

# The Voxelwise Modeling framework: a tutorial introduction to fitting encoding models to fMRI data

Tom Dupré la Tour<sup>1</sup>, Matteo Visconti di Oleggio Castello<sup>1</sup>, and Jack L. Gallant<sup>1,2,\*</sup>

<sup>1</sup>Helen Wills Neuroscience Institute, University of California, Berkeley, CA 94720, USA

<sup>2</sup>Department of Neuroscience, University of California, Berkeley, CA 94720, USA

\*corresponding author: gallant@berkeley.edu

## Abstract

Voxelwise modeling (VM) is a powerful framework for functionally mapping the brain. In the VM framework, features are extracted from the stimulus (or task) and used in an encoding model to predict brain activity. If the encoding model is able to predict brain activity in some part of the brain, then one may conclude that some information represented in the features is also encoded in the brain. In VM, a separate encoding model is fitted on each spatial sample (i.e. each voxel). VM has many benefits compared to other methods for analyzing and modeling neuroimaging data. Most importantly, VM can use large numbers of features simultaneously, which enables the analysis of complex naturalistic stimuli and tasks. Therefore, VM can produce high-dimensional functional maps that reflect the selectivity of each voxel to large numbers of features. Moreover, because model performance is estimated on a separate test dataset not used during fitting, VM minimizes overfitting and inflated Type I error confounds that plague other approaches, and the results of VM generalize to new subjects and new stimuli. Despite these benefits, VM is still not widely used in neuroimaging, partly because no tutorials on this method are available currently. To demystify the VM framework and ease its dissemination, this paper presents a series of hands-on tutorials accessible to novice practitioners. The VM tutorials are based on free open-source tools and public datasets, and reproduce the analysis presented in previously published work.<sup>1</sup>

**Keywords** Neuroimaging; Encoding models; Tutorials; Python

## Introduction

Voxelwise modeling (VM) is a powerful framework for functionally mapping the brain. The VM framework builds upon encoding models to map brain representations from functional magnetic resonance imaging (fMRI) recordings. (See Section 1 for a brief overview of the VM framework.) Over the last two decades, the VM framework has been used to map brain representations generated by visual images [Hansen et al., 2004, Thirion et al., 2006, Kay et al., 2008, Dumoulin and Wandell, 2008, Naselaris et al., 2009, Schönwiesner and Zatorre, 2009, Stansbury et al., 2013, Güçlü and van Gerven, 2015, Eickenberg et al., 2016, St-Yves and Naselaris, 2018, Nonaka et al., 2021, Konkle and Alvarez, 2022], movies [Nishimoto et al., 2011, Huth et al., 2012, Çukur et al., 2013, Eickenberg et al., 2016, Wen et al., 2018, Lescroart and Gallant, 2019, Popham et al., 2021, Khosla et al., 2021, Dupré la Tour et al., 2021], music [Kell et al., 2018], semantic concepts [Mitchell et al., 2008, Wehbe et al., 2014, Deniz et al., 2023], and narrative language [Huth et al., 2016, de Heer et al., 2017, Jain and Huth, 2018, Deniz et al., 2019, Toneva and Wehbe, 2019, Jain et al., 2020, LeBel et al., 2021, Caucheteux and King, 2022].

The VM framework provides several critical improvements over other fMRI data analysis methods. First, most procedures for analyzing fMRI data can only accommodate a small number of different conditions

---

<sup>1</sup>The tutorials are hosted at [https://github.com/gallantlab/voxelwise\\_tutorials](https://github.com/gallantlab/voxelwise_tutorials).

[Friston et al., 1994, Penny et al., 2011, Poline and Brett, 2012]. In contrast, VM can efficiently analyze many different stimulus and task features simultaneously. This enables the analysis of complex naturalistic stimuli and tasks, as advocated for example in [Wu et al., 2006, Hasson et al., 2010, Dubois and Adolphs, 2016, Matusz et al., 2019, Hamilton and Huth, 2020].

Second, experiments in many areas of neuroscience suffer from a lack of reproducibility [Collaboration, 2015] and an inflated rate of false positives (Type I error) [Bennett et al., 2009, Button et al., 2013, Cremers et al., 2017]. This is caused by an over-reliance on null hypothesis testing, an underappreciation of the importance of prediction accuracy and generalization in model selection, and a failure to control overfitting [Wu et al., 2006, Yarkoni and Westfall, 2017, Yarkoni, 2022]. In contrast, VM is a predictive modeling framework that evaluates model performance on a separate test dataset not used during fitting. Thus, VM is not prone to overfitting and inflated Type I error, and generalizes to new subjects and new stimuli.

Third, most studies project data from individuals into a standardized template space and then average over the group, ignoring the substantial individual differences that have been observed in both anatomical and functional fMRI data [Tomassini et al., 2011, Dubois and Adolphs, 2016, Gordon et al., 2017, Valizadeh et al., 2018]. In contrast, to maintain maximal spatial resolution, VM performs all analyses in each subject’s native brain space. Because each subject provides their own fit and test data, each subject provides a complete replication of all hypothesis tests.

Fourth, most methods produce simple statistical brain maps (e.g. SPM [Friston et al., 1994], MVPA [Haxby et al., 2014], RSA [Kriegeskorte et al., 2008]). In contrast, VM produces high-dimensional functional maps that reflect the selectivity of each voxel to thousands of stimulus and task features.

Despite these critical improvements, the use of VM in neuroimaging is still relatively limited. This is likely because many in the fMRI community have not received training in modern methods of data science, and the lack of detailed instructions about how to fit encoding models to fMRI data (but see [Naselaris et al., 2011, Ashby, 2019, chapter 15], and the [speech-model-tutorial](#)). Several recent efforts have sought to provide educational materials to support other neuroimaging methods, such as BrainIAK [Kumar et al., 2020], Neurohackademy, Neuromatch, Dartbrain [Chang et al., 2020b], or Naturalistic-Data [Chang et al., 2020a]. These educational projects cover a wide variety of topics in neuroimaging, and complement other existing resources from well-documented Python toolboxes such as pymvpa [Hanke et al., 2009] and nilearn [Abraham et al., 2014]. However, these existing projects do not provide a sufficient foundation for new users to use VM to implement and test encoding models.

To demystify VM and ease its dissemination, we have created a series of hands-on tutorials that should be accessible to novice practitioners. These tutorials use free open-source tools that have been developed by our lab and by the scientific Python community (see [Section 2](#)). The tutorials are based on a public dataset [Huth et al., 2022] and they reproduce some of the analyses presented in published work from our lab [Nishimoto et al., 2011, Huth et al., 2012, Nunez-Elizalde et al., 2019]. These tutorials were presented at the 2021 conference on Cognitive Computational Neuroscience (CCN; [video recording](#)).

This document provides a guide to the online tutorials. The first section of this guide presents a brief overview of the VM framework. The second section describes the technical implementation of the VM tutorials, listing the different tools used therein and the associated design decisions. The third section provides a brief overview of the VM tutorials content. The fourth section highlights some of the key analyses performed in the tutorials.

## 1 The voxelwise modeling framework

A fundamental problem in neuroscience is to identify the information represented in different brain areas. In the VM framework, this problem is solved using encoding models. An encoding model describes how various features of the stimulus (or task) predict the activity in some part of the brain [Wu et al., 2006]. Using VM to fit an encoding model to blood oxygen level-dependent signals (BOLD) recorded by fMRI involves several steps. First, brain activity is recorded while subjects perceive a stimulus or perform a task. Then, a set of

features (that together constitute one or more *feature spaces*) is extracted from the stimulus or task at each point in time. For example, a video might be represented in terms of amount of motion in each part of the screen [Nishimoto et al., 2011], or in terms of semantic categories of the objects present in the scene [Huth et al., 2012]. Each feature space corresponds to a different representation of the stimulus- or task-related information. The VM framework aims to identify if each feature space is encoded in brain activity. Each feature space thus corresponds to a hypothesis about the stimulus- or task-related information that might be represented in some part of the brain. To test this hypothesis for some specific feature space, a regression model is trained to predict brain activity from that feature space. The resulting regression model is called an *encoding model*. If the encoding model predicts brain activity significantly in some part of the brain, then one may conclude that some information represented in the feature space is also represented in brain activity. To maximize spatial resolution, in VM a separate encoding model is fit on each spatial sample in fMRI recordings (that is on each voxel), leading to *voxelwise encoding models*.

