
	RotNet	2029632	8896
	Instance Recognition	2029632	8896
	SimSiam	2029632	8896
	MoCov2	2029632	8896
	SimCLR	2029632	8896
Dual Stream	Untrained	5806848	14656
	Supervised (CIFAR-10)	5806848	19392
	Supervised (ImageNet)	5806848	14656
	Autoencoder	5806848	14656
	Depth Prediction	5806848	14656
	RotNet	5806848	14656
	Instance Recognition	5806848	14656
	SimSiam	5806848	14656
	MoCov2	5806848	14656
	SimCLR	5806848	14656
Six Stream	Untrained	16780800	28480
	Supervised (CIFAR-10)	16780800	37824
	Supervised (ImageNet)	16780800	28480
	Autoencoder	16780800	28480
	Depth Prediction	16780800	28480
	RotNet	16780800	28480
	Instance Recognition	16780800	28480
	SimSiam	16780800	28480
	MoCov2	16780800	28480
	SimCLR	16780800	28480
AlexNet	Supervised (ImageNet)	57003840	19072
AlexNet (224 px)	Supervised (ImageNet)	57003840	204672
AlexNet	Instance Recognition	57022528	19072
ResNet-18	Supervised (ImageNet)	11176512	143872
ResNet-18	Instance Recognition	11176512	143872
VGG16	Supervised (ImageNet)	134260544	133120
VGG16 (224 px)	Supervised (ImageNet)	134260544	1538560
MouseNet of Shi et al. (2020)	Supervised	5974858	823296
MouseNet Variant	Supervised	5974858	823296

Table S4: **Parameter and unit counts for each model.** Each model is summarized by its total number of trainable parameters (parameter count) and the total number of features used in neural predictions (unit count), excluding those specific to the loss function itself. Unless otherwise stated, each model is trained on 64×64 pixels images.

6.8 Evaluating Model Performance on Downstream Visual Tasks

To evaluate transfer performance on downstream visual tasks, we used the activations from the outputs of the shallow, intermediate, and deep modules of our StreamNet variants. We also included the average-pooling layer in all the variants (the model layer prior to the fully-connected readout layer). The dimensionality of the activations was then reduced to 1000 dimensions using principal components analysis (PCA), if the number of features exceeded 1000. PCA was not used if the number of features was less than or equal to 1000. A linear readout on these features was then used to perform five transfer visual tasks.

For the first four object-centric visual tasks (object categorization, pose estimation, position estimation, and size estimation), we used a stimulus set that was used previously in the evaluation of neural network models of the primate visual system (Schrimpf et al., 2018; Rajalingham et al., 2018; Zhuang et al., 2021). The stimulus set consists of objects in various poses (object rotations about the x, y, and z axes), positions (vertical and horizontal coordinates of the object), and sizes, each from eight categories. We then performed five-fold cross-validation on the training split of the low variation image subset ("Var0" and "Var3", defined in Majaj et al., 2015) consisting of 3200 images, and computed the performance (metrics defined below) on the test split of the high variation set ("Var6") consisting of 1280 images. Ten different category-balanced train-test splits were randomly selected, and the performance of the best model layer (averaged across train-test splits) was reported for each model. All images were resized to 64×64 pixels prior to fitting, to account for the visual acuity adjustment. The final non-object-centric task was texture recognition, using the Describable Textures Dataset (Cimpoi et al., 2014).

Object Categorization We fit a linear support vector classifier to each model layer activations that were transformed via PCA. The regularization parameter,

$$\begin{aligned} C \in [&10^{-8}, 5\times10^{-8}, 10^{-7}, 5\times10^{-7}, 10^{-6}, 5\times10^{-6}, 10^{-5}, 5\times10^{-5},\\ &10^{-4}, 5\times10^{-4}, 10^{-3}, 5\times10^{-3}, 10^{-2}, 5\times10^{-2}, 10^{-1}, 5\times10^{-1},\\ &1, 5, 10^{2}, 5\times10^{2}, 10^{3}, 5\times10^{3}, 10^{4}, 5\times10^{4},\\ &10^{5}, 5\times10^{5}, 10^{6}, 5\times10^{6}, 10^{7}, 5\times10^{7}, 10^{8}, 5\times10^{8}], \end{aligned} \tag{17}$$

was chosen by five-fold cross validation. The categories are Animals, Boats, Cars, Chairs, Faces, Fruits, Planes, and Tables. We reported the classification accuracy average across the ten train-test splits.

