Supporting Information

Yamins et al. 10.1073/pnas.1403112111

SI Text

Data Collection. We collected neural data, assessed human behavior, and tested models on a common image set. In this section, we discuss this image set and the data collection methods used.

Array electrophysiology. Neural data were collected in the visual cortex of two awake behaving rhesus macaques (*Macaca mulatta*, 7 and 9 kg) using parallel multielectrode array electrophysiology recording systems (Cerebus System; BlackRock Microsystems). All procedures were done in accordance with National Institute of Health guidelines and approved by the Massachusetts Institute of Technology (MIT) Committee on Animal Care guidelines. Six 96-electrode arrays (3 arrays each in two monkeys) were surgically implanted in anatomically determined V4, posterior inferior temporal (IT), central IT, and anterior IT regions (1). Of these, 296 neural sites (168 in IT and 128 in V4) were selected as being visually driven with a separate image set. Fixating animals were presented with testing images in pseudorandom order with image duration comparable to those in natural primate fixations (2). Images were presented one at a time on an LCD screen (SyncMaster 2233RZ at 120 Hz; Samsung) for 100 ms, occupying a central 8° visual angle radius on top of a gray background, followed by a 100-ms gray blank period with no image shown. Eye movements were monitored by a video tracking system (EyeLink II; SR Research), and animals were given a juice reward each time central fixation was maintained for six successive image presentations. Eye movement jitter within 2° from a 0.25° red dot at the center of screen was deemed acceptable, whereas presentations with large eye movements were discarded. In each experimental block, responses were recorded once for each image, resulting in 25–50 repeat recordings of the each testing image.

For each image repetition and electrode, scalar firing rates were obtained from spike trains by averaging spike counts in the period 70–170 ms after stimulus presentation, a measure of neural response that has recently been shown to match behavioral performance characteristics very closely (3). Background firing rate, defined as the mean within-block spike count for blank images, was subtracted from the raw response. Additionally, the signal was normalized such that its per-block variance is 1. Final neuron output responses were obtained for each image and site by averaging over image repetitions. Recordings took place daily over a period of several weeks, during which time neuronal selectivity patterns at each recording site were typically stable. Based on firing rates and spike-sorting analysis, we estimate that each individual electrode multiunit site in this study picks up potentials from one to three single neural units. To determine whether results would likely differ for direct single-unit recordings, we sorted single units from the multiunit IT data by using affinity propagation (4) together with the method described in ref. 5. Of these units, 21 had internal trial-to-trial consistency with an *r* value of 0.3. We assessed the hierarchical modular optimization (HMO) model's prediction ability for these single units, obtaining a median of $50.4 \pm 2.2\%$ explained variance, very close to that obtained directly from the multiunit data. Moreover, we supplemented with serially sampled, single-electrode recording (6,7) and found that neuronal populations from arrays have very similar patterns of image encoding as assembled single-electrode unit populations.

Test stimulus set. The test stimulus set (Fig. 1*A*) consisted of 5,760 images of 64 distinct objects chosen from one of eight categories (animals, boats, cars, chairs, faces, fruits, planes, tables), with eight specific exemplars of each category (e.g., BMW, Z3, Ford, etc., within the car category). The set was designed specifically to (*i*) include a range of everyday objects, (*ii*) support both coarse, basic-level category comparisons (e.g., animals vs. cars) and finer subordinate level distinctions (e.g., distinguish among specific cars) (8), and (*iii*) require strong tolerance to object viewpoint variation, e.g., pose, position, and size. Objects were placed on realistic background images, which were chosen randomly to prevent correlation between background content and object class identity.

Object view parameters were chosen randomly from uniform ranges at three levels of variation (low, medium, and high), and images were rendered using the photorealistic Povray package (9). The parameter ranges for the three variation levels were as follows:

i) Low variation: All objects placed at image center ($x = 0$, $y = 0$), with a constant scale factor ($s = 1$) translating to objects occluding 40% of image on longest axis, and held at a fixed reference pose ($rxy = rxz = ryz = 0$).

ii) Medium variation: Object position varies within one-half multiple of total object size ($|x|, |y| \leq 0.3$), varying in scale between $s = 1/1.3 \sim 0.77$ and $s = 1.3$, and between $-45°$ and $45°$ of in-plane and out-of-plane rotation ($\leq 45°$).

iii) High variation: Object position varies within one whole multiple of object size ($|x|, |y| \leq 0.6$), varying in scale between $s = 1/1.6 \sim 0.625$ and $s = 1.6$, and between $-90°$ and $90°$ of in-plane and out-of-plane rotation ($\leq 90°$).

Crowd-sourced human psychophysics. Data on human object recognition judgement abilities shown in Fig. 2*B* and Fig. S6 were obtained using Amazon's Mechanical Turk crowdsourcing platform, an online task marketplace where subjects can complete short work assignments for a small payment. A total of 104 observers participated in one of three visual task sets: an eight-way classification of images of eight different cars, an eight-way classification of images of eight different faces, or an eight-way categorization of images of objects from eight different basic-level categories. Observers completed these 30- to 45-min tasks through Amazons Mechanical Turk. All of the results were confirmed in the laboratory setting with controlled viewing conditions, and virtually identical results were obtained in the laboratory and web populations (Pearson $\text{correlation} = 0.94 \pm 0.01$). For the eight-way basic-level categorization task set, each human observer ($n = 29$) judged a subset of 400 randomly sampled images with blocks for each of the three variation levels (400 of 640 for low variation and 400 of 2,560 for medium and high variation levels). For the eight-way car ($n = 39$) and eight-way face ($n = 40$) identification task sets, each observer saw all 80 images at the low variation level and all 320 images at both medium and high variation levels. The presentation of images were randomized and counterbalanced so that the number of presentations of each class was the same in the given variation level. Each trial started with a central fixation point that lasted for 500 ms, after which an image appeared at the center of the screen for 100 ms; following a 300-ms delay, the observer was prompted to click one of eight response images that matched the identity or category of the stimulus image. Response images were shown from a fixed frontal viewpoint and remained constant throughout a trial block. All human studies were done in accordance with the MIT Committee on the Use of Humans as Experimental Subjects.

Performance was determined by computing accuracies for each task. For a given eight-way task set and variation level (e.g., high-variation basic-level categorization, medium-variation car subordinate identification, etc.), we constructed the raw 8×8 confusion matrix for each individual observer and computed the population confusion matrix summing raw confusion matrices across individuals.

From the population confusion matrix, we computed accuracy values for each task of recognizing one target class against seven distractor classes (a.k.a. binary task). We obtained 72 binary task accuracies by performing this procedure over all combinations of three task sets and three variation levels (3 task sets × 8 targets per task set × 3 levels of variation). We used standard signal detection theory to compute population accuracy from the population confusion matrix definition. The pooled performance scores were highly consistent, with a median (taken over the 72 tasks) Spearman-Brown corrected split-half Pearson-coefficient self-consistency of 0.99. To estimate the subject to subject variability, we selected one subject from each task set and combined the task performance of the three task sets to produce 72 individual human accuracies.

Data Analysis and Metrics. In this section, we discuss metrics that were used to characterize neural data and measure models' match to neurons. These metrics apply to any model and neural population and not just output from hierarchical feedforward neural networks. The only restriction on a model for our methods to be applicable is that the model be image computable, e.g., it is a rule for producing output on any arbitrary stimulus set and does not explicitly rely on stimuli being in a particular subclass of images (10, 11).

Individual neural site predictions. As described in the main text, we used a standard methodology for assessing a model's ability to predict individual sites (12, 13), in which each site is modeled as a linear combination of model outputs. In this procedure, linear regression was used to determine weightings of top-level model outputs which best fit a given neurons' output on a randomly chosen subset of the testing images. The remaining images were used to measure the accuracy of the prediction. Results from multiple random subsets were assessed independently and averaged to ensure statistical validity. Linear regressor results are reported for 10 splits of cross-validation, using 50%/50% train/test splits. Regression weights were obtained using a simple partial least squares (PLS) regression procedure, using 25 retained components (14, 15). For each measured site, separate neural response predictions and cross validated goodness-of-fit r^2 values were obtained. The percentage of explained variance was then computed on a per-site basis by normalizing the r^2 prediction value for that site by the site's Spearman-Brown corrected split-half self-consistency over image presentation repetitions.

To help interpret the meaning of this linear regression technique, consider a hypothetical case in which the responses for all IT neurons in one source animal are known on a set of image stimuli, and the goal is to use these data to predict the response of a random sample of IT neurons from a second target animal. This is a problem of neuron identification, e.g., for each target neuron in the target animal, determining which neuron(s) in the source animal correspond to that target neuron. Although it is known that at the population code level the IT responses of several different animals (and even different primate species) are similar (16), it is not known to what extent there is a 1-to-1 matching of responses between individual neural sites. There is likely to be significant individual variability between the specific tuning curves of units present in different animals, and it is not clear whether the IT units in all animals can be thought of as independent samples from a single master distribution of IT-like neurons. Hence, to explain a given IT unit in the target animal's IT might require linear combinations of multiple source animal IT units, even if a complete sample of neurons from the source animal was available. In more mathematical terms, it is plausible that the best linear fit from one animal's IT to another's would not be particularly sparse. Because it is currently not yet known how sparse the between-animal mapping actually is, in the present work each model's output units is treated a basis from which any observed IT must be constructed, with no prior on the expected sparsity of the weighted sums. Although in our experiments we did collect responses from units in two animals, we do not have enough units from either animal separately to draw a meaningful conclusion as to what the empircal sparsity distribution is, because accurate estimation would likely require on the order of $\sim 10^3$ units from a single animal. If recordings in multiple animals with enough units and images to assess cross-animal fitting sparsity becomes available, such data will be useful to falsify our—or any—model or IT, because the distributions of sparsenesses of the linear mappings from the model to any one population should match the typical animal-to-animal sparseness distribution.

This observation helps clarify the relationship between our work and some existing work on neural fitting (10, 11, 17). In that line of work, which has provided very useful insight into the units up to the V4 area, a different nonlinear model—roughly equivalent to a single convolutional neural network (CNN) network in our model, described below—is fitted separately for each observed visual neuron. Unlike that work, the present results yield a generative model of a neural population as a whole, one that can fit not just the tuning curves of observed neurons but also predicts what types of neurons a typical sample population should a priori contain.