Before fitting a voxelwise encoding model, it is sometimes possible to estimate an upper bound of the model prediction accuracy in each voxel. In VM, this upper bound is called the noise ceiling, and it is related to a quantity called the explainable variance [Sahani and Linden, 2003, Hsu et al., 2004, Schoppe et al., 2016]. The explainable variance quantifies the fraction of the variance in the data that is consistent across repetitions of the same stimulus. Because an encoding model makes the same predictions across repetitions of the same stimulus, the explainable variance is the fraction of the variance in the data that can be explained by the model.

To estimate the prediction accuracy of an encoding model, the model prediction is compared with the recorded brain response. However, higher-dimensional encoding models are more likely to overfit to the training data. Overfitting causes inflated prediction accuracy on the training set and poor prediction accuracy on new data. To minimize the chances of overfitting and to obtain a fair estimate of prediction accuracy, the comparison between model predictions and brain responses must be performed on a separate test data set that was not used during model training. The ability to evaluate a model on a separate test data set is a major strength of the VM framework. It provides a principled way to build complex models while limiting the amount of overfitting. To further reduce overfitting, the encoding model is regularized. In VM, regularization is obtained by ridge regression [Hoerl and Kennard, 1970], a common and powerful regularized regression method.

To take into account the temporal delay between the stimulus and the corresponding BOLD response (i.e. the hemodynamic response), the features are duplicated multiple times using different temporal delays [Dale, 1999]. The regression then estimates a separate weight for each feature and for each delay. In this way, the regression builds for each feature the best combination of temporal delays to predict brain activity. This combination of temporal delays is sometimes called a finite impulse response (FIR) filter [Goutte et al., 2000]. By estimating a separate FIR filter per feature and per voxel, VM does not assume a unique hemodynamic response function.

After fitting the regression model, the model prediction accuracy is projected on the cortical surface for visualization. Our lab created the pycortex [Gao et al., 2015] visualization software specifically for this purpose. These prediction-accuracy maps reveal how information present in the feature space is represented across the entire cortical sheet. (Note that VM can also be applied to other brain structures, such as the cerebellum [LeBel et al., 2021] and the hippocampus; however, those structures are more difficult to visualize computationally.) In an encoding model, all features are not equally useful to predict brain activity. To interpret which features are most useful to the model, VM uses the fit regression weights as a measure of relative importance of each feature. A feature with a large absolute regression weight has a large impact on the predictions, whereas a feature with a regression weight close to zero has a small impact on the predictions. Overall, the regression weight vector describes the *feature tuning* of a voxel, that is the feature combination that would maximally drive the voxel's activity. To visualize these high-dimensional feature tunings over all voxels, feature tunings are projected on fewer dimensions with principal component analysis, and the first few principal components are visualized over the cortical surface [Huth et al., 2012, Huth et al., 2016]. These feature-tuning maps reflect the selectivity of each voxel to thousands of stimulus and task features.

In VM, comparing the prediction accuracy of different feature spaces within a single data set amounts to comparing competing hypotheses about brain representations. In each brain voxel, the best-predicting feature space corresponds to the best hypothesis about the information represented in that voxel. However, many

voxels represent multiple feature spaces simultaneously. To take this possibility into account, in VM a joint encoding model is fit on multiple feature spaces simultaneously. The joint model automatically combines the information from all feature spaces to maximize the joint prediction accuracy.

Because different feature spaces used in a joint model might require different regularization levels, VM uses an extended form of ridge regression that provides a separate regularization parameter for each feature space. This extension is called banded ridge regression [Nunez-Elizalde et al., 2019]. Banded ridge regression also contains an implicit feature-space selection mechanism that tends to ignore feature spaces that are non-predictive or redundant [Dupré la Tour et al., 2022]. This feature-space selection mechanism helps to disentangle correlated feature spaces and it improves generalization to new data.

To interpret the joint model, VM implements a variance decomposition method that quantifies the separate contributions of each feature space. Variance decomposition methods include variance partitioning [Lescroart et al., 2015], the split-correlation measure [St-Yves and Naselaris, 2018], or the product measure [Dupré la Tour et al., 2022]. The obtained variance decomposition describes the contribution of each feature space to the joint encoding model predictions.

## 2 Tutorials design

The VM tutorials are written in the Python programming language, and therefore benefit from a collection of freely-available open-source tools developed by the scientific Python community. This section lists these tools and describes the associated design decisions.

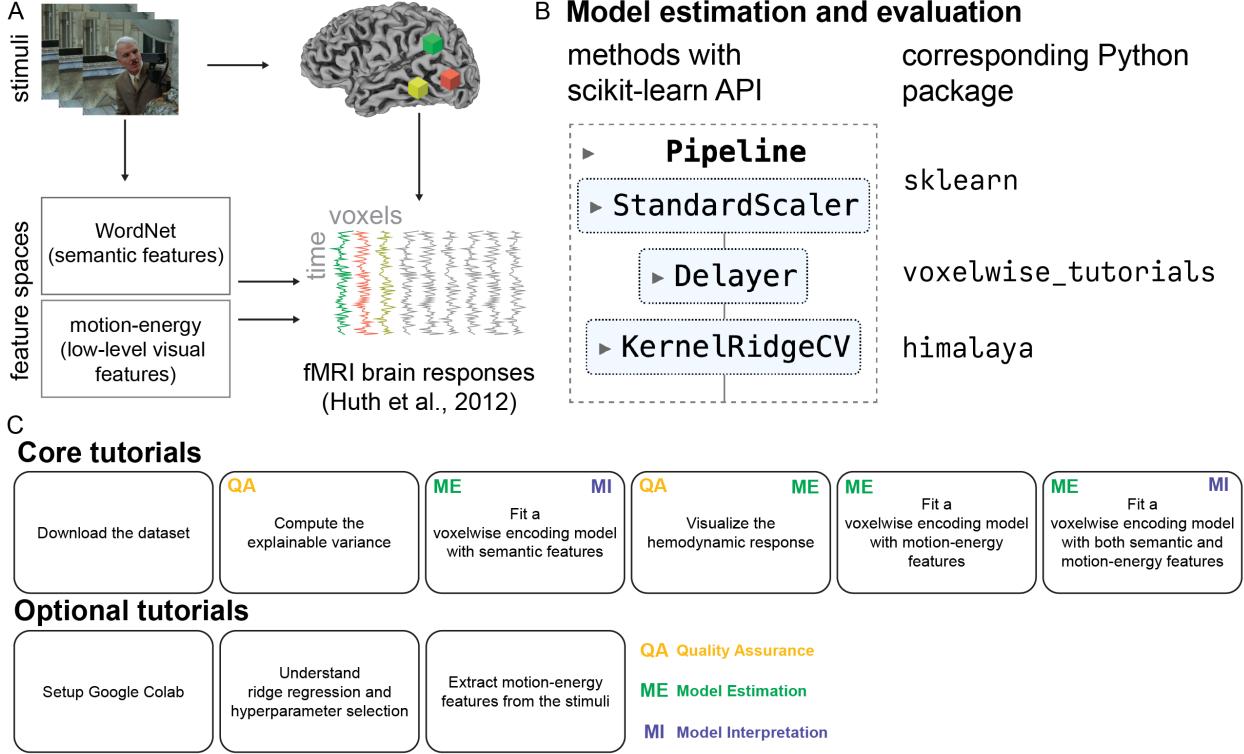
**Basic knowledge required.** The VM tutorials require some (beginner) skills in Python and numpy [Harris et al., 2020]. For an introduction to Python and numpy, we refer the reader to existing resources such as the [scientific Python lectures](#), or the numpy [absolute beginner guide](#). Additionally, the VM tutorials use terminology from scikit-learn [Pedregosa et al., 2011]. For an introduction to scikit-learn and its terminology, we refer the reader to the scikit-learn [getting-started guide](#) and [glossary of common terms](#).