Position Estimation We predicted both the vertical and the horizontal locations of the object center in the image. We used Ridge regression where the regularization parameter was selected from:

$$\alpha = 1/C, \tag{18}$$

where C was selected from the list defined in (17). For each network, we reported the correlation averaged across both locations for the best model layer.

Pose Estimation This task was similar to the position prediction task except that the prediction target were the z-axis (vertical axis) and the y-axis (horizontal axis) rotations, both of which ranged between -90 degrees and 90 degrees. The $(0, 0, 0)$ angle was defined in a per-category basis and was chosen to make the $(0, 0, 0)$ angle "semantically" consistent across different categories. We refer the reader to Hong et al. (2016) for more details. We used Ridge regression with α chosen from the range in (18).

Size Estimation The prediction target was the three-dimensional object scale, which was used to generate the image in the rendering process. This target varied between 0.625 to 1.6, which was a relative measure to a fixed canonical size of 1. When objects were at the canonical size, they occluded around 40% of the image on the longest axis. We used Ridge regression with α chosen from the range in (18).

Texture Recognition We trained linear readouts of the model layers on texture recognition using the Describable Textures Dataset (Cimpoi et al., 2014), which consists of 5640 images organized according to 47 categories, with 120 images per category. We used ten category-balanced train-test splits, provided by their benchmark. Each split consists of 3760 train-set images and 1880 test-set images. A linear support vector classifier was then fit with C chosen in the range (17). We reported the classification accuracy average across the ten train-test splits.

