Neural Regression, Representational Similarity, Model Zoology & Neural Taskonomy at Scale in Rodent Visual Cortex

Colin Conwell
Department of Psychology
Harvard University
conwell@g.harvard.edu

David Mayo
CSAIL & CBMM
MIT
dmayo2@csail.mit.edu

Michael A. Buice
Modeling, Analysis & Theory
Allen Institute for Brain Science
michaelbu@alleninstitute.org

Boris Katz
CSAIL & CBMM
MIT
boris@csail.mit.edu

George A. Alvarez
Department of Psychology
Harvard University
alvarez@wjh.harvard.edu

Andrei Barbu
CSAIL & CBMM
MIT
abarbu@csail.mit.edu

Abstract

How well do deep neural networks fare as models of mouse visual cortex? A majority of research to date suggests results far more mixed than those produced in the modeling of primate visual cortex. Here, we perform a large-scale benchmarking of dozens of deep neural network models in mouse visual cortex with multiple methods of comparison and multiple modes of verification. Using the Allen Brain Observatory's 2-photon calcium-imaging dataset of activity in over 59,000 rodent visual cortical neurons recorded in response to natural scenes, we replicate previous findings and resolve previous discrepancies, ultimately demonstrating that modern neural networks can in fact be used to explain activity in the mouse visual cortex to a more reasonable degree than previously suggested. Using our benchmark as an atlas, we offer preliminary answers to overarching questions about levels of analysis (e.g. do models that better predict the representations of individual neurons also predict representational geometry across neural populations?); questions about the properties of models that best predict the visual system overall (e.g. does training task or architecture matter more for augmenting predictive power?); and questions about the mapping between biological and artificial representations (e.g. are there differences in the kinds of deep feature spaces that predict neurons from primary versus posteromedial visual cortex?). Along the way, we introduce a novel, highly optimized neural regression method that achieves SOTA scores (with gains of up to 34%) on the publicly available benchmarks of primate BrainScore. Simultaneously, we benchmark a number of models (including vision transformers, MLP-Mixers, normalization free networks and Taskonomy encoders) outside the traditional circuit of convolutional object recognition. Taken together, our results provide a reference point for future ventures in the deep neural network modeling of mouse visual cortex, hinting at novel combinations of method, architecture, and task to more fully characterize the computational motifs of visual representation in a species so indispensable to neuroscience.

Preprint. Under review.

1 Introduction

To date, the most successful models of biological visual cortex are object-recognizing deep neural networks applied to the prediction of neural activity in primate visual cortex [1–5]. The predictive power of these models is so substantial that they can be used for synthesizing stimuli that drive neural activity beyond its typical range [6, 7]. The success of these models in predicting mouse visual cortex, however, has been a bit more modest. Later, fully connected layers seem to predict mouse visual cortex better than early convolutional layers — contrasting the traditional schematic wherein the simple and complex cells found in primate visual cortex [8] are best modeled by early convolutional and pooling layers [9]. Some have even suggested that randomly initialized networks are as predictive of rodent brains as trained networks, while still outperforming handcrafted features [10]. Here, we re-examine at large scale the state of neural network modeling in the visual cortices of mice, using many thousands of neurons, over 110 distinct neural network models, and two methods of mapping models to brain. We summarize this survey in four main results:

1. Training matters. The randomly initialized variants of some convolutional architectures fare well when predicting individual neural responses, but representational geometry is always better captured by features learned in service of some task. (Segmentation seems best.)
2. Features of intermediate complexity dominate in the prediction of all cortical sites, but there may be a slight upwards gradient in complexity from primary visual cortex onwards.
3. Taskonomic tools that have previously been shown to approximate functional organization in primates fail to strongly differentiate anatomical regions in mice, with the same kinds of tasks dominant across multiple, distinct neural sites.
4. When aggregated in similar ways, representational similarity and neural regression methods capture similar trends in the kinds of feature spaces that best predict the biology.

2 Methods

2.1 Datasets

For neural data, we use the Allen Brain Observatory Visual Coding[1] dataset [11] collected with two-photon calcium-imaging from the visual cortex of 256 awake adult transgenic mice and consisting of approximately 59,610 unique, individual neurons. Calcium-imaging fluorescence patterns are preprocessed and deconvolved by the Allen Institute[2]. The neurons sampled include neurons from 6 visual cortical areas at 4 cortical depths across 12 genetic cre lines. The visual experiments recorded activity for both artificial images (e.g., diffraction gratings) and 118 natural scenes. We analyze only the latter to ensure comparable inputs to what is typically used in the training of deep nets. Each natural scene is displayed 50 times over the course of an assay.

To ensure an optimal signal to noise ratio, we perform a significant amount of subsetting on the full neural population, beginning by subsetting only excitatory neurons. Recent analyses suggest neural activity throughout mouse visual cortex is often impacted by extraneous, external body movements [12]. For this reason, we subsequently filter out any neurons whose peak responses to the presentation of natural scene images are significantly modulated by the mouse's running speed, using an ANOVA metric provided by the Allen Institute. We further subselect neurons by assessing their split-half reliability across trials (with each split-half constituting 25 of 50 presentations for each image), keeping only those neurons exhibiting 0.8 reliability and above. This thresholding still leaves 6619 neurons for analysis, is in line with prior work on primates, and supports, for example, the construction of cortical representational dissimilarity matrices (RDMs) with split-half reliabilities as high as 0.93. (More details on the relationship between our metrics and neural reliability, including visualizations of some of our results across many degrees of thresholding, can be found in the appendix.)

2.2 Model Zoology

To explore the influence of model architecture on predictive performance, we use 26 model architectures from the Torchvision (PyTorch) model zoo [13] and 66 model architectures from the

[1] Available with a non-commercial license under the Allen Institute terms of use: http://www.alleninstitute.org/legal/terms-use/

[2] More details are available in the whitepapers released with the observatory data: http://observatory.brain-map.org/visualcoding/transgenic

Timm [14] model zoo [15–49]. These models include convolutional networks, vision transformers, normalization-free networks and MLP-Mixer models; all models are feed-forward. For each of these models, we extract the features from one trained and one randomly initialized variant (using whatever initialization scheme the model authors deemed best) so as to better disentangle what training on object recognition affords us in terms of predictive power.

2.3 Neural Taskonomy

Model zoology provides decent perspective on the computations related to object recognition, but the responsibilities of the visual cortex extend far beyond identifying the category of an object. To probe a wider range of tasks, we turn to Taskonomy: a single architecture trained on 24 different common computer vision tasks [50], ranging from autoencoding to edge detection. The model weights we use are from updated PyTorch implementations of the original Tensorflow models [51]. Key to the engineering of Taskonomy is the use of an encoder-decoder design in which only the construction of the decoder varies across tasks. While recent analyses using a similar approach in human visual cortex with fMRI data [52] have tended to focus only on the latent space of each task's encoder, we choose to extract representations across all layers, better situating Taskonomy within the same empirical paradigm that has so far defined the modeling of object recognition in the primate brain. For further clarity, we cluster the 24 tasks according to their 'Taskonomic' category — a total of 5 clusters (2D, 3D, semantic, geometric or other) that we further collapse into 4 clusters (lumping the only member of the 'other' category — a denoising autoencoder — in with its closest cousin — a vanilla autoencoder in the '2D' category). These purely data-driven clusters are derived from estimates of how effectively a set of features learned for one task transfer to (or boost the performance in) another task [50]. Use of the Taskonomy models provides a unique opportunity to test variance in training regimes without the confound of simultaneous changes in architecture.

2.4 Comparing Representations across Biological & Artificial Networks

Two methods predominate in the comparison of neural recordings to deep neural networks: at the most abstract level, one of these compares representational geometries computed across the activations of many individual neurons [53, 54]; the other attempts to predict the activity of individual neurons directly [54, 55]. Both of these techniques are grounded in the use of image-computable models and a shared stimulus set, but differ in the types of transformation applied to the neural activity generated by those stimuli. Given the difference in both target (neural populations versus individual neurons) and transforms (correlation matrices versus dimensionality reduction) we attempt a variant of each type of analysis here, comparing the two directly on the exact same neural data, with the same models and the same stimulus set, and in a granular, layer-by-layer fashion. A more comprehensive review of neural mapping methods is provided in the appendix.

2.4.1 Representational Similarity Analysis

To compare the representational geometries of a given model to the representational geometries of the brain, we begin by computing classic representational dissimilarity matrices (RDMs) [56]. We compute RDMs by correlating the activations of all neurons in a given neural site to the 118 images in the stimulus set, extracting only the upper triangle from the resultant matrices (one for each of the 6 cortical areas surveyed). We compute RDMs for the artificial networks in similar fashion, aggregating the responses of the artificial neurons in a given layer, before aggregating them once more into a correlation matrix. We then measure the relationship between the RDMs computed from the biological and artificial networks with a second-order Pearson correlation between the flattened upper triangles of each. The resultant coefficient constitutes the score for how well a given model layer predicts the representational geometry of a given cortical area.