We also implemented two stronger tests of generalization: (*i*) object-level generalization, in which the regressor sample set contained images of only 32 object exemplars (four in each of eight categories), with results assessed only on the remaining 32 objects, averaging results across many such object splits, and (*ii*) category-level generalization, in which the regressor sample set contained images of only half the categories (eight objects in each of, e.g., animal, boat, car, and chair categories), with results assessed only on images of the other categories (eight objects in face, fruit, plant, and table categories), averaged across many such category splits. Fig. S9 shows neural fitting results for object and category generalizations.

Prediction accuracy remains high for the object-level generalization, suggesting that the HMO model is effective at the generalizing neural predictions across a wide range of natural image variability. Neuron-level predictions of all models fall off somewhat in the category generalization case, although relative magnitude and ordering between models are preserved. To interpret this, it is again useful to consider the hypothetical animal-to-animal neural identification task described above. Even with completely comprehensive source animal response data (e.g., all the units in IT—the perfect model), the neuron identification task involves some uncertainty. If the training image stimulus set is not comprehensive enough to completely identify the target neuron, predictions from source to target will break down on images outside that image set. When the data used to identify the target neuron are narrowed to a very limited semantic slice of image space (e.g., a fraction of the object categories), it is expected that it will become difficult to identify that specific neuron from responses to just those images. For example, if all of the images in the training set were only of simple shapes of a uniform size and geometry, it would be impossible to effectively carry out the neuron identification procedure (via linear regression or any other technique). It is instructive to compare this to the results for the population level coding (see section on Representational Dissimilarity Matrices below), where even in the category generalization case, predictions remain accurate.

Linear classifier analysis. Object recognition performance was assessed by training linear classifiers on model and neural output. Linear classifiers are a standard tool for analyzing the performance capacity of a featural representation of stimulus data on discrete classification problems (6, 18). For any fixed population of output features (from either a model or neural population), a linear classifier determines a linear weighting of the units which best predicts classification labels on a sample set of training images. Category predictions are then made for stimuli held out from the weight training set, and accuracy is assessed on these held-out images. To reduce the noise in estimating accuracy values, results are averaged over a number of independent splittings of the data into training and testing portions. In our case, the output features of a model on each stimulus are (by definition) the set of scalar values for each top-level model unit when evaluated on that stimulus, a typical procedure from computer vision studies (19, 20). For neuronal sites, the output features are defined as the vector of scalar firing rates for each unit, as is typical in neural decoding studies (6, 18).

To measure performance, we trained SVM (15) classifiers with *l*2 regularization for three types of tasks supported by the testing image set, including eight-way basic category classification (i.e., animals vs. boats vs. cars, etc.), eight-way car identification (astra vs. beetle vs. clio, etc.), and eight-way face identification, separately for each of the three levels of variation in the testing image set. Eight-way task choices were computed as a maximum over margins from eight binary one-vs.-all (OVA) classifiers (15). Fig. 2*B* shows cross-validated performance accuracies (defined as the fraction of correct predictions averaged over test splits) for the eight-way basic categorization task at the three variation levels. Fig. S6 shows accuracies for subordinate identification tasks as well.

The values shown in Fig. 2*B* and Fig. S6 are for classifiers trained with 75% train/25% test splits, averaged over 20 random category-balanced splits. However, the absolute values of performance for a linear classifier depend on the choice of the number of training examples used. To ensure that our conclusions were not dependent this choice, we computed performance curves for varying numbers of training examples. Although absolute performances did vary as a function of training examples, we found that the relative ordering of performances did not (Fig. S7*A*). Moreover, representations that were effective at high variation level (e.g., the IT neuronal population and the HMO model units) achieved most of their performance with comparatively small numbers of training examples.

Absolute performance also varies with the number of features used—the number of neuronal sites sampled in the case of neural data, or the number of top-end units in the case of models. As with the number of training examples, we would like to be sure that our results do not depend strongly on the number of sampled units. However, the analysis of dependence on number of units is somewhat less straightforward than analysis of training set size dependence, because it is not immediately clear how to fairly equate one neural unit with a fixed number of sample model units. Ideally, we would have extremely large numbers of both kinds of units and then simply make comparisons on complete population samples. Given the limitations of neural data collection, the limiting factor in this work is the number of neural sites sampled. We believe, however, that for the three key comparisons that we make, sample sizes issues do not strongly impact our results:

i) IT neural sample vs. V4 neural sample: At approximately the same number of neural samples (168–128), the performance values at high variation image set are extremely widely separated. Although it is unlikely that this difference is due to neural sample size, to ensure that this is true, we computed performance curves for subsamples of the population of different sizes (Fig. S7*B*), averaging over many subsamples of each fixed size. At all sizes, the IT population strongly outperformances the V4 population. Because these subsample curves appear to have a predictably logarithmic shape, we also fit the data to a logarithmic functional form to extrapolate approximately how many units would be required to achieve the performance measured from the human behavioral experiments. Our estimate suggests that ~1,050 ± 300 IT units would be consistent with human performance, whereas $\sim 10^7$ V4 units would be required. Such estimates are necessarily very rough, but they illustrate the magnitude of the differences between these neural populations.

ii) IT neural sample vs. existing comparison models: In all cases the models sampled produced more output features than we had neural sites (4,096 in the case of HMAX, 24,316 for the V2-like model, and 86,400 for the V1-like model; see below for more information on the these models). The results in Fig. 2*B* show performances computed with the total number of model features in each case. The implication of this is that, even with thousands or tens of thousands of features, these models are not able to equal the performance level of even 168 randomly chosen IT units. Equating the number of features, either by increasing the number IT samples or decreasing the number of model features, would only make the magnitude of the gap larger.

iii) IT neural sample vs. the HMO model outputs. Our claim is that the HMO model is plausibly correct, i.e., it achieves roughly the right performance for a reasonable number of samples. The HMO model performs at approximately human levels with 1,250 top-end outputs, within the sampling error of the number of IT units suggested by extrapolation to achieve human performance. We also subsampled the HMO model to have as many features as our IT sample (168) and found that, although the performance degraded somewhat, it did not drop below measured IT performance levels. However, it is certainly possible that investigating the detailed dependence of model and IT performance on number of samples would allow us to falsify the HMO model. This falsification would be of interest for spurring future work, but for the reasons described above, it would be unlikely to invalidate the claims made in the present work.

Population code representational dissimilarity matrices. Given stimuli $S = s_1, \ldots, s_k$ and vectors of neural population responses $R = \vec{r}_1, \ldots, \vec{r}_k$ in which r_{ij} is the response of the *j*th neuron to the *i*th stimulus, we following ref. 16 by defining the representational dissimilarity matrix (RDM) as

$$\mathrm{RDM}(R)_{ij} = 1 - \frac{\mathrm{cov}\left(\vec{r}_i, \vec{r}_j\right)}{\sqrt{\mathrm{var}\left(\vec{r}_i\right) \cdot \mathrm{var}\left(\vec{r}_j\right)}}.$$

RDM structure is indicative of a range of behavior that a given neural population can support (21), and two populations can have similar RDMs on a given stimulus set (and similar population-level classification performance) even if the low-level details of the neural responses are somewhat different. Because they involve correlations over the feature dimension, RDMs alleviate some of the ambiguities just discussed in analyzing individual units. We produced RDMs for the IT and V4 neural populations, as well as for each of the model-based synthetic IT and V4 neural populations using weights obtained from the regressions for the individual site fits (Fig. 3 *D* and *E* and Fig. S8). Following Kriegeskorte (21), we measured similarity between population representations by assessing the Spearman rank correlations between the RDMs for the two populations. In addition to the standard image-level RDM, in which each pair of test images gives rise to an element of the RDM, we also computed object-level RDMs by averaging population responses for each object before computing correlations (so that each pair of objects gives rise to an element of the 64 × 64 object-level RDM). Similarity of the HMO model object-level RDMs with the IT object-level RDMs are what shown and quantified in Fig. 3 *D* and *E*.

The RDM for the IT neural population we measured has clear block-diagonal structure—associated with IT's exceptionally high categorization performance—as well as off-diagonal structure that characterizes the IT neural representation more finely than any single performance metric. In contrast, the RDM for the V4 population shows how high levels of variation blur out explicit categorical structure for intermediate visual areas, providing a clear visualization of the contrasting population responses underlying the high-variation V4-IT performance gap shown in Fig. 2*B*.

We also computed RDMs for object- and category-level generalizations, using the weightings from the regressions produced as described above in the section on individual neural site predictions. It is instructive to notice that the HMO model maintains high levels of IT similarity even at category-level generalizations (Fig. 3 *D* and *E*), suggesting that, although individual IT units may be hard to predict from a semantically narrow slice of image space (e.g., half the categories only), the overall population code structure remains well predicted.

Modeling. ***Comparison models.*** We compared results for performance, single site neural fitting, and population-level similarity for a variety of computational models, including the following:

i) The trivial Pixel control, in which 256 × 256 square images were flattened into a 65,536-dimensional feature representation. The pixel features provided a control against the most basic types of low-level image confounds.

ii) The baseline SIFT computer vision model (22). This model provided another control against low-level image confounds.

iii) An optimized V1-like model (23), built on grid of Gabor edges at a variety of frequencies, phases, and orientations. This model provided an approximation of a comparison point to lower levels in the ventral visual stream.

iv) A recent V2-like model (24), composed of conjunctions of Gabors. This model provides an approximation of the second level of the ventral stream.

v) HMAX (19, 25), a multilayer convolutional neural network model targeted at modeling higher ventral cortex. Because it is a deep network, HMAX has large IT-like receptive fields. HMAX is one of main existing first-principles-based models that attempts to build up invariance through hierarchical alternation of simple and complex cell-like layers.

vi) PLOS09, a recent three-layer convolutional neural network (26), which also has large IT-like receptive fields and which was discovered via a high-throughput screening procedure that was a predecessor to the HMO procedure.

Ideal observer semantic models. As shown in Figs. 3 and 5, we also computed the IT and V4 predictivity for ideal-observer semantic models (27). Although these ideal observers are not image computable models, given our perfect knowledge of image metadata, we were able to compute explained variance percentages using the same linear regression protocol applied to the image-computable models. We evaluated two ideal observers:

i) A category ideal observer. This model has eight features, one for each of the eight categories present in the test image set. For each image, the *i*th feature is 1 if the image contains an object of category *i*; otherwise, it is 0. For each IT unit, the eight linear regression weights for this feature set effectively describe how much each category contributes to that unit's response.

ii) An all-variable ideal observer. This model is given oracular access to all metadata parameter variables for the images, with one feature for each of 64 object identities (similar to the category ideal observer features above), in additional to features reporting object position, size, scale and image background.

