# Mouse visual cortex as a limited resource system that self-learns an ecologically-general representation

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

Stanford University

**Abstract**

Studies of the mouse visual system have revealed a variety of visual brain areas that are thought to support a multitude of behavioral capacities, ranging from stimulus-reward associations, to goal-directed navigation, and object-centric discriminations. However, an overall understanding of the mouse's visual cortex, and how it supports a range behaviors, remains unknown. Here, we take a computational approach to help address these questions, providing a high-fidelity quantitative model of mouse visual cortex and identifying key structural and functional principles underlying that model's success. Structurally, we find that a comparatively shallow network structure with a low-resolution input is optimal for modeling mouse visual cortex. Our main finding is functional – that models trained with task-agnostic, self-supervised objective functions based on the concept of contrastive embeddings are much better matches to mouse cortex, than models trained on supervised objectives or alternative self-supervised methods. This result is very much unlike in primates where prior work showed that the two were roughly equivalent, naturally leading us to ask the question of why these self-supervised objectives are better matches than supervised ones in mouse. To this end, we show that the self-supervised, contrastive objective builds a general-purpose visual representation that enables the system to achieve better transfer on out-of-distribution visual, scene understanding and reward-based navigation tasks. 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

The mouse has become an indispensable model organism in systems neuroscience, allowing unprecedented genetic and experimental control at the level of cell-type specificity in individual circuits (Huberman and Niell, 2011). Moreover, studies of mouse visual behavior have revealed a multitude of abilities, ranging from stimulus-reward associations, to goal-directed navigation, and object-centric discriminations (Glickfeld and Olsen, 2017). Thus, the mouse animal model bridges fine-grained, low-level experimental control with high-level behavior. Understanding the principles underlying the structure and function of the mouse visual system and how it relates to more commonly studied visual organisms such as macaque, is therefore important.

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

Computational models are an effective tool for understanding how mouse visual cortex is capable of supporting such behaviors and for providing a normative account for its structure and function. They allow us to identify the key ingredients leading to the model with the best quantitative agreement with the neural data. We can also assess the functional similarities and differences between rodent and primate visual cortex, which otherwise would be hard to capture in the absence of a model from prior literature, beyond failures to be homologous in higher visual areas past V1. Furthermore, these models provide a natural starting point for understanding higher-level processing downstream of the visual system, such as in memory and its role during the navigation of rich visual environments (O'Keefe and Dostrovsky, 1971; Hafting et al., 2005; Kropff et al., 2015; Solstad et al., 2008; Sargolini et al., 2006). Without an explicit model of a visual system, it is difficult to disentangle the contributions to neural response variance of the visual system from those of higher-level cognitive phenomena. Understanding the computations underlying higher cognition and motor control in rodents will therefore critically depend on an understanding of the upstream sensory areas that they depend on.

Deep convolutional neural networks (CNNs) are a class of models that have had 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, such models have been poor predictors of neural responses in mouse visual cortex (Cadena et al., 2019b; de Vries et al., 2020). Our hypothesis is that this failure can be understood via and be remedied by the goal-driven modeling approach (Yamins and DiCarlo, 2016). This approach posits that normative models in neuroscience should pay careful attention to the objective functions (i.e., behavior), architectures (i.e., neural circuit), and data stream (i.e., visual input). These structural and functional ingredients should be finely tuned to the biology and the ecology of the organism under study.

In this work, we build a substantially improved model of mouse visual cortex by better aligning the objective function, architecture, and visual input with those of the mouse visual system. Firstly, from an objective function point of view, the primate ventral stream models 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. Instead, we find that models trained using self-supervised, contrastive algorithms provide the best correspondence to mouse visual responses. Interestingly, this situation is different than in primates, where prior worked showed that the two were roughly equivalent (Zhuang et al., 2021). Secondly, in terms of the architecture, these primate ventral visual stream 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; Gămănuţ and Shimaoka, 2021). By varying the number of linear-nonlinear layers in the models, we find that models with fewer linear-nonlinear layers can achieve neural predictivity performance that is better or on par with very deep models. 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. Indeed, we find that model fidelity can be improved by training them on lower-resolution images. Ultimately, the confluence of these ingredients, leads to a model, known as "Contrastive ShallowNet", that best matches mouse visual cortex thus far.

We then address the question of why the Contrastive ShallowNet is better at neural predictivity from an ecological point of view, especially the role of contrastive, self-supervised learning, which is novel and not expected by the known physiology and behavioral experiments in mouse.

To address this question, we use Contrastive ShallowNet to assess out-of-distribution generalization from its original training environment, including using the visual encoder as the front-end of a bio-mechanically realistic virtual rodent operating in an environment that supports spatially-extended reward-based navigation. We show that visual representations of this model lead to improved transfer performance over its supervised counterpart across environments, illustrating the congruence between the task-transfer performance and improved neural fidelity of the computational model.

Taken together, our best models of the mouse visual system suggest that it is a shallower, general-purpose system operating on comparatively low-resolution inputs. These identified factors therefore provide interpretable insight into the confluence of constraints that may have given 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 Modeling mouse visual cortex

## 2.1 Building quantitatively accurate models

Prior models of mouse visual cortex can be improved by varying three ingredients to better match the biology and the ecology of mouse visual cortex. Before model development, however, we must determine the appropriate procedure by which to evaluate models. As in prior work on modeling primate visual cortex, we "map" model responses to biological responses and the ability of the model responses to recapitulate biological responses determines the model's neural fidelity (Yamins et al., 2014; Khaligh-Razavi and Kriegeskorte, 2014; Schrimpf et al., 2018; Zhuang et al., 2021).

How should one map from model responses to biological responses? We identified the best-performing mapping function by assessing a variety of functions to map responses from one animal to those of another (see Methods; Figure S1A). By performing this animal-to-animal comparison, we not only identify the mapping function for model evaluation in a principled manner, but also identify neural response variance that is common across animals and therefore should be "explained" by models. This common neural response variance across animals is denoted as the "inter-animal consistency". Across all the mapping functions we used for mapping animals to animals, we found that partial least-squares (PLS) regression to yield the most consistent fits across visual areas. Thus, we used PLS regression to map from model responses to biological responses (we observe similar results with non-regression-based metrics in the Supplement).

With this mapping and evaluation procedure, we can then develop models to better match mouse visual responses. Our best models attained neural predictivity of $90\%$ of the inter-animal consistency, much better than the prior high-resolution, deep, and task-specific model (VGG16), which attained $56.27\%$ of this ceiling (Figure 1A). We also attain neural predictivity improvements over prior work (Bakhtiari et al., 2021; Shi et al., 2020), especially the latter "MouseNet" of Shi et al. (2020), which attempts to map details of the mouse connectome (Harris et al., 2016; Knox et al., 2018) onto a CNN architecture. This is another conceptual motivation for our structural-and-functional goal-driven approach, as the higher-level constraints are easier to interrogate than to incorporate and assume individual biological details, as this can be a very under constrained procedure.

In the subsequent subsections, we distill the three factors that contributed to models with improved correspondence to the mouse visual areas: objective function, input resolution, and architecture.

### 2.2 Objective function: Training models on self-supervised, contrastive objectives, instead of supervised objectives, improves correspondence to mouse visual areas

The success in modeling the human and the non-human primate visual system has largely been driven by convolutional neural networks trained in a supervised manner on ImageNet (Deng et al., 2009) to perform object categorization (Khaligh-Razavi and Kriegeskorte, 2014; Bashivan et al., 2019). This suggests that models trained with category-label supervision learn useful visual representations that are well-matched to those of the primate ventral visual stream (Schrimpf et al., 2018; Zhuang et al., 2021). Thus, although biologically implausible for the rodent, category-label supervision is a useful starting point for building baseline models and indeed, models trained in this manner are much improved over the prior primate model (black bars in Figure 1A).

These improved, category-label-supervised models, however, cannot be an explanation for how the mouse visual system developed in the first place. In particular, it is unclear whether or not rodents can perform well on large-scale object recognition tasks when trained, such as tasks where there are hundreds of labels. For example, they obtain approximately 70% on a two-alternative forced-choice object classification task (Froudarakis et al., 2020). Furthermore, the categories of the ImageNet dataset are human-centric and therefore not relevant for rodents.

We therefore turned to more ecological supervision signals and self-supervised objective functions, where high-level category labeling is not necessary. These objectives could possibly lead to models with improved biological plausibility and may provide more general goals for the models, based on natural image statistics, beyond the semantic specifics of (human-centric) object categorization. Among the more ecological *supervised* signals, we consider less categorization (with 10 classes rather than 1000 (Krizhevsky et al., 2009)), and depth prediction (Zhang et al., 2017), as a visual proxy for whisking Zhuang et al. (2017a). Turning to *self-supervision*, early self-supervised objectives include sparse autoencoding (Olshausen and Field, 1996), which has been shown to be successful at producing Gabor-like functions reminiscent of the experimental findings in the work of Hubel and Wiesel (1962). These objectives, however, have not led to quantitatively accurate models of higher visual cortex (Yamins et al., 2014; Zhuang et al., 2021). Further developments in computer vision have led to other self-supervised algorithms, motivated by the idea that "non-semantic" features are highly related to the higher-level, semantic features (i.e., category labels), such as predicting the rotation angle of an image (Gidaris et al., 2018). Although these objectives are very simple, models optimized on them do not result in "powerful" visual representations for downstream tasks. We trained models on these objectives and showed that although they improve neural predictivity over the prior primate model (VGG16), they do not outperform category-label-supervised models (Figure 1A; compare pink, blue, and orange with black).

Further developments in self-supervised learning provided a new class of *contrastive* objectives. These objectives are much more powerful than prior self-supervised objectives described above, as it has been shown that models trained with contrastive objectives leads to visual representations that can support strong performance on downstream object categorization tasks. 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. We trained models using the family of contrastive objective functions including instance recognition (IR, Wu et al., 2018), a 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). Models trained with these contrastive objectives resulted in higher neural predictivity across all the visual areas than models trained on supervised object categorization, depth prediction, and less-powerful self-supervised algorithms (red bars in Figure 1A).