**Tutorial format.** The VM tutorials are open source (BSD-3-Clause license) and freely available to the scientific community. They are under continuous development and open to external contributions. To track changes of the codebase over time, the code is version-controlled using git and hosted on the [Github platform](#). To make sure the code remains functional during development, the code is continuously tested using pytest and Github Actions.

To satisfy different user preferences, the VM tutorials are available in multiple formats. First, the VM tutorials can be downloaded and run locally, either as Python scripts or as jupyter notebooks [Kluyver et al., 2016]. Second, the notebooks can be run for free in the cloud (with GPU support) using [Google Colab](#). In this case, we concatenated all notebooks into a single notebook for convenience. Third, the VM tutorials can be explored pre-rendered on a dedicated [VM tutorials website](#), hosted for free by Github Pages. The website is built with sphinx [Brandl, 2010] and numpydoc. The python scripts are rendered in a notebook-like layout using sphinx-gallery [Nájera et al., 2020].

**Dataset.** The short-clips dataset used in the VM tutorials contains BOLD fMRI measurements made while human subjects viewed a set of naturalistic, short movie clips [Nishimoto et al., 2011, Huth et al., 2012]. The dataset is publicly available on the free [GIN platform](#) [Huth et al., 2022]. For convenience, the dataset is additionally hosted on a premium cloud provider with high bandwidth (Wasabi). To download the dataset, a custom user-friendly data loader is available in the VM tutorials. By using git-annex and datalad [Halchenko et al., 2021], the data loader can seamlessly switch between the different data sources depending on their availability. The data is stored using the HDF5 format, which enables direct access to array slices without loading the entire file in memory. The read-write operations are performed using h5py.

**Regression methods.** Voxelwise encoding models are based on regression methods such as ridge regression [Hoerl and Kennard, 1970] and banded ridge regression [Nunez-Elizalde et al., 2019]. Because a different



**Figure 1: Overview of the Voxelwise Modeling tutorials.** A) The tutorials are based on the public data from [Huth et al., 2012]. In that experiment, participants watched a series of short clips without sound while their brain activity was measured with fMRI. The public dataset contains two feature spaces: WordNet, which quantifies the semantic information in the stimuli, and motion-energy, which quantifies low-level visual features. Both feature spaces are used in the tutorials. B) Model estimation and model evaluation methods are implemented with a fully compatible scikit-learn API. For example, a voxelwise encoding model estimated with regularized regression is implemented as a scikit-learn Pipeline consisting of a StandardScaler step (a preprocessing scikit-learn step to standardize the features prior to model fitting), a Delayer step (a method implemented in the associated voxelwise\_tutorials Python package to delay the features prior to model fitting), and a KernelRidgeCV step (an efficient implementation of kernel ridge regression with cross-validation, implemented in the himalaya package developed by our lab). C) The current version of the tutorials consists of nine separate tutorials, six core tutorials, and three optional tutorials. The core tutorials cover aspects of the Voxelwise Modeling framework related to quality assurance, model estimation, and model interpretation. The optional tutorials include a tutorial for setting up Google Colab (which is useful when teaching these tutorials in a class); a tutorial about ridge regression, cross-validation, and the effect of hyperparameter selection on model prediction; and a tutorial that shows how to use pymoten (a Python package developed by our lab) to extract motion-energy features from the stimuli provided in the public dataset.

model is estimated for every voxel, and because a typical fMRI dataset at 3T contains about  $10^5$  voxels, fitting voxelwise encoding models can be computationally challenging. To address this challenge, the VM tutorials use algorithms optimized for large numbers of voxels, implemented in `himalaya` [Dupré la Tour et al., 2022]. To further improve computational speed, `himalaya` provides three different computational backends to fit regression models either on CPU (using `numpy` [Harris et al., 2020] and `scipy` [Virtanen et al., 2020]) or on GPU (using either `pytorch` [Paszke et al., 2019] or `cupy` [Nishino and Loomis, 2017]). The VM tutorials also use `scikit-learn` [Pedregosa et al., 2011] to define the regression pipeline and the cross-validation scheme.

**Additional tools.** Beyond the software infrastructure that underlies VM, the tutorials also provide a custom Python package called `voxelwise_tutorials`. This package contains a collection of helper functions used throughout the VM tutorials. For example, it includes tools to define the regression pipeline, and generic visualization tools based on `matplotlib` [Hunter, 2007], `networkx` [Hagberg et al., 2008], and `nltk` [Bird and Loper, 2004]. To preserve subject privacy, the dataset used in the VM tutorials does not contain any anatomical information. For this reason, the cortical visualization package `pycortex` [Gao et al., 2015] cannot be used. Instead, the `voxelwise_tutorials` package provides tools to replicate `pycortex` visualization functions using subject-specific `pycortex` mappers that are included with the dataset.

### 3 Tutorials overview

In the VM framework, a typical study can be decomposed into seven steps: (I) experimental design, (II) data collection, (III) data preprocessing, (IV) feature space extraction, (V) voxelwise encoding model estimation, (VI) model evaluation, (VII) model interpretation. These seven steps are described in detail in our upcoming review paper [Visconti di Oleggio Castello et al., 2024]. The present tutorials are designed as a complement to this review, proposing a practical implementation of the following steps: (IV) feature space extraction, (V) voxelwise encoding model estimation, (VI) model evaluation, (VII) model interpretation (Figure 1).

The VM tutorials are organized into nine notebooks that are best worked through in order. These notebooks are rendered in the [VM tutorials website](#). This section briefly describes the content of each notebook. The next section showcases four notebooks that cover key aspects of voxelwise modeling related to quality assurance, model estimation and evaluation, and model interpretation.

- 1. Download the dataset.** This notebook describes how to download the `short-clips` dataset [Huth et al., 2012, Huth et al., 2022]. This dataset contains BOLD fMRI responses in human subjects viewing a set of natural short movie clips. The dataset contains responses from five subjects, and the VM analysis can be replicated in each subject independently.
- 2. Setup of Google Colab (optional).** This optional notebook describes how to set up the VM tutorials to run in the cloud using Google Colab. This notebook should be skipped when running the VM tutorials on a local machine.
- 3. Compute the explainable variance.** This notebook provides a first glance at the data and describes how to compute the explainable variance. The explainable variance quantifies the fraction of the variance in the data that is consistent across repetitions of the same stimulus. This notebook also demonstrates how to plot summary statistics of each voxel onto a subject-specific flattened cortical surface using Pycortex mappers. This visualization technique is used multiple times in the subsequent notebooks.
- 4. Understand ridge regression and hyperparameter selection (optional).** This optional notebook presents ridge regression [Hoerl and Kennard, 1970], a regularized regression method used extensively in VM. The notebook also describes how to use cross-validation to select the optimal hyperparameter in ridge regression. This notebook is not specific to VM, and can be skipped by readers who are already familiar with ridge regression and hyperparameter selection.
- 5. Fit a voxelwise encoding model with semantic features.** This notebook shows how to fit a voxelwise encoding model to predict BOLD responses. The encoding model is fit with semantic features

extracted from the movie clips, reproducing part of the analysis presented in [Huth et al., 2012].