7 Supplemental Figures

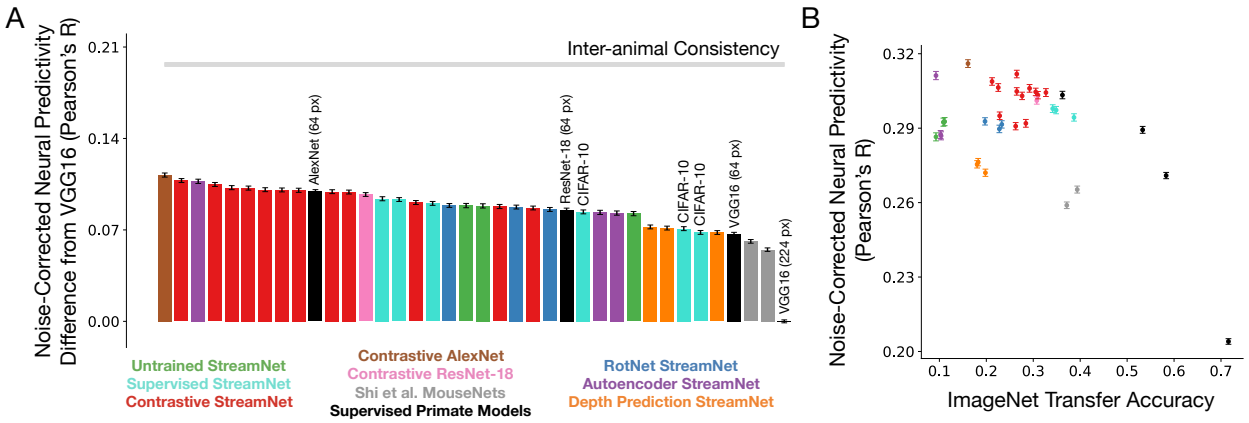

Figure S1: **Shallow architectures trained with contrastive objective functions yield the best matches to the neural data (calcium imaging dataset).** As in Figure 5, but for the calcium imaging dataset. **A**. The median and s.e.m. neural predictivity, using PLS regression, across neurons in all mouse visual areas except VISrl. $N = 16228$ units in total (VISrl is excluded, as mentioned in Section 2). Red denotes our StreamNet models trained on contrastive objective functions, blue denotes our StreamNet models trained on RotNet, turquoise denotes our StreamNet models trained in a supervised manner on ImageNet and on CIFAR-10, green denotes untrained models (random weights), orange denotes our StreamNet models trained depth prediction, purple denotes our StreamNet models trained on autoencoding, brown denotes contrastive AlexNet, pink denotes contrastive ResNet-18 (both trained on instance recognition), black denotes the remaining ImageNet supervised models (primate ventral stream models), and grey denotes the MouseNet of Shi et al. (2020) and our variant of this architecture. Actual neural predictivity performance can be found in Table S3. **B**. Each model's performance on ImageNet is plotted against its median neural predictivity across all units from each visual area. All ImageNet performance numbers can be found in Table S3. Color scheme as in **A**.

