STATISTICAL INFERENCE ON REPRESENTATIONAL GEOMETRIES

A PREPRINT

Heiko H. Schütt *
Zuckerman Institute, Columbia University
New York City, NY 10027, USA
hs3110@columbia.edu

Alexander D. Kipnis †
Zuckerman Institute, Columbia University
New York City, NY 10027, USA
alexander.kipnis@tue.mpg.de

Jörn Diedrichsen
Western University
London, Ontario, Canada
joern.diedrichsen@googlemail.com

Nikolaus Kriegeskorte
Zuckerman Institute, Columbia University
New York City, NY 10027, USA
nk2765@columbia.edu

December 20, 2021

ABSTRACT

Neuroscience has recently made much progress, expanding the complexity of both neural-activity measurements and brain-computational models. However, we lack robust methods for connecting theory and experiment by evaluating our new big models with our new big data. Here we introduce a new inferential methodology to evaluate models based on their predictions of representational geometries. The inference can handle flexible parametrized models and can treat both subjects and conditions as random effects, such that conclusions generalize to the respective populations of subjects and conditions. We validate the inference methods using extensive simulations with deep neural networks and resampling of calcium imaging and functional MRI data. Results demonstrate that the methods are valid and conclusions generalize correctly. These data analysis methods are available in an open-source Python toolbox.

***K*eywords** Representational similarity analysis · toolbox · neuroscience · data analysis

1 Introduction

Experimental neuroscience has recently made rapid progress with technologies for measuring neural population activity. Spatial and temporal resolution, as well as the coverage of measurements across the brains of animals and humans have all improved considerably [1, 2, 3, 4, 5, 6, 7]. Activity is measured using a wide range of techniques, including electrode recordings [8, 9, 1], calcium imaging [3], functional magnetic resonance imaging [fMRI; 4, 6, 7], and scalp electro- and magnetoencephalography [EEG and MEG; 10, 11]. In parallel to the advances in measuring brain activity, theoretical neuroscience has substantially scaled up brain-computational models that implement computational theories [e.g., 12, 13, 14, 15]. The engineering advances associated with deep learning [e.g., 16, 17] provide powerful tools for modeling brain information processing for complex, naturalistic tasks [18]. How to leverage the new big data to evaluate the new big models, however, is an open problem [19, 20, 21, 22].

*Also at: Center for Neural Science, New York University, New York, USA
†present address: Max Planck Institute for Biological Cybernetics, 72076 Tuebingen, Germany

An important concept for understanding how neural population codes is the concept of *representational geometry* [23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41]. Neural activity patterns that represent particular pieces of mental content, such as the stimuli presented in a neurophysiological experiment, can be viewed as points in the multivariate neural population response space of a brain region. The representational geometry is the geometry of these points. The geometry is characterized by the matrix of distances among the points. This distance matrix abstracts from the roles of individual neurons and provides a summary characterization of the neural population code that can be directly compared among animals and between brain and model representations (e.g. a layer of a neural network model). The representational geometry provides a multivariate characterization of a neural population code that generalizes linear decoding analyses. A linear decoding analysis reveals whether particular information is amenable to linear readout. The full distance matrix captures what information is available to any linear decoder [42].

A popular method for analyzing representational geometries [43] on which we build here is representational similarity analysis [RSA 44, 45]. RSA is a two step process (Fig. 1): In the first step, RSA characterizes the representational geometry by estimating the representational distance for each pair of experimental conditions (e.g. different stimuli), and assembles these in a representational dissimilarity matrix (RDM). We use the more general term “dissimilarity" here to include dissimilarity measures that are not distances or metrics in the mathematical sense [46, 42]. An RDM is computed for the neural population in our brain region of interest and for each model representation. In the second step, each model is evaluated by the accuracy of its prediction of the data RDM.

RSA is widely used [43, 35, 42] and has gained additional popularity since image-computable representational models like deep neural networks have become more commonly available [e.g. 47, 33, 48, 49, 12, 50]. There has been important recent progress with the estimation of the representational distances and better measures of RDM prediction accuracy. For estimation, biased and unbiased distance estimators with improved reliability have been proposed [51, 46]. For better quantification of the RDM prediction accuracy, the sampling distribution of distance estimators has been derived and measures of RDM prediction accuracy that take the dependencies between dissimilarity estimates into account have been proposed [52].

Here, we introduce a comprehensive novel methodology for statistical inference on models that predict representational geometries (Fig. 1). We propose bootstrapping methods that can statistically support generalization to new subjects, new conditions, or both simultaneously, as required to support the theoretical claims that we wish to make. We extend these bootstrapping methods with crossvalidation to enable inference on flexible models, i.e. models with parameters fitted to the data [53, 54]. This is important, because theories do not always make a specific prediction for the representational geometry. There may be unknown parameters, such as the relative prevalences of different tuning functions in the neural population or properties of the measurement process.

We thoroughly validate the new inference methods using simulations and neural activity data. Extensive simulations based on deep neural network models and models of the measurement process, where ground truth is known, confirm the validity of the inference procedures and their ability to generalize to the populations of subjects and/or conditions. Using real data from fMRI (human) and calcium imaging (mouse), we confirm that conclusions generalize from an experimental data set (subset of real data) to the entire data set (which serves as a stand-in for the population). The methodology is available in a new open-source RSA toolbox in Python (https://github.com/rsagroup/rsatoolbox).

2 Results

2.1 Methods for inference on representational geometries

2.1.1 Estimating the representational geometry from data

The methodology we introduce here is independent of the method for estimating the data RDM for each subject. For inference in our simulations and tests, we use the crossvalidated Mahalanobis (crossnobis) estimator [46, 45, 27]. The crossnobis RDM estimate is recommended for fMRI data (under most circumstances), as it is unbiased by measurement noise and can take noise covariance into account to improve reliability [46, 27]. For data from other measurement modalities with non-Gaussian noise (e.g. Poisson noise), alternative crossvalidated distance estimators can be used (Online Methods 5.1.1).

2.1.2 Estimating model performance at predicting representational geometries

The second step of RSA is to evaluate the accuracy with which each model predicts the measured representational geometry. To evaluate the RDM prediction accuracy, we need a measure for the similarity of two RDMs. Such a measure should meet at least the following two requirements: First, it should be maximal if the two RDMs are identical. Second, it should be invariant to scaling (as the dissimilarities usually do not have commensurable units for models and data).

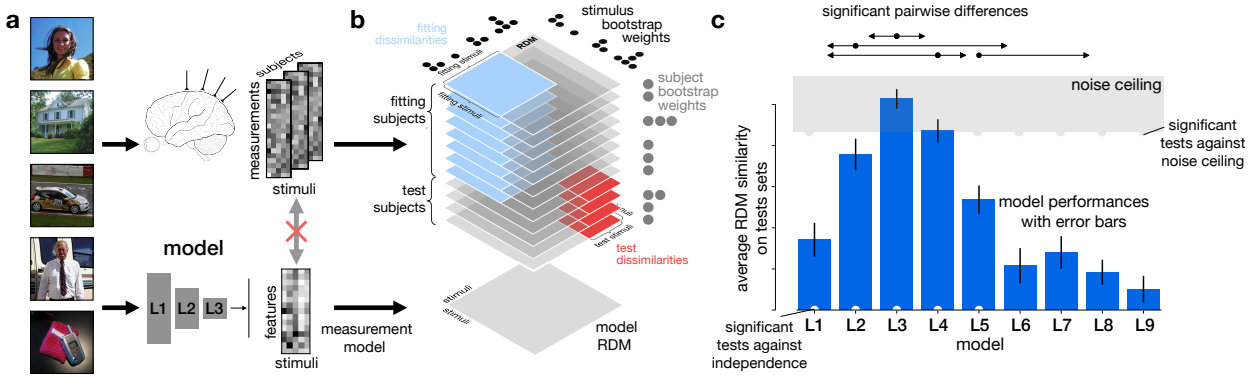

Figure 1: **Overview of model-comparative inference. a** Multiple conditions are presented to observers and to models (here different stimulus images). The brain measurements during the presentation produce a set of measurements for each stimulus and subject, potentially with repetitions; A model yields a feature vector per stimulus. Importantly, no mapping between brain measurement channels and model features is required. **b**: To compare the two representation we compute a representational dissimilarity matrix (RDM) measuring the pairwise dissimilarities between conditions for each subject and each model. For model comparison we perform crossvalidation nested within a bootstrap to remove overfitting bias and to estimate our uncertainty about the model performances. **c**: Based on our uncertainty about model performances we can perform various statistical tests, which are marked in the graphical display. Dots along the noise ceiling and the axis mark models that are significantly different from the noise ceiling or chance performance respectively. Pairwise differences are summarized by arrows. Each arrow indicates that the model marked with the dot performed significantly better than the model the arrow points at and all models further away in the direction of the arrow. (image credit: Ecoset [49] and Wiki Commons)

Popular measures include various correlation coefficients and the cosine similarity (Online Methods 5.1.2). Correlation coefficients and cosine similarity both ignore that the entries of an estimated data RDM are not independent. Taking into account the dependencies between the dissimilarity estimates by whitening of the sampling distribution of the RDM [55, 56] leads to less variable model performance estimates. This can improve the power of model-comparative inference to approach the theoretical limit achieved by a likelihood-ratio test [52].

2.1.3 Estimating the variance of model-performance estimates for generalization to new subjects and conditions

For frequentist inference, we need to estimate how variable the model-performance estimates would be if we repeated the experiment. Technically, we need to estimate the covariance matrix of model performance estimates, which allows us to compute variances for each individual model's performance and for the difference between any pair of models. Past studies have used Student's t-test and non-parametric alternatives on the single-subject model performance estimates for model comparisons. However, inference then only takes the variability over subjects into account and thus does not justify generalization to different experimental conditions (e.g. different stimuli).

Computational neuroscience usually pursues insights that also generalize to a broader population of conditions [57]. Although this generalization may be justified by prior knowledge, the field also needs inference methods that enable researchers to support generalization to the population of conditions statistically when a sufficient sample of conditions can be studied. We therefore developed uncertainty estimates that can treat the measured conditions as a random sample from a population whether the subjects are treated as a random sample or as fixed.

If we intend to generalize merely to new measurements of the same conditions in the same subjects, we require repeated measurement blocks, each of which enables us to obtain an independent RDM estimate. We can then use the variability across blocks for inference, treating blocks the way we would treat subjects if we intended to generalize to new subjects.

If we intend to generalize to either new subjects or new conditions, we can bootstrap resample either subjects or conditions to perform inference. Bootstrap resampling estimates the variance of experimental outcomes by repeatedly sampling data sets from the measured data, effectively treating the measured data as an approximation of the population [58]. For generalization only across subjects, we additionally have the option to use t-tests or rank-sum tests, because all our model performance measures are means across independently drawn single-subject model performances. For a single condition, by contrast, there is no representational geometry prediction that we could evaluate.

To enable simultaneous generalization to both populations, we introduce a novel two-factor bootstrap method. Perhaps surprisingly, simultaneous bootstrap resampling of subjects and conditions (two-factor bootstrap, [59]) overaccounts for variability not coupled to either factor, such as measurement noise. Our simulations and theory show that the two-factor bootstrap triple-counts the variance contributed by the measurement noise (Online Methods 5.1.4, Fig. 3 c, Fig. 6 c). This effect is not unique to RSA; it appears for any type of experiment in which two factors (here subject and condition) jointly determine the experimental outcome. To correct the estimate, we introduce a novel corrected two-factor bootstrap procedure to estimate the variance: We compute bootstrap samples for each of the two factors separately and for both factors simultaneously. We then linearly combine the variances from these three bootstraps to cancel the surplus contribution from the measurement noise.

2.1.4 Evaluating the performance of flexible models

Although computational models may be substantially constrained by task training and prior neuroscientific data, uncertainty often remains. We therefore need to be able to test *flexible models*, i.e. models that have parameters to be fitted to the brain-activity data. Two elements that often require fitting are feature weights and the measurement model. Feature weighting is required when a model does not specify a priori how prevalent different tuning profiles are in the neural population or in the measured signals. For example, for deep neural network representations to match brain responses well, it is usually necessary to weight the features [e.g. 33, 53, 48, 59]. A flexible measurement model may be necessary, because the process of measurement may subsample, average or distort neural responses in ways that the computational model does not account for. For example, models rarely specify the spatial range across which fMRI voxels average the neural activity or which neurons are preferentially sampled by electrophysiological recordings [54].

To account for the overfitting of flexible models, we use a crossvalidated estimate of the model performance. On each fold of crossvalidation, the models are fitted and evaluated with separate partitions of the data, such that each subject and stimulus is assigned to either the training or test set. This entails that a substantial portion of the data is unused on each fold: Each fold excludes all dissimilarities measured in fitting subjects among test conditions and vice versa, as well as all dissimilarities between fitting and test sets of conditions (gray regions in Fig. 1 b). We use a complete set of k partitions of the subject set and a complete set of l partitions of the condition set to define the test sets. We perform $k \cdot l$ folds, one for each combination of a subject and a conditions partition, such that all subjects and conditions appear equally often in the test sets. Across a cycle of $k \cdot l$ folds, this standard crossvalidation procedure yields stable estimates of model performance unbiased by differences in model complexity.

Wrapping bootstrap resampling around the crossvalidated performance estimation enables us to estimate the variances and covariances of our estimates of model performance. We described above that subjects, conditions, or both may be bootstrap-resampled to support different levels of generalization of our conclusions. Our crossvalidation procedure, similarly, can account for overfitting not just to measurement noise but also to peculiarities of the sampled subjects and conditions, as models are tested on subjects and conditions not used in fitting.

Crossvalidation introduces another source of variance, because different random assignments of the conditions and subjects to the crossvalidation folds lead to different model-performance estimates. This effect is particularly large for RSA, because any assignment to crossvalidation folds leaves some measured dissimilarities out of all test sets. In particular, crossvalidation across conditions relies on evaluation on smaller RDMs containing only the dissimilarities among test conditions. Dissimilarities between conditions in different test sets never enter evaluation.

If we used a single crossvalidation cycle inside the bootstrap, we would misinterpret the excess variance caused by the random assignment to crossvalidation folds as part of the variance over repetitions of the experiment, although it is truly just computational and can be averaged out across cycles with different random partitions of subjects and conditions. Simply using a large number of crossvalidation cycles is prohibitively expensive in terms of computation because the crossvalidation is performed for each bootstrap sample. However, if we have at least two crossvalidation cycles with different random partitions for each bootstrap sample, we can estimate the excess variance caused by random partitioning (Online methods 5.1.4). Because the number of bootstrap samples is large, this estimate is quite accurate even when we have only two crossvalidation cycles for each bootstrap sample (Fig. 4 a). We use this estimate of the excess variance to correct the estimate of the (co-)variance of model-performance estimates. To choose how many crossvalidation cycles should be performed per bootstrap sample, we repeatedly estimated the variance for the same datasets while increasing either the number of crossvalidation cycles per bootstrap sample or the number of bootstrap samples. We found that increasing the number of bootstrap samples more effectively stabilizes our uncertainty estimate than increasing the number of crossvalidation cycles (Fig 4 b). Our recommendation, thus, is to use two crossvalidation cycles (the minimum needed to correct for excess variance).

This crossvalidation approach provides model-performance estimates that are not biased by overfitting of flexible models. Fixed and flexible models with different numbers of parameters can be robustly compared with generalization

over conditions and subjects (for details on the inference algorithm and types of flexible model supported, see Online Methods 5.1.4 and 5.1.6).

2.1.5 Frequentist tests for model evaluation and comparison

Based on the uncertainty estimates, we construct frequentist tests to compare models to each other. The default method is a t-test based on bootstrap estimated variances (Online Methods 5.1.5). In addition to comparing models to each other, we compare models to chance performance and to a noise ceiling. The noise ceiling provides an estimate of the performance the true (data-generating) model would achieve. A model that approaches the noise ceiling (i.e. is not significantly below the noise ceiling) cannot be statistically rejected. We would need more data to reveal any remaining shortcomings of the model. The noise ceiling is not 1, because even the true group RDM would not perfectly predict all subjects' RDMs. We estimate an upper and a lower bound for the true model's performance [45]. The RDM that performs best on the measured data on average across subjects provides an upper bound on model performance. For example, for Pearson correlation as the RDM comparison measure, the best performing RDM is the average RDM after normalizing the RDM-vector for each subject to unit variance. To estimate a lower bound, we use crossvalidation, computing the best performing RDM for a subset of the subjects and evaluating on the held-out subjects. For most RDM comparison measures, the best performing RDM can be derived analytically and tests against the lower bound of the noise ceiling can be handled largely like comparisons between models (Online Methods 5.1.3).

2.2 Validation of statistical inference

We validate the inference methods using simulations, functional MRI data and neural data. First, we use abstract simulations to establish that the statistical tests employed are valid given correct estimates of our uncertainty about model performance. Second, we show that our uncertainty estimates about model performance correctly capture the true variability for different generalization schemes in more realistic simulated scenarios. In such scenarios, we also evaluate the power afforded by different RDM comparison measures. Third, we validate the inference procedure for flexible models, confirming that our bootstrap-wrapped crossvalidation scheme correctly accounts for the overfitting of flexible models. Fourth, we validate the methods for real data, acquired with functional MRI in humans and calcium imaging in mice.

2.2.1 Test validity

A frequentist test is valid when the rate of false positives (i.e. the rate of positive results when the null hypothesis is true) does not exceed the specified error rate α, which we set to 5%. Here we check the validity of model-comparative inference, where the null hypothesis is that the two models perform equally well at explaining the representational geometry. We simulate scenarios where two models may predict distinct geometries, but perform equally well on average at predicting the true representational geometry.

To simulate situations where two different models perform equally well, we generated condition-response matrices (containing an activity level for each condition and response channel combination) using matrix-normal models. A matrix normal distribution over matrices yields matrices with normally distributed cells whose covariance is separable into a covariance matrix across rows and one across columns. In our case, rows correspond to the experimental conditions (e.g. stimuli) and the columns correspond to measurement channels (e.g. neurons or voxels). For matrix-normal data, the covariance across conditions captures the similarity among condition-related response patterns and determines the expected squared Euclidean-distance RDM [27]. The covariance among channels only scales the covariance of the distance estimates. This relationship enables us to generate matrix-normal data for arbitrary choices of the expected squared Euclidean-distance RDM. To model the null hypothesis, we choose two models that predict distinct RDMs and generate data, such that the expected data RDM is equally similar to both model RDMs (details in Online Methods 5.2.1).

We also validated simple dependence tests, where a single model is tested and the null hypothesis is that its performance is at chance level. To do so, we performed similar matrix-normal data simulations, evaluating a model that predicts a random RDM on matrix-normal data consistent with an independent random expected data RDM.