6. **Visualize the hemodynamic response.** This notebook describes how to use finite-impulse response (FIR) filters in voxelwise encoding models to take into account the hemodynamic response in BOLD signals.
7. **Extract motion-energy features from the stimuli (optional).** This optional notebook describes how to use pymoten [Nunez-Elizalde et al., 2021] to extract motion-energy features from the stimulus. Motion-energy features are low-level visual features extracted from the movie clips stimuli using a collection of spatio-temporal Gabor filters. This computation can take a few hours, so we have included precomputed motion-energy features in the short-clips dataset.
8. **Fit a voxelwise encoding model with motion-energy features.** This notebook shows how to fit a voxelwise encoding model with different features than in Notebook 5. Here, the encoding model is fit with motion-energy features extracted from the movie clips, reproducing part of the analysis presented in [Nishimoto et al., 2011]. Motion-energy features are low-level visual features extracted in Notebook 7. The voxelwise encoding model is fit using the same method as with semantic features (see Notebook 5). The semantic model and the motion-energy model are then compared over the cortical surface by comparing their prediction accuracies.
9. **Fit a voxelwise encoding model with both semantic and motion-energy features.** This notebook shows how to fit a voxelwise encoding model with two feature spaces jointly, the semantic and the motion-energy feature spaces.

## 4 Key tutorials

### Notebook 3: Compute explainable variance.

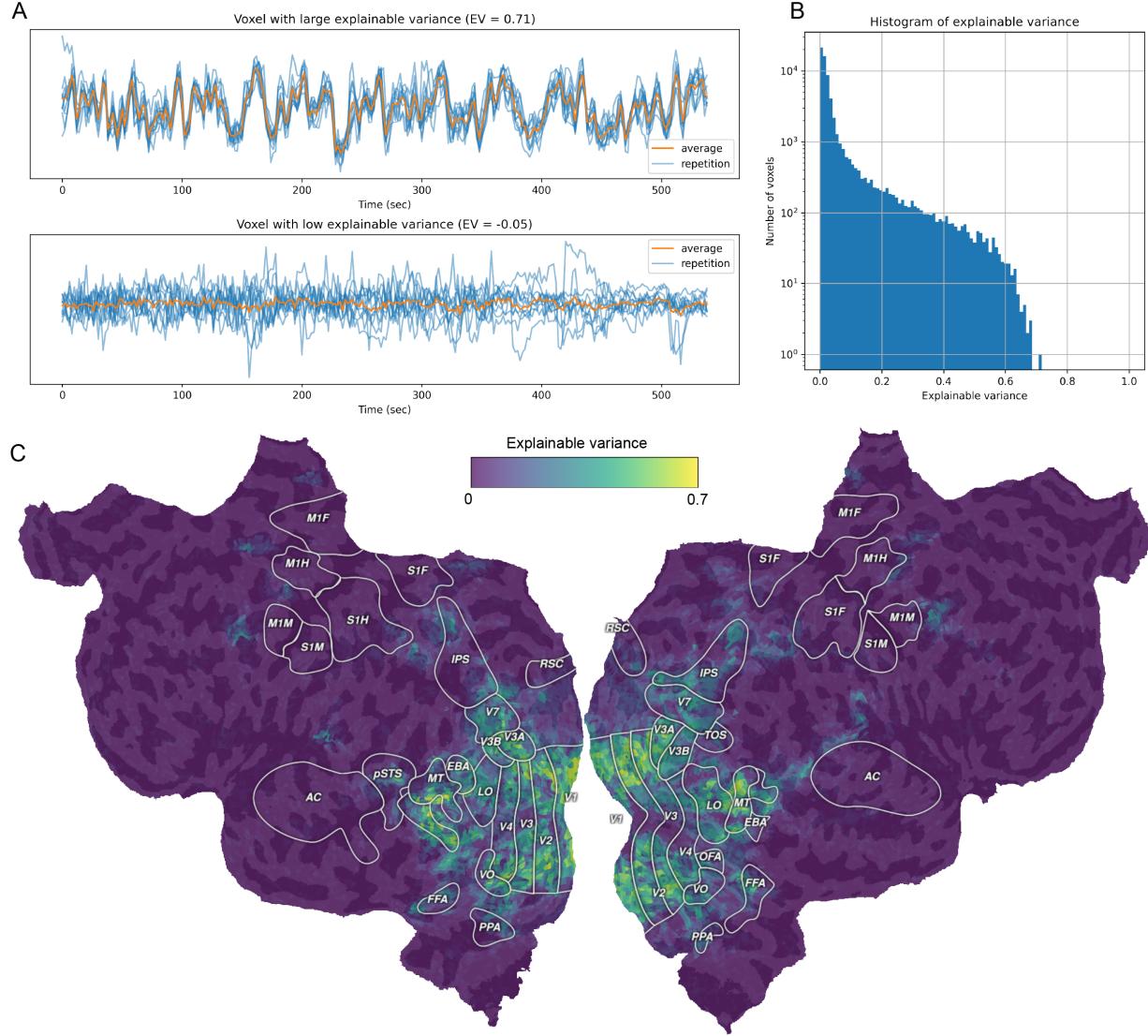
This notebook describes how to compute and interpret explainable variance. Explainable variance is proportional to the maximum prediction accuracy that can be achieved by any voxelwise encoding model, given the available test data [Hsu et al., 2004, Schoppe et al., 2016, Sahani and Linden, 2003]. In practice this is quantified as the fraction of the variance in the data that is consistent across repetitions of the same stimulus. In the voxelwise modeling framework, computing and visualizing explainable variance is a key quality assurance step, ensuring that the measured data provides enough signal to reliably estimate voxelwise encoding models (Figure 2).

### Notebook 5: Fit a voxelwise encoding model with semantic features

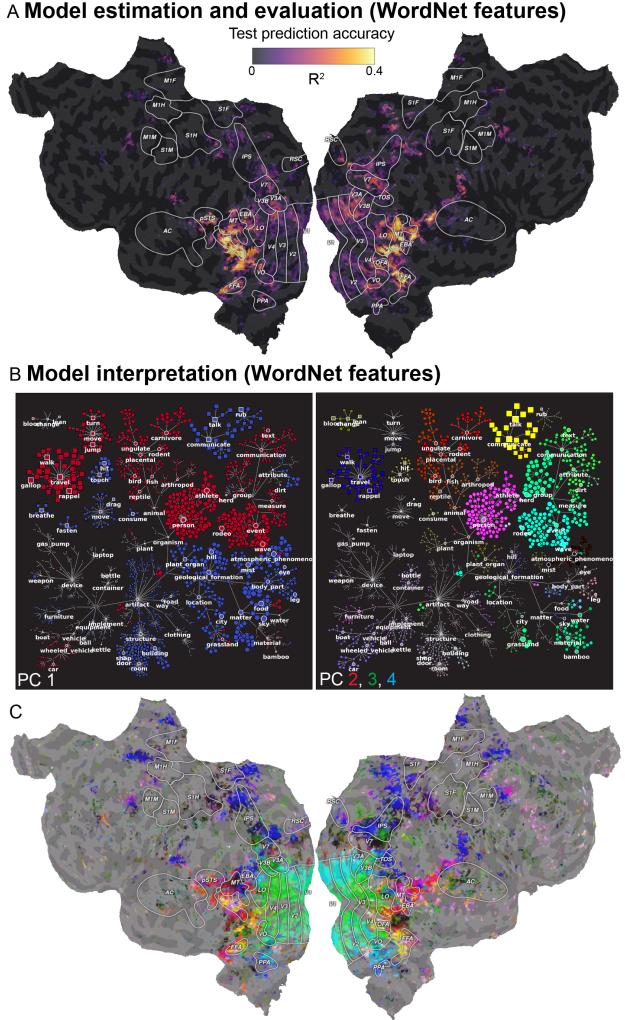
This notebook demonstrates how to fit a voxelwise encoding model to predict brain responses from semantic features. The semantic features included in the dataset consist of WordNet labels [Miller, 1995, Huth et al., 2012] that have been annotated manually for each two-second block of the movie clips. The voxelwise encoding model is fit with ridge regression, and cross-validation is used to select the optimal regularization hyperparameter. Model prediction accuracy is estimated on a separate test set, and projected onto a subject-specific flattened cortical surface. Finally, the regression weights are summarized using principal component analysis. The first few principal components are projected on a flattened cortical surface, and interpreted using graphs that represent the relations between the semantic features (Figure 3).