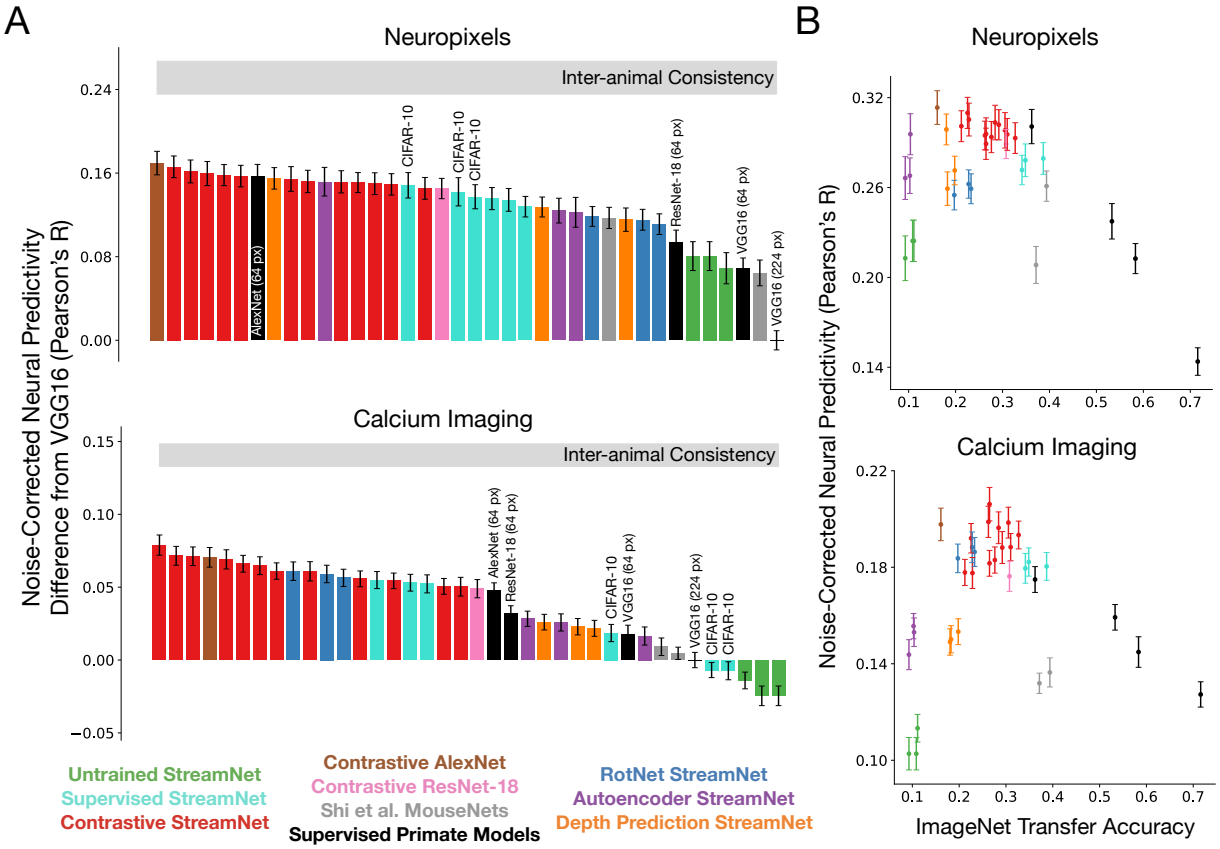

Figure S2: **Shallow architectures trained with contrastive objective functions yield the best matches to the neural data (RSA). A**. The median and s.e.m. noise-corrected neural predictivity, using RSA, across $N = 39$ and $N = 90$ animals for the Neuropixels and calcium imaging dataset respectively (across all visual areas, with VISrl excluded for the calcium imaging dataset, as mentioned in Section 2). Red denotes our StreamNet models trained on contrastive objective functions, blue denotes our StreamNet models trained on RotNet, turquoise denotes our StreamNet models trained in a supervised manner on ImageNet and on CIFAR-10, green denotes untrained models (random weights), orange denotes our StreamNet models trained depth prediction, purple denotes our StreamNet models trained on autoencoding, brown denotes contrastive AlexNet, pink denotes contrastive ResNet-18 (both trained on instance recognition), black denotes the remaining ImageNet supervised models (primate ventral stream models), and grey denotes the MouseNet of Shi et al. (2020) and our variant of this architecture. **B**. We plot each model's performance on ImageNet against its median neural predictivity, using RSA, across visual areas. All ImageNet performance numbers can be found in Table S3. Color scheme as in **A**.

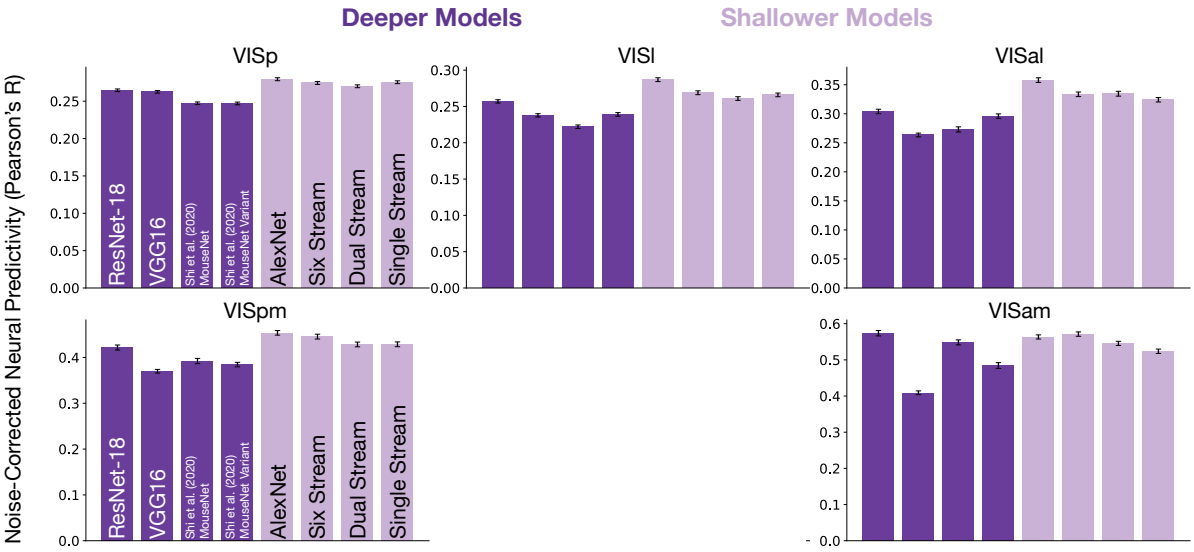

Figure S3: **Hierarchically shallow models achieve competitive neural predictivity performance (calcium imaging dataset).** As in Figure 2C, AlexNet and our StreamNet variants (light purple) were trained in a supervised manner on ImageNet and provide neural predictivity on the calcium imaging dataset that is better or at least as good as those of deeper architectures (dark purple). Refer to Table S1 for N units per visual area. As mentioned in Section 2, visual area VISrl was removed from the calcium imaging neural predictivity results.