If the IT or V4 explained variance for these (or any) ideal observers were close to 100%, then they would provide a conceptually interpretable explanation of neural variation, a very scientifically desirable result. In fact, the explained variance percentages are significantly less than 100% for the ideal observers we tested (although of course other better ones might be found, e.g., by taking into account 3D object curvature). These ideal observers therefore serve as useful controls to which other computational models can be compared. For example, the ideal category model serves to control for the minimum amount of IT explained variance that should be expected from any model that has high categorization performance. Insofar as a model with high categorization performance explains more explained variance than the ideal category model, that additional predictivity can be attributed to the constraints of the model class.

CNN model class. Here, we mathematically specify the basic class of CNN models used in this paper. These principles are consistent with a large parameter space of possible networks. The specific parameterized space of networks we use is close to that described in Pinto et al. (26), with one, two, or three convoluational layers. Each layer is characterized by a fixed set of parameters, but parameter values can differ between layers. This parameter space expresses an inclusive version of the hierarhical feedforward network concept and contains models similar to that used in many previous studies for different parameter values (19, 23, 24, 25).

More specifically, each individual layer is composed of operations including local pooling, normalization, thresholding, and filterbank convolution, which are combined as follows:

$$N_\Theta(X) = \text{Normalize}_{\theta_N}\left(\text{Pool}_{\theta_p}\{\text{Threshold}_{\theta_T}[\text{Filter}_{\theta_F}(X)]\}\right), \quad \textbf{[S1]}$$

where X is a 2D input image. The subscripts $\Theta = (\theta_p, \theta_N, \theta_T, \theta_F)$ denote the specific parameter choices for the constituent operations, setting radii, exponents, and thresholds, as described in Pinto et al. (26). Similar to previous studies, we also use randomly chosen filterbank templates in all models, but additionally allow the mean and variance of the filterbank to vary as parameters. Functions of the form N_Θ are the simplest computational units that we operate on and are thought to be plausible representations of what happens in a single cortical layer (28). To produce deep CNNs, layers of the form N_Θ are stacked hierarchically:

$$\ldots \mathbf{P}^{\ell-1}_{\theta_{P,\ell-1}} \xrightarrow{\text{Filter}} \mathbf{F}^{\ell}_{\theta_{F,\ell}} \xrightarrow{\text{Threshold}} \mathbf{T}^{\ell}_{\theta_{T,\ell}} \xrightarrow{\text{Pool}} \mathbf{N}^{\ell}_{\theta_{P,\ell}} \xrightarrow{\text{Normalize}} \mathbf{P}^{\ell}_{\theta_{N,\ell}} \ldots, \quad \textbf{[S2]}$$

where ℓ is layer number and the initial input at the 0th layer is the image pixel array X. We denote such a stacking operation as $\otimes$, so that the stacked hierarchical model can be written as

$$\mathbf{N} \equiv \otimes_{i=1}^{k} N_{\Theta_i}.$$

Let $\mathcal{N}_k$ denote the space of all stacked networks (N) of depth k or less. In this study, our CNNs are networks of depth $k = 3$ or less.

Mixture networks. We extend the class of CNNs by a fourth principle, namely that, at any stage, networks can consist of mixtures of CNNs where each component has a potentially distinct set of parameters (e.g., pooling size and number of filters), representing different types of units with different response properties (29). Such mixture networks may combine components of differing complexity, which correspond to anatomical bypass connections within the ventral stream (30) (Fig. S2*B*).

For a mathematical formulation of this idea, note that because the networks in $\mathcal{N}_3$ are convolutional, they can be combined in a standard fashion. Specifically, given a sequence of individual modules $\mathbf{N}(\Theta_{i1}, \Theta_{12}, \ldots, \Theta_{in_i})$ for $i \in [1, \ldots, J]$, possibly of different depths, the mixture network is defined by aligning the module output layers along the spatial convolutional dimension. Because the outputs of each of the modules is a 3D tensor, this alignment is well defined up to a rescaling factor in the spatial dimension. We denote this alignment operation by the symbol $\oplus$, so that a combined mixture network can be written as

$$\mathbb{N} \equiv \oplus_{i=1}^{J} \mathbf{N}(\Theta_{i1}, \Theta_{12}, \ldots, \Theta_{in_i}).$$

The total output of networks of this form is also a 3D tensor, so they too can be stacked with the $\otimes$ operation to form more complicated, deeper hierarchies. By definition, the full class $\mathbb{N}$ consists of all of the networks formed by iteratively composed the stacking ($\otimes$) operation and the combination ($\oplus$) operation. Conceptually, members of $\mathbb{N}$ are nonlinear mixtures of modules chosen from a base class of simpler homogenous neural networks (e.g., the elements of $\mathcal{N}$). Schematically, $\otimes$ is a vertical composition relationship, increasing the depth complexity of the network. Biologically, it is plausible to think of $\otimes$ as corresponding to producing complex nonlinear representations by feedforward layering. Conversely, $\oplus$ is a horizontal composition relationship, increasing the breadth complexity of the network. Biologically, this may correspond to the idea of mixing heterogenous populations of different types of units in a given area.

HMO. The HMO procedure (31) is a computational optimization procedure designed to identify high-performing network architectures from the space $\mathbb{N}$. Intuitively, it is a version of adaptive boosting in which rounds of optimization are interleaved with boosting and hierarchical stacking (32). The process first analyses error patterns in the recognition predictions of candidate networks, picking complementary components, e.g., those with optimally nonoverlapping errors. Subsequent rounds of optimization attempt to optimize a criteria weighted toward those stimuli that are misclassified by the first-round results. As a result, complementary components emerge without having to prespecify the corresponding subtasks semantically (or in any other way), mapping the complex structure of high-variation recognition problems onto the parameter space of neurally plausible computations. These components are then aligned along their convolutional dimensions and used as inputs to repeat the same procedure hierarchically to build more complex nonlinearities. Although other possible optimization procedures could potentially be used to create high-performing neural networks (33), the HMO process may be particularly efficient because it explicitly takes advantage of the complementary strengths of different components within the large space of network architectures.

This section describes details of the HMO procedure. Suppose that $N\in\mathcal{N}$ and S is a screening stimulus set. Let E be the binary-valued classification correctness indicator, assigning to each stimulus image s 1 or 0 according to whether the screening task prediction was right or wrong, where the prediction for each s was made by using maximum correlation classifiers (MCCs) (34) on the output features of N with threefold cross-validation (see *Materials and Methods* describing screening set metric). Let

$$\mathrm{performance}(N,S)=\sum_{s\in S}E[N(s)].$$

To efficiently find N that maximizes $\mathrm{performance}(N,S)$, the HMO procedure follows these steps:

i) Optimization: Optimize the performance function within the class of single-stack networks of some fixed depth d_1, obtaining an optimization trajectory of networks in $\mathcal{N}_{d_1}$ (Figs. S2*C* and S5*A*, *Left*). The optimization procedure that we use is hyperparameter tree parzen estimator, as described in ref. 35. This procedure is effective in large parameter spaces that include discrete and continuous parameters.

ii) Boosting: Consider the set of networks explored during step 1 as a set of weak learners and apply a standard boosting algorithm (Adaboost) to identify some number of networks $\mathrm{N}_{11},\ldots,\mathrm{N}_{1l_1}$ whose error patterns are complementary (Fig. S2*C*, *Right*).

iii) Combination: Form the heterogenous network $\mathbb{N}_1=\oplus_i\mathrm{N}_{1i}$ and evaluate $E[\mathbb{N}_1(s)]$ for all $s\in S$.

iv) Error-based reweighting: Repeat step 1, but reweight the scoring to give the jth stimulus s_j weight 0 if N_1 is correct in s_j and 1 otherwise. That is, the performance function to be optimized for N is now

$$\sum_{s\in S}E[N_1(s)]\times E[N(s)].$$

Repeat step 2 on the results of the optimization trajectory obtained to get models $N_{21},\ldots N_{2k_2}$, and repeat step 3 (Figs. S2*C* and S5*A*, *Right*). Steps 1, 2, and 3 are repeated K times.

After K repetitions of this process, we will have obtained a mixture network $N=\oplus_{i\le K,j\le k_i}N_{ij}$. The process can then simply be terminated or repeated with the output of N as the input to another stacked network. In the latter case, the next layer is chosen using the model class $\mathcal{N}_{d_2}$ to draw from, for some fixed depth d_2, and using the same adaptive hyperparameter boosting procedure. The metaparameters of the HMO procedure include the numbers of components $l_1, l_2, \ldots$ to be selected at each boosting round, the number of times K that the interleaved boosting and optimization is repeated, and the number of times M that this procedure is stacked. For the purposes of this work, we fixed the metaparameters $K=3$, $l_1=l_2=l_3=10$, and $M=2$ (with $d_1=3$, $d_2=1$).

Model screening procedure. To construct a specific model network, we applied HMO to a screening task (Figs. S1*B* and S5). Like the testing set, the screening set was designed to be very challenging—having high levels of object pose, position, and scale variation (36). However, to ensure that a fair test could be made, in all other regards, the screening images were distinct from the testing image set, containing objects in totally nonoverlapping semantic categories, using none of the same background scenes, lighting, or noise conditions. The image set used for the HMO screening procedure consisted of 4,500 images of 36 distinct objects, chosen from one of nine categories, including bodies, building, flowers, guns, musical instruments, jewelry, shoes, tools, and trees. As in the testing set, in the high-variation subset, objects were shown in varying positions, sizes, and poses, placed in a variety of uncorrelated natural backgrounds scenes. Lighting was provided by ambient environment reflection, and speckle noise was added to simulate natural image distortions. Images were rendered with the Panda3d package (37).

The relationship between the screening set and testing set is intended to be similar to that between any two typical samples of natural images: having some high-level natural statistical commonalities, but otherwise quite different specific content. For this reason, any performance increases that could be demonstrated to transfer from the screening to the testing set are likely to also transfer, at least to some extent, to other high-variation image sets.