2.4.2 Neural Regression (Encoding Models)

To more directly compare the biological and artificial neural activations in our data, we use a style of regression made popular in the modeling of primate visual cortex, epitomized by BrainScore [4]. Variants of this approach abound, but most consist of extracting model activations, performing dimensionality reduction, and then some form of cross-validated penalized or principal components regression. The dimensionality-reduced feature spaces of the model are used as the regressors of the activation patterns in a given neuron. After testing a number of these variants, we settled on sparse random projection for dimensionality reduction (which proved far more computationally efficient

than standard PCA, without sacrifice in terms of regression scores), followed by ridge regression (in place of the more frequently used partial least squares regression).

The details of our method (programmed with [57]) are as follows: Given a network, we first extract a predetermined number of sparse random projections (4096, in this case) from each layer — in line with the Johnson-Lindenstrauss lemma for the number of observations (images shown to the mice) in our data set. After extracting the random projections from the activations in each individual model layer, we regress these projections on each individual neuron using ridge regression (with a lambda penalty of 1.0). The use of a penalized regression in this case allows us to monopolize generalized cross-validation (a linear algebraic form of leave-one-out cross-validation), yielding a set of predictions for the activity of each neuron for each image[3]. We then compute the Pearson correlation between the predicted and actual activity for each neuron to obtain a score per neuron per model layer, which we then aggregate by taking the mean of neural scores across cortical area.

2.5 Model Rankings

To rank the models according to how well they predict the variance in a given cortical area, we take the max across layers. In effect, this requires that a model 'commit' only one layer to the prediction of each area. In the case of our neural regression metric we call these scores the 'SRP-Ridge Max'; in the case of our representational similarity metric we call these scores the 'RSA Max'. A final mean taken over the SRP-Ridge Max and RSA Max scores per model per cortical area yields our overall model ranking.

2.6 Non-Neural Network Baselines

Prior to the ascendancy of neural network models, a significant amount of time and craft was invested in the hand-engineering of features to simultaneously facilitate image recognition and capture meaningful subsets of neural variance. In this work, we test how well a small subset of those features are able to explain the variance in rodent visual cortex, using both our neural encoding and representational similarity metrics. Our non-neural network baselines consist of random fourier features [58] (computed specifically to match the dimensionality of our neural network predictors), handcrafted gabor filters and GIST (spatial envelope) descriptors [59].

3 Results

3.1 Does our neural regression method work?

To ascertain whether our novel neural regression method works, we first verify its efficacy on a known benchmark: the activity of 256 cells in the V4 and IT regions of two Rhesus macaque monkeys, a core component of BrainScore [4]. BrainScore's in-house method involves a combination of principal components analysis (for dimensionality reduction) and k-fold cross-validated partial least squares regression (for the linear mapping of model to brain activity). Here, we exchange principal components analysis for sparse random projection and partial least squares regression for ridge regression with generalized cross-validation. We compute the scores for each benchmark in the same fashion as BrainScore: as the Pearson correlation coefficient between the actual and predicted (cross-validated) activity of the biological neurons in the V4 and IT samples.

Taking for example a standard AlexNet architecture, our neural regression method yields gains of 16.5% (from 0.550 to 0.641) & 16.9% (from 0.508 to 0.593) on reported scores for V4 and IT, respectively. Across 6 other Torchvision architectures we tested with scores posted on the BrainScore leaderboard, our method yields gains on average of 13% for V4 and 23% for IT, and at its best yields a gain of 34% for SqueezeNet1-0 predicting IT. We consider this a strong validation of our neural regression method, which is both less computationally expensive, far faster and (to the extent that the generalized cross validation represents the optimal approximation of how well the mappings fit to our models might generalize to novel biological samples) more accurate than the combination of PCA and PLS. (More detailed results and speed tests may be found in the appendix.)

[3]The use of generalized cross-validation is particularly beneficial in datasets with fewer probe images, where k-fold cross-validation means losing a significant degree of information in each fit.

3.2 How do trained models compare to randomly initialized models?

Previous work in the deep neural network modeling of mouse visual cortex found that a randomly initialized VGG16 predicted neural responses as well, if not slightly better than, a VGG16 trained on ImageNet [10], suggesting that the neural predictivity of the features produced by a trained object recognition model are perhaps no better than the features produced by a randomly initialized one. Our results, on the other hand, suggest that the neural predictivity of trained versus randomly initialized models more generally depends on both the particular model being tested and the particular method used to produce the mappings between model and brain.

At the level of individual neurons (neural regression), 17 of the 92 model architectures we tested had randomly initialized variants that either matched or outperformed their ImageNet-trained counterparts. Replicating previous findings, we found these 17 architectures to include VGG16, as well as all 3 other VGG variants (11, 13 & 19), AlexNet, the DenseNet architectures (121, 169, 201), and almost all of the normalization-free architectures. Despite this, a paired student's t-test of the scores across *all* models demonstrates that ImageNet-trained architectures are overall more performant than their randomly initialized counterparts ($p = 1.29 \times 10^{-40}$, Cohen's $d = 2.40$). At the level of representational geometry, ImageNet-trained models categorically outperform their randomly initialized counterparts, and by a large margin ($p = 2.29 \times 10^{-29}$, Cohen's $d = 3.45$).

Taken together, these results strongly affirm that *training matters*, and that randomly initialized features can only go so far in the prediction of meaningful neural variance. Differences between ImageNet-trained and randomly initialized models are shown in Figure 1.

3.3 What kinds of architectures best predict rodent visual cortex?

The overall best architecture for predicting mouse visual cortex across both individual neurons (SRP-Ridge) and representational geometries (RSA) was an Inception-ResNet hybrid (Inception-ResNet-V2). There is a small, positive correlation between the depth of a model (the number of distinct layers) for both the RSA-Max metric and SRP-Ridge metric (Spearman $r = 0.22, p = 0.001$ and $r = 0.192, p = 0.007$, respectively), and a small, negative correlation for the total number of trainable parameters in the RSA Max metric (Spearman $r = -0.18, p = 0.007$). The latter of these is most likely driven by the relatively poor performance of parameter-dense architectures like VGG.

Strikingly, trends previously noted in macaques [60] fail to materialize here. In particular, models with higher top-1 accuracies on ImageNet do not perform significantly better than models with lower top-1 accuracies. This relative parity is driven in large part it seems by newer models like EfficientNets, which across the board have dominant scores on ImageNet, but sometimes middling or poor scores in the predictions of rodent visual cortex we've tabulated here.

Compared to all other architecturess, transformers on average fare slightly worse in the RSA Max metric (student's $t = -3.96, p = 0.004$, Hedge's $g = -0.913$), but moderately better in the SRP-Ridge Max metric (student's $t = 2.45, p = 0.023$, Hedge's $g = 0.633$). Other contrasts (including between models with and without residual connections or normalization) are available in the appendix.

Strikingly, transformers and MLP-Mixers boast the largest differences between ImageNet-trained and randomly initialized variants in the SRP-Ridge Max metric, with all pairwise t-tests significant at $alpha = 0.05$ after Bonferroni correction for multiple comparisons. This strongly suggests that the advantage of those randomly initialized variants that matched or outperformed their ImageNet-trained counterparts is an advantage conferred by properties of convolutional architectures (e.g., translation invariance), and not necessarily an advantage shared across random feature spaces writ large. The rankings of these and other architectures may be found in Figure 1.

3.4 What kinds of tasks best predict rodent visual cortex?

The overall best Taskonomy encoder across both the RSA and SRP-Ridge Max is 2D segmentation (ranking second and first respectively; see Figure 2). At the level of individual neurons (SRP-Ridge), 2D tasks (keypoints, autoencoding, inpainting) dominate. At the level of representational geometry (RSA), all 2D tasks but 2D segmentation fall to the bottom of the rankings, and Semantic tasks (object recognition and semantic segmentation) rise to 2nd and 3rd place.