All model-comparative and dependence tests were found to be valid. Inflated false-positive rates were observed only for bootstrap tests when used on a small sample of subjects (<20). This scenario is known to produce underestimates of the variance of the model performances by a factor $\frac{n}{n-1}$ for n subjects [e.g. 58, chapter 5.3]. Also all tests that aim for generalization across conditions were conservative, showing false-positive rates substantially below 5%. This is expected, because we did not include any random selection of conditions in our models, but enforced the H_0 for the exact measured conditions. We conclude that the tests are valid when we can adequately estimate the variance of the

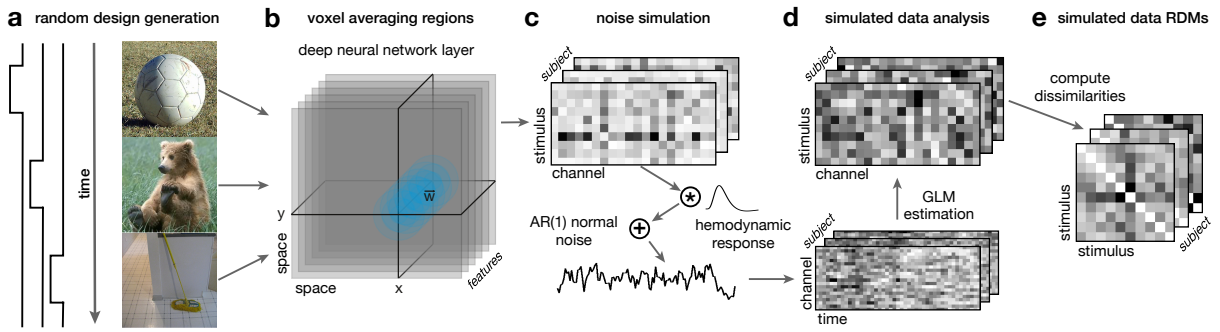

Figure 2: **Illustration of the deep neural network based simulations for fMRI-like data.** The aim of the analyses was always to infer which layer of AlexNet the simulation was based on. **a**: Stimuli are chosen randomly from ecoset [49] and we simulate a simple rapid event related experimental design. **b**: "True" average response per voxel to a stimulus are based on local averages of the internal representations of Alexnet. To simulate the response of a voxel to a stimulus we choose a (x,y)-position uniformly randomly and take a weighted average of the activities around that location. As weights we choose a Gaussian in space and independently draw a weight per feature between 0 and 1. **c**: To generate a simulated voxel timecourse we generate the undistorted timecourses of voxel activities, convolve them with a standard hemodynamic response function and add temporally correlated normal noise. **d**: To estimate the response of a voxel to a stimulus we estimate a standard GLM to arrive at a noisy estimate of the true channel responses we started with in C. **e**: From the estimated channel responses we compute the stimulus by stimulus dissimilarity matrices. These dissimilarity matrices can then be compared to the dissimilarity matrices computed based on the full deep neural network representations from the different layers.

model performances. The following simulations serve to assess under more realistic conditions whether our methods correctly estimate the variance of the model performances.

2.2.2 Evaluation criteria for inference procedures

To evaluate alternative inference procedures, we perform simulations that reveal (1) that the estimates of the uncertainty of the model-performance estimates are accurate, and (2) which model comparison methods are most sensitive to subtle differences between models. To measure whether our bootstrap methods correctly estimate the uncertainty of the model-performance estimates, we compute the relative uncertainty (RU). The RU is the standard deviation of the bootstrap distribution of model-performance estimates σ_{boot} divided by the true standard deviation of model-performance estimates σ_{true} as observed over repeated simulations:

$$\mathrm{RU} = \frac{\sigma_{boot}}{\sigma_{true}} = \sqrt{\frac{\frac{1}{N}\sum_{i=1}^{N}\sigma_i^2}{\sigma_{true}^2}}, \tag{1}$$

where σ_i^2 is the variance estimator of the bootstrap in simulated data set i of the N simulations. Ideally, we would like the bootstrap-estimated variance to match the true variance such that the RU is 1.

To measure how well our analysis discriminates the different models (e.g. layers of a deep neural network), we use a signal-to-noise ratio (SNR). The signal is the magnitude of model-performance differences, which is measured as the variance across models of their average of performance estimates across simulations. The noise is the nuisance variation, which includes subject and condition sample variation along with measurement noise. The noise is measured as the average across models of the variance of performance estimates across simulations. This results in the following formula, in which $\mathrm{Perf}_{i,m}$ is the performance of model m of M in repetition i of N repetitions of the simulation:

$$\mathrm{SNR} = \frac{\mathrm{Var}_m\left(\frac{1}{N}\sum_{i=1}^{N}\mathrm{Perf}_{i,m}\right)}{\frac{1}{M}\sum_{i=1}^{M}\mathrm{Var}_i(\mathrm{Perf}_{i,m})}, \tag{2}$$

A higher signal-to-noise ratio indicates greater sensitivity to differences in model performance: differences between models are larger relative to the variation of model-performance estimates over repeated simulations. Note that this measure does not depend on the accuracy of the bootstrap because the bootstrap estimates of the variances do not enter this statistic. The SNR exclusively measures how large differences between models are compared to the level of nuisance variation we simulate, which may include random sampling of conditions, subjects, or both (in addition to measurement noise).

2.2.3 Validity of generalization to new subjects and conditions

To test whether our inference methods correctly generalize to new subjects and conditions, we performed a simulation that includes random sampling of both subjects and conditions (Fig 2). We used the internal representations of the deep convolutional neural network model AlexNet [47] to generate fMRI-like simulated data. In each simulated scenario, one of the layers of AlexNet served as the true (data-generating) model, while all layers were considered as candidate models in the inferential model comparisons. We simulated true voxel responses as local averages of the activities of close-by units in the feature maps of layers of the model. The response of each simulated voxel was a local average of unit responses, weighted according to a 2D Gaussian kernel over the locations of the feature map multiplied by a vector of nonnegative random weights (drawn uniformly from the unit interval) across the features. We then simulated hemodynamic-response time courses and added measurement noise. The covariance structure of the noise was determined by the overlap of the simulated voxels' averaging regions over space and a first-order auto-regressive model over time. The simulated data were subjected to a standard general linear model (GLM) analysis to estimate the condition-response matrix. Variation over conditions was generated by using randomly sampled natural images from ecoset [49] as input to the AlexNet model. Variation over subjects was generated by randomly choosing a new location and a new vector of feature weights for each voxel of a new simulated subject.

We simulated $N = 100$ datasets for each parameter setting to estimate how variable the model-performance estimates truly are. In analysis, we must estimate our uncertainty about model performance from a single dataset. To estimate how accurate these estimates were, we compared the uncertainty estimates used by different inference procedures (including different bootstrap methods) to the true variability. This comparison is a better check that our inference is accurate than false-positive rates of the model comparison tests, because our simulations do not contain situations that correspond to the H_0 of two different models with equal performance. As expected, the false-positive rates for tests against the true model were very low (not shown) whenever the type of bootstrap matches the desired level of generalization because the true layer has a higher average performance than the other models. At the 5% significance level, the proportion of cases where any other layer performed significantly better than the true (data-generating) layer was only 1.524% and tests against the best other layer (chosen based on all data) significantly favor this other layer in only 0.694% of cases. Multiple comparison correction would reduce these false-positive rates even further.

To test generalization to either across conditions or across subjects, we kept the other dimension constant in some of our simulations. In these simulations, the true variance across conditions is overestimated by bootstrap resampling of conditions (rendering the inference conservative) when we have less than about 40 conditions (Fig 3 a). The true variance across subjects is underestimated by bootstrap resampling of subjects (invalidating the inference) when we have very few subjects (Fig. 3 b). This downward bias corresponds to the $\frac{n}{n-1}$ factor between the sample variance and the unbiased estimate for the population variance and can thus easily be corrected. For experiments with 20 or more subjects and 40 or more conditions, the bootstrap variance estimates are fairly accurate (Fig. 3 a and b).

To test our novel bootstrap method to generalize to subjects and conditions simultaneously, we varied both stimuli and voxel sampling in our simulations. The corrected variance estimate reflects the overall variation caused by random sampling of subjects and conditions and by measurement noise much more accurately than the uncorrected estimate (Fig. 3 c). Due to the true difference in model performance these statistics are more informative than false positive rates, which remained low for this generalization as for the overall numbers mentioned above. Cases where a wrong model significantly outperformed the true model occurred in only 0.3 % of simulations with the corrected two-factor bootstrap even without any multiple comparison correction.

Overall, we found that our two-factor bootstrap method yields accurate estimates of the variance across the simulated populations of subjects and conditions when the dataset is large enough ($\geq$ 20 subjects, $\geq$ 40 conditions) and the type of bootstrap matches the desired level of generalization.

When looking at the model-discriminative signal-to-noise ratio, we first observe that increasing the number of measurements always increases the power of model-comparative inference. The power increases with the amount of data according to a power law (straight line in log-log plot; Fig. 3 d-f). Such a relationship holds whether we increase the number of conditions, the number of subjects, or the number of repetitions per condition. This result is expected and validates the signal-to-noise ratio as a measure of experimental power. In general, increasing the number of measurements helps most for the factor over which generalization is harder. For example, in our deep neural network

based simulations, the variability over subjects is smaller than the variability across conditions (Fig. 3 g). In this simulation, it thus increases statistical power more to collect data for more conditions. When there is more variability across subjects, the opposite might be true. An intermediate voxel size (Gaussian kernel width) yielded the highest model discriminability as measured by the SNR (Fig. 3 h, see Online methods 5.2.3 for more discussion on this topic).

2.2.4 Power afforded by different RDM comparison measures for model comparisons

An important question is how to measure RDM prediction accuracy for model evaluation. We ran the same analysis with different RDM comparison measures on the same datasets in a separate simulation (Online methods 5.2.2).

We found that different types of rank correlation are all similarly good at discriminating models (Fig. 4 c). Proper evaluation of models predicting tied dissimilarities requires Kendall's τ_a [45] or ρ_a, a rarely used variant of Spearman's rank correlation coefficient without correction for ties, analogous to Kendall's τ_a (derivation in Supplementary information 5.1.2). We recommend ρ_a over τ_a for its lower computational cost and analytically derived noise ceiling.

If we are willing to assume that the representational dissimilarity estimates are on an interval scale, we expect to be able to achieve greater model discriminability with RDM comparison measures that are not just sensitive to ranks. In this context, we compare the Pearson correlation and cosine similarity and whitened RDM comparison measures, which we introduced recently [52]. The whitened measures boost the power of inferential model comparisons, by accounting for the anisotropic sampling distribution of RDM estimators. To further increase our model-comparative power, both the whitened and the unwhitened cosine similarity assume a ratio-scale for the representational dissimilarities, which requires that indistinguishable conditions have an expected dissimilarity of zero. This assumption is justified when using a crossvalidated distance estimator [45, 46], which provides unbiased dissimilarity estimates with an interpretable zero point.

Consistent with the theoretical expectations, we observe greatest model discriminability for the whitened cosine similarity, which assumes ratio-scale dissimilarities, intermediate discriminability for the whitened Pearson correlation, and somewhat lower model discriminability for the unwhitened Pearson correlation and the unwhitened cosine similarity. Rank correlation coefficients performed surprisingly well, matching or even outperforming unwhitened Pearson correlation and unwhitened cosine similarity (Fig. 4 c). They provide an attractive alternative to the whitened criteria when researchers wish to make weaker assumptions about their model predictions.

2.2.5 Validation of inference on flexible models

To test our method for inference on flexible models, we made a variant of the deep neural network simulation in which we do not assume to know the range within which voxels average local neural responses. This leaves two parameters of the models to be fitted: the size of the voxels' averaging pools and how the different features (model unit activities) at each retinotopic location are reflected in the voxel responses. In the simulated truth, we set the spatial weights for each voxel to a Gaussian with a standard deviation of 5% of the image size (FWHM $\approx 11.77\%$) and randomly weighted the feature maps with a weight vector drawn independently for each voxel and feature, uniformly at random from the unit interval (details in Online Methods 5.2.2).

We then used models that a researcher could generate without knowing the ground truth of how voxels average local features. As building blocks for the models, we computed RDMs for different voxel averaging pool sizes and for different methods to deal with averaging across feature maps. To capture voxel averaging across retinotopic locations, we smoothed the feature maps with Gaussians of different sizes. To capture voxel averaging across feature maps, we (1) generated RDMs computed after taking the average across feature maps at each location ('avg'), (2) computed the expected RDM for the weight sampling implemented in the simulation ('weighted'), or (3) computed RDMs without any feature-map averaging ('full').

We combined these building blocks into two types of flexible model: *selection models* and *nonnegative linear-combination models*. In a selection model, fitting is implemented as selection of the best among a finite set of RDMs. Here we defined one selection model for each method of combining the feature maps. Each selection model contained RDMs computed for different sizes of the local averaging pool. In linear-combination models, fitting consists in finding nonnegative weights for a set of basis RDMs, so as to maximize RDM prediction accuracy. The RDMs contain estimates of the squared Mahalanobis distances, which sum across sets of tuned neurons that jointly form a population code. As component RDMs we chose the four extreme cases of RDM generation: no pooling across space or averaging across the whole image, each paired with either 'full' or 'avg' treatment of the feature maps. The resulting four-RDM-component linear model approximates the effect of computing the RDM from voxels that reflect the average activity over retinotopic patches of different sizes [54]. For the averaging across feature maps, which uses random weights, there is a strong motivation for using a linear model: When the voxel activities are weighted averages of the underlying neurons with the weights drawn independently from the same distribution, the expected squared Euclidean RDM is exactly a linear

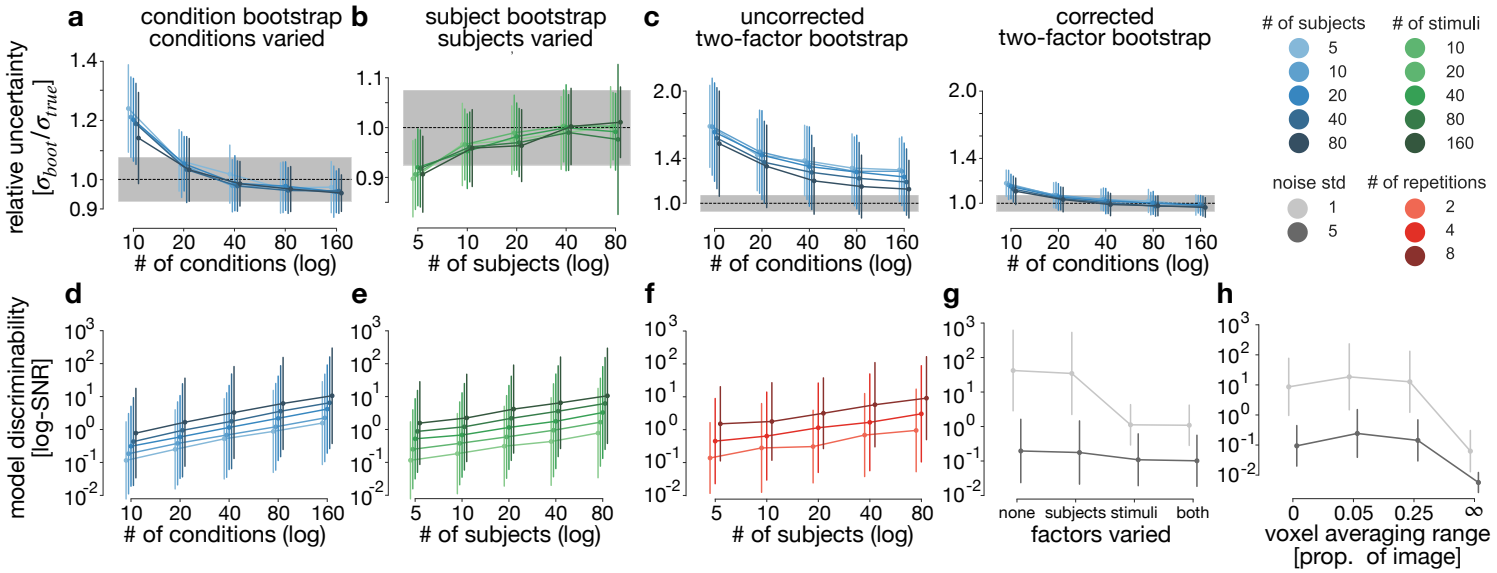

Figure 3: **Results of the deep-neural-network-based simulations. a-c**: Relative uncertainty, i.e. the bootstrap estimate of the standard deviation of model-performance estimates divided by the true standard deviation over repeated simulations. Dashed line and grey box indicate the expected value and standard deviation due to the number of simulations per condition. **a**: Bootstrap resampling of conditions when repeated simulations use random samples of conditions and a fixed set of subjects. **b**: Bootstrap resampling of subjects when simulations use random samples of subjects (simulated voxel placements) and a fixed set of conditions. **c**: Direct comparison of the uncorrected and corrected two-factor bootstraps (see 5.1.4 for details) for simulations that varied both conditions and subjects. **d-h**: Signal-to-noise ratio (Eq. 2), a measure of sensitivity to differences in model performance, for the different inference procedures and simulated scenarios. Infinite voxel averaging range refers to voxels averaging across the whole feature map.

combination of the RDM computed based on the univariate population-average responses and the RDM based on all neurons (Online methods 5.1.7). For comparison, we also included fixed RDM models, corresponding to component RDMs of the fitted models.

We found that our bootstrap-wrapped crossvalidation with the two-factor bootstrap and the excess-variance correction yielded accurate estimates of the uncertainty. The relative uncertainties were close to 1 (Fig. 4 e). The model discriminability (SNR) was primarily determined by how accurately the different models were able to recreate the true measurement model (Fig. 4 f). The highest SNRs were achieved when the assumed model matched exactly (weighted feature treatment and voxel size 0.05), but the model variants which allowed for some fitting still yield high SNRs. Analyses that take the averaging across space and features into account yielded the highest average model performance for the true model. In contrast, analyses that ignore averaging over space or features (the full feature set selection model and some of the fixed models) not only lead to lower SNRs (as seen in Fig. 4 f), but also systematically selected the wrong layer yielding a higher average performance for a different layer than the one we used for generating the data (not visible in the figure).

In situations when the true voxel sampling is unknown, flexible models that are allowed to fit the voxel sampling to data can thus be used effectively to determine the underlying representation. In contrast, fixed, wrong assumptions about the voxel sampling can lead to low model discriminability (SNR), and even to consistently wrong conclusions (not shown).