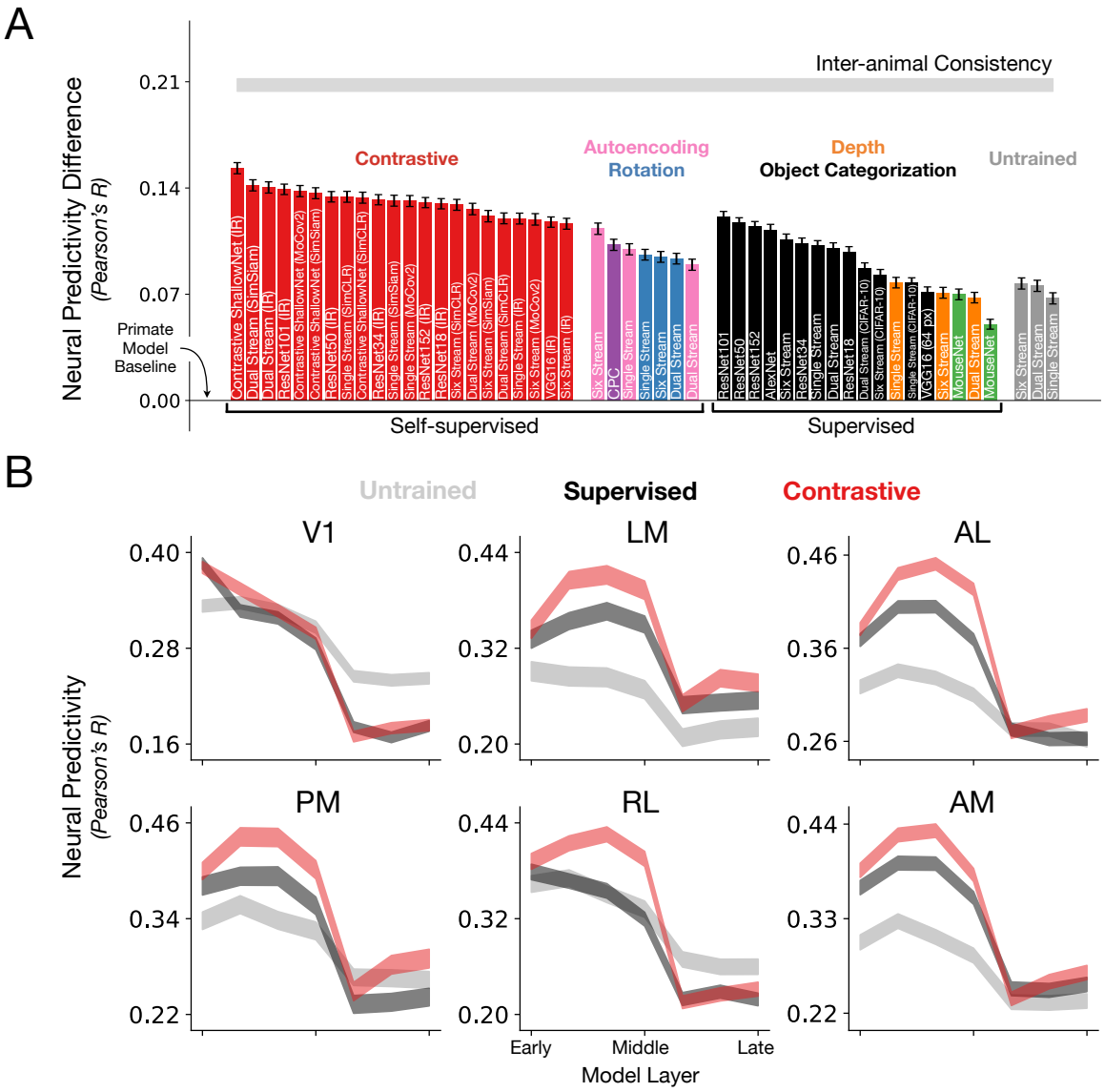


Figure 1: **Substantially improving neural response predictivity of models of mouse visual cortex. A**. The median and s.e.m. neural predictivity difference with the prior-used, primate model of VGG16 trained on 224 px inputs (de Vries et al., 2020; Cadena et al., 2019b; Shi et al., 2019), under PLS regression, across units in all mouse visual areas ($N = 1731$ units in total). Absolute neural predictivity (always computed from the model layer that best predicts a given visual area) for each model can be found in Table S3. Our best model is denoted as "Contrastive ShallowNet (IR)" on the far left. CPC denotes contrastive predictive coding (Oord et al., 2018; Bakhtiari et al., 2021). All models are trained on 64 px inputs unless otherwise stated in figure. **B**. Training a model on a contrastive objective improves neural predictivity across all visual areas. For each visual area, the neural predictivity values are plotted across all model layers for an untrained AlexNet, Supervised (ImageNet) AlexNet and Contrastive AlexNet (ImageNet, instance recognition) – the latter's first four layers form "Contrastive ShallowNet (IR)", the best model of mouse visual cortex. Shaded regions denote mean and s.e.m. across units.

We hone in on the contribution of the contrastive objective function (over a supervised objective) to neural predictivity by fixing the architecture to be AlexNet, while varying the objective function[1]. Across all the visual areas, there is an improvement in neural predictivity simply by using a powerful contrastive algorithm (red vs. black in Figure 1B). The first four layers of Contrastive AlexNet (red, Figure 1B), where neural predictivity is maximal across visual areas, forms our model for mouse visual cortex, named "Contrastive ShallowNet".

### 2.3 Data stream: Training models on images of lower resolution improves correspondence to mouse visual areas

The visual acuity of mice is known to be lower than the visual acuity of primates (Prusky et al., 2000; Kiorpes, 2019). Thus, more accurate models of mouse visual cortex must be trained and evaluated at image resolutions that are lower than those used in the training of models of the primate visual system. We investigated how neural predictivity of two strong *contrastive* models varied as a function of the image resolution at which they were trained.

Two models were used in the exploration of image resolution's effects on neural predictivity. Contrastive AlexNet was trained with image resolutions that varied from $64 \times 64$ pixels to $224 \times 224$ pixels, as $64 \times 64$ pixels was the minimum image size for AlexNet due to its architecture. The image resolution upper bound of $224 \times 224$ pixels is the image resolution that is typically used to train neural network models of the primate ventral visual stream (Schrimpf et al., 2018). We also investigated image resolution's effects using another model known as "Contrastive StreamNet". This model was based on the first four layers of AlexNet, as in Contrastive ShallowNet. Unlike Contrastive ShallowNet, StreamNet incorporates dense skip connections, known from the feedforward connectivity of the mouse connectome (Harris et al., 2016; Knox et al., 2018), as well as multiple parallel streams (schematized in Figure S2). This model was trained using a contrastive objective function (instance recognition) at image resolutions that varied from $32 \times 32$ pixels to $224 \times 224$ pixels, thus enabling us to explore a lower range of image resolutions than the original AlexNet.

Training models using resolutions lower than what is used for models of primate visual cortex improves neural predictivity across all visual areas, but not beyond a certain resolution, where neural predictivity decreases (Figure 2). Although the input resolution of $64 \times 64$ pixels may not be optimal for every architecture, it was 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. Prior retinotopic map studies (Zhuang et al., 2017b) estimate a visual coverage range in V1 of 60-90 degrees, and we found $64 \times 64$ pixels to be roughly optimal for the models (Figure 2) and was also used by Shi et al. (2019), by Bakhtiari et al. (2021), and in the MouseNet of Shi et al. (2020). Although downsampling training images is a reasonable proxy for the mouse retina, as has been done in prior modeling work, more investigation into appropriate image transformations may be needed.

Overall, these data show that optimization using a simple change in the image statistics (i.e., data stream) is crucial to obtain improved models of mouse visual encoding. This suggests that mouse visual encoding is the result of "task-optimization" at a lower resolution than what is typically used for primate ventral stream models.

[1]Note that this contribution to neural predictivity extends beyond the augmentation used for each objective function. We show in Figure S8 that using image augmentations intended for contrastive losses but with supervised losses does not improve the original neural predictivity of the supervised objective (with the supervised augmentations).

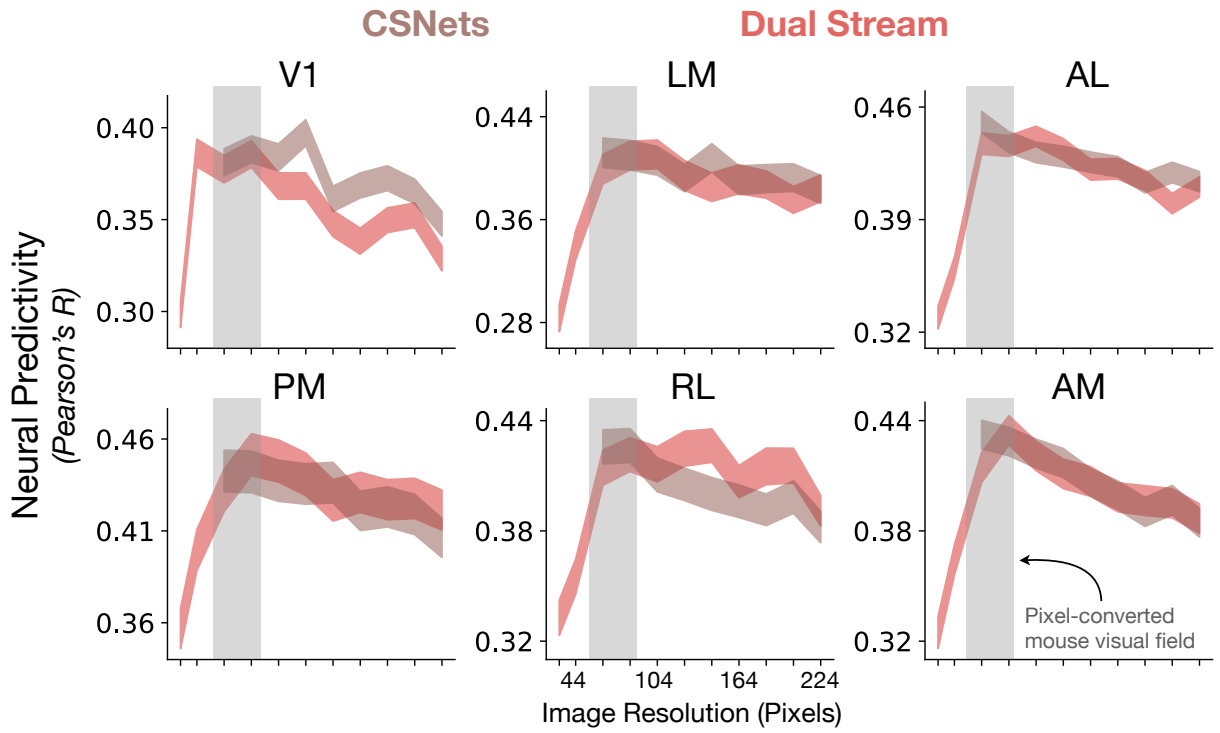


Figure 2: **Lower-resolution training leads to improved neural response predictivity (Neuropixels dataset).** Contrastive AlexNet (from which Contrastive ShallowNet, abbreviated as "CSNet", is derived from its first four layers) and Contrastive StreamNet (Dual Stream) were trained using images of increasing resolutions. For AlexNet, the minimum resolution was $64 \times 64$ pixels. For Dual StreamNet, the minimum resolution was $32 \times 32$ pixels. The median and s.e.m. neural predictivity across all units of all visual areas is plotted against the image resolution at which the models were trained. Reducing the image resolution (but not beyond a certain point) during representation learning improves the match to all the visual areas. The conversion from image resolution in pixels to visual field coverage in degrees is under the assumption that the upper bound on mouse visual acuity is $0.5$ cycles per degree (Prusky et al., 2000) and that the Nyquist limit is $2$ pixels per cycle, leading to a conversion ratio of $1$ pixel per degree.