This reshifting in rank presents a curious case for interpretation, suggesting most likely that while the representations of individual neurons may be coordinated more by the lower level, less abstract features necessary for performing well on most 2D tasks, the overall neural population codes are

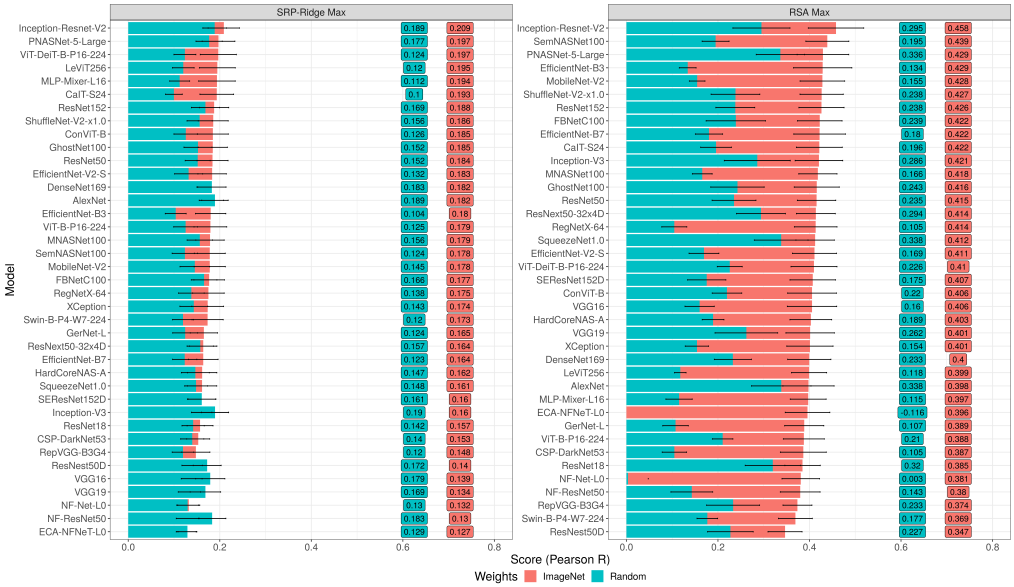


Figure 1: Rankings across 40 of 92 model architectures (deliberately sampled to demonstrate variety), sorted by the scores of the ImageNet-trained variants (red) of each. Instances in which the randomly initialized variants (blue) outperform their ImageNet-trained counterparts are visible in those rows where the blue entirely overlaps the red. Error bars are 95% bootstrapped confidence intervals across the 6 cortical areas. As in [10], some randomly initialized and ImageNet-trained models like VGG16 have equal scores in neural regression, but ImageNet-trained models categorically dominate in representational similarity. The full set of 92 rankings is available in the appendix.

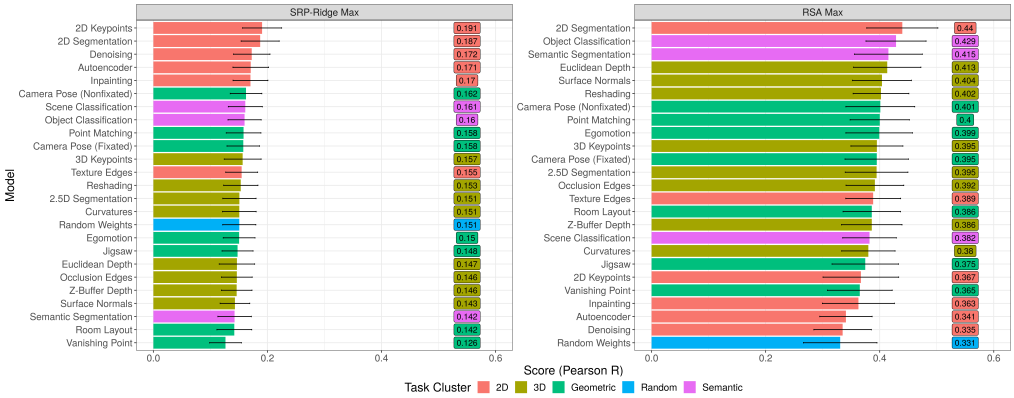


Figure 2: Rankings across Taskonomy encoders, color coded by the Taskonomic cluster to which each belongs. Notice the contrast between the SRP-Ridge Max (which favors 2D tasks) and RSA Max metric (which favors Semantic tasks), but also the relative rank of 2D Segmentation in both.

coordinated more by the parsing of the visual input into ethologically and spatially relevant units via the segmentation and classification tasks. Notably, the original research from which these PyTorch models were adopted offers an auxiliary data point that may anchor this interpretation more concretely. The top 3 models in our RSA Max metric (2D segmentation, object classification, semantic segmentation) are likewise in the top 5 of a ranking the original researchers produced by pitting the Taskonomy encoders against one another as pretrained 'perceptual systems' for reinforcement learning agents learning to navigate a virtual environment (see [51], figure 13 in the appendix). This raises the possibility that the reason these models are optimally predicting the visual neural population code for mice is simply because that code is coordinated in service of navigation.

3.5 How well do non-neural network baselines predict rodent visual cortex?

Non-neural network baselines somewhat uniformly fail to predict cortical activity. We tested three baselines: 1) a bank of Gabor filters of applied to 8x8 grids of each image, 2) the PCs of the resultant feature matrices (i.e. the Gist descriptors [59])), 3) and the max across 600 iterations of 4096 random Fourier features (a dimensionality matching that of our SRPs). Ridge regressed with generalized cross-validation, these feature models score 0.032, 0.026 and -0.014, respectively.

3.6 Does training or architecture matter more for better prediction?

The range in scores between the best and worst performing model architecture trained on ImageNet is 0.209 to 0.121 (0.088) for the SRP-Ridge Max and 0.458 to -0.117 (0.575) for the RSA Max metric (excluding the two normalization-free architectures that produced negative scores and are otherwise significant outliers, the range is more like 0.458 to 0.347 (0.111)); the range between the best and worst performing model in Taskonomy is 0.190 to 0.126 (0.064) for the SRP-Ridge Max and 0.440 to 0.331 (0.109) for RSA Max. These results point ambiguously in the direction of architecture as mattering slightly more, but it seems perfectly possible that novel combinations of architecture and task could push this range further than modifications to either in isolation.

3.7 How 'deep' are the layers that best predict rodent visual cortex?

Across all ImageNet-trained architectures, regardless of metric, the features most predictive of rodent visual cortex are found about a third of the way into the model (see the peaks in the lowess curves in Figure 4). These early to intermediate visual features go beyond basic edge detection but are far from the highly abstracted representations adjacent to final fully connected layers. Across Taskonomy encoders, 2D & Geometric tasks yield their best features in earlier layers; 3D & Semantic tasks yield their best features in more intermediate and later layers. (These motifs vary only slightly by cortical area, a result we discuss in the next section and in Figure 3.)

3.8 Are there differences in model predictions across cortical area?

Here, we attempt to answer this question from two perspectives: that of function and that of hierarchy.

Research into primate visual cortex over the last two decades has unveiled a significant degree of functional (not just anatomical) organization [61–63], with distinct subregions defined in large part by their activity in response to different kinds of stimuli. To replicate this in mouse visual cortex we search for Taskonomic organization, a proxy of functional organization wherein distinct neural sites are better or worse predicted by the features from different taskonomy encoders. Curiously, and in contrast to previous findings in human fMRI [52], it seems to be the case that the scores of different Taskonomic clusters are relatively consistent across cortical area; see Figure 3 (left).) This suggests that mouse visual cortex may be more functionally (or Taskonomically) homogenous than primate visual cortex, with anatomical descriptors providing little to no cue of functional difference. (For breakdowns of Taskonomy scores across genetic cre line instead of cortical area, see the appendix). Another alternative is that the tasks of computer vision are just not so neatly mapped onto the tasks of biological vision in mice.

Another possible difference across cortical area is one of hierarchy. In primate visual cortex, there seems to be a distinct information processing hierarchy along the ventral visual stream [64–66], with posterior sites like V1 and V3 defined by features like oriented edge detectors, and more anterior sites like V4 and IT defined by more complex morphologies. While there continues to be some debate as to whether a similar hierarchy exists in rodent visual cortex, the relative depth at which our models optimally predict cortical activity does unveil at least a preliminary gradient. For details, see Figure 3.

3.9 How do the predictions compare across RSA and neural regression?

While prior work has attempted to answer this question theoretically [67], it's rarely the case that representational similarity and neural regression are compared directly and empirically. Here, we compare our RSA and SRP-Ridge metric both at the level of overall rankings (taking the max across layers) and at the level of individual layers, the latter of which provides a much more detailed assessment of how different feature spaces map to cortical representation.