2.2.6 Validation with functional MRI data

The simulations we presented so far enabled us to test all statistical inference we perform. To test our methods on real data, we decided to re-sample data from a large openly available fMRI experiment in which humans viewed pictures from ImageNet [60]. These data contain various noise sources, individual differences, signal shapes, and distributions that are difficult to simulate accurately. We used a data-based simulation to create realistic synthetic data, whose ground-truth RDM we knew (Fig. 5). By subsampling from this dataset, we generated smaller datasets to test inference with bootstrapping over stimuli. We used the entire dataset as a stand-in for the population a researcher might wish to generalize to. For each brain region, we computed the “true" mean RDM using all data (all runs and subjects). Each of

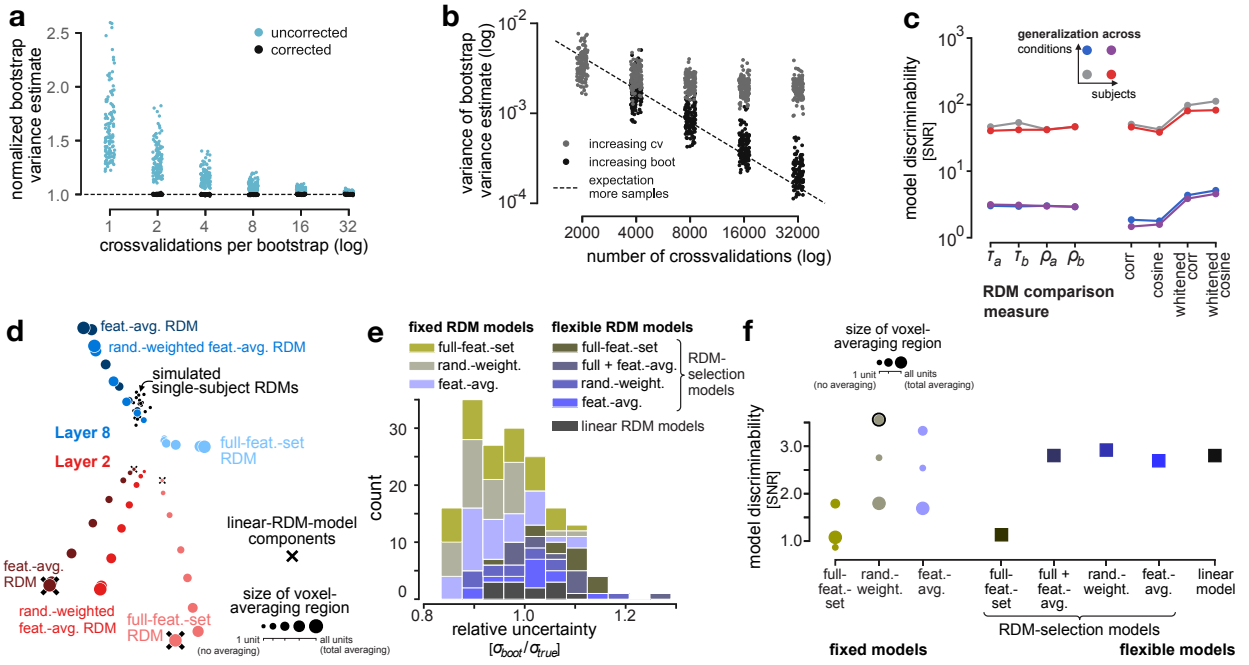

Figure 4: **Flexible model tests using bootstrap-crossvalidation a**: Unbiased estimates of the variance of model performance estimates (dashed line) require either many crossvalidation cycles (light blue dots) or the proposed correction formula (back dots). Each model in each simulated dataset contributes one dot to each point cloud in this plot corresponding to the average estimated variance across 100 repeated analyses. All variance estimates of a model are divisively normalized by the average corrected variance estimate for this model over all numbers of crossvalidation cycles for the dataset. For many crossvalidation cycles, the uncorrected and corrected estimates converge, but the correction formula yields this value even when we use only two crossvalidation cycles (Online methods, 5.1.4). **b**: Variance of the corrected bootstrap variance estimate across multiple estimations on the same dataset comparing the use of more crossvalidation folds per bootstrap sample (gray, $2, 4, 8, 16, 32$ crossvalidations at 1000 bootstrap samples) to using more bootstrap samples (black, $1000, 2000, 4000, 8000, 16000$ bootstrap samples with 2 crossvalidations per sample). More bootstrap samples are more efficient at stabilizing our bootstrap estimates of the variance of model performance estimates. Increasing the number of bootstraps decreases the variance roughly at the $N^{-\frac{1}{2}}$ rate expected for sampling approximations indicated by the dashed line. **c**: Results of a separate simulation comparing different RDM comparison measures for fixed models on the same collection of data sets. Model discriminability (signal-to-noise ratio) is highest for whitened cosine RDM similarity, which requires a crossvalidated distance estimator, and reasonable for rank-correlation coefficients.**d**: MDS arrangement of the RDMs for one simulated dataset. Colored circles show the predictions based on one correct and one wrong layer changing the voxel averaging region and the treatment of features ('full', 'weighted', & 'avg', as described in the text). Fixed models correspond to single choice of model RDM for each Layer. Selection models select the best fitting voxel size from the RDMs presented in one color (or two for 'both'). Crosses mark the four components of the linear model for Layer 2. The small black dots represent simulated subject RDMs without fMRI noise. **e**: Histogram of relative uncertainties $\sigma_{boot}/\sigma_{true}$, showing that the bootstrap-wrapped crossvalidation accurately estimates the variance of the performance estimates across many different inference scenarios. **f**: Signal-to-noise ratios showing the model discriminabilities for the same models as in D. The black circle marks the model used for data simulation. For details on the simulations, see Online Methods 5.2.2.

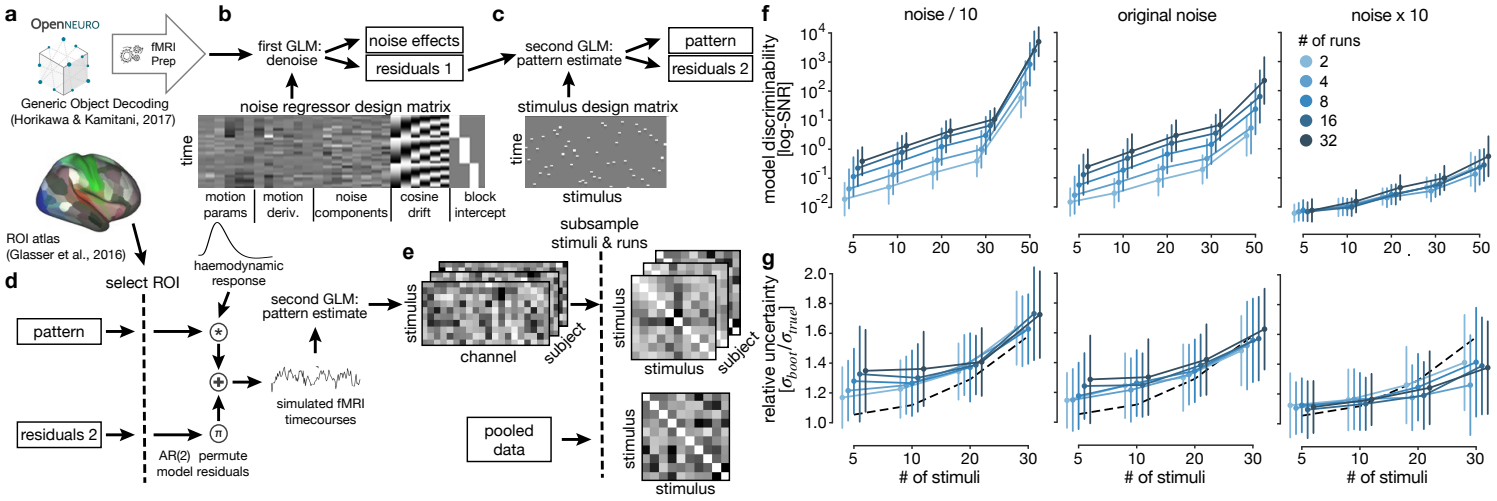

Figure 5: **fMRI-data based simulation a**: These simulations are based on a dataset of neural recordings for 50 stimuli in 5 human observers [60], which were each shown 35 times. To extract stimulus responses from these data we perform 2 GLM steps as in the original publication. **b**: In the first step we regress out diverse noise estimators (provided by fMRIprep) from pooled fMRI runs. **c**: We then apply a second GLM separately on each run to extract the stimulus responses. **d**: We then extract ROIs based on an atlas [61], randomly chose differently sized subsets of runs resp. stimuli to enter further analyses. To simulate realistic noise, we estimate an AR(2) model on the second GLM's residuals, permute and filter them to keep their original autocorrelation structure, and finally scale them by the factors 0.1, 1, and 10. To generate simulated timecourses, we add these altered residuals to the GLM prediction. We then rerun the second GLM on the simulated data and use the Beta-coefficient maps for following steps. **e**: Finally, we compute crossnobis RDMs and perform RSA based on the overall RDM across all subjects. **f**: Results of the simulations, separately for each noise scaling factor. The signal-to-noise ratio shows the same increase as for our abstract simulation with an additional bonus when all 50 stimuli are sampled such that there is no more random selection. **g**: The relative uncertainty shows a fairly close agreement with the increase predicted due to sampling without replacement (indicated by the dashed line).

these ground-truth RDMs for different brain regions served as a model RDM. The model comparison we attempted aims to recover which brain area a dataset was sub-sampled from. This method allows us to check whether our uncertainty estimates are correct for a range of model performances.

We varied the strength of noise, the number of runs and the number of stimuli (i.e. viewed images) but not the number of subjects as the original dataset contains only 5 (prohibiting informative resampling of subjects). To increase the variability of the resampled datasets beyond sampling from the 35 measurement runs and to vary the noise strength, we created a new timecourse for each sampled run while preserving the spatial structure and serial autocorrelation of the noise: For this we estimated a second-order autoregressive model ($AR(2)$) separately for each run's GLM residuals, permuted the AR-model's residuals and added the results back to the GLM's predicted timecourse (see Fig. 5 a-e and Online Methods 5.2.4 for details). We repeated each simulated experiment 24 times and used the relative uncertainty and the model-discriminability SNR as our evaluation criteria.

Results were largely similar to those of the neural-network-based simulations (Fig. 5 f & g). For the relative uncertainty, which measures the accuracy of our bootstrap variance estimates, we see a convergence towards the expected ratio (dashed line), which is no longer 1, because we sample stimuli without replacement in our simulations. The variance of our evaluations will thus be smaller than expected for sampling from the infinite population by a factor of approximately $1 - \frac{n}{N}$ and the expected relative uncertainty becomes the inverse of the square root of this value, which we indicate with a dashed line in the figure. For the model-discriminability SNR, we find the same power-law increase with the number of conditions and the number of runs used as data, while there is an additional jump, when we sample the stimuli exhaustively, which entirely eliminates this noise source. Also, we see that increasing the noise magnitude drastically reduces the SNR.

2.2.7 Validation with calcium imaging data

We can also adjudicate among models of the representational geometry on the basis of direct neural measurements, such as electrophysiological recordings or calcium imaging data. These measurement modalities have drastically different statistical properties than fMRI. To test our methods for this kind of data, we performed a re-sampling simulation

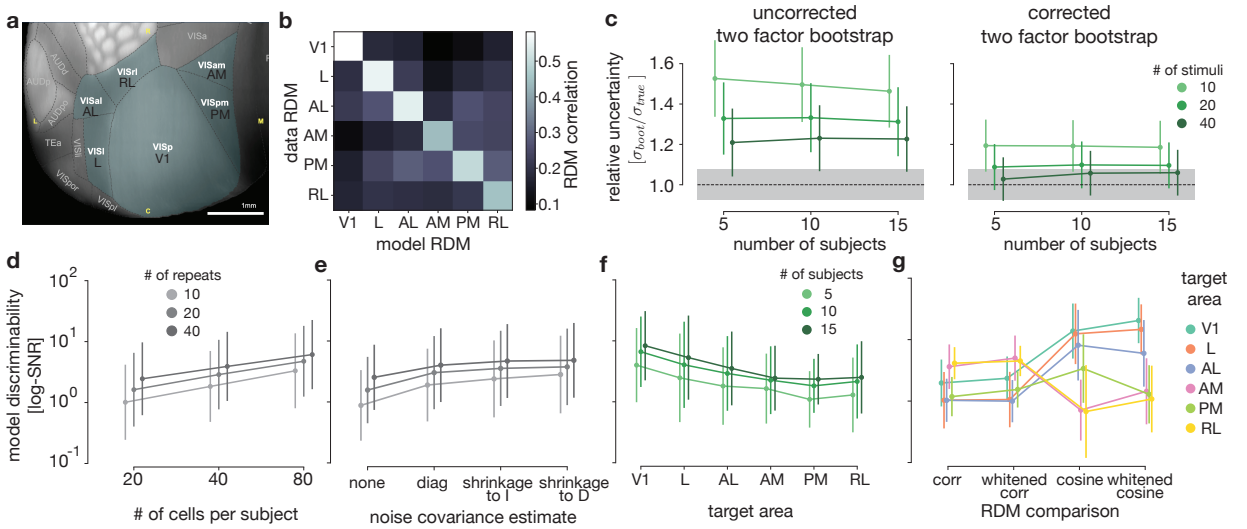

Figure 6: **results in mice with calcium imaging data**. **a**: Mouse visual cortex areas used for analyses and resampling simulations (Image credit: Allen Institute). **b**: Overall similarities of the representations in different cortical areas in terms of their RDM correlations. For each mouse and cortical area ("data RDM", vertical), the RDM was correlated with the average RDM across all other mice ("model RDM", horizontal), for each other cortical area. We plot the average across mice of the crossvalidated RDM correlation (leave-one-mouse-out crossvalidation). The prominent diagonal shows the replicability across mice and the distinctness between cortical areas of the representational geometries. **c**: Relative uncertainty for the two-factor bootstrap methods. The gray box indicates the range of results expected from simulation variability if the bootstrap estimates were perfectly accurate. The correction is clearly advantageous here although the method is still slightly conservative (overestimating the true standard deviation σ_{true} of model-performance evaluations) for small numbers of stimuli. For 40 or more stimuli, the corrected two-factor bootstrap correctly estimates the variance of model-performance evaluations. **d**: Signal to noise ratio validation: The SNR grows with the number of cells per subject and the number of repeats per stimulus. **e**: Signal to noise ratio for different noise covariance estimates. Taking a diagonal covariance estimate into account, i.e. normalizing cell responses by their standard deviation is clearly advantageous. The shrinkage estimates provide a marginal improvement over that. **f**: Signal to noise ratio for data sampled from different areas. **g**: Which measure is optimal for discriminating the models depends on the data generating area. On average there is an advantage of the cosine similarity over the RDM correlation and of the whitened measures over the un-whitened ones.

based on a large calcium-imaging dataset of responses of mouse visual cortex to natural images [62]. This dataset contains recordings from six visual brain areas: primary visual cortex (V1), lateralintermediate (LM), posteromedial (PM), rostrolateral (RL), anteromedial (AM), and anterolateral (AL) visual area (Fig. 6 a).

As in the previous section, we used the overall pooled RDMs for the different areas as models and re-sampled different numbers of stimulus repetitions, neurons, mice, and stimuli to vary the amount of information afforded by the experiment. We repeated each simulated experiment 100 times and used the relative uncertainty and the model-discriminability SNR as our evaluation criteria to assess the validity of our methods and to determine which methods for computing and comparing RDMs are most effective.

First, we analyzed the overall discriminability of the brain areas (Fig. 6 b). Using all data, the different brain areas identified by the dataset providers can be discriminated well, although the areas vary in the reliability of the estimated RDMs (Fig. 6 b & f).

Second, we used the relative uncertainty to test whether our variance estimates are correct for this data. For this simulation, we always resampled all factors (subjects, stimuli, runs and cells). Correspondingly, we used bootstrapping over both subjects and stimuli. We observed that the corrected two-factor bootstrap is substantially more accurate than the uncorrected two-factor bootstrap, the latter being overly conservative, substantially overestimating the true variance (Fig. 6 c).

Third, to understand how the model-comparative power depends on the scenario, we analyzed the model-discriminability SNR. We found that more subjects, more stimuli, more runs, and more cells all increased the SNR just as in our fMRI and neural-network-based simulations (Fig. 6 d). Furthermore, we find that taking the noise covariance into account for

computing the crossnobis RDMs in the first-level analysis improves the signal-to-noise ratio (Fig. 6 e). Univariate noise normalization (implemented as a diagonal noise covariance matrix) is better than no noise normalization. Multivariate noise normalization is better than univariate noise normalization. For multivariate noise normalization, we tested two different shrinkage estimators with different targets: a multiple of the identity and the diagonal matrix of variances. These two variants perform similarly. Beyond that, we find that different brain areas and thus different true RDMs yield different performance patterns across the different RDM comparison methods (Fig. 6 g). For some, cosine similarity performs better, for some Pearson correlation works better and while the whitened performance measures are better on average there are also cases where the un-whitened measures perform slightly better. Thus, it remains dependent on the concrete experiment, which measure performs best at comparing RDMs.

3 Discussion

We present new methods for inferential evaluation and comparison of models that predict representational geometries. The inference procedures enable simultaneous generalization to new subjects and new conditions, treat flexible models correctly using crossvalidation, and use RDM comparison measures whose power approaches the theoretical limit according to the Neyman-Pearson Lemma [52]. For fixed as well as flexible models, our inference methods support all combinations of generalization: to new subjects, to new conditions, neither, or both. We validated the methods using simulated data as well as calcium imaging and fMRI data, showing that the inference is reliable. The methods are available as part of an open-source Python toolbox.

3.1 Generalizing to new subjects and/or new conditions

Generalization to neither subjects nor conditions means that we are drawing conclusions that are expected to hold only for replications of the experiment in the same animals using the same conditions. Generalization to new subjects may not be possible, for example, in case studies or when the number of animals (e.g. two macaques) is insufficient. Generalization to new conditions is not needed when all conditions relevant to our claims have been sampled. For example, [63] studied the representational similarity of finger movements in primary motor cortex. All five fingers were sampled in the experiments and there are no other fingers to generalize to. When generalizing to replications with the same subjects and conditions, we need separate data partitions to estimate the variability of the model performance estimates. We can then use classical tests (e.g. t-test or rank-sum test) to test for significant differences between models.

If generalization to the population of subjects is desired, we need a sufficiently large sample of subjects. We can then evaluate each model for each subject and use classical t-tests or rank-sum tests, which treat subject as a random sample from a population. These tests are valid, controlling false-positive rates at their nominal values in simple simulations (Online methods 5.1.5). The variance across subjects then is a good estimate of the variance across the population of subjects, as long as we do not need to generalize beyond the exact set of experimental conditions used in the experiment.

We often would like our inference to generalize to new conditions drawn from the same population of conditions as the experimental conditions. For example, when evaluating computational models of vision, we are not usually interested in determining which models dominate just for the particular visual stimuli presented in our experiment. We are interested in models that dominate for a population of similar stimuli. Since RDM prediction accuracy cannot be assessed for single conditions, we require a bootstrapping approach. We bootstrap-resample the conditions set and evaluate all models on each sample. This procedure correctly estimates the variability of results for these settings and t-tests based on the estimated variances provide valid frequentist tests.

If we want to generalize simultaneously across conditions and subjects, then the corrected two-factor bootstrap approach (Online methods 5.1.4) provides accurate estimates of our uncertainty about model performances. These uncertainty estimates support valid inferential model comparisons, which we expect to generalize to new subjects and conditions drawn from the respective populations. Using t-tests based on the estimated variances, we can again test all comparisons of interest.

3.2 Inference on fixed and flexible models

If we are interested in flexible models, then our performance evaluation must not be biased by overfitting. To avoid this bias, we use a novel crossvalidation scheme that enables us to evaluate models' predictive accuracy when simultaneously generalizing to new subjects and/or new conditions. This crossvalidation is nested in our bootstrap procedure for estimating uncertainty. By using two crossvalidation runs for each bootstrap sample, we can accurately remove the excess variance introduced by crossvalidation. These methods provide a computationally efficient estimate of the variances and covariances of model performances for flexible models, which enables us to use a t-test to inferentially compare models to each other, to chance performance, and to the lower bound of the noise ceiling.

The methods are fully general in that inference can be performed on any model for which the user provides a fitting and an RDM prediction method. In practice, the complexity of the models is limited by the requirement that we need to fit each model thousands of times to randomly drawn data. Thus, we need a sufficiently fast and reliable fitting method for the model.