### Notebook 6: Visualize the hemodynamic response

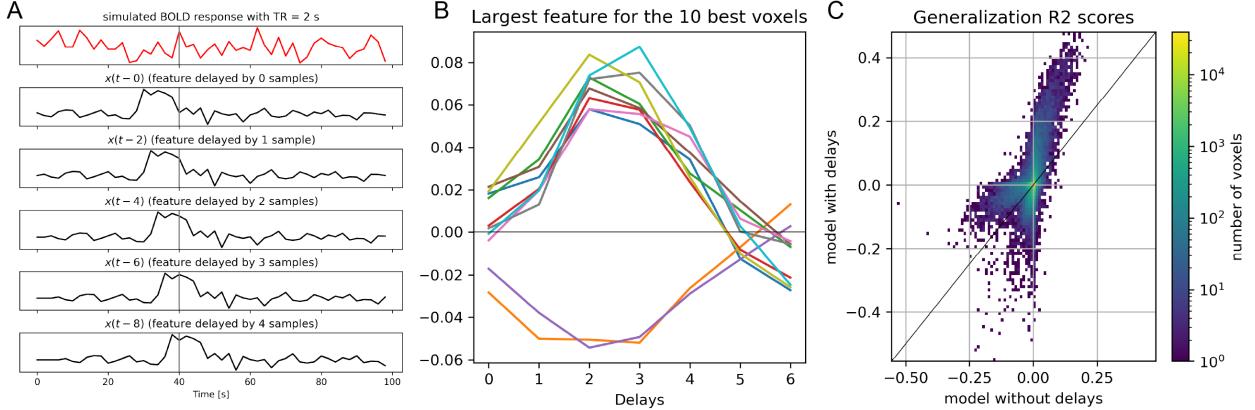
This notebook demonstrates how finite-impulse response (FIR) filters are used in voxelwise encoding models to take into account the delayed hemodynamic response in measured BOLD responses. With the FIR method, the features are duplicated multiple times with different temporal delays prior to model fitting. In this way, the encoding model estimates a separate weight per feature, per delay, and per voxel. This approach accounts for the variability of HRFs across voxels and brain areas, and it improves model prediction accuracy (Figure 4).



**Figure 2: Explainable variance.** Explainable variance is proportional to the maximum prediction accuracy that can be achieved by any voxelwise encoding model, given the available test data. For each voxel, explainable variance quantifies the variance in the measured signal that is repeatable. A) Measured brain responses to ten repetitions of the same test stimulus for a voxel with high explainable variance (top) and a voxel with low explainable variance (bottom). B) Histogram of explainable variance across the cortex of one participant. C) Explainable variance estimates projected onto the cortical surface of one participant. A voxelwise encoding model will be able to predict responses only in voxels with explainable variance greater than zero. Voxels with explainable variance equal or close to zero are either too noisy or not activated reliably by the experiment.



**Figure 3: Fitting a voxelwise encoding model with semantic features.** The tutorial *Fit a voxelwise encoding model with semantic features* shows how to estimate, evaluate, and interpret a voxelwise encoding model fit with the WordNet feature space. This tutorial replicates the analyses of the original publication in a single participant [Huth et al., 2012]. A) Model prediction accuracy on the held-out test set. The voxelwise encoding model based on semantic WordNet features predicts brain responses in category-specific high-level visual areas, but because low-level and high-level features are correlated, it also predicts response in motion-sensitive and early visual areas. The tutorial in Notebook 9 will show how to use banded ridge regression to disentangle low-level visual information from semantic information. B) The estimated model weights are interpreted according to the WordNet labels by performing principal component analysis (PCA), and then visualizing the model weight PCs on the WordNet graph. Each node in the graph is one of the WordNet labels that were used to tag the stimulus, and the edges of the graph correspond to the relationship “is a” [Huth et al., 2012]. The first PC (left) distinguishes objects and actions with motion (*gallop, travel, person*) from buildings and static objects (*structure, room, clothing*). The second, third, and fourth PCs (right) are visualized on the same WordNet graph by coloring each PC according to the red/green/blue channels. C) The model weights are projected onto the second, third, and fourth PC to interpret the semantic tuning of voxels across the cortex. For example, voxels in the fusiform face area (FFA) and in the extrastriate body area (EBA) are colored in red, purple, and yellow, corresponding to person-related, animal-related, and communication-related WordNet labels.



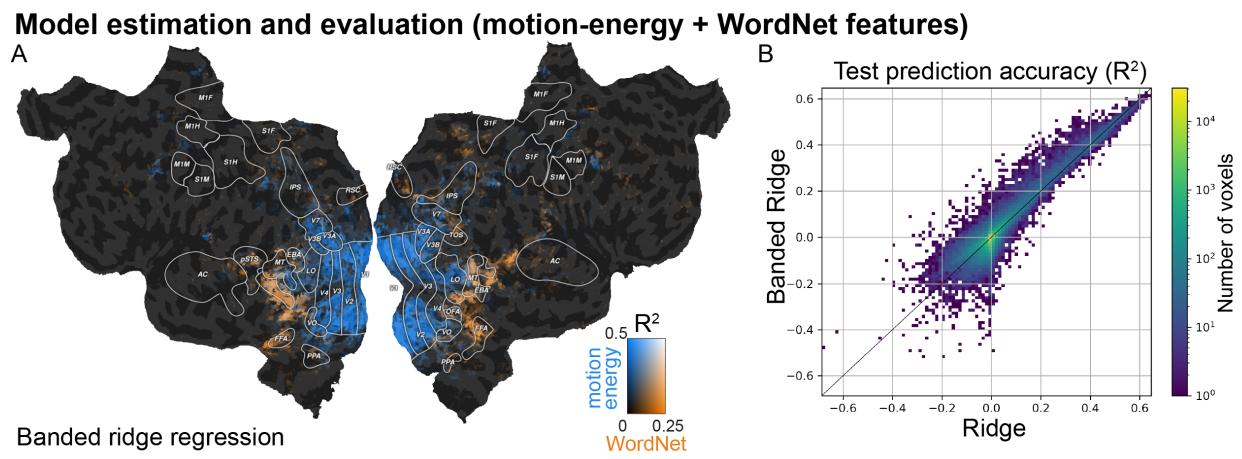
**Figure 4: Delaying features to account for the hemodynamic response function.** Brain responses measured by fMRI are delayed in time due to the dynamics underlying neurovascular coupling. To account for this delay, in the voxelwise modeling framework features are delayed and concatenated prior to model fitting. This procedure corresponds to using a finite impulse response (FIR) model to estimate the hemodynamic response function (HRF) in each voxel. A) Timecourse of simulated BOLD responses in a voxel (in red) and of a simulated feature (in black). To predict the BOLD response, the corresponding feature is delayed by 0, 1, 2, 3, and 4 samples, corresponding to 0, 2, 4, 6, and 8 s at a TR of 2 s. In this way, the response at time  $t = 40$  s is predicted by a linear combination of the features occurring at times 32, 34, 36, 38, and 40 s. B) The model weights associated with one feature are displayed for ten voxels. Using an FIR model allows recovering the shape of the HRF in each voxel. C) Two-dimensional histogram comparing the test prediction accuracy of a model without delays (x-axis) and a model with delays (y-axis). Using delays improves model prediction accuracy by accounting for the hemodynamic delay in the measured BOLD responses.