### 2.4 Architecture: Shallow models suffice for improving correspondence to mouse visual cortex

Anatomically, the mouse visual system has a shallower hierarchy relative to that of the primate visual system (see, e.g., Harris et al., 2019; Siegle et al., 2021; Felleman and Van Essen, 1991; Gămănuţ and Shimaoka, 2021). Analyzing the reliability of the neural responses for each visual area provides further support for the relatively shallow functional hierarchy (Figure S1B, C). Thus, more biologically-plausible models of mouse visual cortex should have fewer linear-nonlinear layers than those of primate visual cortex. By plotting a model's neural predictivity against its number of linear-nonlinear operations, we indeed found that very deep models do not outperform shallow models (e.g., models with less than 12 linear-nonlinear operations), despite changes across loss functions and input resolutions (Figure 3). This observation is in direct contrast with prior observations in the primate visual system whereby networks with fewer than 18 linear-nonlinear operations predict macaque visual responses less well than models with at least 50 linear-nonlinear operations (Schrimpf et al., 2018).

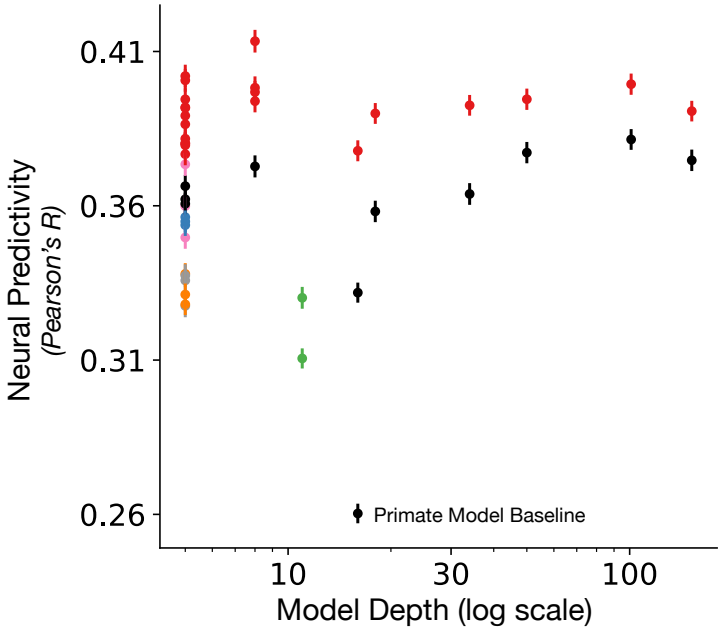


Figure 3: **Neural predictivity vs. model depth.** A model's median neural predictivity across all units from each visual area is plotted against its depth (number of linear-nonlinear layers; in log-scale). Models with fewer linear-nonlinear layers can achieve neural predictivity performances that outperform or are on par with those of models with many more linear-nonlinear layers. Color scheme as in Figure 1A.

## 3 Mouse visual cortex as a general-purpose visual system

Our results show that a model optimized on a self-supervised contrastive objective the most quantitatively accurate model of the mouse visual system. However, unlike supervised object categorization or other more-classical forms of self-supervised learning such as sparse autoencoding, whose ecological function is directly encoded in the loss function (e.g., predator recognition or

metabolically-efficient dimension reduction), the functional utility of self-supervision via a contrastive objective is not as apparent. This naturally raises the question of what behavioral function(s) optimizing a contrastive self-supervision objective might enable for the mouse from an ecological fitness viewpoint.

Analyzing the spectrum of models described in the above section, we first observed that performance on ImageNet categorization is uncorrelated with improved neural predictivity for mouse visual cortex (Figure 4), unlike the well-known correlation in primates (Figure 4; inset). In seeking an interpretation of the biological function of the contrastive self-supervised objective, we were thus prompted to consider behaviors beyond object-centric categorization tasks. We hypothesized that since self-supervised loss functions are typically most effective in task-agnostic stimulus domains where goal-specific labels are unavailable, optimizing for such an objective might enable the rodent to *transfer* well to complex multi-faceted ecological tasks in novel domains affording few targeted opportunities for hyper-specialization.

To test this hypothesis, we used a recently-developed "virtual rodent" framework (adapted from Merel et al. (2020) and also used by Lindsay et al. (2021)), in which a biomechanically-validated mouse model is placed in a simulated 3D maze-like environment (Figure 5A). We used the model that best corresponds to mouse visual responses as the visual system of the simulated mouse, coupled to a simple actor-critic reinforcement-learning architecture (Figure 5B). We then trained the simulated mouse in several visual contexts with varying objective functions, and evaluated those models' ability to *transfer* to a variety of tasks in novel environments, including both a reward-based navigation task, as well as several object-centric visual categorization and estimation tasks (Figure 5C).

We first evaluated a simulated mouse whose visual system was pretrained with ImageNet images in terms of its ability to transfer to reward-based navigation in the Maze environment, training just the reinforcement learning portion of the network on the navigation task with the pretrained visual system held constant. As a supervised control, we performed the same transfer training procedure using a visual front-end created using supervised (ImageNet) object categorization pretraining. We found that the simulated mouse with the contrastive self-supervised visual representation was able to reliably obtain substantially higher navigation rewards than its counterpart with category-supervised visual representation (Figure 6A).

Conversely, we trained the simulated mouse directly in the Maze environment. In one variant, we trained the visual system on the contrastive self-supervised objective (but on the images of the Maze environment). In a second variant, we trained the agent end-to-end on the reward navigation task itself – the equivalent of "supervision" in the Maze environment. We then tested both models on their ability to transfer to out-of-sample visual categorization and estimation tasks. Again, we found that the self-supervised variant transfers significantly better than its "supervised" counterpart (blue vs. purple bars in Figure 6B).

Returning to analyses of neural predictivity, we found that both self-supervised models (trained in either environment) were better matches to mouse visual cortex neurons than the supervised counterparts in their respective environments (red and blue vs. black and purple in Figure 6C). This result illustrates the congruence between the task-transfer performance and improved neural fidelity of the computational model. It is also noteworthy that, comparing models by training environment rather than objective function, the ImageNet-trained models produce more neurally-consistent models than the Maze-trained models, both for supervised and self-supervised objectives (red vs. blue and black vs. purple in Figure 6C). Using the contrastive self-supervised objective function is by itself enough to raise the Maze-trained model to the predictivity level of the ImageNet-trained supervised model, but the training environment makes a significant contribution. This suggests that while the task domain and biomechanical model of the Maze environment

are realistic, future work will likely need to improve the realism of the simulated image distribution.

Overall, these results suggest that contrastive embedding methods have achieved a generalized improvement in the quality 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. While we expect the calculation of rewards to be performed *outside* of the visual system, we expect the mouse visual system would support the visual aspects of the transfer tasks we consider, as we find that higher model areas best support these scene understanding transfer tasks (not unlike the primate, where downstream visual areas support a variety of visual recognition tasks).

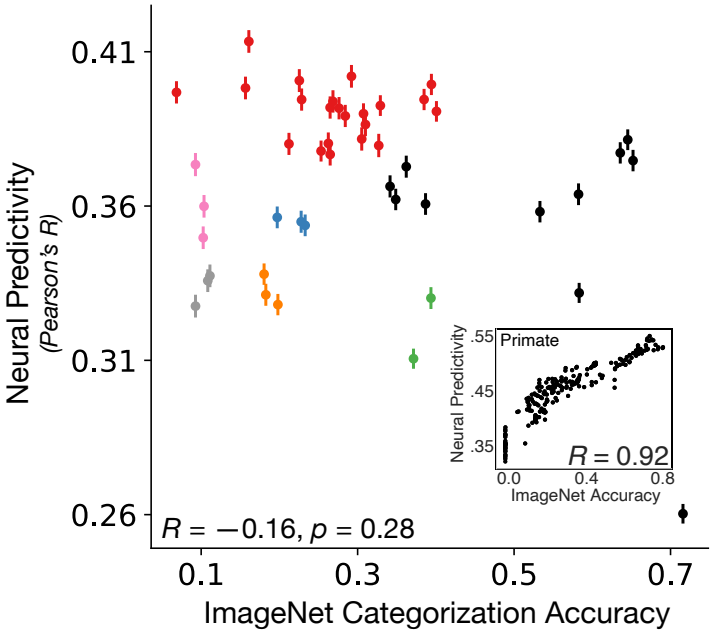


Figure 4: **Neural predictivity is *not* correlated with object categorization performance on ImageNet.** Each model's (transfer or supervised) object categorization performance on ImageNet is plotted against its median neural predictivity across all units from all visual areas. All ImageNet performance values can be found in Table S3. **Inset**. Primate ventral visual stream neural predictivity from BrainScore is correlated with ImageNet categorization accuracy (adapted from Schrimpf et al., 2018). This relationship is in stark contrast to our finding in mouse visual cortex where higher ImageNet categorization accuracy is *not* associated with higher neural predictivity. Color scheme as in Figure 1A. See Figure S3 for neural predictivity on the calcium imaging dataset.