If fitting the model so often is not feasible or if the data RDMs do not provide sufficient constraints, one solution is to fit all models using a separate training set of neural data before the inferential analyses. This approach is appropriate when many parameters are to be fitted, as is the case in nonlinear systems identification approaches as well as encoding models [64], where a large set of neural training data is required. All conclusions are then conditional on the training set, i.e. the tested generalization is towards testing the same models fitted on the same training data evaluated on new test data. Our methods offer fitting of lower-parametric models as part of the model-comparative inference, obviating the need for a separate neural dataset for fitting in many scenarios.

3.3 How many subjects, conditions, repetitions, and measurement channels?

Statistical inference gains power when more data are collected along any dimension. More independent measurement channels, more subjects, more conditions and more repetitions all help. How much data is needed along each of these dimensions depends on the experiment. The most helpful dimension to extend is the one that currently limits generalization. When crossvalidation across repeated measurements (i.e. crossnobis RDMs) is used to eliminate the bias of the distance estimates, using more repetitions brings an additional performance bonus because it reduces the variance contributed by the unbiased estimates [52, Online methods 5.2.3].

3.4 Which distance estimator and RDM comparison measure?

The statistical inference procedures introduced here work for any choice of distance estimator and RDM comparison measure. However, the choice of representational-distance estimator and RDM comparison measure affects the power of model comparative inference and the meaning of the inferential results.

For computing the RDM, we tested only variations of the crossnobis (crossvalidated Mahalanobis) distance estimator, as recommended based on earlier research [46]. The crossnobis estimator can use different noise covariance estimates to normalize patterns, such that the noise distribution becomes approximately isotropic. The noise covariance matrix can be the identity (no normalization), diagonal (univariate normalization), or a full estimate (multivariate normalization). Consistent with previous findings, our results suggest that univariate noise normalization is always preferable to no normalization, and that multivariate noise normalization using a shrinkage estimate of the noise covariance [65, 66] helps in some circumstances and never hurts model discrimination.

For evaluating RDM predictions, we can distinguish RDM comparison methods by the scale they assume for the distance estimates: ordinal, interval, or ratio. For ordinal comparisons, the different rank correlation coefficients perform similarly. We recommend ρ_a for its computational efficiency and analytically derived noise ceiling. For interval- and ratio-scale comparisons, a more complex pattern emerges. In particular whether cosine similarities (ratio scale) or Pearson correlations (interval scale) work better depends on the structure of the model RDMs to be compared. We recently proposed whitened variants of the cosine similarity and Pearson correlation, which take into account that the distance estimates in an RDM are not independent [52]. The whitened RDM comparison measure were more sensitive to subtle differences in model performance when evaluated on fixed models (Fig. 4 c). In the simulations based on the calcium imaging data, whitened RDM comparison measures still performed better on average, but there were brain areas that were easier to identify by using the unwhitened comparison measures.

3.5 Alternative approaches

We present a frequentist inference methodology that uses crossvalidation to obtain point estimates of model performance and bootstrapping to estimate our uncertainty about them. Bayesian alternatives deserve consideration. For example, a Bayesian approach has been proposed to alleviate the bias of distance estimates [51]. This Bayesian estimate makes more detailed assumptions about the trial dependencies than our crossvalidated distance estimators, which remove the bias. The Bayesian estimate might be preferable for its higher stability when its assumptions hold and could be used in combination with our model-comparative inference methods. For model comparisons, Bayesian inference is also an interesting alternative to the frequentist methods we discuss here [54]. Our whitened RDM comparison methods can be motivated as approximations to the likelihood for a model and we reported recently that they afford similar power as likelihood-based inference with normal assumptions [52]. Thus, frequentist inference using the whitened RDM comparison measures is related to Bayesian inference with a uniform prior across models. In the Bayesian framework, generalization to the populations of subjects and conditions would require a prior model of how RDMs vary across

subjects and conditions though, which we are currently lacking. Until such models and the Bayesian inference for them are developed, the frequentist methods we present here remain the only method for generalization to the populations of subjects and conditions.

Another strongly related method for comparing models to data in terms of their geometry is pattern component modeling [55], which compares conditions in terms of their co-variance over measurement channels instead of their representational dissimilarities. This approach is deeply related to representational similarity analysis [27]. Pattern component modeling is somewhat more rigid than RSA as the theory is based on normal distributions, but it has advantages in terms of analytical solutions. In particular, the likelihood of models can be directly evaluated and likelihood-ratio tests performed. Due to the direct evaluation of likelihoods, this framework can be combined with Bayesian inference more easily and recently a variational Bayesian analysis was presented for this model [67].

Another powerful approach to inference on brain-computational models is to fit encoding models that predict measured brain-activity data instead of representational geometries [e.g. 64, 68, 69, 70, 71, 27, 72]. This approach was originally developed in the context of low-dimensional models and measurements. When models and measurements are both high dimensional, even a linear encoding model can be severely under-constrained [73, 74]. As a result, an encoding model requires a combination of substantial fitting data and strong priors on the weights. The predictive model that is being evaluated comprises the encoding model and the priors on its weights [27], which complicates the interpretation of the results [73, 75]. Both model performances and the fitted weights can then be highly uncertain and/or dependent on the details of the assumed encoding model. The additional data and assumptions needed to fit complex encoding models motivate the use of methods as proposed here that do not require fitting of a high-parametric mapping from model to measured brain activity.

4 Conclusion

We present a comprehensive new methodology for inference on models of representational geometries that enables neuroscientists to draw conclusions that generalize to new subjects and conditions, can handle flexible models, and is more powerful than previous approaches. The validity of the methods has been established through extensive simulations and using real neural data. These methods enable neuroscientists working with humans and animals to evaluate complex brain-computational models with measurements of neural population activity. As we enter the age of big models and big data, we hope these methods will help connect computational theory to neuroscientific experiment.

References

[1] Parvizi, J. and Kastner, S. (2018). Promises and limitations of human intracranial electroencephalography. *Nature neuroscience*, 21(4):474–483.

[2] Abbott, J., Ye, T., Krenek, K., Gertner, R. S., Ban, S., Kim, Y., Qin, L., Wu, W., Park, H., and Ham, D. (2020). A nanoelectrode array for obtaining intracellular recordings from thousands of connected neurons. *Nature Biomedical Engineering*, 4(2):232–241.

[3] Wang, T. and Xu, C. (2020). Three-photon neuronal imaging in deep mouse brain. *Optica*, 7(8):947.

[4] Allen, E. J., St-Yves, G., Wu, Y., Breedlove, J. L., Dowdle, L. T., Caron, B., Pestilli, F., Charest, I., Hutchinson, J. B., Naselaris, T., and Kay, K. (2021). A massive 7T fMRI dataset to bridge cognitive and computational neuroscience. Preprint, Neuroscience.

[5] Guo, Z., Wang, L., Ji, B., Xi, Y., Yang, B., and Liu, J. (2021). Flexible, multi-shank stacked array for high-density omini-directional intracortical recording. In *2021 IEEE 34th International Conference on Micro Electro Mechanical Systems (MEMS)*, pages 540–543.

[6] Uğurbil, K. (2021). Ultrahigh field and ultrahigh resolution fmri. *Current Opinion in Biomedical Engineering*, page 100288.

[7] Bandettini, P. A., Huber, L., and Finn, E. S. (2021). Challenges and opportunities of mesoscopic brain mapping with fMRI. *Current Opinion in Behavioral Sciences*, 40:189–200.

[8] Jun, J. J., Steinmetz, N. A., Siegle, J. H., Denman, D. J., Bauza, M., Barbarits, B., Lee, A. K., Anastassiou, C. A., Andrei, A., Aydın, Ç., et al. (2017). Fully integrated silicon probes for high-density recording of neural activity. *Nature*, 551(7679):232–236.

[9] Steinmetz, N. A., Koch, C., Harris, K. D., and Carandini, M. (2018). Challenges and opportunities for large-scale electrophysiology with neuropixels probes. *Current opinion in neurobiology*, 50:92–100.

[10] Baillet, S. (2017). Magnetoencephalography for brain electrophysiology and imaging. *Nature Neuroscience*, 20(3):327–339.

[11] Craik, A., He, Y., and Contreras-Vidal, J. L. (2019). Deep learning for electroencephalogram (EEG) classification tasks: A review. *Journal of Neural Engineering*, 16(3):031001.

[12] Kriegeskorte, N. (2015). Deep neural networks: a new framework for modeling biological vision and brain information processing. *Annual review of vision science*, 1:417–446.

[13] Kell, A. J., Yamins, D. L., Shook, E. N., Norman-Haignere, S. V., and McDermott, J. H. (2018). A task-optimized neural network replicates human auditory behavior, predicts brain responses, and reveals a cortical processing hierarchy. *Neuron*, 98(3):630–644.e16.

[14] Kubilius, J., Schrimpf, M., Kar, K., Rajalingham, R., Hong, H., Majaj, N., Issa, E., Bashivan, P., Prescott-Roy, J., Schmidt, K., Nayebi, A., Bear, D., Yamins, D. L., and DiCarlo, J. J. (2019). Brain-like object recognition with high-performing shallow recurrent anns. In Wallach, H., Larochelle, H., Beygelzimer, A., d' Alché-Buc, F., Fox, E., and Garnett, R. (Editors), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

[15] Zhuang, C., Yan, S., Nayebi, A., Schrimpf, M., Frank, M. C., DiCarlo, J. J., and Yamins, D. L. K. (2021). Unsupervised neural network models of the ventral visual stream. *Proceedings of the National Academy of Sciences*, 118(3):e2014196118.

[16] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d' Alché-Buc, F., Fox, E., and Garnett, R. (Editors), *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

[17] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

[18] LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

[19] Stevenson, I. H. and Kording, K. P. (2011). How advances in neural recording affect data analysis. *Nature Neuroscience*, 14(2):139–142.

[20] Sejnowski, T. J., Churchland, P. S., and Movshon, J. A. (2014). Putting big data to good use in neuroscience. *Nature Neuroscience*, 17(11):1440–1441.

[21] Smith, S. M. and Nichols, T. E. (2018). Statistical Challenges in "Big Data" Human Neuroimaging. *Neuron*, 97(2):263–268.

[22] Kriegeskorte, N. and Douglas, P. K. (2018). Cognitive computational neuroscience. *Nature neuroscience*, 21(9):1148–1160.

[23] Shepard, R. N. and Chipman, S. (1970). Second-order isomorphism of internal representations: Shapes of states. *Cognitive psychology*, 1(1):1–17.

[24] Edelman, S., Grill-Spector, K., Kushnir, T., and Malach, R. (1998). Toward direct visualization of the internal shape representation space by fmri. *Psychobiology*, 26(4):309–321.

[25] Edelman, S. (1998). Representation is representation of similarities. *The Behavioral and brain sciences*, 21(4):449.

[26] Norman, K. A., Polyn, S. M., Detre, G. J., and Haxby, J. V. (2006). Beyond mind-reading: multi-voxel pattern analysis of fmri data. *Trends in cognitive sciences*, 10(9):424–430.

[27] Diedrichsen, J. and Kriegeskorte, N. (2017). Representational models: A common framework for understanding encoding, pattern-component, and representational-similarity analysis. *PLOS Computational Biology*, 13(4):e1005508.

[28] Kriegeskorte, N., Mur, M., Ruff, D. A., Kiani, R., Bodurka, J., Esteky, H., Tanaka, K., and Bandettini, P. A. (2008). Matching categorical object representations in inferior temporal cortex of man and monkey. *Neuron*, 60(6):1126–1141.

[29] Kriegeskorte, N., Mur, M., and Bandettini, P. A. (2008). Representational similarity analysis - connecting the branches of systems neuroscience. *Frontiers in Systems Neuroscience*, 2.

[30] Connolly, A. C., Guntupalli, J. S., Gors, J., Hanke, M., Halchenko, Y. O., Wu, Y.-C., Abdi, H., and Haxby, J. V. (2012). The representation of biological classes in the human brain. *Journal of Neuroscience*, 32(8):2608–2618.

[31] Xue, G., Dong, Q., Chen, C., Lu, Z., Mumford, J. A., and Poldrack, R. A. (2010). Greater Neural Pattern Similarity Across Repetitions Is Associated with Better Memory. *Science*, 330(6000):97–101.

[32] Khaligh-Razavi, S.-M. and Kriegeskorte, N. (2014). Deep supervised, but not unsupervised, models may explain it cortical representation. *PLoS computational biology*, 10(11):e1003915.

[33] Yamins, D. L., Hong, H., Cadieu, C. F., Solomon, E. A., Seibert, D., and DiCarlo, J. J. (2014). Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the National Academy of Sciences*, 111(23):8619–8624.

[34] Cichy, R. M., Pantazis, D., and Oliva, A. (2014). Resolving human object recognition in space and time. *Nature Neuroscience*, 17(3):455–462.

[35] Haxby, J. V., Connolly, A. C., and Guntupalli, J. S. (2014). Decoding neural representational spaces using multivariate pattern analysis. *Annual review of neuroscience*, 37:435–456.

[36] Freeman, J. B., Stolier, R. M., Brooks, J. A., and Stillerman, B. S. (2018). The neural representational geometry of social perception. *Current Opinion in Psychology*, 24:83–91. Social Neuroscience.

[37] Kietzmann, T. C., Spoerer, C. J., Sörensen, L. K., Cichy, R. M., Hauk, O., and Kriegeskorte, N. (2019). Recurrence is required to capture the representational dynamics of the human visual system. *Proceedings of the National Academy of Sciences*, 116(43):21854–21863.

[38] Stringer, C., Pachitariu, M., Steinmetz, N., Carandini, M., and Harris, K. D. (2019). High-dimensional geometry of population responses in visual cortex. *Nature*, page 1.

[39] Chung, S., Lee, D. D., and Sompolinsky, H. (2018). Classification and geometry of general perceptual manifolds. *Physical Review X*, 8(3):031003.

[40] Chung, S. and Abbott, L. (2021). Neural population geometry: An approach for understanding biological and artificial neural networks. *arXiv preprint arXiv:2104.07059*.

[41] Kriegeskorte, N. and Wei, X.-X. (2021). Neural tuning and representational geometry. *Nature Reviews Neuroscience*, 22(11):703–718.

[42] Kriegeskorte, N. and Diedrichsen, J. (2019). Peeling the onion of brain representations. *Annual review of neuroscience*, 42:407–432.

[43] Kriegeskorte, N. and Kievit, R. A. (2013). Representational geometry: integrating cognition, computation, and the brain. *Trends in Cognitive Sciences*, 17(8):401–412.

[44] Kriegeskorte, N., Mur, M., Ruff, D. A., Kiani, R., Bodurka, J., Esteky, H., Tanaka, K., and Bandettini, P. A. (2008). Matching Categorical Object Representations in Inferior Temporal Cortex of Man and Monkey. *Neuron*, 60(6):1126–1141.

[45] Nili, H., Wingfield, C., Walther, A., Su, L., Marslen-Wilson, W., and Kriegeskorte, N. (2014). A Toolbox for Representational Similarity Analysis. *PLOS Computational Biology*, 10(4):e1003553.

[46] Walther, A., Nili, H., Ejaz, N., Alink, A., Kriegeskorte, N., and Diedrichsen, J. (2016). Reliability of dissimilarity measures for multi-voxel pattern analysis. *NeuroImage*, 137:188–200.

[47] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105.

[48] Khaligh-Razavi, S.-M., Henriksson, L., Kay, K., and Kriegeskorte, N. (2017). Fixed versus mixed RSA: Explaining visual representations by fixed and mixed feature sets from shallow and deep computational models. *Journal of Mathematical Psychology*, 76:184–197.

[49] Mehrer, J., Kietzmann, T. C., and Kriegeskorte, N. (2017). Deep neural networks trained on ecologically relevant categories better explain human IT. *Conference on Cognitive Computational Neuroscience*.

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

[51] Cai, M. B., Schuck, N. W., Pillow, J. W., and Niv, Y. (2019). Representational structure or task structure? Bias in neural representational similarity analysis and a Bayesian method for reducing bias. *PLOS Computational Biology*, 15(5):e1006299.

[52] Diedrichsen, J., Berlot, E., Mur, M., Schütt, H. H., Shahbazi, M., and Kriegeskorte, N. (2020). Comparing representational geometries using whitened unbiased-distance-matrix similarity. *arXiv:2007.02789 [stat]*. ArXiv: 2007.02789.

[53] Khaligh-Razavi, S.-M. and Kriegeskorte, N. (2014). Deep Supervised, but Not Unsupervised, Models May Explain IT Cortical Representation. *PLOS Computational Biology*, 10(11):1–29.

[54] Kriegeskorte, N. and Diedrichsen, J. (2016). Inferring brain-computational mechanisms with models of activity measurements. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 371(1705):20160278.

[55] Diedrichsen, J., Yokoi, A., and Arbuckle, S. A. (2018). Pattern component modeling: A flexible approach for understanding the representational structure of brain activity patterns. *NeuroImage*, 180:119–133.

[56] Shahbazi, M., Shirali, A., Aghajan, H., and Nili, H. (2021). Using distance on the Riemannian manifold to compare representations in brain and in models. *NeuroImage*, page 118271.

[57] Yarkoni, T. (2020). The generalizability crisis. *Behavioral and Brain Sciences*, page 1–37.

[58] Efron, B. and Tibshirani, R. J. (1994). *An introduction to the bootstrap*. CRC press.

[59] Storrs, K. R., Kietzmann, T. C., Walther, A., Mehrer, J., and Kriegeskorte, N. (2021). Diverse Deep Neural Networks All Predict Human Inferior Temporal Cortex Well, After Training and Fitting. *Journal of Cognitive Neuroscience*, pages 1–21.

[60] Horikawa, T. and Kamitani, Y. (2017). Generic decoding of seen and imagined objects using hierarchical visual features. *Nature Communications*, 8(1):15037.

[61] Glasser, M. F., Coalson, T. S., Robinson, E. C., Hacker, C. D., Harwell, J., Yacoub, E., Ugurbil, K., Andersson, J., Beckmann, C. F., Jenkinson, M., Smith, S. M., and Van Essen, D. C. (2016). A multi-modal parcellation of human cerebral cortex. *Nature*, 536(7615):171–178.

[62] de Vries, S. E. J., Lecoq, J. A., Buice, M. A., Groblewski, P. A., Ocker, G. K., Oliver, M., Feng, D., Cain, N., Ledochowitsch, P., Millman, D., Roll, K., Garrett, M., Keenan, T., Kuan, L., Mihalas, S., Olsen, S., Thompson, C., Wakeman, W., Waters, J., Williams, D., Barber, C., Berbesque, N., Blanchard, B., Bowles, N., Caldejon, S. D., Casal, L., Cho, A., Cross, S., Dang, C., Dolbeare, T., Edwards, M., Galbraith, J., Gaudreault, N., Gilbert, T. L., Griffin, F., Hargrave, P., Howard, R., Huang, L., Jewell, S., Keller, N., Knoblich, U., Larkin, J. D., Larsen, R., Lau, C., Lee, E., Lee, F., Leon, A., Li, L., Long, F., Luviano, J., Mace, K., Nguyen, T., Perkins, J., Robertson, M., Seid, S., Shea-Brown, E., Shi, J., Sjoquist, N., Slaughterbeck, C., Sullivan, D., Valenza, R., White, C., Williford, A., Witten, D. M., Zhuang, J., Zeng, H., Farrell, C., Ng, L., Bernard, A., Phillips, J. W., Reid, R. C., and Koch, C. (2020). A large-scale standardized physiological survey reveals functional organization of the mouse visual cortex. *Nature Neuroscience*, 23(1):138–151.