### Notebook 9: Fit a voxelwise encoding model with both semantic and motion-energy features

This notebook demonstrates how to fit a voxelwise encoding model with two feature spaces jointly: the semantic and the motion-energy feature spaces (see also Notebooks 5, 7, 8). Because the joint model uses two feature spaces, the voxelwise encoding model is fit with banded ridge regression, as presented in [Nunez-Elizalde et al., 2019]. Banded ridge regression adapts the regularization strength for each feature space in each voxel. This improves prediction accuracy by performing implicit feature-space selection [Dupré la Tour et al., 2022]. Then, to interpret the contribution of each feature space to the joint model, the joint prediction accuracy is decomposed between both feature spaces (Figure 5).

## Conclusion

These voxelwise modeling tutorials provide a solid foundation for creating encoding models with fMRI data. They provide practical examples on how to perform key aspects of voxelwise modeling, such as performing quality assurance by computing explainable variance, fitting and evaluating voxelwise encoding models with cross-validation and regularized regression, and using PCA to interpret model weights. However, these tutorials are not exhaustive. There are other advanced aspects of voxelwise modeling that are not yet included in these tutorials. These advanced topics include experimental design, fMRI data acquisition and preprocessing, and using the encoding model framework to analyze measurements made by other means such as neurophysiology or optical imaging [Wu et al., 2006]. Because these tutorials are publicly available and open source, they might be augmented in the future to provide more information about these advanced topics.



**Figure 5: Fitting a voxelwise encoding model with both semantic and motion-energy features.** The tutorial *Fit a voxelwise encoding model with both semantic and motion-energy features* shows how to use banded ridge regression to evaluate the contribution of different feature spaces in predicting brain responses. In banded ridge regression, for each voxel two regularization hyperparameters are optimized by performing cross-validation separately for each feature space. A) Split prediction accuracy on the test set for the motion-energy feature space (blue) and for the WordNet feature space (orange). Compared to the encoding model fit with only the semantic WordNet feature space (Figure 2), the banded ridge model with both feature spaces disentangles the contribution of low-level visual information on the model prediction from the contribution of semantic information. The WordNet feature space predicts only in category-specific visual areas, while the motion-energy feature space predicts only in early visual areas and motion-sensitive areas. B) The tutorial also demonstrates the advantage of using banded ridge regression instead of regular ridge regression with more than one feature space. The two-dimensional histogram shows the test prediction accuracy of two encoding models estimated with ridge (x-axis) and banded ridge (y-axis) using both the motion-energy and the WordNet feature spaces. The banded ridge regression model results in more accurate predictions than the ridge regression model.

## Data/code availability statement

The code of the VM tutorials is available at [https://github.com/gallantlab/voxelwise\\_tutorials](https://github.com/gallantlab/voxelwise_tutorials). The short-clips dataset [Nishimoto et al., 2011, Huth et al., 2012] used in the VM tutorials is available at <https://gin.g-node.org/gallantlab/shortclips> [Huth et al., 2022].

## Authors contributions

Conceptualization, TDIT; Software, TDIT and MVdOC; Writing - Original Draft, TDIT; Writing - Review & Editing, TDIT, MVdOC and JLG; Funding acquisition, JLG.

## Funding

This work was supported by grants from the Office of Naval Research (N00014-20-1-2002, N00014-15-1-2861, 60744755-114407-UCB, N00014-22-1-2217). The original data acquisition was supported by grants from the National Eye Institute (EY019684), and from the Center for Science of Information (CSOI), an NSF Science and Technology Center, under grant agreement CCF-0939370.

## Declaration of competing interests

The authors declare no competing financial interests.

## Acknowledgments

We thank Fatma Deniz, Mark Lescroart, and the other past and present members of the Gallant lab for fruitful discussions and comments.

## References

- [Abraham et al., 2014] 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*, page 14.
- [Ashby, 2019] Ashby, F. G. (2019). *Statistical analysis of fMRI data*. MIT press.
- [Bennett et al., 2009] Bennett, C. M., Miller, M. B., and Wolford, G. L. (2009). Neural correlates of interspecies perspective taking in the post-mortem atlantic salmon: An argument for multiple comparisons correction. *Neuroimage*, 47(Suppl 1):S125.
- [Bird and Loper, 2004] Bird, S. and Loper, E. (2004). Nltk: the natural language toolkit. Association for Computational Linguistics.
- [Brandl, 2010] Brandl, G. (2010). Sphinx documentation. URL <http://sphinx-doc.org/sphinx.pdf>.
- [Button et al., 2013] Button, K. S., Ioannidis, J., Mokrysz, C., Nosek, B. A., Flint, J., Robinson, E. S., and Munafò, M. R. (2013). Power failure: why small sample size undermines the reliability of neuroscience. *Nature reviews neuroscience*, 14(5):365–376.
- [Caucheteux and King, 2022] Caucheteux, C. and King, J.-R. (2022). Brains and algorithms partially converge in natural language processing. *Communications biology*, 5(1):1–10.
- [Chang et al., 2020a] Chang, L., Manning, J., Baldassano, C., de la Vega, A., Fleetwood, G., Geerligs, L., Haxby, J., Lahnakoski, J., Parkinson, C., Shappell, H., Shim, W. M., Wager, T., Yarkoni, T., Yeshurun, Y., and Finn, E. (2020a). naturalistic-data-analysis/naturalistic\_data\_analysis: Version 1.0.
- [Chang et al., 2020b] Chang, L. J., Huckins, J., Cheong, J. H., Brietzke, S., Lindquist, M. A., and Wager, T. D. (2020b). dartbrains: An online open access resource for learning functional neuroimaging analysis methods in Python.