In terms of overall rankings, the Spearman rank order correlation between the two methods is either 0.56 ($p = 8.36 \times 10^{-19}$) or 0.59 ($p = 3.17 \times 10^{-12}$), depending on whether you include or exclude

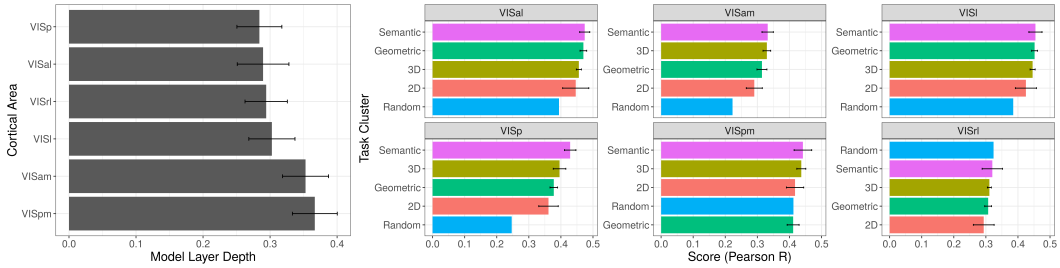


Figure 3: The average depth (from 0 to 1) of layers from ImageNet-trained architectures that score maximally on different cortical areas (left) and breakdowns across cortical area of the scores on Taskonomy (right). Displayed here, for simplicity's sake, are only the results from the RSA (SRP-Ridge versions are available in the appendix). The plot on the left demonstrates that there does indeed seem to be a hierarchy that loosely follows the anatomy of the rodent visual system, with earlier layers predicting the primary visual area (VISp) and later layers predicting anteromedial and posteromedial visual areas (VISam & VISpm). The plot on the right shows that there don't seem to be major 'Taskonomic' dissociations across cortical area. Semantic tasks dominate in all but VISrl where, surprisingly, the randomly initialized version of Taskonomy is ascendant — though we caveat this by noting VISrl boasts the overall lowest scores of any of the cortical areas.

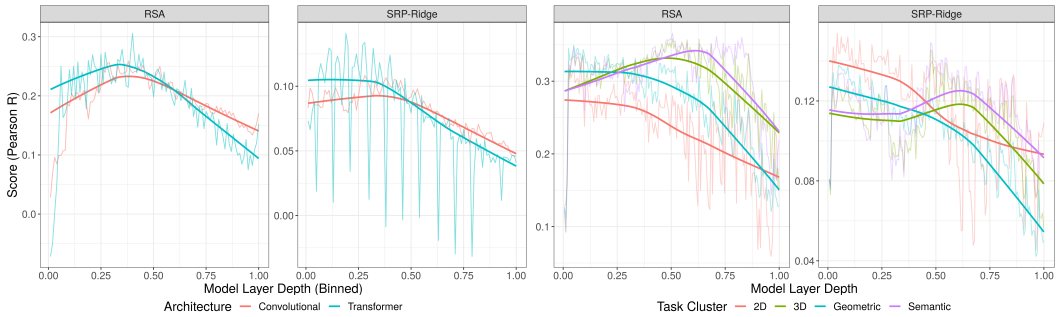


Figure 4: Aggregate layer by layer comparisons across ImageNet-trained models (left) and Taskonomy models (right). The horizontal axis shows relative model layer depth, from the first to the last layer. Because the models on the left vary significantly in size, we discretize the relative depths into bins of size 0.01 (100 bins). The jagged, semitransparent lines are the mean scores at a given depth. (These are particularly jagged for transformers due to the heterogeneous nature of the computational blocks, which engenders large troughs in predictive power). The opaque, smoothed trend lines are the outputs of locally weighted linear regressions (lowess) computed at each point with a span of 2/3 the width of the axis. Plots show that covolutional networks and transformers have similar predictive power at similar depths despite their significant computational differences. (right) Features most predictive of cortical responses across the Taskonomy models vary in their relative depth increasing in later layers. The reversal in the ranks of 2D models between RSA Max and SRP Ridge Max is seen as a vertical shift in the intercept of the curve relating depth to score – though note that the slope and shape of the curve remain similar across metrics. 2D models are most predictive in the early layers, but are nevertheless superseded by other tasks in the RSA Max.

the randomly initialized architectures. In terms of layer by layer comparisons, we decompose the Spearman correlation across distinct combinations of model and cortical area. The average coefficient between the two methods, along with bootstrapped 95% confidence intervals is 0.468 [0.447,0.489] or 0.626 [0.613,0.639], again depending on the inclusion or exclusion of the random models. This suggests a significant degree of overlap between the kinds of features that optimally predict the representations of both individual neurons and neural populations. Of course, the averages here obscure some meaningful subtrends and idiosyncrasies; see Figure 4. Also note that while the overall trends are similar, the scores for the neural encoding method are significantly lower than scores for the representational similarity analysis. With more probe images, we expect that the parametric linear mapping would be more competitive with the nonparameteric distances of representational similarity.

4 Discussion

Our intent with this work was to provide a preliminary atlas for further ventures into the deep neural network modeling of rodent visual cortex. To this end, we have deliberately invested in introspective analyses of the tools we used, as well as the curation of deep neural networks we hope will provide informative waypoints. Obviously, the atlas is far from complete. Newer model classes like self-supervised models, recurrent models [68, 69], equivariant models [70], and models used in robotics (e.g. for visual odometry [71]) are promising candidates for inclusion in future benchmarks, and our neural encoding & representational similarity metrics are just two of many variants.

Nevertheless, the results we have presented here indicate that neural recordings from the visual brains of mice can be compared to neural networks in much the same way as recordings from the visual brains of monkeys. Having as reference two animal models that occupy very different ecological niches and are separated by tens of million years of evolution makes it far more likely that insights into vision gleaned across both are truly fundamental to the process of making perceptual sense of the world — and are not just some idiosyncratic quirk specific to any one evolutionary trajectory. Primate and rodent vision differ significantly even in fairly basic ways: mice lack a fovea, have a retina dominated by rods for vision under low light, and spatial acuity less than 20/1000 [72], making their primary visual system more akin to the primate peripheral system. It is possible that mice rely on vision as a sort of broad bandpass filter for lower-frequency, dynamic stimuli that the mouse can then flee, fight or further investigate with its whiskers — perhaps its most sophisticated sensory organ (and also, it seems, a compelling candidate for neural network modeling [73]).

The dominance in our Taskonomy results of 2D segmentation, object recognition and semantic segmentation (all tasks that have elsewhere been shown to provide fertile, transferable features for the simulation of robotic navigation) underscores the indispensable point that perceptual systems should always be considered in service of behavior. The unparalleled access, resolution, and control afforded by rodent neuroimaging have already revolutionized our understanding of the relationship between perceptual representation and behavioral output. Combined with novel methods like the embedding of neural networks in virtual agents [74] in ecologically realistic environments, this kind of data may well provide a testbed for better situating the tasks of computer vision in the broader behavioral context of agentic scene understanding.

Perhaps the most immediately pressing future direction of this work is manifest in the simple reality that even with our most predictive models, we've explained only a fraction of the highly reliable neural variance this dataset has to offer. One of the best models in any neural site is Taskonomy's 2D Segmentation in the anterolateral visual area (VISal), with an RSA Max score of 0.538, translated to an R^2 of 0.289. This constitutes approximately a third of the total explainable variance in this site, defined in this case simply as the square of the neural RDM's split-half reliability, $R^2 = 0.773$. Only novel combinations of architecture, task and methodology will close this gap and more fully model the rich diversity and fiendish complexity of biological brains at scale — even the very smallest ones.

References

[1] 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.

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

[3] 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, 2019.

[4] Martin Schrimpf, Jonas Kubilius, Ha Hong, Najib J. Majaj, Rishi Rajalingham, Elias B. Issa, Kohitij Kar, Pouya Bashivan, Jonathan Prescott-Roy, Franziska Geiger, Kailyn Schmidt, Daniel L. K. Yamins, and James J. DiCarlo. Brain-score: Which artificial neural network for object recognition is most brain-like? *bioRxiv preprint*, 2018.

[5] Rishi Rajalingham, E.B. Issa, P. Bashivan, K. Kar, K. Schmidt, and J.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. *The Journal of Neuroscience*, 38(33): 7255–7269. doi: 10.1523/JNEUROSCI.0388-18.2018. URL https://doi.org/10.1523/JNEUROSCI.0388-18.2018.

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

[7] Edgar Y Walker, Fabian H Sinz, Erick Cobos, Taliah Muhammad, Emmanouil Froudarakis, Paul G Fahey, Alexander S Ecker, Jacob Reimer, Xaq Pitkow, and Andreas S Tolias. Inception loops discover what excites neurons most using deep predictive models. *Nature neuroscience*, 22(12):2060–2065, 2019.

[8] David H Hubel and Torsten N Wiesel. Receptive fields and functional architecture of monkey striate cortex. *The Journal of physiology*, 195(1):215–243, 1968.