The screening objective sought to minimize classification performance error on the 36-way object classification task (no categorical semantic information was used), as assessed by training unregularized MCC classifiers with threefold cross-validated 50%/50% train/test splits. Using the HMO procedure on this screening set, we generated a network HMO_0, which produces 1,250-dimensional feature vectors for any input stimulus. HMO_0 is the model that we refer to throughout the paper as the HMO model and that we used for all testing evaluation.

In the optimization, candidate networks were first evaluated on overall performance metric, and performance gradients in parameter space were identified as seen in the trend toward decreasing screening loss (step 1; Fig. S5*A*, *Left*, blue dots). Ten components were identified by boosting (step 2) and combined. In subsequent rounds (Fig. S5*A*, *Right*, red dots), the optimization criterion was biased toward weighting more heavily errors of the architectures from earlier rounds (step 4). Decreasing loss in these later rounds indicates that models are improving at the subset of images that confused the components identified in round 1. The complementary model components identified in the two different optimization rounds were associated with different directions in the overall large parameter space of possible neural-like computations that effectively solve different subtasks of the overall recognition task (Fig. S5*B*). As expected, training performance increases as components are combined (Fig. 4*C*).

Assessment. As described in the main text, we then assessed the HMO_0 model against the testing dataset (Fig. S1*A*). The HMO_0 model showed high performance on testing set, as described in the main text, Fig. 2*B*, and Fig. S6. Comparisons to neural data showed that the HMO_0 model also had significantly power to explain neural data, both at the individual site level (Figs. 3 *A*–*C* and 5) and the population level (Fig. 3 *D* and *E* and Fig. S8). The HMO model is a significantly closer match to IT population representations at all variation levels, but the difference is especially evident at the high variation level that most clearly exposes how the high-level IT representation differs from the lower-level V4 representation (Fig. S8, black bars).

Subsequently, we determined the stability of the HMO procedure by running it on a variety of alternative screening sets with different choices of objects and categories, varying the numbers of within-category exemplars and varying amounts of semantic similarity to the testing set. Performance and neural fitting ability were largely stable to these changes. Although some of these later models exhibited higher performance and neural explanatory power than the initial HMO_0 model, to prevent domain overfitting, we report only the results of the initial model HMO_0 constructed before any testing set results were obtained.

It is important to note how our screening process connects to the evaluation of other models. In the cases of the SIFT, V1-like, and V2-like models, we did not pretrain those models using the screening set: this is because those models do not accept pretraining data at all. In the case of HMAX, which does accept pretraining data, we used the testing data itself for pretraining, to give that model that highest chance of performance success. Separately, we also performed a pretraining of the HMAX model using the screening set and then reextracted it on the testing set, but found that this only further decreased final performance and neural fit results of the HMAX model (e.g., learned parameters did not effectively transfer from the screening to the testing set).

Another issue relevant to comparison of models is the question of numbers of total internal units. In the mixture models that we used to create the HMO model, the numbers of filters at each layer were kept very small (≤ 24) to ensure that a total combined model composed of several such components would not be unmanageably large. In the HMO_0 model, the total number of units is approximately the same as that in the HMAX model, and the total number of output features is somewhat smaller (1,250 vs. 4,096).

Correlation experiments. Performance and neural predictivity results suggest that as performance on high-variation tasks increases, metrics of neural similarity also increase (Fig. 1b). To determine whether this correlation is a general feature of the deep feedforward architectures defined here, we ran several additional high-throughput experiments, evaluating a large number of candidate model architectures and measuring categorization performance and IT neural predictivity for each model (Fig. 1a and Fig. S3). Specifically, we performed three high-throughput searches of the parameter space $\mathcal{N}_3$ described in the above:

i) Random selection. We drew several thousand randomly sampled models from the parameter space $\mathcal{N}_3$. For each one, we computed linear classifiers for performance and linear regressors for IT predictivity, as described above. Each green point in Fig. S1*A* corresponds to one such model. In this condition, there is a significant correlation between performance and IT predictivity ($r = 0.55$, $n = 2{,}016$). Negative values on the *y* axis correspond to models having negative goodness of fit (the r^2 coefficient of determination statistic), due to overfitting on the training images. Fig. S3, *Left*, shows model performance for as a function of time during the procedure; the lack of any trend corresponds to random sampling of models.

ii) Performance optimization. Using the recently developed Hyperopt metaparameter optimization algorithm (35), we performed a directed search for network parameters that maximized performance on the high-variation eight-way categorization task (Fig. S1*A*, blue points). This optimization was carried out using the recently developed hyperparameter optimization algorithm Hyperopt (35). Via this optimization, absolute performance and fitting values were significantly improved compared with the random condition. Moreoever, although the optimization was done without reference to any neural data, the correlation between performance and IT predictivity actually increased significantly ($r = 0.78$, $n = 2{,}043$). Fig. S3, *Center*, shows the optimization criterion as a function of time step during the optimization procedure; the upward trend is due to the optimization process. Although the optimization gains toward the end of the optimization process are slow and appear to be plateau, small improvements are still observed.

iii) IT predicitivity optimization. In the third experiment, we directly optimized model architecture for IT predictivity, this time without reference to performance (Fig. S1*A*, orange dots). The correlation is comparable to the performance-optimized condition ($r = 0.80$, $n = 1{,}876$), but the optimization plateau occurs significantly earlier (Fig. S3, *Right*; we repeated the optimization multiple times, and obtained the same result each time; this suggests that continued optimization would not be effective). Moreover, the best-performing models from the performance-optimization experiment predict IT neural output as well as the models explicitly optimized for the predictivity objective, whereas the reverse does not hold.

The results of these experiments support three inferences. First, model performance is modestly correlated with neural predictivity in a random selection regime. Second, optimization pressure for either metric produces markedly better cross-validated accuracy on the optimized axis, and in doing so, significantly strengthens the correlations with the other nonoptimized metric. Third, when optimizing for performance, the best-performing models predict neural output approximately, as well as the most predictive models selected explicitly for neural predictivity, but not vice versa. The feedforward model architecture class itself imposes a relationship between high-level behavior (performance) and more detailed neural mechanisms, but directed optimization focuses on a region within network parameter space where this constraint is much stronger.

The inclusion of the category ideal observer (purple square in Fig. 3*D*) shows an effective negative control on the performance-predictivity relationship: it lies significantly off the main trend, making it visually clear how the correlation arises from a combination of architectural and performance constraints working in concert.* However, this ideal observer is not an image-computable model, It would be especially instructive to identify a image-computable algorithm that achieved invariant object recognition high performance but low neural IT neural consistency. If such an algorithm existed, its architecture might illustrate a very nonneural solution to object recognition tasks as a purely computer vision problem. With current understanding, we cannot rule out the possibility that such an algorithm does not exist—e.g., recent high-performing computer vision systems are deep convolutional neural networks (33).

Fig. 1*A* also implies that, even with intensive optimization, individual models in the $\mathcal{N}_3$ are limited in performance and

*Note that a converse control, in which a model has very high neural consistency for a population of IT units but low performance, cannot exist. IT units are already known to have high performance, so any model that matches IT units sufficiently well must also have high performance.

neural prediction ability, underscoring the need for an enlarged model class. However, further analysis of the results of these optimization experiments provides insight into how to construct a more effective model class. In Fig. S10, we show scatter plots of model performance on pairs of binary subtasks, e.g., performance on the two-way cars-vs.-planes task compared with performance on the two-way boats-vs.-chairs task. These plots show that, as the optimization algorithm explores parameter space, it identifies mutually exclusive subspaces that are effective for some of the natural subtasks defined in the overall task space. The highest performing architectures for one subtask are often significantly suboptimal for other subtasks, leading to V-shaped subtask-vs.-subtask scatter plots. In choosing a single architecture that is best for overall performance, the optimization is forced to tradeoff performance on some of these subtasks.

The effectiveness of optimized mixture models (such as HMO) may be understood in the context of Fig. S10, which suggests that models composed of mixtures from the $\mathcal{N}_3$ class might be significantly more effective than any single model alone. Such mixtures are also suggested by the observation from neurophysiology studies that patches within IT are selectively responsive for distinct object classes (38–40). Intuitively, such subregions might correspond to architecturally specialized structures within the larger feedforward class. Mixture models avoid the tradeoffs inherent in individual feedforward structures by combining several pareto-optimal network architectures. By identifying particularly effective mixture combinations, the HMO procedure overcomes these limitations efficiently. In addition, however, a key ingredient for the HMO model's success is that the components constituting the model, which were (by construction) complementary on the original screening set, were still complementary on the testing set. This holds even though the testing set had entirely distinct object categories, so the basis on which the complementarity of the components was originally discovered—nonoverlapping error patterns in screening set object identity judgements—is no longer even applicable. This strongly rules out image domain-specific overfitting and suggests that mixture components discovered by performance optimization may form a generically useful visual representational basis that can be recombined to solve new object recognition problems. In fact, achieving high performance and neural fitting capability appears to require diversity in many of the parameters of the constituent components (Fig. S11).

Model parameter diversity analysis. We characterized model parameters in terms of per-component tuning specificity vs. intercomponent diversity. Tuning specificity is a measure of how specifically each parameter needed to be tuned to produce optimal performance. To compute this, we analyzed the distribution of each parameter's values along the optimization trajectory near the optimal point using the concept of entropy. By definition, the entropy of (N samples from) a distribution P is

$$E(P) = \log(N) - \frac{1}{N}\sum_i n_i \log(n_i),$$

where N is the number of samples from the distribution, the sum is taken over possible values i of the distribution, and n_i is the number of samples with value i.

Suppose an optimal module component Θ^* occurs at timepoint t^* in the trajectory of one optimization run in the HMO process. Then, let $P_{p,k}(\Theta^*)$ be the distribution of values of parameter p in the k-neighborhood around t^* in the optimization trajectory

$$P_{p,k}(\Theta^*) = (\text{value of parameter } p \text{ at time points } t \in [t^* - k, \ldots, t, \ldots, t^* + k]).$$

The specificity of parameter p around optimal point Θ^* is, by definition,

$$-E\left[P_{p,k}(\Theta^*)\right].$$

Intuitively, this is because, if the distribution $P_{p,k}(\Theta^*)$ had high entropy, this indicated that the value of the parameter near the optimal point did not matter very much and therefore was not tuned very specifically. If, on the other hand, the distribution had low entropy, it was tightly clustered around one or a few optimal values that the optimization had identified as being important, suggesting it was highly tuned. For the purposes, we took $k = 25$ time steps, but values were not strongly sensitive to k with the range of 10–100. For each parameter p, we report the median tuning specificity of that parameter, taken over all component modules.