- [Collaboration, 2015] Collaboration, O. S. (2015). Estimating the reproducibility of psychological science. *Science*, 349(6251):aac4716.
- [Cremers et al., 2017] Cremers, H. R., Wager, T. D., and Yarkoni, T. (2017). The relation between statistical power and inference in fMRI. *PloS one*, 12(11):e0184923.
- [Çukur et al., 2013] Çukur, T., Nishimoto, S., Huth, A. G., and Gallant, J. L. (2013). Attention during natural vision warps semantic representation across the human brain. *Nature neuroscience*, 16(6):763–770.
- [Dale, 1999] Dale, A. M. (1999). Optimal experimental design for event-related fMRI. *Human brain mapping*, 8(2–3):109–114.
- [de Heer et al., 2017] de Heer, W. A., Huth, A. G., Griffiths, T. L., Gallant, J. L., and Theunissen, F. E. (2017). The hierarchical cortical organization of human speech processing. *Journal of Neuroscience*, 37(27):6539–6557.
- [Deniz et al., 2019] Deniz, F., Nunez-Elizalde, A. O., Huth, A. G., and Gallant, J. L. (2019). The representation of semantic information across human cerebral cortex during listening versus reading is invariant to stimulus modality. *Journal of Neuroscience*, 39(39):7722–7736.
- [Deniz et al., 2023] Deniz, F., Tseng, C., Wehbe, L., la Tour, T. D., and Gallant, J. L. (2023). Semantic representations during language comprehension are affected by context. *Journal of Neuroscience*, 43(17):3144–3158.
- [Dubois and Adolphs, 2016] Dubois, J. and Adolphs, R. (2016). Building a science of individual differences from fMRI. *Trends in cognitive sciences*, 20(6):425–443.
- [Dumoulin and Wandell, 2008] Dumoulin, S. O. and Wandell, B. A. (2008). Population receptive field estimates in human visual cortex. *Neuroimage*, 39(2):647–660.
- [Dupré la Tour et al., 2021] Dupré la Tour, T., Lu, M., Eickenberg, M., and Gallant, J. L. (2021). A finer mapping of convolutional neural network layers to the visual cortex. In *SVRHM 2021 Workshop@ NeurIPS*.
- [Dupré la Tour et al., 2022] Dupré la Tour, T., Eickenberg, M., Nunez-Elizalde, A. O., and Gallant, J. L. (2022). Feature-space selection with banded ridge regression. *NeuroImage*, 264:119728.
- [Eickenberg et al., 2016] Eickenberg, M., Gramfort, A., Varoquaux, G., and Thirion, B. (2016). Seeing it all: Convolutional network layers map the function of the human visual system. *NeuroImage*, 152:184–194.
- [Friston et al., 1994] Friston, K. J., Holmes, A. P., Worsley, K. J., Poline, J.-P., Frith, C. D., and Frackowiak, R. S. (1994). Statistical parametric maps in functional imaging: a general linear approach. *Human brain mapping*, 2(4):189–210.
- [Gao et al., 2015] Gao, J. S., Huth, A. G., Lescroart, M. D., and Gallant, J. L. (2015). Pycortex: an interactive surface visualizer for fMRI. *Frontiers in neuroinformatics*, 9:23.
- [Gordon et al., 2017] Gordon, E. M., Laumann, T. O., Gilmore, A. W., Newbold, D. J., Greene, D. J., Berg, J. J., Ortega, M., Hoyt-Drazen, C., Gratton, C., Sun, H., et al. (2017). Precision functional mapping of individual human brains. *Neuron*, 95(4):791–807.
- [Goutte et al., 2000] Goutte, C., Nielsen, F. A., and Hansen, K. (2000). Modeling the hemodynamic response in fmri using smooth fir filters. *IEEE transactions on medical imaging*, 19(12):1188–1201.
- [Güçlü and van Gerven, 2015] Güçlü, U. and van Gerven, M. A. (2015). Deep neural networks reveal a gradient in the complexity of neural representations across the ventral stream. *Journal of Neuroscience*, 35(27):10005–10014.
- [Hagberg et al., 2008] Hagberg, A., Swart, P., and S Chult, D. (2008). Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States).
- [Halchenko et al., 2021] Halchenko, Y. O., Meyer, K., Poldrack, B., Solanki, D. S., Wagner, A. S., Gors, J., MacFarlane, D., Pustina, D., Sochat, V., Ghosh, S. S., Mönch, C., Markiewicz, C. J., Waite, L., Shlyakhter, I., de la Vega, A., Hayashi, S., Häusler, C. O., Poline, J.-B., Kadelka, T., Skytén, K., Jarecka, D., Kennedy, D., Strauss, T., Cieslak, M., Vavra, P., Ioanas, H.-I., Schneider, R., Pfüger, M., Haxby, J. V., Eickhoff, S. B., and Hanke, M. (2021). DataLad: distributed system for joint management of code, data, and their relationship. *Journal of Open Source Software*, 6(63):3262.
- [Hamilton and Huth, 2020] Hamilton, L. S. and Huth, A. G. (2020). The revolution will not be controlled: natural stimuli in speech neuroscience. *Language, cognition and neuroscience*, 35(5):573–582.
- [Hanke et al., 2009] Hanke, M., Halchenko, Y. O., Sederberg, P. B., Hanson, S. J., Haxby, J. V., and Pollmann, S. (2009). PyMVPA: A python toolbox for multivariate pattern analysis of fMRI data. *Neuroinformatics*, 7(1):37–53.
- [Hansen et al., 2004] Hansen, K. A., David, S. V., and Gallant, J. L. (2004). Parametric reverse correlation reveals spatial linearity of retinotopic human v1 bold response. *Neuroimage*, 23(1):233–241.

- [Harris et al., 2020] Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., et al. (2020). Array programming with numpy. *Nature*, 585(7825):357–362.
- [Hasson et al., 2010] Hasson, U., Malach, R., and Heeger, D. J. (2010). Reliability of cortical activity during natural stimulation. *Trends in cognitive sciences*, 14(1):40–48.
- [Haxby et al., 2014] Haxby, J. V., Connolly, A. C., Guntupalli, J. S., et al. (2014). Decoding neural representational spaces using multivariate pattern analysis. *Annual review of neuroscience*, 37(1):435–456.
- [Hoerl and Kennard, 1970] Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67.
- [Hsu et al., 2004] Hsu, A., Borst, A., and Theunissen, F. E. (2004). Quantifying variability in neural responses and its application for the validation of model predictions. *Network: Computation in Neural Systems*, 15(2):91–109.
- [Hunter, 2007] Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in science & engineering*, 9(3):90–95.
- [Huth et al., 2016] Huth, A. G., De Heer, W. A., Griffiths, T. L., Theunissen, F. E., and Gallant, J. L. (2016). Natural speech reveals the semantic maps that tile human cerebral cortex. *Nature*, 532(7600):453–458.
- [Huth et al., 2022] Huth, A. G., Nishimoto, S., Vu, A. T., Dupré la Tour, T., and Gallant, J. L. (2022). Gallant lab natural short clips 3T fMRI data. *G-Node*.
- [Huth et al., 2012] Huth, A. G., Nishimoto, S., Vu, A. T., and Gallant, J. L. (2012). A continuous semantic space describes the representation of thousands of object and action categories across the human brain. *Neuron*, 76(6):1210–1224.
- [Jain and Huth, 2018] Jain, S. and Huth, A. (2018). Incorporating context into language encoding models for fMRI. In *Advances in neural information processing systems*, pages 6628–6637.
- [Jain et al., 2020] Jain, S., Vo, V., Mahto, S., LeBel, A., Turek, J. S., and Huth, A. (2020). Interpretable multi-timescale models for predicting fmri responses to continuous natural speech. *Advances in Neural Information Processing Systems*, 33:13738–13749.
- [Kay et al., 2008] 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.
- [Kell et al., 2018] 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.
- [Khosla et al., 2021] Khosla, M., Ngo, G. H., Jamison, K., Kuceyeski, A., and Sabuncu, M. R. (2021). Cortical response to naturalistic stimuli is largely predictable with deep neural networks. *Science Advances*, 7(22):eabe7547.
- [Kluyver et al., 2016] Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S., and Willing, C. (2016). Jupyter Notebooks – a publishing format for reproducible computational workflows. In Loizides, F. and Schmidt, B., editors, *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, pages 87 – 90. IOS Press.
- [Konkle and Alvarez, 2022] Konkle, T. and Alvarez, G. A. (2022). A self-supervised domain-general learning framework for human ventral stream representation. *Nature communications*, 13(1):1–12.
- [Kriegeskorte et al., 2008] Kriegeskorte, N., Mur, M., and Bandettini, P. A. (2008). Representational similarity analysis-connecting the branches of systems neuroscience. *Frontiers in systems neuroscience*, page 4.
- [Kumar et al., 2020] Kumar, M., Ellis, C. T., Lu, Q., Zhang, H., Capotă, M., Willke, T. L., Ramadge, P. J., Turk-Browne, N. B., and Norman, K. A. (2020). BrainIAK tutorials: User-friendly learning materials for advanced fMRI analysis. *PLoS computational biology*, 16(1):e1007549.
- [LeBel et al., 2021] LeBel, A., Jain, S., and Huth, A. G. (2021). Voxelwise encoding models show that cerebellar language representations are highly conceptual. *Journal of Neuroscience*, 41(50):10341–10355.
- [Lescroart and Gallant, 2019] Lescroart, M. D. and Gallant, J. L. (2019). Human scene-selective areas represent 3d configurations of surfaces. *Neuron*, 101(1):178–192.
- [Lescroart et al., 2015] Lescroart, M. D., Stansbury, D. E., and Gallant, J. L. (2015). Fourier power, subjective distance, and object categories all provide plausible models of bold responses in scene-selective visual areas. *Frontiers in computational neuroscience*, 9:135.
- [Matusz et al., 2019] Matusz, P. J., Dikker, S., Huth, A. G., and Perrodin, C. (2019). Are we ready for real-world neuroscience?