## 4 Discussion

In this work, we showed that comparatively shallow architectures trained with contrastive objectives operating on lower-resolution images most accurately predict static-image-evoked neural responses across multiple mouse visual areas, surpassing the predictive power of supervised methods and approaching the inter-animal consistency. The fact that these goal-driven constraints lead to a better match to visual responses, even in "behaviorally-free" data where mice are passively viewing stimuli, suggests that these constraints may be good descriptions of the evolutionary and

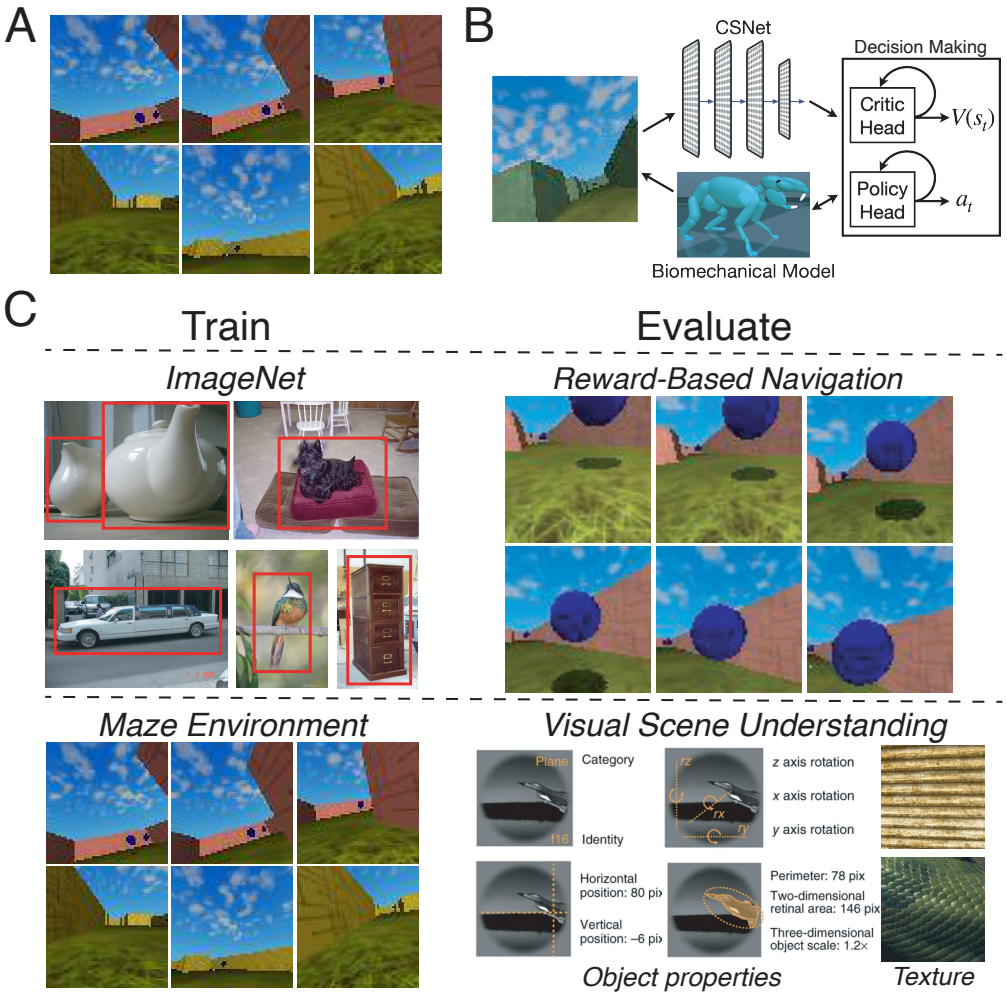


Figure 5: **Evaluating the generality of learned visual representations.** **A**. Each row shows an example of an episode used to train the reinforcement learning (RL) policy in an offline fashion (Wang et al., 2020). The episodes used to train the virtual rodent (adapted from Merel et al., 2020) were previously generated and are part of a larger suite of RL tasks for benchmarking algorithms (Gulcehre et al., 2020). In this task, the goal of the agent is to navigate the maze to collect as many rewards as possible (blue orbs shown in the first row). **B**. A schematic of the RL agent. The egocentric visual input is fed into the visual backbone of the model, which is fixed to be the first four convolutional layers of either the contrastive or the supervised variants of AlexNet. The output of the visual encoder is then concatenated with the virtual rodent's proprioceptive inputs before being fed into the (recurrent) critic and policy heads. The parameters of the visual encoder are *not* trained, while the parameters of the critic head and the policy head are trained. **C**. A schematic of the out-of-distribution generalization procedure. Visual encoders are either trained in a supervised or self-supervised manner on ImageNet or the Maze environment, and then evaluated on reward-based navigation or datasets (from Hong et al., 2016) consisting of object properties (category, pose, position, and size) and different textures.

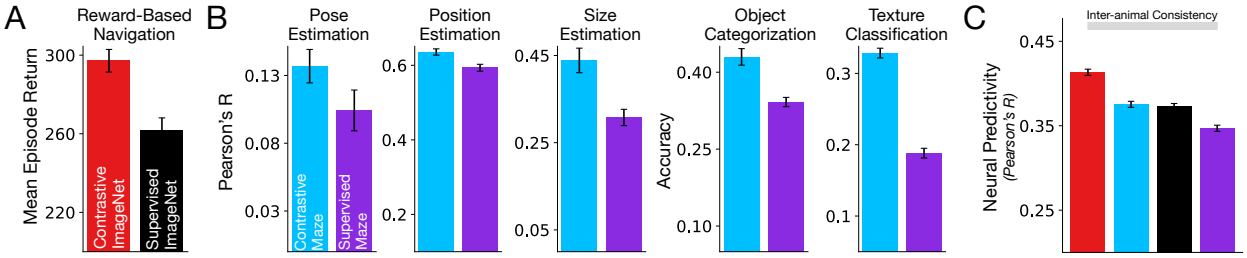


Figure 6: **Self-supervised, contrastive visual representations better support transfer performance on downstream, out-of-distribution tasks.** Models trained in a contrastive manner (using either ImageNet or the egocentric maze inputs; red and blue respectively) lead to better transfer on out-of-distribution downstream tasks than models trained in a supervised manner (i.e., models supervised on labels or on rewards; black and purple respectively). **A**. Models trained on ImageNet, tested on reward-based navigation. **B**. Models trained on egocentric maze inputs ("Contrastive Maze", blue) or supervised on rewards (i.e., reward-based navigation; "Supervised Maze", purple), tested on visual scene understanding tasks: pose, position, and size estimation, and object and texture classification. **C**. Median and s.e.m. neural predictivity across units in the Neuropixels dataset.

the developmental drivers of mouse visual cortical structure and function.

In the primate ventral visual stream, models trained on contrastive objectives led to neural predictivity performance that was *on par* with that of supervised models (Zhuang et al., 2021), suggesting they are a more ecologically-valid proxy for a categorization-specialized system. This is in stark contrast with our findings in our models of mouse visual cortex – we found that models trained on contrastive objectives *substantially surpassed* the neural predictivity of their supervised counterparts. We investigated the advantages that contrastive objectives might confer over supervised objectives for mouse visual representations and found that they provide representations that are generally improved over those obtained by supervised methods. The improvement in generality of contrastive models enabled better transfer to a diverse range of downstream behaviors in novel, *out of distribution* environments, including reward-based navigation in egocentric maze environments and visual scene understanding.

As mentioned previously, the goal-driven modeling approach allows us to understand the principles that govern the system under study and further allows for direct comparisons across systems. Our high-fidelity model of mouse visual cortex and the principles underlying its construction can be compared with models of primate visual cortex. While the primate ventral visual stream is well-modeled by a deep hierarchical system and object-category learning, mouse as a model visual system has not had such a coherent account heretofore. Our results demonstrate that both primate and rodent visual systems are highly constrained, albeit for different functional purposes. The results summarized above further suggest that mouse visual cortex is a light-weight, shallower, low-resolution, and general-purpose visual system in contrast to the deep, high-resolution, and more task-specific visual system of primates, suggested by prior work (Zhuang et al., 2021).

Although we have made progress in modeling the mouse visual system in three core ways (the choice of objective function, data stream, and architecture class), some limitations remain both in the modeling and in the neural data.

On the architectural front, 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., 2022).

By incorporating recurrent connections in the architectures, we can probe the functionality of these feedback connections using self-supervised loss functions in scenarios with temporally-varying, dynamic inputs. For example, given that powerful self-supervised methods obtained good visual representations of static images, it would be interesting to explore a larger spectrum of self-supervised 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).

Constraining the input data so that they are closer to those received by the mouse visual system was important for improved neural fidelity. Our resizing (i.e., downsampling) of the images to be smaller during training acted as a proxy for low-pass filtering. We believe that future work could investigate other appropriate low-pass filters and ecologically-relevant pixel-level transformations to apply to the original image or video stream.

Our inter-animal consistency analyses make recommendations for qualities of experiments that are likely to provide data that will be helpful in more sharply differentiating models. Under all mapping functions, a log-linear extrapolation analysis revealed that as the number of units increases, the inter-animal consistency approaches one more rapidly for the Neuropixels dataset, than for the calcium imaging dataset, indicating the higher reliability of the Neuropixels data (Figure S6). Moreover, when assessing inter-animal consistency, correlation values between animals were significantly higher for the training set than for the test set, indicating that the number of stimuli could be enlarged to close this generalization gap (Figure S7A). As a function of the number of stimuli, the test set inter-animal consistencies steadily increased, and would likely continue to increase substantially if the dataset had more stimuli (Figure S7B). Thus, while much focus in experimental methods has been on increasing the number of neurons in a dataset (Steinmetz et al., 2021), our analyses indicate that increasing the number of stimuli may drastically improve model identification. Doing so would likely raise the inter-animal consistency, providing substantially more dynamic range for separating models in terms of their ability to match the data, potentially allowing us to obtain more specific conclusions about which circuit structure(s) (Collins et al., 2017; Bergstra et al., 2015) and which (combinations of) objectives (e.g., Wu et al., 2018; Chen and He, 2020; Chen et al., 2020a) best describe mouse visual cortex.