Intercomponent diversity is a measure of how variable a parameter is between the component modules. This was measured by computing, for each pair of components, how well separated the distributions of the parameter's values around each component were from each other. More formally, the d-prime discriminability index, d', for two distributions P_1 and P_2 is defined by

$$d'(P_1, P_2) = \frac{|\langle P_1\rangle - \langle P_2\rangle|}{\sqrt{0.5[\mathrm{var}(P1) + \mathrm{var}(P2)]}}.$$

(The sample d' uses the sample versions of the mean and variances.) Suppose Θ_1^* and Θ_2^* are two optimal components chosen by the HMO procedure. Then we measure separability for these two components as

$$d'\left[P_{p,k}(\Theta_1^*), P_{p,k}(\Theta_2^*)\right].$$

For each parameter p, we define intercomponent diversity as the median of this separation value taken over all pairs of components Θ_1 and Θ_2. The higher the diversity, the more different the components were from each other, and vice versa.

Parameters that have both high tuning specificity and high intercomponent diversity are both critical for performance and required to be heterogenous. Our results highlight certain types of parameters as being simultaneously highly tuned and diverse. This is particularly true for two broad classes of parameters, as can be seen Fig. S11, upper right: (*i*) local filter statistics, including filter mean and spread and (*ii*) the pooling exponents trading off between max-pooling and average-pooling (41). Other types of parameters are highly tuned but less diverse (nonlinear activation thresholds; lower right), whereas some appear less important overall (higher-level pooling and normalization kernel sizes; lower left). Interestingly, we observe that the parameter controlling the number of network layers (depth) is both comparatively highly tuned and diverse suggests that allowing network modules of different levels of complexity in the heterogeneous models is important for achieving high model performance. As a result, the final model has a significant proportion of lower-complexity units projecting directly to the final layer, suggesting that bypass connections (e.g., projects from V1 to V4 or V2 to IT) may be a key functional feature of the ventral stream (30).

Taken together, these results point to a computationally rigorous explanation for why heterogeneity is observed in the receptive fields of ventral stream neurons both at the unit and subarea levels (29, 38, 30, 42).

1. Felleman DJ, Van Essen DC (1991) Distributed hierarchical processing in the primate cerebral cortex. *Cereb Cortex* 1(1):1–47.
2. DiCarlo JJ, Maunsell JHR (2000) *Inferotemporal Representations Underlying Object Recognition in the Free Viewing Monkey* (Society for Neuroscience, New Orleans).
3. Majaj N, Hong H, Solomon E, DiCarlo J (2012) A unified neuronal population code fully explains human object recognition. *Computational and Systems Neuroscience* (COSYNE). Available at http://cosyne.org/cosyne12/Cosyne2012_program_book.pdf. Accessed March 3, 2014.
4. Frey BJ, Dueck D (2007) Clustering by passing messages between data points. *Science* 315(5814):972–976.
5. Quiroga RQ, Nadasdy Z, Ben-Shaul Y (2004) Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering. *Neural Comput* 16(8):1661–1687.
6. Hung CP, Kreiman G, Poggio T, DiCarlo JJ (2005) Fast readout of object identity from macaque inferior temporal cortex. *Science* 310(5749):863–866.
7. Rust NC, Dicarlo JJ (2010) Selectivity and tolerance ("invariance") both increase as visual information propagates from cortical area V4 to IT. *J Neurosci* 30(39):12978–12995.
8. Rosch E, Mervis CB, Gray WD, Johnson DM (1976) Basic objects in natural categories. *Cognit Psychol* 8(3):382–439.
9. Plachetka T (1998) POV ray: Persistence of vision parallel raytracer. *Proceedings of the Springer Conference on Computer Graphics*, ed Szirmay-Kalos L (SCCG, Budmerice, Slovakia), pp 123–129.
10. Connor CE, Brincat SL, Pasupathy A (2007) Transformation of shape information in the ventral pathway. *Curr Opin Neurobiol* 17(2):140–147.
11. Brincat SL, Connor CE (2004) Underlying principles of visual shape selectivity in posterior inferotemporal cortex. *Nat Neurosci* 7(8):880–886.
12. Carandini M, et al. (2005) Do we know what the early visual system does? *J Neurosci* 25(46):10577–10597.
13. Cadieu C, et al. (2007) A model of V4 shape selectivity and invariance. *J Neurophysiol* 98(3):1733–1750.
14. Helland I (2006) Partial least squares regression. *Encyclopedia of Statistical Sciences* (John Wiley & Sons, Hoboken, NJ).
15. Pedregosa F, et al. (2011) Scikit-learn: Machine learning in Python. *J Mach Learn Res* 12:2825–2830.
16. Kriegeskorte N (2009) Relating population-code representations between man, monkey, and computational models. *Front Neurosci* 3(3):363–373.
17. Sharpee TO, Kouh M, Reynolds JH (2013) Trade-off between curvature tuning and position invariance in visual area V4. *Proc Natl Acad Sci USA* 110(28):11618–11623.
18. Rust NC, Mante V, Simoncelli EP, Movshon JA (2006) How MT cells analyze the motion of visual patterns. *Nat Neurosci* 9(11):1421–1431.
19. Mutch J, Lowe DG (2008) Object class recognition and localization using sparse features with limited receptive fields. *International Journal of Computer Vision* 80(1): 45–57.
20. LeCun Y, Bengio Y (2003) Convolutional networks for images, speech, and time series. *The Handbook of Brain Theory and Neural Networks*, ed Arbib MA (MIT Press, Cambridge MA), pp 255–258.
21. Kriegeskorte N, et al. (2008) Matching categorical object representations in inferior temporal cortex of man and monkey. *Neuron* 60(6):1126–1141.
22. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60(2):91–110.
23. Pinto N, Cox DD, DiCarlo JJ (2008) Why is real-world visual object recognition hard? *PLOS Comput Biol* 4(1):e27.
24. Freeman J, Simoncelli EP (2011) Metamers of the ventral stream. *Nat Neurosci* 14(9): 1195–1201.
25. Serre T, Oliva A, Poggio T (2007) A feedforward architecture accounts for rapid categorization. *Proc Natl Acad Sci USA* 104(15):6424–6429.
26. Pinto N, Doukhan D, DiCarlo JJ, Cox DD (2009) A high-throughput screening approach to discovering good forms of biologically inspired visual representation. *PLOS Comput Biol* 5(11):e1000579.
27. Geisler WS (2003) Ideal observer analysis. *The Visual Neurosciences*, eds Chalupa L, Werner J (MIT Press, Boston), pp 825–837.
28. DiCarlo JJ, Zoccolan D, Rust NC (2012) How does the brain solve visual object recognition? *Neuron* 73(3):415–434.
29. Martin KA, Schröder S (2013) Functional heterogeneity in neighboring neurons of cat primary visual cortex in response to both artificial and natural stimuli. *J Neurosci* 33(17):7325–7344.
30. Nakamura H, Gattass R, Desimone R, Ungerleider L (2011) The modular organization of projections from areas v1 and v2 to areas v4 and teo in macaques. *J Neurosci* 14:1195–1201.
31. Yamins D, Hong H, Cadieu C, Dicarlo J (2013) Hierarchical modular optimization of convolutional networks achieves representations similar to macaque it and human ventral stream. *Adv Neural Inf Process Syst* 26:3093–3101.
32. Schapire RE (1999) *Theoretical Views of Boosting and Applications*, Lecture Notes in Computer Science (Springer, Berlin), Vol 1720, pp 13–25.
33. Krizhevsky A, Sutskever I, Hinton G (2012) ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems* 25:1097–1105.
34. Buciu I, Pitas I (2003) ICA and Gabor representation for facial expression recognition. *Proceedings of the 2003 International Conference on Image Processing* (Institute of Electrical and Electronics Engineers, Piscataway, NJ), Vol 2, pp 855–858.
35. Bergstra J, Yamins D, Cox D (2013) Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. *Proceedings of The 30th International Conference on Machine Learning*, eds Dasgupta S, McAllester D (MIT Press, Cambridge, MA).
36. DiCarlo JJ, Cox DD (2007) Untangling invariant object recognition. *Trends Cogn Sci* 11(8):333–341.
37. Goslin M, Mine MR (2004) The Panda3D graphics engine. *Computer* 37(10):112–114.
38. Downing PE, Chan AW, Peelen MV, Dodds CM, Kanwisher N (2006) Domain specificity in visual cortex. *Cereb Cortex* 16(10):1453–1461.
39. Kanwisher N, McDermott J, Chun MM (1997) The fusiform face area: A module in human extrastriate cortex specialized for face perception. *J Neurosci* 17(11):4302–4311.
40. Freiwald WA, Tsao DY (2010) Functional compartmentalization and viewpoint generalization within the macaque face-processing system. *Science* 330(6005):845–851.
41. Riesenhuber M, Poggio T (2000) Models of object recognition. *Nat Neurosci* 3(Suppl): 1199–1204.
42. Chelaru MI, Dragoi V (2008) Efficient coding in heterogeneous neuronal populations. *Proc Natl Acad Sci USA* 105(42):16344–16349.