- [Miller, 1995] Miller, G. A. (1995). Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.
- [Mitchell et al., 2008] Mitchell, T. M., Shinkareva, S. V., Carlson, A., Chang, K.-M., Malave, V. L., Mason, R. A., and Just, M. A. (2008). Predicting human brain activity associated with the meanings of nouns. *Science*, 320(5880):1191–1195.
- [Naselaris et al., 2011] Naselaris, T., Kay, K. N., Nishimoto, S., and Gallant, J. L. (2011). Encoding and decoding in fMRI. *Neuroimage*, 56(2):400–410.
- [Naselaris et al., 2009] Naselaris, T., Prenger, R. J., Kay, K. N., Oliver, M., and Gallant, J. L. (2009). Bayesian reconstruction of natural images from human brain activity. *Neuron*, 63(6):902–915.
- [Nishimoto et al., 2011] Nishimoto, S., Vu, A. T., Naselaris, T., Benjamini, Y., Yu, B., and Gallant, J. L. (2011). Reconstructing visual experiences from brain activity evoked by natural movies. *Current Biology*, 21(19):1641–1646.
- [Nishino and Loomis, 2017] Nishino, R. and Loomis, S. H. C. (2017). Cupy: A numpy-compatible library for nvidia gpu calculations. *31st conference on neural information processing systems*, page 151.
- [Nonaka et al., 2021] Nonaka, S., Majima, K., Aoki, S. C., and Kamitani, Y. (2021). Brain hierarchy score: Which deep neural networks are hierarchically brain-like? *IScience*, 24(9):103013.
- [Nunez-Elizalde et al., 2021] Nunez-Elizalde, A. O., Deniz, F., Dupré la Tour, T., Visconti di Oleggio Castello, M., and Gallant, J. L. (2021). Pymoten: scientific Python package for computing motion energy features from video. *Zenodo*.
- [Nunez-Elizalde et al., 2019] Nunez-Elizalde, A. O., Huth, A. G., and Gallant, J. L. (2019). Voxelwise encoding models with non-spherical multivariate normal priors. *NeuroImage*, 197:482–492.
- [Nájera et al., 2020] Nájera, O., Larson, E., Estève, L., Varoquaux, G., Liu, L., Grobler, J., de Andrade, E. S., Holdgraf, C., Gramfort, A., Jas, M., Nothman, J., Grisel, O., Varoquaux, N., Gouillart, E., Luessi, M., Lee, A., Vanderplas, J., Hoffmann, T., Caswell, T. A., Sullivan, B., Batula, A., jaeilepp, Robitaille, T., Appelhoff, S., Kunzmann, P., Geier, M., Lars, Sundén, K., Stańczak, D., and Shih, A. Y. (2020). sphinx-gallery/sphinx-gallery: Release v0.7.0. Technical report.
- [Paszke et al., 2019] 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 *Advances in Neural Information Processing Systems*, pages 8024–8035.
- [Pedregosa et al., 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.
- [Penny et al., 2011] Penny, W. D., Friston, K. J., Ashburner, J. T., Kiebel, S. J., and Nichols, T. E. (2011). *Statistical parametric mapping: the analysis of functional brain images*. Elsevier.
- [Poline and Brett, 2012] Poline, J.-B. and Brett, M. (2012). The general linear model and fMRI: does love last forever? *Neuroimage*, 62(2):871–880.
- [Popham et al., 2021] Popham, S. F., Huth, A. G., Bilenko, N. Y., Deniz, F., Gao, J. S., Nunez-Elizalde, A. O., and Gallant, J. L. (2021). Visual and linguistic semantic representations are aligned at the border of human visual cortex. *Nature neuroscience*, 24(11):1628–1636.
- [Sahani and Linden, 2003] Sahani, M. and Linden, J. (2003). How linear are auditory cortical responses. *Advances in neural information processing systems*, 15:301–308.
- [Schönwiesner and Zatorre, 2009] Schönwiesner, M. and Zatorre, R. J. (2009). Spectro-temporal modulation transfer function of single voxels in the human auditory cortex measured with high-resolution fmri. *Proceedings of the National Academy of Sciences*, 106(34):14611–14616.
- [Schoppe et al., 2016] Schoppe, O., Harper, N. S., Willmore, B. D., King, A. J., and Schnupp, J. W. (2016). Measuring the performance of neural models. *Frontiers in computational neuroscience*, 10:10.
- [St-Yves and Naselaris, 2018] St-Yves, G. and Naselaris, T. (2018). The feature-weighted receptive field: an interpretable encoding model for complex feature spaces. *NeuroImage*, 180:188–202.
- [Stansbury et al., 2013] Stansbury, D. E., Naselaris, T., and Gallant, J. L. (2013). Natural scene statistics account for the representation of scene categories in human visual cortex. *Neuron*, 79(5):1025–1034.
- [Thirion et al., 2006] Thirion, B., Duchesnay, E., Hubbard, E., Dubois, J., Poline, J.-B., Lebihan, D., and Dehaene, S. (2006). Inverse retinotopy: inferring the visual content of images from brain activation patterns. *NeuroImage*, 33(4):1104–1116.

- [Tomassini et al., 2011] Tomassini, V., Jbabdi, S., Kincses, Z. T., Bosnell, R., Douaud, G., Pozzilli, C., Matthews, P. M., and Johansen-Berg, H. (2011). Structural and functional bases for individual differences in motor learning. *Human brain mapping*, 32(3):494–508.
- [Toneva and Wehbe, 2019] Toneva, M. and Wehbe, L. (2019). Interpreting and improving natural-language processing (in machines) with natural language- processing (in the brain). In *Advances in Neural Information Processing Systems*, pages 14928–14938.
- [Valizadeh et al., 2018] Valizadeh, S. A., Liem, F., Mérillat, S., Hänggi, J., and Jäncke, L. (2018). Identification of individual subjects on the basis of their brain anatomical features. *Scientific reports*, 8(1):1–9.
- [Virtanen et al., 2020] Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., et al. (2020). Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272.
- [Visconti di Oleggio Castello et al., 2024] Visconti di Oleggio Castello, M., Deniz, F., Dupré la Tour, T., and Gallant, J. L. (2024). Encoding models in functional MRI: the Voxelwise Modeling framework. *In preparation*.
- [Wehbe et al., 2014] Wehbe, L., Murphy, B., Talukdar, P., Fyshe, A., Ramdas, A., and Mitchell, T. (2014). Simultaneously uncovering the patterns of brain regions involved in different story reading subprocesses. *PloS one*, 9(11).
- [Wen et al., 2018] Wen, H., Shi, J., Zhang, Y., Lu, K.-H., Cao, J., and Liu, Z. (2018). Neural encoding and decoding with deep learning for dynamic natural vision. *Cerebral Cortex*, 28(12):4136–4160.
- [Wu et al., 2006] 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.
- [Yarkoni, 2022] Yarkoni, T. (2022). The generalizability crisis. *Behavioral and Brain Sciences*, 45.
- [Yarkoni and Westfall, 2017] Yarkoni, T. and Westfall, J. (2017). Choosing prediction over explanation in psychology: Lessons from machine learning. *Perspectives on Psychological Science*, 12(6):1100–1122.