[9] Jianghong Shi, Eric Shea-Brown, and Michael Buice. Comparison against task driven artificial neural networks reveals functional properties in mouse visual cortex. In *Advances in Neural Information Processing Systems*, pages 5765–5775, 2019.

[10] S. A. Cadena, F. H. Sinz, T. Muhammad, E. Froudarakis, E. Cobos, E. Y. Walker, J. Reimer, M. Bethge, A. Tolias, and A. S. Ecker. How well do deep neural networks trained on object recognition characterize the mouse visual system? 2019.

[11] 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.

[12] Simon Musall, Matthew T Kaufman, Ashley L Juavinett, Steven Gluf, and Anne K Churchland. Single-trial neural dynamics are dominated by richly varied movements. *Nature neuroscience*, 22(10):1677–1686, 2019.

[13] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[14] Ross Wightman. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019.

[15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[16] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Conference on Computer Vision and Pattern Recognition*, pages 4700–4708, 2017.

[17] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[18] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V Le. Mnasnet: Platform-aware neural architecture search for mobile. In *Conference on Computer Vision and Pattern Recognition*, pages 2820–2828, 2019.

[19] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[21] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Conference on Computer Vision and Pattern Recognition*, pages 1492–1500, 2017.

[22] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Conference on Computer Vision and Pattern Recognition*, pages 6848–6856, 2018.

[23] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[24] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[25] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

[26] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection, 2020.

[27] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning, 2016.

[28] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks, 2020.

[29] Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation, 2019.

[30] Qilong Wang, Banggu Wu, Pengfei Zhu, Peihua Li, Wangmeng Zuo, and Qinghua Hu. Eca-net: Efficient channel attention for deep convolutional neural networks, 2020.

[31] Bichen Wu, Xiaoliang Dai, Peizhao Zhang, Yanghan Wang, Fei Sun, Yiming Wu, Yuandong Tian, Peter Vajda, Yangqing Jia, and Kurt Keutzer. Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search, 2019.

[32] Mingxing Tan and Quoc V. Le. Mixconv: Mixed depthwise convolutional kernels, 2019.

[33] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2020.

[34] Xiang Li, Wenhai Wang, Xiaolin Hu, and Jian Yang. Selective kernel networks, 2019.

[35] Chenxi Liu, Barret Zoph, Maxim Neumann, Jonathon Shlens, Wei Hua, Li-Jia Li, Li Fei-Fei, Alan Yuille, Jonathan Huang, and Kevin Murphy. Progressive neural architecture search, 2018.

[36] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. Squeeze-and-excitation networks, 2019.

[37] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows, 2021.

[38] François Chollet. Xception: Deep learning with depthwise separable convolutions, 2017.

[39] Niv Nayman, Yonathan Aflalo, Asaf Noy, and Lihi Zelnik-Manor. Hardcore-nas: Hard constrained differentiable neural architecture search, 2021.

[40] Stéphane d'Ascoli, Hugo Touvron, Matthew Leavitt, Ari Morcos, Giulio Biroli, and Levent Sagun. Convit: Improving vision transformers with soft convolutional inductive biases, 2021.

[41] Weijian Xu, Yifan Xu, Tyler Chang, and Zhuowen Tu. Co-scale conv-attentional image transformers, 2021.

[42] Kai Han, Yunhe Wang, Qi Tian, Jianyuan Guo, Chunjing Xu, and Chang Xu. Ghostnet: More features from cheap operations, 2020.

[43] Ben Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. Levit: a vision transformer in convnet's clothing for faster inference, 2021.

[44] Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, and Alexey Dosovitskiy. Mlp-mixer: An all-mlp architecture for vision, 2021.

[45] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, and Hartwig Adam. Searching for mobilenetv3, 2019.

[46] Andrew Brock, Soham De, Samuel L. Smith, and Karen Simonyan. High-performance large-scale image recognition without normalization, 2021.

[47] Andrew Brock, Soham De, and Samuel L. Smith. Characterizing signal propagation to close the performance gap in unnormalized resnets, 2021.

[48] Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, and Jian Sun. Repvgg: Making vgg-style convnets great again, 2021.

[49] Hugo Touvron, Piotr Bojanowski, Mathilde Caron, Matthieu Cord, Alaaeldin El-Nouby, Edouard Grave, Armand Joulin, Gabriel Synnaeve, Jakob Verbeek, and Hervé Jégou. Resmlp: Feedforward networks for image classification with data-efficient training, 2021.

[50] Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3712–3722, 2018.

[51] Alexander Sax, Jeffrey O Zhang, Bradley Emi, Amir Zamir, Silvio Savarese, Leonidas Guibas, and Jitendra Malik. Learning to navigate using mid-level visual priors. *arXiv preprint arXiv:1912.11121*, 2019.

[52] Aria Wang, Michael Tarr, and Leila Wehbe. Neural taskonomy: Inferring the similarity of task-derived representations from brain activity. In *Advances in Neural Information Processing Systems*, pages 15475–15485, 2019.

[53] Nikolaus Kriegeskorte, Marieke Mur, Douglas A Ruff, Roozbeh Kiani, Jerzy Bodurka, Hossein Esteky, Keiji Tanaka, and Peter A Bandettini. Matching categorical object representations in inferior temporal cortex of man and monkey. *Neuron*, 60(6):1126–1141, 2008.

[54] Umut Güçlü and Marcel AJ van Gerven. Increasingly complex representations of natural movies across the dorsal stream are shared between subjects. *NeuroImage*, 145:329–336, 2017.

[55] Matteo Carandini, Jonathan B Demb, Valerio Mante, David J Tolhurst, Yang Dan, Bruno A Olshausen, Jack L Gallant, and Nicole C Rust. Do we know what the early visual system does? *Journal of Neuroscience*, 25(46):10577–10597, 2005.

[56] N. Kriegeskorte, M. Mur, D.A. Ruff, R. Kiani, J. Bodurka, H. Esteky, and P.A. Bandettini. Matching categorical object representations in inferior temporal cortex of man and monkey. *Neuron*, 60(6):1126–1141. doi: 10.1016/j.neuron.2008.10.043. URL https://doi.org/10.1016/j.neuron.2008.10.043.

[57] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[58] Ali Rahimi, Benjamin Recht, et al. Random features for large-scale kernel machines. In *NIPS*, volume 3, page 5. Citeseer, 2007.

[59] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision*, 42(3):145–175, 2001.

[60] Kamila Maria Jozwik, Martin Schrimpf, Nancy Kanwisher, and James J DiCarlo. To find better neural network models of human vision, find better neural network models of primate vision. *BioRxiv*, page 688390, 2019.

[61] Nancy Kanwisher. Functional specificity in the human brain: a window into the functional architecture of the mind. *Proceedings of the National Academy of Sciences*, 107(25):11163–11170, 2010.

[62] Kalanit Grill-Spector and Kevin S Weiner. The functional architecture of the ventral temporal cortex and its role in categorization. *Nature Reviews Neuroscience*, 15(8):536–548, 2014.

[63] Dwight J Kravitz, Kadharbatcha S Saleem, Chris I Baker, and Mortimer Mishkin. A new neural framework for visuospatial processing. *Nature Reviews Neuroscience*, 12(4):217–230, 2011.

[64] D.J. Felleman and D.C. Van Essen. Distributed hierarchical processing in the primate cerebral cortex. *Cerebral Cortex*, 1(1):1–47.

[65] J.J. DiCarlo, D. Zoccolan, and N.C. Rust. How does the brain solve visual object recognition? *Neuron*, 73(3):415–434. doi: 10.1016/j.neuron.2012.01.010. URL https://doi.org/10.1016/j.neuron.2012.01.010.

[66] Nicole C Rust and James J DiCarlo. Selectivity and tolerance ("invariance") both increase as visual information propagates from cortical area v4 to it. *Journal of Neuroscience*, 30(39):12978–12995, 2010.

[67] Jörn Diedrichsen and Nikolaus Kriegeskorte. Representational models: A common framework for understanding encoding, pattern-component, and representational-similarity analysis. *PLoS computational biology*, 13(4):e1005508, 2017.

[68] K. Kar, J. Kubilius, K. Schmidt, E.B. Issa, and J.J. DiCarlo. Evidence that recurrent circuits are critical to the ventral stream's execution of core object recognition behavior. *Nature Neuroscience*. doi: 10.1038/s41593-019-0392-5. URL https://doi.org/10.1038/s41593-019-0392-5.