We endeavor that our work in modeling mouse visual cortex will meaningfully drive future experimental and computational studies in mice of other sensory systems and of visually-guided behaviors. The input-domain-agnostic nature of these contrastive objectives suggest the tantalizing possibility that they might be used in other sensory systems, such as barrel cortex or the olfactory system. By building high-fidelity computational models of sensory cortex, we believe that they can be integrated with models of higher-order systems (e.g., medial temporal lobe), with the goal of providing us with greater insight into how sensory experience contributes to adaptive or maladaptive behaviors.

## 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.

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.

Răzvan Gămănuţ and Daisuke Shimaoka. Anatomical and functional connectomes underlying hierarchical visual processing in mouse visual system. *Brain Structure and Function*, pages 1–19, 2021.

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.

Caglar Gulcehre, Ziyu Wang, Alexander Novikov, Thomas Paine, Sergio Gómez, Konrad Zolna, Rishabh Agarwal, Josh S Merel, Daniel J Mankowitz, Cosmin Paduraru, Gabriel Dulac-Arnold, Jerry Li, Mohammad Norouzi, Matthew Hoffman, Nicolas Heess, and Nando de Freitas. RL unplugged: A suite of benchmarks for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 7248–7259, 2020.

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.

Torkel Hafting, Marianne Fyhn, Sturla Molden, May-Britt Moser, and Edvard I Moser. Microstructure of a spatial map in the entorhinal cortex. *Nature*, 436(7052):801–806, 2005.

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.

David H Hubel and Torsten N Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of Physiology*, 160(1):106–154, 1962.

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

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.

Emilio Kropff, James E Carmichael, May-Britt Moser, and Edvard I Moser. Speed cells in the medial entorhinal cortex. *Nature*, 523(7561):419–424, 2015.

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.

Grace W Lindsay, Josh Merel, Tom Mrsic-Flogel, and Maneesh Sahani. Divergent representations of ethological visual inputs emerge from supervised, unsupervised, and reinforcement learning. *arXiv preprint arXiv:2112.02027*, 2021.

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.

Josh Merel, Diego Aldarondo, Jesse Marshall, Yuval Tassa, Greg Wayne, and Bence Ölveczky. Deep neuroethology of a virtual rodent. *International Conference on Learning Representations*, 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. Recurrent connections in the primate ventral visual stream mediate a tradeoff between task performance and network size during core object recognition. *Neural Computation*, 34:1652–1675, 2022.

John O'Keefe and Jonathan Dostrovsky. The hippocampus as a spatial map: Preliminary evidence from unit activity in the freely-moving rat. *Brain research*, 1971.

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.

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.

Francesca Sargolini, Marianne Fyhn, Torkel Hafting, Bruce L McNaughton, Menno P Witter, May-Britt Moser, and Edvard I Moser. Conjunctive representation of position, direction, and velocity in entorhinal cortex. *Science*, 312(5774):758–762, 2006.

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.

Trygve Solstad, Charlotte N Boccara, Emilio Kropff, May-Britt Moser, and Edvard I Moser. Representation of geometric borders in the entorhinal cortex. *Science*, 322(5909):1865–1868, 2008.

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.

Ziyu Wang, Alexander Novikov, Konrad Zolna, Josh S Merel, Jost Tobias Springenberg, Scott E Reed, Bobak Shahriari, Noah Siegel, Caglar Gulcehre, Nicolas Heess, et al. Critic regularized regression. *Advances in Neural Information Processing Systems*, 33:7768–7778, 2020.

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, 2017a.

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.

Jun Zhuang, Lydia Ng, Derric Williams, Matthew Valley, Yang Li, Marina Garrett, and Jack Waters. An extended retinotopic map of mouse cortex. *elife*, 6:e18372, 2017b.

# 5 Methods

## 5.1 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 V1 (VISp), LM (VISl), AL (VISal), RL (VISrl), AM (VISam), and PM (VISpm) 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 $75$th 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 5.3).

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 RL of the calcium imaging dataset (green in Figure S1A). This difference, however, was not observed for responses in RL 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 field-of-view for RL 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 for the calcium imaging data, 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.

## 5.2 Noise-Corrected Neural Predictivity

### 5.2.1 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 regression, we use an $\alpha = 1$, following the `sklearn.linear_model` convention. 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 S1B, 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.

| Dataset Type | Visual Area | Total Units | Total Specimens |
|---|---|---|---|
| Calcium Imaging | V1 | 7080 | 29 |
| | LM | 4393 | 24 |
| | AL | 2064 | 9 |
| | RL | 1116 | 8 |
| | AM | 847 | 9 |
| | PM | 1844 | 19 |
| Neuropixels | V1 | 442 | 8 |
| | LM | 162 | 6 |
| | AL | 396 | 6 |
| | RL | 299 | 7 |
| | AM | 257 | 7 |
| | PM | 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.

Motivated by the justification given in Section 5.3 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 5.3.2, 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 $r^\ell$ be the set of model responses at model layer $\ell \in L$, $M$ be the mapping, and let $\mathrm{s}$ 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 5.3.5).

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 $224 \times 224$ pixels, and finally resized the stimulus to $64 \times 64$ pixels. For models trained using the CIFAR-10 dataset, this resizing yielded a $64 \times 81$ pixels stimulus.

#### 5.2.2 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 $r^\ell$ be the model responses at model layer $\ell$ and let $\mathrm{s}$ 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 $\mathsf{A}$ is the trial-wise, deterministic model features (which have a mapping consistency of 1 as a result) and a single target animal $\mathsf{B}$, which is the pseudo-population response:

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

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 5.3.5.

If the fits are performed separately for each animal, then $\mathsf{B}$ 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, $\mathsf{RSA}\left(\mathrm{s}_1^{\mathsf{B}}, \mathrm{s}_2^{\mathsf{B}}\right)$.

## 5.3 Inter-Animal Consistency Derivation

### 5.3.1 Single Animal Pair

Suppose we have neural responses from two animals $\mathsf{A}$ and $\mathsf{B}$. Let $\mathrm{t}_i^p$ 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 $\mathrm{t}_i^p$, so let $\mathrm{s}_{j,i}^p$ be the $j$th set of $n$ trials of $\mathrm{t}_i^p$. 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 $\mathsf{B}$ given, as input, the true neural responses of animal $\mathsf{A}$ 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 $\mathsf{B}$ given, as input, the trial-average of noisy sample 1 on the train set stimuli of animal $\mathsf{A}$.

With these definitions in hand, the inter-animal mapping consistency from animal $\mathsf{A}$ to animal $\mathsf{B}$ 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:

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

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 ($\mathrm{t}$) by their measurable estimates ($\mathrm{s}$), starting from the original quantity in Equation (3). First, we make the approximation that:

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

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:

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

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:

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

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):

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

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

which taken together implies the following:

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

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

#### 5.3.2 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.

#### 5.3.3 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:

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

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:

$$\left\langle \mathsf{RSA}\left(\mathrm{t}^{\mathsf{A}}, \mathrm{t}^{\mathsf{B}}\right)\right\rangle_{\mathsf{A}\in\mathcal{A}:(\mathsf{A},\mathsf{B})\in\mathcal{A}\times\mathcal{A}} \sim \left\langle \frac{\mathsf{RSA}\left(\mathrm{s}_1^{\mathsf{A}}, \mathrm{s}_2^{\mathsf{B}}\right)}{\sqrt{\widetilde{\mathsf{RSA}}\left(\mathrm{s}_1^{\mathsf{A}}, \mathrm{s}_2^{\mathsf{A}}\right) \times \widetilde{\mathsf{RSA}}\left(\mathrm{s}_1^{\mathsf{B}}, \mathrm{s}_2^{\mathsf{B}}\right)}} \right\rangle_{\mathsf{A}\in\mathcal{A}:(\mathsf{A},\mathsf{B})\in\mathcal{A}\times\mathcal{A}} . \tag{10}$$

#### 5.3.4 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.

#### 5.3.5 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,

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

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

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

### 5.4 StreamNet Architecture Variants

We developed shallower, multiple-streamed architectures for mouse visual cortex, shown in Figure 1A. 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.

### 5.5 Neural Network Training Objectives

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

| Module Name | Output Size | **Single** $(N = 1)$ | **Dual** $(N = 2)$ | **Six** $(N = 6)$ |
|---|---|---|---|---|
| Input | $64 \times 64$ | N/A | N/A | N/A |
| Shallow | $7 \times 7$ | $(64, 11, 4, 2)$ | $(64, 11, 4, 2)$ | $(64, 11, 4, 2)$ |
| Intermediate | $3 \times 3$ | $\begin{bmatrix}(192, 5, 1, 2)\\(384, 3, 1, 1)\end{bmatrix}$ | $\begin{bmatrix}(192, 5, 1, 2)\\(384, 3, 1, 1)\end{bmatrix} \times 2$ | $\begin{bmatrix}(192, 5, 1, 2)\\(384, 3, 1, 1)\end{bmatrix} \times 6$ |
| Deep | $3 \times 3$ | 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). Note that the first convolutional layer has max pooling of stride 2, as in AlexNet. 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.

#### 5.5.1 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 $i$th image, $\boldsymbol{c}_i \in [0, C-1]$ is the category index of the $i$th image (zero-indexed), and $\boldsymbol{\theta}$ are the model parameters. Equation (11) was minimized using stochastic gradient descent (SGD) with momentum (Bottou, 2010).

**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.

**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) 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 $224 \times 224$ pixels were applied to the image and depth map (which were both subsequently resized to $64 \times 64$ 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 $45$.

#### 5.5.2 Self-supervised 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 $64 \times 64$ pixels image, $\lambda$ 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 $0.0005$.

**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:

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

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:

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

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):

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

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 $0.0001$.