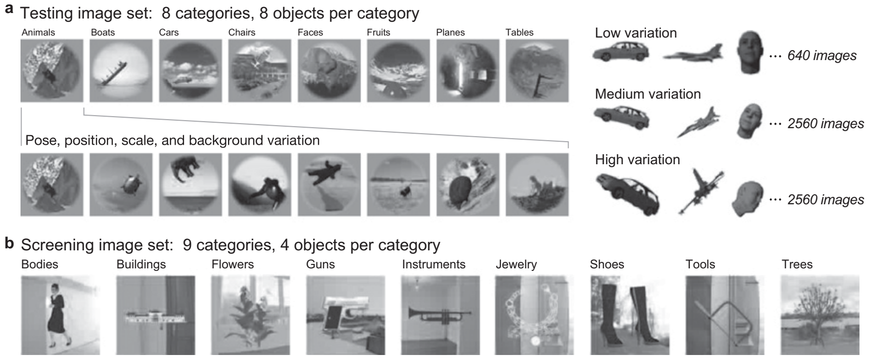

Fig. S1. (*A*) The neural representation benchmark (1) testing image set on which we collected neural data and evaluated models contained 5,760 images of 64 objects in eight categories. The image set contained three subsets, with low, medium, and high levels of object view variation. Images were placed on realistic background scenes, which were chosen randomly to be uncorrelated with object category identity. (*B*) The screening image set used to discover the HMO model contained 4,500 images of 36 objects in nine categories. As with any two uncorrelated samples of images from the world—such as those images seen during development vs. those seen in adult life—the overall natural statistics of the screening set images were intended to be roughly similar to those of the testing set, but the specific content was quite different. Thus, the objects, semantic categories, and background scenes used in screening were totally nonoverlapping with those used in the testing set. Moreoever, different camera, lighting and noise conditions, and a different rendering software package, were used.

1. Cadieu C, et al. (2013) The neural representation benchmark and its evaluation on brain and machine. *International Conference on Learning Representations*. arXiv:1301.3530.

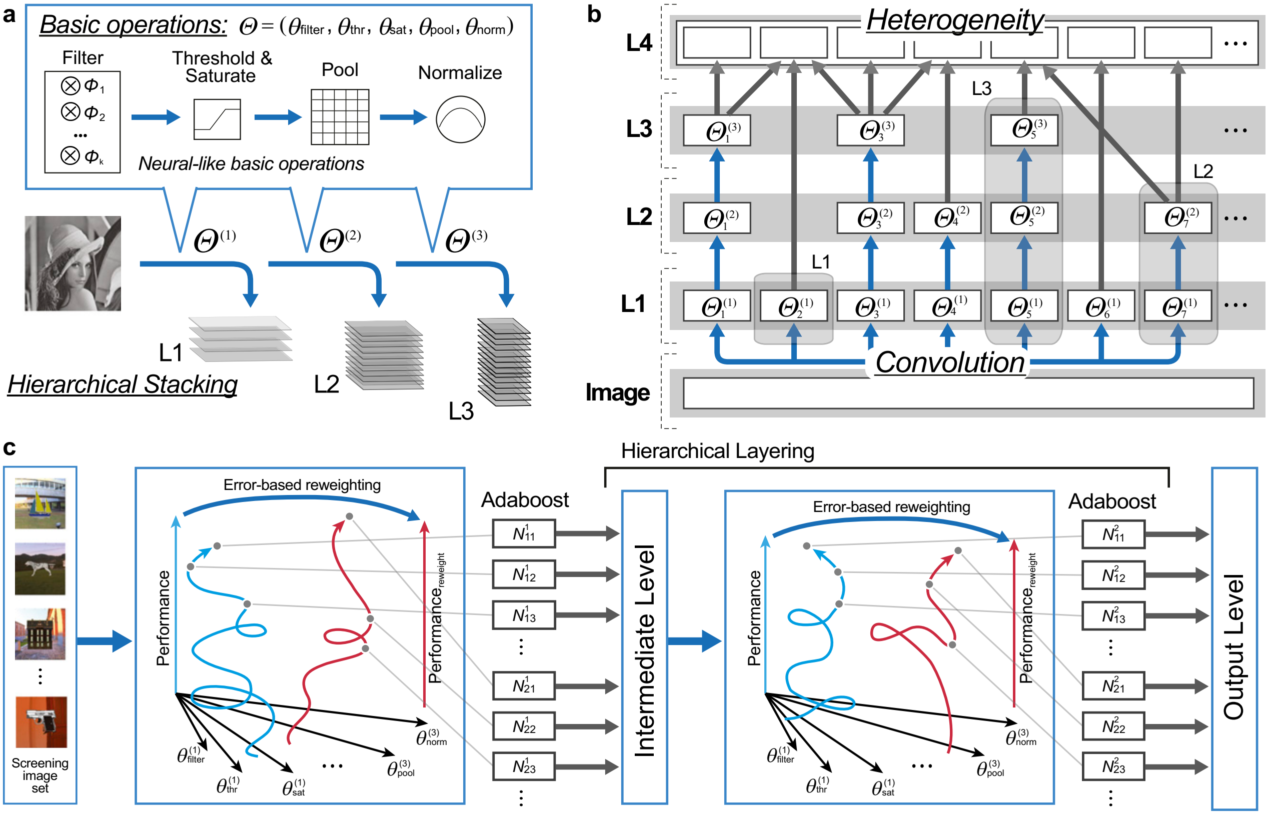

Fig. S2. In this work, we use CNN models. CNNs consist of a series of hierarchical layers, with bottom layers accepting inputs directly from image pixels, with units form the top and intermediate layers used to support training linear classifiers for performance evaluation and linear regressors for predicting neural tuning curves. (*A*) Following a line of existing work, we limited the constituent operations in each layer of the hierarchy to linear-nonlinear (LN) compositions including (*i*) a filtering operation, implementing AND-like template matching; (*ii*) a simple nonlinearity, e.g., a threshold; (*iii*) a local pooling/aggregation operation, such as softmax; and (*iv*) a local competitive normalization. These layers are combined to produce low complexity (L1), intermediate complexity (L2), and high complexity (L3) networks. All operations are repeated convolutionally at each spatial position, corresponding to the general retinotopic organization in the ventral stream. (*B*) In creating the HMO model, we allow mixture of several of these elements to model heterogenous neural populations, each acting convolutionally on the input image. The networks are structured in a manner consistent with known features of the ventral stream, as a series of areas of roughly equal complexity, but which permit bypass projections. (*C*) HMO is a procedure for searching the space of CNN mixtures to maximize object recognition performance. With several rounds of optimization, HMO creates mixtures of component modules that specialize in subtasks, without needing to prespecify what these subtasks should be. Errors from earlier rounds of optimization are analyzed and used to reweight subsequent optimization toward unsolved portions of the problem. The complementary component modules that emerge via this process are then combined and used as input to repeat the procedure hierarchically (*Materials and Methods* and *SI Text*).

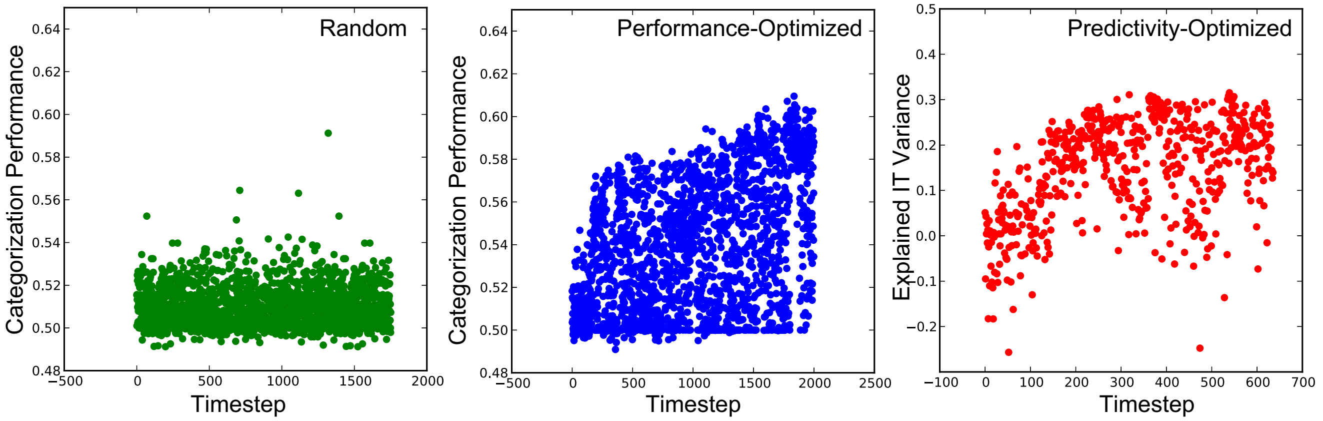

Fig. S3. Optimization time traces for the high-throughput experiments shown in Fig. 1. In the performance and fitting-optimized the *y* axis shows the optimization criterion—in the random selection case (*Left*), no optimization was done, and the performance data were ignored.

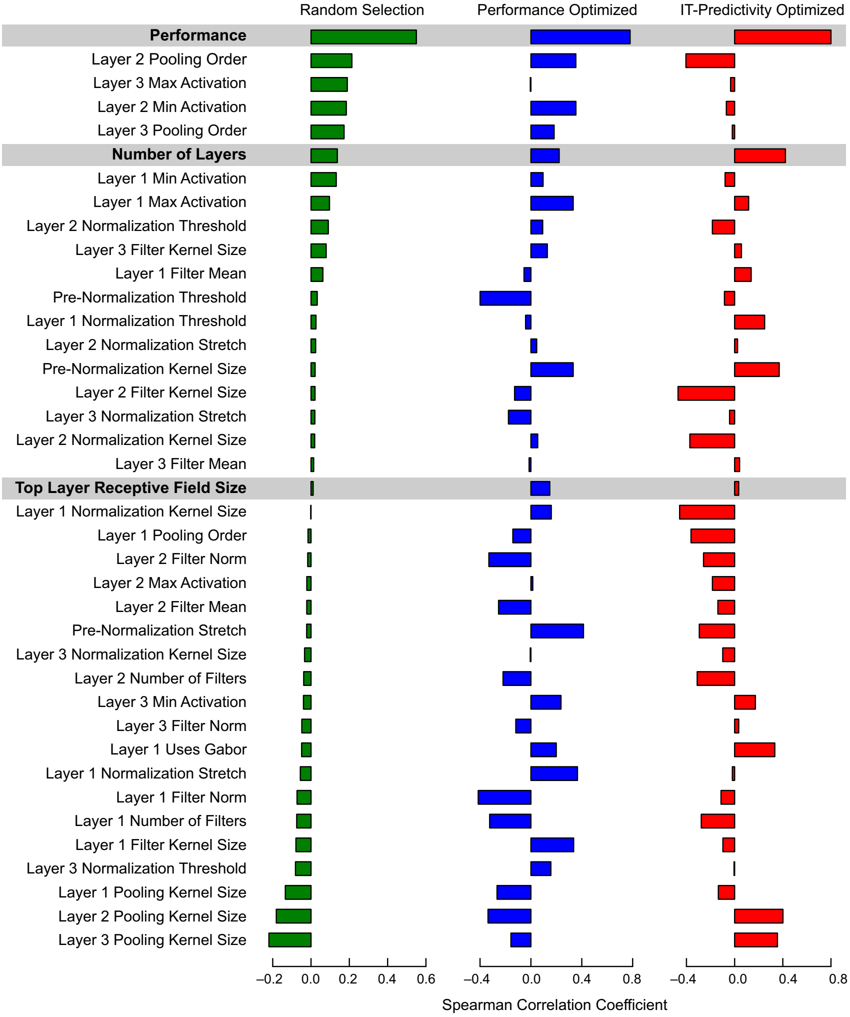

Fig. S4. Correlation of model parameters with IT predictivity for the three high-throughput experiments shown in Fig. 1. Parameters for which the correlation is significantly different from 0 are shown. Also included are several additional metrics that are not direct model parameters but that represent measurable quantities of interest for each model, e.g., model object recognition performance. The *x* axis is Spearman *r* correlation of the given parameter with IT predictivity for the indicated model selection procedure, including random (*Left*, green bars), performance-optimized (*Center*, blue bars), and IT predictivity optimized (*Right*, red bars). Parameters are ordered by correlation value for the random condition. Performance strongly correlates with IT predictivity in all selection regimes. Number of layers (model depth) consistently correlates as well, but much more weakly. Interestingly, one obvious metric—receptive field size at the top model layer—is only very weakly associated with predictivity, because, although the best models tended to have larger receptive field sizes, a large number of poor models also shared this characteristic.