[69] Jonas Kubilius, Martin Schrimpf, Ha Hong, Najib J. Majaj, Rishi Rajalingham, Elias B. Issa, Kohitij Kar, Pouya Bashivan, Jonathan Prescott-Roy, Kailyn Schmidt, Aran Nayebi, Daniel Bear, Daniel L. K. Yamins, and James J. DiCarlo. Brain-Like Object Recognition with High-Performing Shallow Recurrent ANNs. In H. Wallach, H. Larochelle, A. Beygelzimer, F. D'Alché-Buc, E. Fox, and R. Garnett, editors, *Neural Information Processing Systems (NeurIPS)*, pages 12785—-12796. Curran Associates, Inc., 2019. URL http://papers.nips.cc/paper/9441-brain-like-object-recognition-with-high-performing-shallow-recurrent-anns.

[70] Ivan Ustyuzhaninov, Santiago A Cadena, Emmanouil Froudarakis, Paul G Fahey, Edgar Y Walker, Erick Cobos, Jacob Reimer, Fabian H Sinz, Andreas S Tolias, Matthias Bethge, et al. Rotation-invariant clustering of neuronal responses in primary visual cortex. In *International Conference on Learning Representations*, 2019.

[71] Sen Wang, Ronald Clark, Hongkai Wen, and Niki Trigoni. Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks. *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017. doi: 10.1109/icra.2017.7989236. URL http://dx.doi.org/10.1109/ICRA.2017.7989236.

[72] GT Prusky and RM Douglas. Characterization of mouse cortical spatial vision. *Vision research*, 44(28):3411–3418, 2004.

[73] Chengxu Zhuang, Jonas Kubilius, Mitra JZ Hartmann, and Daniel L Yamins. Toward goal-driven neural network models for the rodent whisker-trigeminal system. In *Advances in Neural Information Processing Systems*, pages 2555–2565, 2017.

[74] Josh Merel, Diego Aldarondo, Jesse Marshall, Yuval Tassa, Greg Wayne, and Bence Ölveczky. Deep neuroethology of a virtual rodent. *arXiv preprint arXiv:1911.09451*, 2019.

[75] Seyed-Mahdi Khaligh-Razavi and Nikolaus Kriegeskorte. Deep supervised, but not unsupervised, models may explain it cortical representation. *PLoS computational biology*, 10(11), 2014.

[76] 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.

[77] David A Klindt, Alexander S Ecker, Thomas Euler, and Matthias Bethge. Neural system identification for large populations separating" what" and" where". *arXiv preprint arXiv:1711.02653*, 2017.

[78] Ari Morcos, Maithra Raghu, and Samy Bengio. Insights on representational similarity in neural networks with canonical correlation. In *Advances in Neural Information Processing Systems*, pages 5727–5736, 2018.

[79] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. *arXiv preprint arXiv:1905.00414*, 2019.

[80] Natalia Y Bilenko and Jack L Gallant. Pyrcca: regularized kernel canonical correlation analysis in python and its applications to neuroimaging. *Frontiers in neuroinformatics*, 10:49, 2016.

[81] Ryan M Rifkin and Ross A Lippert. Notes on regularized least squares. 2007.

[82] Leyla Tarhan and Talia Konkle. Reliability-based voxel selection. *NeuroImage*, 207:116350, 2020.

A Appendix

A.1 Code for Reproducibility

Code for the replication of our analysis may be found at this anonymous GitHub repository: https://github.com/annonymousecode/Model-Zoology-and-Neural-Taskonomy (License GPL v2)

A.2 Compute Required

We used a single machine with 8 Nvidia RTX 3090 GPUs, 755gb of RAM, and 96 CPUs. GPUs were used only for extracting model activations, and could (without major slowdown) be removed from the analytic pipeline. Dimensionality reduction and regression computations were CPU and RAM intensive. Replicating all of our results would take approximately two weeks on a similar machine.

A.3 Ethics Statement

Lest our science forget the life that powers it, we must note that behind the phenomenal dataset provided by the Allen Institute are 256 laboratory mice, each of which was subjected to multiple surgeries, a highly invasive neuroimaging technique and genetic engineering. The moral parameters of this particular praxis of neuroscience are contentious, and not without reason. While we believe centralized, comprehensive and (most importantly) public datasets like those provided by the Allen Institute may actually decrease the total number of laboratory animals required for similar kinds of empirical projects, we acknowledge with solemnity the cost to life required.

A.4 Deeper Dive: How do our neural benchmarking methods compare to others?

In the main analysis, we roughly group existing methods for comparing the responses of deep neural networks to neural responses recorded from brain tissue into two categories: neural regression and representational similarity analysis. In reality, this division is often not so neatly dichotomous. Some of the first brain-to-network comparisons availed themselves of both these methods simultaneously; Yamins et al. [1] citing Carandini et al. [55] and Kriegeskorte et al. [53] used linear regression for mapping responses in individual neural sites and representational similarity analysis for populations. Other seminal work comparing deep nets to primate visual cortex pioneered distinctive variants of each. Güçlü and van Gerven [54] employed regression in the form of encoding models to assess the hierarchical correspondence between earlier and later layers of processing in vivo and silico. Khaligh-Razavi and Kriegeskorte [75] built representational dissimilarity matrices by "remixing" and "reweighting" model features according to their performance in a support vector machine classifier trained on major categorical divisions in the stimulus set. Zhuang et al. [76] citing Klindt et al. [77] uses a form of masked regression to better account for spatial information (e.g. properties of the receptive field) in the target feature spaces. In the context specifically of comparisons to rodent neurophysiology, Cadena et al. [10]'s neural encoding method predicts spike rate with a core feature model (VGG16) in tandem with a "shifter" network and "modular" network that correct for extraneous influences on recorded brain activity (including eye movements and running speed). A possible third strain of methods that doesn't fit so neatly into the binary of regression versus representational similarity are canonical correlation and alignment methods. These techniques leverage what is often assumed to be an underlying latent space of similarity shared across divergent high-dimensional datasets to assess (via projection) the shared variance between them. Canonical correlation and alignment methods are popular in both the machine learning [78, 79] and neuroimaging communities [80], but have so far been applied mostly to comparison within, rather than across, domains and neural substrates. The relative advantages of these various approaches as they pertain to characterizing the representational structure of biological brains is largely uncertain, with a comprehensive comparison of techniques on the same dataset seemingly absent from the literature.

The current standard for high-throughput benchmarking of neural data on neural models is perhaps that of Schrimpf et al. [4] in BrainScore, a method that consists of a partial least squares (PLS) regression fit individually to each neural site (in their case, a cluster of neurons around a given electrode in a microarray), wherein the regressand is the responses from that site and the regressors are the principal components of a target model's feature space. The end product of this process is a Pearson correlation coefficient (unadjusted or reliability-corrected) quantifying the relationship between actual neural activity and the neural activity predicted by the linear mapping. While effective, this combination of principal components analysis and partial least squares regression tends to be a computationally expensive process – often prohibitively so in the absence of cloud or cluster

computing. The final approach we use in the primary manuscript is a more computationally efficient version of this process. The reasoning behind the particular neural regression we use (assessing the tradeoff between accuracy and computational traction) may be found in the section below.

A.5 How do different neural regression methods trade off in terms of speed & accuracy?

Given the many variants of neural regression used in the analysis of the human and nonhuman primate brain (and to a lesser extent the rodent brain), we experimented with a number of possible approaches before settling on the one detailed in the primary manuscript. Attempting to directly mirror the approach described in Schrimpf et al. [4], we began with a method combining principal components analysis for dimensionality reduction with partial least squares regression for neural prediction. So as to capture more dimensions of variance in a given model's feature spaces, and not 'double dip' meaningful dimensions of variance with the regression to follow, Brain-Score computes a set of principal components on the features from an auxiliary set of held-out ImageNet images, then extracts the loadings of the features from the target stimulus set on these same components. These loadings are subsequently made the regressors in a partial least squares regression of 25 components, with a given neural site (the activity from a microelectrode array) as the regressand. The most computationally intensive step of this process is the calculation of the PCA on the features from the auxiliary Imagenet images – requiring in larger models like VGG16 upwards of 450GB of RAM for a single layer. The prohibitively large expense of this PCA prompted us to search both for alternative dimensionality reduction techniques, as well as for the possibility of extracting fewer total dimensions with whatever technique we chose. A summary of the outputs of this search (using only a small subset of models and a fraction of our total neural data) may be found in Figure 5.