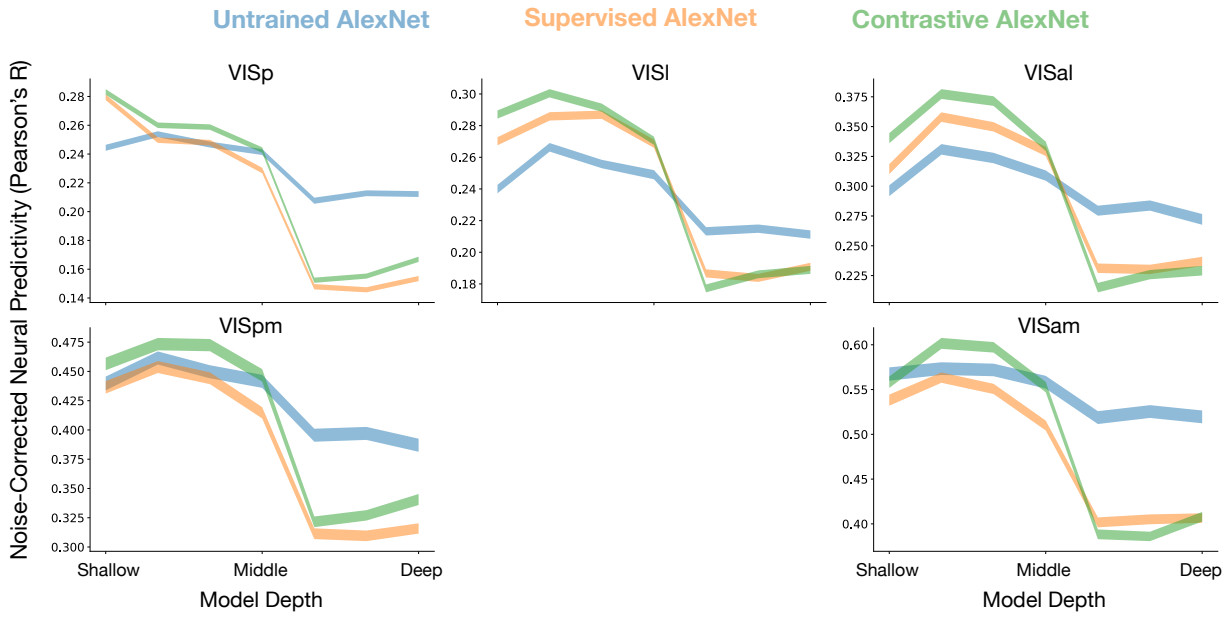

Figure S4: **Unsupervised models better predict the neural responses in mouse visual cortex (calcium imaging dataset).** As in Figure 3, AlexNet was either untrained (blue), trained in a supervised manner (orange) or trained in an unsupervised manner (green). We observe that the first four convolutional layers provide the best fits to the neural responses for all the visual areas while the latter three layers are not very predictive for any visual area. This suggests that an even shallower architecture may be suitable and further corroborates our architectural decision in Figure 2B. As mentioned in Section 2, visual area VISrl was removed from the calcium imaging neural predictivity results.