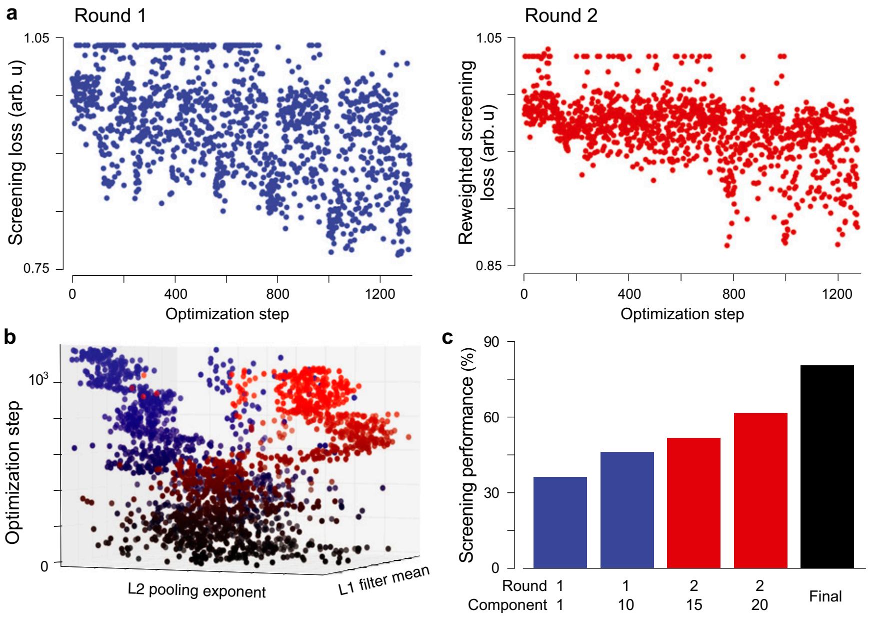

Fig. S5. (*A*) Optimization loss traces during the HMO procedure showing decreased loss as optimization proceeded. (*B*) Parameter-space trajectories during two optimization rounds shown in *A* (round 1 are blue dots; round 2 are red dots). This 3D plot shows parameter values for two chosen parameters (L1 filter mean and L2 pooling exponent) out of many, but it is evident that subsequent rounds of optimization (e.g., red) gravitate toward different paremeter combinations (i.e., different network architetures) than earlier rounds of optimization (e.g., blue). (*C*) Training performance as a function of model complexity, showing dramatic increases as components from round 1 (blue bars) and round 2 (red bars) were added. The final model (black bar) consists of 30 components identified with three complementary rounds of optimization, plus one L1 layer that, anatomically, stacks on top of those 30 components and, functionally, produces nonlinear combinations of their outputs.

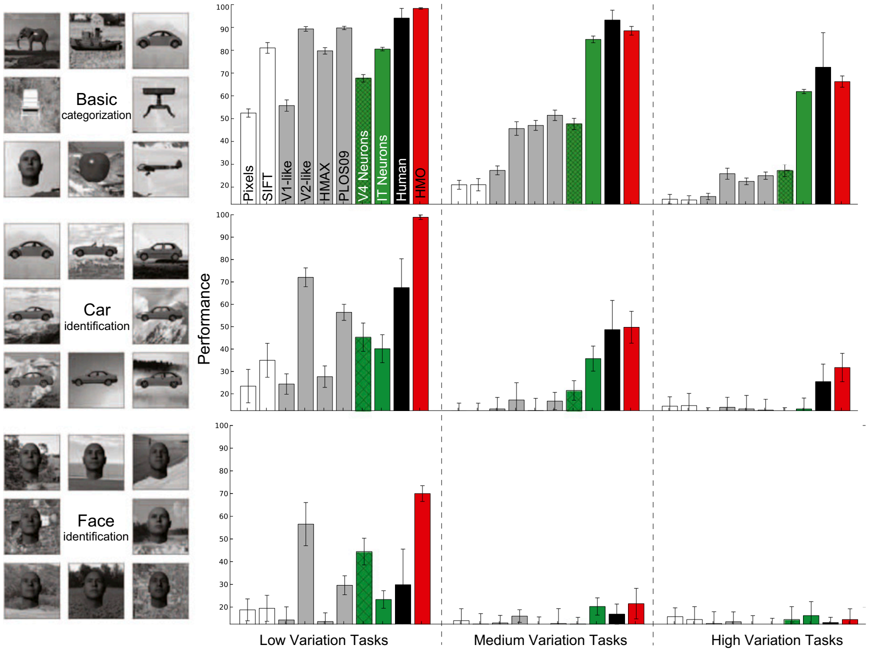

Fig. S6. Classification results for tasks including basic eight-way basic categorization (*Top*), eight-way subordinate car identification (*Middle*), and eight-way subordinate face identification (*Bottom*). Each task was assessed at low, medium, and high levels of image variation (*SI Text*). Comparison was made between neural data, human data, existing models from the literature, and the HMO model outputs. The tasks span a wide range of difficulty, from low-variation basic eight-way categorization where humans perform at greater than 95% accuracy to high-variation subordinate face identification, where human performance is indistinguishable from chance.

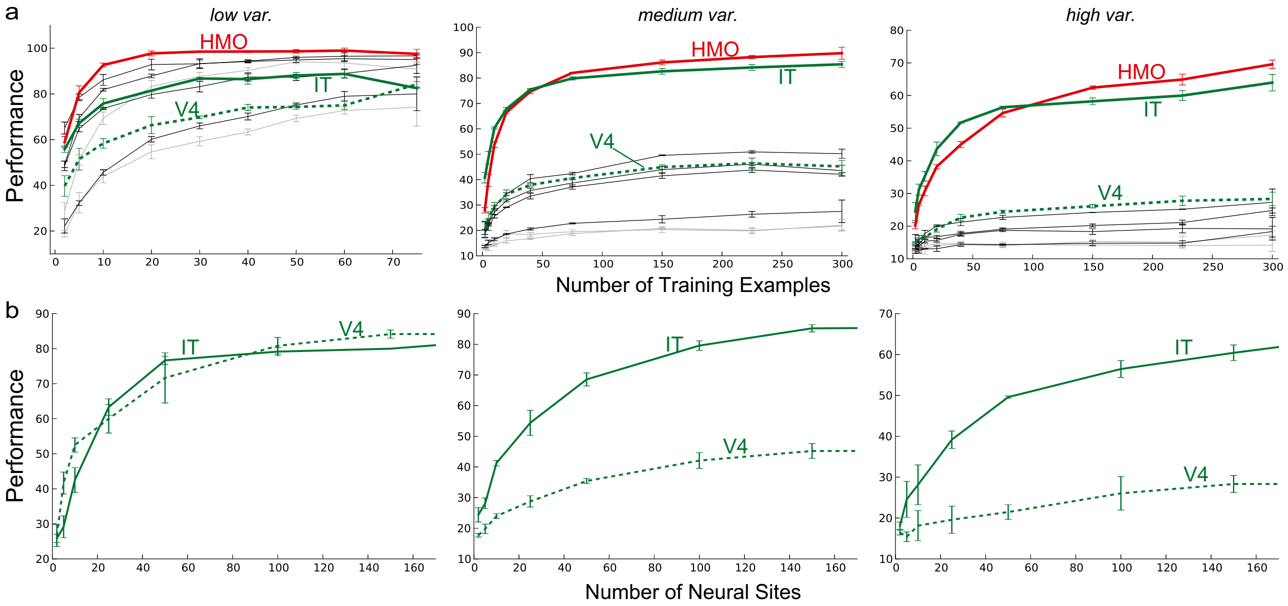

Fig. S7. (*A*) Dependence of performance on number of training examples for models and neural populations. HMO model is shown in red; IT population in solid green; V4 in dotted green; all other control models are shown in black. (*B*) Direct comparison of dependence of performance on number of neural sites, for the IT (solid green) and V4 (dotted green) neural populations.