What this search made clear (at least in the context of our specific neural data and stimulus set) is that approaches involving PCA were doubly suboptimal, taking orders of magnitude longer to compute, and actually costing a nontrivial portion of score. In the PLS regressions as well, it quickly became clear that 10 components yielded scores comparable (if not equivalent) with those of twice or thrice as many components, suggesting nothing was to be gained from more components apart from longer compute times. Finally, while somewhat less definitive than the 2 previous points, our tests did suggest that using 4096 sparse random projections was roughly comparable to using 8192 sparse random projections – and translated to about 0.75 of the compute time per projection. When in an additional test a ridge regression performed in roughly 0.29 seconds what a PLS regression with 10 components performed in roughly 1.31 seconds (representing an over 200% percent gain in speed, and consistent across 10 iterations), while producing only a 0.004 difference in average R^2, we abandoned PLS regression entirely in favor of ridge regression. Switching to ridge regression meant we could also make use of generalized cross-validation [57, 81], cutting the time required for k-fold cross-validation from roughly 1.05 seconds to 0.25 seconds. While it is most certainly the case that these tests were not comprehensive enough in terms of models or neural data to cover the full range of contingencies and idiosyncrasies of analysis, we felt the empirical justification for the use of a sparse random projection and ridge regression approach was sufficient. In future work, we intend to expand our testing regimen to see if the empirical advantage holds in a wider range of cases.

A.6 How does reliability thresholding impact our benchmark scores?

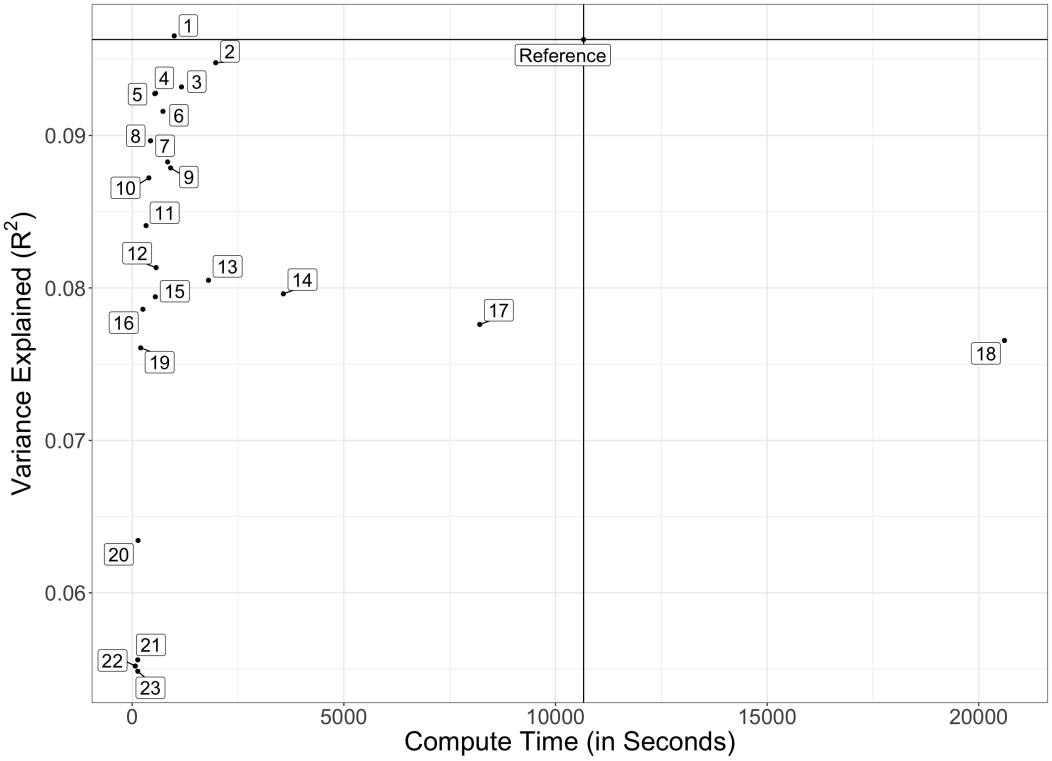
In the main analysis, we subselected from the greater pool of available neurons only those neurons with split-half reliabilities of 0.8 and above. In Figure 6, we show the impact of different degrees of thresholding on the scores for a majority of our models.

A.7 Addendum: Does our neural regression method work?

In the main analysis, we offer a few examples of the scores our method obtains in the BrainScore benchmarks of macaque V4 and IT. Reproduced in Figure 7 are the scores for the full subset of models we tested.

A.8 Addendum: What kinds of architectures best predict rodent visual cortex?

In the main analysis, the rankings of the various architectures were subsampled for illustration. Reproduced in Figure 8 are the scores for all models.



PLS Regression with 25 Components and Different Dimensionality-Reduced Regressors

ID / Rank	Dimensionality Reduction Technique Applied to the Regressors
Reference	No Dimensionality Reduction; All Features Included
1	1:8192 SRPs calculated directly on the Stimulus Set
2	1:8192 Feature SRPs calculated with 9216 ImageNet Images
3	1:4096 Feature SRPs calculated with 9216 ImageNet Images
4	1:4096 SRPs calculated directly on the Stimulus Set
5	SRPs corresponding to Embedding Quality of 0.1
6	2048 Randomly Selected Features
7	1:2048 Feature SRPs calculated with 9216 ImageNet Images
8	4096 Randomly Selected Features
9	1:1024 Feature PCs calculated with 1024 ImageNet Images
10	1:2048 SRPs calculated directly on the Stimulus Set
11	2048 Randomly Selected Features
12	1:1024 Feature SRPs calculated with 9216 ImageNet Images
13	1:1024 Feature PCs calculated with 9216 ImageNet Images
14	1:2048 Feature PCs calculated with 9216 ImageNet Images
15	1:1024 Feature SRPs calculated with 1024 ImageNet Images
16	1:1024 SRPs calculated directly on the Stimulus Set
17	1:4096 Feature PCs calculated with 9216 ImageNet Images
18	1:8192 Feature PCs calculated with 9216 ImageNet Images
19	SRPs corresponding to Embedding Quality of 0.25
20	SRPs corresponding to Embedding Quality of 0.5
21	SRPs corresponding to Embedding Quality of 0.75
22	All PCs calculated directly on the Stimulus Set
23	SRPs corresponding to Embedding Quality of 0.99

Figure 5: Variance explained versus compute time for a PLS Regression with 25 components and different types of dimensionality reduction techniques applied to the regressors. Descriptions of the dimensionality reduction steps associated with each data point are provided in the table above. The vertical and horizontal ablines triangulate a reference point for all methods: that is, a regression in which all features from a given model layer are used simultaneously without dimensionality reduction. Notice, that at least one PCA-based method (18) takes longer than this full regression.

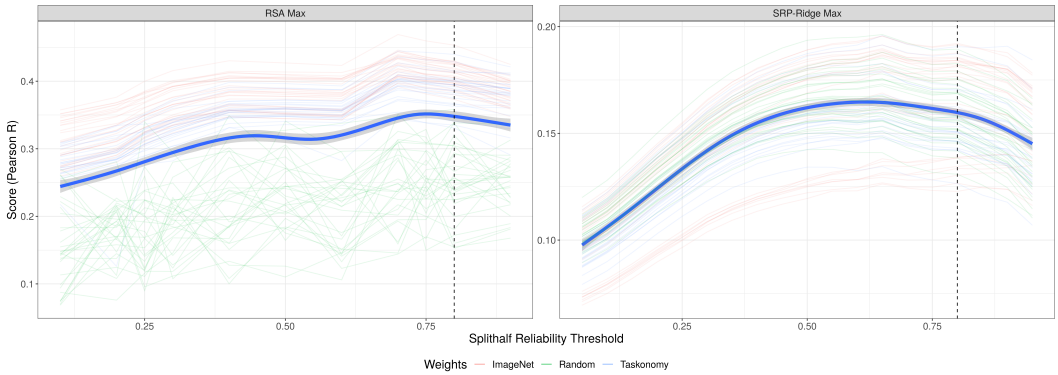


Figure 6: Scores for a variety of models with both the RSA Max (left) and SRP-Ridge Max (right) metrics at different levels of reliability thresholding. The jagged, semitransparent lines are the scores for individual models. The smooth, opaque line is the output of a generalized additive smoother fit across all models. (Error bars are bootstrapped 95% confidence interval across models). The dotted vertical line is the threshold we use in the main analysis. Based on these results, one might argue a more performative threshold would have been closer to 0.7 or 0.75. In the future, we plan to more closely emulate methods designed to derive the optimal threshold empirically (e.g. reliability-based voxel selection in human fMRI [82]).