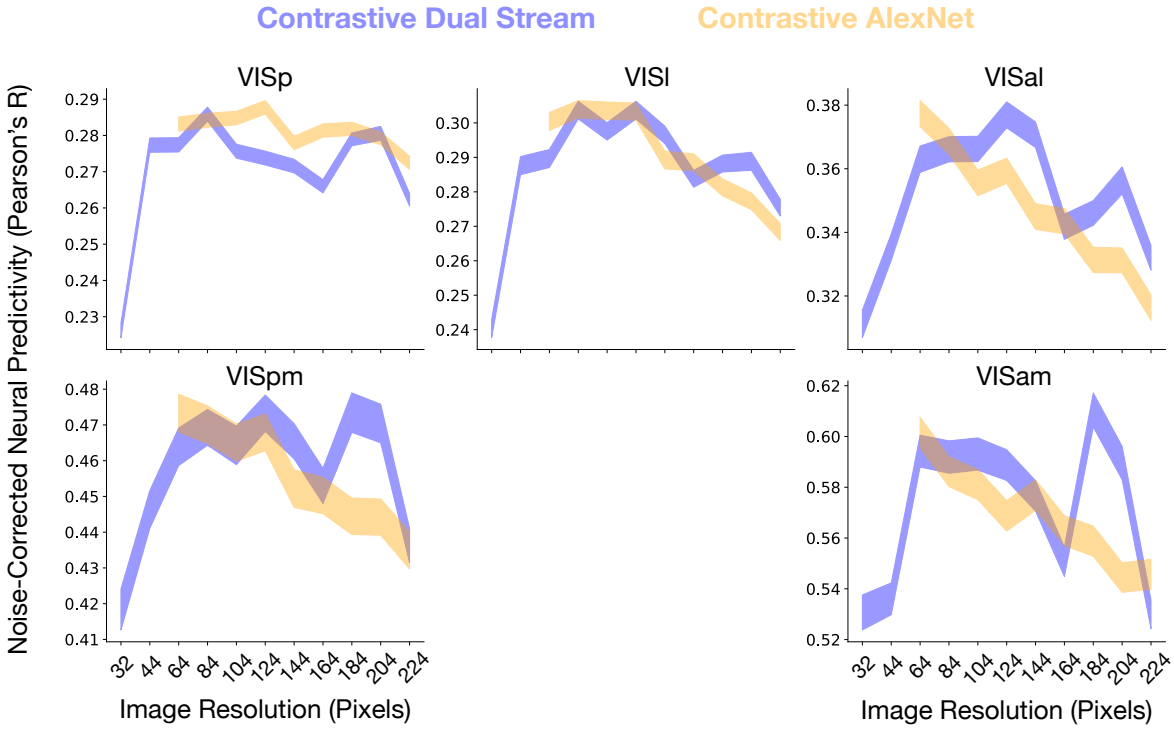

Figure S5: **Lower image resolution during model training improves task-optimized neural predictivity (calcium imaging dataset)** As in Figure 4, models with "lower visual acuity" were trained using lower resolution ImageNet images. Each image was downsampled from 224×224 pixels, the image size typically used to train primate ventral stream models, to various image sizes (image sizes on horizontal axis). Our dual stream variant (blue) and AlexNet (orange) were trained using various image sizes on instance recognition and their neural predictivity performances were computed for each mouse visual area. Training models on resolutions lower than 224×224 pixels generally led to improved correspondence with the neural responses for both models. The median and s.e.m. across neurons in each visual area is reported. As mentioned in Section 2, visual area VISrl was removed from the calcium imaging neural predictivity results. Refer to Table S1 for N units per visual area.