[63] Ejaz, N., Hamada, M., and Diedrichsen, J. (2015). Hand use predicts the structure of representations in sensorimotor cortex. *Nature Neuroscience*, 18(7):1034–1040.

[64] Wu, M. C.-K., David, S. V., and Gallant, J. L. (2006). Complete functional characterization of sensory neurons by system identification. *Annu. Rev. Neurosci.*, 29:477–505.

[65] Ledoit, O. and Wolf, M. (2004). Honey, I shrunk the sample covariance matrix. *The Journal of Portfolio Management*, 30(4):110–119.

[66] Schäfer, J. and Strimmer, K. (2005). A Shrinkage Approach to Large-Scale Covariance Matrix Estimation and Implications for Functional Genomics. *Statistical Applications in Genetics and Molecular Biology*, 4(1).

[67] Friston, K. J., Diedrichsen, J., Holmes, E., and Zeidman, P. (2019). Variational representational similarity analysis. *NeuroImage*, 201:115986.

[68] Kay, K. N., Naselaris, T., Prenger, R. J., and Gallant, J. L. (2008). Identifying natural images from human brain activity. *Nature*, 452(7185):352–355.

[69] Dumoulin, S. O. and Wandell, B. A. (2008). Population receptive field estimates in human visual cortex. *Neuroimage*, 39(2):647–660.

[70] Naselaris, T., Kay, K. N., Nishimoto, S., and Gallant, J. L. (2011). Encoding and decoding in fMRI. *NeuroImage*, 56(2):400–410.

[71] Wandell, B. A. and Winawer, J. (2015). Computational neuroimaging and population receptive fields. *Trends in cognitive sciences*, 19(6):349–357.

[72] Cadena, S. A., Denfield, G. H., Walker, E. Y., Gatys, L. A., Tolias, A. S., Bethge, M., and Ecker, A. S. (2019). Deep convolutional models improve predictions of macaque V1 responses to natural images. *PLOS Computational Biology*, 15(4):e1006897.

[73] Cadena, S. A., Sinz, F. H., Muhammad, T., Froudarakis, E., Cobos, E., Walker, E. Y., Reimer, J., Bethge, M., Tolias, A. S., and Ecker, A. S. (2019). How well do deep neural networks trained on object recognition characterize the mouse visual system? In *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*, page 5.

[74] Kornblith, S., Norouzi, M., Lee, H., and Hinton, G. (2019). Similarity of neural network representations revisited. *Proceedings of the 36th International Conference on Machine Learning*, page 11.

[75] Kriegeskorte, N. and Douglas, P. K. (2019). Interpreting encoding and decoding models. *Current Opinion in Neurobiology*, 55:167–179.

[76] Storrs, K. R., Khaligh-Razavi, S.-M., and Kriegeskorte, N. (2020). Noise ceiling on the crossvalidated performance of reweighted models of representational dissimilarity: Addendum to Khaligh-Razavi & Kriegeskorte (2014). Preprint, Neuroscience.

[77] Kriegeskorte, N. and Mur, M. (2012). Inverse mds: Inferring dissimilarity structure from multiple item arrangements. *Frontiers in psychology*, 3:245.

[78] Kemeny, J. G. (1959). Mathematics without Numbers. *Daedalus*, 88(4):577–591.

[79] Young, H. P. and Levenglick, A. (1978). A Consistent Extension of Condorcet's Election Principle. *SIAM Journal on Applied Mathematics*, 35(2):285–300. Publisher: Society for Industrial and Applied Mathematics.

[80] Ali, A. and Meilă, M. (2012). Experiments with Kemeny ranking: What works when? *Mathematical Social Sciences*, 64(1):28–40.

[81] Kendall, M. G. (1948). *Rank correlation methods.* Griffin.

[82] Mehrer, J., Spoerer, C. J., Jones, E. C., Kriegeskorte, N., and Kietzmann, T. C. (2021). An ecologically motivated image dataset for deep learning yields better models of human vision. *Proceedings of the National Academy of Sciences*, 118(8):e2011417118.

[83] Pedregosa, F., Eickenberg, M., Ciuciu, P., Thirion, B., and Gramfort, A. (2015). Data-driven HRF estimation for encoding and decoding models. *NeuroImage*, 104:209–220.

[84] Bodurka, J., Ye, F., Petridou, N., Murphy, K., and Bandettini, P. (2007). Mapping the MRI voxel volume in which thermal noise matches physiological noise—Implications for fMRI. *NeuroImage*, 34(2):542–549.

[85] Chaimow, D., Yacoub, E., Uğurbil, K., and Shmuel, A. (2018). Spatial specificity of the functional MRI blood oxygenation response relative to neuronal activity. *NeuroImage*, 164:32–47.

[86] Weldon, K. B. and Olman, C. A. (2021). Forging a path to mesoscopic imaging success with ultra-high field functional magnetic resonance imaging. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 376(1815):20200040.

[87] Esteban, O., Markiewicz, C., Blair, R. W., Moodie, C., Isik, A. I., Erramuzpe Aliaga, A., Kent, J., Goncalves, M., DuPre, E., Snyder, M., Oya, H., Ghosh, S., Wright, J., Durnez, J., Poldrack, R., and Gorgolewski, K. J. (2018). fMRIPrep: a robust preprocessing pipeline for functional MRI. *Nature Methods*.

[88] Esteban, O., Blair, R., Markiewicz, C. J., Berleant, S. L., Moodie, C., Ma, F., Isik, A. I., Erramuzpe, A., Kent, M., James D. andGoncalves, DuPre, E., Sitek, K. R., Gomez, D. E. P., Lurie, D. J., Ye, Z., Poldrack, R. A., and Gorgolewski, K. J. (2018). fmriprep. *Software*.

[89] Gorgolewski, K., Burns, C. D., Madison, C., Clark, D., Halchenko, Y. O., Waskom, M. L., and Ghosh, S. (2011). Nipype: a flexible, lightweight and extensible neuroimaging data processing framework in python. *Frontiers in Neuroinformatics*, 5:13.

[90] Gorgolewski, K. J., Esteban, O., Markiewicz, C. J., Ziegler, E., Ellis, D. G., Notter, M. P., Jarecka, D., Johnson, H., Burns, C., Manhães-Savio, A., Hamalainen, C., Yvernault, B., Salo, T., Jordan, K., Goncalves, M., Waskom, M., Clark, D., Wong, J., Loney, F., Modat, M., Dewey, B. E., Madison, C., Visconti di Oleggio Castello, M., Clark, M. G., Dayan, M., Clark, D., Keshavan, A., Pinsard, B., Gramfort, A., Berleant, S., Nielson, D. M., Bougacha, S., Varoquaux, G., Cipollini, B., Markello, R., Rokem, A., Moloney, B., Halchenko, Y. O., Wassermann, D., Hanke, M., Horea, C., Kaczmarzyk, J., de Hollander, G., DuPre, E., Gillman, A., Mordom, D., Buchanan, C., Tungaraza, R., Pauli, W. M., Iqbal, S., Sikka, S., Mancini, M., Schwartz, Y., Malone, I. B., Dubois, M., Frohlich, C., Welch, D., Forbes, J., Kent, J., Watanabe, A., Cumba, C., Huntenburg, J. M., Kastman, E., Nichols, B. N., Eshaghi, A., Ginsburg, D., Schaefer, A., Acland, B., Giavasis, S., Kleesiek, J., Erickson, D., Küttner, R., Haselgrove, C., Correa, C., Ghayoor, A., Liem, F., Millman, J., Haehn, D., Lai, J., Zhou, D., Blair, R., Glatard, T., Renfro, M., Liu, S., Kahn, A. E., Pérez-García, F., Triplett, W., Lampe, L., Stadler, J., Kong, X.-Z., Hallquist, M., Chetverikov, A., Salvatore, J., Park, A., Poldrack, R., Craddock, R. C., Inati, S., Hinds, O., Cooper, G., Perkins, L. N., Marina, A., Mattfeld, A., Noel, M., Snoek, L., Matsubara, K., Cheung, B., Rothmei, S., Urchs, S., Durnez, J., Mertz, F., Geisler, D., Floren, A., Gerhard, S., Sharp, P., Molina-Romero, M., Weinstein, A., Broderick, W., Saase, V., Andberg, S. K., Harms, R., Schlamp, K., Arias, J., Papadopoulos Orfanos, D., Tarbert, C., Tambini, A., De La Vega, A., Nickson, T., Brett, M., Falkiewicz, M., Podranski, K., Linkersdörfer, J., Flandin, G., Ort, E., Shachnev, D., McNamee, D., Davison, A., Varada, J., Schwabacher, I., Pellman, J., Perez-Guevara, M., Khanuja, R., Pannetier, N., McDermottroe, C., and Ghosh, S. (2018). Nipype. *Software*.

[91] Tustison, N. J., Avants, B. B., Cook, P. A., Zheng, Y., Egan, A., Yushkevich, P. A., and Gee, J. C. (2010). N4itk: Improved n3 bias correction. *IEEE Transactions on Medical Imaging*, 29(6):1310–1320.

[92] Avants, B., Epstein, C., Grossman, M., and Gee, J. (2008). Symmetric diffeomorphic image registration with cross-correlation: Evaluating automated labeling of elderly and neurodegenerative brain. *Medical Image Analysis*, 12(1):26–41.

[93] Zhang, Y., Brady, M., and Smith, S. (2001). Segmentation of brain MR images through a hidden markov random field model and the expectation-maximization algorithm. *IEEE Transactions on Medical Imaging*, 20(1):45–57.

[94] Dale, A. M., Fischl, B., and Sereno, M. I. (1999). Cortical surface-based analysis: I. segmentation and surface reconstruction. *NeuroImage*, 9(2):179–194.

[95] Klein, A., Ghosh, S. S., Bao, F. S., Giard, J., Häme, Y., Stavsky, E., Lee, N., Rossa, B., Reuter, M., Neto, E. C., and Keshavan, A. (2017). Mindboggling morphometry of human brains. *PLOS Computational Biology*, 13(2):e1005350.

[96] Fonov, V., Evans, A., McKinstry, R., Almli, C., and Collins, D. (2009). Unbiased nonlinear average age-appropriate brain templates from birth to adulthood. *NeuroImage*, 47, Supplement 1:S102.

[97] Greve, D. N. and Fischl, B. (2009). Accurate and robust brain image alignment using boundary-based registration. *NeuroImage*, 48(1):63–72.

[98] Jenkinson, M., Bannister, P., Brady, M., and Smith, S. (2002). Improved optimization for the robust and accurate linear registration and motion correction of brain images. *NeuroImage*, 17(2):825–841.

[99] Cox, R. W. and Hyde, J. S. (1997). Software tools for analysis and visualization of fmri data. *NMR in Biomedicine*, 10(4-5):171–178.

[100] Power, J. D., Mitra, A., Laumann, T. O., Snyder, A. Z., Schlaggar, B. L., and Petersen, S. E. (2014). Methods to detect, characterize, and remove motion artifact in resting state fmri. *NeuroImage*, 84(Supplement C):320–341.

[101] Behzadi, Y., Restom, K., Liau, J., and Liu, T. T. (2007). A component based noise correction method (CompCor) for BOLD and perfusion based fmri. *NeuroImage*, 37(1):90–101.

[102] Satterthwaite, T. D., Elliott, M. A., Gerraty, R. T., Ruparel, K., Loughead, J., Calkins, M. E., Eickhoff, S. B., Hakonarson, H., Gur, R. C., Gur, R. E., and Wolf, D. H. (2013). An improved framework for confound regression and filtering for control of motion artifact in the preprocessing of resting-state functional connectivity data. *NeuroImage*, 64(1):240–256.

[103] Lanczos, C. (1964). Evaluation of noisy data. *Journal of the Society for Industrial and Applied Mathematics Series B Numerical Analysis*, 1(1):76–85.

[104] Abraham, A., Pedregosa, F., Eickenberg, M., Gervais, P., Mueller, A., Kossaifi, J., Gramfort, A., Thirion, B., and Varoquaux, G. (2014). Machine learning for neuroimaging with scikit-learn. *Frontiers in Neuroinformatics*, 8.

5 Online Methods

The methods section for this paper is separated into two parts: First, we describe the RSA analysis pipeline we propose in more detail. In the second part, we describe the simulation methods we used to test our pipeline for this paper.

5.1 Details of RSA

The inference method we describe here represents a new pipeline for representational similarity analysis. Nonetheless, some parts of the analysis appeared in earlier or concurrent publications [44, 45, 46, 76].

5.1.1 Computing representational dissimilarity matricies

How representational dissimilarity is best quantified and inferred from raw data depends on the type of measurements taken. For functional Magnetic Resonance Imaging (fMRI) for example, it is beneficial to take the noise covariance across voxels into account by computing Mahalanobis distances [46]. For electro-physiological recordings of individual neurons one should take the poisson nature of the variability into account by either transforming the measured spiking rates or by using a KL-divergence based dissimilarity measure based on the poisson-distribution [42]. And inferring dissimilarities from behavior based on categorization or arrangement data uses entirely different techniques [77].

In the formal mathematical sense a distance or metric is a function that takes two points from the space as input and computes a real number from them subject to the following three rules: (1) The result is larger or equal to zero, with equality only if the two points are equal. (2) Symmetry, i.e. the result is the same if the two points are swapped. (3) The triangle inequality, i.e. the sum of distances from a to b and b to c is less or equal to the distance of a to c for all choices of the three points. For a dissimilarity, we drop requirement (3), which allows symmetric divergences between probability distributions for example. Additionally, we allow estimators for dissimilarities that may return negative values in violation of (1). To do so, we weaken (1) to be necessary only in expectation.

In principle any dissimilarity measure on the measured representation vectors can be used to quantify the dissimilarities between conditions. Popular choices in the past were squared and un-squared euclidean distances, cosine distances, linear decoding accuracy and correlations distances. Earlier publications comparing different measures of dissimilarity found simple correlation distances to be less informative for comparing models than other methods and the underlying decoding axes to be better than decoding accuracy [46]. Here, we focus on crossnobis RDMs, which were found to work best for fMRI-like data [46], have clear interpretations as linear decodeability of differences and whose sampling distributions can be described analytically [52]. Two aspects of these dissimilarities warrant further discussion: crossvalidation of dissimilarities and taking noise covariance into account.

Crossvalidated dissimilarities One important aspect of the first level analysis is that naive estimates of representational similarity can be severely biased [46, 51, 52] towards a structure dictated by the structure of the experiment rather than the structure of the representations. To avoid this problem one can use crossvalidated distances, which combine difference estimates from independent measurements such that the dissimilarity estimate is unbiased. Crossvalidation loses less statistical power the more repetitions can be used [52]. When crossvalidation is used with Mahalanobis distances this is referred to as the cross-nobis distance [46]. Alternatively, one can employ Bayesian inference [51], which can provide estimates which are not biased by the experimental structure either.

For our simulations in this paper we used crossvalidated Mahalanobis (*Crossnobis*) dissimilarities throughout. For $N \geq 2$ repeated measurements of response patterns $\mathbf{x}_{mi}, m = 1 \dots N, i = 1 \dots K$ (for K conditions), the crossnobis estimator $\hat{d}_{ij}$ is defined as:

$$\hat{d}_{ij} = \frac{1}{N(N-1)} \sum_{m} \sum_{n \neq m} (\mathbf{x}_{mi} - \mathbf{x}_{mj})^T \Sigma^{-1} (\mathbf{x}_{ni} - \mathbf{x}_{nj}) \tag{3}$$

for an estimate noise covariance matrix Σ.

where Σ is an estimate of the noise (co-)variance matrix. The crossnobis estimator provides unbiased estimates of the underlying dissimilarities. As in the non-crossvalidated Mahalanobis distance, the linear transformation of of the response dimensions (using the noise precision matrix Σ^{-1}) improves reliability [46, 45] and renders the estimates monotonically related to the linear decoding accuracy for each pair of conditions, when a fixed Gaussian error distribution is assumed.

Noise covariance estimation To take the noise covariance into account we first need to estimate it. To do so, we can use one of two sources of information: We either estimate the covariance based on the variation of the repeated

measurements around their mean or based on the residuals of a first level analysis which estimated the patterns from the raw data. For fMRI for example, these residuals would be the residuals of the first level GLM. In either case we may have relatively little data to estimate a large noise covariance matrix. Making this feasible usually requires regularization. To do so the following methods are available:

- When the estimation task is judged to be entirely impossible one can reduce the Mahalanobis and Crossnobis back to the euclidean and crossvalidated euclidean distances by using the identity matrix instead.
- As a univariate simplification one can estimate a diagonal matrix which only takes the variances of voxels into account.
- For estimating a full covariance one may use a shrinkage estimate, which "shrinks" the sample covariance towards a simpler estimate of the covariance like a multiple of the identity or the diagonal of variances [65, 66]. The amount of shrinkage used in these methods fortunately can be estimated quite accurately based on the data directly such that these methods do not require parameter adjustment.

We implemented these different methods and test them against each other in the main text. Overall the shrinkage estimates perform best, while the other techniques are equally good in some situations. Using the sample covariance directly is not advisable unless an unusually large amount of data exists for this estimation, in which case the shrinkage estimates converge towards the sample covariance anyway.

Poisson-like variability Instead of the Gaussian variability implied by the Euclidean and Mahalanobis dissimilarity measures discussed above, noise is often assumed to be poisson or at least to have its variance increase linearly with mean activation. This is used primarily when the spiking variability of neurons is thought to be the main noise source as in electrophysiological recordings. For this case we discuss two possible solutions.

The first alternative, discussed by [42] is to use a variance stabilizing transform, i.e. to apply a square root to all dimensions of all representations and use an RDM based on the transformed values. This has the advantage, that the covariances can be taken into account.

The second alternative, which we first introduce here is to use a symmetrized KL-divergence of Poisson distributions with mean firing rates given by the feature values. This approach automatically takes the increased variance at larger activation levels into account and inherits nice information theoretic and decoding based interpretations from the KL divergence.

The KL-divergence of two Poisson distributions with mean rates λ_1 and λ_2 is given by:

$$KL(\lambda_1||\lambda_2) = \sum_{k=0}^{\infty} P(k|\lambda_1) \log \frac{P(k|\lambda_1)}{P(k|\lambda_2)} \tag{4}$$

$$= \lambda_1 \log \frac{\lambda_1}{\lambda_2} + \lambda_2 - \lambda_1 \tag{5}$$

Based on this we can compute the symmetrized version of the KL:

$$KL_{sym}(\lambda_1, \lambda_2) = KL(\lambda_1||\lambda_2) + KL(\lambda_2||\lambda_1) \tag{6}$$

$$= (\lambda_1 - \lambda_2)(\log \lambda_1 - \log \lambda_2) \tag{7}$$

To get a crossvalidated version of this dissimilarity we can calculate the difference in logarithms from one crossvalidation fold and the difference between raw values for a different fold and average across all pairs of different crossvalidation folds.