Model	Weights	V4	IT
alexnet	ImageNet	0.6413	0.5939
densenet121	ImageNet	0.6555	0.6298
googlenet	ImageNet	0.6551	0.6278
mnasnet0_5	ImageNet	0.6554	0.6228
mnasnet1_0	ImageNet	0.6632	0.6398
mobilenet_v2	ImageNet	0.6592	0.6295
resnet101	ImageNet	0.6690	0.6193
resnet18	ImageNet	0.6492	0.6233
resnet34	ImageNet	0.6560	0.6282
resnet50	ImageNet	0.6649	0.6216
resnext50_32x4d	ImageNet	0.6474	0.6285
shufflenet_v2_x0_5	ImageNet	0.6530	0.6087
shufflenet_v2_x1_0	ImageNet	0.6538	0.6296
squeezenet1_0	ImageNet	0.6652	0.6157
squeezenet1_1	ImageNet	0.6602	0.6061
vgg11	ImageNet	0.6735	0.6202
vgg13	ImageNet	0.6748	0.6201
vgg16	ImageNet	0.6750	0.6260
vgg19	ImageNet	0.6733	0.6250
wide_resnet50_2	ImageNet	0.6606	0.6355

Figure 7: Scores for a subset of models tested on the macaque V4 & IT benchmarks of BrainScore. Corresponding scores for models in our set that overlap with models tested by the BrainScore team may be found at: https://www.brain-score.org/

Figure 8: The RSA Max and SRP-Ridge Max scores (with bootstrapped 95% confidence intervals) for all model architectures tested (both ImageNet-trained and randomly initialized variants).

A.9 Addendum: Are there differences in model predictions across cortical area?

Figure 3 in the main paper shows only the differences with respect to the RSA analysis. Here, in Figure 9 we show the differences with respect to the SRP-Ridge analysis.

A.10 Are there differences in model predictions across *genetic cre line*?

In the main analysis, we aggregate neurons by anatomical region (cortical area); another method of aggregating neurons is by genetic cre line. Aggregating in this way changes the overall focus of the benchmarking, from asking 'where' certain models fare best in predicting visual cortical activity to 'with what cell types'. The kinds of representational idiosyncrasies that characterize different cell types are beyond the scope of this paper. Nevertheless, as a sampler for those interested, aggregate Taskonomy scores across cre line are provided in Figure 10.

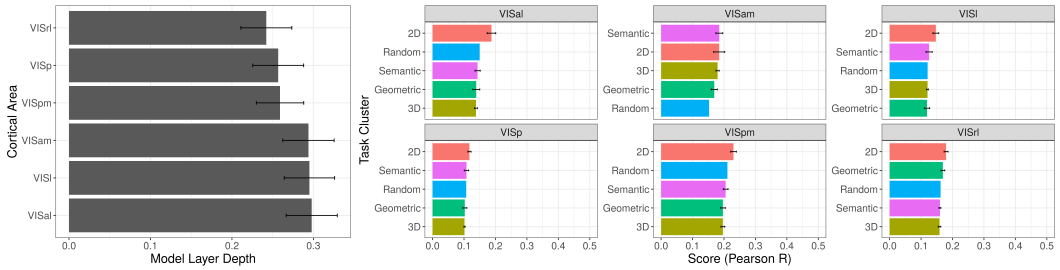


Figure 9: The average depth (from 0 to 1) of layers from ImageNet-trained architectures that score maximally on different cortical areas (left) and breakdowns across cortical area of the scores on Taskonomy (right). Displayed here are the results from the SRP-Ridge analysis. The hierarchy (left) is a bit less clear than in the case of the RSA analysis (though primary visual area (VISP) is still predicted by earlier layers and anteromedial visual area (VISam) is still predicted by later layers). As in the RSA analysis, there do not seem to be clear 'Taskonomic' dissociations across cortical area. 2D tasks dominate in all but VISam where Semantic tasks are tied almost exactly.

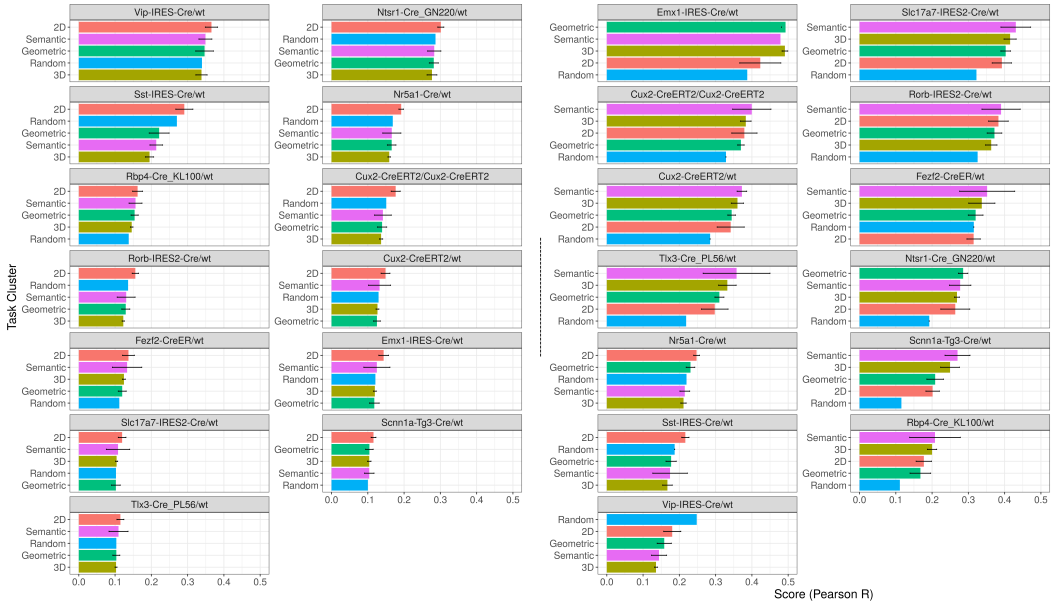


Figure 10: Taskonomy scores across genetic cre line. On the left is SRP-Ridge Max metric; on the right is the RSA Max metric. The facets are shown in descending order by the overall magnitude of their mean scores. Of note: Large-scale motifs present when aggregating across cortical area (such as the dominance of 2D models in SRP-Ridge; the dominance of semantic models in the RSA Max; and the lack of clear Taskonomic dissociations) are recapitulated across cre line. Nevertheless, some differences are salient. For example, the aggregate scores for certain cre lines using the SRP-Ridge Max metric are much higher on average than the scores obtained in any cortical area.

Area	Structure	Abbreviation
V1	Primary Visual Area	VISp
LM	Lateral Visual Area	VISl
AL	Anterolateral Visual Area	VISal
PM	Posteromedial Visual Area	VISpm
RL	Rostrolateral Visual Area	VISrl
AM	Anteromedial Visual Area	VISam

Figure 11: A glossary of areas in the mouse visual cortex.

Task	Cluster	Definition
Autoencoder	2D	Image compression and decompression
Object Classification	Semantic	1000-way object classification (knowledge distillation from ImageNet).
Scene Classification	Semantic	Scene Classification (knowledge distillation from MIT Places).
Curvatures	3D	Magnitude of 3D principal curvatures
Denoising	Other	Uncorrupted version of corrupted image.
Euclidean Depth	3D	Depth estimation
Z-Buffer Depth	3D	Depth estimation.
Occlusion Edges	3D	Edges which include parts of the scene.
Texture Edges	2D	Edges computed from RGB only (texture edges).
Egomotion	Geometric	Odometry (camera poses) given three input images.
Camera Pose (Fixated)	Geometric	Relative camera pose with matching optical centers.
Inpainting	2D	Filling in masked center of image.
Jigsaw	Geometric	Putting scrambled image pieces back together.
2D Keypoints	2D	Keypoint estimation from RGB-only (texture features).
3D Keypoints	3D	3D Keypoint estimation from underlying scene 3D.
Camera Pose (Nonfixated)	Geometric	Relative camera pose with distinct optical centers.
Surface Normals	Other	Pixel-wise surface normals.
Point Matching	Geometric	Classifying if centers of two images match or not.
Reshading	3D	Reshading with new lighting placed at camera location.
Room Layout	Geometric	Orientation and aspect ratio of cubic room layout.
Semantic Segmentation	Semantic	Pixel-wise semantic labeling (via knowledge distillation from MS COCO).
Unsupervised 2.5D Segmentation	3D	Segmentation (graph cut) on RGB-D-Normals-Curvature image.
Unsupervised 2D Segmentation	2D	Segmentation (graph cut) on RGB.
Vanishing Point	Geometric	Three Manhattan-world vanishing points.

Figure 12: Definitions of the tasks that the Taskonomy models are trained to perform.

A.11 Glossary of Visual Cortical Areas in Mouse Brain

Reproduced in Figure 11 is a glossary of visual cortical areas in the mouse brain. More information about the Allen Brain Observatory visual coding dataset may be found at their website: http://observatory.brain-map.org/visualcoding

A.12 Taskonomy Task Definitions

Reproduced in Figure 12 are Taskonomy's official definitions of its constituent tasks. Further information is available at their website: http://taskonomy.stanford.edu