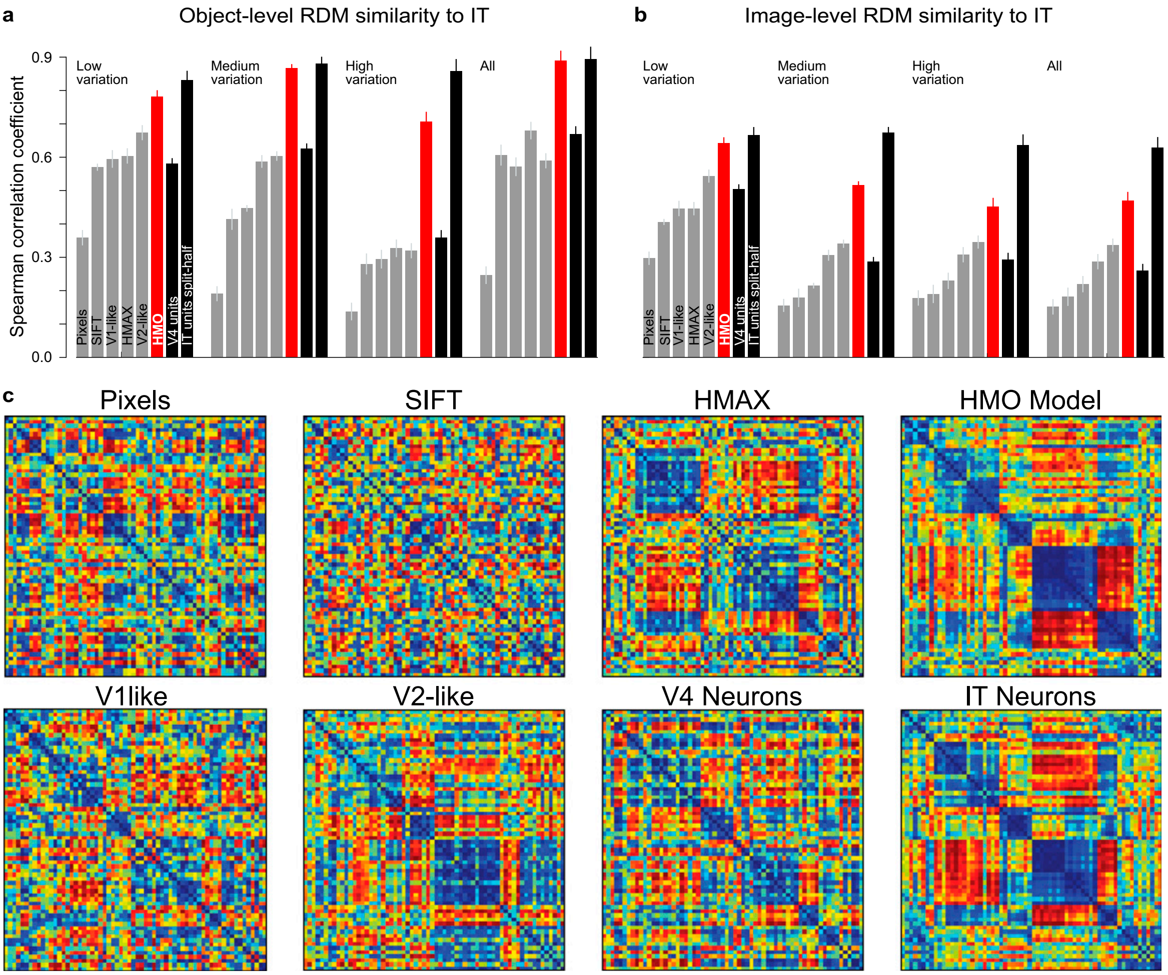

Fig. S8. Additional RDM comparisons to IT population structure. As in Fig. 3 *D* and *E*, each bar shows the Spearman correlation of an RDM for a model (or V4 population) with the RDM for the IT neural population on the same stimulus set. We show comparisons for three subsets of the test image set separated by variation level (Low, Medium, and High), as well as for the whole stimulus set (All). (*A*) Comparisons of RDMs at the object level, in which population representation vectors are averaged on a per-object basis before taking the pairwise correlations to make the RDM matrices. (*B*) More detailed image-level RDMs comparisons, with each stimulus represented separately. (*C*) Object-level RDMs for a variety of models and the V4 and IT neural populations.

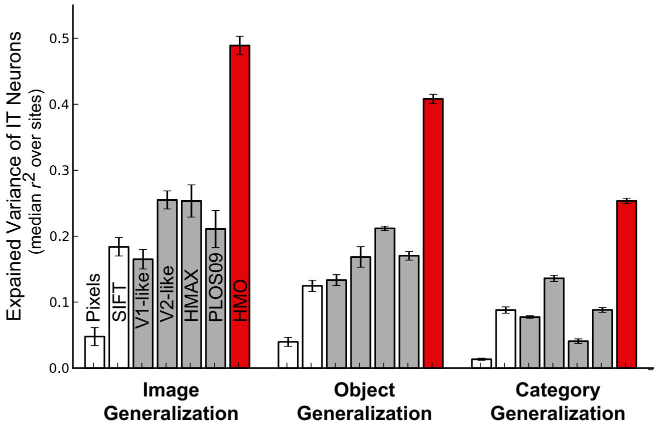

Fig. S9. IT explained variance for each model, fit with training/test image splits generated by (*1*) image generalization, a random selection process in which train and test splits contain images of the same 64 objects, but on different backgrounds and at widely different poses, positions, and sizes; (*2*) object generalization, in which train and test images are split so that they contain no overlapping objects, so that predictions are tested for generalization across object identity as well as position, pose, size and background variation; and (*3*) category generalization, in which train an test images are split so that they contain no overlapping categories, so that predictions are tested across category boundaries as well. Fig. 3*E* shows the corresponding results at the population RDM level.

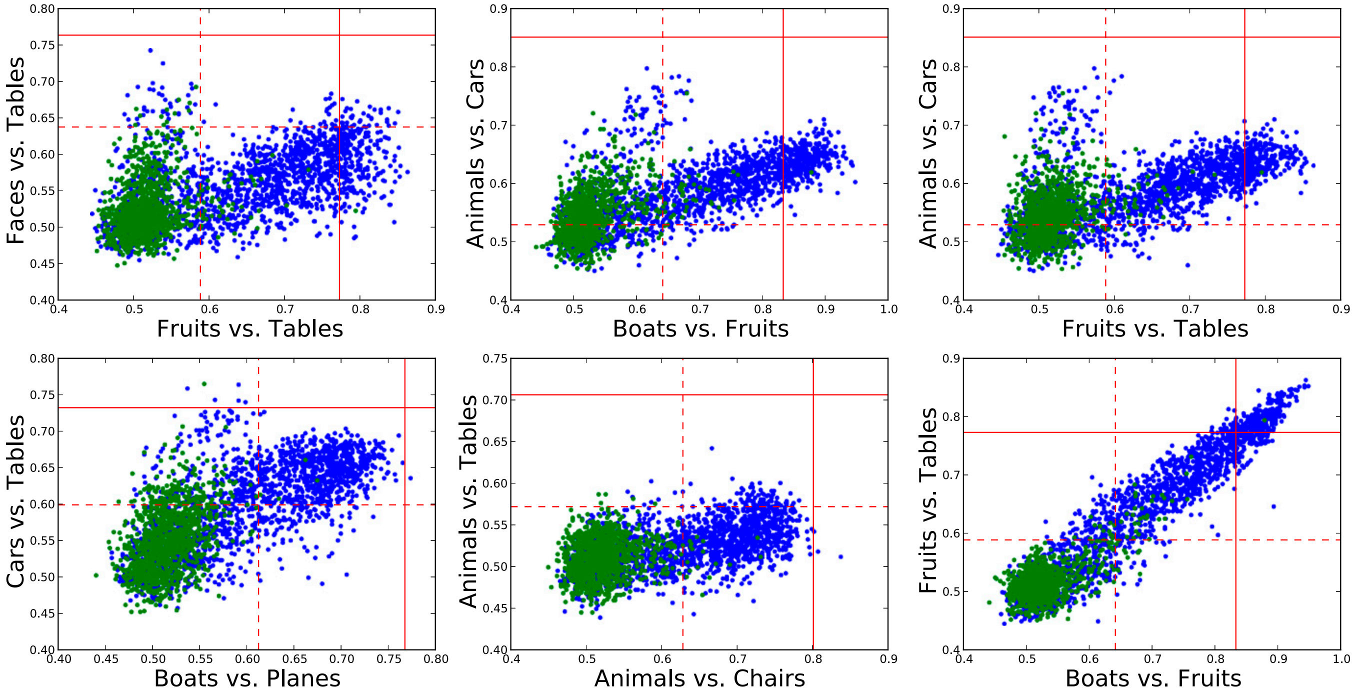

Fig. S10. Tradeoffs between subtask-optimal architectures. Each panel shows pairwise relative performance of the models from the high-throughput experiments in Fig. 1*A* and Fig. S3 on a variety of binary subtasks. As in that figure, random selections are shown in green and performance-optimized selections are shown in blue. Sometimes performance on one binary subtask—e.g., Boats-vs.-Fruits and Fruits-vs.-Tables (lower right corner panel)—directly correlates with performance on another. More commonly, there is a tradeoff between subtask performance in the models explored during optimization, leading to the V pattern observed in subtask pairs. Because the procedure was maximizing overall performance (as opposed to performance on any one subtask), one arm of the V is heavier than the other, corresponding to the optimization process being forced to make a single choice in each of these tradeoffs.

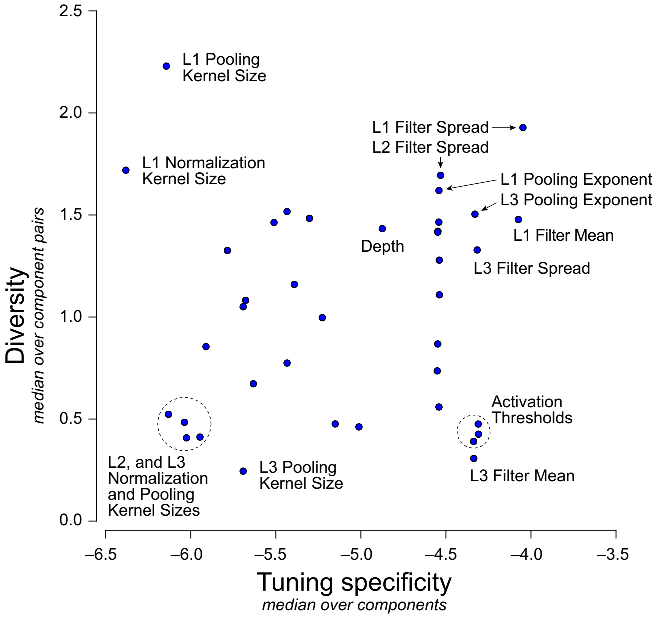

Fig. S11. Characterization of selected model parameters in terms of per-component tuning specificity vs. intercomponent diversity. Each point in this plot represents an architectural parameter in the HMO model. Parameters in the upper right corner are highly tuned but also highly diverse in their tunings between model components. See *SI Text* for the definition of diversity and tuning specificity.