This KL-divergence based dissimilarity is theoretically more interpretable than the square root transform, but comes with two small drawbacks: First, the underlying firing rates cannot be 0 as a poisson distribution which never fires is infinitely different from all others. This can be easily fixed by using a weak prior on the firing rate, which results in a non-zero estimated firing rate. Second, there is no straight forward way to include a noise-covariance into the dissimilarity. While such noise correlations are much weaker than correlations between nearby voxels in fMRI or nearby electrodes in MEG, correlated noise may still limit discriminability based on large neural populations. Thus, there might be situations, where this is a good reason to prefer the square root transform.

5.1.2 Comparing RDMs

The second level analysis is comparing a set of measured data-RDMs (from a set of subjects) to the RDMs predicted by different models. There are various measures to compare RDMs, which are chosen based on what aspects the model

RDMs are meant to capture. The strictest measure to use would be to simply compute a euclidean distance between a model-RDM and the reference-RDMs. In virtually all cases, we cannot predict the exact magnitude of the distances though, as the signal to noise ratio varies between subjects and measurement sessions. Allowing an overall scaling of the RDMs leads to the cosine similarity. If we additionally drop the assumption that a predicted difference of 0 corresponds to a measured dissimilarity of 0, we can use a correlation between RDMs. For the cosine similarity and correlation between RDMs we recently proposed whitened variants which take the correlations between the different entries of the RDM into account [52]. Finally, we can drop the assumption of a linear relationship between RDMs by using rank correlations like Kendall's τ or Spearman's ρ. For this lowest bar for a relationship Kendall's τ_a or randomized rank breaking for Spearman's ρ_a are usually preferred over a standard Spearman's ρ or Kendall's τ_b and τ_c, which all favor RDMs with tied ranks [45]. As we discuss below there is a direct formula for the expected Spearman's rho under random tiebraking, which we prefer now for computational efficiency reasons. Concretely, these different comparison metrics are defined as follows:

Cosine similarity The most stringent similarity measure for RDMs is the cosine similarity. For two vectorized RDMs $\mathbf{r}_1$ and $\mathbf{r}_2$ it is defined as:

$$\frac{\mathbf{r}_1^T \mathbf{r}_2}{\sqrt{\mathbf{r}_1^T \mathbf{r}_1 \, \mathbf{r}_2^T \mathbf{r}_2}} \tag{8}$$

Correlation When a dissimilarity of 0 is not interpretable as indistinguishable, the average dissimilarity can be removed by using the Pearson correlation as a similarity measure. It is defined as:

$$\frac{(\mathbf{r}_1 - \bar{\mathbf{r}}_1)^T (\mathbf{r}_2 - \bar{\mathbf{r}}_2)}{\sqrt{(\mathbf{r}_1 - \bar{\mathbf{r}}_1)^T (\mathbf{r}_1 - \bar{\mathbf{r}}_1) \, (\mathbf{r}_2 - \bar{\mathbf{r}}_2)^T (\mathbf{r}_2 - \bar{\mathbf{r}}_2)}}, \tag{9}$$

where the bar indicates the mean of the vector.

Whitened similarity measures We recently derived a formula for the covariance of RDM entries, which arises because all dissimilarities of a single condition are based on the same measurements of that condition [55, 52]. Based on a simplified estimate of this covariance V we can then compute a whitened cosine similarity and a whitened correlation defined as follows:

$$\frac{\mathbf{r}_1^T V^{-1} \mathbf{r}_2}{\sqrt{\mathbf{r}_1^T V^{-1} \mathbf{r}_1 \, \mathbf{r}_2^T V^{-1} \mathbf{r}_2}} \tag{10}$$

$$\frac{(\mathbf{r}_1 - \bar{\mathbf{r}}_1)^T V^{-1} (\mathbf{r}_2 - \bar{\mathbf{r}}_2)}{\sqrt{(\mathbf{r}_1 - \bar{\mathbf{r}}_1)^T V^{-1} (\mathbf{r}_1 - \bar{\mathbf{r}}_1) (\mathbf{r}_2 - \bar{\mathbf{r}}_2)^T V^{-1} (\mathbf{r}_2 - \bar{\mathbf{r}}_2)}} \tag{11}$$

As we derived in our earlier publication these measures are exactly equivalent to a linear centered kernel alignment (CKA) [52] with or without removing the mean activation vector. This equivalent formulation can be computed faster as it avoids the inversion of V. Thus, our implementation uses this equivalent formulation for faster computation in the background.

Spearman's ρ_a In previous versions of RSA it was argued that Kendall's τ_a is preferable over other rank-correlation methods, because it does not prefer predictions with tied ranks over the random orderings of the same entries [45]. However, Kendall's τ-type correlation coefficients are considerably slower to compute than Spearman's ρ-type correlation coefficients. Moreover, finding the RDM with the highest average τ_a for a given set of data RDMs (upper bound of the noise ceiling) is equivalent to the Kemeny-Young method for preference voting [78, 79], which is NP-hard to compute and practically too slow to compute for our application [80].

These problems can be alleviated by using the expected Spearman's ρ under random tie breaking as an evaluation criterion instead. The coefficient ρ_a was described by Kendall [81, chapter 3.8]. For a vector $\mathbf{x} \in \mathbb{R}^n$, let $Rae(\mathbf{x})$ be the distribution of random-among-equals rank-transforms, where each unique value in $\mathbf{x}$ is replaced with its integer rank and, in the case of a set of tied values, a random permutation of the corresponding ranks. For each draw $\tilde{\mathbf{x}} \sim Rae(\mathbf{x})$, thus, $x_i < x_j \Rightarrow \tilde{x}_i < \tilde{x}_j$. However, for pairs (i, j), where $x_i = x_j$, the ranks will fall in order $\tilde{x}_i < \tilde{x}_j$ or $\tilde{x}_i > \tilde{x}_j$ with equal probability. The set of values $\{\tilde{x}_i | 1 \leq i \leq n\}$ is $\{1, \ldots, n\}$. The ρ_a correlation coefficient is defined as:

$$\rho_a(\mathbf{x}, \mathbf{y}) = \mathbb{E}_{\substack{\tilde{\mathbf{x}} \sim Rae(\mathbf{x}) \\ \tilde{\mathbf{y}} \sim Rae(\mathbf{y})}} \left[\rho(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}) \right] \tag{12}$$

For this expectation there is a direct formula:

$$\rho_a(\mathbf{x}, \mathbf{y}) = \mathbb{E}_{\substack{\tilde{\mathbf{a}}=\tilde{\mathbf{x}}-\frac{1}{n}\sum_{i=1}^{n} i,\tilde{\mathbf{x}}\sim Rae(\mathbf{x}) \\ \tilde{\mathbf{b}}=\tilde{\mathbf{y}}-\frac{1}{n}\sum_{i=1}^{n} i,\tilde{\mathbf{y}}\sim Rae(\mathbf{y})}} \left[\frac{\tilde{\mathbf{a}}^\top \tilde{\mathbf{b}}}{\|\tilde{\mathbf{a}}\|_2 \|\tilde{\mathbf{b}}\|_2} \right] \tag{13}$$

$$= \frac{12}{n^3 - n} \mathbb{E}_{\tilde{\mathbf{a}}}[\tilde{\mathbf{a}}]^\top \mathbb{E}_{\tilde{\mathbf{b}}}[\tilde{\mathbf{b}}] \tag{14}$$

$$= \frac{12\mathbf{x}^\top \mathbf{y}}{n^3 - n} - \frac{6(n-1)^2}{n(n+1)} \tag{15}$$

Thus, computing this value ρ_a does not require drawing actual random tie breaks.

5.1.3 Noise ceilings

There is a general interest to compute a noise ceiling for the performance of models, i.e. a measure for how good a model could be on the given data. For representational dissimilarity analysis, it is common to infer both an upper and a lower bound on model performance [45].

The upper bound is constructed by computing the RDM which performs best among all RDMs. Obviously, no model can perform better than this best RDM. For most RDM comparison measures, this only requires taking a mean after adequate normalisation of the single subject RDMs. For cosine similarity, they are normalized to unit norm. For Pearson correlation, the RDM vectors are normalized to zero mean and unit standard deviation. For the whitened measures the normalization is based on the norm induced by the noise precision instead, i.e. subject RDM vectors $\mathbf{r}$ are divided by $\sqrt{\mathbf{r}^T \Sigma^{-1} \mathbf{r}}$ instead of the standard euclidean norm $\sqrt{\mathbf{r}^T \mathbf{r}}$. For the Spearman correlation, subject RDM vectors are first transformed to ranks.

For Kendall's τ_a, there is no simple efficient method to find the optimal RDM for a dataset, which is one of the reasons for using the Spearman rank correlation for RDM comparisons. If Kendall τ based inference is chosen nonetheless, the problem can be solved approximately by applying techniques for Kemeny-Young voting [80] or by simply using the average ranks, which is a reasonable approximation, especially if the rank transformed RDMs are similar across subjects. In our current toolbox we use this approximation without further adjustment.

For the lower bound, leave-one-out crossvalidation over subjects can be used. To do this, each subject is once selected as the left out subject and the best RDM to fit all other subjects is computed. The expected average performance of this RDM is a lower bound on the true models performance, because fitting all distance independently is technically a very flexible model, which performs the same generalization as the tested models. As all other models it should thus perform worse than or equal to the correct model.

When flexible models are used, such that crossvalidation over conditions is performed, the computation of noise ceilings needs to take this into account [76]. Essentially, the computation of the noise ceilings is then restricted to the test sets of the crossvalidation, which takes into account that only parts of the RDMs are ever used for evaluation.

5.1.4 Variance estimation

Additional to the point estimate of model performances we want to estimate how certain we should be about our evaluations. In the frequentist framework this corresponds to an estimate how variable our evaluation results would be if we repeated the experiment.

Subject variance The easiest to estimate variance is the variance our results would have if we repeated the experiment with new subjects, but the same conditions, as all our evaluations are simple averages across subjects. Thus, an estimate of the variance can always be obtained by dividing the sample variance over subjects by the number of subjects.

Bootstrapping conditions The dependence of the evaluation on the conditions is more complicated. Thus, we use bootstrapping [58] to estimate the variance we expect over repetitions of the experiment with new conditions but the same subjects. To do so, we sample sets of conditions with replacement from the set of measured conditions and generate the data-RDM and the model RDMs for this sample of conditions. The variance of the model performances on these resampled RDMs is then an estimate of the variance over experiments with different stimulus choices.

The bootstrap samples of conditions contain repetitions of the same condition. The dissimilarity between a stimulus and itself will be 0 in the data and any model such that every model would correctly predict these self-dissimilarities. Thus,

including these self-dissimilarities would bias all model performances upward. To avoid this, we exclude them from the evaluation.

Two-factor bootstrap We are often interested in estimating the variance of model performance estimates that would result across experiments if we sampled both new subjects and new conditions for each experiment. This variance estimate supports inference that generalizes to the population of conditions and subjects. To estimate this variance, we resort to a novel bootstrapping method.

Simulations and mathematical analysis show consistently that the variance estimate based on simply resampling both subjects and conditions simultaneously is too large, i.e. the results differ more between bootstrap samples from the same dataset than they differ between repetitions of the simulated experiment. This is surprising, because simultaneous botstrapping of both factors imitates the sampling process of the experiment. The simultaneous bootstrap technique has in fact been used in earlier publications [e.g. 59]. However, we now understand this method to be severely conservative and not optimally sensitive.

The reason for this is that bootstrapping over the two dimensions overaccounts for variations, which are independent of both condition and subject. The bootstrap variance can be separated into a contribution of the conditions, a contribution of the subjects, and a contribution of the interaction of subjects and conditions or measurement noise. The interaction cannot be distinguished from the measurement noise. However, the independent noise contribution enters not only its own term, but also the two others. The total variance estimated by the bootstrap corresponds to the sum of (1) the variance due to measurement noise and interactions between subjects and conditions, (2) the variance across subjects in the sample, which contains both the variation due to the sample of subjects and the variation due to measurement noise and interactions, (2) the variance across conditions, which contains both the variation due to the sample of conditions and (again) the variation due to the measurement noise and interactions. Thus, both the variance over subjects and the variance over conditions contain another contribution of equal size to the independent variance component additional to the variance that is attached to them in the true underlying model.

This is not specific to RSA. Even in a simple additive model, the subject and stimulus independent noise component enters the bootstrap variance three times. Whenever there is substantial measurement noise this leads to a severe overestimation of the uncertainty.

We can compute the variances caused by either condition choice and measurement noise or subject sampling and measurement noise using bootstrapping over only conditions or only subjects. Combining the variance estimates obtained from these two σ^2_{stim} and σ^2_{subj} with the variance estimate obtained from the uncorrected two-factor bootstrap, we can compute an estimate $\hat{\sigma}^2$, which has the right expectation for the additive model:

$$\hat{\sigma}^2_{subj} \approx \sigma^2_{subj} + \sigma^2_{noise} \tag{16}$$

$$\hat{\sigma}^2_{stim} \approx \sigma^2_{stim} + \sigma^2_{noise} \tag{17}$$

$$\hat{\sigma}^2_{both} \approx \sigma^2_{subj} + \sigma^2_{stim} + 3\sigma^2_{noise} \tag{18}$$

$$\Rightarrow \hat{\sigma}^2 = 2(\hat{\sigma}^2_{subj} + \hat{\sigma}^2_{stim}) - \hat{\sigma}^2_{both} \tag{19}$$

$$\approx \sigma^2_{subj} + \sigma^2_{stim} + \sigma^2_{noise} \tag{20}$$

We show in multiple simulations that this estimate approximates the correct variance better than the uncorrected two-factor bootstrap, although it should be noted that we derived the combination formula for the bootstraps based on an additive model, which is not directly applicable here.

To avoid impossible estimates and reduce the variance of the estimator somewhat we bound this estimator from above and below based on bounds we can derive based on these bootstrap variances. We use both σ^2_{subj} and σ^2_{stim} as lower bounds for the estimate as the variances they estimate are smaller than the true variance. As an upper bound, we use σ^2_{double}, our original, too large estimate. This biases the estimate again, but yields variance estimates that are strictly positive and less variable.

Bootstrap-wrapped crossvalidation To avoid overfitting, when using flexible models, we suggest to use crossvalidation. Crossvalidation means that we separate the dataset into separate test-groups. We then fit the models to all but one group and evaluate on the left out group. Taking the average over the training groups yields a single performance estimate. For the same reason as for bootstrapping, crossvalidation should be done over both conditions and subjects.

As the RDM for the test-set must contain multiple values to allow any sensible comparison the smallest possible number of conditions to perform crossvalidation is 6. By default we start with 2 folds and start using 3 folds at 12 conditions, 4

folds at 24 conditions and 5 folds at 40 conditions and stick to 5 folds from there on. These numbers seem to work reasonably well, but were chosen ad hoc.

To estimate the uncertainty about our crossvalidated model performances we use a bootstrap, which creates samples from our dataset. We then perform crossvalidation on each of these samples. We call this bootstrap-wrapped crossvalidation.

Different assignments of the conditions and subjects to the different folds lead to different evaluations of the models. When we assign conditions to groups in RSA, this effect is particularly strong, because dissimilarities between conditions in separate groups do not enter the evaluation. The variance in the evaluations created by this random assignment is generated only by our analysis though and should not increase our uncertainty about the model performance. Indeed, we can define the 'true' model performance on a dataset as the average over all possible crossvalidation fold assignments.

Unfortunately, computing model performance under all possible crossvalidation assignments will usually be prohibitively expensive in terms of computation time, especially, when we want to compute this inside a bootstrap sampling procedure. We can instead get an approximation by sampling $n_{cv} > 1$ different randomly chosen crossvalidation-fold assignments for each bootstrap sample.

The expected model performance across randomly chosen fold-assignments is exactly the mean model performance over all possible assignments such that the mean is a consistent estimate of model performance. However, the bootstrap wrapped crossvalidation estimate of the variance of the model performances with n_{cv} randomly chosen crossvalidation samples will be larger than the variance of the exact means σ_{boot}. When we assume that the variance of randomly chosen fold assignments around their mean σ^2_{cv} is equal for each bootstrap sample the overall variance $\sigma^2_{bootcv,n_{cv}}$ is:

$$\sigma^2_{bootcv,n_{cv}} = \sigma^2_{boot} + \frac{\sigma^2_{cv}}{n_{cv}} \tag{21}$$

When we have multiple fold-assignments for each bootstrap sample it is straightforward to compute an estimate for the variance we would have gotten if we had drawn only a single fold assignment $\sigma^2_{bootcv,1}$, as the variance of all individual fold assignments from all bootstrap samples.

Using these two variance estimates we can simply solve for the variance contributions of the fold assignment and of the bootstrap:

$$\sigma^2_{cv} = \frac{n_{cv}}{n_{cv}-1}\left(\sigma^2_{bootcv,1} - \sigma^2_{bootcv,n_{cv}}\right) \tag{22}$$

$$\sigma^2_{boot} = \sigma^2_{bootcv,n_{cv}} - \frac{1}{n_{cv}-1}\left(\sigma^2_{bootcv,1} - \sigma^2_{bootcv,n_{cv}}\right) \tag{23}$$

Thus, we can directly compute an estimate of the variance with all fold assignments from a simulation containing 2 or more crossvalidation assignments for each bootstrap sample. We show in the main text that the average estimate is independent of n_{cv} (Fig. 4 a). Additionally, the variance in the estimate reduces more, when we increase the number of bootstrap samples than it decreases when we increase the number of fold assignments per bootstrap sample. Thus, we currently recommend using only 2 fold assignments per bootstrap sample.

In principle, one can go even further and use only few randomly chosen test-sets per bootstrap sample, i.e. stop running full crossvalidations, which put each stimulus and each subject into the test-set once. In preliminary simulations (not shown), this procedure added substantial additional variance though, which was not compensated by running correspondingly more bootstrap samples. Thus, using full crossvalidations seems to be more efficient.

5.1.5 Tests

Given an uncertainty estimate for the model performances there is a collection of tests available to compare model performances against each other, to the noise ceiling, or to chance performance.

As we base our uncertainty estimates on a bootstrap there are two types of tests we can use for these comparisons: A percentile test based on the bootstrap samples or a T-test based on the estimated variances.

For the percentile test, we calculate the bootstrap distribution for the differences and then test by checking whether the difference expected under the H_0 (usually 0) lies within the simple percentile bootstrap confidence interval. It is possible to generate more exact confidence intervals like bias corrected and accelerated intervals based on the bootstrap samples, which result in potentially better tests. In our simulations and experience with natural data, model performances tend

to be fairly symmetrically distributed around the original model evaluation though, such that additional corrections seemed unnecessary.

For the T-test we use the variance estimated from the bootstrap and use the number of observations minus one as the degrees of freedom. When bootstrapping across both subjects and conditions, we used the smaller number to stay conservative. This essentially follows [58, chapter 12.4]

The T-test has some advantages over the percentile bootstrap [58]: First, reliable estimates for the tails of the bootstrap distribution require many bootstrap samples. Especially, when smaller α levels are requested or corrections for multiple comparisons are used this can become computationally expensive or unreliable. Second, for small sample sizes the bootstrap distribution does not take the uncertainty about the variance of the distribution into account. This is a similar error as taking the standard normal instead of a T-distribution to define confidence intervals. Third, the bootstrap distributions are discrete, which is a bad approximation in the tails of the distribution. For example, a sample of 5 RDMs which are all positively related to the model is declared significantly related to the model at any α level, because all bootstrapped average evaluations are at least as high as the lowest individual evaluation. Fourth, for the 2D bootstrap and the bootstrap-crossvalidation we can give corrections for the variance, but lack techniques to generate adequate bootstrap samples directly.