## 5.6 Top-1 Validation Set Performance

### 5.6.1 Performance of primate models on $224 \times 224$ pixels and $64 \times 64$ pixels ImageNet

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

| Architecture | Image Size | Objective Function | Top-1 Accuracy |
|---|---|---|---|
| AlexNet | $224 \times 224$ | Supervised (ImageNet) | 56.52% |
| | $64 \times 64$ | | 36.22% |
| VGG16 | $224 \times 224$ | | 71.59% |
| | $64 \times 64$ | | 58.32% |
| ResNet-18 | $224 \times 224$ | | 69.76% |
| | $64 \times 64$ | | 53.31% |
| ResNet-34 | $224 \times 224$ | | 73.31% |
| | $64 \times 64$ | | 58.24% |
| ResNet-50 | $224 \times 224$ | | 76.13% |
| | $64 \times 64$ | | 63.57% |
| ResNet-101 | $224 \times 224$ | | 77.37% |
| | $64 \times 64$ | | 64.53% |
| ResNet-152 | $224 \times 224$ | | 78.31% |
| | $64 \times 64$ | | 65.21% |

### 5.6.2 Performance of StreamNet Variants on $64 \times 64$ pixels CIFAR-10 and $64 \times 64$ pixels ImageNet

Here we report the top-1 validation set accuracy of our model variants trained in a supervised manner on $64 \times 64$ 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% |

#### 5.6.3 Transfer Performance of StreamNet Variants on $64 \times 64$ pixels ImageNet Under Linear Evaluation for Models Trained with Self-supervised Objectives

In this subsection, we report the top-1 ImageNet validation set performance under linear evaluation for models trained with self-supervised objectives. After training each model on a self-supervised 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 self-supervised 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 $90$.

### 5.7 Reinforcement Learning Task

A set of state-action-reward-state tuples (i.e., $(\boldsymbol{s}_t, \boldsymbol{a}_t, r_t, \boldsymbol{s}_{t+1})$) were generated in prior work (Merel et al., 2020) and were used to train (in an offline fashion) the reinforcement learning (RL) agent. We used an offline RL algorithm known as critic-regularized regression (Wang et al., 2020). Except for the visual encoder, the architecture of the RL agent was identical to that used by Wang et al. (2020) (c.f. Figure 3 in Wang et al., 2020). Four different visual encoders based on the AlexNet architecture were used:

- Contrastive ImageNet: AlexNet trained using ImageNet in a contrastive manner (instance recognition). Up to the first four layers (forming ShallowNet) are implanted into the virtual rodent as its visual system, as these were the layers that best matched mouse visual areas. This visual encoder is the best model of mouse visual cortex (Figure 1). Its weights were held fixed during training of the RL agent.
- Supervised ImageNet: AlexNet trained using ImageNet in a supervised manner. Up to the first four layers (forming ShallowNet) are implanted into the virtual rodent as its visual system, as these were the layers that best matched mouse visual areas. Its weights were held fixed during training of the RL agent.
- Contrastive Maze: AlexNet trained using the egocentric maze inputs from the virtual rodent reward-based navigation task. Up to the first four layers (forming ShallowNet) are implanted into the virtual rodent as its visual system, as these were the layers that best matched mouse visual areas. Its weights were then held fixed during training of the RL agent.
- Supervised Maze: The first four convolutional layers of AlexNet (ShallowNet) trained end-to-end on the virtual rodent reward-based navigation task.

After training the agent until policy loss convergence by $10\,000$ steps (once), the agent was evaluated on $300$ episodes. Each model was trained twice (i.e., two different random seeds), so each model was evaluated on $300 \times 2 = 600$ episodes in total and we report the average reward across all $600$ episodes.

### 5.8 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 in-

| 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% |
| AlexNet | SimSiam | 6.84% | 39.68%; 30.95% |
| AlexNet | MoCov2 | 15.67% | 39.83%; 31.20% |
| AlexNet | SimCLR | 26.85% | 39.39%; 30.27% |
| ResNet-18 | Supervised | 53.31% | 35.82%; 28.93% |
| ResNet-18 | Instance Recognition | 30.75% | 38.99%; 30.11% |
| ResNet-34 | Supervised | 58.24% | 36.38%; 28.42% |
| ResNet-34 | Instance Recognition | 32.91% | 39.26%; 30.93% |
| ResNet-50 | Supervised | 63.57% | 37.72%; 30.70% |
| ResNet-50 | Instance Recognition | 38.49% | 39.45%; 31.12% |
| ResNet-101 | Supervised | 64.53% | 38.15%; 31.05% |
| ResNet-101 | Instance Recognition | 39.43% | 39.94%; 31.60% |
| ResNet-152 | Supervised | 65.21% | 37.74%; 30.67% |
| ResNet-152 | Instance Recognition | 40.08% | 39.07%; 31.30% |
| VGG16 | Supervised | 58.32% | 33.19%; 27.09% |
| VGG16 (224 px) | Supervised | 71.59% | 26.03%; 20.40% |
| VGG16 | Instance Recognition | 25.33% | 37.78%; 28.66% |
| MouseNet of Shi et al. (2020) | Supervised | 37.14% | 31.05%; 25.89% |
| MouseNet Variant | Supervised | 39.37% | 33.02%; 26.53% |
| CPC of Bakhtiari et al. (2021) | Contrastive Predictive Coding | N/A | 36.28%; 29.67% |

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 self-supervised 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 1C and S3. Unless otherwise stated, each model is trained and validated on $64 \times 64$ pixels images. Note that the first four layers of AlexNet Instance Recognition, SimSiam, MoCov2, and SimCLR, form "Contrastive ShallowNet" (CSNets) for their respective contrastive objectives.

cluded 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 medium and high variation image subsets ("Var3" and "Var6", defined in Majaj et al., 2015) consisting of $3840$ images, and computed the performance (metrics defined below) on the test split of the medium and high variation sets ("Var3" and "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 $64 \times 64$ 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 $\alpha$ 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 $\alpha$ chosen from the range in (18).

**Texture Classification** 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 consisted 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.

## Data Availability

The calcium imaging and Neuropixels datasets were both obtained from the Allen Brain Atlas using the AllenSDK package (`https://allensdk.readthedocs.io/en/latest/index.html`).

## Code Availability

All code can be accessed by emailing the corresponding authors.

# 6 Supplemental Figures

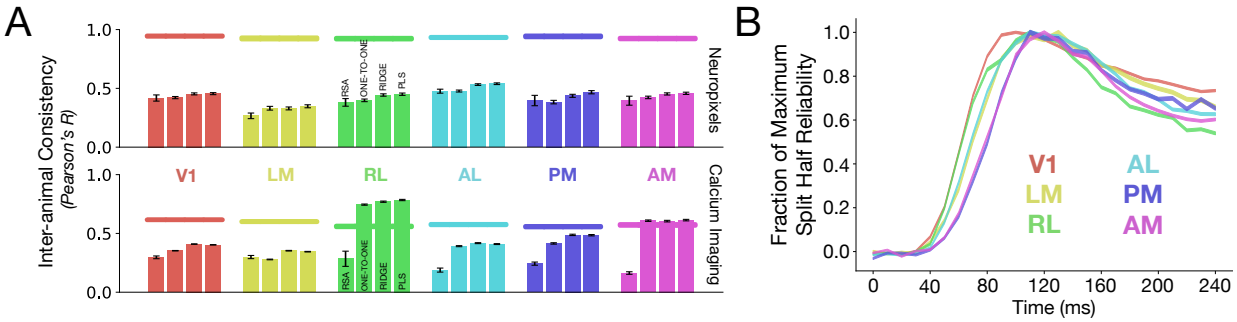


Figure S1: **Neural response consistency and functional hierarchy in mouse visual cortex. A**. Inter-animal consistency was computed using different linear maps, showing that PLS regression provides the highest consistency. Horizontal bars at the top are the median and s.e.m. of the internal consistencies of the neurons in each visual area. Refer to Table S1 for $N$ units per visual area. **B**. The fraction of maximum split-half reliability is plotted as a function of time (in 10-ms time bins) for each visual area. The time-series show that the visual areas can be grouped into roughly three to four modules based on the timing of the responses.

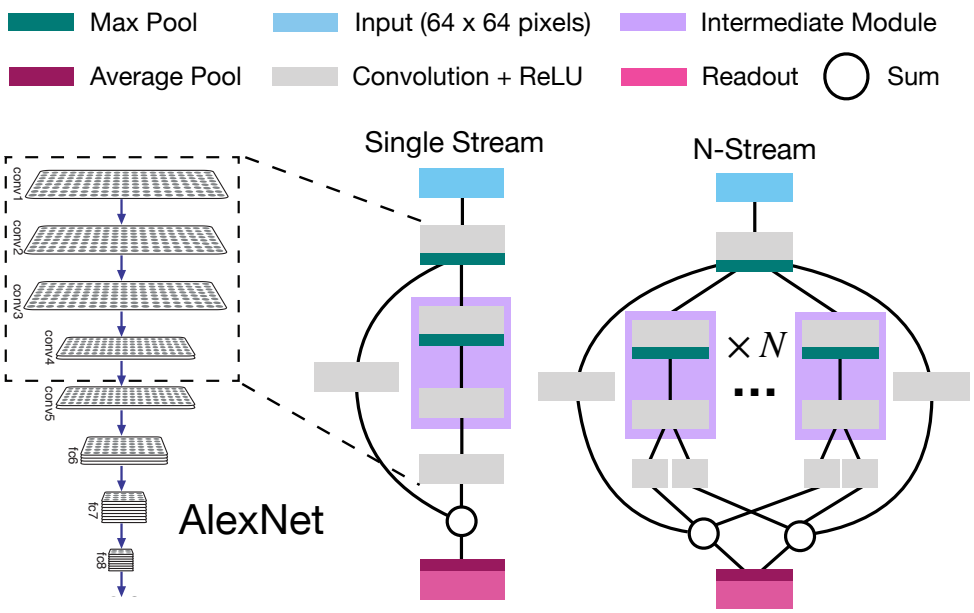


Figure S2: **StreamNet architecture schematic.** The first four convolutional layers of AlexNet best corresponded to all the mouse visual areas) These convolutional layers were used as the basis for our StreamNet architecture variants. The number of parallel streams, $N$, was varied to be one (single-stream), two (dual-stream) or six (six-stream).