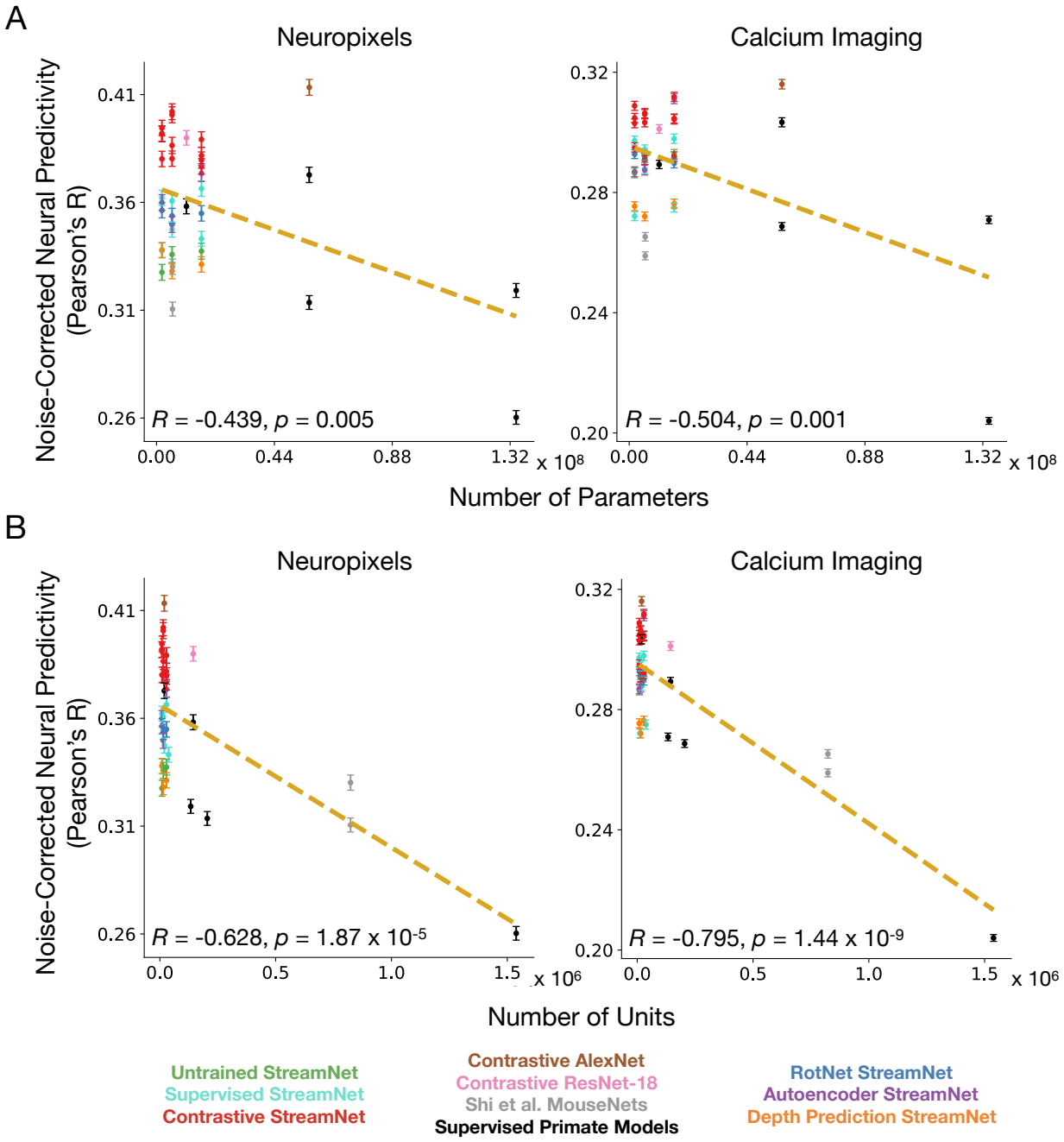

Figure S6: **Increasing neural network size can decrease the model's neural predictivity of responses in mouse visual areas.** **A**. Each model's neural predictivity is plotted as a function of its architecture size in terms of number of parameters, for both Neuropixels and calcium imaging datasets. **B**. Each model's neural predictivity is plotted as a function of its architecture size in terms of number of units, for both Neuropixels and calcium imaging datasets. The median and s.e.m. neural predictivity across neurons for each model is reported in all panels. Refer to Table S3 for the neural predictivity values of each model and to Table S4 for the parameter and unit counts of each model.