We should also note that we expect the T-distribution to be a good approximation for our case: We expect fairly symmetric distributions for the differences between models and average them across subjects, which should lead to a quick convergence towards a normal distribution for the model performances and their differences.

5.1.6 Flexible models

As model types we implement 3 types of flexible models additional to the standard fixed model, which represents a single RDM to be tested:

1. a selection model, which states that one of a set of RDMs is the correct one
2. a one dimensional manifold model, which consists of an ordered list of RDMs and is allowed to linearly interpolate between neighboring RDMs
3. a linear mixing model, which states that the RDM is a (positively) weighted sum of a set of RDMs

These models all aim to represent the flexibility necessary to represent the uncertainty about the measurement model in different ways. It is important to allow models to fit to the data adjusting aspects of the measurement model, because the spatial smoothing and weighted averaging of features due to measurement methods can strongly influence the resulting RDMs, which can lead to wrong conclusions and generally bad model performance when the models are compared to measured RDMs [54].

The choice model implements the flexibility of the measurement model in perhaps the simplest way, by allowing a choice among a set of RDMs produced from the model under different assumptions about the effect of the measurement. For training we can simply evaluate each possible RDM on the training data and choose the best performing one as the model prediction for evaluation. This model implies no structure in which RDMs can be predicted, but can only handle a finite set of RDMs.

The one dimensional manifold model implements an ordered set of RDMs, where the model is allowed to interpolate between each pair of consecutive RDMs. This representation is helpful if the uncertainty about the measurements effect can be well summarized by one continuous parameter like the width of a smoothing kernel. Then we can sample a set of values for this parameter and use the simulated results as the basis RDMs for this kind of model. Then the model will provide an approximation to the continuous set of RDMs predicted by changing the parameter without requiring a method to optimize the parameter directly.

Finally, the linear mixing model represents the effect of weighting orthogonal features. In this case the overall RDM is a weighted sum of the RDMs generated by the individual features. Whenever the original model has a feature-based representation it can be sensible to assume that these features are represented with different weights or are differently amplified by the measurement process.

Another application of the linear mixing model stems from the observation that the expected RDM for measurements that independently randomly weight features is a linear combination of two RDMs, one based on treating features separately and one based on averaging all features before computing the RDM (see below).

Our methodology is not specific to these types of models and can be easily extended to other types of models. To do so, the only requirement is that there is a reasonably efficient fitting method to infer the best fitting parameters for a given

dataset of training RDMs. Indeed, new model types can be slotted into our toolbox by users by implementing only two functions: One that predicts a RDM based on a parameter vector and one that fits the parameter to a dataset.

As argued before these flexible models should be evaluated on different data than the ones used for fitting the model, e.g. by crossvalidation as explained above.

5.1.7 Random feature weighting

If measurements weight features identically and independently we can directly compute what the expected squared euclidean RDM for the measurements is. We use this calculation both to justify a linear weighting model and to compute the correct models in some of our simulations.

Formally we can show that this is true by the following calculation: Let $w_i v$ be the weighting for the i-th feature in the v-th voxel for two patterns $\mathbf{x}$ and $\mathbf{y}$ with feature values x_i and y_i. Then the expected squared euclidean distance in voxel space can be written as:

$$\begin{aligned}
\mathbb{E}\left[\frac{1}{N_v}\sum_v\left(\sum_i w_{iv}(x_i-y_i)\right)^2\right] &= \mathbb{E}\left[\left(\sum_i w_i(x_i-y_i)\right)^2\right] && (24)\\
&= \mathbb{E}\left[\sum_i w_i^2(x_i-y_i)^2\right]+\mathbb{E}\left[\sum_i\sum_{j\neq i} w_i w_j(x_i-y_i)(x_j-y_j)\right] && (25)\\
&= \mathrm{Var}\left[w\right]\sum_i(x_i-y_i)^2+\mathbb{E}^2\left[w\right]\sum_i\sum_j(x_i-y_i)(x_j-y_j) && (26)\\
&= \mathrm{Var}\left[w\right]\sum_i(x_i-y_i)^2+\mathbb{E}^2\left[w\right]\left(\sum_i x_i-\sum_i y_i\right)^2 && (27)
\end{aligned}$$

, i.e. in this case the expected RDM is a linear combination of the RDM based on individual features and the RDM based on the average across features weighted by the variance and the squared expected value of the weight distribution respectively. As averaging or filtering across space is interchangeable with feature weighting, we can also use this calculation to compute the expected RDM for models that combine averaging over space and across features as in our simulations. Then the RDM at some level of averaging over space is still always a linear combination of the feature-averaged and feature-separate RDMs at that level of spatial averaging.

5.2 Simulation and Evaluation

To evaluate our methods we use four kinds of simulations. First, we implement a simple simulation using a normal distribution for the original measurements, which corresponds to the matrix normal generative model we used for theoretical derivations elsewhere [52]. Second, we implement simulations based on deep neural networks and a simple approximation of voxel sampling. By choosing a new random voxel sampling per subject and using different randomly selected input images, we can test our methods with systematic variations across conditions and/or subjects. Third, we implement a simulation based on real fMRI data recombining measurements signals and noise to keep all complications found in true fMRI data. Last, we present simulations based on calcium imaging data from mice [62].

5.2.1 Matrix-normal simulation

As the simplest model for generating data for RSA we used a matrix normal model. For this model we start with a RDM. As the RDM can be computed from the co-variance matrix between conditions [27], we can find a covariance matrix that results in the given RDM and then generate responses with this covariance. This random pattern will then have the desired RDM.

Concretely, the second moment matrix G between conditions across voxels can be computed from the squared euclidean distance matrix as follows:

$$G = -\frac{1}{2}(HDH) \tag{28}$$

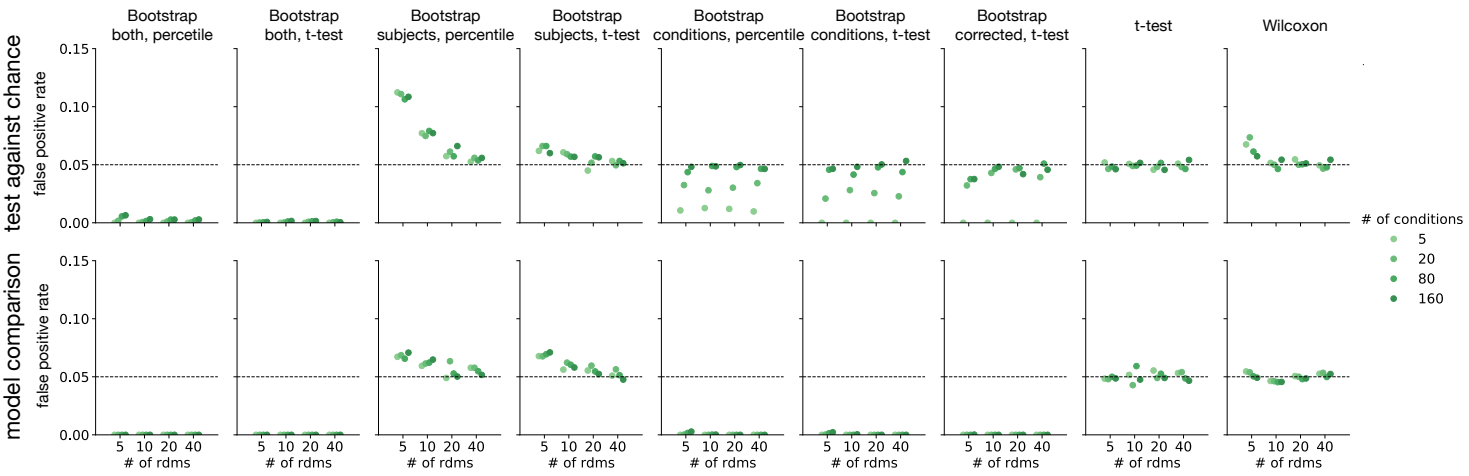

Figure 7: Basic Evaluation of the tests based on normally distributed simulated data. Each plot shows the false positive rate plotted against the number of RDMs and the number of conditions used. Ideal tests should fall on the dotted line at the nominal alpha level of 5%. For the model comparison test simulations, model performance was exactly equal on the conditions measured. This explain the very low false positive rate for bootstrapping across conditions for this test.

Using a centering matrix $\mathbf{H} = \mathbf{I}_k - \frac{1}{K}\mathbf{1}_k$, with $\mathbf{1}_k$ being a square matrix of ones. A dataset with this covariance across conditions has D as its squared euclidean RDM.

We can easily generate Gaussian data with a given second moment matrix and can thus generate data with any required RDM.

Comparison against 0: To generate H_0 data for testing our comparisons of models against 0 we choose both the model and the data RDMs as the distances between independent drawn Gaussian noise samples.

Model comparisons To generate H_0 data for model comparisons we first generate two random model-RDMs from independent standard normal noise data. We then normalize the model RDMs to have 0 mean and standard deviation of 1. Then we average the two RDMs, which yields a matrix with equal correlation to the two models. As a last step, we then subtract the minimum, to yield only positive distances and add the maximum distance to all distances once, such that the triangle inequality is guaranteed. As this last step only shifted the distance vector by a constant, the final distance vector still has the exact same correlation with the two model predictions.

These methods effectively draw the covariance over conditions from a standardized Wishart distribution with as many degrees of freedom as the number of measurement channels.

Data generation In both cases, we find a new configuration of data points that produce the desired RDM for each subject by converting the RDM into the second moment matrix via equation 28 and drawing random normal data as described at the beginning of this section. We then add additional iid. normal "measurement noise" to each entry of the data matrix. From this data matrix we then compute a squared euclidean distance RDM per subject and use this as the data RDM to enter our inference process. Finally, we run our inference methods on these data RDMs and the original model RDMs to check whether the false positive rate is low enough.

Selected conditions For each test and setting we generated 50 randomly drawn model RDMs and 100 data sets for each of these RDMs. We always used 200 measurement dimensions and tested all combinations of the following factors: 5, 10, 20 or 40 subjects, 5, 20, 80 or 160 conditions and all test types. As tests we used percentile tests and t-tests based on bootstrapping both dimensions, subjects only or conditions only, a standard t-test across subjects and a Wilcoxon rank sum test. For the corrected bootstrap, we only used the t-test based on the estimated variances, because we cannot draw bootstrap samples based on our correction.

Detailed Results For the tests for dependence and the tests for model comparison we plotted the results in Figure 7. For large datasets, the false positive rate converged towards a value at or below 5% in our simulations. This held for various bootstrap tests, as well as for t- and Wilcoxon tests. However, we observed a few interesting patterns (Online Methods: Fig. 7). Using bootstrap resampling of subjects only, we observed inflated false-positive rates of up to about

7% for small datasets when using the t-test and up to 12% for the simple percentile bootstrap test. These slightly too large false positive rates are due to the bootstrap estimating the biased variance estimate (dividing by N instead of $N-1$). For more than 20 subjects, we cannot distinguish the percentage from 5%. For the t-test and Wilcoxon rank-sum test, there were no such caveats as they consistently achieved a false-positive rate of about 5%.

When bootstrap resampling the conditions, the tests were conservative, achieving a false-positive rate below 1%, lower than the nominal 5% (at the expense of power). This held whether or not subjects were treated as a random effect: The t-tests based on either the corrected or the uncorrected two-factor bootstrap similarly had false-positive rates below 1%. This conservatism is expected because our matrix-normal simulations assumed a fixed set of conditions for which the models perform exactly equally well, rather than a random sample of conditions, which would lead to larger variance of the model performances.

Additionally, we ran a similar simulation, to test the tests against the noise ceiling. To do so, we generated a single random model and used the same RDM also for data generation. The results of this simulation are quickly summarized however, because the lower noise ceiling never significantly outperformed the true model indicating that the comparison against the lower noise ceiling is a very conservative test. This is most likely due to the difference between the lower bound on the noise ceiling and the true noise ceiling.

5.2.2 Neural network based simulation

Our simulations were based on the activities in the convolutional layers of AlexNet [47] in response to randomly chosen images from the ecoset validation set [82]. For each stimulus we computed the activities in the convolutional layer and took randomly chosen local averages to simulate the averaging of voxels. We then generated fMRI-like measurement timecourses to a randomly ordered short event related design by convolution with a hemodynamic response function and addition of autoregressive noise. We then ran a GLM analysis to estimate the response strength to each stimulus. From these estimated voxel responses, we computed data RDMs per subject and ran our proposed analysis procedures to compute model performances of different models which we also based on the convolutional layers of AlexNet.

For the network we used the implementation available for pytorch through the torchvision package [16].

Stimuli Stimuli were chosen independently from the validation set of ecoset by first choosing a category randomly and then sampling an image randomly from that category. These stimuli are natural images with categories chosen to approximate the relevance for human observers. The validation set contain 565 categories with 50 images each, i.e. 28250 images in total.

Noise-free Voxel response To compute the response strength of a voxel to a stimulus we computed a local average of the feature maps. We first convolved the feature maps with a Gaussian representing the spatial extend of the voxels, whose size we defined by its standard deviation relative to the overall size of the feature map. A voxel with size 0.05 would thus correspond to a Gaussian averaging area whose standard deviation is 5% of the size of the feature map. Voxel locations were then chosen uniformly randomly over the locations within the feature map. To average across features, we chose a weight for each feature and each voxel uniformly between 0 and 1 and then took the weighed sum as the voxel response.

fMRI simulation To generate timecourses we assumed a measurement was taken every 2 seconds and a new stimulus was presented during every second measurement, with no stimulus presented in the measurement intervals between stimulus presentations.

To generate a simulated fMRI response, we computed the stimulus by voxel response matrix and normalized it per subject to have equal averaged squared value. We then converted this into timecourses following the usual GLM assumptions and convolved the predictions with a hemodynamic response function. We set the hrf to the standard sum of two gamma distributions as assumed in SPM [83], normalized to an overall sum of 1.

We then added noise from an auto-regressive model of rank 1 (AR1) with covariance between pairs of voxels given by the overlap of the weighting functions of their weights. To control the strength of the autocorrelation, we set the coefficient for the previous datapoint to 0.5. To enforce the covariance between voxels, we multiplied the noise matrix with the cholesky decomposition of the desired covariance. To control the overall noise strength we scaled the final noise by a constant.

Each stimulus was presented once per run, with multiple stimulus presentations implemented as multiple runs.

Analysis To analyse the simulated data we ran a standard GLM analysis which yielded a β-estimate for each presented stimulus for each run of the experiment.

To compute RDMs we used Crossnobis distances based on leave one out crossvalidation over runs and the covariance of the residuals of the GLM. For this step we used the function implemented in our toolbox.

Fixed model definition As models to be compared we used the different layers of AlexNet. To generate an optimal model RDM we applied two transformations to mimic the average effect of voxel sampling. First we convolved the representation with the spatial receptive field of the voxels to mimic the spatial averaging effect. To capture the effect of pooling the features with non-negative weights, we then computed a weighted sum of the RDM containing the features separately and one RDM based on the summed response across features weighted with weights 1 and 3.

This weighting computes the expected euclidean distance of patterns under our random weighting scheme as we showed above (Online Methods 5.1.7: For our $w_i \sim U(0,1)$ the expected value is $\mathbb{E}(w) = \frac{1}{2}$ and the variance is $\mathrm{Var}(w_i) = \frac{1}{12}$ such that the weights for the RDM based on the individual features is $\frac{1}{4}$ and the weight for the RDM based on the summed feature response is $\frac{1}{12}$, i.e. a 3:1 weighting.

Based on this weighting we generated a fixed model for each individual processing step in Alexnet including the non-linearities and pooling operations resulting in 12 models predicting a fixed RDM.

Tested conditions For the large deep neural network based simulation underlying the results in Figure 3, we chose a base set of factors which we crossed with all other conditions and a separate set of factors which were not crossed with each other but only with the base set.

Into the base set of factors we included the following factors: Which experimental parameters were changed over repetitions of the experiment (None, subjects, stimuli or both) and which bootstrapping method we applied (over stimuli, over subjects, over both or applying the bootstrap correction). We applied all 4 bootstrapping conditions to the simulations in which none of the parameters were varied, the fitting ones to the subject and stimulus varying simulations and the bootstrap with and without correction for to the simulations were both parameters varied over repetitions resulting in 8 conditions for variation and bootstrap. Additionally, we included the number of subjects (5, 10, 20, 40 or 80) and the number of stimuli (10, 20, 40, 80, 160). For each set of conditions we thus ran $8 \times 5 \times 5 = 200$ conditions.

Other factors we varied were: The number of repeats, which we set to 4 usually and tested 2 and 8. The layer we used to simulate the data, which we usually set to layer number 8 which corresponds to the output of the 3rd convolutional layer, and also tried 2, 5, 10 and 12, which correspond to the other 4 convolutional layers of AlexNet. The size of the voxels which we usually set to 0.05, i.e. we set the standard deviation of the Gaussian to 5% of the size of the feature map. As variations we tried 0, 0.25 and ∞, i.e. no spatial pooling, a quarter of the size of the feature map as standard deviation and an average over the whole feature map. Finally, we varied the number of voxels, which we usually set to 100, but tried 10 and 1000 additionally. In total we thus ran $3 + 5 + 4 + 3 = 15$ sets of conditions with 200 conditions each resulting in 3000 conditions, with a grand total of $300\,000$ simulations.

Bootstrap-crossvalidation To test the precision and consistency of the calculations for the bootstrap wrapped crossvalidation (Fig. 4 a & b), we needed repeated analyses for the same datasets. For this simulations we thus simulated only 10 datasets for the standard conditions, 20 subjects and 40 stimuli, while varying both stimuli and subjects and then ran repeated analyses on these datasets. For each setting we ran 100 repeated analysis of each dataset. As conditions we chose 2,4,8,16, and 32 crossvalidation assignments for 1000 bootstrap samples and additionally variants with only 2 crossvalidation folds and 2000, 4000, 8000, or 16000 bootstrap samples.

RDM comparison measures To evaluate different comparison measures (Fig 4 c) we simulated data with our standard conditions and 10 subjects, 40 stimuli and 2 repeats, changing which parameters varied over repetitions of the experiment as in the main simulation, but omitted all bootstrapping. For this simulation we enforced that the first simulation in each condition used the same stimuli and subjects. The different measures for comparing RDMs were here applied to the same experimental data.

Flexible model treatment To test whether our methods are adequate for estimating the variability for model performances of flexible models (Fig. 4 d, e & f), we ran our standard settings for 20 subjects and 40 stimuli and drawing new subjects and new stimuli, while replacing the fixed models per layer with flexible models of different kinds.

We generated models by combining models with different assumptions about the voxel pooling pattern: We varied two factors: (1) how feature weighting was handled: full, i.e. predicted distances are euclidean distances in the original feature space, avg, i.e. distances are the differences in the average activation across features, or 'weighted', i.e. the weighted average of these two models, that corresponds to the expected RDM under the weight sampling we simulated. (2) how averaging over space was handled.