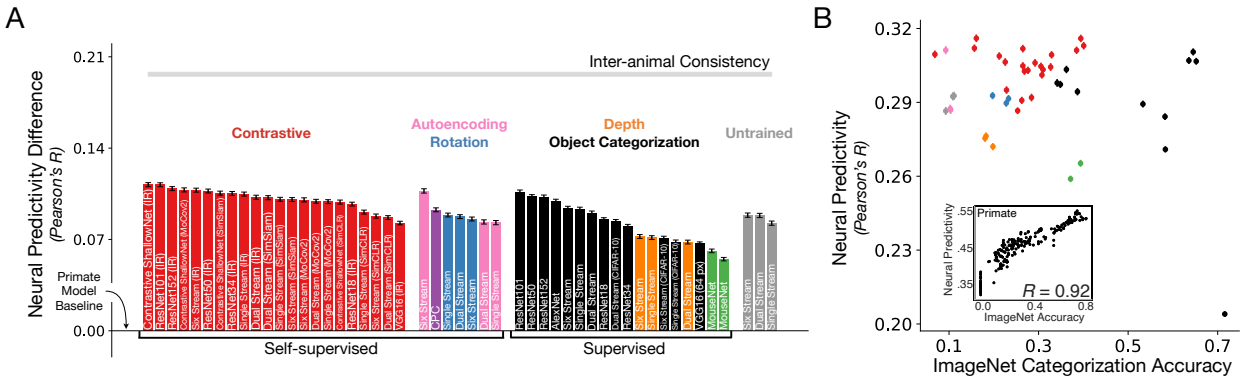


Figure S3: **Shallow architectures trained with contrastive objective functions yield the best matches to the neural data (calcium imaging dataset).** As in Figure 1, but for the calcium imaging dataset. **A**. The median and s.e.m. neural predictivity, using PLS regression, across neurons in all mouse visual areas except RL. $N = 16228$ units in total (RL is excluded, as mentioned in Section 5.1). 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. **Inset**. Primate ventral visual stream neural predictivity from BrainScore is correlated with ImageNet categorization accuracy (adapted from Schrimpf et al., 2018). All ImageNet performance numbers can be found in Table S3. Color scheme as in **A** and Figure 1A.

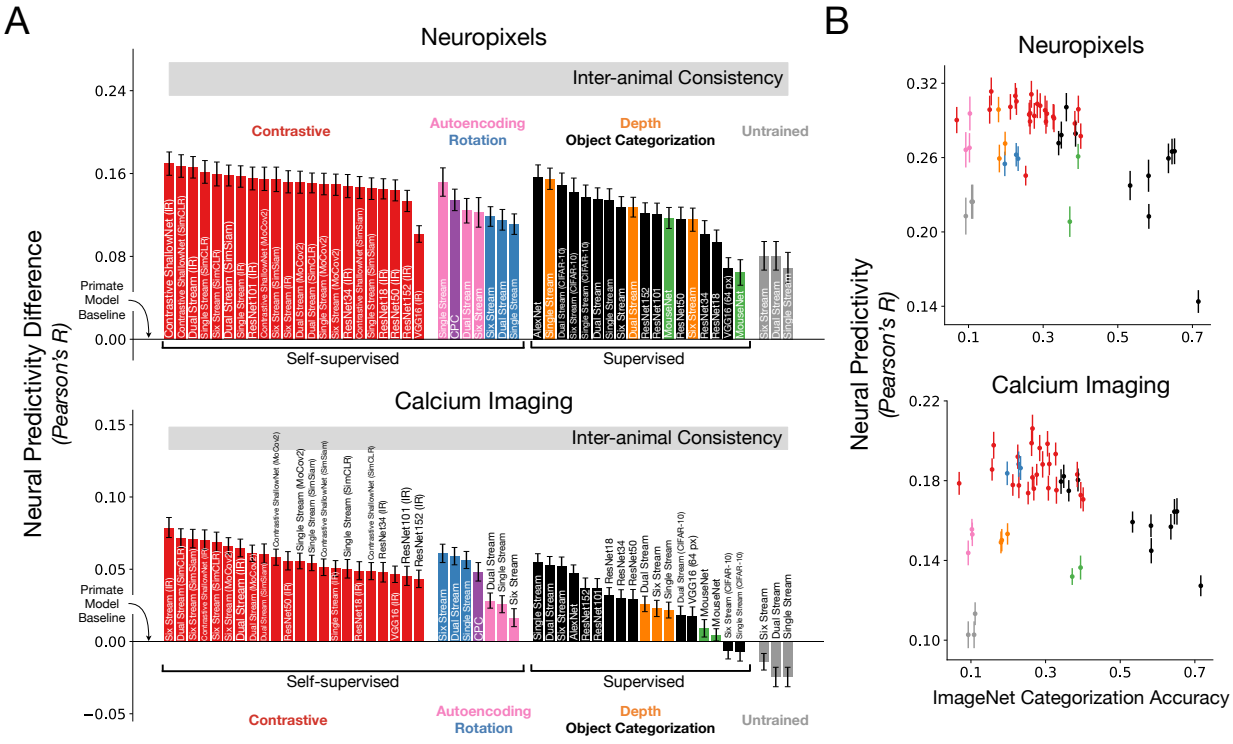


Figure S4: **Shallow architectures trained with contrastive objective functions yield the best matches to the neural data (RSA). A**. The median and s.e.m. noise-corrected neural predictivity, using RSA, across $N = 39$ and $N = 90$ animals for the Neuropixels and calcium imaging dataset respectively (across all visual areas, with RL excluded for the calcium imaging dataset, as mentioned in Section 5.1). **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** and Figure 1A.

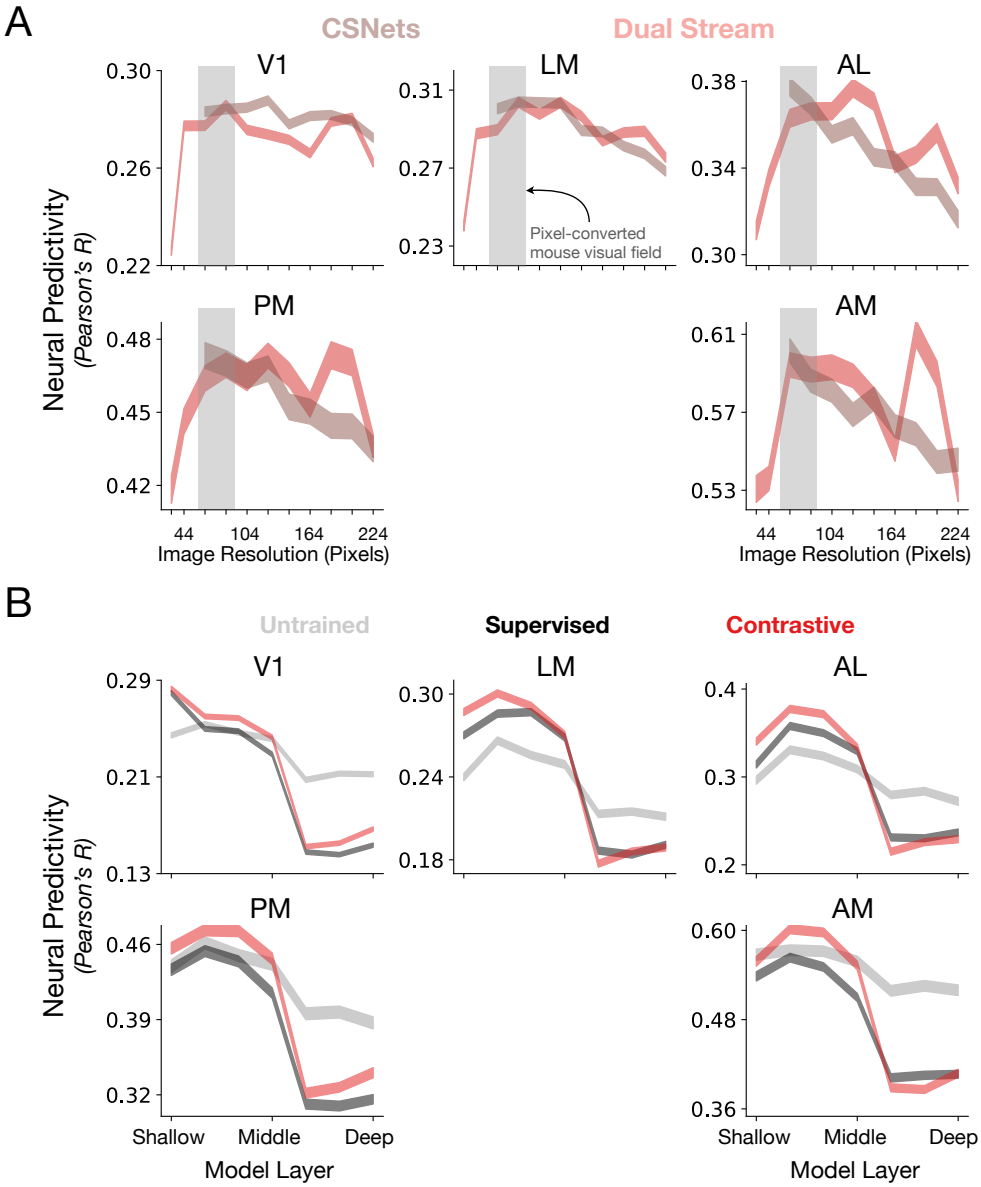


Figure S5: **Structural and functional factors leading to improved neural response predictivity (calcium imaging dataset). A**. As in Figure 2, our dual stream variant (red) and Contrastive ShallowNet (brown; "CSNet") were trained using lower resolution ImageNet images and in a contrastive manner. Each image was downsampled from $224 \times 224$ pixels, the image size typically used to train primate ventral stream models, to various image sizes. Training models on resolutions lower than $224 \times 224$ pixels generally led to improved neural predictivity. The median and s.e.m. across neurons in each visual area is reported. As mentioned in Section 5.1, visual area RL was removed from the calcium imaging neural predictivity results. Refer to Table S1 for $N$ units per visual area. **B**. As in Figure 1B, AlexNet was either untrained, trained in a supervised manner (ImageNet) or trained in an self-supervised manner (instance recognition). 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. As mentioned in Section 5.1, visual area RL was removed from the calcium imaging neural predictivity results.