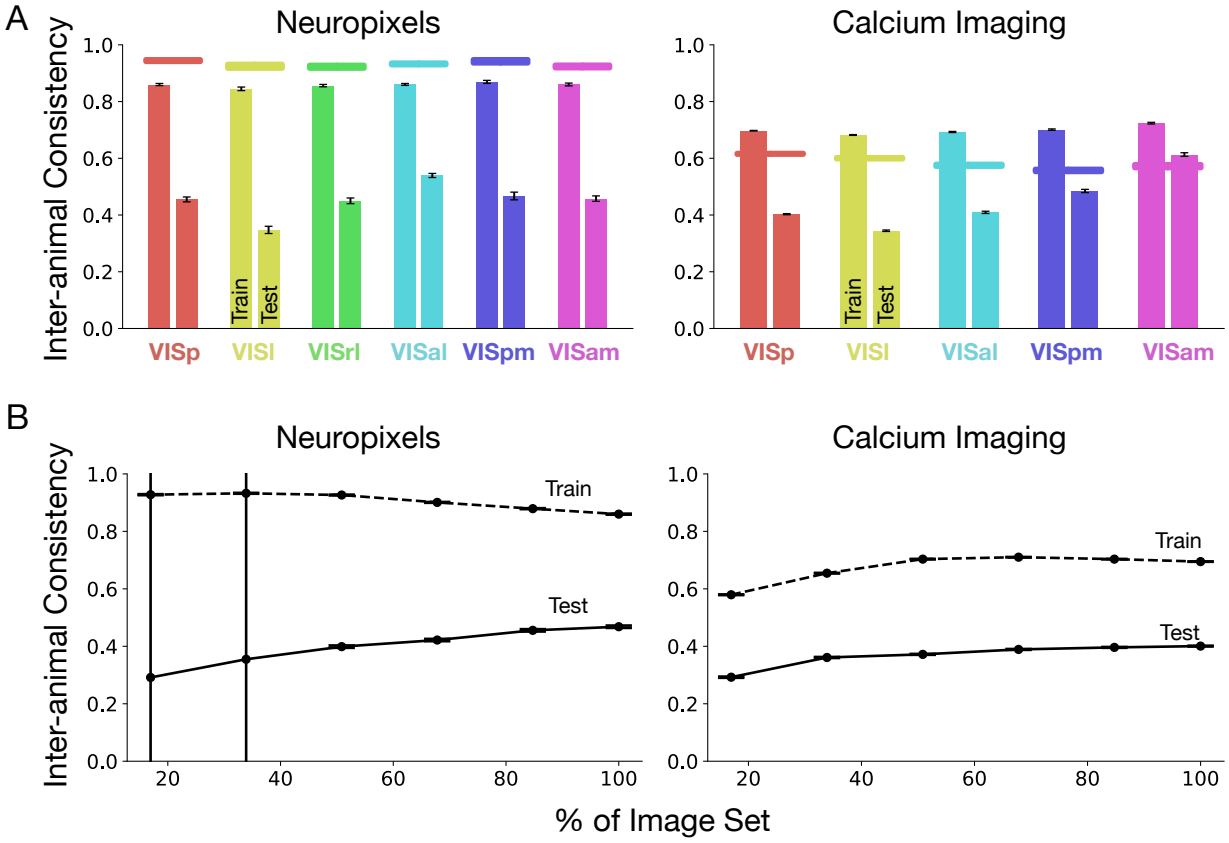

Figure S7: **Inter-animal consistency can increase with more stimuli. A**. Inter-animal consistency under PLS regression evaluated on the train set (left bars for each visual area) and test set (right bars for each visual area), for both Neuropixels and calcium imaging datasets. The horizontal lines are the internal consistency (split half reliability). **B**. Inter-animal consistency under PLS regression on the train set (dotted lines) and test set (straight lines), aggregated across visual areas. Each dot corresponds to the inter-animal consistency evaluated across 10 train-test splits, where each split is a sample of the natural scene image set corresponding to the percentage (x-axis). Note that VISrl is excluded for calcium imaging, as explained in the text. The median and s.e.m. across neurons is reported for both panels. Refer to Table S1 for N units per visual area.