We first used different kinds fixed models, which serve as the building blocks for the flexible models. We varied two aspects of the measurement models applied: How large voxels are assumed to be (no pooling, $std = 5\%$ of the image size and pooling over the whole feature maps) and how the features pooling is handled (no pooling, average feature or the correct weighting assumed for the fixed models previously). These 3×3 combinations are the 9 fixed model variants.

We then generated selection models which had a range of voxel sizes to choose from (no pooling, std $= 1\%, 2\%, 5\%, 10\%, 20\%, 50\%$ of the image size and pooling across the whole feature map). For the treatment of pooling over features we used 4 variants: For the first three called full, average and weighted we used one of the types of fixed models to generate the RDMS. For the last we allowed both the RDMs used by the full models and the ones used by the average models as a choice.

As an example of a linearly weighted model we generated a model which was allowed to use a linear weighting the four corner cases: no feature pooling and no spatial pooling, average feature and no spatial pooling, a global average per feature map and the RDM induced by pooling over all locations and features. The model was then allowed to predict any linear combination of the features to fit the data RDMs.

5.2.3 Signal to noise ratio evaluation

To quantify how much increasing the number of measurements along one of the experimental factors improved signal to noise ratio, we can use the slope of a regression line for the signal-to-noise ratio against the number of measurements in log-log space, which corresponds to the exponent of the power law relationship (see Fig. 3 in the main text). We observe that increasing the number of conditions ($slope = 0.935$) is slightly more effective than increasing the number of subjects ($slope = 0.690$), and increasing the number of repeated measurements is most effective ($slope = 1.581$), probably due to the crossvalidation we employ. The crossvalidation across repeated measurements we use to yield unbiased distance estimates produces $\frac{m}{m-1}$ times the variance in the original RDM entries compared to the biased estimates without crossvalidation. This provides an additional benefit for increasing the number of repeated independent measurements.

The signal-to-noise ratio depends on the sources of nuisance variation included in the simulations. In these particular simulations, resampling the conditions increases nuisance variation more than resampling the subjects (Fig. 3 g). This indicates that inference generalizing across conditions is harder than inference generalizing across subjects in these simulations. For small noise levels the SNR is much higher when nothing is varied over repetitions or only subjects are varied than when the stimuli are also varied. At large measurement noise levels this effect disappears, because the measurement noise becomes the dominant factor.

The intuition to explain our observations about the signal-to-noise ratio is that it is most helpful to take more samples along the dimension which currently causes most variation in the results. Clearly our variation in stimuli caused more variance than our variation in voxel sampling to simulate subject variability, which causes sampling more stimuli to be more effective. Also, we simulate sufficiently high noise levels, such that reducing measurement noise remains effective. Beyond this effect, more repetitions are more profitable due to crossvalidation, which increases variance less the more independent measurements are already available.

Additionally, we observe that an intermediate voxel size (Gaussian kernel width) yields the highest model discriminability as measured by the SNR (see Fig. 3 h in the main text). When each voxel averages over a large area, information in fine-grained patterns of activity is lost, which is detrimental to model selection. The fall-off for very small voxels in our simulations is due to randomly sampled voxels covering the feature map less well leading to greater variability. In real fMRI experiments, we don't expect this effect to play a role, as we expect voxels to always cover the whole brain area, such that smaller voxels correspond to more voxels, which are clearly beneficial for better model selection. We do nonetheless expect a fall of for small voxel sizes for real fMRI experiments as well, because very small voxel sizes lead to a steep increase in instrument noise for fMRI and the BOLD signal itself is not perfectly local to the neurons that cause it [84, 85, 86]. Thus, the dependence on voxel averaging size is what we expect for real fMRI experiments as well, albeit for different reasons. Also, it might be informative for other measurement methods like electro-physiology, that a local average can be preferable over perfectly local measurements for model selection, when the number of measured channels is limited.

5.2.4 fMRI data based simulation

With our fMRI data based simulation we aim to show that our analyses are correct and functional for real fMRI data which may contain additional statistical regularities, which we did not take into account in our deep neural network based simulations. To do so, we took a large published dataset of fMRI responses to images and sampled from this

dataset to generate datasets across which we would like to generalize. All scripts for the fMRI data based simulation are openly available on `https://github.com/adkipnis/fmri-simulations`.

Dataset For these simulations we used data from [60][3]. This dataset contains fMRI data collected from 5 subjects viewing natural images selected from ImageNet or imagining images from a category. For our simulations we used only the "test" datasets, which contain 50 different images from distinct categories, which were each presented 35 times to each subject giving us an overall reliable signal and repetitions to resample from.

We used the automatic MRI preprocessing pipeline implemented in fMRIPrep 1.5.2. ([87]; [88]; RRID:SCR_016216), which is based on *Nipype* 1.3.1 ([89]; [90]; RRID:SCR_002502). This program was also used to produce the following description of the preprocesing performed:

Anatomical data preprocessing The T1-weighted (T1w) image was corrected for intensity non-uniformity (INU) with `N4BiasFieldCorrection` [91], distributed with ANTs 2.2.0 [92, RRID:SCR_004757], and used as T1w-reference throughout the workflow. The T1w-reference was then skull-stripped with a *Nipype* implementation of the `antsBrainExtraction.sh` workflow (from ANTs), using OASIS30ANTs as target template. Brain tissue segmentation of cerebrospinal fluid (CSF), white-matter (WM) and gray-matter (GM) was performed on the brain-extracted T1w using `fast` [FSL 5.0.9, RRID:SCR_002823, 93]. Brain surfaces were reconstructed using `recon-all` [FreeSurfer 6.0.1, RRID:SCR_001847, 94], and the brain mask estimated previously was refined with a custom variation of the method to reconcile ANTs-derived and FreeSurfer-derived segmentations of the cortical gray-matter of Mindboggle [RRID:SCR_002438, 95]. Volume-based spatial normalization to one standard space (MNI152NLin2009cAsym) was performed through nonlinear registration with `antsRegistration` (ANTs 2.2.0), using brain-extracted versions of both T1w reference and the T1w template. The following template was selected for spatial normalization: *ICBM 152 Nonlinear Asymmetrical template version 2009c* ([96], RRID:SCR_008796; TemplateFlow ID: MNI152NLin2009cAsym).

Functional data preprocessing For each of the 35 BOLD runs found per subject (across all tasks and sessions), the following preprocessing was performed. First, a reference volume and its skull-stripped version were generated using a custom methodology of *fMRIPrep*. The BOLD reference was then co-registered to the T1w reference using `bbregister` (FreeSurfer) which implements boundary-based registration [97]. Co-registration was configured with six degrees of freedom. Head-motion parameters with respect to the BOLD reference (transformation matrices, and six corresponding rotation and translation parameters) are estimated before any spatiotemporal filtering using `mcflirt` [FSL 5.0.9, 98]. BOLD runs were slice-time corrected using `3dTshift` from AFNI 20160207 [99, RRID:SCR_005927]. The BOLD time-series, were resampled to surfaces on the following spaces: *fsaverage5*, *fsaverage6*. The BOLD time-series (including slice-timing correction when applied) were resampled onto their original, native space by applying the transforms to correct for head-motion. These resampled BOLD time-series will be referred to as *preprocessed BOLD in original space*, or just *preprocessed BOLD*. The BOLD time-series were resampled into standard space, generating a *preprocessed BOLD run in ['MNI152NLin2009cAsym'] space*. First, a reference volume and its skull-stripped version were generated using a custom methodology of *fMRIPrep*. Several confounding time-series were calculated based on the *preprocessed BOLD*: framewise displacement (FD), DVARS and three region-wise global signals. FD and DVARS are calculated for each functional run, both using their implementations in *Nipype* [following the definitions by 100]. The three global signals are extracted within the CSF, the WM, and the whole-brain masks. Additionally, a set of physiological regressors were extracted to allow for component-based noise correction [*CompCor*, 101]. Principal components are estimated after high-pass filtering the *preprocessed BOLD* time-series (using a discrete cosine filter with 128s cut-off) for the two *CompCor* variants: temporal (tCompCor) and anatomical (aCompCor). tCompCor components are then calculated from the top 5% variable voxels within a mask covering the subcortical regions. This subcortical mask is obtained by heavily eroding the brain mask, which ensures it does not include cortical GM regions. For aCompCor, components are calculated within the intersection of the aforementioned mask and the union of CSF and WM masks calculated in T1w space, after their projection to the native space of each functional run (using the inverse BOLD-to-T1w transformation). Components are also calculated separately within the WM and CSF masks. For each CompCor decomposition, the *k* components with the largest singular values are retained, such that the retained components' time series are sufficient to explain 50 percent of variance across the nuisance mask (CSF, WM, combined, or temporal). The remaining components are dropped from consideration. The head-motion estimates calculated in the correction step were also placed within the corresponding confounds file. The confound time series derived from head motion estimates and global signals were expanded with the inclusion of temporal derivatives and quadratic terms for each [102]. Frames that exceeded a threshold of 0.5 mm FD or 1.5 standardized DVARS were annotated as motion outliers. All resamplings can be performed with *a single interpolation step* by composing all the pertinent transformations (i.e. head-motion transform matrices, susceptibility distortion correction when available, and co-registrations to anatomical and output spaces). Gridded (volumetric) resamplings were performed using

[3] as available from `https://openneuro.org/datasets/ds001246/versions/1.2.1`

`antsApplyTransforms` (ANTs), configured with Lanczos interpolation to minimize the smoothing effects of other kernels [103]. Non-gridded (surface) resamplings were performed using `mri_vol2surf` (FreeSurfer).

Many internal operations of *fMRIPrep* use *Nilearn* 0.5.2 [104, RRID:SCR_001362], mostly within the functional processing workflow. For more details of the pipeline, see the section corresponding to workflows in *fMRIPrep*'s documentation.

The above boilerplate text was automatically generated by fMRIPrep with the express intention that users should copy and paste this text into their manuscripts *unchanged.* It is released under the CC0 license.

Region selection Visual areas were defined according to the surface atlas by Glasser et al. [61]. For our simulations we used the following 10 visual areas as ROIs, joining areas from the atlas to avoid too small ROIs (the name of the areas in the atlas is given in brackets): V1 (V1), V2 (V2), V3 (V3), V4 (V4), ventral visual complex (VVC), ventromedial visual area (VMV1, VMV2, VMV3), parahippocampal place area (PHA1, PHA2, PHA3), fusiform face area (FFC), inferotemporal cortex (TF, PeEc), and MT / MST (MT, MST). The areas were selected separately for the two hemispheres.

To map the atlas onto individual subject's brain space we used the mappings estimated by FreeSurfer with fmriprep's standard settings. The Glasser Atlas was registered to each participant's native space with mri_surf2surf , and voxels labeled using mri_annotation2label . Next, each ROI was mapped to native T1w volumetric space with mri_label2vol . To cover as many contiguous voxels as possible, the resulting masks were inflated with mri_binarize and every voxel outside of the volume between the pial surface and white matter was eroded with mris_calc . To convert the resulting masks to T2*w space we used custom python scripts: First, masks were smoothed with a Gaussian kernel of FWHM = 3 mm, resampled to T2*w space using nearest neighbor interpolation, and finally thresholded. The threshold for each mask was set to equalize mask volume between T1w and T2*w space. Finally, voxels with multiple ROI assignments were removed from all ROIs but the one with the highest pre-threshold value. Voxels outside the fMRIPrep-generated brain mask were removed from all generated 3d-masks of ROIs.

General Linear Modeling For extracting response patterns from the measurements we used two General Linear Models (GLMs). In the first GLM, we regressed out noise sources and in the second we estimate stimulus responses. This two step process is advantageous in this case where stimulus predictors and noise predictors are highly collinear [4]: It allows us to attribute all variance that could be attributed to the noise sources uniquely to them and not to effects of stimulus presentation. This is closer to the original papers analysis, leads to higher reliability and generates the second GLM as a stage at which we can adequately model the noise with a relatively simple AR(2) model.

General Linear Modeling was performed in SPM12 (`http://www.fil.ion.ucl.ac.uk/spm`). No spatial smoothing was applied, models were estimated using Restricted Maximum Likelihood on top of Ordinary Least Squares and auto-correlations were taken into account using SPM's inbuilt AR(1) method.

For the first GLM, we used the following noise regressors from the ones provided by fMRIprep: An intercept for each run, the 6 basic motion parameters and their derivatives, 6 cosine basis functions to model drift, FD, DVARS and the first 6 aCompCors with the largerst eigenvalues. All runs were pooled to get the best noise parameter estimates possible.

We interpret the residuals from the first GLM as a denoized version of the fMRI signal and use them as input for a second GLM separately for each run to estimate stimulus effects: Stimulus-specific regressors were generated by convolving stimulus onset time-series with the canonical HRF without derivatives. From this GLM we kept the estimated β coefficients $\hat{\beta}_i \in \mathbb{R}^p$ for each stimulus and the residuals r_i for further processing.

Resampling To sample a single run for further analysis, we randomly chose a run from the measured data without replacement. To expand the set of possible datasets, we then generated a new simulated BOLD signal $\tilde{y}_i$ for each voxel i at the stage of the second level GLM. To do so, we model the data as a GLM with an AR(2) model for the noise and then generate a new timecourse by permuting the residuals η_i of the AR(2) model. As we apply the same permutation to each voxel, this procedure largely preserves spatial noise covariance.

In mathematical formulas this process can be described as follows: Let p be the number of stimuli, n be the number of scans per run, and $y_i \in \mathbb{R}^n$ be the denoized BOLD-response of voxel i. We can then use the design matrix of the run $X \in \mathbb{R}^{n \times p}$, the point estimate $\hat{\beta}_i \in \mathbb{R}^p$ for the parameter values and corresponding residuals $r_i \in \mathbb{R}^n$ estimated by SPM to simulate a new data run:

[4] As there is only one presentation of each stimulus per run and as they cover the whole run they can together form almost any sufficiently slow variation.

$$\tilde{y}_i = X\hat{\beta}_i + \lambda \cdot \tilde{r}_i, \qquad \tilde{r}_{i,t} = \hat{w}_{i,1}\tilde{r}_{i,t-1} + \hat{w}_{i,2}\tilde{r}_{i,t-2} + \tilde{\eta}_{i,t} \tag{29}$$

where $w_{i,1}$ and $w_{i,2}$ are the estimated parameters of an AR(2) model fitted to the residuals $\mathbf{r}_i$. Its residuals are denoted η_i and were randomly permuted to give $\tilde{\eta}_i$ using the same permutation for all voxels in a run.

We then saved this dataset in the same format as the original data and re-ran the second level GLM on these simulated data to generate noisy estimates stimulus responses in each voxel.

To generate a dataset for a simulation, we randomly selected a subset of runs and stimuli for the analysis.

RDM calculation & comparison We use crossnobis RDMs for this simulation, testing 4 different estimates for the noise covariance: We either use the identity, univariate noise normalization, or a shrinkage estimate of the covariance based on the covariance of the residuals, or based on the covariance of the individual runs' mean-centered β estimates.

For comparing RDMs, we use the cosine similarity throughout.

Model RDMs We use the RDMs of different ROIs as models, effectively testing how well our methods recover the data generating ROI. The model RDM for each ROI is the pooled RDM across all subjects and runs computed by the same noise normalization method as the one used for the data RDMs. Data for these RDMs stemmed from the original results of the second level GLM, making them less noisy than any RDM stemming from the simulated data.

Simulation design For each condition we ran 24 repeats to estimate the true variability of results and ran all combinations of the following conditions: We used $2, 4, 8, 16$, or 32 runs per simulation (5 variants). We used $5, 10, 20, 30$, or 50 stimuli (5 variants). We scaled the noise by $0.1, 1.0$, or 10.0 (3 variants). We used each of the 20 ROIs for data generation once (20 variants). And we used the 4 methods for estimating the noise covariance (4 variants). Resulting in $24 \cdot 5 \cdot 5 \cdot 3 \cdot 20 \cdot 4 = 144,000$ different simulations.

5.2.5 Calcium imaging data based simulation

For the calcium imaging data based simulation we used the Allen institutes mouse visual coding calcium imaging data available at https://observatory.brain-map.org/visualcoding/ [62]. Detailed information on the recording techniques can be obtained from the original publications and with the dataset.

We used the 'natural scenes' data, which consists of measured calcium responses to 118 natural scenes. The natural scenes were shown for 250ms each without an inter stimulus interval in random order. In each session each image was present 50 times.

From this dataset, we selected all experimental sessions, which contained a natural scenes experiment. Additionally, we restricted ourselves to three relatively broad cre driver lines, which target excitatory neurons relatively broadly: 'Cux2-CreERT2', 'Emx1-IRES-Cre', and 'Slc17a7-IRES2-Cre'. For further analyses we ignore which driver line was used to achieve enough data for resampling. This resulted in 174 experimental sessions from 91 mice with 146 cells recorded on average (range: 18-359). Of these recordings, 35 came from laterointermediate area, 32 from posteromedial visual area, 23 from rostrolateral visual area, 46 from primary visual cortex, 16 from anteromedial area and 22 from anterolateral area.

To quantify the response of a neuron to the stimuli, we used the fully pre-processed $\frac{df}{F}$ traces as provided by the dataset. We then extract the measurements from the frame after the one marked as stimulus onset till the stated stimulus endframe resulting in six or seven frames per stimulus presentation. As a response per neuron we then simply took the average of these frames.

To compute RDMs based on this data, we used Crossnobis distances based on different estimates of the noise covariance matrix based on the variance of the stimulus repetitions around the average neural response for each stimulus. We either used: an identity matrix, effectively calculating a crossvalidated euclidean distance; a diagonal matrix of variances, corresponding to univariate noise normalization; a shrinkage estimate towards a constant diagonal matrix [65], or a shrinkage estimate shrunk towards the diagonal of sample variances [66].

To generate new datasets, we randomly sampled subsets of stimuli, mice, runs, and cells from a brain area without replacement. To exclude possible interactions we avoided sampling multiple sessions recorded from the same mouse by sampling the mice and then randomly sampling from the sessions of each mouse, if there were more than one. For this dataset, we did not use any further processing of the data.

As conditions for this simulation, we performed all combinations of the following factors: 20, 40 or 80 cells per experiment; 5 10 or 15 mice; 10, 20 or 40 stimuli; 10, 20 or 40 stimulus repeats; the four types of noise covariance

estimates; 4 types of rdm comparison: cosine similarity, correlation, whitened cosine similarity and whitened correlation; whether the bootstrap was corrected; and the 6 brain areas. This resulted in $3 \times 3 \times 3 \times 3 \times 4 \times 4 \times 2 \times 6 = 15552$ simulation conditions for which we simulated 100 simulations each.

As models for the simulations we used the average RDM for each brain area as a fixed RDM model for that brain area. Thus the models are not independent from the data in our main simulation. This is not problematic for checking the integrity of our inference methods, but does not show that we can indeed differentiate brain areas based on their RDMs. To show that retrieving the brain area is possible as displayed in Fig. 6 b in the main text, we performed leave one out crossvalidation across mice, i.e. we chose the RDM models for the brain areas based on all but one mice and evaluated the RDM correlation with the left out mouse's RDM.