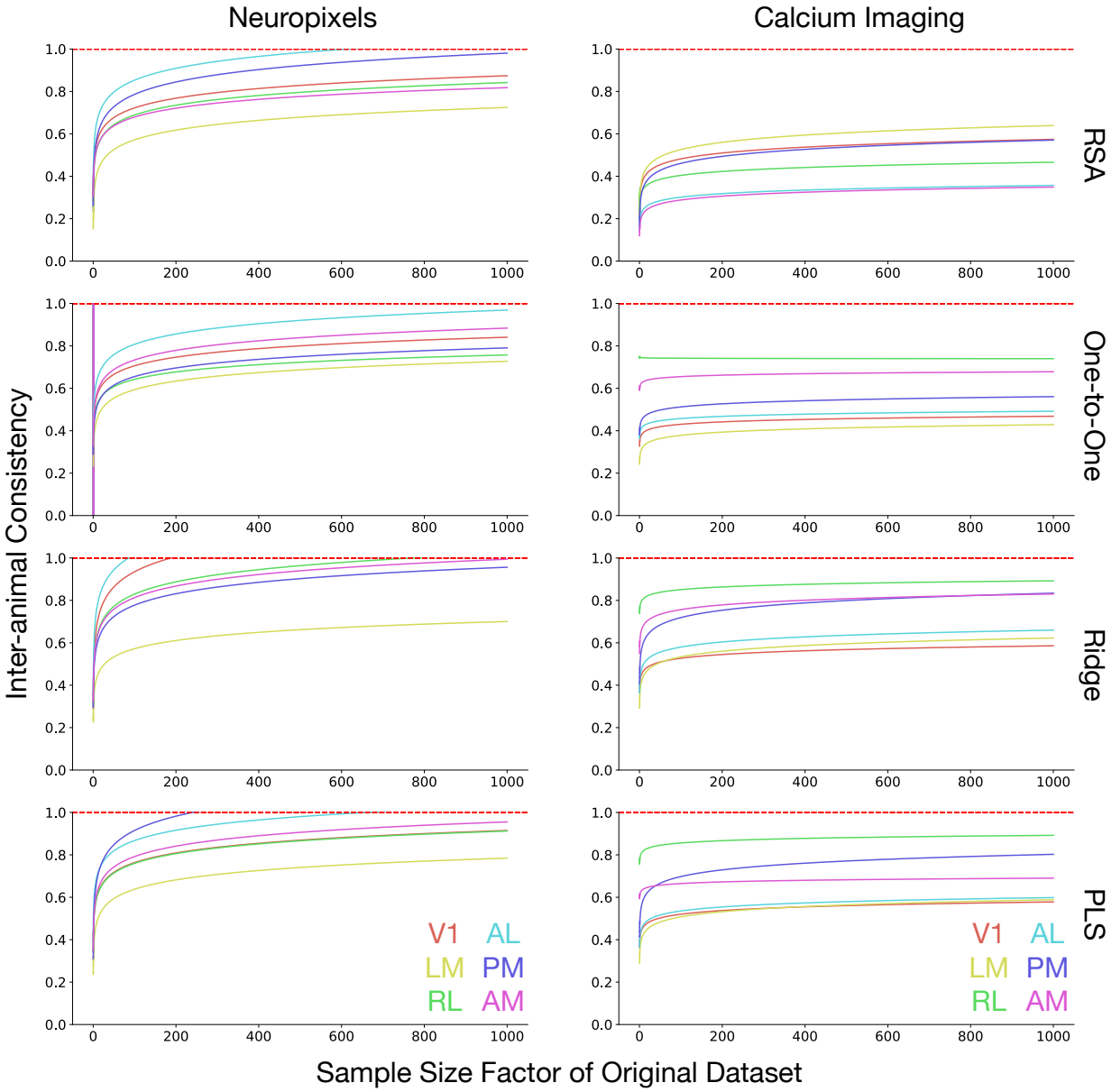


Figure S6: **Inter-animal consistency as a function of the number of units in the datasets for different mapping transforms.** Inter-animal consistency was extrapolated across the number of units in each dataset using a log-linear function ($f(n) = a \log_{10}(n) + b$, where $a$ and $b$ are fit as parameters via least squares, and $n$ is the sample size factor). This analysis reveals that inter-animal consistency of the Neuropixels dataset approaches 1.0 more rapidly than it does for the calcium imaging dataset. Inter-animal consistency evaluated at a sample size factor of one indicates the consistency when all the existing units in the datasets are used (i.e., inter-animal consistency values reported in Figure S1A).

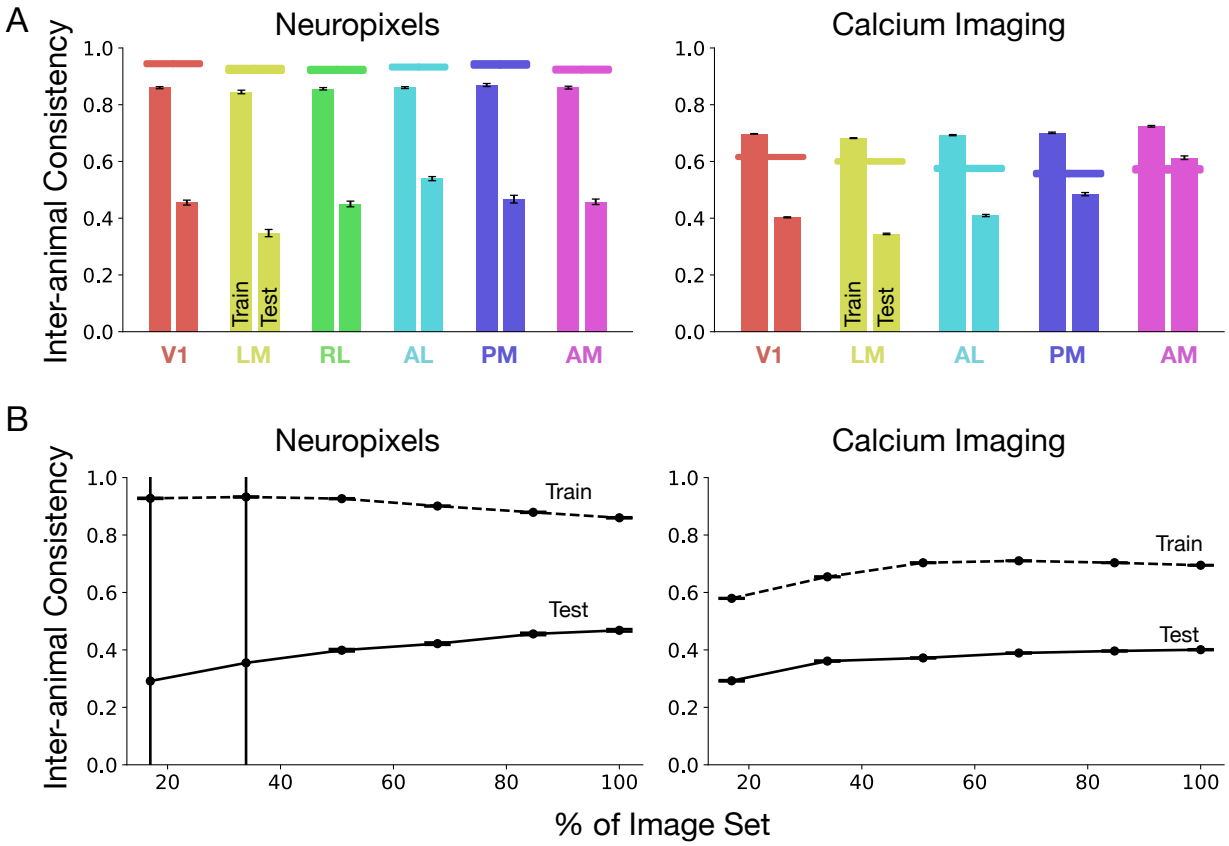


Figure S7: **Inter-animal consistency can increase with more stimuli. A**. Inter-animal consistency under PLS regression evaluated on the train set (left bars for each visual area) and test set (right bars for each visual area), for both Neuropixels and calcium imaging datasets. The horizontal lines 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 RL is excluded for calcium imaging, as explained in the text (Section 5.1). The median and s.e.m. across neurons is reported for both panels. Refer to Table S1 for $N$ units per visual area.

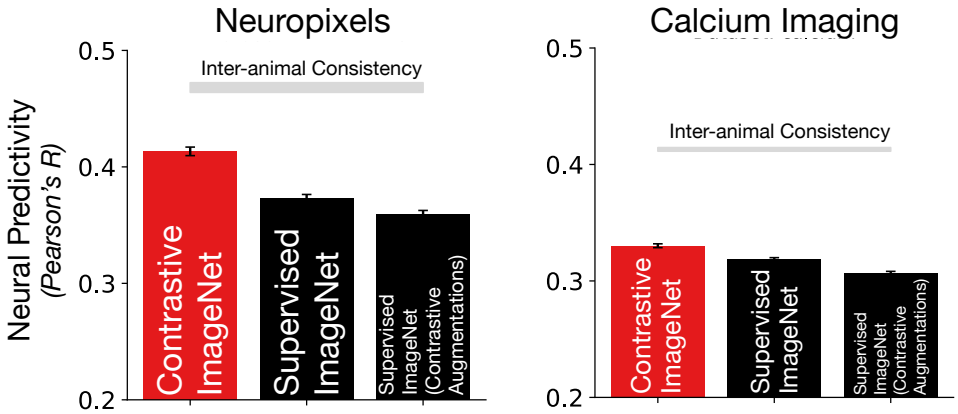


Figure S8: **Data augmentations alone do not lead to improved neural predictivity.** Here we compare the neural predictivity of AlexNet trained in three different ways. Contrastive ImageNet is an AlexNet trained using instance recognition on ImageNet with augmentations that are part of the contrastive algorithm (random crop, random color jitter, random grayscale, random horizontal flip). Supervised ImageNet is an AlexNet trained on ImageNet in a supervised manner with a smaller set of augmentations (random crop and random horizontal flip). Supervised ImageNet (contrastive augmentations) is an AlexNet trained on ImageNet in a supervised manner with the augmentations used in the instance recognition algorithm. This control model allows us to ascertain whether the improved neural predictivity of the Contrastive ImageNet model (red) is due to the contrastive loss function itself or due to the larger set of image augmentations used during model training. In both neural response datasets, we can conclude that data augmentations alone do not contribute to improved correspondence with the mouse visual areas.